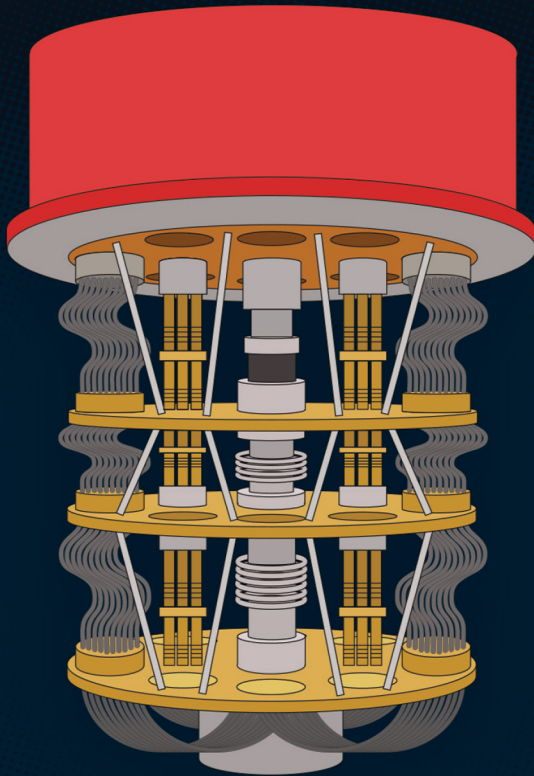


Principles of Superconducting Quantum Computers



Daniel D. Stancil | Gregory T. Byrd



WILEY

Principles of Superconducting Quantum Computers

Principles of Superconducting Quantum Computers

Daniel D. Stancil
North Carolina State University
Raleigh, North Carolina

Gregory T. Byrd
North Carolina State University
Raleigh, North Carolina

WILEY

This edition first published 2022
© 2022 by John Wiley & Sons, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by law. Advice on how to obtain permission to reuse material from this title is available at <http://www.wiley.com/go/permissions>.

The right of Daniel D. Stancil and Gregory T. Byrd to be identified as the authors of this work has been asserted in accordance with law.

Registered Office

John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA

Editorial Office

111 River Street, Hoboken, NJ 07030, USA

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Limit of Liability/Disclaimer of Warranty

The contents of this work are intended to further general scientific research, understanding, and discussion only and are not intended and should not be relied upon as recommending or promoting scientific method, diagnosis, or treatment by physicians for any particular patient. In view of ongoing research, equipment modifications, changes in governmental regulations, and the constant flow of information relating to the use of medicines, equipment, and devices, the reader is urged to review and evaluate the information provided in the package insert or instructions for each medicine, equipment, or device for, among other things, any changes in the instructions or indication of usage and for added warnings and precautions. While the publisher and authors have used their best efforts in preparing this work, they make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

A catalogue record for this book is available from the Library of Congress

Hardback ISBN: 9781119750727; ePub ISBN: 9781119750741; ePDF ISBN: 9781119750734;
Obook ISBN: 9781119750758

Cover image: © Adaptrographics/Getty Images
Cover design by Wiley

Set in 9.5/12.5pt STIXTwoText by Integra Software Services Pvt. Ltd, Pondicherry, India

10 9 8 7 6 5 4 3 2 1

Dedicated to the pioneers of the first Quantum Revolution, who paved the way.

Contents

List of Figures	<i>xiii</i>
List of Tables	<i>xxv</i>
Preface	<i>xxvii</i>
Acknowledgments	<i>xxix</i>
About the Companion Website	<i>xxxix</i>

1	Qubits, Gates, and Circuits	1
1.1	Bits and Qubits	1
1.1.1	Circuits in Space vs. Circuits in Time	1
1.1.2	Superposition	1
1.1.3	No Cloning	3
1.1.4	Reversibility	3
1.1.5	Entanglement	3
1.2	Single-Qubit States	4
1.3	Measurement and the Born Rule	5
1.4	Unitary Operations and Single-Qubit Gates	6
1.5	Two-Qubit Gates	8
1.5.1	Two-Qubit States	8
1.5.2	Matrix Representation of Two-Qubit Gates	9
1.5.3	Controlled-NOT	11
1.6	Bell State	12
1.7	No Cloning, Revisited	13
1.8	Example: Deutsch's Problem	15
1.9	Key Characteristics of Quantum Computing	18
1.10	Quantum Computing Systems	18
1.11	Exercises	22
2	Physics of Single Qubit Gates	25
2.1	Requirements for a Quantum Computer	25
2.2	Single Qubit Gates	25
2.2.1	Rotations	25
2.2.2	Two State Systems	33
2.2.3	Creating Rotations: Rabi Oscillations	38
2.3	Quantum State Tomography	42
2.4	Expectation Values and the Pauli Operators	44

2.5	Density Matrix	45
2.6	Exercises	48
3	Physics of Two Qubit Gates	51
3.1	$\sqrt{i}$ SWAP Gate	51
3.2	Coupled Tunable Qubits	53
3.3	Cross Resonance Scheme	55
3.4	Other Controlled Gates	57
3.5	Two-Qubit States and the Density Matrix	59
3.6	Exercises	62
4	Superconducting Quantum Computer Systems	63
4.1	Transmission Lines	63
4.1.1	General Transmission Line Equations	63
4.1.2	Lossless Transmission Lines	65
4.1.3	Transmission Lines with Loss	67
4.2	Terminated Lossless Line	71
4.2.1	Reflection Coefficient	71
4.2.2	Power (Flow of Energy) and Return Loss	72
4.2.3	Standing Wave Ratio (SWR)	73
4.2.4	Impedance as a Function of Position	74
4.2.5	Quarter Wave Transformer	76
4.2.6	Coaxial, Microstrip, and Coplanar Lines	77
4.3	<i>S</i> Parameters	80
4.3.1	Lossless Condition	81
4.3.2	Reciprocity	81
4.4	Transmission (ABCD) Matrices	81
4.5	Attenuators	85
4.6	Circulators and Isolators	87
4.7	Power Dividers/Combiners	89
4.8	Mixers	92
4.9	Low-Pass Filters	95
4.10	Noise	97
4.10.1	Thermal Noise	97
4.10.2	Equivalent Noise Temperature	99
4.10.3	Noise Factor and Noise Figure	100
4.10.4	Attenuators and Noise	101
4.10.5	Noise in Cascaded Systems	103
4.11	Low Noise Amplifiers	104
4.12	Exercises	105
5	Resonators: Classical Treatment	107
5.1	Parallel Lumped Element Resonator	107
5.2	Capacitive Coupling to a Parallel Lumped-Element Resonator	109
5.3	Transmission Line Resonator	111
5.4	Capacitive Coupling to a Transmission Line Resonator	113
5.5	Capacitively-Coupled Lossless Resonators	117

5.6	Classical Model of Qubit Readout	120
5.7	Exercises	124
6	Resonators: Quantum Treatment	127
6.1	Lagrangian Mechanics	127
6.1.1	Hamilton's Principle	127
6.1.2	Calculus of Variations	128
6.1.3	Lagrangian Equation of Motion	129
6.2	Hamiltonian Mechanics	130
6.3	Harmonic Oscillators	131
6.3.1	Classical Harmonic Oscillator	131
6.3.2	Quantum Mechanical Harmonic Oscillator	133
6.3.3	Raising and Lowering Operators	135
6.3.4	Can a Harmonic Oscillator Be Used as a Qubit?	137
6.4	Circuit Quantum Electrodynamics	138
6.4.1	Classical LC Resonant Circuit	138
6.4.2	Quantization of the LC Circuit	139
6.4.3	Circuit Electrodynamics Approach for General Circuits	140
6.4.4	Circuit Model for Transmission Line Resonator	141
6.4.5	Quantizing a Transmission Line Resonator	144
6.4.6	Quantized Coupled LC Resonant Circuits	144
6.4.7	Schrödinger, Heisenberg, and Interaction Pictures	147
6.4.8	Resonant Circuits and Qubits	150
6.4.9	The Dispersive Regime	153
6.5	Exercises	156
7	Theory of Superconductivity	159
7.1	Bosons and Fermions	159
7.2	Bloch Theorem	161
7.3	Free Electron Model for Metals	163
7.3.1	Discrete States in Finite Samples	163
7.3.2	Phonons	166
7.3.3	Debye Model	167
7.3.4	Electron–Phonon Scattering and Electrical Conductivity	168
7.3.5	Perfect Conductor vs. Superconductor	170
7.4	Bardeen, Cooper, and Schrieffer Theory of Superconductivity	172
7.4.1	Cooper Pair Model	172
7.4.2	Dielectric Function	175
7.4.3	Jellium	176
7.4.4	Scattering Amplitude and Attractive Electron–Electron Interaction	179
7.4.5	Interpretation of Attractive Interaction	180
7.4.6	Superconductor Hamiltonian	181
7.4.7	Superconducting Ground State	182
7.5	Electrodynamics of Superconductors	185
7.5.1	Cooper Pairs and the Macroscopic Wave Function	185
7.5.2	Potential Functions	186
7.5.3	London Equations	187
7.5.4	London Gauge	189

7.5.5	Penetration Depth	190
7.5.6	Flux Quantization	191
7.6	Chapter Summary	192
7.7	Exercises	193
8	Josephson Junctions	195
8.1	Tunneling	195
8.1.1	Reflection from a Barrier	196
8.1.2	Finite Thickness Barrier	198
8.2	Josephson Junctions	200
8.2.1	Current and Voltage Relations	200
8.2.2	Josephson Junction Hamiltonian	203
8.2.3	Quantized Josephson Junction Analysis	205
8.3	Superconducting Quantum Interference Devices (SQUIDs)	207
8.4	Josephson Junction Parametric Amplifiers	208
8.5	Exercises	209
9	Errors and Error Mitigation	211
9.1	NISQ Processors	211
9.2	Decoherence	212
9.3	State Preparation and Measurement Errors	214
9.4	Characterizing Gate Errors	215
9.5	State Leakage and Suppression Using Pulse Shaping	219
9.6	Zero-Noise Extrapolation	220
9.7	Optimized Control Using Deep Learning	223
9.8	Exercises	225
10	Quantum Error Correction	227
10.1	Review of Classical Error Correction	227
10.1.1	Error Detection	228
10.1.2	Error Correction: Repetition Code	228
10.1.3	Hamming Code	229
10.2	Quantum Errors	230
10.3	Detecting and Correcting Quantum Errors	232
10.3.1	Bit Flip	232
10.3.2	Phase Flip	234
10.3.3	Correcting Bit and Phase Flips: Shor's 9-Qubit Code	235
10.3.4	Arbitrary Rotations	236
10.4	Stabilizer Codes	238
10.4.1	Stabilizers	238
10.4.2	Stabilizers for Error Correction	239
10.5	Operating on Logical Qubits	242
10.6	Error Thresholds	243
10.6.1	Concatenation of Error Codes	243
10.6.2	Threshold Theorem	244
10.7	Surface Codes	245
10.7.1	Stabilizers	246

10.7.2	Error Detection and Correction	247
10.7.3	Logical X and Z Operators	250
10.7.4	Multiple Qubits: Lattice Surgery	253
10.7.5	CNOT	257
10.7.6	Single-Qubit Gates	258
10.8	Summary and Further Reading	259
10.9	Exercises	261
11	Quantum Logic: Efficient Implementation of Classical Computations	263
11.1	Reversible Logic	264
11.1.1	Reversible Logic Gates	264
11.1.2	Reversible Logic Circuits	266
11.2	Quantum Logic Circuits	268
11.2.1	Entanglement and Uncomputing	269
11.2.2	Multi-Qubit Gates	270
11.2.3	Qubit Topology	270
11.3	Efficient Arithmetic Circuits: Adder	272
11.3.1	Quantum Ripple-Carry Adder	273
11.3.2	In-Place Ripple-Carry Adder	275
11.3.3	Carry-Lookahead Adder	277
11.3.4	Adder Comparison	281
11.4	Phase Logic	283
11.4.1	Controlled-Z and Controlled-Phase Gates	283
11.4.2	Selective Phase Change	285
11.4.3	Phase Logic Gates	287
11.5	Summary and Further Reading	288
11.6	Exercises	289
12	Some Quantum Algorithms	291
12.1	Computational Complexity	291
12.1.1	Quantum Program Run-Time	292
12.1.2	Classical Complexity Classes	292
12.1.3	Quantum Complexity	293
12.2	Grover's Search Algorithm	294
12.2.1	Grover Iteration	294
12.2.2	Quantum Implementation	296
12.2.3	Generalizations	299
12.3	Quantum Fourier Transform	299
12.3.1	Discrete Fourier Transform	300
12.3.2	Inverse Discrete Fourier Transform	300
12.3.3	Quantum Implementation of the DFT	301
12.3.4	Encoding Quantum States	302
12.3.5	Quantum Implementation	304
12.3.6	Computational Complexity	306
12.4	Quantum Phase Estimation	307
12.4.1	Quantum Implementation	307
12.4.2	Computational Complexity and Other Issues	308
12.5	Shor's Algorithm	309

12.5.1	Hybrid Classical-Quantum Algorithm	309
12.5.2	Finding the Period	310
12.5.3	Computational Complexity	314
12.6	Variational Quantum Algorithms	314
12.6.1	Variational Quantum Eigensolver	316
12.6.2	Quantum Approximate Optimization Algorithm	320
12.6.3	Challenges and Opportunities	323
12.7	Summary and Further Reading	324
12.8	Exercises	325

Bibliography	327
---------------------	-----

Index	339
--------------	-----

List of Figures

1.1 Interpretation of classical versus quantum NOT gates.	2
1.2 NAND circuit diagram.	4
1.3 Circuit representation of Eq. (1.31). In a quantum circuit diagram, the operation goes from left to right, while the matrix expression is shown going from right to left. The final box is a measurement in the standard basis, resulting in a classical bit.	8
1.4 Symbol for a CNOT gate, and its effect on basis states.	11
1.5 Circuit for creating an entangled state known as a Bell State. When the two qubits are measured, they will either both be 0, or they will both be 1.	12
1.6 Result of executing the circuit 1024 times on a quantum simulator, compared with executing the circuit 1024 times on a real IBM quantum computer.	13
1.7 Hypothetical cloning operator, that creates an exact and independent copy of unknown quantum state $ \alpha\rangle$. The text will show that such an operator cannot be implemented.	14
1.8 Conceptual illustration of the Deutsch Problem.	15
1.9 Reversible circuit for calculating $f(x)$.	16
1.10 Implementations of black-box function U_f for Deutsch's problem. Top output is $ y \oplus f(x)\rangle$, and bottom output is $ x\rangle$.	17
1.11 Implementation of Deutsch's algorithm. The dashed box is equivalent to U in Figure 1.8.	17
1.12 System diagram for a superconducting quantum computer.	20
2.1 Rotation of a vector of length r CCW around the z axis.	26
2.2 Illustration of how two consecutive rotations can be replaced with a single equivalent rotation.	29
2.3 Representation of a single qubit state on the Bloch Sphere (created in part using [2]).	31
2.4 Precession of spin vector for a particle with positive charge in a z -directed magnetic field.	36
2.5 Solutions to the coupled mode equations for $\Delta = 0$ and $\Delta/\kappa = 3$. For the case of Rabi oscillations, $ \kappa t = \Omega_R t/2$.	42
2.6 Rotations enabling measurement of the projections of the state vector along the x and y axes. (a) The Hadamard gate corresponds to rotation of the state vector around an axis in the x - z plane making a 45° angle with	

the z axis. This rotates the x component to z axis. (b) z rotation of -90° followed by a Hadamard rotation to estimate the projection along y .	43
2.7 (a) Bloch sphere representation of a mixed state, and (b) the result of applying a Hadamard gate (H) to the mixed state. Plots generated using Qiskit [2].	47
3.1 Common symbols for the SWAP, i SWAP, and $\sqrt{i$ SWAP two-qubit gates.	52
3.2 Operations needed to convert a $\sqrt{i$ SWAP' gate to a CNOT.	53
3.3 Controlled- U gate. If the control qubit is $ 1\rangle$, the U gate is applied to the target qubit. Otherwise, the gate is not applied.	57
3.4 Implementation of controlled- U gate using Eq. (3.39).	58
4.1 Ladder line used for radio frequency transmission.	64
4.2 Equivalent circuit for a transmission line. (a) Lumped elements can be used so long as the distance Δz is small compared to the distance traveled during a period of the signal. (b) Single section along the line for analysis.	64
4.3 A transmission line terminated with a load impedance. Note that two coordinate systems have their origins at the load location: the coordinate z increases to the right, and the coordinate l increases to the left.	72
4.4 Voltage standing wave pattern along a transmission line with a mismatched load.	75
4.5 Impedance looking into a terminated transmission line of length l .	75
4.6 A real impedance can be matched to a lossless transmission line using a quarter wavelength line whose impedance is the geometric mean of the load and line to be matched. This is referred to as a quarter wave matching transformer.	76
4.7 Some commonly-used types of transmission lines. Since there are multiple ways to drive the coplanar guide, the most common source connection is explicitly shown. All of the dielectric materials are assumed to be nonmagnetic for our purposes.	78
4.8 Incoming and outgoing wave amplitudes used in the definitions of S parameters.	80
4.9 Definition of voltages and currents for the ABCD transmission matrix.	82
4.10 Definitions for constructing the ABCD matrix of a section of transmission line.	82
4.11 Circuit for an attenuator.	86
4.12 Circulator Circuit diagrams. In actual devices, the three ports are symmetrically placed at 120° angles, as represented on the left. However, in drawing circuit diagrams, it is often convenient to show the ports at right angles, as shown in the center. As shown on the right, you can make an isolator by connecting one of the ports to a matched load.	88
4.13 Wilkinson power divider.	89
4.14 Quadrature hybrid 4-port network. For clarity each transmission line is represented by a single line, with the return conductors understood (e.g., an implied ground plane).	90
4.15 Even and odd mode analysis of a quadrature hybrid coupler. All impedances are shown normalized to Z_0 .	90

4.16	Commonly-used symbol for a quadrature hybrid. A signal applied to port 1 is evenly split into quadrature signals at ports 2 and 3.	92
4.17	Mixer Circuit diagrams. In an ideal mixer, the signal at the output terminal is the product of the signals applied to the two input ports.	92
4.18	(a) Circuit to shape a microwave pulse. (b) Example of creating a pulse with a Gaussian pulse shape.	93
4.19	(a) Circuit to recover the cosine (in-phase, or I) and sine (quadrature, or Q) components of an RF signal. (b) The amplitude and phase plotted on the IQ plane. (c) Raw I-Q signals measured on ibm-q-armonk, a 1 qubit demonstration processor on the IBM Q Network. The means for each state are indicated by the large markers.	94
4.20	Low-pass filter circuits.	95
4.21	Frequency response of the T network low-pass filter.	96
4.22	Circuit illustrating thermal noise power from a resistor at temperature T . (a) Resistor at temperature T coupled to a load through a lossless bandpass filter with bandwidth B . (b) Equivalent circuit explicitly showing a noise voltage source that depends on temperature.	97
4.23	Quantum noise as a function of temperature normalized to the photon energy. Solid line is the exact expression (4.161), while the dashed line is the Rayleigh-Jeans approximation (4.160).	99
4.24	Noise added by a circuit with power gain G . For an amplifier, $G > 1$, while for an attenuator $G < 1$.	100
4.25	Thermal noise from a passive element such as an attenuator at temperature T . (a) Power applied to port 2 is partially reflected and partially transmitted to the output, with the balance dissipated as heat. The power dissipated is visualized as being conveyed to a fictitious port 3 connected to a thermal reservoir at temperature T . (b) The total power out has a contribution from the input as well as the thermal reservoir attached to the fictitious port 3.	102
4.26	Noise in a system of cascaded components.	103
4.27	Layered structure of different semiconductor materials used to separate the donor impurities from the donor electrons to achieve minimal scattering. Structures of different materials like this are referred to as <i>heterostructures</i> . Owing to the different band structure of AlGaAs and GaAs, donor electrons from the heavily-doped AlGaAs layer become trapped on the GaAs side of the interface between intrinsic GaAs and AlGaAs, forming a “2D electron gas.” This concentration of electrons is depicted by the shaded region. A heterostructure similar to this is a key feature of HEMTs.	105
5.1	Resonator circuits.	108
5.2	Equivalent circuits for capacitively-coupled lumped-element resonator.	110
5.3	Capacitively-coupled transmission line resonator.	113
5.4	Near the n th resonant frequency of the transmission line, the capacitively-coupled transmission line resonator can be modeled as a capacitively-coupled lumped-element resonator.	114
5.5	Equivalent circuits used to calculate the insertion loss and return loss of a capacitively-coupled transmission line resonator.	115

5.6 Characteristics of capacitively-coupled transmission line resonator. Note the significantly different frequency scales on the horizontal axis as well as the different vertical scales for $ S_{11} $. Parameters used are $L_n = 0.453 \mu\text{H/m}$, $Z_c = R_L = 50 \Omega$, $f_{0=5} \text{ GHz}$, $Q_{\text{int}} = 2.3 \times 10^5$, and $l = 11.04 \text{ mm}$. These values are comparable to experimentally-measured parameters. Based on [26].	116
5.7 Two LC resonant circuits coupled by a capacitor.	117
5.8 Coupling between lossless LC resonators for $C_g/\sqrt{C_A C_B} = 0.05$.	120
5.9 Tire swings suspended from a common support rope have similar behavior to coupled LC resonators.	122
5.10 Geometry of a transmon qubit coupled to a transmission line resonator.	122
5.11 Equivalent circuits for transmission line resonator coupled to a transmon qubit.	123
5.12 Magnitude and phase of the reflected signals from the resonator in the dispersive limit. Parameters used in the calculations are: $Z_c = 50 \text{ Ohms}$, $f_q = \omega_q/2\pi = 5 \text{ GHz}$, $f_0 = \omega_0/2\pi = 7 \text{ GHz}$, $Q_0 = 2.3 \times 10^5$, $C_{12} = 0.5 \text{ fF}$, $C_{23} = C_{45} = C_{47} = C_{12}/2$, $\kappa = 7 \text{ MHz}$.	125
6.1 Classical harmonic oscillator consisting of a mass m attached to an anchored spring with force constant K .	132
6.2 Quantum mechanical harmonic oscillator (a) and perturbed harmonic oscillator (b). The wave functions are plotted in arbitrary amplitude units at the associated energy level.	135
6.3 LC resonant circuit.	138
6.4 Lumped circuit model for an open-circuited section of transmission line of length $N\Delta z = \ell$.	141
6.5 Capacitively-coupled LC resonant circuits.	145
7.1 The Fermi-Dirac distribution gives the probability that a Fermion state with a given energy is occupied at a specified temperature. The lower the temperature, the sharper the transition from occupied to not-occupied at $\mathcal{E} = \mu$.	160
7.2 The Bose-Einstein distribution gives the expected number of Bosons in a state with energy $\mathcal{E}$ at the temperature T . When the number of particles is not conserved (e.g., with photons and phonons), the chemical potential μ is set to zero.	161
7.3 Depiction of a periodic potential created by the sum of Coulomb potentials from an array of ions (represented by dots).	162
7.4 Cubic lattice in k -space resulting from periodic boundary conditions.	164
7.5 Spring-mass model for phonons on a 1-D lattice.	166
7.6 Graphical depiction of electron scattering by absorption of a phonon (left), and by emission of a phonon (right).	168
7.7 Visualizing conduction as the displacement of the Fermi sea. Left: in the absence of an electric field, the k -vectors cancel in pairs. Right: the application of an electric field shifts the Fermi sphere, resulting in a small sliver of volume near the Fermi surface in which the k -vectors are not canceled.	170

7.8	A rectangular wire used to relate the voltage to the current in normal and perfect conductors.	170
7.9	A superconductor expels magnetic flux (top), while static magnetic flux can penetrate a perfect conductor (bottom).	171
7.10	The potential is negative (attractive) for $\omega < \omega_q$ in the Jellium model. The BCS theory simply assumes that the potential is a negative constant up to the maximum phonon frequency ω_D .	180
7.11	Diagrammatic representation of two electrons interacting via the emission and absorption of a phonon.	181
7.12	The delayed response to the presence of an electron creates a local concentration of positive charge that can attract another electron.	181
7.13	The probability that a pair state is occupied as a function of energy near the Fermi surface. The width of the transition is approximately 2Δ .	184
7.14	Geometry used to understand the London gauge.	189
7.15	Behavior of the magnetic field and superconducting current at the interface between a superconductor and a normal region. (Note that the field and current have different scales.)	191
7.16	Geometry of a superconducting sample containing a hole used to calculate the magnetic flux quantum.	191
8.1	Geometry for a particle incident from the left on a potential barrier filling the half-space for $x > 0$. The wave function plot is for the case $\mathcal{E} = V/2$.	197
8.2	Geometry for a particle incident from the left on a potential barrier with finite width. The wave function plot is for the case $\mathcal{E} = V/2$.	198
8.3	A Josephson junction is formed by two superconductors separated by a thin insulating layer.	200
8.4	Circuit symbols for a Josephson junction. Left: equivalent circuit showing the tunneling junction in parallel with the junction capacitance. Right: compact symbol including both the tunneling junction and the capacitance.	203
8.5	Non-linear resonant circuit formed by shunting a Josephson tunnel junction with a capacitor. Here C is the total parallel capacitance including the intrinsic junction capacitance.	204
8.6	Two Josephson junctions in parallel forming a Superconducting Quantum Interference Device, or SQUID.	207
8.7	Example geometry for applying a current to tune a SQUID-type transmon qubit. The interdigital pattern is a capacitance in parallel with a SQUID. A current is applied to create a magnetic field linking the SQUID, appearing as the small “Y”-shaped structure near the center of the interdigital structure. <i>Source</i> : [59].	208
8.8	Parametric reflection amplifiers using variable inductors.	209
9.1	Measurement of T_1 relaxation time.	212
9.2	Measurement of T_2 dephasing time.	213
9.3	Measurement error calibration matrix M_e for three qubits. Element (i,j) shows the probability of measuring i after preparing state $ j\rangle$. Thus, each column j corresponds to the outcomes when preparing state $ j\rangle$.	

Diagonal elements show probabilities of correct measurement, while off-diagonals represent probabilities of errors.	214
9.4 Mitigating measurement errors using the calibration matrix from Figure 9.3. Prepared state is $(000\rangle + 111\rangle)/\sqrt{2}$.	215
9.5 Randomized benchmarking circuit. A random selection of m operators is followed by a single operator representing the inverse of the previous m operators. With no errors, the output will be $ 0\rangle^{\otimes n}$.	216
9.6 Randomized benchmarking experiment on IBM Q hardware.	217
9.7 Interleaved Randomized Benchmarking (IRB) adds a specific Clifford gate, C , after each randomized gate. The relationship between the exponential curves with and without C characterize the error rate for C .	218
9.8 Optimized spectrum with a notch at the anharmonicity value of -300 MHz.	220
9.9 Error-mitigated results for a single-qubit randomized benchmarking circuit. The four lower curves are the measured results with stretch factors $c = 1$ (red), $c = 2$ (orange), $c = 3$ (yellow), and $c = 4$ (green). The upper curves are the Richardson extrapolations to the first (light blue), second (dark blue), and third order (violet). The color bars represent histogram frequencies of 100 data points for each sequence size. <i>Source</i> : [81].	222
9.10 Error-mitigated results for traversal of a path along the Bloch sphere. The first-order mitigated result (blue) reaches the target state of $ 1\rangle$, while the experimental results show the effects of decoherence. Stretch factors are $c = 1$ (red), $c = 2$ (orange), and $c = 3$ (green). <i>Source</i> : [81].	222
9.11 Error-mitigated results for finding the ground state energy of a lithium-hydride (LiH) molecule using VQE on a four-qubit system. Stretch factors are 1, 1.1, 1.25, and 1.5, and the mitigated result uses first-order Richardson extrapolation. The inset shows the topology of the qubits involved in the experiment. <i>Source</i> : [81].	223
9.12 Sample control pulses generated by deep reinforcement learning (DRL) agent for the $R_X(\pi/2)$ gate. (a) Standard IBM DRAG pulse with a duration of 35ns. (b) Result of randomized benchmark characterization. The gray area shows the spread of data points and the black line is the exponential fit. The number is the errors per gate (EPG). (c)–(d) Stepwise DRL pulse (28.4ns) and RB results. Different colors represent the real (in-phase) and imaginary (out-of-phase) components of the waveform. RB fidelity is slightly better and spread of data is slightly smaller. (e)–(f) Stepwise DRL pulse with a 12.4ns duration, with similar fidelity. <i>Source</i> : [83].	224
9.13 Implementation of a three-qubit Toffoli gate (see Section 11.1.1).	225
10.1 Classical noisy channel.	228
10.2 Encoding data for error detection/correction: k -bit data is encoded using n bits to transmit over the channel. Decoding detects/corrects errors and transforms back to k bits.	228
10.3 Error correction using the 3-bit repetition code.	229
10.4 Hamming [7,4,3] code. Data bits are d_i and parity bits are c_i . Syndrome bits (s) indicate the position of a single bit flip error.	230

10.5 Quantum channel with Pauli errors.	231
10.6 Quantum error correction: encoding, error, correction.	232
10.7 Three-qubit error correction code for bit flips only.	233
10.8 Three-qubit error correction code for phase flips only.	234
10.9 Shor's $[[9,1,3]]$ code that corrects a single bit flip or phase flip.	235
10.10 Correction circuit for Shor's $[[9,1,3]]$ code.	236
10.11 Partial rotation around the x -axis.	237
10.12 Measuring $Z_1 Z_0$ for the two-qubit bit flip detection code. The top two qubits are the state to be measured, and the bottom qubit is an ancilla. The three circuits are equivalent, just drawn in different ways.	238
10.13 Projective measurement for an n -bit observable M with eigenvalues ± 1 .	239
10.14 Stabilizer version of 3-qubit bit flip error detection.	240
10.15 Error correction circuit for the $[[7, 1, 3]]$ Steane code.	241
10.16 Encoding circuit for the $[[7, 1, 3]]$ Steane code.	242
10.17 The logical $\bar{X}$ operator is transversal for the $[[7,1,3]]$ code, consisting of a single X gate on each physical qubit. Logical operators for Z , H , S , and CNOT are also transversal.	243
10.18 Performing the T operation on encoded data.	243
10.19 Estimated logical error rates for a concatenated Steane $[[7, 1, 3]]$ code for various physical error rates. For error rates below the threshold (p_{th}), additional levels of concatenation can reduce the logical error rate.	245
10.20 Surface code, composed by a square mesh of physical qubits. Data qubits (white) are interleaved with <i>measure-X</i> qubits (teal/dark gray) and <i>measure-Z</i> qubits (yellow/light gray). The direction of arrows show the direction of CNOT operations for projective measurements.	246
10.21 Surface code error cycle. The top row is a measure-Z circuit (ZZZZ stabilizer); the second row is a measure-X circuit (XXXX stabilizer). The sequence of gates allows for all measurements to happen in parallel across the entire surface.	248
10.22 When a bit flip (X) error occurs on a data qubit, the two measure-Z qubits will flip their measurement outcomes, indicated by circles and a change in color. This can be used as a <i>syndrome</i> to detect a bit flip error on a specific data qubit.	248
10.23 A phase flip error affects the outcome of its two adjacent X stabilizers. If both bit and phase flips occur, all four neighboring stabilizers will flip their outcomes.	249
10.24 When multiple errors occur in the same vicinity, they may not be uniquely distinguishable. In general, many different chains of bit flip errors can result in the same syndrome measurement. In the case of four errors (right), no change in the measurement outcome occurs. This actually not an error at all, because the resulting state is trivially related to the original quiescent state $ \psi\rangle$.	249
10.25 Logical X and Y operations on the surface code array.	251
10.26 Any path of X operators connecting one X boundary to the other is equivalent to the logical $\bar{X}$ operator, up to a global phase.	251
10.27 Rough merge of two planar qubits.	254
10.28 Smooth split of a planar qubits.	256

10.29	Preparation for a CNOT operation on control qubit $ C\rangle$ and target qubit $ T\rangle$, with the help of an intermediate qubit $ INT\rangle$.	257
10.30	State injection. (a) Prepare a single data qubit with the desired state $ \psi\rangle$, and other qubits as $ 0\rangle$. (b) Entangle data qubits to create multi-qubit state. (c) Stabilize to create logical state $ \psi\rangle_L$.	259
10.31	Errors in surface code for Exercise 10.4.	261
11.1	Three-qubit quantum adder.	264
11.2	A NOT gate is reversible. By using another NOT gate, we recover the original input x .	265
11.3	The reversible XOR gate requires that one of the inputs be carried over to the output. The result is the classical equivalent of the quantum CNOT gate.	265
11.4	To make an AND gate reversible, we must remember both of its input bits. Otherwise, for $xy = 0$, there's no way to know whether $x = 0$, $y = 0$, or both.	265
11.5	The Toffoli gate has two control bits and a target bit. The target is flipped if and only if <i>both</i> control bits are 1. The gate symbol for Toffoli is shown in (a), while (b) shows an implementation using classical gates.	265
11.6	The Toffoli gate is universal for Boolean logic, because it can be used to implement the NAND gate. Implementations of several fundamental gates are shown here. An open circle is a <i>negative</i> control; the control is active if the input bit is 0.	266
11.7	Examples of multi-controlled Toffoli (MCT) gates.	266
11.8	Reversible four-input AND circuit. The temporary values held in the ancilla bits t_1 and t_0 make this circuit unusable as a general reusable computing block; the ancilla bits cannot be reused in other parts of a larger circuit.	267
11.9	Reversible four-input AND circuit; ancilla bits t_1 and t_0 are uncomputed and restored to zero.	267
11.10	Reversible fanout using a CNOT. While this works for classical circuits, copying (cloning) of an arbitrary state is not permitted in quantum circuits (Section 11.2).	267
11.11	General reversible circuit (a) for irreversible function $f(x)$. Using zero inputs (b) generates $f(x)$, and the circuit is its own inverse (c).	267
11.12	Sketch of a generalized reversible circuit. Circuit F replaces AND gates with Toffoli gates, using additional ancilla bits as needed. Some CNOT gates (illustrated by the MCT in the middle) XOR the results with the y bits to produce the output $y \oplus f(x)$. Then circuit F^{-1} uncomputes the ancilla bits and restores the x bits as needed.	268
11.13	Reuse of ancilla bits (dashed lines) reduces the number of bits/qubits required for the circuit. (Adapted from [8].)	268
11.14	A CNOT can be used to replicate computational basis states (a), but for a superposition, the resulting state is entangled (b).	269
11.15	Action of Toffoli gate on superposition states.	269
11.16	An entangled Bell state is disentangled by a CNOT gate.	270
11.17	Implementation of Toffoli gate [9].	270

11.18	Qubit topologies for some superconducting quantum systems.	271
11.19	The direction of a CNOT gate can be reversed using Hadamard gates.	271
11.20	The SWAP gate exchanges the state of two qubits. It can be implemented using three CNOT gates.	271
11.21	SWAP gates may be needed to move qubit states into a position where they can interact. In the figure, we move the top and bottom qubits to the middle, perform a CNOT, and then move them back.	272
11.22	Classical non-reversible one-bit full adder.	273
11.23	Simple quantum adder. Carry output from bit i becomes an input for bit $i + 1$.	273
11.24	Three-bit ripple-carry adder.	274
11.25	Three-bit ripple-carry adder with reclaimed ancillas.	274
11.26	Three-bit ripple-carry adder, redrawn to better show circuit depth.	275
11.27	MAJ (majority) and UMA (unmajority and add) functions for in-place ripple-carry adder. The three-CNOT version of UMA on the right is used in the optimized adder circuit.	275
11.28	Full adder built with MAJ and UMA.	276
11.29	Three-bit ripple adder with one ancilla bit.	276
11.30	Optimized three-bit ripple adder with one ancilla bit.	277
11.31	Carry-lookahead circuit for 8-bit adder. All carry bits are generated using $O(\log n)$ levels of logic. Block $[i,j]$ calculates both propagate $P[i,j]$ and generate $G[i,j]$ signals. To compute $c_8 = G[0,8]$, only the shaded blocks are required.	278
11.32	Quantum carry-lookahead adder for 8-bit integers [127].	280
11.33	Controlled-Z (CZ) and controlled-phase (CP) gates.	284
11.34	CZ gate acting on an equal superposition of basis states. Only the state where both qubits are one is flipped.	284
11.35	Phase kickback using controlled- T gate. The target qubit is apparently unaffected, while the phase rotation occurs on the control qubit.	285
11.36	Selective phase rotation of basis states in $ A\rangle$ for which the predicate S evaluates as true. U_S is a unitary function that calculates the predicate S using an ancilla qubit.	286
11.37	Selective phase flip (π -rotation) of basis states in $ A\rangle$ for which the predicate S evaluates as true. U_S is a unitary function that calculates the predicate S using an ancilla qubit.	286
11.38	Phase kickback using CZ and CNOT gates.	286
11.39	Phase logic gates: performs phase rotation of λ on computational basis states that satisfy the Boolean operator.	287
11.40	Flipping the phase when $(a + \bar{b}) \cdot c$ is true. If the input starts as an equal superposition, the output state is shown below the circuit.	288
11.41	Interleave function for Exercise 11.4.	289
11.42	Max function for Exercise 11.6.	290
12.1	Grover search initialization. All states are in equal superposition. The correct answer ($x = 4$) is shown in black.	295
12.2	First iteration of Grover search. On the left, the oracle changes the sign of the correct answer, reducing the mean amplitude. On the right, all	

amplitudes are inverted around the mean, increasing the amplitude of the correct answer.	295
12.3 Second iteration of Grover search, further increasing the amplitude of the correct answer.	296
12.4 Third iteration of Grover search. In this case, the amplitude of the correct answer is reduced from its previous value.	296
12.5 Grover search algorithm for n qubits. Boxes <i>flip phase</i> and <i>invert</i> form one iteration, which is repeated $k = (\pi/4)\sqrt{N}$ times, where $N = 2^n$.	297
12.6 QFT of the state $ 2\rangle$.	302
12.7 QFT of the state $(2\rangle + 3\rangle)/\sqrt{2}$.	302
12.8 Result of QFT applied to $(1\rangle + 15\rangle)/\sqrt{2}$ yields a real-valued Fourier-basis state. Amplitudes are shown in the plot on the right, clearly showing one period of a sinusoid.	302
12.9 Circuit for creating a quantum-encoded square wave state with frequency = 1.	303
12.10 Quantum-encoded square wave state with frequency = 1 generated from the circuit in Figure 12.9.	303
12.11 Output of QFT applied to square wave.	303
12.12 Magnitudes of basis states from Figure 12.11. The same pattern is observed when taking the DFT of a time-domain square wave.	304
12.13 QFT for four qubits.	304
12.14 Relative phase rotations for each qubit in a 4-qubit QFT. Output state is a tensor product of these qubit states.	305
12.15 First stage of 4-qubit QFT.	305
12.16 Inverse QFT for four qubits.	305
12.17 Quantum Phase Estimation. Following our normal convention, the least significant bit qubit for the phase is at the top.	307
12.18 Factoring integer M using Shor's algorithm. The gray box is the only part of the algorithm that runs on a quantum computer.	310
12.19 For $a = 3$ and $M = 35$, the period of $a^x \bmod M$ is 12. Finding the period by brute force scales exponentially with m , the number of bits in M .	311
12.20 Modular exponentiation circuit using repeated modular multiplication, where $U^k y\rangle = a^{k,y} \bmod M\rangle$.	312
12.21 Quantum circuit for period-finding in Shor's algorithm. M is the integer to be factored, a is a number that is relatively prime to M , and r is the period of $a^k \bmod M$. The number of qubits n is chosen such that $M^2 \leq 2^n < 2M^2$. The measured eigenphase approximates a multiple of $1/r$, and classical post-processing determines r .	313
12.22 Components of a Variational Quantum Algorithm (VQA). Parameterized state preparation and measurement occur on the quantum processor, followed by classical optimization of the cost function.	315
12.23 Expressibility of various ansatzes, illustrated by 2000 sampled states [170].	318
12.24 Examples of problem-agnostic ansatz circuits from Sim <i>et al.</i> (Redrawn from [170].). R_x , R_y , and R_z gates are parameterized. In general, CNOT gates can be replaced with controlled- $\{Z, R_z, R_x, R_y\}$. Each circuit is a <i>layer</i> that can be repeated.	318

12.25	Example of MaxCut: Partition a graph such that the number of edges crossing between the partitions is maximized.	321
12.26	Circuit to rotate qubits i and j by $-\gamma$, if they are different.	322
12.27	QAOA circuit for MaxCut example in Figure 12.25. Dashed box is repeated p times with different parameters γ_i and β_i in each iteration. Dot notation is used for phase rotation gates to save space.	323

List of Tables

4.1 Useful ABCD matrices.	83
6.1 Classical parameters and the corresponding quantum operators.	134
7.1 Energy and momentum conservation in electron-phonon single scattering events.	168
7.2 Comparison of $2\Delta(0)/k_B T_c$ with the BCS theory prediction for some technologically important superconductors.	185
10.1 Effects of bit-flip error channel on a 3-qubit replication code.	233
10.2 Effects of phase-flip error channel on a 3-qubit phase replication code.	234
10.3 Stabilizer generators for Steane $[[7, 1, 3]]$ code.	241
10.4 Syndrome measurements for all single-qubit errors in the Steane code. Each syndrome bit corresponds to the measurement of the corresponding operator from Table 10.3—e.g., bit 0 corresponds to measuring M_0 .	241
11.1 Boolean algebra operators.	264
11.2 Comparison of in-place ripple-carry and carry-lookahead adder circuits. For this table, depth only counts Toffoli gates.	281
11.3 Comparison of adder circuits for 8-bit binary numbers. The bottom two rows show the results of compiling the circuit to a specific quantum computer topology. <i>Sparse</i> is the 53-qubit IBM Rochester topology, and <i>Full mesh</i> is Google Sycamore topology, both shown in Figure 11.18.	282
11.4 Controlled-Z (CZ) gate, where q_0 is the control and q_1 is the target.	284
12.1 Result of conditional phase rotations, the first four stages of Figure 12.13. The swap circuits at the end of the circuit reverse the order of the qubits, in order to match the desired rotations in Figure 12.14.	306

Preface

Over the past several years progress in quantum computing technology and algorithms has accelerated rapidly. We believe that Electrical and Computer Engineers have much to contribute to this work, and one of the goals of this book is to help introduce those with ECE backgrounds to this exciting area.

Our interest in quantum computing was initiated by the discussions between NC State and IBM about establishing the IBM Quantum Hub at NC State, and began in earnest in the spring of 2018. During this semester we both sat in on a Computer Science special topics seminar led by professors Frank Mueller and Patrick Dreher. This led to a joint CSC/ ECE special topics course on quantum computing in the fall of 2018 that was team-taught by professors Mueller and Dreher along with one of us (GTB).

As we have delved more deeply into quantum computing over the past several years, we have discovered that most of the rapidly-growing literature is addressed to those with backgrounds in physics, mathematics, and/or computer science, and often assumes a body of shared prerequisite knowledge and terminology that is not typical for Electrical and Computer Engineers. Electrical Engineers studying semiconductor physics do study quantum mechanics, but from a very different perspective. Band theory, effective mass, tunneling, and perhaps the hydrogen atom are covered, but quantum computing involves 2-level systems, state vectors and rotations, Hamiltonians, and unitary operators—topics not normally emphasized in device physics courses. Further, there is very little available in the current introductory literature explaining *how the systems operate*.

Just as semiconductor physics became a standard component of electrical engineering curricula beginning in the early 1960s, we believe that quantum computing will become integral parts of ECE curricula in the coming decades. The span of ECE covers the entire range of technologies underpinning quantum computing, including device physics and modeling, nanofabrication, RF and optical systems, signal processing, information theory, error correction and coding, transpilers and compilers, system architecture, and algorithms and applications.

At the time of this writing, there are competing technologies for the implementation of quantum computing, including trapped ions, quantum dots, topological structures, and artificial atoms made with superconducting devices. While we believe Electrical and Computer Engineers will be critical to the development of all of these systems, we have chosen in this volume to concentrate on superconducting technologies relying on Josephson junction transmons.

The intended audience is advanced undergraduate students and first-year graduate students in Electrical and Computer engineering. In presenting topics, we have tried to bridge the gaps that we have encountered between the prevailing literature and the backgrounds of Electrical and Computer engineers.

We offer this work not as an exposition by authorities, but rather as an introduction by technological “pilgrims” hoping to help other “pilgrims” along this exciting journey.

D. D. Stancil

G. T. Byrd

Raleigh, NC

Acknowledgments

We are grateful to our colleagues and students at NC State for many helpful conversations that have deepened and clarified our understanding. In addition, we would like to acknowledge helpful comments on portions of the manuscript by J. C. Bardin and N. Earnest-Noble. Of course, any remaining errors are ours.

We would also like to express appreciation to Brett Kurzman, Sarah Lemoire, Becky Cowan, and the capable team at Wiley for their assistance and guidance.

About the Companion Website

This book is accompanied by a companion website:

www.wiley.com/go/stancil/principlesofsuperconductingquantumcomputers

The website includes:

Instructor manual

1

Qubits, Gates, and Circuits

1.1 Bits and Qubits

Digital systems that are most familiar are based on binary digits, or “bits.” Each bit can take on either the value “1” or “0,” and any arbitrary data can be represented by such a binary representation. In addition, any arbitrary logical operation can be implemented using bits. We will refer to these familiar systems as “classical” systems, since they are governed by the everyday laws of classical physics.

Quantum computing is different from classical computing in a number of significant ways. The fundamental unit of information in quantum computing is the *qubit* (pronounced “KEW-bit”), short for quantum bit. The capabilities and behavior of qubits are quite different than bits, and we begin by pointing out and discussing the key differences as a launching point for our study of quantum computing.

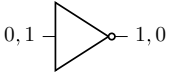
1.1.1 Circuits in Space vs. Circuits in Time

A simple classical logic circuit is represented by the NOT gate shown in Figure 1.1(a). The NOT gate turns a “0” into “1” and vice versa. In this circuit diagram the horizontal direction represents *space*, i.e., the input and output of the circuit are physically accessible from different points in the circuit, and they can be measured simultaneously.

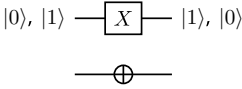
The quantum version of the NOT gate is the *X* gate shown in Figure 1.1(b). For qubits, the “0” and “1” states are normally written $|0\rangle$, and $|1\rangle$, respectively. We will discuss the meaning of this notation in more detail in a future section, but for now just consider them to be labels for the two states. In this case, the horizontal direction represents *time*, i.e., the input and output of the circuit represent the state of the *same* qubit after performing the *X* gate operation. In other words, unlike the usual structure of classical logic, a quantum gate represents an operation that you perform on a *single* qubit or set of qubits. The output effectively overwrites the input, and every time a gate is applied it changes the state of the qubit.

1.1.2 Superposition

A classical bit must either be a “0” or a “1.” In contrast, a qubit can also be in a *superposition state* that is part “0” and part “1.” If the qubit is in a “1” or “0” state, we say



(a) Classical NOT Circuit diagram. The horizontal direction represents *space*, i.e., the input and output of the circuit are physically accessible from different points in the circuit, and they can be measured simultaneously.



(b) Quantum X gate circuit (quantum version of the NOT gate). The horizontal direction represents *time*, i.e., the input and output of the circuit represent the state of the same qubit after performing the X gate operation. The lower part of the Figure shows an alternate symbol for the quantum NOT gate.

Figure 1.1 Interpretation of classical versus quantum NOT gates.

the qubit is in a *basis* state.¹ For basis states, the state of the qubit can be measured any number of times without changing the state, much like measuring the state of a classical gate. A superposition state will also yield either a “0” or a “1” when measured, with a probability determined by the mixture. In this case the action of making the measurement will “collapse” the superposition state onto one of the constituent basis states, and the information stored in the superposition state will be lost. For example, if $|\psi\rangle$ happens to represent a superposition state, then measuring the qubit at any time will collapse the state to either $|0\rangle$ or $|1\rangle$, destroying the information in the superposition state. From this point on, repeated measurement of the qubit will always yield the same result as the first measurement, just like a classical bit.

Mathematically the superposition state can be written

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (1.1)$$

where α and β are complex constants.

As mentioned, if such a superposition state is measured, it will always give either $|0\rangle$ or $|1\rangle$ but with probabilities of each determined by α and β . Specifically, the probabilities of measuring the two possible outcomes are given by

$$Pr[0] = |\alpha|^2, Pr[1] = |\beta|^2. \quad (1.2)$$

If these are the only two possible outcomes of the measurement, then the probabilities must sum to 1, or

$$|\alpha|^2 + |\beta|^2 = 1. \quad (1.3)$$

This ability to represent superposition states is one of the secrets to the power of quantum computing: there is a sense in which the qubits are able to explore multiple possibilities in parallel.

¹ To be more precise, $|0\rangle$ and $|1\rangle$ are known as the “computational basis” or “standard basis” states. For this chapter, we will restrict our discussion to the standard basis and will simply refer to basis states.

1.1.3 No Cloning

In a classical logic circuit we can measure the state of a bit at any time, and make as many copies of the state as we want. We can also do this for a qubit if it is known to be in one of the basis states; as described above, we can measure the $|0\rangle$ or $|1\rangle$ state without disturbing it, and we can make as many copies of $|0\rangle$ or $|1\rangle$ as needed.

However, it turns out that it is not possible to create a precise, independent copy of an *arbitrary* quantum state. This is known as the *no-cloning theorem*. We'll show a proof in Section 1.7, but for now let us consider the challenges this poses to the quantum programmer.

For example, we can't get estimates of α or β from running a circuit once. Cloning would allow me to run the circuit, make lots of copies of the result, and then measure each copy to estimate $|\alpha|^2$ and $|\beta|^2$ by the probability of measuring $|0\rangle$ or $|1\rangle$. Without cloning, we can only measure the result once. To get many measurements, we must run the same computation many times to produce (hopefully!) the same result over and over.

We cannot make copies of states for breakpoints, or to help with debugging or error recovery. It is also challenging to apply different operators to a single state during the course of a computation. Classical programmers take the ability to copy a state for granted, and this limitation requires some adjustment.

It turns out that it is possible to copy a quantum state using entanglement (Section 1.1.5), but only by destroying the state being copied. This represents a *transfer* of state rather than a copy, and is referred to as *teleportation*.

1.1.4 Reversibility

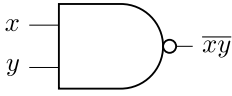
Classical NOT gates are reversible; i.e., two NOT gates in series returns the bit to its original state. However, the situation is different for classical logic gates with multiple inputs. As an example, consider the NAND Gate shown in Figure 1.2. It is not possible to uniquely determine the input bits from the output bit. Because of this, conventional multi-input logic is *irreversible*.²

In contrast, quantum gates are *reversible*. As a result, even though the output overwrites the input, the input is not lost since the effect of a series of gate operations can be reversed by applying the appropriate inverse operations.

1.1.5 Entanglement

The fact that we can only calculate the probability of measurement results—not the precise results—on qubits in superposition states allows for a phenomenon that has no classical analog: *entanglement*. If two qubits are entangled, then the states are correlated even though the outcome of a measurement on either of the qubits can only be predicted by its probability. For example, if two qubits are entangled, then a measurement on one of the qubits will give a result with probability determined by its superposition state, just as would happen with an isolated qubit. However, once the

² Note: There are classical multi-input logic gates that *are* reversible, called “conservative logic” gates. To date these have not been widely used, but they have the advantage of making it possible in principle to compute without dissipating power [1].



(a) NAND circuit diagram

x	y	$\overline{xy}$
0	0	1
0	1	1
1	0	1
1	1	0

(b) NAND truth table

Figure 1.2 NAND circuit diagram.

state of one of the qubits has been collapsed by a measurement, the entanglement means that we know exactly what we will get if we measure the second qubit.

It is a bit like flipping two coins at the same time, and having them always come up the same, i.e., both heads or both tails. Or alternatively, having them always come up opposite—one heads and the other tails. Prior to making a measurement, the states of both of the qubits have not yet been determined. However, when the state of one is collapsed by a measurement, the second qubit somehow instantaneously “knows” the result. This is true regardless of how far apart the qubits are. For example, suppose we entangle two qubits, send one to New York and the other to Tokyo, then make prior arrangements to measure both at the same time. We will discover that the results of the measurements will be correlated even though there was not enough time for a signal traveling the speed of light to travel between the qubits. Einstein called this “spooky action at a distance.”

If it were possible to control the result of the first measurement, then it would appear that we could communicate faster than the speed of light—violating the principle of special relativity. However, the fact that we cannot control the result of the first measurement means that we cannot actually send any information using this mechanism.³ So we can rest comfortably knowing that relativity has not been violated.

We will see that the phenomena of *superposition* and *entanglement* give quantum computing its unusual and powerful capabilities.

1.2 Single-Qubit States

Since there are two components in a qubit’s state, we can represent the state as a two-dimensional vector, referred to as a *state vector*. The basis states can be written

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (1.4)$$

It follows that we can express a general superposition state as a weighted sum of the basis states:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \quad (1.5)$$

³ Entanglement can be used to increase the *capacity* of a communication channel, however, using a protocol known as *super-dense coding*.

The notation $|\psi\rangle$ is called a “ket.” The transpose complex conjugate⁴ of a ket is denoted $\langle\psi|$ and is called a “bra”:

$$|\psi\rangle^\dagger = \langle\psi| = [\alpha^* \quad \beta^*], \quad (1.6)$$

where the $\dagger$ superscript indicates the transpose complex conjugate, also referred to as the Hermitian conjugate, or adjoint. The “bra” and “ket” terminology may seem strange until you multiply them together to form a “bra-ket.” For example,

$$\langle\psi|\psi\rangle = [\alpha^* \quad \beta^*] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha^* \alpha + \beta^* \beta \quad (1.7)$$

$$= |\alpha|^2 + |\beta|^2. \quad (1.8)$$

(Note that when written as a bra-ket, only a single vertical bar is used in the center.) The bra-ket operation (i.e., multiplying a bra and ket) is also referred to as an “inner product.” Referring to (1.8), we see that the inner product of a state vector with itself gives the sum of the probabilities that each of the basis states will be obtained in a measurement. Since the sum of the probabilities of all possible outcomes must equal 1, we see that the state vectors must be properly normalized to a length of unity.

1.3 Measurement and the Born Rule

We have previously stated that the probability of measuring a given component of a superposition state is given by the magnitude squared of its coefficient. The act of measuring requires an apparatus that interacts with the qubit in order to extract information. The rules of quantum mechanics tell us that the apparatus can only give partial information, related to a set of basis states. For now, we will assume that measurement is always with respect to the standard basis states, $|0\rangle$ and $|1\rangle$, which is the case for most quantum computing systems. However, it is possible for a measurement apparatus to be associated with a different set of basis states. (We will discuss measurement in more detail in later chapters.)

When a qubit is measured: (a) the state is changed to one of the basis states associated with the measurement, and (b) the measurement apparatus tells us the resulting state. In general, the probability that a state $|\psi\rangle$ will be found in the basis state $|a\rangle$ when measured is given by

$$Pr[|a\rangle] = |\langle\psi|a\rangle|^2. \quad (1.9)$$

This is called the *Born Rule*. For example, the probability that the outcome of measuring the state $|\psi\rangle$ above is $|0\rangle$, $|1\rangle$ is given by

$$Pr[|0\rangle] = |\langle\psi|0\rangle|^2 = |\alpha|^2 |\langle 0|0\rangle|^2 = |\alpha|^2, \quad (1.10)$$

$$Pr[|1\rangle] = |\langle\psi|1\rangle|^2 = |\beta|^2 |\langle 1|1\rangle|^2 = |\beta|^2, \quad (1.11)$$

as we found before.

⁴ The transpose of a matrix or vector is formed by interchanging the rows and columns. The transpose of a column vector gives a row vector. The transpose complex conjugate is formed by taking the transpose of the vector or matrix and then taking the complex conjugate of each element.

1.4 Unitary Operations and Single-Qubit Gates

We refer to a transformation from one quantum state to another as a *gate*. The effect of a single qubit gate is to change α and β into a new mixture α' and β' :

$$\begin{aligned}\alpha' &= U_{11}\alpha + U_{12}\beta \\ \beta' &= U_{21}\alpha + U_{22}\beta.\end{aligned}\tag{1.12}$$

This can be written as a matrix equation

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}\tag{1.13}$$

$$|\psi'\rangle \equiv U |\psi\rangle.\tag{1.14}$$

Since the length of the state vector must always be unity, we are only allowed to use matrices U that conserve the length of the vector. In other words, $\langle\psi'|\psi'\rangle = \langle\psi|\psi\rangle = 1$. This puts a very important constraint on the matrix U :

$$\langle\psi'|\psi'\rangle = \langle\psi|U^\dagger U|\psi\rangle = 1,\tag{1.15}$$

using the following observation:

$$\langle\psi'| = (U |\psi\rangle)^\dagger = \langle\psi| U^\dagger.\tag{1.16}$$

Since $\langle\psi|\psi\rangle = 1$, we conclude that

$$U^\dagger U = I,\tag{1.17}$$

where I is the identity matrix

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.\tag{1.18}$$

Matrices that satisfy this requirement are called *unitary* matrices. We can view these matrices as performing an operation on a qubit by changing the mixture of basis states. Consequently, the matrices U are also referred to as *unitary operators*.

The identity matrix I can be considered to be the simplest “gate” and leaves the state vector unchanged. Classically, the NOT gate is the only non-trivial single-bit gate. In contrast, there are many non-trivial single qubit quantum gates (technically, the number of 2×2 unitary matrices is unlimited). The most common non-trivial single qubit gates are the Pauli-X (X), Pauli-Y (Y), Pauli-Z (Z), and Hadamard (H) gates defined as follows:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\tag{1.19}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},\tag{1.20}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},\tag{1.21}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (1.22)$$

To get an understanding of what these gates do, consider applying an X gate to the “ground” state $|0\rangle$:

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle. \quad (1.23)$$

Similarly,

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle. \quad (1.24)$$

We see then that the X gate is a “bit flip” gate, and transforms $|0\rangle$ into $|1\rangle$ and vice versa. This, then is the analog of the classical NOT gate. You should verify the following results from applying Y , Z , and H gates:

$$Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle, \quad (1.25)$$

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle, \quad (1.26)$$

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad (1.27)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

In addition, it is interesting to note that each one of these matrices is its own Hermitian conjugate. Consequently, these four gates have the property that applying them twice gives the identity matrix:

$$X^2 = Y^2 = Z^2 = H^2 = I. \quad (1.28)$$

Note that while X , Y , and Z gates transform between the $|0\rangle$ and $|1\rangle$ states, the Hadamard gate actually creates a *superposition* state, and therefore will prove to be particularly useful. The states resulting from applying H to the basis states are given their own names:

$$|+\rangle \equiv H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad (1.29)$$

$$|-\rangle \equiv H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \quad (1.30)$$

Multiple gates can be sequentially applied to a qubit by matrix multiplication:

$$XYH|0\rangle \quad (1.31)$$

This expression says that we start with the ground state, then apply a Hadamard gate, followed by a Pauli- Y and then a Pauli- Z . Note that this process conceptually moves from right to left. This computation can be represented by a quantum circuit diagram as shown in Figure 1.3. Note that the process moves from left to right on the circuit

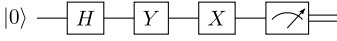


Figure 1.3 Circuit representation of Eq. (1.31). In a quantum circuit diagram, the operation goes from left to right, while the matrix expression is shown going from right to left. The final box is a measurement in the standard basis, resulting in a classical bit.

diagram: we begin with the ground state on the left, then apply a Hadamard gate, a Pauli-Y gate, and a Pauli-X.

Finally, after performing a quantum calculation, we generally measure each qubit. The symbol for a measurement is shown as the last element in Figure 1.3. This action collapses the final state onto one of the basis states. The outcome of a measurement is a classical bit that is stored in a classical register. “Quantum wires” are denoted by solid lines, and “classical wires” are denoted with double lines. A quantum wire simply represents a time-interval over which the state is kept unchanged.

1.5 Two-Qubit Gates

It is also possible to have gates with multiple qubits as inputs. However, unlike classical multi-bit gates, the fact that quantum gates are reversible requires that there must be the same number of output qubits as input qubits. In this section we will explore some common two-qubit gates.

1.5.1 Two-Qubit States

When we consider a system with two qubits, we don’t just consider each qubit independently of the other. Instead, this forms a two-qubit system with its own set of basis states. If we know the state of each qubit, then the combined two-qubit state is described using the *tensor product* of the two state vectors, defined as follows:

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a \begin{bmatrix} c \\ d \end{bmatrix} \\ b \begin{bmatrix} c \\ d \end{bmatrix} \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}. \quad (1.32)$$

Using the standard basis, the basis states for a two-qubit system are defined by combinations of the $|0\rangle$ and $|1\rangle$ states:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ 0 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (1.33)$$

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ 0 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad (1.34)$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ 1 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad (1.35)$$

$$|11\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ 1 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (1.36)$$

Any two-qubit state can be written as a linear combination of the basis states:

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle. \quad (1.37)$$

Two-qubit state vectors are also normalized:

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1. \quad (1.38)$$

As we will see later, while every two-qubit state can be written in the form of Eq. (1.37), not every two-qubit state can be written as the tensor product of single-qubit states.

This can be generalized into a system with n qubits, requiring state vectors with 2^n components with 2^n complex coefficients.

1.5.2 Matrix Representation of Two-Qubit Gates

Just as single qubit gates can be represented by 2×2 matrices, an n -qubit gate can be represented by a $2^n \times 2^n$ matrix. Consequently two-qubit gates require the construction of 4×4 unitary matrices. Given two single-qubit operators A and B , the tensor product is defined as:

$$A \otimes B \equiv \begin{bmatrix} A_{11}B & A_{12}B \\ A_{21}B & A_{22}B \end{bmatrix} = \begin{bmatrix} A_{11} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} & A_{12} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \\ A_{21} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} & A_{22} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \end{bmatrix}, \quad (1.39)$$

which creates a 4×4 matrix.

Suppose we wanted to construct a two-qubit circuit starting in the state $|10\rangle$ with an X gate applied to the left qubit, and a Y gate applied to the other. Mathematically this

would be written

$$X \otimes Y |10\rangle. \quad (1.40)$$

Referring to (1.24) we see that the X gate will simply flip the left qubit, and referring to (1.25) we see that the Y gate will flip the right qubit and add the coefficient i . We conclude that

$$X \otimes Y |10\rangle = i |01\rangle. \quad (1.41)$$

To see how this would be implemented using the matrix representation, we first construct the $X \otimes Y$ matrix:

$$X \otimes Y = \begin{bmatrix} 0 & \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} & 1 & \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \\ 1 & \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} & 0 & \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{bmatrix}. \quad (1.42)$$

Completing the calculation gives the expected result:

$$X \otimes Y |10\rangle = \begin{bmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = i \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = i |01\rangle. \quad (1.43)$$

A particularly interesting two-qubit circuit is formed by applying a Hadamard gate to each qubit in the ground state: $H \otimes H |00\rangle$. Let us first compute $H \otimes H$:

$$H \otimes H = \frac{1}{2} \begin{bmatrix} 1 & \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & 1 & \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ 1 & \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & -1 & \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \quad (1.44)$$

Completing the calculation gives:

$$H \otimes H |00\rangle = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \quad (1.45)$$

Note that the resulting state vector can be decomposed into a sum of all of the two-qubit basis states:

$$\frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (1.46)$$

or alternatively

$$H \otimes H |00\rangle = \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle). \quad (1.47)$$

We see that application of Hadamard gates to each qubit creates an equally weighted superposition of all possible basis states. This is often a very useful starting point for a quantum calculation.

Although the matrix representation can be helpful in understanding the operations, calculations can often be done more compactly once the effect of the gates are understood. For example, we could write $H \otimes H |00\rangle = H \otimes H |0\rangle |0\rangle$, apply the Hadamard gates to each qubit, and simplify:

$$\begin{aligned} H \otimes H |0\rangle |0\rangle &= H |0\rangle \otimes H |0\rangle \\ &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle). \end{aligned} \quad (1.48)$$

We conclude this section with a comment on notation. A more compact notation is often used for situations where the same operator is applied across multiple qubits; i.e., $H \otimes H$ is alternatively written $H^{\otimes 2}$, $H \otimes H \otimes H = H^{\otimes 3}$, etc.

1.5.3 Controlled-NOT

The gates that we have considered so far involve operations that are independently applied to separate qubits—there is no qubit–qubit interaction. If we are to entangle two qubits, then we need classes of gates where the operation on one qubit depends on the state of another. One of the most important such gates is the controlled-NOT, or CNOT gate. For this gate, one of the input qubits is the “control,” and the other is the “target.” If the control qubit is zero, then nothing is done to the target qubit, but if the control qubit is one, then the target qubit is flipped. For example, if the right qubit in our notation is the control and the left qubit is the target, then the CNOT gate transforms the basis states as follows:

$$U_{\text{CN}} |00\rangle = |00\rangle; U_{\text{CN}} |01\rangle = |11\rangle; U_{\text{CN}} |10\rangle = |10\rangle; U_{\text{CN}} |11\rangle = |01\rangle. \quad (1.49)$$

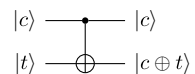
The effect of a CNOT can be compactly represented by $U_{\text{CN}} |t\rangle |c\rangle = |c \oplus t\rangle |c\rangle$, where $\oplus$ represents exclusive-OR or modulo-2 addition (e.g., $0+1=1$, but $1+1=0$). The matrix representation of the CNOT gate is

$$U_{\text{CN}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (1.50)$$

and the circuit symbol is shown in Figure 1.4.

It is worth noting at this point that we are putting the least-significant qubit at the top of a circuit diagram, and on the right on the state labels used with kets. This labels states with the natural binary order. However, there are different conventions in use, and this

Figure 1.4 Symbol for a CNOT gate, and its effect on basis states.



can be a point of confusion. Some authors put the top-most qubit in a circuit diagram on the *left* when they label kets. In this alternate notation, $U'_{CN} |c\rangle |t\rangle = |c\rangle |c \oplus t\rangle$ so that

$$U'_{CN} |00\rangle = |00\rangle; U'_{CN} |01\rangle = |01\rangle; U'_{CN} |10\rangle = |11\rangle; U'_{CN} |11\rangle = |10\rangle. \quad (1.51)$$

The matrix representation of the CNOT gate in this alternate convention is

$$U'_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (1.52)$$

We will consistently use the first convention, with the least-significant qubit (top-most on a circuit diagram) on the *right* when writing state labels.

1.6 Bell State

Consider the circuit shown in Figure 1.5. The circuit can be described mathematically as $U_{CN} (I \otimes H) |00\rangle$. Here the expression $I \otimes H$ simply means a Hadamard gate is applied to the right qubit, and the identity matrix applied to the left qubit (which leaves the left qubit unchanged). Completing the calculation gives

$$U_{CN} (I \otimes H) |00\rangle = U_{CN} |0\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \quad (1.53)$$

$$= \frac{1}{\sqrt{2}} U_{CN} (|00\rangle + |01\rangle) \quad (1.54)$$

$$= \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle). \quad (1.55)$$

Note that there is no way to factor this state into (qubit 1 state) $\otimes$ (qubit 0 state). This is known as a *Bell State*, and it is an example of an *entangled* state, as described in Section 1.1.5.

If we measure the state of one of the qubits in (1.55), then we will obtain either a zero or a one with equal probability, as we would expect. The unusual behavior happens when we measure the second qubit: if we measure $|0\rangle$ on the first qubit, then we know the state has collapsed to $|00\rangle$, so a measurement of the second qubit will also yield $|0\rangle$. Similarly, if the first measurement yields $|1\rangle$, then the state must have collapsed to

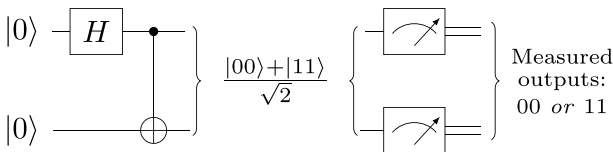


Figure 1.5 Circuit for creating an entangled state known as a Bell State. When the two qubits are measured, they will either both be 0, or they will both be 1.

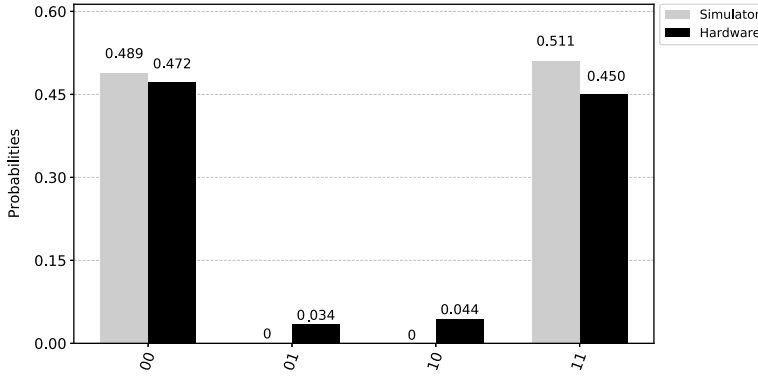


Figure 1.6 Result of executing the circuit 1024 times on a quantum simulator, compared with executing the circuit 1024 times on a real IBM quantum computer.

$|11\rangle$, so a measurement of the second qubit will also yield $|1\rangle$. The measurements are correlated, no matter how far apart the physical qubits might be.

We simulated the Bell circuit using IBM’s Qiskit [2], and also executed the circuit on real IBM Quantum hardware. A histogram illustrating the results of simulating the circuit 1024 times (each execution is referred to as a “shot”) is shown in Figure 1.6. From (1.55), we see that the probability amplitudes of the states $|00\rangle$ and $|11\rangle$ are both equal to $1/\sqrt{2}$. Since the probability of measuring a state is the absolute magnitude squared of the amplitude, the probability of measuring each state is $1/2$. Consistent with this, the measurements from the 1024 shots are roughly equally divided between the states $|00\rangle$ and $|11\rangle$, with no other states occurring.

The simulator gives the result expected from an error-free quantum processor. In contrast, the quantum processors available today are noisy and contain errors. As an illustration, Figure 1.6 also shows the result of executing 1024 shots on a real IBM Quantum processor. Although the states $|00\rangle$ and $|11\rangle$ do occur most frequently, the error states $|01\rangle$ and $|10\rangle$ occasionally occur as well. Fortunately there are techniques to reduce and partially mitigate such noise (Chapter 9), and these techniques represent an active area of research.

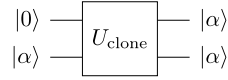
1.7 No Cloning, Revisited

With a better understanding of quantum states and operations, we are now ready to construct a proof of the no-cloning theorem. The proof relies on the fact that unitary operators are *linear*; when applied to a sum of states, the operator operates independently on each component:

$$U(|\psi\rangle + |\phi\rangle) = U|\psi\rangle + U|\phi\rangle. \quad (1.56)$$

Suppose we have an operator U_{clone} that takes two qubits, as shown in Figure 1.7. When the second qubit is $|0\rangle$, it is replaced with an exact copy of the first qubit. The

Figure 1.7 Hypothetical cloning operator, that creates an exact and independent copy of unknown quantum state $|\alpha\rangle$. The text will show that such an operator cannot be implemented.



proof will show that such an operator cannot exist, because it is not compatible with the principle of linearity.

Further suppose that we have two states:

$$|\alpha\rangle = a_0 |0\rangle + a_1 |1\rangle \quad (1.57)$$

$$|\beta\rangle = b_0 |0\rangle + b_1 |1\rangle. \quad (1.58)$$

By the definition of cloning:

$$U_{\text{clone}} |\alpha\rangle |0\rangle = |\alpha\rangle |\alpha\rangle = |\alpha\alpha\rangle, \quad (1.59)$$

$$U_{\text{clone}} |\beta\rangle |0\rangle = |\beta\rangle |\beta\rangle = |\beta\beta\rangle.$$

Now consider a new state $|\delta\rangle = (|\alpha\rangle + |\beta\rangle)/\sqrt{2}$. By the definition of cloning:

$$\begin{aligned} U_{\text{clone}} |\delta\rangle |0\rangle &= |\delta\rangle |\delta\rangle \\ &= \frac{|\alpha\rangle + |\beta\rangle}{\sqrt{2}} \otimes \frac{|\alpha\rangle + |\beta\rangle}{\sqrt{2}} \\ &= \frac{1}{2} (|\alpha\alpha\rangle + |\alpha\beta\rangle + |\beta\alpha\rangle + |\beta\beta\rangle). \end{aligned} \quad (1.60)$$

However, by the linearity of unitary operators:

$$\begin{aligned} U_{\text{clone}} |\delta\rangle |0\rangle &= \frac{1}{\sqrt{2}} U_{\text{clone}} (|\alpha\rangle |0\rangle + |\beta\rangle |0\rangle) \\ &= \frac{|\alpha\alpha\rangle + |\beta\beta\rangle}{\sqrt{2}}. \end{aligned} \quad (1.61)$$

Since Eqs. (1.60) and (1.61) cannot both be true, there is no unitary U_{clone} that can perform the cloning operation for all states.

We stated earlier that we *can* clone a (computational) basis state. This can be done with the CNOT gate, with the first qubit as the control. (With our bottom-to-top ordering, this corresponds to the U'_{CN} operator from (1.52).) Suppose state $|\psi\rangle$ is either $|0\rangle$ or $|1\rangle$, but we don't know which.

$$U'_{\text{CN}} |\psi\rangle |0\rangle = \begin{cases} U'_{\text{CN}} |00\rangle = |00\rangle, & \text{if } |\psi\rangle = |0\rangle \\ U'_{\text{CN}} |10\rangle = |11\rangle, & \text{if } |\psi\rangle = |1\rangle \end{cases} = |\psi\psi\rangle. \quad (1.62)$$

If we apply the circuit from Figure 1.5 to an arbitrary state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, we get a result that looks sort of like cloning, but not quite:

$$U_{\text{CN}}(I \otimes H) |0\rangle |\psi\rangle = \frac{\alpha |00\rangle + \beta |11\rangle}{\sqrt{2}}. \quad (1.63)$$

The result is not cloning because the two qubits are entangled. We did not succeed in creating two independent copies of $|\psi\rangle$. This is a useful construct, however, and can be

extended to create n -qubit states that look like this:

$$\frac{\alpha |0000 \dots\rangle + \beta |1111 \dots\rangle}{2^{n/2}}.$$

These states will be useful for quantum error correction codes (Chapter 10).

1.8 Example: Deutsch's Problem

We now have enough quantum computing “machinery” to work out a simple example of a quantum algorithm. The example algorithm is admittedly not very useful, but it provides some important insights into how quantum computers can outperform classical computers.

Consider a function that takes a single binary digit as input, and provides a single binary digit as output. There are four possible functions: $f_1(x) = 0$, $f_2(x) = 1$, $f_3(x) = x$, and $f_4(x) = \bar{x}$. Suppose someone gave us an implementation of one of these functions in a black box, and asked us to determine whether $f(0) = f(1)$, or if $f(0) \neq f(1)$. Classically, we could do this in two function calls, one with $x = 0$, and one with $x = 1$. However, using a quantum algorithm we can answer this question with a single function call! Let's see how this could be done.

First, suppose that we have the gate U shown in Figure 1.8(a) for implementing the function $f(x)$. The function is implemented by changing the sign of the state: if $f(x) = 0$ then the sign is unchanged, and if $f(x) = 1$ then the sign of the state is flipped. We will return to the question of how to implement this gate shortly, but for now let us assume that the gate is given to us, and our job is to determine if $f(0) = f(1)$ or $f(0) \neq f(1)$. Now consider the circuit shown in Figure 1.8(b). Applying the Hadamard gate to the input $|0\rangle$ gives

$$\psi_1 = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle). \quad (1.64)$$

Note that the Hadamard gate enables us to apply U to *both* $|0\rangle$ and $|1\rangle$ at the *same time*. This is referred to as *quantum parallelism* and is one of the secrets behind the power of quantum computing (although there are some key qualifications that make this somewhat less exciting than it would seem at first!).

Continuing with the calculation, we next compute $\psi_2 = U\psi_1$ to obtain

$$\psi_2 = \frac{1}{\sqrt{2}} [(-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle]. \quad (1.65)$$

Figure 1.8 Conceptual illustration of the Deutsch Problem.

$$|x\rangle \longrightarrow \boxed{U} \longrightarrow (-1)^{f(x)} |x\rangle$$

(a) “Black Box” for evaluating the function $f(x)$.

$$|0\rangle \longrightarrow \boxed{H} \longrightarrow \psi_1 \longrightarrow \boxed{U} \longrightarrow \psi_2 \longrightarrow \boxed{H} \longrightarrow H\psi_2$$

(b) Circuit for illustrating Deutsch's Problem.

Applying the second Hadamard gate gives the final output:

$$H\psi_2 = \frac{1}{2}(-1)^{f(0)}[|0\rangle + |1\rangle] + \frac{1}{2}(-1)^{f(1)}[|0\rangle - |1\rangle] \quad (1.66)$$

$$= \frac{1}{2}[(-1)^{f(0)} + (-1)^{f(1)}]|0\rangle + \frac{1}{2}[(-1)^{f(0)} - (-1)^{f(1)}]|1\rangle. \quad (1.67)$$

Examining (1.67), we see that if $f(0) = f(1)$ the second term vanishes and the output is $|0\rangle$. In contrast, when $f(0) \neq f(1)$, then the first term vanishes and the output is $\pm |1\rangle$. Therefore, if we measure the output as $|0\rangle$, we know that $f(0) = f(1)$, while if we measure $|1\rangle$, we know that is not the case. (Our measuring device will ignore the minus sign and show us a result of “1” if the output state is $-|1\rangle$.)

Arranging for the amplitudes of the correct states to add and others to cancel is an example of *quantum interference*, another key principle underlying quantum computing. So we see that in a single function call, we are able to determine whether or not $f(x)$ is balanced or constant—a feat that would only be possible classically with two function calls!

Now let us turn our attention to how the gate U can be realized. First, we have to confess that the circuit in Figure 1.8 is hiding something: we need another qubit to implement the U gate.

Consider the function $f(x) = 0$. We want a unitary U_f that converts $|x\rangle$ to $|f(x)\rangle$, but this is clearly not reversible. Given an output of $|0\rangle$, there’s no way to know whether the input is $|1\rangle$ or $|0\rangle$. To make the function reversible, we need to include both $f(x)$ and x in the output, so that we can reconstruct the input when reversing the operation.

For two outputs, we need two inputs. We typically use $|0\rangle$ as second input, but to be more general, let’s assume that the second input can be any input $|y\rangle$. With $|x\rangle$ and $|y\rangle$ as the inputs, the outputs will be $|x\rangle$ and $|y \oplus f(x)\rangle$, where $\oplus$ is the standard Boolean exclusive-OR (XOR) operation. As shown in Figure 1.9, this function is its own inverse; if we use $|x\rangle$ and $|y \oplus f(x)\rangle$ as the input, we get $|x\rangle$ and $|y \oplus f(x) \oplus f(x)\rangle = |y\rangle$ as the output.

Figure 1.10 shows the four alternative implementations of U_f , depending on which version of $f(x)$ we want. From left to right:

- $f_1(x) = 0$, do nothing to y , equivalent to $y \oplus 0$.
- $f_2(x) = 1$, flip y using a NOT gate, equivalent to $y \oplus 1$.
- $f_3(x) = x$, use a CNOT gate to flip y if $x = 1$, equivalent to $y \oplus x$.
- $f_4(x) = \bar{x}$, flip x and then XOR with y using a CNOT gate, equivalent to $y \oplus \bar{x}$. We then use another NOT gate to restore x to its original value.

Now, using our U_f circuit, we will compute $(-1)^{f(x)}|x\rangle$ using a technique known as *phase kickback*. If we set $|y\rangle = |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, then the top output of U_f will

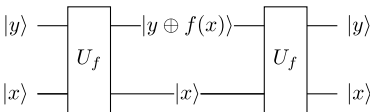


Figure 1.9 Reversible circuit for calculating $f(x)$.

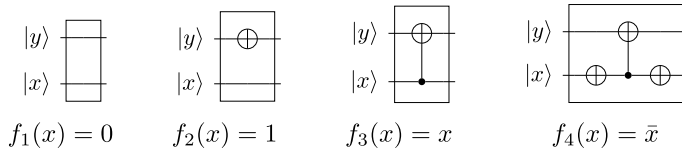


Figure 1.10 Implementations of black-box function U_f for Deutsch's problem. Top output is $|y \oplus f(x)\rangle$, and bottom output is $|x\rangle$.

remain $|-\rangle$ and the bottom output will be $|x\rangle$ if $f(x) = 0$ and will be $-|x\rangle$ if $f(x) = 1$. Let's do the math.

First, consider what happens if we flip $|-\rangle$ using the X gate:

$$\begin{aligned}
 X|-\rangle &= \frac{1}{\sqrt{2}}X(|0\rangle - |1\rangle) \\
 &= \frac{1}{\sqrt{2}}X(|1\rangle - |0\rangle) \\
 &= -|-\rangle.
 \end{aligned} \tag{1.68}$$

In the cases where U_f flips $|y\rangle$, we have:

$$U_f|x\rangle|-\rangle = |x\rangle(-|-\rangle) = (-|x\rangle)|-\rangle. \tag{1.69}$$

It might seem that we “magically” moved the minus sign from one qubit to the other, but remember that this is really a two-qubit state, not two individual qubits. And -1 is just a constant, so we can associate it with either part of the two-qubit state.

Figure 1.11 shows the under-the-covers implementation of the circuit from Figure 1.8, where the dashed box replaces U . We add our ancilla qubit as part of this circuit. We typically assume that qubits are initialized in state $|0\rangle$, so we use HX to create the desired $|-\rangle$ state. Then we have our black-box function, U_f , which has the desired sign-flipping effect on $|x\rangle$. Finally, it is good practice to “uncompute” the ancilla bit, restoring it $|0\rangle$, so that the qubit can be used elsewhere in a larger circuit.

If the notions of reversible logic and phase kickback seem strange, don't worry. We'll have a lot more to say about them in Chapters 11 and 12.

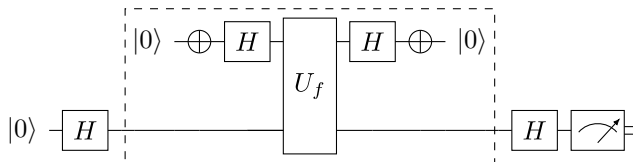


Figure 1.11 Implementation of Deutsch's algorithm. The dashed box is equivalent to U in Figure 1.8.

1.9 Key Characteristics of Quantum Computing

The Bell and Deutsch examples have illustrated several important characteristics of quantum computing:

- *Results are (usually) statistical:* When a classical program is executed, you obtain the same result each time. However, when a quantum circuit is executed, multiple results are possible with probabilities determined by the magnitude squared of the amplitudes of each state. Consequently, in general a circuit must be executed a large number of times, with the results of the computation extracted from a histogram of the measured outcomes.

It is possible, however, for the outcome to be one particular state with probability 1. This is the case for Deutsch's algorithm, for example. We know the answer for certain after one execution (assuming the quantum machine is error-free). So it's not the case that a quantum algorithm *must* be probabilistic, but the most interesting algorithms tend to be that way.

- *Quantum parallelism:* Arranging for the input state to be a superposition allows the calculation to consider multiple cases at once. However, it is not as easy to capitalize on this as it might sound. As indicated in the previous bullet, even though the output state may contain the complete solution, a single measurement will yield only one state with a probability given by the squared magnitude of the amplitude of the state in the solution of the problem.
- *Exponential scaling:* The number of cases that can be considered scales as 2^N , where N is the number of qubits. Beyond about 50 qubits, a general quantum processor cannot be simulated by a supercomputer; said differently, a processor with of order 50 or more qubits is capable of computations not possible on the best classical computers. However, if the quantum program is restricted to certain types of gates, then the operation of the quantum computer can be efficiently simulated by a classical computer.
- *Quantum interference:* When multiple cases are considered using superposition, the goal of the quantum circuit is to arrange for the amplitudes of correct answer(s) to add constructively, while arranging for the incorrect answer(s) to add destructively.
- *Asking the right question:* Although the output state of a quantum calculation will generally contain information about many possibilities, making a measurement collapses the state and therefore only gives a single result. In the Deutsch Problem, two classical function calls would not only tell you whether the function was constant or balanced, but it would tell you precisely what the function was. In contrast, the quantum calculation answers the question about whether the function is constant or balanced in one function call (which requires consideration of both cases), but it does not tell you which of the two possible functions it is.

1.10 Quantum Computing Systems

At this point, you may be asking: what kind of physical system exhibits the behavior that we can exploit for quantum computing? Any two-state quantum mechanical system can represent a qubit, and there are several possibilities, such as the spin of an electron, the polarization of a photon, or the energy level of an electron in a charged ion. Any of these systems can be used to build a quantum computer, but there are tradeoffs

regarding how the qubits can be manufactured and controlled, and how they interact with one another.

In this book, we concentrate on one specific technology for creating qubits and quantum computing systems: superconducting circuits. Unlike many competing technologies, superconducting qubits are macroscopic in size and are based on well-known nanofabrication technologies. They represent the current technology of choice for several companies building quantum computer systems, including IBM, Google, and Rigetti.

A large part of this book, Chapters 2–8, is devoted to a detailed explanation of these devices and how to control them to carry out the fundamental operations of a quantum computer, described above. In this section, before diving into the details, we provide a high-level overview of a superconducting quantum computer.

As we will see in Chapter 2, we will need to couple the qubit to a signal whose frequency depends on the energy difference between the $|0\rangle$ and $|1\rangle$ states, i.e., the ground and excited states. In superconducting quantum computers, this energy difference corresponds to a microwave frequency near 5 GHz. Consequently, we must design a microwave system to control and measure superconducting qubit states.

The general features of the microwave system to control and readout superconducting qubits are shown in Figure 1.12. A key feature is that the qubits must be held at a very low temperature, near absolute zero. Consequently the qubits must be located in a cryogenic refrigerator. To understand why this is necessary, we want to make sure that if we put the qubit in the ground state, it stays in the ground state. In other words, we want to make sure that it is unable to absorb enough energy incidentally from its environment to make a transition. A circuit in equilibrium at temperature T can emit and absorb photons with the energy kT , where k is Boltzmann's constant. The energy of a photon is $\hbar\omega$, where ω is the frequency and $\hbar$ is Planck's constant divided by 2π . We want to make sure that $kT \ll \hbar\omega$. For $\omega/(2\pi) = 5$ GHz, this means that $T \ll 0.24$ K. In state-of-the-art dilution refrigerators, the temperature of the qubits can be held at 10–15 mK. In this range of temperatures, thermal excitation of 5 GHz qubits can be neglected.

Referring again to Figure 1.12, the quantum processor (QP) containing the qubits is located at the bottom left of the refrigerator. In addition to being kept very close to absolute zero temperature, the quantum processor is also sensitive to stray magnetic fields, so it is further placed inside a magnetic shield within the coldest stage of the refrigerator.

The round component just above the quantum processor is a *circulator*. This is a non-reciprocal microwave component in which energy can only propagate between ports in the direction of the circulating arrow. By non-reciprocal, we mean that the behavior of the component is different if you interchange the input with the output. For example, the RF signal from the control electronics enters the circulator at the top port; the energy “circulates” around to the port to which the quantum processor is connected. Any reflected energy from the quantum processor, e.g., containing information about the state of a qubit, then re-enters the bottom port of the circulator. However, since the circulator is non-reciprocal, instead of returning to the input port on the top side of the circulator, it flows instead to the port on the right and is conveyed to the circulator located in the center. We will return to this center circulator in a moment, but let us first consider the chain of coaxial cables and attenuators conveying the control signal to the first circulator.

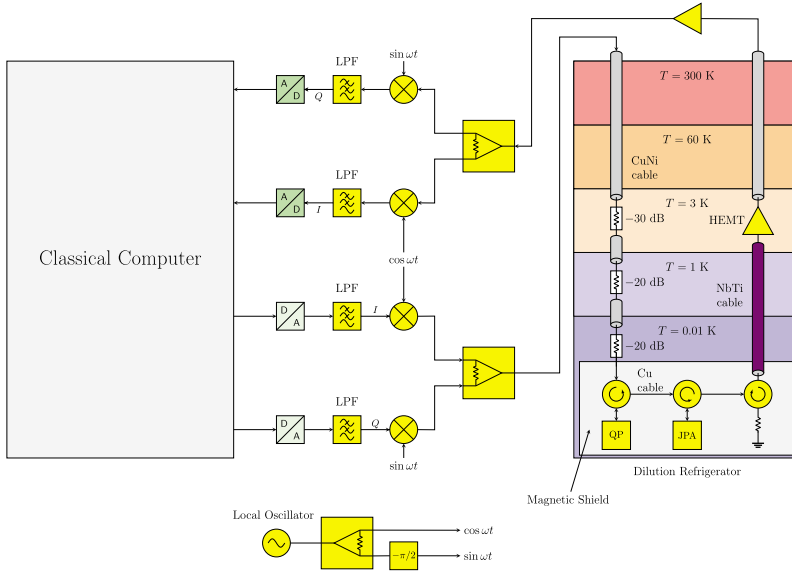


Figure 1.12 System diagram for a superconducting quantum computer.

If we simply used a copper coaxial cable to carry the signal from the room-temperature electronics into the refrigerator, we would have at least two significant problems. First, copper is a good conductor of heat as well as electricity, so the copper cable would convey heat from the upper stages to the lower stages, making it very difficult to reach the temperatures required at the lower stages. To address this, coaxial cables made of an alloy of copper and nickel are used instead. This alloy has very low thermal conductivity to assist with thermally isolating the stages, while having an acceptable electrical conductivity.

The second problem is that a cable coming straight from the outside environment would convey significant noise from the environment into the refrigerator. To combat this, attenuators are placed in the lower stages. These attenuators reduce noise power from the upper stages, but introduce their own noise proportional to their equilibrium temperature. Consequently at the lowest stage, the thermal noise is minimized by the very low temperature of the attenuator. Of course these attenuators also reduce the amplitude of the control signal, so we must make sure that the signal level produced by the signal source is strong enough to produce a satisfactory control signal at the quantum processor.

Returning to the signal reflected from the quantum processor, upon entering the center circulator, the energy is transferred to the bottom port and delivered to a Josephson Junction Parametric Amplifier (JPA). This is a quantum-limited amplifier, meaning that the noise it introduces to the circuit is close to the fundamental minimum allowed by quantum theory. The JPA works in reflection mode, so the amplified reflected signal is returned to the circulator and transferred to the final circulator at the bottom right.

The circulator at the bottom right is operated as an *isolator*. Power applied to the input (left port) is delivered to the output (top port), but any power reflected from

impedance mismatches further down line, e.g., at the input of the HEMT amplifier, will be delivered to the matched load attached to the bottom port. In this way, the very sensitive quantum processor and quantum limited amplifier are isolated from reflected power or noise from the upper stages.

The measured signal at this point is very weak, and we definitely would not want to use a stack of attenuators on the output line! Instead, a coaxial cable made of an alloy of Niobium and Titanium (NbTi) is used to convey the signal to the 3 K stage. NbTi is a Type II superconductor with a transition temperature of about 10 K, so it provides an extremely low loss path to the 3 K stage. At the 3 K stage, there is a more conventional low-noise amplifier (LNA), but made using a high electron mobility transistor (HEMT)—a type of transistor that is known for producing very low noise. The signal from the output of the HEMT LNA is further amplified at room temperature before delivery to the signal processing electronics.

Returning to the control-signal electronics, any band-limited signal centered on the frequency ω can be represented by sine and cosine components:

$$f(t) = I(t) \cos \omega t + Q(t) \sin \omega t, \quad (1.70)$$

where $I(t)$ and $Q(t)$ are “in-phase” and “quadrature” functions whose time variations are slow compared with the period $T = 2\pi/\omega$. Digital samples of I and Q are converted to band-limited analog signals by digital-to-analog converters followed in general by low-pass filters to eliminate high-frequency components resulting from aliasing. The circular components in Figure 1.12 containing “ $\times$ ” are called *mixers*, and produce an output that is simply the product of the two input signals. For example, the bottom-most mixer produces $Q(t) \sin \omega t$ on its output. Similarly, the output of the next-to-bottom mixer is $I(t) \cos \omega t$. The outputs of the two mixers are applied to a power combiner, shown as the square component with two terminals on the left, and one on the right. The output of the combiner is the sum of the two input signals, creating the desired general signal of the form given by Eq. (1.70). This circuit is referred to as an *IQ modulator*.

There are different ways of combining RF signals, but the particular one shown is known as a Wilkinson combiner, or Wilkinson divider. Note that this component is reciprocal, so it can be used either as a combiner or a divider. An attractive feature of the Wilkinson circuit when used as a divider is that if all of the ports are matched, it does not introduce any loss.

The output signal from the quantum processor is applied to a signal processing circuit at the top center of Figure 1.12 that is very similar to the IQ modulator used to generate the control signal. The signal is first split into equal parts using a Wilkinson divider, and the divided signals are multiplied by either $\cos \omega t$ or $\sin \omega t$ before being applied to low-pass filters. The effect of the low-pass filters is to integrate the applied signal, so since the cross term proportional to $\cos \omega t \sin \omega t$ averages to zero, the filter outputs are proportional to either $I(t)$ or $Q(t)$. These signals are then digitized and analyzed by the classical computer.

With this high-level description as motivation, we are now ready to discuss in detail the principles that underlie the hardware and software of superconducting quantum computing.

- Chapters 2 and 3 explore the quantum physics that determine the behavior of qubits and gates. This gives us the tools to understand the fundamentals of quantum states and how they can be manipulated.
- Chapters 4–8 show how qubits are constructed from superconducting devices, how they are coupled to each other with microwave transmission lines, and how they are controlled and measured by external systems.
- Chapters 9 and 10 discuss how imperfections in the systems we build affect the quantum information we are trying to process. In the near term, we need mechanisms to characterize and compensate for errors, while the long-term hope is that we will have a sufficient number of high-quality qubits to correct errors dynamically and sustain long-running, fault-tolerant quantum computations.
- Finally, Chapters 11 and 12 describe the computations that can be accomplished using qubits and gates, and the potential for applications beyond the capabilities of classical computers.

Our hope is to lay a firm foundation for those new to the quantum computing field, whether students or practicing engineers, as a first step toward tackling the many research and engineering challenges that are needed to make large-scale quantum computers a reality.

1.11 Exercises

- 1.1 How many basis states are there for a three-qubit system? Show the vectors for the (computational) basis states.
- 1.2 There are four Bell states that can be created by entangling two qubits. In addition to the state shown in Figure 1.5, the three additional states are listed below. Construct a circuit for generating each state.
 - (a) $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$
 - (b) $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$
 - (c) $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$
- 1.3 Prove the following equivalencies.
 - (a) $HZH = X$
 - (b) $HXH = Z$
 - (c) $HYH = -Y$
 - (d) $\text{CNOT}_{1,0} = H^{\otimes 2} \text{CNOT}_{0,1} H^{\otimes 2}$
 In (d), $\text{CNOT}_{i,j}$ means a CNOT with qubit i as the control and qubit j as the target. $H^{\otimes 2}$ means a Hadamard gate applied to both qubits.
- 1.4 Create a quantum circuit that swaps two qubit states. In other words: $|ab\rangle \mapsto |ba\rangle$. *Hint:* Consider this classical algorithm that swaps two numbers x and y using an exclusive-OR (XOR) instruction.

$$x \leftarrow x \text{ XOR } y$$

$$y \leftarrow y \text{ XOR } x$$

$$x \leftarrow x \text{ XOR } y$$

- 1.5** Suppose we design a superconducting qubit where the energy difference between $|0\rangle$ and $|1\rangle$ is around 10 GHz. What is the temperature needed to minimize the effect of thermal energy on the qubit, assuming that the qubit is in thermal equilibrium with its environment?

- 1.6** Consider the following three-qubit quantum state:

$$|q_2q_1q_0\rangle = \frac{|010\rangle + |101\rangle + |001\rangle + |110\rangle}{2}$$

Is qubit q_2 entangled with the other two qubits? Explain why or why not.

- 1.7** Suppose we have a way of measuring a qubit in the $|+\rangle$ and $|-\rangle$ basis. As a reminder:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

- (a) Given a qubit in the $|0\rangle$ state, what is the probability of measuring $|+\rangle$? (*Hint: Use the Born Rule.*)
 (b) Given a qubit in the $|1\rangle$ state, what is the probability of measuring $|+\rangle$?
 (c) Given a qubit $|R\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$, what is the probability of measuring $|-\rangle$?

- 1.8** Given a qubit

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1+i}{2}|1\rangle$$

- (a) What is the probability of measuring $|0\rangle$?
 (b) What is the probability of measuring $|+\rangle$?
 (c) What is the probability of measuring $|-\rangle$?

2

Physics of Single Qubit Gates

2.1 Requirements for a Quantum Computer

As you might expect, having some way to represent and manipulate qubits is a basic requirement of any quantum computer. In thinking through all of the key requirements, DiVincenzo [3] identified the following five basic requirements:

1. A scalable physical system with well-characterized qubits
2. The ability to initialize the state of the qubits to a simple fiducial state, such as $|000\dots\rangle$
3. Long relevant decoherence times, much longer than the gate operation time
4. A “universal” set of quantum gates
5. A qubit-specific measurement capability

In this Chapter we will discuss aspects of two of these requirements, but a bit out of DiVincenzo’s order:

- We will see that arbitrary manipulations of qubits can be interpreted as rotations, so we begin by discussing rotations—a subtopic of Requirement 4;
- We will then explore the physics of two-state systems needed to represent qubits—a subtopic of Requirement 1.

We will delay discussions of entanglement, measurement, initialization, and decoherence time to subsequent chapters.

2.2 Single Qubit Gates

2.2.1 Rotations

In the last chapter we encountered several types of gates and described them by 2×2 matrices. To understand more generally what gates do to qubits we need to understand *rotations*. We will then see that the gates we have discussed can be viewed as particular rotations.

You may have encountered rotations in a classical physics or mechanics class, so we will begin with a brief review of rotations of classical vectors in 3-space. However, in quantum mechanics the physical state is represented by the *state vector*, so we need to understand how these state vectors are rotated.

For a single qubit the state vector is represented by a 2×1 matrix (or vector), and in other areas of quantum mechanics such state vectors are referred to as *spinors*. The origin of this term comes from the fact that a particle with spin in a magnetic field is one of the simplest two-state systems, and in this context the 2×1 state vectors represent the spin state. However, the concepts are valid for any two-state quantum mechanical system.

2.2.1.1 Classical Rotations

The rotation of a classical vector around the z axis by the angle θ in the counter-clockwise (CCW) direction is illustrated in Figure 2.1. From the geometry of the figure we see that the vector of length r along the x axis rotates into $(r \cos \theta, r \sin \theta, 0)$. Following a similar construction, a vector of length r along the y axis rotates into $(-r \sin \theta, r \cos \theta, 0)$. In general we can write

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (2.1)$$

We refer to the square matrix as the *rotation matrix* for rotations about the z axis:

$$R_z^c(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.2)$$

Here the superscript c indicates a classical physics rotation.

Following similar procedures, rotations about the x axis and y axis are described by the following matrices:

$$R_x^c(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}, \quad (2.3)$$

$$R_y^c(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}. \quad (2.4)$$

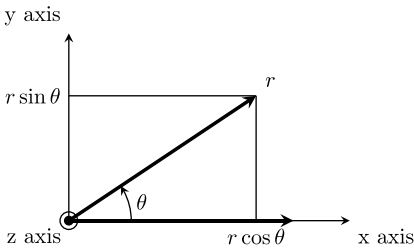


Figure 2.1 Rotation of a vector of length r CCW around the z axis.

2.2.1.2 Rotation of the Quantum Mechanical State Vector

In quantum mechanics, vectors result from taking the expectation value¹ of vector operators for a specific quantum state. For example, to determine the direction of an electron's spin, we take the expectation value of the spin operator using the electron's state vector. To rotate the measured spin vector, we need to rotate the electron *state vector* in such a way that the vector obtained by taking the expectation rotates in the manner we expect. As mentioned above, it is sometimes helpful to refer to electron spin as an example, but our discussion applies to any two-state quantum mechanical system.

For this discussion, consider the general vector operator $\mathbf{A}$:

$$\mathbf{A} = \hat{\mathbf{x}}A_x + \hat{\mathbf{y}}A_y + \hat{\mathbf{z}}A_z, \quad (2.5)$$

where $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ refer to unit vectors in the x , y , and z direction, respectively. A_x refers to the matrix component of $\mathbf{A}$ that operates along the x axis.

For the quantum state $|\psi\rangle$, the expectation of $\mathbf{A}$ is given by $\langle\psi|\mathbf{A}|\psi\rangle$. Now suppose we would like to rotate the system so that $\langle\psi|\mathbf{A}|\psi\rangle$ rotates by an angle θ in the CCW direction around the z axis. We want to find a unitary operator U such that $|\psi'\rangle = U|\psi\rangle$, and

$$\langle\psi'|\mathbf{A}|\psi'\rangle = R_z^c(\theta)\langle\psi|\mathbf{A}|\psi\rangle. \quad (2.6)$$

Substituting in for $|\psi'\rangle$ leads to the following constraint on U

$$\langle\psi|U^\dagger\mathbf{A}U|\psi\rangle = \langle\psi|R_z^c(\theta)\mathbf{A}|\psi\rangle \quad (2.7)$$

$$U^\dagger\mathbf{A}U = R_z^c(\theta)\mathbf{A}. \quad (2.8)$$

Here we have used the fact that $R_z^c(\theta)\langle\psi|\mathbf{A}|\psi\rangle = \langle\psi|R_z^c(\theta)\mathbf{A}|\psi\rangle$, which is easily verified by direct calculation. We have also used the fact that $(U|\psi\rangle)^\dagger = \langle\psi|U^\dagger$.

To simplify our consideration of (2.8), let us consider an infinitesimal rotation ϵ about the axis $\hat{\mathbf{n}}$ (a unit vector along the desired axis of rotation). In this case we anticipate that U will depart only slightly from the identity operator, so to first order in small quantities we can express U with the first two terms in a Maclaurin series:

$$U = I - i\frac{\epsilon}{2}\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}. \quad (2.9)$$

Here $\boldsymbol{\sigma}$ is an arbitrary Hermitian² vector operator that is yet to be determined. The fact that $\boldsymbol{\sigma}$ is Hermitian along with the coefficient i on the second term ensures that U is indeed unitary to first order in ϵ . (We could have absorbed the factor of $-1/2$ into the so-far-unspecified $\boldsymbol{\sigma}$, but including it explicitly as shown will lead us to the definition of $\boldsymbol{\sigma}$ that is most commonly used.)

1 The expectation value is the average outcome of a measurement for the operator.

2 A Hermitian operator $\mathbf{A}$ is one for which $\mathbf{A}^\dagger = \mathbf{A}$.

For the case of a rotation around the z axis, the x component of (2.8) becomes

$$\left(I + i\frac{\epsilon}{2}\sigma_z\right)A_x\left(I - i\frac{\epsilon}{2}\sigma_z\right) = A_x - \epsilon A_y \quad (2.10)$$

$$A_x + i\frac{\epsilon}{2}\sigma_z A_x - i\frac{\epsilon}{2}A_x\sigma_z + \left(\frac{\epsilon}{2}\right)^2\sigma_z A_x\sigma_z = A_x - \epsilon A_y \quad (2.11)$$

$$\sigma_z A_x - A_x\sigma_z = 2iA_y. \quad (2.12)$$

On the right-hand side of (2.10), we have used the small angle approximations to the elements of R_z^c : $\cos \epsilon \approx 1$, $\sin \epsilon \approx \epsilon$. Going from (2.11) to (2.12), we have neglected terms of second order in ϵ .

Similar calculations for the y and z components of (2.8) lead to

$$\sigma_z A_y - A_y\sigma_z = -2iA_x \quad (2.13)$$

$$\sigma_z A_z - A_z\sigma_z = 0. \quad (2.14)$$

Now if A_m, σ_n were ordinary numbers, equations (2.12)–(2.14) would imply that $A_x = A_y = 0$, since ordinary numbers commute. However, if these operators are represented by *matrices*, then more general solutions can exist since matrix multiplication is not commutative. Equations (2.12)–(2.14) are referred to as *commutation* relations, since they specify the difference obtained by reversing the order of the operators. A more compact notation is often used to specify commutation relations:

$$[A, B] \equiv AB - BA. \quad (2.15)$$

Rotations about the x and y axes lead to additional commutation relations which can be written

$$[\sigma_x, A_x] = 0, \quad [\sigma_x, A_y] = i2A_z, \quad [\sigma_x, A_z] = -i2A_y, \quad (2.16)$$

and

$$[\sigma_y, A_x] = -i2A_z, \quad [\sigma_y, A_y] = 0, \quad [\sigma_y, A_z] = i2A_x. \quad (2.17)$$

Since equations (2.12)–(2.14) were obtained without restrictions on the vector operator A , they must also be satisfied if we substitute the vector operator $\mathbf{A} = \boldsymbol{\sigma}$. This leads to the commutation relations for the components of $\boldsymbol{\sigma}$:

$$[\sigma_x, \sigma_y] = i2\sigma_z \quad (2.18)$$

$$[\sigma_y, \sigma_z] = i2\sigma_x \quad (2.19)$$

$$[\sigma_z, \sigma_x] = i2\sigma_y. \quad (2.20)$$

We conclude that for Eq. (2.9) to properly rotate a quantum state vector, the components of the vector operator $\boldsymbol{\sigma}$ must satisfy the commutation relations (2.18)–(2.20). You can easily verify by direct substitution that the following matrices satisfy these commutation relations:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (2.21)$$

In physics these matrices are called the *Pauli matrices*, and they may look vaguely familiar—they are identical to the X, Y, Z gate matrices we encountered in the previous chapter! In quantum computing you will often see these matrices referred to by both notations. The gate notation is commonly used when quantum circuits are being

discussed, while the Pauli notation is often used when the Hamiltonian (or operator for the total energy of a system) of a physical system is being constructed.

In general, four parameters are required to uniquely define a 2×2 matrix. Since the identity matrix along with the three Paulis are linearly independent, we can express any 2×2 matrix A as

$$A = a_0 I + a_1 \sigma_x + a_2 \sigma_y + a_3 \sigma_z. \quad (2.22)$$

So far we have seen how to perform infinitesimal rotations, but what about finite rotations? To make this transition, we note that any two consecutive rotations can be replaced by a single equivalent rotation, as illustrated in Figure 2.2. The figure illustrates rotations about different axes, but let us consider the simpler case of multiple small rotations about the same axis resulting in an equivalent large rotation; in other words we apply the rotation operator N times with $\epsilon \approx \theta/N$ to achieve the large rotation θ . The equivalent unitary operator for the total rotation can be approximated

$$U(\theta) \approx \left(I - i \frac{\theta}{2N} \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \right)^N. \quad (2.23)$$

In the limit $N \rightarrow \infty$, we obtain a closed-form expression for the unitary operator for the rotation of quantum states:

$$U(\theta) = \lim_{N \rightarrow \infty} \left(I - i \frac{\theta}{2N} \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \right)^N \quad (2.24)$$

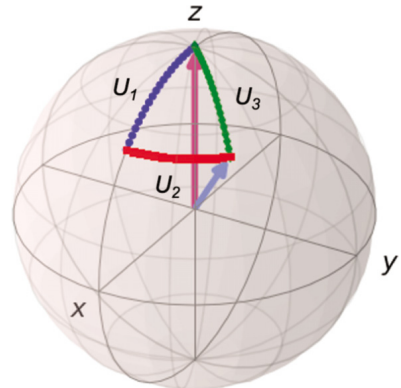
$$= \exp \left(-i \frac{\theta}{2} \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \right). \quad (2.25)$$

It may seem a bit strange to have matrices as arguments to the exponential function, but this is simply defined by the conventional Maclaurin series

$$\exp \left(-i \frac{\theta}{2} \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \right) = \sum_{k=0}^{\infty} \frac{(-i)^k \left[\frac{\theta}{2} \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \right]^k}{k!}. \quad (2.26)$$

Having written this series, with a little additional work we can obtain an alternative form for U that is often useful in quantum computing. First, we separate the even and

Figure 2.2 Illustration of how two consecutive rotations can be replaced with a single equivalent rotation.



odd terms in the series:

$$\exp\left(-i\frac{\theta}{2}\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}\right) = \sum_{p=0}^{\infty} \frac{(-1)^p \left[\frac{\theta}{2}\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}\right]^{2p}}{(2p)!} - i \sum_{q=0}^{\infty} \frac{(-1)^q \left[\frac{\theta}{2}\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}\right]^{2q+1}}{(2q+1)!}. \quad (2.27)$$

Here we have used the observations that $(-i)^{2p} = (-1)^p$ and $(-i)^{2q+1} = -i(-1)^q$. These expressions can be simplified with the aid of the identity

$$(\mathbf{A} \cdot \boldsymbol{\sigma})(\mathbf{B} \cdot \boldsymbol{\sigma}) = (\mathbf{A} \cdot \mathbf{B})I + i(\mathbf{A} \times \mathbf{B}) \cdot \boldsymbol{\sigma}. \quad (2.28)$$

Setting $\mathbf{A} = \mathbf{B} = \hat{\mathbf{n}}$, and using the facts that $\hat{\mathbf{n}}$ is a unit vector and any vector crossed into itself is zero, we conclude that $(\hat{\mathbf{n}} \cdot \boldsymbol{\sigma})^2 = I$. It follows that any even power of $(\hat{\mathbf{n}} \cdot \boldsymbol{\sigma})$ is equal to the identity matrix, and any odd power is simply equal to $(\hat{\mathbf{n}} \cdot \boldsymbol{\sigma})$. These conclusions can be expressed

$$(\hat{\mathbf{n}} \cdot \boldsymbol{\sigma})^{2p} = I, \quad (\hat{\mathbf{n}} \cdot \boldsymbol{\sigma})^{2q+1} = (\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}), \quad (2.29)$$

where p and q are positive integers. These results can be used to further simplify (2.27):

$$\exp\left(-i\frac{\theta}{2}\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}\right) = \sum_{p=0}^{\infty} \frac{(-1)^p \left(\frac{\theta}{2}\right)^{2p}}{(2p)!} - i(\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}) \sum_{q=0}^{\infty} \frac{(-1)^q \left(\frac{\theta}{2}\right)^{2q+1}}{(2q+1)!} \quad (2.30)$$

$$= I \cos \frac{\theta}{2} - i(\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}) \sin \frac{\theta}{2}. \quad (2.31)$$

In the last step we recognized that the first sum is the Maclaurin series for the cosine, and the second sum is the Maclaurin series for the sine function.

Equation (2.31) enables us to easily construct 2×2 matrices for rotations around the principal axes:

$$R_x(\theta) = e^{-i\frac{\theta}{2}\sigma_x} = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \quad (2.32)$$

$$R_y(\theta) = e^{-i\frac{\theta}{2}\sigma_y} = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \quad (2.33)$$

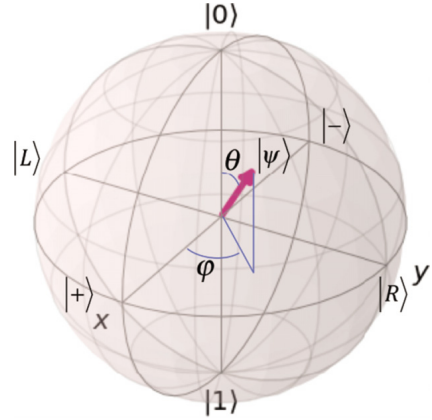
$$R_z(\theta) = e^{-i\frac{\theta}{2}\sigma_z} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \quad (2.34)$$

Note that these have a similar appearance to classical rotation matrices, but the rotation angle enters as $\theta/2$ instead of θ . A consequence of this is that a rotation of 2π does not return to the same state, but rather the negative of the starting state. However, since the measured expectation values are proportional to the magnitude squared of the state vector amplitude, this sign change has no measurable result.

2.2.1.3 Bloch Sphere

Consider a unit vector pointing along the z axis. The vector can be rotated into any other direction represented by a point on a unit sphere with two rotations: first rotate an angle θ about the y axis, then an angle ϕ about the z axis (see Figures 2.2 and 2.3):

Figure 2.3 Representation of a single qubit state on the Bloch Sphere (created in part using [2]).



$$|\psi'\rangle = R_z(\phi)R_y(\theta) |0\rangle \quad (2.35)$$

$$= \begin{bmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{bmatrix} \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (2.36)$$

$$= \begin{bmatrix} e^{-i\phi/2} \cos \frac{\theta}{2} \\ e^{i\phi/2} \sin \frac{\theta}{2} \end{bmatrix} \quad (2.37)$$

$$= e^{-i\phi/2} \cos \frac{\theta}{2} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + e^{i\phi/2} \sin \frac{\theta}{2} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.38)$$

$$= e^{-i\phi/2} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right). \quad (2.39)$$

Since an overall phase will not affect a measurement, we can represent any single qubit state $|\psi\rangle$ as

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle. \quad (2.40)$$

The term *global phase* or *overall phase* is used to describe a phase that affects both basis states. Two quantum states that differ only by a global phase are indistinguishable. The term *local phase* or *relative phase* is used to describe a phase difference between the two basis states. Two quantum states with different local phases are distinctly different states. The usual practice is to associate the phase only with the $|1\rangle$ component, but it's not a requirement; it's the difference in phase that is relevant.

The spherical surface onto which the state vector is mapped using (θ, ϕ) as shown in Figure 2.3 is called the *Bloch Sphere*. Any single qubit state can be represented by a unit vector directed from the origin to a point on the sphere, and the action of a single qubit gate can be visualized as a rotation of the state along the surface of this sphere. You may notice some unexpected aspects of how states are represented on the Bloch Sphere. For example, we know the states $|0\rangle$ and $|1\rangle$ are orthogonal to one another—however, these states are depicted as co-linear on the Bloch Sphere. It is important to keep in mind that this is simply a visualization tool; it is not an accurate depiction of the state vector in a conventional Cartesian vector space.

2.2.1.4 The Most General Unitary

The rotation about y followed by a rotation about z allows you to orient a vector in any direction on the Bloch Sphere as we showed in the previous section. However, in mechanics, there are *three* rotations needed to orient an object arbitrarily, e.g., *roll*, *pitch*, and *yaw*. There are 12 different ways to define the rotations and their order, referred to as *Euler angle conventions*. In the discussion of Euler angle rotations, it is useful to distinguish between the fixed coordinate system, and a mobile coordinate system defined with respect to the object itself. We will use what is referred to as the *ZYZ* convention. In this convention an arbitrary orientation is achieved by the following steps:

- First rotate the system by the angle λ about the z axis of the fixed coordinate system.
- Next rotate the system by the angle θ about the y axis of the fixed coordinate system.
- Finally, rotate the system by the angle ϕ once again about the z axis of the fixed coordinate system.

The *ZYZ* convention is particularly attractive for quantum computing because, as we will see, rotations about the z axis can be experimentally performed with essentially zero error [4].

Since any possible orientation of the state vector can be achieved with this process, the most general unitary operator can be written

$$U(\phi, \theta, \lambda) = R_z(\phi)R_y(\theta)R_z(\lambda) \quad (2.41)$$

$$= e^{-i(\phi+\lambda)/2} \begin{bmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\phi+\lambda)} \cos \frac{\theta}{2} \end{bmatrix}. \quad (2.42)$$

Recognizing again that an overall phase factor is not observable, we can define the most general unitary operator as³

$$u3(\theta, \phi, \lambda) \equiv \begin{bmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\phi+\lambda)} \cos \frac{\theta}{2} \end{bmatrix}. \quad (2.43)$$

Using the appropriate values of (θ, ϕ, λ) , it is straightforward to verify that $u3$ can reproduce all of the gates encountered so far:

$$u3(0, 0, 0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I, \quad (2.44)$$

³ Note that we have listed the angle parameters in a different order from that in which the rotations are applied. This was done so that our definition of $u3$ agrees with the definition of $u3$ in the IBM Quantum system.

$$u3(\pi, 0, \pi) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = X, \quad (2.45)$$

$$u3(\pi, \pi/2, \pi/2) = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = Y, \quad (2.46)$$

$$u3(0, 0, \pi) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Z, \quad (2.47)$$

$$u3(\pi/2, 0, \pi) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = H. \quad (2.48)$$

IBM Qiskit defines two other unitaries as special cases of $u3$:

$$u1(\lambda) = u3(0, 0, \lambda) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix}, \quad (2.49)$$

$$u2(\phi, \lambda) = u3(\pi/2, \phi, \lambda) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{bmatrix}. \quad (2.50)$$

$u1$ allows application of a quantum phase, whereas $u2$ enables the creation of superposition states.

2.2.2 Two State Systems

As we have commented, a qubit can be represented by any two-state quantum system. The simplest two-state system is an electron in a magnetic field. Because the electron has both charge and spin, it also possesses a magnetic moment. So just like a compass in the Earth's magnetic field, the energy depends on the orientation of the spin with respect to the magnetic field, with the lowest energy occurring when the magnetic moment and the magnetic field are lined up. In the following sections we will gain a deeper understanding of the behavior of the spin and how the spin's behavior can be manipulated, and relate this back to qubits and gates.

2.2.2.1 Eigenvalues of the Two State Spin System

The expression for the total energy of a system is referred to in physics as the *Hamiltonian*, represented by the symbol $\mathcal{H}$. One of the fundamental postulates of quantum mechanics is that the state vector must satisfy the *time-dependent Schrödinger equation*:

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = \mathcal{H} |\psi\rangle, \quad (2.51)$$

where $\hbar$ is Planck's constant divided by 2π . For a system in a static electromagnetic field, the time-dependent state vector can be written

$$|\psi(t)\rangle = e^{-i\mathcal{E}t/\hbar} |\psi(0)\rangle. \quad (2.52)$$

Substituting (2.52) into (2.51) gives the *time-independent* Schrödinger equation

$$\mathcal{H} |\psi\rangle = \mathcal{E} |\psi\rangle. \quad (2.53)$$

This is of the form of a standard Eigenvalue equation, where $|\psi\rangle$ is an *eigenvector* associated with the *eigenvalue* $\mathcal{E}$. Multiplying from the left by $\langle\psi|$ shows that the eigenvalue is the expectation value of the operator $\mathcal{H}$ associated with the state vector $|\psi\rangle$:

$$\mathcal{E} = \frac{\langle\psi|\mathcal{H}|\psi\rangle}{\langle\psi|\psi\rangle}. \quad (2.54)$$

For a magnetic moment $\boldsymbol{\mu}$ in a magnetic field $\mathbf{B}$, the Hamiltonian is given by

$$\mathcal{H} = -\boldsymbol{\mu} \cdot \mathbf{B}. \quad (2.55)$$

As we would expect intuitively, this simply says that the energy will be lowest (most negative) when $\boldsymbol{\mu}$ and $\mathbf{B}$ are parallel. The energy of a magnetic moment in a magnetic field described by Eq. (2.55) is referred to as the *Zeeman energy*, named after the Dutch physicist Pieter Zeeman.

If a charged particle were moving in a circle, the motion would give rise to angular momentum given by $\mathbf{L} = \mathbf{r} \times \mathbf{p}$, where $\mathbf{r}$ is the position of the particle, and $\mathbf{p}$ is its momentum. This is referred to as *orbital* angular momentum. Also, if the particle has charge, then the motion constitutes a current loop resulting in a magnetic moment. Since both the angular momentum and the magnetic moment arise from the same particle motion, the magnetic moment and angular momentum are proportional to one another: $\boldsymbol{\mu} = \gamma_L \mathbf{L}$, where $\gamma_L = q/(2m_q)$ is called the *gyromagnetic ratio*, q is the charge of the particle (negative for an electron), and m_q is the mass of the particle.

The particle may also possess *intrinsic* angular momentum in addition to that arising from orbital motion. This intrinsic angular momentum is called *spin* and is denoted $\mathbf{S}$. It behaves similarly to orbital angular momentum, but with several important differences: (a) orbital angular momentum can have values of $n\hbar$ where n is an integer (including zero), while spin angular momentum of a single fundamental matter particle can only take on the values $\pm\hbar/2$ and cannot be zero; (b) the gyromagnetic ratio for orbital angular momentum is $\gamma_L = q/(2m_q)$, while that for spin is $\gamma_S = q/m_q$; and (c) while orbital angular momentum can be quantitatively explained by a circular motion of the particle, no non-relativistic motion can quantitatively explain spin. Consequently it is viewed as an intrinsic property of the particle, just as mass and charge are intrinsic properties.

Although in general the total angular momentum of a particle can have both orbital and spin contributions, for simplicity we consider the case when there is only spin angular momentum, and the charge is positive (e.g., for a proton). With these observations in mind, the Hamiltonian (2.55) can be written

$$\mathcal{H} = -\gamma_S \mathbf{S} \cdot \mathbf{B} = -\frac{q}{m_q} \mathbf{S} \cdot \mathbf{B}, \quad (2.56)$$

where the spin operator $\mathbf{S}$ is given by

$$\mathbf{S} = \frac{\hbar}{2} \boldsymbol{\sigma}, \quad (2.57)$$

$\boldsymbol{\sigma}$ is the Pauli vector we previously encountered:

$$\boldsymbol{\sigma} = \hat{x}\sigma_x + \hat{y}\sigma_y + \hat{z}\sigma_z, \quad (2.58)$$

and σ_j are Pauli matrices given by Eq. (2.21).

Consider the case with the magnetic field directed along the z axis, or $\mathbf{B} = \hat{\mathbf{z}}B_0$. The Hamiltonian can be written

$$\mathcal{H}_0 = -\frac{1}{2}\hbar\omega_0\sigma_z, \quad (2.59)$$

where

$$\omega_0 = \frac{q}{m_q}B_0. \quad (2.60)$$

Using this form of the spin Hamiltonian, the time-independent Schrödinger equation (2.53) becomes

$$\begin{bmatrix} \hbar\omega_0/2 + \mathcal{E} & 0 \\ 0 & \hbar\omega_0/2 - \mathcal{E} \end{bmatrix} |\psi\rangle = 0. \quad (2.61)$$

Setting the determinant equal to zero gives the characteristic equation

$$\left(\frac{\hbar\omega_0}{2} + \mathcal{E}\right)\left(\frac{\hbar\omega_0}{2} - \mathcal{E}\right) = 0 \quad (2.62)$$

which has solutions

$$\mathcal{E} = \pm \frac{\hbar\omega_0}{2}. \quad (2.63)$$

It is easy to verify that the eigenvector for the eigenvalue $\mathcal{E} = -\hbar\omega_0/2$ is $|0\rangle$, and the eigenvector for the eigenvalue $\mathcal{E} = +\hbar\omega_0/2$ is $|1\rangle$.

2.2.2.2 Larmor Precession

Using the basic form from Eq. (2.52), we can write the general solution to the time-varying Schrödinger equation as

$$|\psi(t)\rangle = c_0 e^{i\omega_0 t/2} |0\rangle + c_1 e^{-i\omega_0 t/2} |1\rangle \quad (2.64)$$

$$\equiv c_0 |\psi_0(t)\rangle + c_1 |\psi_1(t)\rangle. \quad (2.65)$$

A particularly interesting case results when we choose $c_0 = \cos(\theta/2)$ and $c_1 = \sin(\theta/2)$:

$$|\psi(t)\rangle = \cos \frac{\theta}{2} e^{i\omega_0 t/2} |0\rangle + \sin \frac{\theta}{2} e^{-i\omega_0 t/2} |1\rangle. \quad (2.66)$$

To interpret this state, let us first find the expectation value⁴ of the x component of the spin operator $S_x = \hbar\sigma_x/2$:

$$\langle S_x \rangle = \langle \psi(t) | \frac{\hbar}{2} \sigma_x | \psi(t) \rangle \quad (2.67)$$

$$= \frac{\hbar}{2} \left(\cos \frac{\theta}{2} e^{-i\omega_0 t/2} \langle 0 | + \sin \frac{\theta}{2} e^{+i\omega_0 t/2} \langle 1 | \right) \quad (2.68)$$

$$\times \sigma_x \left(\cos \frac{\theta}{2} e^{i\omega_0 t/2} |0\rangle + \sin \frac{\theta}{2} e^{-i\omega_0 t/2} |1\rangle \right). \quad (2.69)$$

Multiplying this expression out and using the relations $\sigma_x |0\rangle = |1\rangle$, $\sigma_x |1\rangle = |0\rangle$, and $\langle m|n\rangle = \delta_{m,n}$ where $\delta_{m,n}$ is the Kronecker delta and is equal to 1 if $m = n$ and 0

⁴ The notation $\langle A \rangle$ represents the expectation value of an operator A . If we want to describe the expectation with respect to a specific state $|\psi\rangle$, we often write $\langle A \rangle_\psi = \langle \psi | A | \psi \rangle$.

otherwise, we obtain

$$\langle S_x \rangle = \frac{\hbar}{2} \sin \theta \cos(\omega_0 t). \quad (2.70)$$

Calculating the expectation values of S_y and S_z in a similar way gives

$$\langle S_y \rangle = -\frac{\hbar}{2} \sin \theta \sin(\omega_0 t), \quad (2.71)$$

and

$$\langle S_z \rangle = \frac{\hbar}{2} \cos \theta. \quad (2.72)$$

These equations describe a vector making an angle θ with the z axis, and precessing about it with an angular velocity ω_0 , as shown in Figure 2.4. This motion is known as *Larmor Precession*. It is analogous to the motion of a spinning top in the earth's gravitational field owing to the angular momentum of the top.

From our previous discussion, we know that there must be some analogous rotation of the state vector that gives rise to the precession of the expectation of the spin vector. With this in mind, note that Eq. (2.66) is identical with Eq. (2.39) with $\phi = -\omega_0 t$.

2.2.2.3 Coupled Qubit States

When a two-level system is coupled to a driving field at precisely the frequency corresponding to the energy difference between the states, the system will oscillate between the two states at a frequency determined by the strength of the driving field. These oscillations are called *Rabi* oscillations and are the basic mechanism underlying rotations on the Bloch Sphere. In this section we will examine these oscillations and show how they can be used to realize the basic gates used in quantum computing.

Without such a coupling field, the states $|\psi_0(t)\rangle$ and $|\psi_1(t)\rangle$ are orthogonal eigenvectors of the Hamiltonian, as we have seen. For the case of a relatively weak interaction (i.e., the interaction energy much smaller than the energy difference between the states), we can approach this as a perturbation to the uncoupled system. With this approach, the general state vector is always a superposition of $|\psi_0(t)\rangle$ and $|\psi_1(t)\rangle$, but the precise mixture varies slowly in time on the scale of $1/\omega_0$. The time-varying

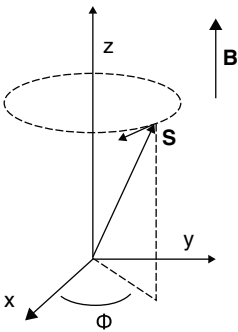


Figure 2.4 Precession of spin vector for a particle with positive charge in a z -directed magnetic field.

coefficients can be found by solving a set of coupled Schrödinger equations:

$$\begin{aligned} i\hbar\partial_t (c_0(t) |\psi_0(t)\rangle) &= \mathcal{H}_0 c_0(t) |\psi_0(t)\rangle + V_{01} |0\rangle\langle 1| c_1(t) |\psi_1(t)\rangle \\ i\hbar\partial_t (c_1(t) |\psi_1(t)\rangle) &= \mathcal{H}_0 c_1(t) |\psi_1(t)\rangle + V_{10} |1\rangle\langle 0| c_0(t) |\psi_0(t)\rangle. \end{aligned} \quad (2.73)$$

Here $V_{01} |0\rangle\langle 1|$ represents the coupling from state $|1\rangle$ to state $|0\rangle$, $V_{10} |1\rangle\langle 0|$ represents the coupling from state $|0\rangle$ to state $|1\rangle$, and we have introduced the short-hand notation $\partial_t \equiv \partial/\partial t$.

Recalling that $|\psi_0\rangle = e^{i\omega_0 t/2} |0\rangle$ and $|\psi_1\rangle = e^{-i\omega_0 t/2} |1\rangle$, the top equation in (2.73), can be written in the somewhat simplified form

$$i\hbar\partial_t (c_0(t) |\psi_0(t)\rangle) = \mathcal{H}_0 c_0(t) |\psi_0(t)\rangle + V_{01} c_1(t) e^{-i\omega_0 t/2} |0\rangle. \quad (2.74)$$

Expanding the derivative on the left-hand-side gives

$$\begin{aligned} i\hbar c_0(t) \partial_t |\psi_0(t)\rangle + i\hbar |\psi_0(t)\rangle \partial_t c_0(t) = \\ \mathcal{H}_0 c_0(t) |\psi_0(t)\rangle + V_{01} c_1(t) e^{-i\omega_0 t/2} |0\rangle. \end{aligned} \quad (2.75)$$

Recognizing that the first terms on both sides of the equation are equal owing to the unperturbed Schrödinger equation, and multiplying from the left by $\langle 0|$ gives

$$\partial_t c_0 = -i \frac{V_{01}}{\hbar} e^{-i\omega_0 t} c_1. \quad (2.76)$$

Following a similar procedure with the second equation leads to

$$\partial_t c_1 = -i \frac{V_{10}}{\hbar} e^{i\omega_0 t} c_0. \quad (2.77)$$

These represent two coupled equations that must be solved simultaneously.

Up to this point, we have not put any constraints on V_{01} and V_{10} . To explore the relationship between these two quantities, recall that the sum of the probabilities that the system is in one state or the other must sum to 1:

$$c_0^* c_0 + c_1^* c_1 = 1. \quad (2.78)$$

It follows that the time derivative of this expression must be zero:

$$\partial_t (c_0^* c_0 + c_1^* c_1) = 0, \quad (2.79)$$

or

$$c_0^* \partial_t c_0 + c_0 \partial_t c_0^* + c_1^* \partial_t c_1 + c_1 \partial_t c_1^* = 0. \quad (2.80)$$

This can be interpreted as requiring “conservation of probability” in the system. Substituting expressions for the derivatives from (2.76) and (2.77) and simplifying gives

$$i c_0 c_1^* e^{i\omega_0 t} (V_{01} - V_{10}) + c.c. = 0, \quad (2.81)$$

where *c.c.* represents the complex conjugate of the first term. We see then that imposing the conservation of probability condition requires $V_{01} = V_{10} \equiv V$. With this condition, the Hamiltonian represented by the right-hand-side of Eq. (2.73) can be written

$$\begin{aligned} \mathcal{H} &= -\frac{1}{2} \hbar \omega_0 \sigma_z + V \sigma_x, \\ &= \mathcal{H}_0 + \mathcal{V}. \end{aligned} \quad (2.82)$$

2.2.3 Creating Rotations: Rabi Oscillations

2.2.3.1 Rotation Operator Approach

Suppose we consider a time-varying coupling term of the form

$$V(t) = \hbar\Omega_R \cos(\omega_d t - \gamma), \quad (2.83)$$

where ω_d and γ are the amplitude and phase, respectively, of an RF drive signal applied to couple the qubit states. The Hamiltonian (2.82) then becomes

$$\mathcal{H} = -\frac{\hbar\omega_0}{2}\sigma_z + \hbar\Omega_R \cos(\omega_d t - \gamma)\sigma_x. \quad (2.84)$$

Recall from our previous discussion that the state vector of a superposition state precesses in a CW direction when viewed from the positive z axis. It is convenient to introduce a coordinate system rotating with the state vector so that the vector appears fixed in the rotating system. We know that the state vector can be rotated about the z axis using the unitary $R_z(\phi)$ (2.34). This is sometimes called an *active* rotation. To rotate the coordinate system instead, we use the inverse unitary, effectively cancelling the apparent rotation of the state. This is referred to as a *passive* rotation. The state vector in the rotating system can therefore be written

$$|\psi(t)\rangle_r = R_z^\dagger(\phi) |\psi(t)\rangle. \quad (2.85)$$

Since Schrödinger's equation must also be satisfied in the rotating frame, let us multiply by $i\hbar$ and take the time derivative:

$$\begin{aligned} i\hbar\partial_t |\psi\rangle_r &= i\hbar\partial_t (R_z^\dagger |\psi\rangle) \\ &= i\hbar\dot{R}_z^\dagger |\psi\rangle + R_z^\dagger (i\hbar\partial_t |\psi\rangle) \\ &= i\hbar\dot{R}_z^\dagger R_z R_z^\dagger |\psi\rangle + R_z^\dagger \mathcal{H} |\psi\rangle \\ &= i\hbar\dot{R}_z^\dagger R_z |\psi\rangle_r + R_z^\dagger \mathcal{H} R_z R_z^\dagger |\psi\rangle \\ &= i\hbar\dot{R}_z^\dagger R_z |\psi\rangle_r + R_z^\dagger \mathcal{H} R_z |\psi\rangle_r \\ &= \mathcal{H}_r |\psi\rangle_r, \end{aligned} \quad (2.86)$$

where

$$\mathcal{H}_r = i\hbar\dot{R}_z^\dagger R_z + R_z^\dagger \mathcal{H} R_z. \quad (2.87)$$

In the above calculations we have used the property $R_z R_z^\dagger = 1$, and the fact that $|\psi\rangle$ satisfies the Schrödinger equation in the laboratory frame. Since the state vector rotation is given by $\phi = -\omega_0 t$,

$$R_z^\dagger(-\omega_0 t) = R_z(\omega_0 t) = e^{-i\frac{\omega_0 t}{2}\sigma_z}. \quad (2.88)$$

The first term of the Hamiltonian in the rotating frame (2.87) can now be written

$$i\hbar\dot{R}_z^\dagger R_z = i\hbar(-i\frac{\omega_0}{2}\sigma_z)R_z^\dagger R_z = +\frac{\hbar\omega_0}{2}\sigma_z. \quad (2.89)$$

To evaluate the remaining terms in the rotating frame Hamiltonian, we have

$$R_z^\dagger(-\omega_0 t)\sigma_z R_z(-\omega_0 t) = \sigma_z, \quad (2.90)$$

and

$$R_z^\dagger(-\omega_0 t) \sigma_x R_z(-\omega_0 t) = \begin{bmatrix} 0 & e^{-i\omega_0 t} \\ e^{i\omega_0 t} & 0 \end{bmatrix}. \quad (2.91)$$

Putting the pieces of the rotating frame Hamiltonian (2.87) together gives

$$\begin{aligned} \mathcal{H}_r &= +\frac{\hbar\omega_0}{2}\sigma_z - \frac{\hbar\omega_0}{2}\sigma_z + \hbar\Omega_R \cos(\omega_d t - \gamma) \begin{bmatrix} 0 & e^{-i\omega_0 t} \\ e^{i\omega_0 t} & 0 \end{bmatrix} \\ &= \frac{\hbar\Omega_R}{2} \begin{bmatrix} 0 & e^{i(\omega_d - \omega_0)t - i\gamma} + e^{-i(\omega_d + \omega_0)t + i\gamma} \\ e^{i(\omega_d + \omega_0)t - i\gamma} + e^{-i(\omega_d - \omega_0)t + i\gamma} & 0 \end{bmatrix}. \end{aligned} \quad (2.92)$$

Normally the drive frequency ω_d will be close to the resonant frequency ω_0 . Consequently this Hamiltonian expression contains frequencies close to zero, and frequencies on the order of twice ω_0 . The high frequency oscillations will tend to average out on the scale of a gate operation, so we neglect these terms going forward. This is called the *rotating wave approximation*. Making this approximation and manipulating the form gives

$$\mathcal{H}_r = \frac{\hbar\Omega_R}{2} \begin{bmatrix} 0 & e^{i(\Delta t - \gamma)} \\ e^{-i(\Delta t - \gamma)} & 0 \end{bmatrix}, \quad (2.93)$$

where $\Delta = \omega_d - \omega_0$. If the drive frequency equals the qubit resonant frequency, then we have

$$\begin{aligned} \mathcal{H}_r &= \frac{\hbar\Omega_R}{2} \begin{bmatrix} 0 & e^{-i\gamma} \\ e^{+i\gamma} & 0 \end{bmatrix} \\ &= \frac{\hbar\Omega_R}{2} (\cos(\gamma)\sigma_x + \sin(\gamma)\sigma_y) \\ &= \frac{\hbar\Omega_R}{2} \hat{\mathbf{n}} \cdot \boldsymbol{\sigma}, \end{aligned} \quad (2.94)$$

where

$$\hat{\mathbf{n}} = \hat{\mathbf{x}} \cos \gamma + \hat{\mathbf{y}} \sin \gamma. \quad (2.95)$$

The time evolution of the state vector in the rotating frame is therefore

$$|\psi(t)\rangle_r = e^{-i\mathcal{H}_r t/\hbar} |\psi(0)\rangle_r = e^{-i\frac{\Omega_R t}{2} \hat{\mathbf{n}} \cdot \boldsymbol{\sigma}} |\psi(0)\rangle_r. \quad (2.96)$$

We see that the unitary operator describing the time evolution is simply a rotation through the angle $\Omega_R t$ about the axis $\hat{\mathbf{n}}$. As an example, for rotations about the x axis, the matrix form of the unitary is

$$e^{-i\frac{\Omega_R t}{2} \sigma_x} = \begin{bmatrix} \cos(\Omega_R t/2) & -i \sin(\Omega_R t/2) \\ -i \sin(\Omega_R t/2) & \cos(\Omega_R t/2) \end{bmatrix}. \quad (2.97)$$

If we start in the ground state $|0\rangle$, the time evolution of the coefficients of the two states is given by

$$\begin{bmatrix} c_0(t) \\ c_1(t) \end{bmatrix} = \begin{bmatrix} \cos(\Omega_R t/2) & -i \sin(\Omega_R t/2) \\ -i \sin(\Omega_R t/2) & \cos(\Omega_R t/2) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\Omega_R t/2) \\ -i \sin(\Omega_R t/2) \end{bmatrix}. \quad (2.98)$$

It follows that the probability of measuring the qubit in one state or the other is given by (upper graph in Figure 2.5):

$$\begin{aligned} P_0 &= |c_0(t)|^2 = (\cos(\Omega_R t/2))^2 = \frac{1}{2} (1 + \cos(\Omega_R t)) \\ P_1 &= |c_1(t)|^2 = (\sin(\Omega_R t/2))^2 = \frac{1}{2} (1 - \cos(\Omega_R t)). \end{aligned} \quad (2.99)$$

2.2.3.2 Rotations about z

Rotations about z are different from the rotations about x, y owing to the relationship between the rotating frame and the precession. One way that a z rotation could be realized is to simply detune ω_d from ω_0 by a small amount. This would cause a relative shift between the state vector and the rotating frame that would increase in time. When the desired phase shift is accumulated, then control pulses return to the condition $\omega_d = \omega_0$.

However, a more precise way to realize a z rotation is simply to control the phase of the generator. We see from (2.94) that the drive signal phase γ enables the placement of the axis of rotation for the Rabi oscillation anywhere in the $x - y$ plane. This effectively implements an arbitrary z rotation. With modern signal generators, the phase can be changed essentially instantaneously and with negligible error [4].

2.2.3.3 Coupled-Mode Theory Approach

Let us return to the coupled equations (2.76) and (2.77), and consider solving them directly as an alternate way to obtain the behavior of the system. Substituting the driving term (2.83) into the first coupled equation (2.76) gives

$$\begin{aligned} \partial_t c_0 &= -i\Omega_R \cos(\omega_d t - \gamma) e^{-i\omega_0 t} c_1 \\ &= -i\frac{\Omega_R}{2} (e^{i(\omega_d t - \gamma)} + e^{-i(\omega_d t - \gamma)}) e^{-i\omega_0 t} c_1 \\ &= -i\frac{\Omega_R}{2} (e^{i(\omega_d - \omega_0)t - i\gamma} + e^{-i(\omega_d + \omega_0)t + i\gamma}) c_1. \end{aligned} \quad (2.100)$$

Noting that on the scale of a cycle of $\omega_d - \omega_0$ the high frequency oscillations from the last term in (2.100) will approximately average to zero, we can once again apply the rotating wave approximation and neglect the high frequency term. The coupled equation (2.100) can then be approximated by

$$\partial_t c_0 = -i\frac{\Omega_R}{2} e^{i\Delta t - i\gamma} c_1, \quad (2.101)$$

where $\Delta = \omega_d - \omega_0$. Following a similar procedure, Eq. (2.77) in the rotating wave approximation becomes

$$\partial_t c_1 = -i\frac{\Omega_R}{2} e^{-i\Delta t + i\gamma} c_0. \quad (2.102)$$

Equations (2.101) and (2.102) can be written as the coupled-mode equations⁵

$$\partial_t c_0 = \kappa_{01} c_1 e^{i\Delta t}, \quad (2.103)$$

$$\partial_t c_1 = \kappa_{10} c_0 e^{-i\Delta t}, \quad (2.104)$$

where

$$\kappa_{01} = -i \frac{\Omega_R}{2} e^{-i\gamma} = -\kappa_{10}^*. \quad (2.105)$$

Solving (2.103) for c_1 and substituting into (2.104) gives a second-order differential equation for c_0 :

$$\ddot{c}_0 - i\Delta \dot{c}_0 + |\kappa|^2 c_0 = 0, \quad (2.106)$$

where $|\kappa_{01}| = |\kappa_{10}| = |\kappa|$, and $|\kappa| = \Omega_R/2$.

Exponential solutions of (2.106) can be obtained from the roots of the corresponding characteristic equation

$$r^2 - i\Delta r + |\kappa|^2 = 0. \quad (2.107)$$

The roots of this equation are

$$\begin{aligned} r_{a,b} &= i \left(\Delta \pm \sqrt{\Delta^2 + 4|\kappa|^2} \right) / 2 \\ &= i (\Delta \pm \Omega) / 2, \end{aligned} \quad (2.108)$$

where

$$\Omega = \sqrt{\Delta^2 + 4|\kappa|^2} = \sqrt{\Delta^2 + \Omega_R^2}. \quad (2.109)$$

Two independent exponential solutions are therefore

$$c_{0a} = e^{i(\Delta/2)t + i\Omega t/2}, \quad c_{0b} = e^{i(\Delta/2)t - i\Omega t/2}. \quad (2.110)$$

Taking the sum and difference of these expressions allows us to obtain alternate independent solutions as

$$c_{0a} = e^{i(\Delta/2)t} \cos(\Omega t/2), \quad c_{0b} = e^{i(\Delta/2)t} \sin(\Omega t/2). \quad (2.111)$$

The general solution can be expressed as an arbitrary linear combination of these two independent solutions. Consequently we can write

$$c_0(t) = e^{i(\Delta/2)t} (A \cos(\Omega t/2) + B \sin(\Omega t/2)). \quad (2.112)$$

The general expression for c_1 can be obtained from (2.103). After some straightforward algebra, the solutions can be expressed in terms of the initial values at $t = 0$:

$$c_0(t) = e^{i(\Delta/2)t} \left[c_0(0) \cos(\Omega t/2) + \frac{2\kappa_{01}c_1(0) - i\Delta c_0(0)}{\Omega} \sin(\Omega t/2) \right], \quad (2.113)$$

$$c_1(t) = e^{-i(\Delta/2)t} \left[c_1(0) \cos(\Omega t/2) + \frac{2\kappa_{10}c_0(0) + i\Delta c_1(0)}{\Omega} \sin(\Omega t/2) \right]. \quad (2.114)$$

⁵ Researchers working with integrated optical devices or microwave circuits and devices may recognize (2.101) and (2.102) from coupled-mode theory; e.g., see [5–7].

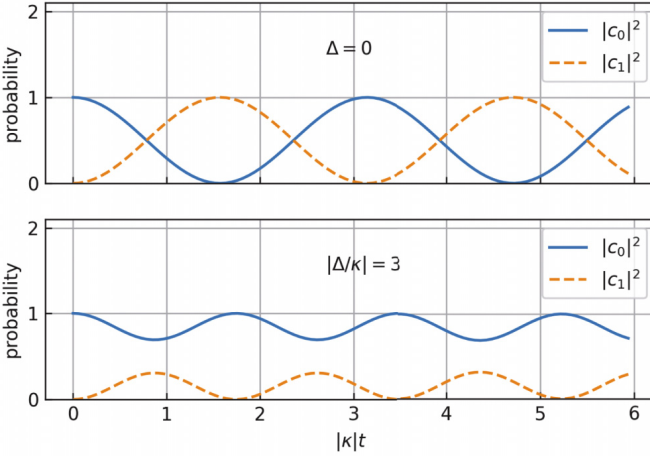


Figure 2.5 Solutions to the coupled mode equations for $\Delta = 0$ and $\Delta/\kappa = 3$. For the case of Rabi oscillations, $|\kappa t| = \Omega_R t/2$.

As an example, consider the case $c_0(0) = 1$, $c_1(0) = 0$. The probabilities of obtaining the states are

$$|c_0(t)|^2 = \cos^2(\Omega t/2) + \left(\frac{\Delta}{\Omega}\right)^2 \sin^2(\Omega t/2), \quad (2.115)$$

$$|c_1(t)|^2 = \left(\frac{2|\kappa|}{\Omega}\right)^2 \sin^2(\Omega t/2). \quad (2.116)$$

These solutions are plotted for the two cases $\Delta = 0$ and $\Delta/\kappa = 3$ in Figure 2.5. Note that significant coupling only occurs when $\Delta \approx 0$.

2.3 Quantum State Tomography

Suppose we have executed several rotation commands on a qubit by applying the appropriate RF pulses. How can we verify if we obtained the state we intended? Referring to Figure 2.3, we could determine the state $|\psi\rangle$ if we could measure the projection of the state on each axis. As we have discussed, making a measurement on the state will collapse the state onto either $|0\rangle$ or $|1\rangle$ with probabilities determined by the squares of the magnitudes of the amplitudes of these components of the state. To estimate these probabilities, we need to prepare and measure the same state a large number of times. The percentage of times we obtain $|0\rangle$ allows us to estimate this probability, and similarly for $|1\rangle$. This suggests that we could obtain the angle θ from the calculation

$$\cos \theta = P_0(|\psi\rangle) - P_1(|\psi\rangle), \quad (2.117)$$

where $P_0(|\psi\rangle)$ is the probability of obtaining $|0\rangle$ and $P_1(|\psi\rangle)$ is the probability of obtaining $|1\rangle$ when the state $|\psi\rangle$ is measured. But if we can only measure the projections along the z axis, how do we determine the projections along x and y ? We can do this by rotating the state vector so that the component we want to measure is along z . For example, consider a rotation of π around the axis $\hat{\mathbf{n}} = (\hat{\mathbf{x}} + \hat{\mathbf{y}})/\sqrt{2}$, as shown in Figure 2.6(a).

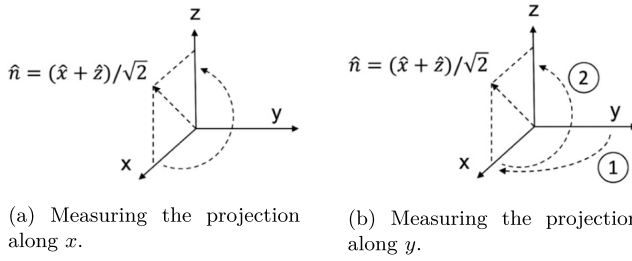


Figure 2.6 Rotations enabling measurement of the projections of the state vector along the x and y axes. (a) The Hadamard gate corresponds to rotation of the state vector around an axis in the x - z plane making a 45° angle with the z axis. This rotates the x component to z axis. (b) z rotation of -90° followed by a Hadamard rotation to estimate the projection along y .

From (2.31) we have

$$\begin{aligned}
 U_n(\pi) &= I \cos \frac{\pi}{2} - i \frac{1}{\sqrt{2}} (\sigma_x + \sigma_z) \sin \frac{\pi}{2} \\
 &= -i \frac{1}{\sqrt{2}} (\sigma_x + \sigma_z) \\
 &= -i \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\
 &= -iH.
 \end{aligned} \tag{2.118}$$

Apart from a global phase, we see that this rotation is represented simply by the Hadamard gate. To see why this rotation is of interest, recall that the state represented by a vector along x is $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, and the state represented by a vector along $-x$ is $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Applying a Hadamard gate to these states gives

$$H|+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle, \tag{2.119}$$

and

$$H|-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle. \tag{2.120}$$

We see then that the Hadamard gate rotates the state vector so that its x component lies along the z axis. Subsequently making measurements enables the component of the state vector along x to be estimated. Thus we can write

$$\sin \theta \cos \phi = P_0(H|\psi\rangle) - P_1(H|\psi\rangle). \tag{2.121}$$

Finally, we can estimate the projection along the y axis by first rotating the y axis into the x axis using a z rotation of $-\pi/2$, followed by a Hadamard gate to again rotate x into z as shown in Figure 2.6(b). Note that the z rotation operator can be written

$$R_z(\phi) = e^{-i\phi/2} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \equiv e^{-i\phi/2} P(\phi), \tag{2.122}$$

where $P(\phi)$ is called a *phase gate*, and introduces a phase shift of ϕ between the states $|0\rangle$ and $|1\rangle$.⁶ When written this way it is easy to see that a Z gate is created by a rotation of π (cf. (2.47)). At this point it is convenient to introduce another gate, $S = Z^{1/2}$:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \quad (2.123)$$

In terms of this gate, we see that a z rotation of $-\pi/2$ is represented by $S^\dagger$:

$$P(-\pi/2) = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} = S^\dagger. \quad (2.124)$$

We conclude that to estimate the projection along the y axis, we apply the operator $HS^\dagger$ followed by a measurement. Similar to before we can therefore write

$$\sin \theta \sin \phi = P_0(HS^\dagger |\psi\rangle) - P_1(HS^\dagger |\psi\rangle), \quad (2.125)$$

and the azimuth angle ϕ on the Bloch sphere can be estimated from⁷

$$\tan \phi = \frac{P_0(HS^\dagger |\psi\rangle) - P_1(HS^\dagger |\psi\rangle)}{P_0(H |\psi\rangle) - P_1(H |\psi\rangle)}. \quad (2.126)$$

Having estimated both θ and ϕ enables us to locate the state vector on the Bloch sphere, and we can express the state vector in the z basis using (2.40).

2.4 Expectation Values and the Pauli Operators

In Section 2.2.2.2 we considered the case where the state vector represented the spin of a particle. To find the expectation values of the x, y, z components of the spin, we applied the relevant Pauli operator to the state vector. In other words, to obtain the expected value of the x component we calculated $\langle \psi | \sigma_x | \psi \rangle$, and similarly for y and z .

Based on this, one might think that to construct a quantum circuit to measure the x component of a qubit state vector, it would be necessary to apply σ_x (i.e., an X gate) to the qubit of interest prior to measurement. However, from the previous section we see that simply preparing a state multiple times and taking the average of the measurements yields an estimate for the z component. Similarly, to estimate the x or y components, it is simply necessary to perform suitable rotations to align the desired component with the z axis, then again prepare and measure the state multiple times to estimate the average. Said differently, $\langle \psi | \sigma_x | \psi \rangle$ represents mathematical operations that yield the expected value of the x component of spin—or more generally of the state vector. But by definition, (an approximation to) the expectation value is obtained by taking a large number of measurements of the x component and averaging them. Since in quantum computing we actually take measurements, we obtain estimates of the state vector components by taking the averages of large numbers of measurements, rather than doing calculations involving the Pauli matrices.

⁶ For a single qubit operation, the difference between R_z and P is a global phase, and is inconsequential. However, in multi-qubit states, R_z and P may not be equivalent.

⁷ Note that a 4-quadrant arctangent function should be used to invert this relation, in general.

As an example, suppose we prepared and measured an identical state N times (each time is referred to as a “shot”). Out of these shots, $|0\rangle$ is measured N_0 times, and $|1\rangle$ is measured N_1 times, where $N_0 + N_1 = N$. We obtain an estimate of the z component of the state as follows:

$$\langle\psi|\sigma_z|\psi\rangle = \cos\theta = P_0(|\psi\rangle) - P_1(|\psi\rangle) \approx \frac{N_0 - N_1}{N}. \quad (2.127)$$

Clearly the more shots we execute (larger N), the better estimate of the z component we obtain. To make this more quantitative, we note that by the Central Limit Theorem, as N becomes large, the variance of the estimate is σ^2/N , where the variance of an individual measurement is σ^2 . Since the measurement can only yield ± 1 , we can estimate an upper limit on the error as the sample standard deviation $1/\sqrt{N}$. These ideas are explored in more detail in Exercise 2.1.

2.5 Density Matrix

Before we move to more complex qubit interactions, it is useful to introduce another representation of a quantum state, known as a *density matrix* or *density operator*.⁸ The density matrix is a generalization of the state vector representation. It can represent a larger set of possible quantum states, including those that do not lie on the surface of the Bloch sphere.

Until now, we have only considered states that lie on the surface of the Bloch sphere, so let's begin there. Such a state is known as a *pure state*. As we know, it can be precisely represented as a two-element state vector $|\psi\rangle$, and the density matrix ρ is defined as:

$$\rho \equiv |\psi\rangle\langle\psi|. \quad (2.128)$$

Rather than a two-element vector, ρ is a 2×2 -element matrix. Following the definition, here are the density matrices for some states that we have encountered before:

$$|0\rangle\langle 0| = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.129)$$

$$|1\rangle\langle 1| = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.130)$$

$$|+\rangle\langle +| = \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix} \right) = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (2.131)$$

$$|-\rangle\langle -| = \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right) \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \end{bmatrix} \right) = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \quad (2.132)$$

⁸ Density operator is a more general term, as the density matrix depends on the choice of basis vectors for the state space. Since we restrict ourselves to the standard basis, we will use the terms interchangeably.

Note that basis states ($|0\rangle$ and $|1\rangle$) have no off-diagonal entries, but superposition states ($|+\rangle$ and $|-\rangle$) do. Also note that the diagonal entries ρ_{jj} represent the probability of measuring state $|j\rangle$ in the standard basis. Therefore the sum of the diagonal entries, also known as the *trace* of the matrix, will always be 1.

$$\text{Tr}(\rho) = \sum_j \rho_{jj} = 1. \quad (2.133)$$

For a general state $|\psi\rangle = \cos \frac{\theta}{2} + e^{i\phi} \sin \frac{\theta}{2}$,

$$|\psi\rangle\langle\psi| = \begin{bmatrix} \cos^2 \frac{\theta}{2} & e^{-i\phi} \sin \frac{\theta}{2} \cos \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} \cos \frac{\theta}{2} & \sin^2 \frac{\theta}{2} \end{bmatrix}. \quad (2.134)$$

One feature of the density matrix is that it can also represent quantum states that are not on the surface of the Bloch sphere, and thus cannot be defined by a state vector. Such states are known as *mixed states*. One example of a mixed state is the description of the state of one qubit that is part of an entangled group of qubits. We will defer that discussion until after we discuss two-qubit states and gates in more detail.

The second example is when the state is described as a probabilistic *ensemble* of pure states, each with a specified probability. For example, suppose we prepare a qubit in the $|0\rangle$ state, but there is a 10% probability that our preparation is faulty, and the qubit actually ends up in the $|1\rangle$ state. (This is very bad hardware that should probably be retired.) If we want to model the uncertainty in that qubit's state, we use the following definition of the density matrix:

$$\rho \equiv \sum_i p_i |\psi_i\rangle\langle\psi_i|, \quad (2.135)$$

where p_i is the probability of being in state $|\psi_i\rangle$. For our particular example,

$$|\psi_0\rangle = |0\rangle, p_0 = 0.90 \quad (2.136)$$

$$|\psi_1\rangle = |1\rangle, p_1 = 0.10 \quad (2.137)$$

$$\begin{aligned} \rho &= 0.90 |0\rangle\langle 0| + 0.10 |1\rangle\langle 1| \\ &= \begin{bmatrix} 0.90 & 0 \\ 0 & 0.10 \end{bmatrix}. \end{aligned} \quad (2.138)$$

We can distinguish a pure state from a mixed state using the trace of ρ^2 :

$$\text{Tr}(\rho^2) = \begin{cases} 1 & \text{if a pure state} \\ < 1 & \text{if a mixed state.} \end{cases} \quad (2.139)$$

You can verify that this is true for the pure and mixed states shown above.

With respect to the Bloch sphere, a mixed state lies inside the surface; a vector from the origin to the point representing the mixed state is not a unit vector. Any density

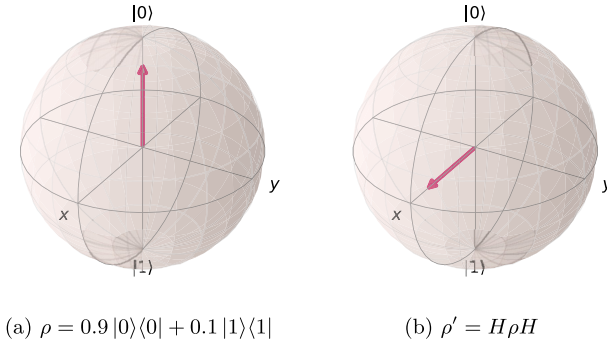


Figure 2.7 (a) Bloch sphere representation of a mixed state, and (b) the result of applying a Hadamard gate (H) to the mixed state. Plots generated using Qiskit [2].

matrix can be written in the following form:

$$\rho = \frac{1}{2} \begin{bmatrix} 1+z & x-iy \\ x+iy & 1-z \end{bmatrix} \quad (2.140)$$

$$= \frac{1}{2} (I + x\sigma_x + y\sigma_y + z\sigma_z), \quad (2.141)$$

where x , y , and z are the Cartesian coordinates in the Bloch sphere representation. (For the proof, see Rieffel and Polak [8], Section 10.1.3.) Using our simple mixed-state example above, the Bloch vector coordinates are $(0, 0, 0.8)$, shown in Figure 2.7, which is clearly not a unit vector.

How does the density matrix change when a unitary U is applied to a quantum state? Each state $|\psi_i\rangle$ in the ensemble becomes $U|\psi_i\rangle$, so the new density matrix is

$$\rho' = \sum_i p_i U |\psi_i\rangle\langle\psi_i| U^\dagger = U\rho U^\dagger. \quad (2.142)$$

As with the state vector, the unitary does not change the length of the Bloch vector associated with the state, only its orientation.

To illustrate, let's apply the Hadamard gate to our example mixed state, recalling that $H = H^\dagger$.

$$\begin{aligned} \rho' &= H\rho H \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0.90 & 0 \\ 0 & 0.10 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0.90 & 0.90 \\ 0.10 & -0.10 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.4 \\ 0.4 & 0.5 \end{bmatrix}. \end{aligned} \quad (2.143)$$

The Bloch coordinates for ρ' are $(0.8, 0, 0)$ —the x and z coordinates have been swapped, as expected.

Finally, we want to be clear to distinguish a mixed state from a pure superposition state. Because the notion of probabilities factors into both, there can be some confusion. The following example illustrates the difference.

First, consider a pure state defined by a state vector:

$$|\psi\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle. \quad (2.144)$$

Next, consider the state represented by a density matrix:

$$\begin{aligned} \rho &= \frac{1}{2} |0\rangle\langle 0| + \frac{1}{2} |1\rangle\langle 1| \\ &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}. \end{aligned} \quad (2.145)$$

If we measure $|\psi\rangle$, we have a 50% probability of observing $|0\rangle$ and a 50% chance of observing $|1\rangle$. The same is true if we measure ρ : 50% $|0\rangle$ and 50% $|1\rangle$. So are the two states the same?

Suppose we use state tomography (Section 2.3) to learn more about the state. We measure $|\psi\rangle$ in the x basis by first computing $H|\psi\rangle$. This time we observe $|0\rangle$ 100% of the time, because $H|\psi\rangle = |0\rangle$. Then we measure $H\rho H$, and we see 50% $|0\rangle$ and 50% $|1\rangle$. Why? First, we observe that $H\rho H = \rho$, so the state has not changed. Second, we consider that the mixed state is $|0\rangle$ with 50% probability—measuring $H|0\rangle = |+\rangle$ will yield 50% $|0\rangle$ and 50% $|1\rangle$. With 50% probability, the mixed state is $|1\rangle$, and measuring $H|1\rangle = |-\rangle$ will also yield 50% $|0\rangle$ and 50% $|1\rangle$. Since this result is very different, it is clear that the two states are not the same.

We can also compute the Bloch sphere coordinates for each state, which again makes it clear that these are very different. The coordinates for $|\psi\rangle$ are $(1, 0, 0)$, while the coordinates for ρ are $(0, 0, 0)$. Note that any rotation of ρ will not change its coordinates, so measuring in any basis will result in the same 50–50 distribution of results. This is known as a *maximally* mixed state, because we have no information about the state along any axis. As a matter of fact, there are a number of different ensembles of pure states that can lead to the same density matrix, which will be made clear in Exercise 2.5.

While there is uncertainty in the measurement of either state, the kinds of uncertainty are different. Measurement of $|\psi\rangle$ exhibits *quantum uncertainty*; we are uncertain of the result because of the intrinsic nature of the quantum state and the nature of measurement. The density matrix, however, represents *classical uncertainty* about which quantum state was prepared.

2.6 Exercises

2.1 In this exercise we explore the statistics of the expected value obtained by executing a large number of “shots” on a quantum circuit. We note that the result of a measurement is either $|0\rangle$ or $|1\rangle$, which we can represent by $+1$ and -1 , respectively. A random variable Z with two possible values is an example of a random variable with a two-point distribution, and the value of Z from the i th measurement we denote z_i . According to the Central Limit Theorem, the distribution of $\sum_i z_i/N$ tends toward a Gaussian (Normal) distribution for large

N , with variance σ^2/N , where σ^2 is the variance of Z . In this exercise we will find an upper bound on the value of σ^2 , and hence an upper bound on the standard deviation estimate $\sigma/\sqrt{N}$.

- (a) Let $\Pr(Z = 1) = p$, and $\Pr(Z = -1) = 1 - p$. Find $E[Z^2]$ and $E[Z]$.
- (b) Find the variance $\text{Var}[Z] = E[Z^2] - E[Z]^2$.
- (c) Show that the maximum value of the variance of Z is 1, so an upper bound on the standard deviation of $\sum_i z_i/N$ is $\sigma \leq 1/\sqrt{N}$.

2.2 A particular state $|\psi\rangle$ has been prepared a large number of times, and the following estimated probabilities are obtained from measurements: $P_0(|\psi\rangle) = 0.8$, $P_1(|\psi\rangle) = 0.2$, $P_0(H|\psi\rangle) = 0.7$, $P_1(H|\psi\rangle) = 0.3$, $P_0(HS^\dagger|\psi\rangle) = 0.6$, $P_1(HS^\dagger|\psi\rangle) = 0.4$. Find estimates of the angles θ, ϕ locating the state on the Bloch Sphere.

2.3 Show that $XYX = -Y$ and use this to prove that $XR_y(\theta)X = R_y(-\theta)$ [9].

2.4 Consider the following definitions:

$$A \equiv R_z(\phi)R_y(\theta/2)$$

$$B \equiv R_y(-\theta/2)R_z(-(\lambda + \phi)/2)$$

$$C \equiv R_z((\lambda - \phi)/2)$$

- (a) Show that $ABC = I$.
- (b) Show that $AXBXC = R_z(\phi)R_y(\theta)R_z(\lambda)$. The purpose of this exercise is to show that any single-qubit unitary U can be implemented using $AXBXC$ with appropriate choices of ϕ, θ , and λ , up to a global phase.

2.5 Consider these two pure states:

$$|a\rangle = \sqrt{0.90}|0\rangle + \sqrt{0.10}|1\rangle, \quad |b\rangle = \sqrt{0.90}|0\rangle - \sqrt{0.10}|1\rangle$$

Compute the density matrix $\rho = \frac{1}{2}|a\rangle\langle a| + \frac{1}{2}|b\rangle\langle b|$. Compare to Eq. (2.138). Does a density matrix uniquely correspond to a particular ensemble of pure states?

3

Physics of Two Qubit Gates

As mentioned in Chapter 1, distinguishing characteristics of quantum computing include the use of the quantum mechanical phenomena of superposition and entanglement. The simplest manifestation of superposition is the case of a single qubit in a state with non-zero probabilities of getting either a 0 or a 1 when measured. We explored the physics underlying the generation of such single-qubit gates in Chapter 2.

The simplest manifestation of entanglement is a two-qubit state that cannot be expressed as a product of single-qubit states. The Bell State described in Section 1.6 is a classic example.

The CNOT gate that we used to create the Bell state combined with single-qubit gates comprise a universal gate set—i.e., a set of gates capable of doing any arbitrary quantum computation¹ [10]. Further, it turns out that *any* two-qubit entangling gate along with single-qubit gates is universal [11]. Consequently a number of entangling schemes have been proposed. We will concentrate on two: (1) coupled qubits that have the same frequency, and (2) coupled qubits with different frequencies where a signal at the frequency of one is applied to the other. This latter scheme is referred to as *cross-resonant entanglement*.

3.1 $\sqrt{i}$ SWAP Gate

It is helpful to begin this discussion by defining the SWAP gate and two of its cousins. As the name implies, a SWAP gate simply interchanges the states on two qubits. In other words, it makes the following transformations: $|00\rangle \rightarrow |00\rangle$, $|01\rangle \rightarrow |10\rangle$, $|10\rangle \rightarrow |01\rangle$, and $|11\rangle \rightarrow |11\rangle$. It is represented by the matrix

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.1)$$

¹ Since any quantum computation can be represented by a unitary matrix, an equivalent definition of a universal gate set is a set capable of representing any arbitrary unitary matrix.

The i SWAP gate is similar, except when the qubit states are different: $|01\rangle \rightarrow i|10\rangle$, $|10\rangle \rightarrow i|01\rangle$. Its matrix representation is

$$i\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.2)$$

The reason for introducing the SWAP and i SWAP gates is really to motivate the $\sqrt{i\text{SWAP}}$:

$$\sqrt{i\text{SWAP}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & i/\sqrt{2} & 0 \\ 0 & i/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.3)$$

As the name implies, it is straightforward to verify that $\sqrt{i\text{SWAP}} \cdot \sqrt{i\text{SWAP}} = i\text{SWAP}$. The common symbols for these gates are shown in Figure 3.1.

As it turns out, a two-qubit gate very similar to the i SWAP occurs naturally when two superconducting qubits are coupled with a capacitor [12]. This “natural” gate is the same as the i SWAP except for the sign on the i terms.² We will call this gate $i\text{SWAP}'$.

Its square root is given by

$$\sqrt{i\text{SWAP}'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & -i/\sqrt{2} & 0 \\ 0 & -i/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.4)$$

This is an entangling gate, since if we start off with either $|01\rangle$ or $|10\rangle$, we end up with a superposition of the two states: a Bell state. It’s not pretty, but the $\sqrt{i\text{SWAP}'}$ can be transformed into a CNOT (within an irrelevant phase factor) by the following operations (see Exercise 3.1):

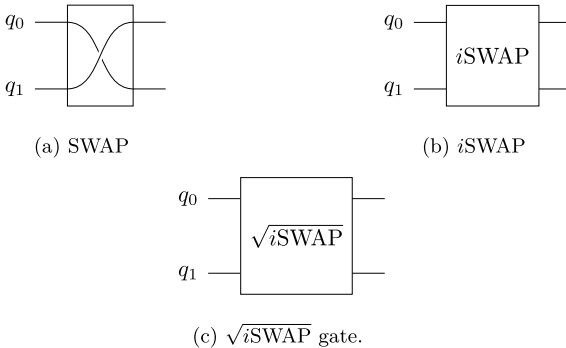


Figure 3.1 Common symbols for the SWAP, i SWAP, and $\sqrt{i\text{SWAP}}$ two-qubit gates.

² Some authors, in fact, define the i SWAP with the negative signs in i ; e.g., see [12, 13].

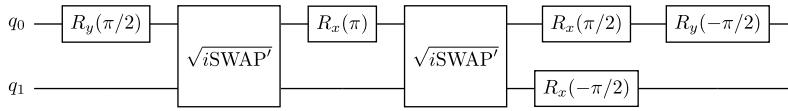


Figure 3.2 Operations needed to convert a $\sqrt{i\text{SWAP}}$ gate to a CNOT.

$$\begin{aligned} \text{CNOT} = & [I \otimes R_y(-\pi/2)][R_x(-\pi/2) \otimes R_x(\pi/2)]\sqrt{i\text{SWAP}}' \\ & \cdot [I \otimes R_x(\pi)]\sqrt{i\text{SWAP}}'[I \otimes R_y(\pi/2)]. \end{aligned} \quad (3.5)$$

The corresponding circuit is shown in Figure 3.2. Note that the order of the operations is opposite that shown in (3.5) since mathematically the operations begin by applying the right-most operator to the state vector, while in the circuit the first operation appears on the left with time increasing to the right.

3.2 Coupled Tunable Qubits

We now turn our attention to the physics leading to the $\sqrt{i\text{SWAP}}$ gate. We begin by considering the case of two coupled qubits with different frequencies, but that can be tuned to the same frequency.³ We will see that this system naturally generates entangled states that can realize the $\sqrt{i\text{SWAP}}$ gate [12].

If the qubits are uncoupled, then the system state is simply represented by the product of the single-qubit states:

$$|\psi\rangle_2 = |\psi_A\rangle \otimes |\psi_B\rangle \quad (3.6)$$

where $|\psi_A\rangle$ and $|\psi_B\rangle$ satisfy

$$i\hbar\partial_t |\psi_{A,B}\rangle = \mathcal{H}_{A,B} |\psi_{A,B}\rangle \quad (3.7)$$

and the single-qubit Hamiltonians are

$$\mathcal{H}_{A,B} = -\frac{1}{2}\hbar\omega_{A,B}\sigma_z. \quad (3.8)$$

Because the qubits are uncoupled, we could (for example) put qubit A in the state $|1\rangle$ and qubit B in the state $|0\rangle$ or vice versa, and the system would remain in this state for all time.

Now consider what would happen if there were some mechanism coupling the two qubits. If the frequencies were very different, then energy conservation would keep them in the initial states regardless of the coupling.⁴

On the other hand, if the frequencies were equal, then there is no fundamental reason why a photon on qubit A could not make a transition to qubit B (assuming B is in the ground state) in the presence of some coupling mechanism. In contrast, if both qubits were in the ground state or both were in the excited state, then the states would again remain stationary in time. In the first case there would be no energy to transfer,

³ We will see in Section 8.3 that tuning can be realized by applying a magnetic field to a qubit made using two Josephson junctions configured as a superconducting quantum interference device (SQUID).

⁴ Here we are neglecting transitions owing to radiation or any other mechanism. If the only states available have energies $\hbar\omega_A$ and $\hbar\omega_B$, then we would have to either create or dispose of energy to make the transition if $\omega_A \neq \omega_B$.

and in the second case one might consider an exchange of photons, but since photons are indistinguishable, the state must remain the same. We can summarize the possible transitions as follows:

$$|\psi_{00}\rangle \rightarrow |\psi_{00}\rangle, \quad |\psi_{01}\rangle \leftrightarrow |\psi_{10}\rangle, \quad |\psi_{11}\rangle \rightarrow |\psi_{11}\rangle, \quad (3.9)$$

where we have introduced the shorthand $|\psi_0\rangle \otimes |\psi_1\rangle = |\psi_{01}\rangle$ with the state of qubit A on the left and qubit B on the right. Since the states $|\psi_{00}\rangle$ and $|\psi_{11}\rangle$ don't really do anything interesting, let's concentrate on what happens to states $|\psi_{01}\rangle$ and $|\psi_{10}\rangle$.

Taking the energies of the states $|\psi_0\rangle$ and $|\psi_1\rangle$ to be $-\hbar\omega/2$ and $\hbar\omega/2$, respectively, the state $|\psi_{01}\rangle$ must satisfy the Schrödinger equation

$$\begin{aligned} i\hbar\partial_t |\psi_{01}\rangle &= [\mathcal{H}_A \otimes I + I \otimes \mathcal{H}_B] |\psi_{01}\rangle \\ &= \frac{\hbar}{2}(-\omega_A + \omega_B) |\psi_{01}\rangle \end{aligned} \quad (3.10)$$

while the state $|\psi_{10}\rangle$ must satisfy

$$i\hbar\partial_t |\psi_{10}\rangle = \frac{\hbar}{2}(\omega_A - \omega_B) |\psi_{10}\rangle. \quad (3.11)$$

In the presence of coupling, the state vector will not satisfy either of these equations, but instead an equation with a Hamiltonian modified by an interaction term.

If the coupling is weak, it is reasonable to guess that the state at an arbitrary time will look like a combination of these two uncoupled states. Since we suspect that the system may transition between the states $|\psi_{10}\rangle$ and $|\psi_{01}\rangle$, let's make an educated guess that an approximation to the state might look like a sum of these states with time-varying coefficients, similar to what we did when describing Rabi oscillations in Section 2.2.2.3:

$$|\psi\rangle = c_{01}(t) |\psi_{01}\rangle + c_{10}(t) |\psi_{10}\rangle. \quad (3.12)$$

In this case the two Schrödinger equations need to be modified by adding terms that couple the equations:

$$i\hbar\partial_t (c_{01}(t) |\psi_{01}\rangle) = \frac{\hbar}{2}(-\omega_A + \omega_B)c_{01}(t) |\psi_{01}\rangle + \kappa |01\rangle\langle 10| c_{10}(t) |\psi_{10}\rangle \quad (3.13)$$

$$i\hbar\partial_t (c_{10}(t) |\psi_{10}\rangle) = \frac{\hbar}{2}(\omega_A - \omega_B)c_{10}(t) |\psi_{10}\rangle + \kappa |10\rangle\langle 01| c_{01}(t) |\psi_{01}\rangle. \quad (3.14)$$

Here we have introduced κ as a phenomenological constant to describe the coupling.

Proceeding as we did in Section 2.2.2.3, we see that applying the product rule to the time derivative on the left-hand-side generates two terms, one of which cancels the first term on the right-hand-side. This leaves

$$i\hbar\partial_t (c_{01}(t)) |\psi_{01}\rangle = \kappa c_{10}(t) |01\rangle\langle 10| \psi_{10}\rangle \quad (3.15)$$

$$i\hbar\partial_t (c_{10}(t)) |\psi_{10}\rangle = \kappa c_{01}(t) |10\rangle\langle 01| \psi_{01}\rangle. \quad (3.16)$$

Recall that

$$|\psi_{01}\rangle = e^{i\hbar(\omega_A - \omega_B)t/2} |01\rangle$$

and

$$|\psi_{10}\rangle = e^{-i\hbar(\omega_A - \omega_B)t/2} |10\rangle.$$

Making this substitution and multiplying the first and second equations from the left by $\langle 01|$ and $\langle 10|$, respectively, leads to

$$i\hbar\partial_t c_{01}(t) = \kappa c_{10}(t)e^{+i\hbar\Delta t} \quad (3.17)$$

$$i\hbar\partial_t c_{10}(t) = \kappa c_{01}(t)e^{-i\hbar\Delta t}, \quad (3.18)$$

where $\Delta = \omega_B - \omega_A$. The coefficient κ will be determined by the details of the coupling. In Chapter 6 we show that for qubits consisting of nonlinear LC circuits coupled by a capacitor, $\kappa = -ig$ with $g > 0$ (see (6.142) and (6.143)). These equations are then identical in form to the coupled qubit states (2.103) and (2.104)! It follows that the solutions are also given by (2.114) with $c_0 \rightarrow c_{01}$ and $c_1 \rightarrow c_{10}$. Referring to Figure 2.5 we see that significant state conversion only occurs when $\Delta \approx 0$, or in other words when the qubit frequencies are matched. In this case, the probability amplitudes evolve as (see (2.113) and (2.114)):

$$\begin{aligned} c_{01}(t) &= c_{01}(0) \cos gt - ic_{10}(0) \sin gt, \\ c_{10}(t) &= -ic_{01}(0) \sin gt + c_{10}(0) \cos gt. \end{aligned} \quad (3.19)$$

This can be conveniently written in matrix form:

$$\begin{bmatrix} c_{01}(t) \\ c_{10}(t) \end{bmatrix} = \begin{bmatrix} \cos gt & -i \sin gt \\ -i \sin gt & \cos gt \end{bmatrix} \begin{bmatrix} c_{01}(0) \\ c_{10}(0) \end{bmatrix}. \quad (3.20)$$

Making use of the fact that the states $|\psi_{00}\rangle$ and $|\psi_{11}\rangle$ do not change with time, the evolution of the entire 2-qubit system can be written as the 4×4 matrix [12]:

$$\begin{bmatrix} c_{00}(t) \\ c_{01}(t) \\ c_{10}(t) \\ c_{11}(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos gt & -i \sin gt & 0 \\ 0 & -i \sin gt & \cos gt & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{00}(0) \\ c_{01}(0) \\ c_{10}(0) \\ c_{11}(0) \end{bmatrix}. \quad (3.21)$$

We see that for $gt = \pi/4$ this matrix corresponds to the $\sqrt{i\text{SWAP}}$ gate.⁵ The coupling can be effectively turned off by detuning the qubits. So the $\sqrt{i\text{SWAP}}$ gate is formed by tuning the qubits to the same frequency, leaving them tuned for a time $t = \pi/(4g)$, and then detuning them again.

The analysis showing how the coupling terms we have assumed arise from capacitively-coupled superconducting qubits is presented in Sections 6.4.6 and 6.4.7.

3.3 Cross Resonance Scheme

Turning on and off the interaction by tuning the qubits requires tuning control lines in addition to the RF signal lines. This additional complexity impacts the scalability of the system. In contrast, the *cross-resonance* scheme uses fixed-frequency qubits with fixed coupling, and controls the interaction with RF signals—simplifying qubit control [14, 15]. The key idea is that the frequency of qubit 2 is applied to qubit 1, or vice versa. Owing to the coupling, qubit 2 exhibits Rabi oscillations as expected, but it also acquires

⁵ We also see that the $\sqrt{i\text{SWAP}}$ state is generated after a time $gt = 7\pi/4$.

a phase that depends on the state of qubit 1. This provides the basic mechanism for entanglement. This interaction can be controlled simply by varying the amplitude of the applied drive field.

The approximate Hamiltonian for this system with the drive signal applied to qubit 1 is (see (2.84) and (6.155))

$$\mathcal{H} = -\frac{1}{2}\hbar\omega_1\sigma_1^z + \hbar\Omega_1 \cos(\omega_d t) \sigma_1^x - \frac{1}{2}\hbar\omega_2\sigma_2^z + \frac{1}{2}\hbar\omega_{xx}\sigma_2^x\sigma_1^x. \quad (3.22)$$

Here we have also made the substitution $g = \omega_{xx}/2$ to facilitate comparison with the literature [14].

It is worthwhile to pause for a moment to comment on various notations that are in use. Equation (3.22) uses a shorthand for the Pauli operators. Written explicitly in terms of tensor products this becomes

$$\begin{aligned} \mathcal{H} = & -\frac{1}{2}\hbar\omega_1 I \otimes \sigma_z + \hbar\Omega_1 \cos(\omega_d t) I \otimes \sigma_x \\ & -\frac{1}{2}\hbar\omega_2 \sigma_z \otimes I + \frac{1}{2}\hbar\omega_{xx} \sigma_x \otimes \sigma_x. \end{aligned} \quad (3.23)$$

Yet another way of writing this when the connection back to gates is desired is to replace the Paulis with X, Y, Z (Recall from Chapter 1 that in our notation $X = \sigma_x$, etc.). In addition, the tensor product symbol is often omitted, so the Hamiltonian can also be written

$$\mathcal{H} = -\frac{1}{2}\hbar\omega_1 IZ + \hbar\Omega_1 \cos(\omega_d t) IX - \frac{1}{2}\hbar\omega_2 ZI + \frac{1}{2}\hbar\omega_{xx} XX. \quad (3.24)$$

Setting $\omega_d = \omega_2$, making a series of unitary transformations, and applying the rotating wave approximation [14], the effective Hamiltonian can be written

$$\mathcal{H}^{\text{eff}} = \frac{\hbar\omega_{xx}}{4} \frac{1}{1 + (\Delta/\Omega_1)^2} \left(XX + \frac{\Delta}{\Omega_1} XZ \right) \quad (3.25)$$

where $\Delta = |\omega_1 - \omega_2|$. If the qubits are sufficiently detuned so that $\Omega_1 \ll \Delta$, we then have

$$\mathcal{H}^{\text{eff}} \approx \frac{\Omega_1 \hbar\omega_{xx}}{4\Delta} XZ. \quad (3.26)$$

From Schrödinger's equation, the time evolution of the state vector is given by

$$\begin{aligned} |\psi(t)\rangle &= e^{-i\mathcal{H}t/\hbar} |\psi(0)\rangle \\ &= \exp(-i\Omega_1 \omega_{xx} t XZ / (4\Delta)) |\psi(0)\rangle \\ &\equiv \exp(-i\beta \pi XZ / 2) |\psi(0)\rangle \end{aligned} \quad (3.27)$$

where we have introduced the parameter $\beta = \Omega_1 \omega_{xx} t / (2\pi\Delta)$. Using (2.31) to expand the unitary operator, we have

$$|\psi(t)\rangle = [I \cos(\beta\pi/2) - iXZ \sin(\beta\pi/2)] |\psi(0)\rangle. \quad (3.28)$$

It is clear that if $\beta = 1$, then

$$e^{-i\mathcal{H}t/\hbar} = -iXZ, \quad (3.29)$$

or for a general value of β :

$$e^{-i\mathcal{H}t/\hbar} = (-iXZ)^\beta = e^{-i\pi\beta/2} (XZ)^\beta. \quad (3.30)$$

This is significant, since $(XZ)^\beta$ along with the one-qubit unitaries comprises a universal gate set for quantum computing [14]. In particular, for $\beta = 1/2$, we have $(XZ)^{1/2}$ to within a global phase factor, and this unitary can be used to generate a CNOT [14, 16]:

$$\text{CNOT} = (IZ)^{-1/2}(XZ)^{1/2}(XI)^{-1/2}. \quad (3.31)$$

Alternatively, this can be written in terms of single-qubit rotations as

$$\text{CNOT} = [I \otimes R_z(-\pi/2)][XZ]^{1/2}[R_x(-\pi/2) \otimes I]. \quad (3.32)$$

To generate $(XZ)^{1/2}$, we simply need to apply the excitation to qubit 1 for a time duration of

$$t = \frac{\pi\Delta}{\Omega_1\omega_{xx}}. \quad (3.33)$$

We see that increasing the coupling ω_{xx} or drive amplitude Ω_1 shortens the time interval needed, while increasing the frequency separation of the qubits increases the time interval needed.

3.4 Other Controlled Gates

The CNOT is a fundamental two-qubit gate and is necessary for universal quantum computing. It plays a large role in the logic of quantum algorithms, and it often creates entangled states, as we saw with the Bell states in Section 1.6 and Exercise 1.2.

It is often convenient to add a control signal to some other gate, such as controlled- Z , controlled- H , and so forth. We can do this with any single-qubit gate, as shown in Figure 3.3. The matrix describing a controlled- U or CU gate is shown below; the subscripts indicate the control qubit followed by the target qubit, while U_{jk} indicates the matrix element (j, k) of U .

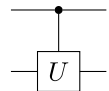
$$\text{CU}_{0,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & U_{00} & 0 & U_{01} \\ 0 & 0 & 1 & 0 \\ 0 & U_{10} & 0 & U_{11} \end{bmatrix}. \quad (3.34)$$

We can implement a controlled- U gate using CNOT and single-qubit gates, using a decomposition that is described in 2.4. First, recall the YZZ decomposition of a unitary U :

$$U = e^{i\alpha}R_z(\phi)R_y(\theta)R_z(\lambda). \quad (3.35)$$

In this version, we include a global phase α . When the gate U is only applied for basis states in which the control qubit is $|1\rangle$, then the global phase becomes a relative phase, because it is not applied to all basis states.

Figure 3.3 Controlled- U gate. If the control qubit is $|1\rangle$, the U gate is applied to the target qubit. Otherwise, the gate is not applied.



We then define unitaries A , B , and C :

$$\begin{aligned} A &\equiv R_z(\phi)R_y(\theta/2) \\ B &\equiv R_y(-\theta/2)R_z(-(\lambda + \phi)/2) \\ C &\equiv R_z((\lambda + \phi)/2) \end{aligned} \quad (3.36)$$

which leads to the following relationships:

$$ABC = I \quad (3.37)$$

$$AXBXC = R_z(\phi)R_y(\theta)R_z(\lambda) \quad (3.38)$$

$$U = e^{i\alpha}AXBXC. \quad (3.39)$$

First, consider how to add a global phase α only if the control bit is $|1\rangle$. The unitary for the global phase is

$$U_\alpha = e^{i\alpha}I = \begin{bmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{bmatrix}, \quad (3.40)$$

and the matrix for the controlled version is

$$\begin{aligned} CU_\alpha &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\alpha} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix} \\ &= I \otimes u1(\alpha). \end{aligned} \quad (3.41)$$

In other words, we can perform a controlled global phase by simply applying a $u1$ gate (Section 2.2.1.4) to the control qubit. This seems odd, but it does what we want: if the control qubit is $|0\rangle$, nothing happens; when the control qubit is $|1\rangle$ the phase is applied, keeping in mind that the phase affects the entire two-qubit state, not just the control qubit.

Then we use Eq. (3.39) and implement the controlled- U using the circuit shown in Figure 3.4. If the control qubit is $|0\rangle$, then the CNOT gates are not applied and $ABC = I$, so nothing happens. If the control qubit is $|1\rangle$, then U is applied to the target qubit.

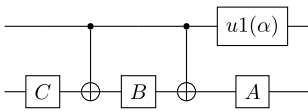


Figure 3.4 Implementation of controlled- U gate using Eq. (3.39).

3.5 Two-Qubit States and the Density Matrix

We are now ready to resume our discussion of the density matrix as a representation of a quantum state. The definition of the density matrix remains the same as before:

$$\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j|. \quad (3.42)$$

If ψ_j is a two-qubit state, then it is a four-element state vector and ρ is a 4×4 -element matrix. We do have to review our definition of pure vs. mixed state, because our previous definition involves the surface of the Bloch sphere. There is no Bloch sphere for a two-qubit state, so a pure state is one that can be described as a linear superposition of two-qubit state vectors. A mixed state is, again, an ensemble of pure states, each with a probability. In other words, a pure state will only have one term in Eq. (3.42).

If we know the density matrices of each of the two qubits, they can be combined using the tensor product to produce a combined density matrix. In the following equations, we use ρ^A to denote the density matrix of qubit A , and we use ρ^{AB} to denote the density matrix of the two-qubit state involving qubits A and B .

$$\rho^{AB} = \rho^A \otimes \rho^B \quad (3.43)$$

As an example, suppose qubit A is $|1\rangle$ and qubit B is $|+\rangle$. We can find the density matrix ρ^{AB} in two ways:

$$\begin{aligned} \rho^{AB} &= |1+\rangle\langle 1+| = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \end{aligned} \quad (3.44)$$

$$\begin{aligned} \rho^{AB} &= \rho_0 \otimes \rho_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \end{aligned} \quad (3.45)$$

Note that $\text{Tr}(\rho^{AB}) = 1$, and $\text{Tr}((\rho^{AB})^2) = 1$, which means it's a pure state.

Now let's consider one of the Bell states: $|\Psi^+\rangle = 1/\sqrt{2}(|01\rangle + |10\rangle)$. The density matrix is:

$$\begin{aligned}\rho^{AB} &= \left(\frac{|01\rangle + |10\rangle}{\sqrt{2}} \right) \left(\frac{\langle 01| + \langle 10|}{\sqrt{2}} \right) \\ &= \frac{|01\rangle\langle 01| + |01\rangle\langle 10| + |10\rangle\langle 01| + |10\rangle\langle 10|}{2} \\ &= \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.\end{aligned}\tag{3.46}$$

This is also a pure state, because $\text{Tr}((\rho^{AB})^2) = 1$. But it is not a product state—it cannot be decomposed into a tensor product of two state vectors or a tensor product of two density matrices.

However, the advantage of the density matrix representation is that we can compute the density matrix of each qubit, even when the total state is not a product state. We do this using an operation called a *partial trace* [17, 18]. To recover the state of qubit A , we “trace out” qubit B from the two-qubit density matrix:

$$\rho^A = \text{Tr}_B(\rho^{AB}).\tag{3.47}$$

For our purposes, we will assume that the density matrix ρ^{AB} is built using the standard basis states ($|0\rangle$ and $|1\rangle$) for both qubits A and B . Then the density matrix can be decomposed as:

$$\rho^{AB} = \sum_{i,j,k,l} c_{ik,jl} |i\rangle\langle j| \otimes |k\rangle\langle l|,\tag{3.48}$$

where $i, j, k, l \in \{0, 1\}$, i and j represent states of A , and k and l are states of B . In this notation, ik means a concatenation of the one-bit numbers i and k . When written as a matrix, constant $c_{ik,jl}$ will appear in row ik and column jl .

$$\rho^{AB} = \begin{bmatrix} c_{00,00} & c_{00,01} & c_{00,10} & c_{00,11} \\ c_{01,00} & c_{01,01} & c_{01,10} & c_{01,11} \\ c_{10,00} & c_{10,01} & c_{10,10} & c_{10,11} \\ c_{11,00} & c_{11,01} & c_{11,10} & c_{11,11} \end{bmatrix}.\tag{3.49}$$

To perform the partial trace $\text{Tr}_B(\rho^{AB})$, we trace only the part of the matrix that is relevant to qubit B . This leaves behind the density matrix associated with only qubit A .

$$\rho^A = \text{Tr}_B(\rho^{AB}) = \sum_{i,j,k,l} c_{ik,jl} |i\rangle\langle j| \text{Tr}(|k\rangle\langle l|)\tag{3.50}$$

$$= \sum_{i,j,k,l} c_{ik,jl} |i\rangle\langle j| \langle l|k\rangle\tag{3.51}$$

$$= \sum_{i,j,k} c_{ik,kl} |i\rangle\langle j|.\tag{3.52}$$

Equation (3.51) exploits the fact that $\text{Tr} |\alpha\rangle\langle\beta| = \langle\beta|\alpha\rangle$ for two states. Since the states $|k\rangle$ and $|l\rangle$ in this equation are basis states, they are orthogonal, meaning that $\langle l|k\rangle = 1$ if $k = l$, and zero otherwise, leading to (3.52).

Therefore, element $a_{i,j}$ of ρ^A is given by:

$$a_{i,j} = \sum_k c_{ik,jk}, \quad (3.53)$$

leading to the following definition of ρ^A :

$$\rho^A = \text{Tr}_B(\rho^{AB}) = \begin{bmatrix} c_{00,00} + c_{01,01} & c_{10,00} + c_{11,01} \\ c_{10,00} + c_{11,01} & c_{11,00} + c_{11,11} \end{bmatrix}. \quad (3.54)$$

Performing a partial trace of A to recover ρ^B follows an equivalent path:

$$\rho^B = \text{Tr}_A(\rho^{AB}) = \sum_{i,j,k,l} c_{ik,jl} \text{Tr}(|i\rangle\langle j|) |k\rangle\langle l| \quad (3.55)$$

$$= \sum_{i,j,k,l} c_{ik,jl} \langle j|i\rangle |k\rangle\langle l| \quad (3.56)$$

$$= \sum_{i,k,l} c_{ik,il} |k\rangle\langle l| \quad (3.57)$$

$$= \begin{bmatrix} c_{00,00} + c_{10,10} & c_{01,00} + c_{11,10} \\ c_{00,01} + c_{10,11} & c_{01,01} + c_{11,11} \end{bmatrix}. \quad (3.58)$$

Returning to the Bell state example (3.46),

$$\rho^A = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \rho^B = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (3.59)$$

You may recognize this as the density matrix of a maximally mixed state (Section 2.5). This is very interesting—the two-qubit state is pure, but the state of each qubit on its own is mixed. Even though we know everything about the two-qubit system, we can say nothing about the individual qubits within that system.

The density matrix and partial trace approach can easily be extended to more than two qubits. Instead of qubit A and qubit B , we can have n -qubit system A and m -qubit system B , and we can extract the density matrix of either system from the joint density matrix ρ^{AB} . The only change is that i and j become n -bit values, while k and l become m -bit values. We can also use any basis sets $\{a_i\}$ and $\{b_i\}$ for A and B , even different ones for each system. The partial trace approach is very powerful, and is particularly useful when describing the interaction of a quantum mechanical system with the surrounding environment.

3.6 Exercises

- 3.1** Verify by direct computation that to within a global phase factor, the following relation is true:

$$\begin{aligned} \text{CNOT} = & [I \otimes R_y(-\pi/2)][R_x(-\pi/2) \otimes R_x(\pi/2)]\sqrt{i\text{SWAP}}' \\ & \cdot [I \otimes R_x(\pi)]\sqrt{i\text{SWAP}}'[I \otimes R_y(\pi/2)]. \end{aligned}$$

- 3.2** Given that

$$I \cos(\beta\pi/2) - iXZ \sin(\beta\pi/2) = e^{-i\pi\beta/2}(XZ)^\beta,$$

find $(XZ)^{1/2}$.

- 3.3** Verify by direct computation that to within a global phase factor, the following relation is true:

$$\text{CNOT} = (IZ)^{-1/2}(XZ)^{1/2}(XI)^{-1/2}.$$

- 3.4** Using the approach in Figure 3.4, create a quantum circuit to implement a controlled- H gate. (Hint: Eq. (2.48) will be helpful.)

- 3.5** Consider the following 3-qubit state:

$$|Q\rangle = |q_2q_1q_0\rangle = \frac{|000\rangle + |001\rangle + |010\rangle + |111\rangle}{2}.$$

What is the density matrix for qubit q_2 ?

4

Superconducting Quantum Computer Systems

With the high-level description in Section 1.10 as motivation, in this chapter we consider the components of the superconducting quantum computer system and their operation in more detail. At RF and microwave frequencies, care must be taken about how components are connected. In particular, when the wavelength of the signal is comparable to circuit/system dimensions, familiar low-frequency circuit concepts such as Kirchoff's voltage law no longer hold. Instead, we must look at the connections between all of the components as *transmission lines* and ensure that we are treating the signals conveyed by these transmission lines correctly. Consequently we begin our in-depth consideration of the system components with transmission lines.

4.1 Transmission Lines

4.1.1 General Transmission Line Equations

One type of transmission line that can be used at radio frequencies is the “ladder line” shown in Figure 4.1. In this type of line, the conductor distance is maintained by insulating spacers placed at regular intervals. As suggested in the diagram, currents flow in the conductors when signals are traveling along the line. These currents create magnetic fields encircling the wires resulting in a small amount of *inductance* per unit length. Since real, normal metals also have some small electrical resistance, the currents also see a small *resistance* per unit length. Similarly, the two conductors may have different potentials at a given location along the line, so that electric fields originate on one wire and terminate on the other. The charge separation associated with this potential difference results in a small *capacitance* per unit length, and if the insulators are not perfect, small leakage currents result in an equivalent *conductance* per unit length between the wires.

These observations suggest the equivalent circuit model shown in Figure 4.2. As suggested by part (a) of the figure, lumped elements can be used to represent the series resistance, series inductance, shunt leakage conductance, and shunt capacitance, provided the distance Δz is small compared to the distance that the signal propagates during one period of oscillation.

To analyze how the voltage and current change along the length of the line, consider one section of the line as shown in Figure 4.2(b). Applying Kirchoff's voltage law around

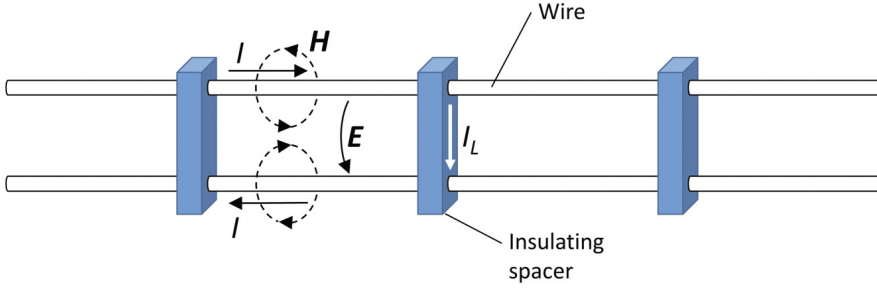
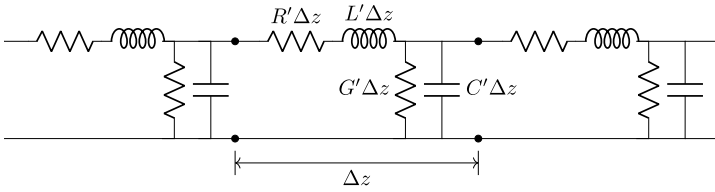
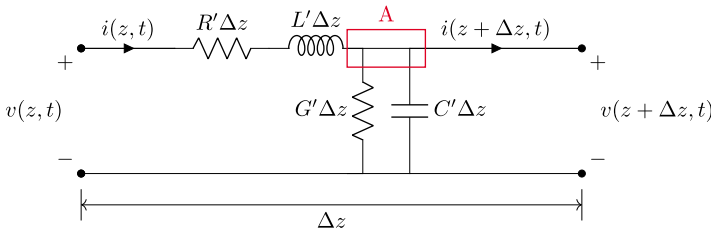


Figure 4.1 Ladder line used for radio frequency transmission.



(a) Equivalent circuit for transmission line.



(b) Single section for analysis.

Figure 4.2 Equivalent circuit for a transmission line. (a) Lumped elements can be used so long as the distance Δz is small compared to the distance traveled during a period of the signal. (b) Single section along the line for analysis.

the outer loop gives

$$v(z, t) - i(z, t)R'\Delta z - L'\Delta z \frac{\partial i(z, t)}{\partial t} - v(z + \Delta z, t) = 0. \quad (4.1)$$

This can be rearranged to obtain

$$\frac{v(z + \Delta z) - v(z, t)}{\Delta z} = -i(z, t)R' - L' \frac{\partial i(z, t)}{\partial t}. \quad (4.2)$$

If we now take the limit as $\Delta z \rightarrow 0$, we have a differential equation relating the voltage $v(z, t)$ and the current $i(z, t)$:

$$\frac{\partial v}{\partial z} = -iR' - L' \frac{\partial i}{\partial t}. \quad (4.3)$$

Next, consider Kirchoff's current law applied to node A in Figure 4.2(b):

$$i(z, t) = i(z + \Delta z, t) + v(z + \Delta z, t)G'\Delta z + C'\Delta z \frac{\partial v(z + \Delta z, t)}{\partial t}. \quad (4.4)$$

Rearranging this and taking the limit as before gives a second differential equation relating the current and voltage:

$$\begin{aligned} \frac{i(z + \Delta z, t) - i(z, t)}{\Delta z} &= -G'v(z + \Delta z, t) - C' \frac{\partial v(z + \Delta z, t)}{\partial t}. \\ \frac{\partial i}{\partial z} &= -G'v - C' \frac{\partial v}{\partial t}. \end{aligned} \quad (4.5)$$

Obtaining the general solution to Eqs. (4.3) and (4.5) is, unfortunately, rather complicated. However, solutions can be obtained easily for two important cases: lossless lines, and the sinusoidal steady state. These will be considered in the next sections.

4.1.2 Lossless Transmission Lines

Since the loss is represented by the series resistance R' and the parallel conductance G' , we need only to set these quantities to zero to obtain the equations for a lossless transmission line. Equations (4.3) and (4.5) become

$$\frac{\partial v}{\partial z} = -L' \frac{\partial i}{\partial t} \quad (4.6)$$

and

$$\frac{\partial i}{\partial z} = -C' \frac{\partial v}{\partial t}. \quad (4.7)$$

To solve these equations, we need to combine them to either eliminate i or v . As an example, let us eliminate v . The first step is to differentiate both sides of Eq. (4.6) with respect to t , and both sides of Eq. (4.7) with respect to z :

$$\frac{\partial^2 v}{\partial t \partial z} = -L' \frac{\partial^2 i}{\partial t^2}, \quad (4.8)$$

$$\begin{aligned} \frac{\partial^2 i}{\partial z^2} &= -C' \frac{\partial^2 v}{\partial z \partial t} \\ &= -C' \frac{\partial^2 v}{\partial t \partial z}. \end{aligned} \quad (4.9)$$

In writing the last line of Eq. (4.9), we have made use of the fact that z and t are independent variables, so the order of differentiation does not matter. The voltage v can now be eliminated from Eqs. (4.8) and (4.9) to obtain

$$\frac{\partial^2 i}{\partial t^2} = \frac{1}{L'C'} \frac{\partial^2 i}{\partial z^2}. \quad (4.10)$$

If we do a similar manipulation to eliminate i instead of v , we will find that the equation for v is identical to Eq. (4.10). This is called the *wave equation*, for reasons that will become clear as we discuss its properties.

It is straightforward to show that any function of the form $f(z \pm ut)$ is a solution to Eq. (4.10). To see this, let $s = z \pm ut$ and substitute $f(s)$ into Eq. (4.10). Using $\partial s / \partial z = 1$ and $\partial s / \partial t = \pm u$, we obtain

$$u^2 \frac{\partial^2 f}{\partial s^2} = \frac{1}{L'C'} \frac{\partial^2 f}{\partial s^2}. \quad (4.11)$$

We conclude that $f(z \pm ut)$ is a solution to the wave equation provided

$$u = \frac{1}{\sqrt{L'C'}}. \quad (4.12)$$

To interpret this, we note that $f(s) = f(z \pm ut)$ represents an arbitrary waveform moving along the z axis. To follow a particular point on the curve to see how fast it is moving, we hold the argument s constant. Taking the time derivative of s and setting it to zero gives

$$\frac{ds}{dt} = \frac{dz}{dt} \pm u = 0, \quad (4.13)$$

or

$$\frac{dz}{dt} = \mp u. \quad (4.14)$$

Consequently, the rate at which z must change to follow a constant point on the curve is given by u . We call u the *phase velocity* of the waveform. The choice of sign simply means that the waveform can travel in either direction along the z axis.

Another important property of the signal that travels along the transmission line can be obtained by using what we have now learned with one of the original lossless transmission line equations, say Eq. (4.6). If both i and v have solutions of the form $f(z \pm ut)$ so that $i = i_0 f(z \pm ut)$ and $v = v_0 f(z \pm ut)$ then Eq. (4.6) can be written

$$v_0 \frac{\partial f}{\partial s} = \mp L' u i_0 \frac{\partial f}{\partial s}. \quad (4.15)$$

Solving for the ratio v_0/i_0 gives

$$\begin{aligned} \frac{v_0}{i_0} &= \mp L' u \\ &= \mp L' \frac{1}{\sqrt{L'C'}} \\ &= \mp \sqrt{\frac{L'}{C'}}. \end{aligned} \quad (4.16)$$

Since the ratio of voltage to current has dimensions of impedance, we define the *characteristic impedance* Z_c of the transmission line as:

$$Z_c = \sqrt{\frac{L'}{C'}}. \quad (4.17)$$

We see, then, that if an arbitrary pulse shape is transmitted along a lossless transmission line, the pulse will propagate without distortion at the speed $1/\sqrt{L'C'}$, and at each point, the ratio of the voltage to the current is given by the characteristic impedance (4.17). When loss is present, both of these observations will need to be modified.

4.1.3 Transmission Lines with Loss

4.1.3.1 Sinusoidal Steady State

As indicated earlier, the presence of loss makes the general solution of the transmission line equations much more difficult, but we can solve them fairly easily for the sinusoidal steady state. In this case we can write¹

$$\begin{aligned} i(z, t) &= \text{Re}\{I(z)e^{j\omega t}\} \\ v(z, t) &= \text{Re}\{V(z)e^{j\omega t}\} \end{aligned} \quad (4.18)$$

where $I(z)$ and $V(z)$ are complex amplitudes that represent the magnitude and phase of the steady state signal as a function of position along the line. In terms of these complex amplitudes, the transmission line Eqs. (4.3) and (4.5) become

$$\frac{\partial V}{\partial z} = -(R' + j\omega L')I, \quad (4.19)$$

$$\frac{\partial I}{\partial z} = -(G' + j\omega C')V. \quad (4.20)$$

As in the lossless case, we proceed to eliminate either I or V to obtain a single differential equation. For example, solving Eq. (4.19) for I and substituting the result into Eq. (4.20) gives

$$\begin{aligned} \frac{\partial}{\partial z} \left(-\frac{1}{R' + j\omega L'} \frac{\partial V}{\partial z} \right) &= -(G' + j\omega C')V \\ \frac{\partial^2 V}{\partial z^2} &= (R' + j\omega L')(G' + j\omega C')V \\ \frac{\partial^2 V}{\partial z^2} &= \gamma^2 V, \end{aligned} \quad (4.21)$$

where we define

$$\gamma^2 = (R' + j\omega L')(G' + j\omega C'). \quad (4.22)$$

Following a similar procedure to eliminate V results in an equation of the same form for I :

$$\frac{\partial^2 I}{\partial z^2} = \gamma^2 I. \quad (4.23)$$

You can easily verify that solutions to Eqs. (4.21) and (4.23) are of the form

$$V(z) = V_0^+ e^{-\gamma z} + V_0^- e^{\gamma z} \quad (4.24)$$

and

$$I(z) = I_0^+ e^{-\gamma z} + I_0^- e^{\gamma z}, \quad (4.25)$$

where we define γ as the positive square root of Eq. (4.22):

$$\gamma = \sqrt{(R' + j\omega L')(G' + j\omega C')} \equiv \alpha + j\beta. \quad (4.26)$$

¹ Here we use the electrical engineering convention for time-dependence $\exp(j\omega t)$, while in quantum mechanics and quantum computing it is more common to use the convention $\exp(-i\omega t)$. So to avoid confusion, in this book we use the convention $j = -i$.

The parameter γ is sometimes referred to as the *propagation constant*, since it describes how the waves change as they move along the z direction. (Note however that while γ is a constant with respect to position and time, it does vary with *frequency*.)

The coefficients of the terms on the right-hand side of Eqs. (4.24) and (4.25) can be related with the aid of Eq. (4.19):

$$\begin{aligned} I(z) &= -\frac{1}{R' + j\omega L'} \frac{\partial V}{\partial z} \\ &= -\frac{\gamma}{R' + j\omega L'} (-V_0^+ e^{-\gamma z} + V_0^- e^{+\gamma z}) \\ &= \frac{\gamma}{R' + j\omega L'} (V_0^+ e^{-\gamma z} - V_0^- e^{+\gamma z}). \end{aligned} \quad (4.27)$$

Comparing Eqs. (4.27) and (4.25), we see that

$$\frac{V_0^+}{I_0^+} = -\frac{V_0^-}{I_0^-} = \frac{R' + j\omega L'}{\gamma}. \quad (4.28)$$

This suggests that we define the characteristic impedance for the lossy line as

$$Z_c \equiv \frac{R' + j\omega L'}{\gamma} = \frac{R' + j\omega L'}{\sqrt{(R' + j\omega L')(G' + j\omega C')}} = \sqrt{\frac{R' + j\omega L'}{G' + j\omega C'}}. \quad (4.29)$$

Note that this reduces to the lossless expression (4.17) when $R' = G' = 0$.

Returning to the time domain, we have

$$\begin{aligned} v(z, t) &= \text{Re} \{ V_0^+ e^{-\gamma z + j\omega t} + V_0^- e^{\gamma z + j\omega t} \} \\ &= \text{Re} \{ |V_0^+| e^{-\alpha z - j\beta z + j\omega t + j\phi^+} + |V_0^-| e^{\alpha z + j\beta z + j\omega t + j\phi^-} \} \\ &= |V_0^+| e^{-\alpha z} \underbrace{\cos(\omega t - \beta z + \phi^+)}_{\text{wave traveling in } +z \text{ direction}} \\ &\quad + |V_0^-| e^{\alpha z} \underbrace{\cos(\omega t + \beta z + \phi^-)}_{\text{wave traveling in the } -z \text{ direction}}. \end{aligned} \quad (4.30)$$

To see why the first and second terms describe waves traveling in the positive and negative z directions, respectively, consider what the value of z must do for the arguments of the cosines to remain constant. For the first term, the value of z must increase as t increases, so the constant point on the wave moves in the positive z direction. Similarly, z must decrease as t increases in the second term for the argument of the cosine to remain constant. To find the speed with which the constant point moves, we set the arguments equal to a constant, solve for z , then differentiate with respect to time. For example, from the argument of the cosine in the first term we have

$$\begin{aligned} \omega t - \beta z + \phi^+ &= \text{const} \\ z &= \frac{\omega}{\beta} t + \frac{\phi^+ - \text{const}}{\beta} \end{aligned}$$

and

$$u = \frac{dz}{dt} = \frac{\omega}{\beta}. \quad (4.31)$$

Signals that have the form of sinusoidal variations moving along a particular direction are referred to as *waves*. In the presence of loss, the sinusoids also decrease in amplitude as they move, or *propagate*, according to the factors $\exp(\pm\alpha z)$ in Eq. (4.30). Solutions of this type are referred to as *damped waves*, and the decrease in amplitude caused by the loss is referred to as *attenuation*.

The attenuation of a transmission line is often described in terms of *decibels per unit length*, or dB/m. The attenuation of a cable in dB is obtained from the ratio of the voltage (or current) magnitude after propagating a distance L , to the voltage (or current) magnitude at the input to the line:

$$\begin{aligned} dB &= -20\log_{10}\left(\frac{v(L)}{v(0)}\right) = -20\log_{10}\left(\frac{e^{-\alpha L}}{e^0}\right) \\ &= 20\alpha L \log_{10}e \\ &= 8.69\alpha L. \end{aligned} \quad (4.32)$$

The attenuation per unit length is therefore

$$dB/m = 8.69\alpha. \quad (4.33)$$

An important parameter of a wave is the distance required to obtain a phase shift of 2π , called the *wavelength*. To find this distance, we take the difference between the values of the argument of one of the cosines in Eq. (4.30) at two difference positions, set this difference equal to 2π , and solve for the necessary separation. For example, for the wave traveling in the $+z$ direction:

$$\begin{aligned} [\omega t - \beta(z - \lambda) + \phi^+] - [\omega t - \beta z + \phi^+] &= 2\pi \\ \beta\lambda &= 2\pi \\ \lambda &= \frac{2\pi}{\beta}. \end{aligned} \quad (4.34)$$

Since $\beta = 2\pi/\lambda$ can be interpreted as the number of wavelengths in a distance of 2π meters, β is referred to as the *wave number*.² These general results for velocity and wavelength can be easily reduced to the lossless case. From Eq. (4.22) with $R' = G' = 0$,

$$\gamma = \sqrt{j\omega L' j\omega C'} = \sqrt{-\omega^2 L' C'} = j\omega\sqrt{L' C'} = j\beta, \quad (4.35)$$

and

$$\beta = \omega\sqrt{L' C'}. \quad (4.36)$$

The phase velocity is therefore

$$u = \frac{\omega}{\beta} = \frac{1}{\sqrt{L' C'}} \quad (4.37)$$

² In some contexts, the quantity $1/\lambda$ is referred to as the “wave number.” In this case it represents the number of wavelengths per meter, as opposed to the number per 2π meters. This alternate definition is sometimes used in optics.

as before. The wavelength in the lossless case is

$$\begin{aligned}\lambda &= \frac{2\pi}{\beta} = \frac{2\pi}{\omega\sqrt{L'C'}} \\ \lambda &= \frac{1}{f\sqrt{L'C'}}.\end{aligned}\tag{4.38}$$

Equations (4.36) and (4.38) relating the angular frequency to the wave number or the frequency to the wavelength are referred to as *dispersion relations*.

4.1.3.2 Low Loss Transmission Lines

In many practical situations, the loss (or attenuation) of a transmission line is small but not negligible. For this reason it is useful to examine the expressions for the transmission line parameters and find approximations that are valid in the small-loss limit. In making these approximations, we will keep only the *lowest-order non-vanishing terms* for the parameters Z_c , α , β , and λ .

Let us first consider the characteristic impedance given by Eq. (4.29). The conditions for low loss are

$$\begin{aligned}R' &\ll \omega L' \\ G' &\ll \omega C' .\end{aligned}\tag{4.39}$$

With this limit in mind, we can write the characteristic impedance as

$$Z_c = \sqrt{\frac{j\omega L' \left(1 + \frac{R'}{j\omega L'}\right)}{j\omega C' \left(1 + \frac{G'}{j\omega C'}\right)}}.\tag{4.40}$$

Now if the conditions given by (4.39) are satisfied, then the terms inside the parentheses in both the numerator and denominator are approximately equal to unity, and the characteristic impedance is well-approximated to lowest-order by that of a lossless line:

$$Z_c \approx \sqrt{\frac{L'}{C'}}.\tag{4.41}$$

Next consider the propagation constant γ , from which we can obtain α and β . From Eq. (4.22) and anticipating the limit (4.39), the propagation constant can be written

$$\begin{aligned}\gamma &= \sqrt{-\omega^2 L' C' \left[1 - \frac{R' G'}{\omega^2 L' C'} - j\omega \left(\frac{L' G' + R' C'}{\omega^2 L' C'}\right)\right]} \\ &= j\omega\sqrt{L' C'} \sqrt{1 - \frac{R' G'}{\omega^2 L' C'} - j\left(\frac{G'}{\omega C'} + \frac{R'}{\omega L'}\right)}.\end{aligned}\tag{4.42}$$

Now consider the terms in the square root carefully. In view of the assumptions (4.39), the third term is linear in small quantities while the second term is quadratic in small quantities. We conclude that to lowest order, the second term can be neglected.

Further, we note that

$$\sqrt{1+x} \approx 1 + x/2 \quad (4.43)$$

if $|x| \ll 1$. Making these two approximations, Eq. (4.42) can be written

$$\begin{aligned} \gamma &\approx j\omega\sqrt{L'C'} \left[1 - \frac{j}{2} \left(\frac{G'}{\omega C'} + \frac{R'}{\omega L'} \right) \right] \\ &= j\omega\sqrt{L'C'} + \frac{1}{2} \left(G' \sqrt{\frac{L'}{C'}} + R' \sqrt{\frac{C'}{L'}} \right) \\ &\approx j\omega\sqrt{L'C'} + \frac{1}{2} (G'Z_c + R'/Z_c). \end{aligned} \quad (4.44)$$

Here we have also used the approximation (4.41) for Z_c in the last step. Equating the real part of this expression with α and the imaginary part with β gives

$$\alpha \approx \frac{1}{2} (G'Z_c + R'/Z_c), \quad (4.45)$$

$$\beta \approx \omega\sqrt{L'C'}. \quad (4.46)$$

It follows that u and λ are also approximated by their lossless values to lowest order:

$$u = \frac{\omega}{\beta} \approx \frac{1}{\sqrt{L'C'}}, \quad (4.47)$$

$$\lambda = \frac{2\pi}{\beta} \approx \frac{1}{f\sqrt{L'C'}}. \quad (4.48)$$

We conclude that in the low loss limit we can approximate the attenuation constant α with (4.45), and the remainder of the parameters with their lossless expressions.

4.2 Terminated Lossless Line

Since the whole point of a transmission line is to carry signals from one place to another, it is critical that we understand what happens when we connect something (like a qubit!) to the end of the transmission line. The thing connected to the end of the transmission line is called the load. As we will see, the amount of power delivered to the load depends critically on how the load impedance compares with the characteristic impedance of the transmission line. For simplicity, we will assume the line is lossless.

4.2.1 Reflection Coefficient

To begin the discussion, consider the geometry shown in Figure 4.3. A transmission line with characteristic impedance Z_c and wave number β is terminated with the impedance Z_L . In the most general case, we imagine that we can have waves traveling both to the

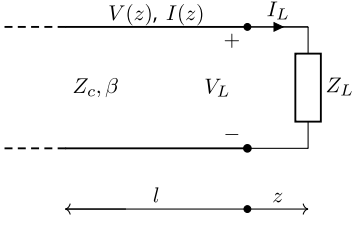


Figure 4.3 A transmission line terminated with a load impedance. Note that two coordinate systems have their origins at the load location: the coordinate z increases to the right, and the coordinate l increases to the left.

right (incident on the load) and to the left (reflected from the load). In this case, general expressions for the voltage and current along the line are

$$V(z) = V_0^+ e^{-j\beta z} + V_0^- e^{j\beta z}, \quad (4.49)$$

$$\begin{aligned} I(z) &= I_0^+ e^{-j\beta z} + I_0^- e^{j\beta z} \\ &= \frac{V_0^+}{Z_c} e^{-j\beta z} - \frac{V_0^-}{Z_c} e^{j\beta z}, \end{aligned} \quad (4.50)$$

where we have used (4.27) and (4.28) to relate the currents and voltages of the incident and reflected waves in terms of the characteristic impedance. In particular, note that the sign of the reflected wave term in the expression for current (4.50) has changed.

At the load, the total voltage and current must be related by Ohm's law:

$$\begin{aligned} Z_L &= \frac{V_L}{I_L} = \frac{V(0)}{I(0)} = \frac{V_0^+ + V_0^-}{\frac{V_0^+}{Z_c} - \frac{V_0^-}{Z_c}} \\ &= Z_c \frac{V_0^+ + V_0^-}{V_0^+ - V_0^-} = Z_c \frac{1 + V_0^-/V_0^+}{1 - V_0^-/V_0^+} \\ &= Z_c \frac{1 + \Gamma}{1 - \Gamma}. \end{aligned} \quad (4.51)$$

Here we have defined the voltage reflection coefficient

$$\Gamma \equiv \frac{V_0^-}{V_0^+}. \quad (4.52)$$

Equation (4.51) can be solved for the reflection coefficient in terms of the impedances:

$$\Gamma = \frac{Z_L - Z_c}{Z_L + Z_c}. \quad (4.53)$$

It is important to note that the only way to prevent some portion of the incident wave from being reflected is for the load impedance to exactly equal the characteristic impedance of the line. This is referred to as a matched load. In practice, the load is almost never perfectly matched, so some portion of the incident signal will be reflected from the load.

4.2.2 Power (Flow of Energy) and Return Loss

Recall that one of the convenient things about phasor (complex) notation is that the time average of the product of two quantities is easily obtained by multiplying one

quantity by the complex conjugate of the other, then taking half the real part of the result. With this tool in hand, let us consider the time averaged power (flow of energy) in the incident and reflected waves on the transmission line. The time averaged power in the incident wave is

$$\begin{aligned}\langle P_{inc} \rangle &= \frac{1}{2} \text{Re} \left\{ V_0^+ (I_0^+)^* \right\} = \frac{1}{2} \text{Re} \left\{ V_0^+ (V_0^+ / Z_c)^* \right\} \\ &= \frac{|V_0^+|^2}{2Z_c}.\end{aligned}\quad (4.54)$$

This is the power incident on the load. However, not all of the incident power is absorbed by the load in general. The power of the wave reflected from the load is

$$\begin{aligned}\langle P_{ref} \rangle &= \frac{1}{2} \text{Re} \left\{ V_0^- (I_0^-)^* \right\} = \frac{1}{2} \text{Re} \left\{ V_0^- (-V_0^- / Z_c)^* \right\} \\ &= -\frac{|V_0^-|^2}{2Z_c}.\end{aligned}\quad (4.55)$$

The minus sign on the reflected power indicates that the energy is actually flowing to the left, away from the load. The ratio of the reflected to incident power in magnitude is given conveniently in terms of the reflection coefficient:

$$\left| \frac{\langle P_{ref} \rangle}{\langle P_{inc} \rangle} \right| = \left| \frac{V_0^-}{V_0^+} \right|^2 = |\Gamma|^2. \quad (4.56)$$

The fraction of power that is reflected is referred to as the return loss and is usually expressed in dB:

$$\begin{aligned}RL &= -10 \log_{10} \left| \frac{\langle P_{ref} \rangle}{\langle P_{inc} \rangle} \right| = -10 \log_{10} |\Gamma|^2 \\ &= -20 \log_{10} |\Gamma|.\end{aligned}\quad (4.57)$$

Note that since $|\Gamma| \leq 1$, the minus sign ensures that the return loss is always positive. There is a common temptation to want to make the “loss” negative (i.e., to say, “the return loss is -10 dB”), but this would result in a double negative; a negative loss would be the same as gain!

In practice, it is commonly assumed that the match is acceptable if at least 90% of the incident power is absorbed by the load. This corresponds to a return loss of 10 dB.

4.2.3 Standing Wave Ratio (SWR)

Another common way of quantifying the degree of mismatch between a load and a line is the standing wave ratio. Since the incident and reflected waves are traveling in opposite directions, they go in and out of phase with each other as you move along the line. Consequently, at some points they add constructively to give a total voltage higher than that of the incident wave, and at other points they add destructively to give a total voltage lower than that of the incident wave. The ratio of the largest value of the voltage magnitude to that of the smallest value is called the voltage standing wave ratio, or VSWR. (You sometimes hear people try to pronounce this acronym, resulting in something that sounds like, “viz-wahr.”) It turns out that the magnitude of the current

fluctuates in a similar manner, and if you take the ratio of the maximum to minimum values you will get the identical numerical result for a lossless line. Consequently it is not really necessary to add the qualifier, “voltage,” and it is also common to simply call this ratio the standing wave ratio, or SWR (not usually pronounced!). The SWR can be obtained from the reflection coefficient as follows:

$$\begin{aligned}
 VSWR = SWR &= \frac{|V|_{\max}}{|V|_{\min}} = \frac{|V_0^+| + |V_0^-|}{|V_0^+| - |V_0^-|} \\
 &= \frac{1 + |V_0^-|/|V_0^+|}{1 - |V_0^-|/|V_0^+|} \\
 SWR &= \frac{1 + |\Gamma|}{1 - |\Gamma|}.
 \end{aligned} \tag{4.58}$$

If the line is perfectly matched, $|\Gamma| = 0$, and the SWR is 1 (or 1:1). On the other hand, if all of the power is reflected, $|\Gamma| = 1$ and the $SWR \rightarrow \infty$. As mentioned previously, a match is usually considered acceptable if 90% of the power is delivered to the load, and this occurs when the Return Loss is 10 dB. This situation corresponds to an SWR of about 2 (or 2:1).

It is also convenient to express the magnitude of the reflection coefficient in terms of the SWR:

$$|\Gamma| = \frac{SWR - 1}{SWR + 1}. \tag{4.59}$$

It is instructive to consider the shape of the voltage waveform along the line in a bit more detail. Referring to the coordinate systems shown in Figure 4.3, the voltage at a distance l to the left of the load is obtained by setting $z = -l$ in Eq. (4.49). Expressed in terms of the reflection coefficient, we obtain

$$\begin{aligned}
 V(-l) &= V_0^+ [e^{j\beta l} + \Gamma e^{-j\beta l}] \\
 &= V_0^+ e^{j\beta l} [1 + \Gamma e^{-j2\beta l}].
 \end{aligned} \tag{4.60}$$

Next, let us explicitly show the magnitude and phase of the reflection coefficient, $\Gamma = |\Gamma| e^{j\theta}$. Substituting this into Eq. (4.60) and taking the magnitude of the entire expression gives

$$|V(-l)| = |V_0^+| |1 + |\Gamma| e^{j(\theta - 2\beta l)}|. \tag{4.61}$$

An example of the voltage magnitude described by this equation is shown in Figure 4.4. Although the wave patterns for the incident and reflected waves move to the right and left, respectively, the pattern for the total voltage magnitude is stationary. This voltage pattern is called the standing wave pattern or envelope.

4.2.4 Impedance as a Function of Position

Although the current pattern has peaks and valleys just like the voltage pattern, the peaks and valleys do not occur at the same locations. As a result, the ratio of voltage to current varies continuously along the line. If the line were cut at some point a distance l from the load as shown in Figure 4.5, the input impedance seen looking into the line

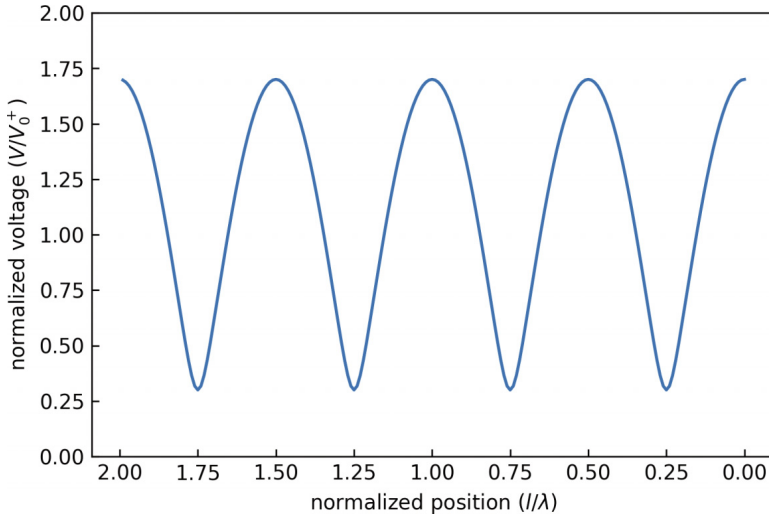
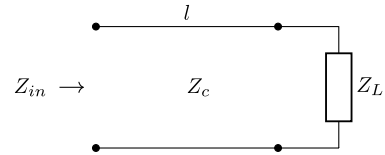


Figure 4.4 Voltage standing wave pattern along a transmission line with a mismatched load.

Figure 4.5 Impedance looking into a terminated transmission line of length l .



at that point would be the ratio of the voltage to current at that distance from the load:

$$\begin{aligned}
 Z_{in} &= \frac{V(-l)}{I(-l)} = \frac{V_0^+ [e^{j\beta l} + \Gamma e^{-j\beta l}]}{\frac{V_0^+}{Z_c} [e^{j\beta l} - \Gamma e^{-j\beta l}]} \\
 &= Z_c \frac{e^{j\beta l} + \Gamma e^{-j\beta l}}{e^{j\beta l} - \Gamma e^{-j\beta l}}.
 \end{aligned} \tag{4.62}$$

This can be expressed entirely in terms of the impedances using Eq. (4.53). Making the substitution for Γ and simplifying gives

$$\begin{aligned}
 Z_{in} &= Z_c \frac{(Z_L + Z_c) e^{j\beta l} + (Z_L - Z_c) e^{-j\beta l}}{(Z_L + Z_c) e^{j\beta l} - (Z_L - Z_c) e^{-j\beta l}} \\
 &= Z_c \frac{Z_L (e^{j\beta l} + e^{-j\beta l}) + Z_c (e^{j\beta l} - e^{-j\beta l})}{Z_L (e^{j\beta l} - e^{-j\beta l}) + Z_c (e^{j\beta l} + e^{-j\beta l})} \\
 &= Z_c \frac{Z_L \cos \beta l + j Z_c \sin \beta l}{j Z_L \sin \beta l + Z_c \cos \beta l} \\
 Z_{in} &= Z_c \frac{Z_L + j Z_c \tan \beta l}{Z_c + j Z_L \tan \beta l}.
 \end{aligned} \tag{4.63}$$

Since the tangent is periodic in π rather than 2π , the impedance is periodic in distances of half a wavelength rather than a wavelength.

Three limiting cases give some insight into the behavior of the impedance function (4.63). In the case of a short circuit, $Z_L = 0$ and the input impedance becomes

$$Z_{sc} = jZ_c \tan \beta l. \quad (4.64)$$

However, if the line is open circuited so that $Z_L \rightarrow \infty$, the input impedance becomes

$$Z_{oc} = -jZ_c \cot \beta l. \quad (4.65)$$

It is interesting to note the relationship

$$Z_{sc}Z_{oc} = Z_c^2. \quad (4.66)$$

This suggests an experimental technique for measuring the characteristic impedance of a transmission line!

If the line is electrically short, $\beta l \ll 1$, the tangent can be approximated by its argument and the cotangent by the inverse of its argument. Comparing with the circuit equations $X_L = j\omega L$ and $X_C = -j/\omega C$, we see that an electrically short length of short-circuited transmission line acts like an inductor whose inductance is proportional to length, and a short length of open-circuited transmission line acts like a capacitor whose capacitance is proportional to length.

Finally, if the line is terminated in the characteristic impedance of the line, i.e., $Z_L = Z_c$, then the input impedance is simply equal to the characteristic impedance regardless of the length of the transmission line.

4.2.5 Quarter Wave Transformer

From the previous discussions, we see that it is generally desirable for a transmission line to be terminated in a matched impedance; i.e., a real impedance equal to the characteristic impedance of the line. Under these conditions, the reflection coefficient will be zero, indicating that all of the incident power is absorbed by the load. Further, the input impedance of the line will simply be equal to the characteristic impedance regardless of the length.

The quarter wave transformer is a common and effective technique used for impedance matching. Consider the transmission line circuit terminated in a real impedance R_L illustrated in Figure 4.6.

For a quarter wavelength section of transmission line, the argument to the tangent functions in Eq. (4.63) is $\beta l = (2\pi/\lambda)(\lambda/4) = \pi/2$. Since the tangent increases without bounds as the argument approaches $\pi/2$, the tangent terms dominate the numerator

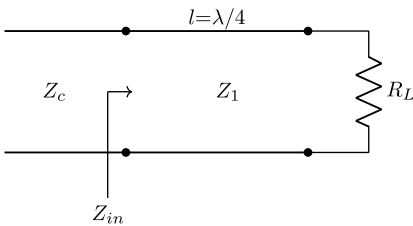


Figure 4.6 A real impedance can be matched to a lossless transmission line using a quarter wavelength line whose impedance is the geometric mean of the load and line to be matched. This is referred to as a quarter wave matching transformer.

and denominator of the expression for the input impedance (4.63), and we find

$$\lim_{\beta l \rightarrow \pi/2} Z_{in} = \frac{Z_1^2}{R_L}. \quad (4.67)$$

If we want to eliminate reflections (i.e., $\Gamma = 0$), then we want $Z_{in} = Z_c$. Imposing this condition on Eq. (4.67) leads to

$$Z_c = \frac{Z_1^2}{R_L}. \quad (4.68)$$

Consequently we see that we can match any real load R_L to a line with impedance Z_c provided we use a quarter-wavelength segment of transmission line whose characteristic impedance is the geometric mean of R_L and Z_c :

$$Z_1 = \sqrt{Z_c R_L}. \quad (4.69)$$

It is important to keep in mind, however, that this match will only be perfect at the single frequency where $l = \lambda/4$. At other frequencies there will be some reflections from the load. The bandwidth over which the SWR is 2:1 or less depends on the degree of mismatch between the load and the line to be matched. For example, if a 122 Ω line is used to match a 300 Ω load to a 50 Ω line, the impedance bandwidth for an SWR < 2:1 is about $\pm 20\%$. It is also important to keep in mind that this technique only works for real load impedances. However, any load impedance can be transformed into a real impedance at a given frequency by adding the appropriate lumped capacitance or inductance in series or parallel with the load. Further, since short segments of transmission line can be made to look like either inductors or capacitors, any load impedance can be transformed to a purely real impedance by putting another appropriate segment of transmission line in series or parallel with the load.

4.2.6 Coaxial, Microstrip, and Coplanar Lines

Although the ladder line is conceptually convenient since the geometry resembles its lumped-circuit model, it is mainly used as a low-loss feed line for high-frequency (i.e., below 30 MHz) antennas. It is not of interest for superconducting quantum computing systems owing to the lack of shielding and incompatibility with nanofabrication methods. Instead, *coaxial* lines (or coax for short) are used to convey signals between component blocks, and *coplanar* transmission lines are typically used to convey signals on quantum computing chips. The simpler *microstrip* lines are used in some components of interest for superconducting computer control and readout systems.

The detailed analysis of the fields and modes of these transmission lines are beyond the scope of our present discussion, but expressions for the characteristic impedance Z_c and effective relative permittivity ϵ_e are summarized below. In all cases, the wavelength along the line is given by

$$\lambda = \frac{\lambda_0}{\sqrt{\epsilon_e}}, \quad (4.70)$$

where λ_0 is the wavelength in free space.

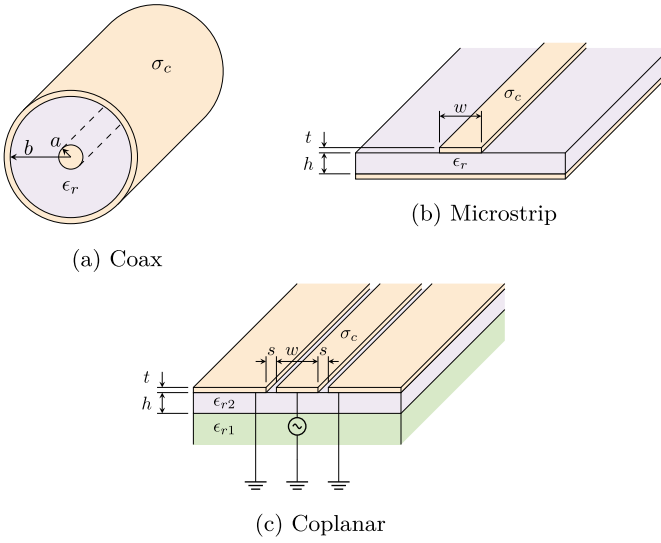


Figure 4.7 Some commonly-used types of transmission lines. Since there are multiple ways to drive the coplanar guide, the most common source connection is explicitly shown. All of the dielectric materials are assumed to be nonmagnetic for our purposes.

4.2.6.1 Coaxial Lines

The field structure and parameters for coaxial transmission lines are discussed in most introductory texts on electromagnetics, e.g., [19]. Referring to Figure 4.7(a), the characteristic impedance is

$$Z_c = \frac{1}{2\pi} \sqrt{\frac{\mu_0}{\epsilon_0 \epsilon_r}} \ln(b/a). \quad (4.71)$$

In this case $\epsilon_c = \epsilon_r$ since the coax is uniformly filled with dielectric.

4.2.6.2 Microstrip Lines

RF and microwave components such as filters, power dividers, and quadrature hybrids are often made using metallized substrates with transmission line patterns etched on one side using a process similar to that used to make printed circuit boards. Figure 4.7(b) describes the geometry and key dimensions. Unlike coax, simple and accurate analytical expressions are not available, and instead expressions are obtained by fitting approximate analytical expressions to the results of numerical modeling. In the common case of RF and microwave components described above, the conductor thickness t is much less than either the conductor width w or the substrate thickness h and can be neglected. Under this assumption, models for the key parameters have been discussed by a number of authors [20–24]. Here we quote the expressions for characteristic impedance given by Pozar [21]:

$$Z_c = \begin{cases} \frac{60}{\sqrt{\epsilon_e}} \ln\left(\frac{8h}{w} + \frac{w}{4h}\right) & \text{for } w/h \leq 1 \\ \frac{120\pi}{\sqrt{\epsilon_e}[w/h + 1.393 + 0.667 \ln(w/h + 1.444)]} & \text{for } w/h \geq 1, \end{cases} \quad (4.72)$$

and the effective relative dielectric constant is approximated by

$$\epsilon_e = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \frac{1}{\sqrt{1 + 12h/w}}. \quad (4.73)$$

4.2.6.3 Coplanar Waveguide

Microstrip lines require metallization on both sides of the circuit substrate. A geometry that is more compatible with nanofabrication techniques is the coplanar waveguide (CPW) shown in Figure 4.7(c). For example, the bottom dielectric could be a silicon substrate, and the top dielectric could be an oxide layer grown on the silicon. The planar metallization pattern could then be formed using conventional photolithographic techniques (similar to those used to make printed circuit boards). In such a case the substrate thickness is so much larger than the oxide thickness and the lateral dimensions of the transmission line that it can be considered infinite. Consequently only the thickness of the top dielectric layer (oxide in our silicon example) affects the transmission line properties.

We have shown the usual excitation connections for this transmission line, since three conductors permit excitation to be applied in multiple ways.

When the conductor thickness t can be neglected, analytical expressions for the characteristic impedance can be obtained using a mathematical technique called conformal mapping [25–27]. The result is

$$Z_c = 4\epsilon_0\epsilon_e \frac{K(k_0)}{K(k'_0)}. \quad (4.74)$$

Here K is the complete elliptic integral of the first kind, and its arguments are defined as

$$k_0 = \frac{w}{w + 2s}, \quad k'_0 = \sqrt{1 - k_0^2}. \quad (4.75)$$

The effective relative permittivity is

$$\epsilon_e = 1 + \frac{1}{2}(\epsilon_{r1} - 1) + q(\epsilon_{r2} - \epsilon_{r1}), \quad (4.76)$$

where

$$q = \frac{1}{2} \frac{K(k)K(k'_0)}{K(k')K(k_0)} \quad (4.77)$$

and

$$k = \frac{\sinh\left(\frac{\pi w}{4h}\right)}{\sinh\left(\frac{\pi(w+2s)}{4h}\right)}, \quad k' = \sqrt{1 - k^2}. \quad (4.78)$$

As an example, [26] analyzed a coplanar waveguide with $w = 10 \mu\text{m}$, $s = 6.6 \mu\text{m}$, $h = 550 \text{ nm}$, and $t = 200 \text{ nm}$. The line was fabricated on a high-resistivity silicon wafer with thickness $500 \mu\text{m}$. The substrate thickness is clearly much larger than the other transmission line dimensions, justifying the assumption of infinite thickness. In contrast, the conductor thickness in this case is not negligible, and the authors found that for their geometry, available analytic expressions such as Eq. (4.76) for the effective

relative permittivity were inaccurate. For such geometries, the effective permittivity must be calculated numerically using tools such as the finite element method.

The impedance expression (4.74) is based on the magnetic contributions to the inductance per unit length, i.e., the conventional inductance resulting from magnetic flux linking the current. However, in Chapter 7, we will see that in superconducting currents there is a kinetic contribution to the inductance associated with the inertia of charge carriers as well. This kinetic contribution to the inductance of superconducting coplanar waveguides has been considered by [26, 28], and is negligible when the temperature is significantly below the superconducting transition temperature, as is the case in quantum computing. Consequently, this kinetic contribution can be neglected in our cases of interest.

4.3 S Parameters

Because voltages and currents vary with position, it is often more convenient to characterize devices and circuits in a language that is more consistent with the notions of incident, reflected, and transmitted waves. Consider the two-port circuit shown in Figure 4.8. The incoming wave voltage amplitudes are denoted by a “+” superscript, and the outgoing wave voltage amplitudes are denoted by a “−” superscript. For a linear circuit, the incoming and outgoing waves are related by

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix}, \quad (4.79)$$

or

$$V^- = SV^+. \quad (4.80)$$

The elements of the S matrix are called S parameters and are defined as follows:

$$\begin{aligned} S_{11} &= \left. \frac{V_1^-}{V_1^+} \right|_{V_2^+=0} = \text{reflection coefficient from port 1 } (\Gamma_1), \\ S_{21} &= \left. \frac{V_2^-}{V_1^+} \right|_{V_2^+=0} = \text{transmission coefficient from port 1 to port 2}, \\ S_{12} &= \left. \frac{V_1^-}{V_2^+} \right|_{V_1^+=0} = \text{transmission coefficient from port 2 to port 1}, \\ S_{22} &= \left. \frac{V_2^-}{V_2^+} \right|_{V_1^+=0} = \text{reflection coefficient from port 2 } (\Gamma_2). \end{aligned} \quad (4.81)$$

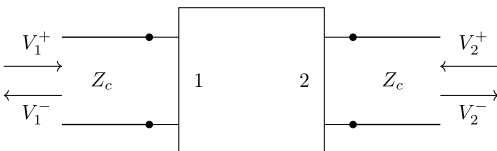


Figure 4.8 Incoming and outgoing wave amplitudes used in the definitions of S parameters.

4.3.1 Lossless Condition

For a lossless network, all the power that goes in must come out. For example, assuming all ports have the characteristic impedance Z_c , we require

$$\frac{|V_1^+|^2}{2Z_c} + \frac{|V_2^+|^2}{2Z_c} - \frac{|V_1^-|^2}{2Z_c} - \frac{|V_2^-|^2}{2Z_c} = 0. \quad (4.82)$$

Equivalently, this can be expressed

$$(V^+)^{\dagger} V^+ - (V^-)^{\dagger} V^- = 0, \quad (4.83)$$

where the superscript $\dagger$ represents the transpose complex conjugate. We can use the S matrix to express this entirely in terms of the incident wave voltage amplitudes:

$$\begin{aligned} (V^+)^{\dagger} V^+ - (SV^+)^{\dagger} SV^+ &= 0 \\ (V^+)^{\dagger} V^+ - (V^+)^{\dagger} S^{\dagger} SV^+ &= 0 \\ (V^+)^{\dagger} (I - S^{\dagger} S) V^+ &= 0. \end{aligned} \quad (4.84)$$

From this we conclude that

$$S^{\dagger} S = I \quad (4.85)$$

and the S matrix of a lossless network must be *unitary*.

4.3.2 Reciprocity

A microwave network is said to be *reciprocal* if the transmission from port i to port j is the same as from j to i . In other words, a reciprocal network will work the same if the input and output are interchanged. In terms of S parameters we require $S_{ij} = S_{ji}$. We conclude that the S matrix of a reciprocal network is symmetric.

4.4 Transmission (ABCD) Matrices

In many cases, microwave networks consist of a cascade of 2-port components. It is therefore useful to define a 2×2 transmission matrix that can be used to obtain the voltage at the output port from that of the input port. Transmission through a cascade of 2-ports can then be obtained by multiplying the transmission matrices. These matrices are also called *ABCD matrices* owing to the common symbols used for the matrix elements.

Unlike S parameters, the ABCD matrix for a 2-port is defined in terms of the total voltages and currents as defined in Figure 4.9. Note that the sign convention for current is directed in on the input, and out on the output. This ensures consistent definitions when multiple two-ports are cascaded. In terms of these quantities, the ABCD matrix is defined as

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix}. \quad (4.86)$$

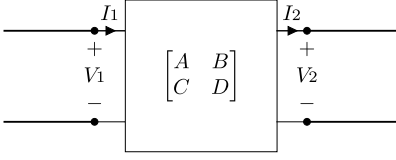


Figure 4.9 Definition of voltages and currents for the ABCD transmission matrix.

From this equation we obtain the definitions of the matrix components as

$$A = \left. \frac{V_1}{V_2} \right|_{I_2=0}, \quad (4.87)$$

$$B = \left. \frac{V_1}{I_2} \right|_{V_2=0}, \quad (4.88)$$

$$C = \left. \frac{I_1}{V_2} \right|_{I_2=0}, \quad (4.89)$$

$$D = \left. \frac{I_1}{I_2} \right|_{V_2=0}. \quad (4.90)$$

As an example of the construction of an ABCD matrix, consider the length of lossless transmission line shown in Figure 4.10. The total voltage and current as functions of position along the line are given by

$$V(z) = V_0^+(e^{-j\beta z} + \Gamma e^{j\beta z}), \quad (4.91)$$

$$I(z) = \frac{V_0^+}{Z_c}(e^{-j\beta z} - \Gamma e^{j\beta z}), \quad (4.92)$$

where Γ is the reflection coefficient at port 2, and V_0^+ is the amplitude of the wave incident wave from the left.

The voltages and currents at the ports are given by

$$V_1 = V(-l) = V_0^+(e^{j\beta l} + \Gamma e^{-j\beta l}), \quad (4.93)$$

$$I_1 = I(-l) = \frac{V_0^+}{Z_c}(e^{j\beta l} - \Gamma e^{-j\beta l}), \quad (4.94)$$

$$V_2 = V(0) = V_0^+(1 + \Gamma), \quad (4.95)$$

$$I_2 = I(0) = \frac{V_0^+}{Z_c}(1 - \Gamma). \quad (4.96)$$

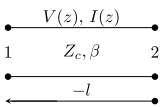


Figure 4.10 Definitions for constructing the ABCD matrix of a section of transmission line.

Referring to Eqs. (4.87) and (4.89), we see that A and C are calculated with $I_2 = 0$. This implies an open circuit at port 2, for which $\Gamma = +1$ from Eq. (4.96). This gives

$$\begin{aligned} A &= \left. \frac{V_1}{V_2} \right|_{I_2=0} = \frac{e^{j\beta l} + e^{-j\beta l}}{2} \\ &= \cos \beta l, \end{aligned} \quad (4.97)$$

and

$$\begin{aligned} C &= \left. \frac{I_1}{V_2} \right|_{I_2=0} = \frac{e^{j\beta l} - e^{-j\beta l}}{2Z_c} \\ &= j \frac{1}{Z_c} \sin \beta l = jY_c \sin \beta l. \end{aligned} \quad (4.98)$$

Similarly, the definitions of B and D from Eqs. (4.88) and (4.90) require $V_2 = 0$, implying a short circuit at port 2, and $\Gamma = -1$ from Eq. (4.95). It follows that

$$\begin{aligned} B &= \left. \frac{V_1}{I_2} \right|_{V_2=0} = \frac{e^{j\beta l} - e^{-j\beta l}}{2/Z_c} \\ &= jZ_c \sin \beta l, \end{aligned} \quad (4.99)$$

and

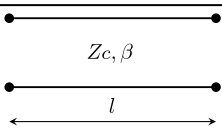
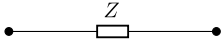
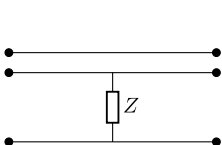
$$\begin{aligned} D &= \left[\frac{I_1}{I_2} \right]_{V_2=0} = \frac{e^{j\beta l} + e^{-j\beta l}}{2} \\ &= \cos \beta l. \end{aligned} \quad (4.100)$$

The ABCD matrix for the section of transmission line is therefore

$$\begin{bmatrix} \cos \beta l & jZ_c \sin \beta l \\ jY_c \sin \beta l & \cos \beta l \end{bmatrix}. \quad (4.101)$$

The ABCD matrices for several useful 2-ports are given in Table 4.1.

Table 4.1 Useful ABCD matrices.

Circuit	ABCD matrix
	$\begin{bmatrix} \cos \beta l & jZ_c \sin \beta l \\ jY_c \sin \beta l & \cos \beta l \end{bmatrix}$
	$\begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} 1 & 0 \\ 1/Z & 1 \end{bmatrix}$

As mentioned above, the input voltages and currents of a series of cascaded two-ports are obtained by simply multiplying the ABCD matrices of the two-ports. For example, the voltage and current at the input to two cascaded two-ports can be obtained in terms of the voltage and current at the output as follows:

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} \begin{bmatrix} V_3 \\ I_3 \end{bmatrix}. \quad (4.102)$$

Although ABCD matrices are convenient for analyzing cascaded networks, ultimately one is often interested in the S parameters of the equivalent 2-port. As an example, consider S_{11} for the general case of a terminated transmission line of impedance Z_1 connected to port 1, and a terminated transmission line of impedance Z_2 connected to port 2 [29]. The voltage at port 2 will be related to the current as $I_2 = V_2/Z_2$:

$$\begin{aligned} \begin{bmatrix} V_1 \\ I_1 \end{bmatrix} &= \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} 1 \\ 1/Z_2 \end{bmatrix} V_2 \\ &= \begin{bmatrix} A + B/Z_2 \\ C + D/Z_2 \end{bmatrix} V_2. \end{aligned} \quad (4.103)$$

The impedance Z_{in1} looking into port 1 is then

$$Z_{in1} = \frac{V_1}{I_1} = \frac{A + B/Z_2}{C + D/Z_2}. \quad (4.104)$$

Since S_{11} is equal to the reflection coefficient Γ , we have

$$S_{11} = \Gamma = \frac{Z_{in1} - Z_1}{Z_{in1} + Z_1}. \quad (4.105)$$

Substituting (4.104) into (4.105) and simplifying gives

$$S_{11} = \frac{A + B/Z_2 - CZ_1 - DZ_1/Z_2}{A + B/Z_2 + CZ_1 + DZ_1/Z_2}. \quad (4.106)$$

To obtain S_{21} , we now return to the top equation in (4.103):

$$V_1 = (1 + S_{11})V_1^+ = (A + B/Z_2)V_2. \quad (4.107)$$

Solving for V_2/V_1^+ , substituting in for S_{11} , and simplifying gives

$$S_{21} = \frac{2}{A + B/Z_2 + CZ_1 + DZ_1/Z_2}. \quad (4.108)$$

To obtain S_{22} we first invert the ABCD matrix to obtain

$$\begin{bmatrix} V_2 \\ I_2 \end{bmatrix} = \frac{1}{AD - BC} \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_1 \\ I_1 \end{bmatrix}. \quad (4.109)$$

However, referring to Figure 4.9, I_1 is pointing into the network and I_2 is pointing out of the network. To consider S_{22} we need to reverse these directions, so we introduce

$I'_1 = -I_1$ and $I'_2 = -I_2$, with $Z_{\text{in}2} = V_2/I'_2$ and $Z_1 = V_1/I'_1$. The inverted ABCD matrix equation then becomes

$$\begin{aligned} \begin{bmatrix} V_2 \\ -I'_2 \end{bmatrix} &= \frac{1}{AD - BC} \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} 1 \\ -1/Z_1 \end{bmatrix} V_1 \\ &= \frac{1}{AD - BC} \begin{bmatrix} D + B/Z_1 \\ -C - A/Z_1 \end{bmatrix} V_1. \end{aligned} \quad (4.110)$$

The impedance $Z_{\text{in}2}$ looking into port 2 is then

$$Z_{\text{in}2} = \frac{V_2}{I'_2} = \frac{D + B/Z_1}{C + A/Z_1}. \quad (4.111)$$

Similar to before, S_{22} is given by

$$\begin{aligned} S_{22} &= \frac{Z_{\text{in}2} - Z_2}{Z_{\text{in}2} + Z_2} \\ &= \frac{D + B/Z_1 - CZ_2 - AZ_2/Z_1}{D + B/Z_1 + CZ_2 + AZ_2/Z_1}. \end{aligned} \quad (4.112)$$

Finally, to obtain S_{12} , we return to the top equation in (4.110):

$$V_2^+(1 + S_{22}) = \frac{D + B/Z_1}{AD - BC} V_1. \quad (4.113)$$

Solving for V_1/V_2^+ , substituting for S_{22} , and simplifying leads to

$$S_{12} = \frac{2(AD - BC)(Z_1/Z_2)}{A + B/Z_2 + CZ_1 + DZ_1/Z_2}. \quad (4.114)$$

In the most common case of $Z_1 = Z_2 \equiv Z_c$, the equations relating S parameters to the ABCD matrix reduce to the following:

$$S_{11} = \frac{A + B/Z_c - CZ_c - D}{A + B/Z_c + CZ_c + D}, \quad (4.115)$$

$$S_{22} = \frac{-A + B/Z_c - CZ_c + D}{A + B/Z_c + CZ_c + D}, \quad (4.116)$$

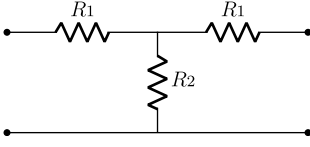
and for reciprocal networks,

$$S_{12} = S_{21} = \frac{2}{A + B/Z_c + CZ_c + D}. \quad (4.117)$$

In obtaining this expression, we see that we must require $AD - BC = 1$ for reciprocal networks with the same impedance at each port.

4.5 Attenuators

Attenuators are reciprocal components that are used to reduce the amplitude of a signal on a transmission line. Ideally, we would like to do this without creating reflections along the line. This means, of course, that we want both ports to be matched to the characteristic impedance of the line. It follows from reciprocity that the attenuator has

**Figure 4.11** Circuit for an attenuator.

the same behavior regardless of which port is used as the input and which is used as the output.

One way to realize such a component is with a symmetric lumped-element circuit as shown in Figure 4.11. The equivalent ABCD matrix for this circuit can be obtained by multiplying three matrices: the first for a series impedance of R_1 , the second for a shunt impedance of R_2 , and the third for another series impedance of R_1 . Referring to Table 4.1 we have:

$$\begin{bmatrix} 1 & R_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1/R_2 & 1 \end{bmatrix} \begin{bmatrix} 1 & R_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 + R_1/R_2 & R_1(1 + R_1/R_2) \\ 1/R_2 & 1 + R_1/R_2 \end{bmatrix}. \quad (4.118)$$

First, let us consider the conditions that provide matched ports. In terms of S parameters, this implies $S_{11} = S_{22} = 0$. From Eq. (4.115), this implies $A + B/Z_c = CZ_c + D$, or $B/Z_c = CZ_c$ since $A = D$ (see Eq. (4.118)). We have then,

$$\begin{aligned} \frac{1}{Z_c} \left(R_1 + \frac{R_1^2}{R_2} \right) &= \frac{Z_c}{R_2} \\ \frac{R_2}{Z_c} \left(\frac{R_1}{R_2} + \frac{R_1^2}{R_2^2} \right) &= \frac{Z_c}{R_2} \end{aligned} \quad (4.119)$$

or

$$R_2 = \frac{Z_c}{\sqrt{r + r^2}}, \quad (4.120)$$

where $r = R_1/R_2$.

Next we want to choose the resistor ratio $r = R_1/R_2$ so that the desired attenuation is obtained. If G is the desired power gain, then we want to require $S_{12} = S_{21} = \sqrt{G}$. Since $A = D$ and $B/Z_c = CZ_c$, S_{12} from Eq. (4.117) can be written

$$\begin{aligned} S_{12} &= \frac{2}{2(A + CZ_c)} \\ &= \frac{2}{2(1 + r + Z_c/R_2)} \\ &= \frac{1}{1 + r + \sqrt{r + r^2}}. \end{aligned} \quad (4.121)$$

With $\sqrt{G} = S_{12}$, this can be written

$$\sqrt{r + r^2} = \frac{1}{\sqrt{G}} - 1 - r. \quad (4.122)$$

Squaring both sides and solving for r gives

$$r = \frac{\sqrt{G}}{2} \left(\frac{1}{\sqrt{G}} - 1 \right)^2. \quad (4.123)$$

To summarize, we first choose G for the desired attenuation, and calculate the required resistor ratio r from Eq. (4.123). Once r is known, R_2 is given by Eq. (4.120), and $R_1 = rR_2$.

4.6 Circulators and Isolators

To explore the implications of reciprocity a bit more, consider a 3-port, lossless network:

$$\begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}. \quad (4.124)$$

If all ports are matched, $S_{11} = S_{22} = S_{33} = 0$. Requiring the S matrix to be unitary leads to

$$\begin{bmatrix} 0 & S_{21}^* & S_{31}^* \\ S_{12}^* & 0 & S_{32}^* \\ S_{13}^* & S_{23}^* & 0 \end{bmatrix} \begin{bmatrix} 0 & S_{12} & S_{13} \\ S_{21} & 0 & S_{23} \\ S_{31} & S_{32} & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.125)$$

$$\begin{bmatrix} |S_{21}|^2 + |S_{31}|^2 & S_{31}^* S_{32} & S_{21}^* S_{23} \\ S_{32}^* S_{31} & |S_{12}|^2 + |S_{32}|^2 & S_{12}^* S_{13} \\ S_{23}^* S_{21} & S_{13}^* S_{12} & |S_{13}|^2 + |S_{23}|^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This represents nine simultaneous nonlinear equations, six of which are independent. To find a solution, let us assume for the moment that $S_{21} \neq 0$. From the equation on the lower left, it immediately follows that $S_{23} = 0$. But in this case, the equation on the lower right requires $|S_{13}| = 1$. This in turn requires $S_{12} = 0$ from the bottom center equation, leading to $|S_{32}| = 1$ from the center equation. The center left equation then gives $S_{31} = 0$. Finally, the top left equation then requires $|S_{21}| = 1$. We conclude that the form of the S matrix is

$$\begin{bmatrix} 0 & 0 & S_{13} \\ S_{21} & 0 & 0 \\ 0 & S_{32} & 0 \end{bmatrix}. \quad (4.126)$$

Note that this equation is not symmetric and so is nonreciprocal; the signal can flow from ports $1 \rightarrow 2$, $2 \rightarrow 3$, and $3 \rightarrow 1$, but not in the reverse direction. This device is called a *circulator*. We conclude that any matched, lossless 3-port circuit must be nonreciprocal. But any circuit made entirely of reciprocal components such as inductors, capacitors, and resistors must be reciprocal, so we conclude that no reciprocal 3-port circuit can be perfectly matched.

It turns out that it is possible to realize a nonreciprocal circuit, but it requires something more than inductors, capacitors, and resistors. In particular, passive nonreciprocal devices can be realized by including magnetized material in the device. The detailed analysis of the electrodynamics of nonreciprocal devices is beyond the scope of this discussion, but we can get some understanding from the behavior of two-state quantum systems discussed in Chapter 2. Generally speaking, the laws of physics have *time-reversal symmetry*, i.e., the equations remain valid with the substitution $t \rightarrow -t$.

We know from Chapter 2 that a spin in a magnetic field precesses in a direction determined by the direction of the applied magnetic field. Since the magnetic moment of a material generally arises from some combination of electron orbital and spin angular momentum, it turns out that when placed in a magnetic field, the magnetization also precesses with a preferred sense.

Now in a reciprocal network, interchanging the input and the output is equivalent to making the substitution $t \rightarrow -t$. However, if magnetic materials are present, simply interchanging the input and output will not reverse the sense of precession; to obtain true time reversal would also require reversing the direction of the magnetic field. So if the direction of the magnetic field remains the same, the circuit will in general have different behavior if the input and output are interchanged. This is the fundamental origin of nonreciprocal behavior in passive devices. The most common implementation of nonreciprocal devices uses insulating magnetic materials called *fer-rites*. For more information about the operation of nonreciprocal ferrite devices, see [21, 30].

Commonly used symbols for circulators are shown in Figure 4.12. In these symbols, the connections can be viewed as signal lines, with an implied ground comprising the second conductor of the transmission lines. In actual devices, the ports are oriented at 120° intervals, as shown on the symbol on the left. However, in drawing circuits, it is often convenient to draw the ports at 90° intervals as shown in the center symbol.

It is interesting to consider how the operation of the device depends on the impedance terminating the ports in the circuit. For example, consider that port 2 is terminated in a matched load. In this case, power that enters port 1 will be transferred to the load at port 2, and no reflections will occur. In this case port 3 sees no signal at all. But what if the load at port 2 is not perfectly matched? In this case there will be some reflected power, which will appear as an input signal to port 2. The power entering port 2 will be conveyed to port 3, and if port 3 is connected to a perfectly matched load as shown in the right symbol in Figure 4.12, the reflected power will be dissipated in the load on port 3 and none of the reflected power will return to the original source at port 1. Devices like this that isolate the source from the load are referred to as *isolators*. They play an important role in preventing noise from reaching a qubit, while still allowing the weak signal from the qubit to be measured.

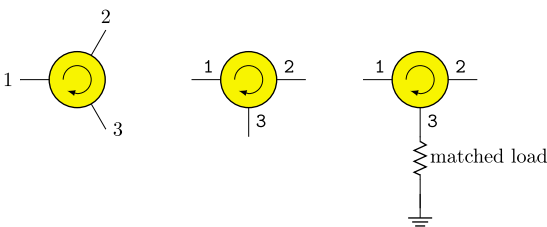


Figure 4.12 Circulator Circuit diagrams. In actual devices, the three ports are symmetrically placed at 120° angles, as represented on the left. However, in drawing circuit diagrams, it is often convenient to show the ports at right angles, as shown in the center. As shown on the right, you can make an isolator by connecting one of the ports to a matched load.

4.7 Power Dividers/Combiners

If we need to connect the same signal to two different high-impedance inputs at low frequencies, we would simply wire them together directly. However, at RF frequencies where transmission line effects must be considered, we need a component called a *splitter* or *divider* to connect a signal to two different loads. The reason is that each connection represents a transmission line, and if the two loads were placed in parallel, the impedance seen by the input line would be $1/2$ its characteristic impedance, resulting in reflections to previous parts of the circuit.

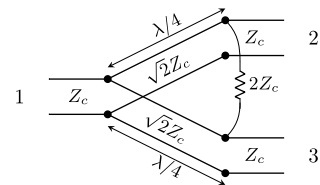
The particular type of divider depicted in Figure 4.13 is known as a *Wilkinson divider*. The input line splits into two $\lambda/4$ lines with characteristic impedance $\sqrt{2}Z_c$, where Z_c is the characteristic impedance of the lines external to the components. From (4.68), the input to each branch of the $\lambda/4$ lines would be $(\sqrt{2}Z_c)^2/Z_c = 2Z_c$. Consequently, when the two branches are placed in parallel, the effective impedance will be Z_c , and the input line will be matched.

Since the output voltages in the two branches are symmetric, there is no potential difference across the resistor, and the resistor has no effect on the circuit. Consequently, in theory a Wilkinson divider can be lossless (in practice there will always be some loss owing to the finite conductivity of the conductors, dielectric losses, etc.).

In contrast, when the circuit is used as a combiner for signals that are not identical, then there will be a potential difference across the resistor and some of the power will be dissipated.³ However, the value of the resistor can be chosen so that ports 2 and 3 are matched (the proper value for the resistor turns out to be $2Z_c$). Consequently although all ports are matched, the circuit is not lossless when used as a combiner. (We saw in the previous section that a lossless 3-port device with all ports matched must be a *circulator* instead of a divider/combiner.) A detailed analysis of the Wilkinson divider can be found in [21].

It is also possible to realize a divider with a 4-port network. Figure 4.14 shows a type of 4-port where the ports are connected with quarter-wave branches of transmission line with the impedances shown. This is a type of branch coupler called a *quadrature hybrid*. As we will see, this coupler divides power applied to port 1 equally between ports 2 and 3, with port 4 isolated (i.e., there is no coupling between port 1 and port 4). In addition, the signals at the output ports 2 and 3 are 90 degrees out of phase—a useful feature enabling sine and cosine signals to be generated from a single local oscillator.

Figure 4.13 Wilkinson power divider.



³ If the amplitudes and phases of the signals to be combined were precisely time-reversed versions of the output signals when used as a splitter, then by reciprocity the device would be lossless. However, in general, arbitrary signals to be combined result in some loss in the resistor.

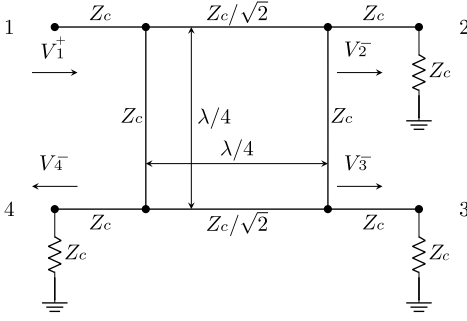
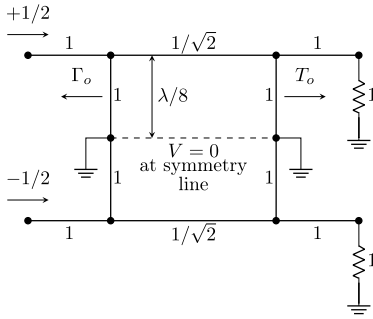
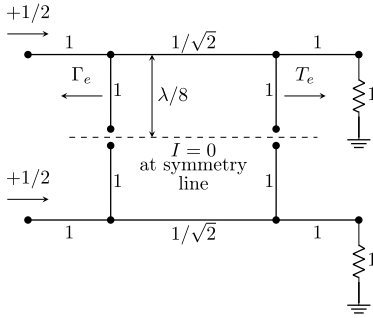


Figure 4.14 Quadrature hybrid 4-port network. For clarity each transmission line is represented by a single line, with the return conductors understood (e.g., an implied ground plane) [24].



(a) Odd mode excitation.



(b) Even mode excitation.

To understand how the quadrature hybrid works, it is helpful to consider the network response to even (in-phase) and odd (out-of-phase) signals applied to ports 1 and 3 as shown in Figure 4.15. In the case of an odd excitation (Figure 4.15(a)), the voltage must be zero along the dashed line owing to symmetry, while the current is zero along the dashed symmetry line for an even excitation (Figure 4.15(b)). In both cases, the two sides of the 4 port network are effectively de-coupled. The resulting 2-ports can then be analyzed to obtain the reflection and transmission coefficients $\Gamma_{e,o}$ and $T_{e,o}$. Adding the signals for the two cases gives the response of the 4-port to a single excitation applied to port 1.

The resulting network S parameters are then given by

$$S_{11} = \frac{1}{2}(\Gamma_e + \Gamma_o), \quad (4.127)$$

$$S_{21} = \frac{1}{2}(T_e + T_o), \quad (4.128)$$

$$S_{31} = \frac{1}{2}(T_e - T_o), \quad (4.129)$$

$$S_{41} = \frac{1}{2}(\Gamma_e - \Gamma_o). \quad (4.130)$$

First, consider the odd excitation as shown in Figure 4.15(a). We can analyze this as the cascade of a quarter wavelength section of transmission line with normalized impedance of $1/\sqrt{2}$ between two shunt impedances consisting of shorted $1/8$ wavelength stubs. From Eq. (4.64) we have $Z_{sc} = j \tan \beta l$ for the normalized impedance of the stub, or with $\beta l = \pi/4$, $Z_{sc} = j$. The ABCD matrix for the network of cascaded 2-ports is:

$$\begin{aligned} \begin{bmatrix} A_o & B_o \\ C_o & D_o \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ -j & 1 \end{bmatrix} \begin{bmatrix} 0 & j/\sqrt{2} \\ j\sqrt{2} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -j & 1 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & j \\ j & 1 \end{bmatrix}. \end{aligned} \quad (4.131)$$

The odd-mode transmission and reflection coefficients are then

$$\Gamma_o = \frac{A_o + B_o - C_o - D_o}{A_o + B_o + C_o + D_o} = \frac{1 + j - j - 1}{1 + j + j + 1} = 0, \quad (4.132)$$

$$T_o = \frac{2}{A_o + B_o + C_o + D_o} = \frac{2}{(1 + j + j + 1)/\sqrt{2}} = \frac{1 - j}{\sqrt{2}}. \quad (4.133)$$

Following a similar procedure for the even-mode excitation in Figure 4.15(b), we note that the normalized input impedance of an open-circuited stub is $Z_{oc} = -j \cot \beta l$, or $Z_{oc} = -j$ for the $\lambda/8$ stub. The ABCD matrix for the even-mode excitation is

$$\begin{aligned} \begin{bmatrix} A_e & B_e \\ C_e & D_e \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ j & 1 \end{bmatrix} \begin{bmatrix} 0 & j/\sqrt{2} \\ j\sqrt{2} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ j & 1 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & j \\ j & -1 \end{bmatrix}. \end{aligned} \quad (4.134)$$

The even-mode transmission and reflection coefficients are then

$$\Gamma_e = \frac{A_e + B_e - C_e - D_e}{A_e + B_e + C_e + D_e} = \frac{-1 + j - j + 1}{-1 + j + j - 1} = 0, \quad (4.135)$$

$$T_e = \frac{2}{A_e + B_e + C_e + D_e} = \frac{2}{(-1 + j + j - 1)/\sqrt{2}} = -\frac{1 + j}{\sqrt{2}}. \quad (4.136)$$

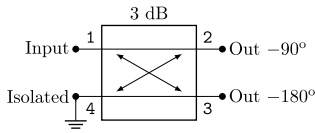


Figure 4.16 Commonly-used symbol for a quadrature hybrid. A signal applied to port 1 is evenly split into quadrature signals at ports 2 and 3.

The key S parameters can now be calculated:

$$S_{11} = \frac{1}{2} (\Gamma_e + \Gamma_o) = 0, \quad (4.137)$$

$$S_{21} = \frac{1}{2} (T_e + T_o) = -\frac{j}{\sqrt{2}}, \quad (4.138)$$

$$S_{31} = \frac{1}{2} (T_e - T_o) = -\frac{1}{\sqrt{2}}, \quad (4.139)$$

$$S_{41} = \frac{1}{2} (\Gamma_e - \Gamma_o) = 0. \quad (4.140)$$

We see then that the power input at port 1 will be equally divided between ports 2 and 3, but the signal at port 2 will be shifted by $-\pi/2$, and the signal at port 3 will be shifted by π . This behavior is summarized in Figure 4.16 in terms of a commonly-used symbol for a quadrature hybrid.

Alternatively, if we take the signal at port 2 as the reference cosine, then port 3 becomes the sine owing to the relative phase shift of $-\pi/2$, providing us with the quadrature local oscillator signals needed for the system shown in Figure 1.12.

4.8 Mixers

In microwave systems, a *mixer* is a device that multiplies two waveforms together:

$$f_{\text{out}}(t) = f_{\text{in } 1}(t)f_{\text{in } 2}(t). \quad (4.141)$$

The commonly-used symbol is shown in Figure 4.17. As with the circulator, the connections represent signal lines with an implied ground (not shown) completing the two conductors needed for a transmission line.

A common way to realize a mixer is to use the non-linear I - V characteristic of a junction diode. The current in a diode is given by

$$I(V) = I_S (e^{qV/nk_B T} - 1), \quad (4.142)$$

where I_S is the saturation current of the diode when reverse biased, q is the electronic charge, k_B is Boltzmann's constant, n is an ideality factor (generally between 1 and 2 for silicon), and T is the absolute temperature. For silicon at room temperature, $k_B T/q$ is about 26 mV. For small voltage excursions about a steady state value $V = V_0 + v$, the

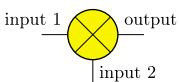


Figure 4.17 Mixer Circuit diagrams. In an ideal mixer, the signal at the output terminal is the product of the signals applied to the two input ports.

diode current can be expanded in a Taylor series:

$$I(V) \approx I_0 + v \left. \frac{dI}{dV} \right|_{V_0} + \frac{1}{2} v^2 \left. \frac{d^2 I}{dV^2} \right|_{V_0} + \dots \quad (4.143)$$

If $v = v_1 + v_2$, then the second-order term in (4.143) will contain product terms of the form (4.141).

Mixers are typically used for two primary functions: controlling the shape of the signal envelope to control a pulse shape or encode information on the signal (*modulation*), or translating a signal either up or down in frequency.

As an example of pulse shaping in quantum computing systems, pulses must be applied to qubits to control rotations on the Bloch sphere, where the duration of the pulse determines the amount of rotation. Figure 4.18(a) shows a circuit for controlling the shape of a pulse, and Figure 4.18(b) shows the waveforms at each terminal. In this context, the $\cos(\omega t)$ term is referred to as the “carrier,” and $f_p(t)$ is referred to as the “baseband pulse.”

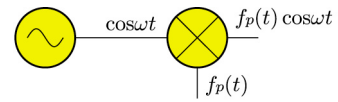
As an example of translating a signal to a different frequency, consider the situation of applying the inputs $f_p(t) \cos \omega_1 t$ and $\cos \omega_2 t$ to the inputs of a mixer. In this type of circuit, the frequency ω_2 is referred to as the “local oscillator” or “LO” frequency. In this case the output signal would be

$$\begin{aligned} f_{\text{out}} &= f_p(t) \cos \omega_1 t \cos \omega_2 t \\ &= \frac{1}{2} f_p(t) [\cos(\omega_1 - \omega_2)t + \cos(\omega_1 + \omega_2)t], \end{aligned} \quad (4.144)$$

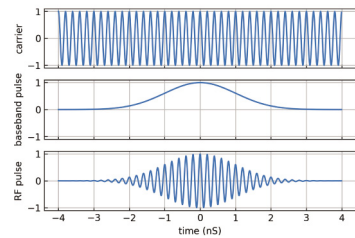
where we have expanded the cosine product with the identity $2 \cos u \cos v = \cos(u - v) + \cos(u + v)$. We see that the output will contain pulses at both the sum and difference frequencies. Typically only one of these will be desired, and the desired signal can be selected using either a low-pass or a high-pass filter. In particular, if we were to choose $\omega_1 = \omega_2$, then use of a low-pass filter would enable recovery of the original pulse $f_p(t)$.

A particularly interesting case is when the pulse carrier frequency has an arbitrary phase with respect to the LO signal. Such a signal can be expressed as the sum of a sine

Figure 4.18 (a) Circuit to shape a microwave pulse. (b) Example of creating a pulse with a Gaussian pulse shape.



(a)



(b)

component and a cosine component whose amplitudes are determined by the phase:

$$f_p(t) \cos(\omega t - \phi) = f_p(t) [\cos \phi \cos \omega t + \sin \phi \sin \omega t]. \quad (4.145)$$

In this case the output of the mixer would be

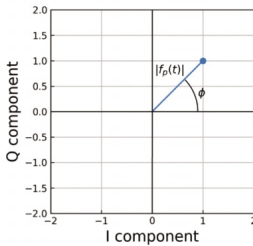
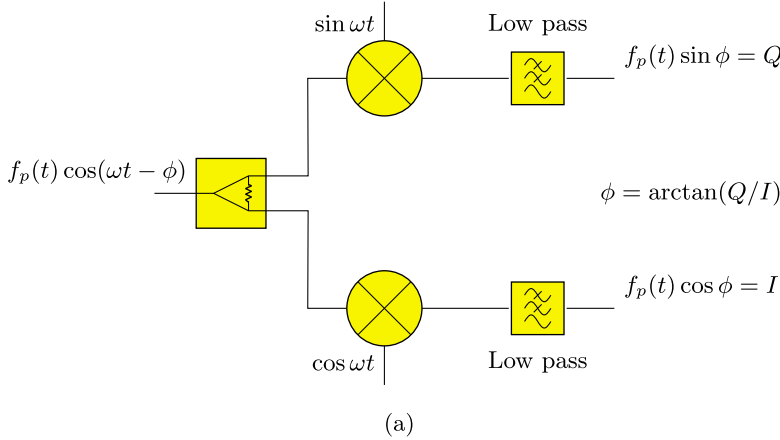
$$\begin{aligned} f_{\text{out}} &= f_p(t) \cos(\omega t - \phi) \cos \omega_{\text{LO}} t \\ &= \frac{1}{2} f_p(t) [\cos((\omega - \omega_{\text{LO}})t - \phi) + \cos((\omega + \omega_{\text{LO}})t - \phi)]. \end{aligned} \quad (4.146)$$

Comparing with (4.145) we see that if $\omega_{\text{LO}} = \omega$, then the difference signal is proportional to the amplitude of the cosine component of the signal pulse. Similarly, if the LO signal is a sine instead of a cosine, the output would be

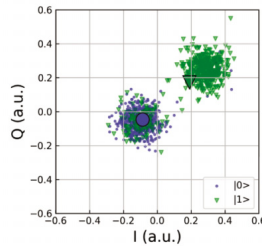
$$\begin{aligned} f_{\text{out}} &= f_p(t) \cos(\omega t - \phi) \sin \omega_{\text{LO}} t \\ &= \frac{1}{2} f_p(t) [\sin((\omega + \omega_{\text{LO}})t - \phi) - \sin((\omega - \omega_{\text{LO}})t - \phi)]. \end{aligned} \quad (4.147)$$

In this case we see that the difference frequency component is proportional to the amplitude of the *sine* component of the signal pulse.

Using these observations, we can construct a circuit to measure the phase of the signal as shown in Figure 4.19(a). The low-pass filters block the 2ω signal component coming from the mixer, leaving only the difference-frequency signal. The *I* and *Q*



(b)



(c)

Figure 4.19 (a) Circuit to recover the cosine (in-phase, or *I*) and sine (quadrature, or *Q*) components of an RF signal. (b) The amplitude and phase plotted on the *IQ* plane. (c) Raw *I-Q* signals measured on *ibm-q-armonk*, a 1 qubit demonstration processor on the IBM Q Network. The means for each state are indicated by the large markers.

outputs can be used to easily construct both the magnitude $|f_p(t)| = \sqrt{I^2 + Q^2}$ and phase $\phi = \arctan Q/I$ of the signal. The magnitude and phase are often plotted on the I - Q plane, as shown in Figure 4.19(b). When the signals reflected from a qubit are plotted on the I - Q plane, the signal points from $|0\rangle$ and $|1\rangle$ states cluster separately, enabling the states to be identified. An example signal constellation showing $|0\rangle$ and $|1\rangle$ states using an IBM Q processor is shown in Figure 4.19(c).

4.9 Low-Pass Filters

As we found in the previous section, the product of two sinusoidal signals obtained from a mixer contains sum and difference frequencies. In the case of the I - Q demodulator shown in Figure 4.19, we are only interested in the difference frequency component, and so we need a low-pass filter—i.e., a filter that only passes frequencies below a certain frequency—to isolate the desired components. In addition, if a digitally-sampled signal is used to define the pulse shape in a circuit such as Figure 4.18, signals will also appear at harmonics of the sampling frequency (these are referred to as *aliased signals*). Again, a low-pass filter is needed to remove these spurious signals from the output.

Simple “ π ” and “ T ” circuits illustrating the operation of a low-pass filter are shown in Figure 4.20. Intuitively, the blocking of high-frequency signals results from the fact that the impedance of the series inductors increases with frequency while the impedance of the shunting capacitors decreases with frequency. As an example, we will analyze the T network shown in Figure 4.20(b).

We have already analyzed a similar T network when we considered the attenuator design shown in Figure 4.11. Making the identifications $R_1 = j\omega L/2$ and $R_2 = 1/(j\omega C)$, we can obtain the ABCD matrix from Eq. (4.118). The result is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 - \omega^2 LC/2 & j\omega L/(1 - \omega^2 LC/4) \\ j\omega C & 1 - \omega^2 LC/2 \end{bmatrix}. \quad (4.148)$$

To see if the network can be matched, let us explore the conditions under which S_{11} (or S_{22}) can vanish. As with the attenuator circuit, $A = D$ owing to symmetry, so referring to Eq. (4.115), $S_{11} = 0$ when $B/Z_c = CZ_c$. This gives

$$\frac{j\omega L}{Z_c} \left(1 - \frac{\omega^2 LC}{4} \right) = j\omega CZ_c, \quad (4.149)$$

or

$$\frac{L}{C} \left(1 - \frac{\omega^2 LC}{4} \right) = Z_c^2. \quad (4.150)$$

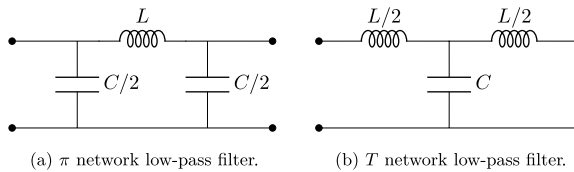


Figure 4.20 Low-pass filter circuits.

The form of this equation suggests the definition of a characteristic frequency

$$\omega_c = \frac{2}{\sqrt{LC}}. \quad (4.151)$$

The significance of this frequency will become apparent shortly. Since we are particularly interested in a match at low frequencies that we want to pass, let us consider Eq. (4.150) in the limit $\omega \rightarrow 0$. We conclude that to ensure the best match, we require⁴

$$\sqrt{\frac{L}{C}} = Z_c. \quad (4.152)$$

Next, to check that the circuit works as a low-pass filter, we want to see how S_{21} varies with frequency. From (4.117) and (4.148), we have

$$S_{21} = \frac{2}{2(1 - \omega^2 LC/2) + (1 - \omega^2 LC/4)j\omega L/Z_c + j\omega CZ_c}. \quad (4.153)$$

Recognizing that $L/Z_c = CZ_c = \sqrt{LC} = 2/\omega_c$, S_{21} can be simplified to

$$S_{21} = \frac{1}{1 - 2j\omega/\omega_c - 2(\omega/\omega_c)^2 + j(\omega/\omega_c)^3}. \quad (4.154)$$

Clearly for small ω we have $S_{21} \approx 1$, as expected. Let us consider the case $\omega = \omega_c$. After substituting and simplifying, we find that

$$S_{21} = -\frac{1}{2}(1 - j), \quad |S_{21}|^2 = \frac{1}{2}. \quad (4.155)$$

We see then that ω_c is the frequency at which the transmitted power drops by half, referred to as the “cutoff” frequency. The magnitude squared of S_{11} and S_{21} are plotted in Figure 4.21, clearly showing the behavior of a low-pass filter.

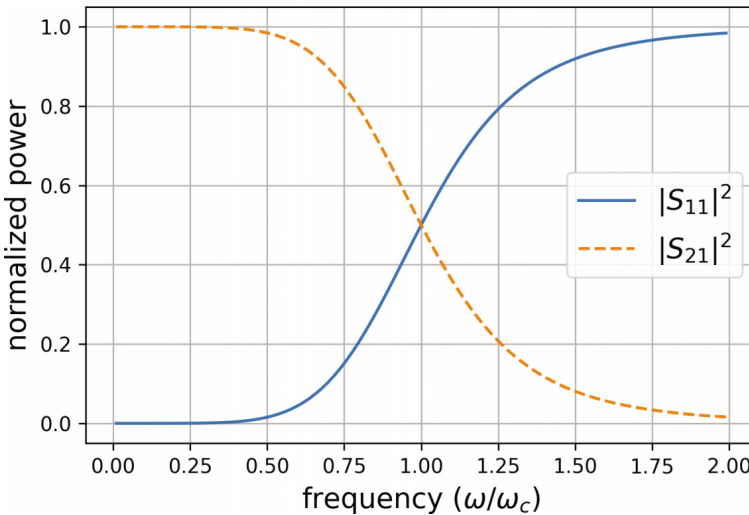


Figure 4.21 Frequency response of the T network low-pass filter.

⁴ Actually, we see that Eq. (4.149) is satisfied at $\omega = 0$ without any requirement on L or C owing to the ω factors on both sides. However this requirement on the ratio of L/C improves the low-frequency match for $\omega \neq 0$.

A more extensive treatment of the design of filters can be found in Pozar [21], but our discussion illustrates the basic principles.

4.10 Noise

Measurements of any kind typically exhibit random errors. We must pay particular attention to these errors when measuring very small quantities, as is the case when measuring the state of a qubit.

4.10.1 Thermal Noise

A fundamental source of error arises from charge vibrations associated with the equilibrium temperature of the physical system. This *thermal noise* or *Johnson noise* is the result of a phenomenon called *black body radiation*. For a resistor at absolute temperature T (K), the power per unit frequency f emitted to the circuit containing the resistor is given by the one-dimensional analogue of the Planck radiation law:

$$P_f = \frac{hf}{e^{hf/kT} - 1}. \quad (4.156)$$

Here h and k are Planck's and Boltzmann's constants, respectively. We can interpret this as the power per unit bandwidth⁵ hf times the probability that the state hf is occupied. The latter is given by the Bose-Einstein distribution function, and is discussed in more detail in Section 7.1. If the resistor is connected to a load through a lossless band pass filter as illustrated in Figure 4.22, the total power injected into the circuit is just $P_f B$, where B is the bandwidth of the filter (Hz). Maximum power will be transferred to

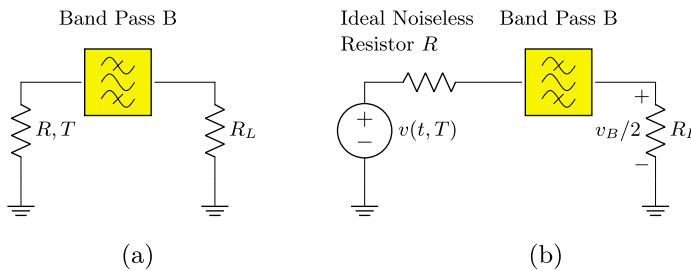


Figure 4.22 Circuit illustrating thermal noise power from a resistor at temperature T . (a) Resistor at temperature T coupled to a load through a lossless bandpass filter with bandwidth B . (b) Equivalent circuit explicitly showing a noise voltage source that depends on temperature.

⁵ Note that power per unit bandwidth has units $(\text{J/s})/(\text{1/s}) = \text{J}$ which is the same as energy. Consequently, depending on the context, hf can be considered either as an energy or power per unit bandwidth.

the load if $Z_L = R$, in which case the load is matched to the source.⁶ The rms voltage dropped across a matched load will be $v_B/2$, if v_B is the band-pass-filtered voltage of an equivalent noise voltage source v (i.e., half of the voltage is dropped across each resistor). This voltage can be related to the power delivered to the load by setting the standard expression for rms power in a resistor equal to the power $P_f B$:

$$\frac{(v_B/2)^2}{R} = \frac{hfB}{e^{hf/kT} - 1}, \quad (4.157)$$

or

$$v_B = \sqrt{\frac{4hfBR}{e^{hf/kT} - 1}}. \quad (4.158)$$

Typically in microwave circuits the frequencies and temperatures are such that $hf \ll kT$. For example, for a 5 GHz signal at room temperature (~ 300 K), $hf/(kT) \approx 8 \times 10^{-4}$. In this case, $\exp(hf/kT) \approx 1 + hf/kT$, and (4.158) can be approximated by

$$v_B \approx \sqrt{4kTBR}. \quad (4.159)$$

In this limit the thermal power delivered to the load $R_L = R$ is

$$P = \frac{(v_B/2)^2}{R} = kTB. \quad (4.160)$$

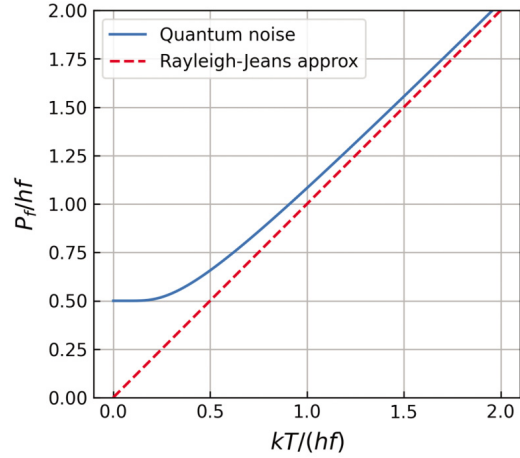
This is called the *Rayleigh-Jeans* approximation. Since the power does not depend on frequency, thermal noise is *white noise* in this limit.

To illustrate the concept we have focused on the power supplied by resistor R to resistor R_L . However, it is clear from the symmetry of the circuit that noise energy also flows in the other direction—from R_L to R . If both resistors have the same temperature, the noise power supplied by each resistor will be equal and the system will be in thermal equilibrium. However, suppose R is associated with a circuit at room temperature, and R_L is associated with the quantum processor at a temperature of a few milli-Kelvins. In this case significant net thermal energy would be flowing into the quantum processor. Not only will this require the refrigerator to work hard to keep the processor at the correct temperature, but the noise power would thermally excite the qubit introducing significant errors. The solution to this problem is to provide signals from room temperature electronics to the processor through cooled attenuators. We will return to this topic in a future section.

In conventional circuits, although the energy of a single photon is much less than the thermal energy, the signal will typically have large numbers of photons so that the signal power will be larger than the noise power. However, in the case of a qubit in thermal equilibrium, if $hf \ll kT$ then the $|1\rangle$ state would be constantly excited by thermal energy, and we would not have the ability to control the state of the qubit. Consequently for qubits we would like to have $hf \gg kT$. The temperature equivalent to the photon energy is $T = hf/k$, which for a 5 GHz signal gives $T \approx 0.23$ K. So to satisfy this condition the temperature of the qubit must be much less than 0.23 K. Using

⁶ Of course an equal amount of power will be delivered from the resistor on the right to the resistor on the left by symmetry, if both start at the same temperature. Otherwise, net energy will flow toward the colder resistor.

Figure 4.23 Quantum noise as a function of temperature normalized to the photon energy. Solid line is the exact expression (4.161), while the dashed line is the Rayleigh-Jeans approximation (4.160).



dilution refrigerators, qubit temperatures can be held as low as 0.01 K. In this limit we must add the zero-point power to (4.156):

$$\begin{aligned}
 P_f &= \frac{hf}{e^{hf/kT} - 1} + \frac{hf}{2} \\
 &= \frac{hf}{2} \frac{e^{hf/kT} + 1}{e^{hf/kT} - 1} \\
 &= \frac{hf}{2} \coth \frac{hf}{2kT}.
 \end{aligned} \tag{4.161}$$

The quantum noise power spectral density (4.161) is compared with the Rayleigh-Jeans approximation (4.160) used at higher temperatures in Figure 4.23. For the case of a 5 GHz qubit at 0.01 K, $kT/(hf) \approx 0.04$, so the noise power is dominated by the zero-point fluctuations.

An attenuator is a passive element with $G < 1$, and is of particular interest for quantum computing systems. Referring to Figure 4.24, we see that if $G \ll 1$ then the noise at the output of the component is dominated by the noise added by the component. Since the attenuator is passive, the added noise is simply the thermal noise corresponding to the temperature of the attenuator.

At very low temperatures it is convenient to introduce the added-noise number [31]

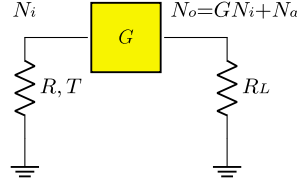
$$n_N = \frac{P_f}{hf}. \tag{4.162}$$

The added-noise number clearly has the quantum limit 1/2, which is sometimes referred to as “one-half photon of noise.”

4.10.2 Equivalent Noise Temperature

It is often the case that the noise introduced by amplifiers and other electronic components is not a strong function of frequency within the system bandwidth. In such cases, we can model the noise with an equivalent temperature; i.e., we can replace the noise

Figure 4.24 Noise added by a circuit with power gain G . For an amplifier, $G > 1$, while for an attenuator $G < 1$.



source with a resistor at temperature T_N and the same noise power would be produced. As an example, consider the circuit shown in Figure 4.24.

The equivalent noise temperature T_N is defined as the increase in input temperature required to account for all the output noise referred to the input:

$$\begin{aligned} GN_i + N_a &= GP_f(T_0 + T_N)B \\ &= \frac{1}{2}GhfB \coth\left(\frac{hf}{2k(T_0 + T_N)}\right). \end{aligned} \quad (4.163)$$

If the noise at the input is the result of a source at the reference temperature T_0 , then

$$N_a + \frac{1}{2}GhfB \coth\left(\frac{hf}{2kT_0}\right) = \frac{1}{2}GhfB \coth\left(\frac{hf}{2k(T_0 + T_N)}\right). \quad (4.164)$$

The IEEE has established a standard [32] for the measurement of noise based on a specific reference temperature.⁷ Assuming we use the standard reference temperature of $T_0 = 290$ K for our input noise source, then $kT_0 \gg hf$ for microwave frequencies, and we can expand the $\coth(x)$ function for small x : $\coth(x) \approx 1 + 1/x$ for $x \ll 1$. Making this approximation in (4.164) and simplifying leads to

$$T_N = \frac{N_a}{GkB}. \quad (4.165)$$

The minimum amount of added noise is the equivalent of $1/2$ photon at the input, or $N_a \geq GhfB/2$. Consequently the quantum limit on the noise temperature of a device is

$$T_N \geq \frac{GhfB/2}{GkB} = \frac{hf}{2k}. \quad (4.166)$$

4.10.3 Noise Factor and Noise Figure

The noise factor F is an engineering metric that is often used to characterize the amount of noise added by an RF component. The concept was introduced by Friis [33] who defined the noise factor as

$$F \equiv \frac{S_i/N_i}{S_o/N_o} \geq 1, \quad (4.167)$$

where S_i/N_i is the signal power to noise power ratio at the input of the component, and S_o/N_o is the signal to noise power ratio at the output of the component. For a

⁷ Clearly there is a computational simplification from choosing a relatively large value for T_0 since this enables the expansion of the $\coth$ function. But why not choose something closer to room temperature than 290 K (63 F)? The value of 290 K was proposed by Friis [33] because it made the value of kT a convenient round number: $kT_0 = 4.00 \times 10^{-21}$ J. This was particularly attractive in an era of hand and slide-rule computation [34].

component with gain G as shown in Figure 4.24, the noise factor can be written

$$\begin{aligned}
 F &= \frac{S_i}{N_i} \frac{GN_i + N_a}{GS_i} \\
 &= 1 + \frac{N_a}{GN_i} \\
 &= \frac{P_f(T_0 + T_N)}{P_f(T_0)} \\
 &\approx 1 + \frac{T_N}{T_0}.
 \end{aligned} \tag{4.168}$$

In the last step we have assumed use of the standard noise temperature of 290 K. If the amount of noise added is negligible compared to the noise at the input to the component, then $F \approx 1$. The noise temperature can also be expressed in terms of the noise factor by solving (4.168) for T_N :

$$T_N = T_0(F - 1). \tag{4.169}$$

The *noise figure* NF is simply the noise factor expressed in dB:

$$NF = 10 \log_{10} F. \tag{4.170}$$

4.10.4 Attenuators and Noise

Let us consider the noise contributed to a circuit by a passive reciprocal element such as an attenuator. Power that is incident on port 2 is split three ways: some of it may be reflected if there is an impedance mismatch at port 2, some of it will be transmitted, and some of it will be dissipated as heat in the element. In Figure 4.25 the power dissipated is visualized as being transmitted to a fictitious third port that is perfectly matched to a thermal reservoir at temperature T . Power conservation allows us to write

$$P_2 = P_r + P_1 + P_d. \tag{4.171}$$

Dividing through by P_2 enables us to write this in terms of S parameters:

$$\begin{aligned}
 1 &= \frac{P_r}{P_2} + \frac{P_1}{P_2} + \frac{P_d}{P_2} \\
 &= |S_{22}|^2 + |S_{12}|^2 + |S_{32}|^2.
 \end{aligned} \tag{4.172}$$

Consequently the magnitude squared of the S parameter for the fictitious port 3 can be obtained in terms of the actual circuit parameters S_{22} and S_{12} :

$$|S_{32}|^2 = 1 - |S_{22}|^2 - |S_{12}|^2. \tag{4.173}$$

However, thermal noise associated with temperature T is also injected into the element via port 3, and the portion of this power appearing at port 2 is simply

$$P_{n2} = P_T |S_{23}|^2. \tag{4.174}$$

Since the element is reciprocal, $|S_{32}| = |S_{23}|$ and $|S_{12}| = |S_{21}|$, and we can use (4.173)

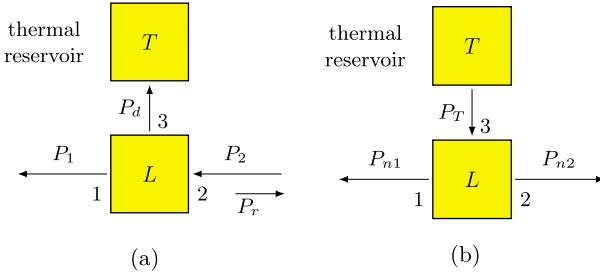


Figure 4.25 Thermal noise from a passive element such as an attenuator at temperature T . (a) Power applied to port 2 is partially reflected and partially transmitted to the output, with the balance dissipated as heat. The power dissipated is visualized as being conveyed to a fictitious port 3 connected to a thermal reservoir at temperature T . (b) The total power out has a contribution from the input as well as the thermal reservoir attached to the fictitious port 3.

to write

$$P_{n2} = P_T (1 - |S_{22}|^2 - |S_{21}|^2). \quad (4.175)$$

If power is also applied to port 1, then the total power output at port 2 is

$$P_o = P_1 |S_{21}|^2 + P_T (1 - |S_{22}|^2 - |S_{21}|^2). \quad (4.176)$$

If the input power is the noise power from a previous stage with effective temperature T_i , then we can write $P_1 = N_i$ and $P_T = N_T$, and the total noise at the output of the attenuator is

$$N_{nT} = N_i |S_{21}|^2 + N_T (1 - |S_{22}|^2 - |S_{21}|^2). \quad (4.177)$$

Usually both circuit ports of the attenuator are matched so that $S_{22} = 0$ and this simplifies to

$$N_{nT} = GN_i + (1 - G)N_T, \quad (4.178)$$

where we have identified $|S_{21}|^2$ with the power gain G of the attenuator.

For a low-loss component, e.g., a low-loss transmission line, $G \approx 1$ and the noise power at the output is essentially the same as the noise power at the input. In contrast, if the attenuation is high so that $G \ll 1$, then the noise output power is just the thermal noise associated with the temperature T .

This has significance for providing control signals to a cryogenic qubit, in that a low-loss transmission line from the room would convey thermal noise power much greater than the energy levels of the qubit, making it impossible to control the energy state. In contrast, if the signal goes through a series of attenuators, each at a colder stage of the refrigerator, then the noise from the control signal will be dominated by the temperature of the last attenuator. If this last attenuator can be held at a comparable temperature to the qubit, then the thermal excitation of the qubit will be minimized. Of course the control signal is also attenuated, so the input signal power to the attenuator chain must be large enough to ensure sufficient amplitude to control the qubit.

Finally, let us consider the noise factor for the attenuator:

$$\begin{aligned} F &= \frac{S/N_i}{GS/(GN_i + (1-G)N_T)} \\ &= 1 + (L-1)\frac{N_T}{N_i}, \end{aligned} \quad (4.179)$$

where we have introduced the loss factor $L = 1/G$ of the attenuator. Note that if the input noise temperature is the same as that of the attenuator, then $F = L$. In case of the IEEE noise factor definition, we have

$$F = 1 + (L-1)\frac{T}{T_0}, \quad (4.180)$$

where $T_0 = 290$ K. Comparing with (4.168) we see that the equivalent noise temperature of an attenuator is

$$T_N = (L-1)T. \quad (4.181)$$

For an attenuator with $L \gg 1$, it may seem odd that the effective noise temperature is much higher than the actual temperature of the attenuator. The reason for this is simply that T_N is defined referred to the input of the device, and a very large temperature at the input is needed to produce the observed noise level at the output of the attenuator!

4.10.5 Noise in Cascaded Systems

Often in microwave systems the signal must pass through several components in sequence, for example if a single component does not provide enough gain or attenuation for the application. Consider the noise in the cascaded system shown in Figure 4.26. The noise power at the output of each component is

$$N_1 = G_1 N_i + N_{a1}, \quad (4.182)$$

$$N_2 = G_1 G_2 N_i + G_2 N_{a1} + N_{a2}, \quad (4.183)$$

$$N_o = G_1 G_2 G_3 N_i + G_2 G_3 N_{a1} + G_3 N_{a2} + N_{a3}. \quad (4.184)$$

If a signal S_i is applied to the input, the noise figure will be

$$\begin{aligned} F &= \frac{S_i}{N_i} \frac{G_1 G_2 G_3 N_i + G_2 G_3 N_{a1} + G_3 N_{a2} + N_{a3}}{G_1 G_2 G_3 S_i} \\ &= 1 + \frac{N_{a1}}{G_1 N_i} + \frac{N_{a2}}{G_1 G_2 N_i} + \frac{N_{a3}}{G_1 G_2 G_3 N_i}. \end{aligned} \quad (4.185)$$

This is more commonly written in the form

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2}. \quad (4.186)$$

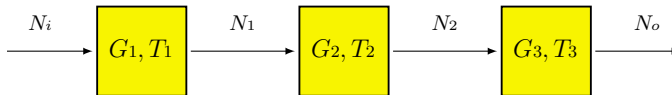


Figure 4.26 Noise in a system of cascaded components.

Note that if $G_1, G_2, G_3 \gg 1$, the noise added by component 1 dominates the added noise in the output, and we have

$$F \approx 1 + \frac{N_{a1}}{G_1 N_i} = F_1. \quad (4.187)$$

In contrast, for attenuators with $G_1, G_2, G_3 \ll 1$, the noise from component 3 dominates the added noise at the output. In this case, the output noise power is primarily determined by the physical temperature of the third attenuator:

$$N_o \approx P_f(T_3)B = \frac{1}{2} h f B \coth \frac{h f}{2 k T_3}. \quad (4.188)$$

We therefore arrive at two key observations: *for a chain of amplifiers, the noise figure of the first amplifier dominates, so it is critical that this first amplifier have the lowest possible noise; but for a chain of attenuators, the noise from the last attenuator dominates, so it is critical for this attenuator to be as cold as possible.*

4.11 Low Noise Amplifiers

In conventional transistor amplifiers, noise arises from resistance in the device. This resistance is caused by scattering of the charge carriers by either lattice vibrations (i.e., *phonons*; see Chapter 7) or departures from the perfect periodicity of the ions in the material from defects in crystal structure or impurities. Scattering from lattice vibrations can be minimized by cooling the device, and material growth techniques can significantly reduce imperfections in the crystal structure. However, charge carriers essential for device operation are provided by dopant impurities, and it is more difficult to reduce this source of resistance. While there are approaches to designing the circuit to minimize the impact of device noise (see for example, Chapter 13 in Lee [34]), it is clearly advantageous to start with a device with as low resistance as possible.

A *high electron mobility transistor*, or HEMT, is a structure that minimizes scattering from impurities by spatially separating the donors supplying the charge carriers, and the channel in which the charge carriers move. This is schematically represented in Figure 4.27. The changes in energy band structure between the layers is such that the donor electrons have the lowest energy if they are in the GaAs layer but very near the interface between the intrinsic layers of AlGaAs and GaAs. The spacer layer minimizes the influence of the donor atoms, and scattering is extremely low in the near-perfect intrinsic semiconductor layers.

The cooling capacity at the lowest-temperature stage in a dilution refrigerator is very limited, and although the losses in HEMT amplifiers are small, they consume too much power to be kept at the lowest temperature stage. However, they can be cooled at the 4K stage, and typically exhibit noise numbers $n_N^{\text{HEMT}} \gtrsim 20$ [35].

Amplifiers that can be supported at the lowest temperature stage are *parametric amplifiers*. Very low power parametric amplifiers can be constructed using superconducting Josephson junctions. We will discuss these more in Chapter 8, but these circuits amplify by periodically varying the inductance in a resonant circuit at, for example, twice the signal frequency. Because the superconducting devices have extremely low loss and operate at the lowest temperature in the refrigerator, noise numbers of order 1–4 can be achieved [35].

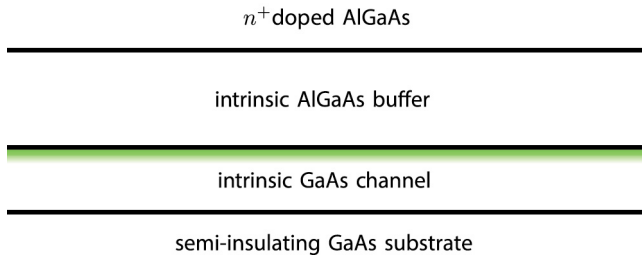


Figure 4.27 Layered structure of different semiconductor materials used to separate the donor impurities from the donor electrons to achieve minimal scattering. Structures of different materials like this are referred to as *heterostructures*. Owing to the different band structure of AlGaAs and GaAs, donor electrons from the heavily-doped AlGaAs layer become trapped on the GaAs side of the interface between intrinsic GaAs and AlGaAs, forming a “2D electron gas.” This concentration of electrons is depicted by the shaded region. A heterostructure similar to this is a key feature of HEMTs.

4.12 Exercises

- 4.1 RG-58 coaxial cable has a characteristic impedance of $50\ \Omega$ and a capacitance per unit length of about $100\ \text{pF/m}$. Find the inductance per unit length.
- 4.2 The return loss from a terminated lossless transmission line is measured to be $15\ \text{dB}$. What percentage of the incident power is absorbed by the load?
- 4.3 What is the characteristic impedance of a quarter-wave matching section designed to match a $300\ \Omega$ transmission line to a $75\ \Omega$ load?
- 4.4 Referring to Figure 4.11, find the values of R_1 and R_2 needed to make a $20\ \text{dB}$ attenuator for a $50\ \Omega$ transmission line.
- 4.5 For the circuit of Figure 4.20(b), find L and C to obtain a cutoff frequency of $100\ \text{MHz}$ when used in a $50\ \Omega$ system.
- 4.6 Calculate the minimum quantum limited noise temperature from (4.166) for an amplifier at $6\ \text{GHz}$.
- 4.7 The noise temperature of a certain HEMT amplifier is $T_N = 4\ \text{K}$. Calculate the noise figure.

5

Resonators: Classical Treatment

Resonators play an important role in superconducting quantum computers. The qubits themselves are nonlinear resonators, and in many architectures, the qubits are coupled to other qubits and to external circuits through transmission line segments functioning as resonators. In quantum computers these resonators are excited with small numbers of photons, and so must ultimately be analyzed quantum mechanically. However, it is helpful to gain intuition about the behavior of both lumped circuits and transmission line resonators using classical analysis. Consequently we will focus on a classical treatment in this chapter, and then consider a quantum mechanical treatment in Chapter 6.

5.1 Parallel Lumped Element Resonator

Consider the lumped-element circuit shown in Figure 5.1(a) consisting of a capacitor, inductor, and resistor in parallel.

The impedance Z_{in} of the circuit is given by:

$$Z_{\text{in}} = \left(\frac{1}{j\omega L} + \frac{1}{R} + j\omega C \right)^{-1}. \quad (5.1)$$

For large R , the impedance will be sharply peaked around $\omega_0 = 1/\sqrt{LC}$. Consequently, let us expand the impedance about ω_0 , i.e., $\omega = \omega_0 + \Delta\omega$:

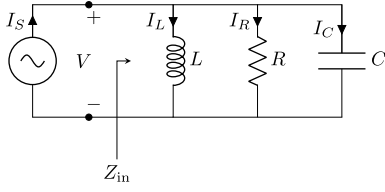
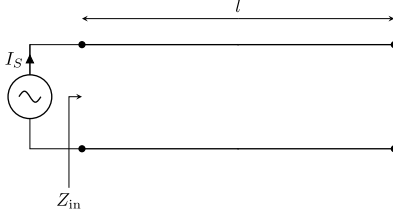
$$Z_{\text{in}} = \left(\frac{1}{jL(\omega_0 + \Delta\omega)} + \frac{1}{R} + jC(\omega_0 + \Delta\omega) \right)^{-1}. \quad (5.2)$$

The first term can be re-written using the series expansion for small x

$$\frac{1}{1+x} \approx 1-x, \quad (5.3)$$

where $x = \Delta\omega/\omega_0$:

$$Z_{\text{in}} \approx \left(\frac{1}{j\omega_0 L} \left(1 - \frac{\Delta\omega}{\omega_0} \right) + \frac{1}{R} + j\omega_0 C + j\Delta\omega C \right)^{-1}. \quad (5.4)$$

(a) *RLC* Lumped element resonator.

(b) Transmission line resonator.

Figure 5.1 Resonator circuits.

This expression can be further simplified by recognizing that the inductive and capacitive reactances are equal and opposite at resonance so that $j\omega_0 C = -1/(j\omega_0 L)$, and also recognizing that $\omega_0^2 L = 1/C$:

$$Z_{in} \approx \left(-\frac{1}{j\omega_0^2 L} \Delta\omega + \frac{1}{R} + j\Delta\omega C \right)^{-1} \quad (5.5)$$

$$\approx \frac{R}{1 + j2RC\Delta\omega}.$$

In the following it is useful to recall that in the sinusoidal steady state we have $I_S = I_0 \exp(j\omega t)$ and $V = V_0 \exp(j\omega t)$. The quality factor (Q) of the resonant circuit is defined as

$$Q \equiv \frac{\omega_0 U}{P_d}, \quad (5.6)$$

where U is the time-averaged energy stored in the circuit and P_d is time average power dissipated. The average energy stored in the capacitor is

$$U_C = \frac{1}{4} C |V_0|^2. \quad (5.7)$$

At resonance, the average energy stored in the inductor is equal to that stored in the capacitor, so the total average stored energy is $U = C|V_0|^2/2$. The average power dissipated is

$$P_d = \frac{|V_0|^2}{2R}. \quad (5.8)$$

Combining this result with the expression for the total energy gives

$$Q = \frac{\omega_0 C |V_0|^2 / 2}{|V_0|^2 / (2R)} = \omega_0 C R = \frac{R}{X_C}, \quad (5.9)$$

where $X_C = 1/(\omega_0 C)$. Since the magnitudes of the inductive and capacitive reactances are equal at resonance, we can write

$$Q = \frac{R}{X_C} = \omega_0 RC = \frac{R}{X_L} = \frac{R}{\omega_0 L}. \quad (5.10)$$

Using these results, we can also write the impedance in terms of the Q :

$$Z_{in} = \frac{R}{1 + j2Q\Delta\omega/\omega_0}. \quad (5.11)$$

In terms of the source current, the power dissipated as a function of frequency is

$$\begin{aligned} P_d &= \frac{1}{2} \text{Re}(I_S V^*) \\ &= \frac{1}{2} \text{Re}(I_S I_S^* Z_{in}^*) \\ &= \frac{|I_0|^2}{2} \text{Re}(Z_{in}^*). \end{aligned} \quad (5.12)$$

To find the real part of the impedance, we multiply and divide by the complex conjugate of the denominator:

$$\begin{aligned} Z_{in} &= \frac{R}{1 + j2Q\Delta\omega/\omega_0} \left(\frac{1 - j2Q\Delta\omega/\omega_0}{1 - j2Q\Delta\omega/\omega_0} \right) \\ &= \frac{R(1 - j2Q\Delta\omega/\omega_0)}{1 + (2Q\Delta\omega/\omega_0)^2}. \end{aligned} \quad (5.13)$$

Taking the real part and substituting into the expression for power dissipated (5.12) gives

$$P_d = \frac{|I_0|^2}{2} \frac{R}{1 + (2Q\Delta\omega/\omega_0)^2}. \quad (5.14)$$

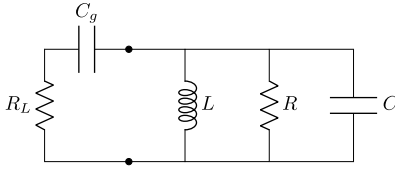
From this equation, we note that the magnitude of the dissipated power will drop to half its peak value if

$$Q = \frac{\omega_0}{2\Delta\omega} \equiv \frac{\omega_0}{BW}, \quad (5.15)$$

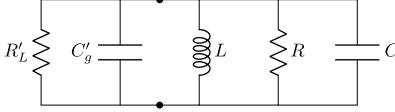
where the bandwidth $BW \equiv 2\Delta\omega$. Said differently, the Q is given by the center frequency divided by the full-width at half maximum of the power dissipated curve.

5.2 Capacitive Coupling to a Parallel Lumped-Element Resonator

Ultimately, to excite or measure the properties of a lumped-element resonator, it must be connected to an external circuit. One way to do this is by way of a coupling capacitor as shown in Figure 5.2(a). The load resistor R_L is the Thevenin equivalent impedance of the circuit to which the resonator is coupled, e.g., a source or transmission line with a $50 \, \Omega$ impedance. Alternatively, near the resonance frequency we can represent the series impedance of the coupling capacitor and external load resistance with an equivalent parallel combination as shown in Figure 5.2(b). To see this, we write the input admittance of the series circuit and isolate the real and imaginary parts:



(a) Capacitively-coupled RLC lumped element resonator.



(b) RLC resonator with Norton equivalent of coupling impedance.

Figure 5.2 Equivalent circuits for capacitively-coupled lumped-element resonator.

$$\begin{aligned}
 Y &= \left(R_L + \frac{1}{j\omega_0 C_g} \right)^{-1} \\
 &= \frac{j\omega_0 C_g}{1 + j\omega_0 R_L C_g} \frac{1 - j\omega_0 R_L C_g}{1 - j\omega_0 R_L C_g} \\
 &= \frac{\omega_0^2 R_L C_g^2}{1 + \omega_0^2 R_L^2 C_g^2} + j\omega_0 \frac{C_g^2}{1 + \omega_0^2 R_L^2 C_g^2} \\
 &\equiv \frac{1}{R'_L} + j\omega_0 C'_g,
 \end{aligned} \tag{5.16}$$

where we have defined the equivalent parallel, or Norton, resistance and capacitance values

$$R'_L = \frac{1 + \omega_0^2 R_L^2 C_g^2}{\omega_0^2 R_L C_g^2}, \quad C'_g = \frac{C_g}{1 + \omega_0^2 R_L^2 C_g^2}. \tag{5.17}$$

If the coupling capacitor is small such that $\omega_0^2 R_L^2 C_g^2 \ll 1$, then we have approximately

$$R'_L \approx \frac{R_L}{\omega_0^2 R_L^2 C_g^2}, \quad C'_g \approx C_g, \tag{5.18}$$

where $R'_L \gg R_L$. The Q of the coupled resonator can be written

$$\begin{aligned}
 \frac{1}{Q} &= \frac{1}{\omega_0 C_{eq} R_{eq}} \\
 &= \frac{1}{\omega_0 C_{eq}} \left(\frac{1}{R} + \frac{1}{R'_L} \right) \\
 &= \frac{1}{\omega_0 C_{eq} R} + \frac{1}{\omega_0 C_{eq} R'_L} \\
 &\approx \frac{1}{Q_{\text{int}}} + \frac{1}{Q_{\text{ext}}},
 \end{aligned} \tag{5.19}$$

where we have made the approximation $C_{eq} = C + C'_g \approx C$ in the last line. Here Q_{int} is the intrinsic Q of the isolated resonator, and Q_{ext} is the external Q , or Q associated with

the loading of the external circuit. The ratio of internal to external Q is referred to as the coupling constant g :

$$g = \frac{Q_{\text{int}}}{Q_{\text{ext}}} = \frac{R}{R'_L}. \quad (5.20)$$

If $g > 1$ the resonator is said to be *over-coupled*, for $g = 1$ the resonator is said to be *critically-coupled*, and for $g < 1$ the resonator is said to be *under-coupled*.

Note that the effective resonant frequency of the loaded circuit is also shifted slightly:

$$\omega'_0 = \frac{1}{\sqrt{LC_{\text{eq}}}} \approx \omega_0 \left(1 - \frac{C_g}{2C} \right). \quad (5.21)$$

5.3 Transmission Line Resonator

Next, let us consider the transmission line segment of length l shown in Figure 5.1(b). Since the segment is open circuited at the right end, the input impedance is given by Eq. (4.65):

$$\begin{aligned} Z_{\text{in}} &= -jZ_c \cot \beta l \\ &= Z_c \coth(j\beta l). \end{aligned} \quad (5.22)$$

In the second line we have re-written this in terms of the hyperbolic cotangent to make it a bit easier to write the corresponding expression for a transmission line with loss. In particular, we simply need to replace $j\beta$ with $\gamma = \alpha + j\beta$:

$$\begin{aligned} Z_{\text{in}} &= Z_c \coth(\alpha + j\beta)l \\ &= Z_c \frac{1 + j \tanh \alpha l \tan \beta l}{\tanh \alpha l + j \tan \beta l}. \end{aligned} \quad (5.23)$$

Here we have used a trigonometric identity to write the impedance using only functions with real arguments. We next simplify this expression by assuming that the transmission line is low-loss, and that the length of the line is very close to an integer multiple of half a wavelength. For a low-loss line, $\alpha l \ll 1$ so that

$$\tanh \alpha l \approx \alpha l. \quad (5.24)$$

If the line is close to an integer multiple of half a wavelength, we can write

$$\beta l = \beta_n l + \Delta \beta l, \quad l = \frac{n\lambda_n}{2}, \quad (5.25)$$

where $\beta_n = 2\pi/\lambda_n$ is the wave number at which the line is a multiple of exactly one half wavelength. It immediately follows that

$$\beta_n l = \frac{2\pi}{\lambda_n} \frac{n\lambda_n}{2} = n\pi. \quad (5.26)$$

We will see that the transmission line segment behaves like a resonator, with this as the resonance condition. The frequencies of these resonances are given by

$$\omega_n = u\beta_n = \frac{1}{\sqrt{L'C'}} \frac{n\pi}{l}, \quad (5.27)$$

from which we can obtain the resonant length l

$$l = \frac{un\pi}{\omega_n}, \quad (5.28)$$

where u is the phase velocity on the transmission line. Using this expression for l we can write

$$\Delta\beta l = \frac{\Delta\omega}{u} \frac{un\pi}{\omega_n} = n\pi \frac{\Delta\omega}{\omega_n}. \quad (5.29)$$

Putting these observations together, for frequencies near one of the resonances we have

$$\tan \beta l = \tan (n\pi + \Delta\beta l) \approx \Delta\beta l = n\pi \frac{\Delta\omega}{\omega_n}. \quad (5.30)$$

Substituting the expressions for $\tanh \alpha l$ and $\tan \beta l$ into the expression for the input impedance (5.23) and keeping only terms that are first-order in small quantities leads to

$$Z_{\text{in}} = \frac{Z_c/(\alpha l)}{1 + j \frac{n\pi}{\alpha l} \frac{\Delta\omega}{\omega_n}}. \quad (5.31)$$

We can cast this into the same form as the impedance for a lumped parallel RLC circuit by defining appropriate equivalent values of R , L , and C . Comparing with (5.5) immediately suggests

$$R_n = \frac{Z_c}{\alpha l} \equiv R. \quad (5.32)$$

With this identification, comparison of the denominators gives

$$2RC_n = n\pi \frac{R_n}{Z_c} \frac{1}{\omega_n}, \quad (5.33)$$

or

$$\begin{aligned} C_n &= \frac{n\pi}{2Z_c} \frac{l\sqrt{L'C'}}{n\pi} \\ &= \frac{l}{2} \sqrt{\frac{C'}{L'}} \sqrt{L'C'} \\ &= \frac{1}{2} C' l \equiv C. \end{aligned} \quad (5.34)$$

Finally, the equivalent inductance can be obtained from the expression for the resonant frequency:

$$\frac{1}{\sqrt{L_n C}} = \frac{1}{\sqrt{L' C'}} \frac{n\pi}{l} = \sqrt{\frac{(n\pi)^2}{L' l C' l}} = \sqrt{\frac{(n\pi)^2}{2L' l C}} \quad (5.35)$$

or

$$L_n = \frac{2L'l}{(n\pi)^2}. \quad (5.36)$$

The Q is given by

$$Q_n = \omega_n RC = \frac{1}{\sqrt{L' C'}} \frac{n\pi}{l} \frac{Z_c}{\alpha l} \frac{C'l}{2} = \frac{n\pi}{2\alpha l}. \quad (5.37)$$

Note that this expression could also be immediately obtained by comparing the denominators of (5.11) and (5.31).

For a given resonator, we see that the Q increases with n . So for applications requiring a large Q (e.g., quantum computing), does this mean that it is better to use higher-order resonances? Closer examination shows that there is a catch: if we want to choose a higher-order resonance but keep the actual frequency the same, then $l = un\pi/\omega_n$. Consequently the increase in the numerator of (5.37) is cancelled by the increase in l in the denominator, so we might as well use the fundamental mode of the resonator ($n = 1$) since it has the smallest physical size. For the fundamental mode resonance with frequency ω_0 , the equivalent lumped L and C values are

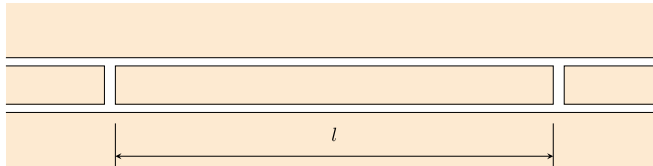
$$C = \frac{\pi}{2} \frac{1}{Z_c \omega_0}, \quad (5.38)$$

$$L = \frac{2}{\pi} \frac{Z_c}{\omega_0}. \quad (5.39)$$

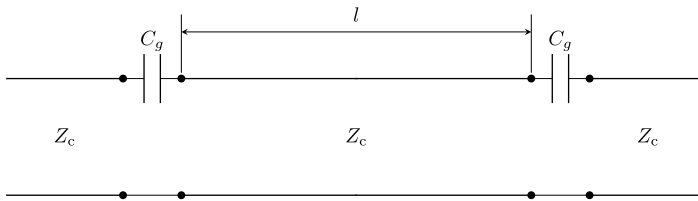
5.4 Capacitive Coupling to a Transmission Line Resonator

One way to create and simultaneously couple to a transmission line resonator is to place small gaps with a spacing of $\lambda/2$ in a transmission line as shown in Figure 5.3(a). The gaps create discontinuities causing reflections and the standing wave characteristic of the resonance, but also provide capacitive coupling to an external circuit.

This circuit can be analyzed using the ABCD matrix approach of Section 4.4. Referring to Figure 5.3(b) and Table 4.1, the ABCD matrix for the capacitively-coupled transmission line resonator is



(a) Top view of capacitively-coupled coplanar transmission line resonator.



(b) Circuit model for transmission line resonator.

Figure 5.3 Capacitively-coupled transmission line resonator.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{j\omega C_g} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cosh \gamma l & Z_c \sinh \gamma l \\ Y_c \sinh \gamma l & \cosh \gamma l \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{j\omega C_g} \\ 0 & 1 \end{bmatrix}. \quad (5.40)$$

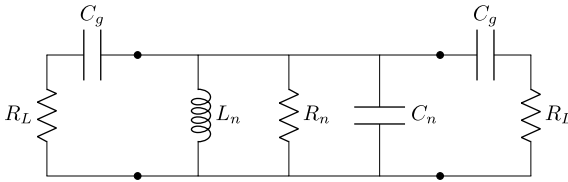
The S parameters can be obtained from the ABCD matrix using Eqs. (4.115), (4.116), and (4.117).

From a system design perspective, it is important to be able to estimate the strength of the transmitted and reflected signals at resonance. To do this, first consider the equivalent circuits near resonance shown in Figure 5.4. As discussed previously in Section 5.3, near resonance the transmission line resonator shown in Figure 5.3(b) can be modeled by the lumped-element circuit shown in Figure 5.4(a). Further, as discussed in Section 5.2, the transmission line loading on both sides can also be represented by Norton equivalents as shown in Figure 5.4(b). Compared to the discussion of Section 5.2, we see that the additional equivalent parallel capacitance from loading is now $2C'_g$, and the equivalent parallel resistive loading is $R'_L/2$. As a result, the coupling coefficient and loaded frequency can be immediately obtained from (5.20) and (5.21):

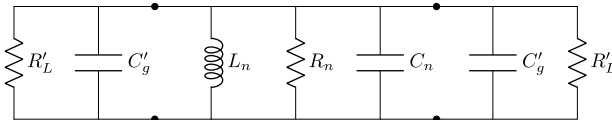
$$g = \frac{Q_{\text{int}}}{Q_{\text{ext}}} = \frac{2R}{R'_L}, \quad (5.41)$$

$$\omega'_0 = \frac{1}{\sqrt{LC_{\text{eq}}}} \approx \omega_0 \left(1 - \frac{C_g}{C} \right). \quad (5.42)$$

At resonance, the reactances cancel, yielding the simplified equivalent circuit shown in Figure 5.5(a). The current I'_L represents the current delivered to the output transmission line or load, and is obtained using the current divider relation:

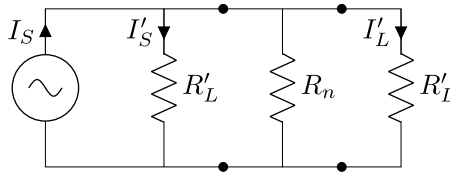


(a) The transmission lines capacitively-coupled to each end of the resonator can be modeled as load resistances $R_L = Z_c$, assuming the transmission lines are lossless and terminated in matched loads.

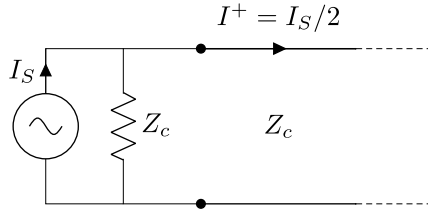


(b) Near the resonant frequency, the series load impedances (Thevenin) presented by the coupled transmission lines can be modeled as effective parallel (Norton) equivalents.

Figure 5.4 Near the n th resonant frequency of the transmission line, the capacitively-coupled transmission line resonator can be modeled as a capacitively-coupled lumped-element resonator.



(a) At resonance, the reactive impedances cancel.



(b) The amplitude of the wave launched by a Norton equivalent source on a matched transmission line is half the source amplitude.

Figure 5.5 Equivalent circuits used to calculate the insertion loss and return loss of a capacitively-coupled transmission line resonator.

$$\begin{aligned}
 I'_L &= I_S \frac{1/R'_L}{\frac{1}{R'_L} + \frac{1}{R_n} + \frac{1}{R'_L}} \\
 &= I_S \frac{R_n/R'_L}{1 + 2R_n/R'_L}.
 \end{aligned} \tag{5.43}$$

To convert this into the parameter S_{21} , we need to divide by the magnitude of the current launched by the source onto the input transmission line. As shown in Figure 5.5(b), the amplitude of the incident current is $I^+ = I_S/2$. Consequently, we have

$$S_{21} = \frac{I'_L}{I^+} = \frac{g}{1 + g}, \tag{5.44}$$

with g given by (5.41). The insertion loss of the resonator in dB is therefore given by

$$IL = -20 \log_{10} \frac{g}{1 + g}. \tag{5.45}$$

We can also use this circuit to estimate the reflection coefficient at resonance. By symmetry, the current I'_S through the source equivalent impedance is the same as I'_L through the load. However, the total current through the source impedance can be understood as the incident current plus the reflected current:

$$I'_S = I^+ (1 - \Gamma). \tag{5.46}$$

Using the fact that $I'_S = I'_L$, we can write

$$\frac{I'_S}{I^+} = 1 - \Gamma = \frac{I'_L}{I^+} = \frac{g}{1 + g}. \tag{5.47}$$

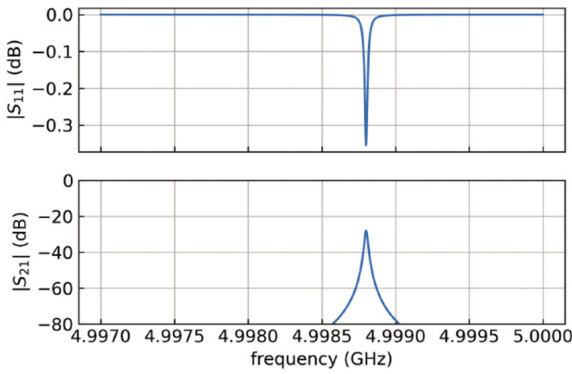
Solving for the reflection coefficient gives

$$\Gamma = S_{11} = \frac{1}{1 + g}. \quad (5.48)$$

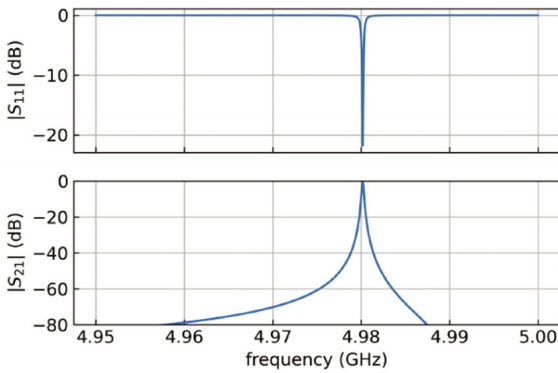
The return loss in dB is therefore

$$RL = -20 \log_{10} \frac{1}{1 + g}. \quad (5.49)$$

The magnitudes of the reflection and transmission S parameters are plotted in Figure 5.6 for under-coupled and over-coupled example cases. As can be seen from these examples, when g is small the frequency shift resulting from coupling is small and the Q remains high, but the reflected and transmitted signals are weak. In contrast, the reflected and transmitted signals are strong for large g , but the frequency shift caused by loading is larger and the Q is reduced.



(a) Under-coupled resonator, $C_g = 0.24$ fF, $g = 0.0416$.



(b) Over-coupled resonator, $C_g = 3.98$ fF, $g = 11.4$.

Figure 5.6 Characteristics of capacitively-coupled transmission line resonator. Note the significantly different frequency scales on the horizontal axis as well as the different vertical scales for $|S_{11}|$. Parameters used are $L_n = 0.453$ μ H/m, $Z_c = R_L = 50$ Ω , $f_0 = 5$ GHz, $Q_{\text{int}} = 2.3 \times 10^5$, and $l = 11.04$ mm. These values are comparable to experimentally-measured parameters. Based on [26].

Typical transmission line resonator circuits for quantum processors are made on high-resistivity silicon substrates, with coplanar resonator structures fabricated in 200 nm thick niobium films [36–38].

5.5 Capacitively-Coupled Lossless Resonators

Circuits important to quantum computing also include qubits coupled to resonators, and qubits coupled to one another. Insight into the behavior of such circuits can be obtained by considering the idealized case of two lossless LC resonators coupled by a capacitor, as shown in Figure 5.7. To analyze this circuit, we write Kirchoff's current law at nodes *A* and *B*:

$$I_{L_A} + I_{C_A} + I_g = 0, \quad (5.50)$$

$$I_{L_B} + I_{C_B} - I_g = 0. \quad (5.51)$$

These equations can be written entirely in terms of the currents in the two inductors using the relations

$$I_{C_A} = C_A \frac{dV_A}{dt}, \quad I_{C_B} = C_B \frac{dV_B}{dt}, \quad (5.52)$$

$$V_A = L_A \frac{dI_{L_A}}{dt}, \quad V_B = L_B \frac{dI_{L_B}}{dt}, \quad (5.53)$$

$$I_g = C_g \frac{d}{dt} (V_A - V_B). \quad (5.54)$$

The result is

$$\frac{d^2 I_{L_A}}{dt^2} + \omega_A^2 I_{L_A} = C_g L_B \omega_A^2 \frac{d^2 I_{L_B}}{dt^2}, \quad (5.55)$$

$$\frac{d^2 I_{L_B}}{dt^2} + \omega_B^2 I_{L_B} = C_g L_A \omega_B^2 \frac{d^2 I_{L_A}}{dt^2}, \quad (5.56)$$

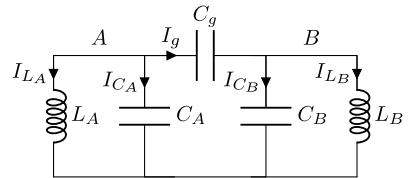
where

$$\omega_A = \frac{1}{\sqrt{L_A(C_A + C_g)}}, \quad (5.57)$$

$$\omega_B = \frac{1}{\sqrt{L_B(C_B + C_g)}}. \quad (5.58)$$

We are particularly interested in the weak coupling limit, when $C_g \ll C_{A,B}$. Substituting Eq. (5.56) into (5.55) and neglecting terms of second-order in the small quantity C_g gives

Figure 5.7 Two LC resonant circuits coupled by a capacitor.



$$\frac{d^2 I_{L_A}}{dt^2} + \omega_A^2 I_{L_A} = C_g L_B \omega_A^2 \omega_B^2 I_{L_B}. \quad (5.59)$$

Similarly, Substituting Eq. (5.55) into (5.56) and again neglecting terms of second-order in the small quantity C_g gives

$$\frac{d^2 I_{L_B}}{dt^2} + \omega_B^2 I_{L_B} = C_g L_A \omega_A^2 \omega_B^2 I_{L_A}. \quad (5.60)$$

For an isolated LC circuit, an initial charge on the capacitor will cause the circuit to oscillate indefinitely at the natural frequency $\omega_0 = 1/\sqrt{LC}$. Consequently, if the two resonant circuits are very weakly coupled, then we expect the behavior to be a small perturbation on the two circuits oscillating at their natural frequencies. In particular, we take the currents in the inductors to be of the form

$$I_{L_A} = \sqrt{\frac{2}{\omega_A L_A}} A(t) e^{j\omega_A t}, \quad (5.61)$$

$$I_{L_B} = \sqrt{\frac{2}{\omega_B L_B}} B(t) e^{j\omega_B t}, \quad (5.62)$$

where $A(t)$ and $B(t)$ are functions that vary slowly compared to the frequencies ω_A, ω_B . The normalization factors are chosen so that the complex power $S = VI^*/2$ in both inductors will have a magnitude of unity if $A = B = 1$.

Using Eq. (5.61), the second time derivative is found to be

$$\begin{aligned} \frac{d^2 I_{L_A}}{dt^2} &= \sqrt{\frac{2}{\omega_A L_A}} \left(\frac{d^2 A}{dt^2} + j2\omega_A \frac{dA}{dt} - \omega_A^2 A \right) e^{j\omega_A t}, \\ &\approx \sqrt{\frac{2}{\omega_A L_A}} \left(j2\omega_A \frac{dA}{dt} - \omega_A^2 A \right) e^{j\omega_A t}. \end{aligned} \quad (5.63)$$

Here we have assumed that A is sufficiently slowly varying on the scale of the period of ω_A that

$$\left| \frac{d^2 A}{dt^2} \right| \ll \left| 2\omega_A \frac{dA}{dt} \right|. \quad (5.64)$$

Substituting (5.63) into (5.59) and simplifying leads to

$$\frac{dA}{dt} = j\kappa B e^{j\Delta t}, \quad (5.65)$$

where

$$\begin{aligned} \kappa &= \frac{1}{2} C_g \omega_A \omega_B \sqrt{\omega_A L_A \omega_B L_B}, \\ &\approx \frac{C_g}{2\sqrt{C_A C_B}} \sqrt{\omega_A \omega_B}, \end{aligned} \quad (5.66)$$

and

$$\Delta = \omega_B - \omega_A. \quad (5.67)$$

Here we have simplified κ by expressing the inductors in terms of the frequencies using (5.57) and (5.58), and neglecting terms that are second order in the small quantity C_g .

Following a similar procedure, the equation for I_{L_B} (5.60) expressed in terms of the amplitudes A and B becomes

$$\frac{dB}{dt} = j\kappa A e^{-j\Delta t}. \quad (5.68)$$

Equations (5.65) and (5.68) are the same coupled-mode equations we found in Section 2.2.3.3. In the present context, the general solutions (2.114) take the form

$$A(t) = e^{j(\Delta/2)t} \left[A(0) \cos(\Omega t/2) + \frac{2\kappa B(0) - jA(0)\Delta}{\Omega} \sin(\Omega t/2) \right], \quad (5.69)$$

$$B(t) = e^{-j(\Delta/2)t} \left[B(0) \cos(\Omega t/2) + \frac{2\kappa A(0) + jB(0)\Delta}{\Omega} \sin(\Omega t/2) \right], \quad (5.70)$$

with

$$\Omega = \sqrt{\Delta^2 + 4\kappa^2}. \quad (5.71)$$

Expressed as real functions of time, the currents are

$$I_{L_A}(t) = \text{Re} \left(\sqrt{\frac{2}{\omega_A L_A}} A(t) e^{j\omega_A t} \right), \quad (5.72)$$

$$I_{L_B}(t) = \text{Re} \left(\sqrt{\frac{2}{\omega_B L_B}} B(t) e^{j\omega_B t} \right). \quad (5.73)$$

As an example, let us consider the case $A(0) = 1$, $B(0) = 0$, and normalize both currents by dividing through by $\sqrt{2/(\omega_A L_A)}$. This gives

$$\hat{I}_{L_A}(t) = \cos(\omega_{\text{ave}} t) \cos(\Omega t/2) + \frac{\Delta}{\Omega} \sin(\omega_{\text{ave}} t) \sin(\Omega t/2), \quad (5.74)$$

$$\hat{I}_{L_B}(t) = \frac{2\kappa}{\Omega} \sqrt{\frac{\omega_A L_A}{\omega_B L_B}} \cos(\omega_{\text{ave}} t) \sin(\Omega t/2). \quad (5.75)$$

Here we have used the observation that $\Delta/2 + \omega_A = -\Delta/2 + \omega_B = (\omega_A + \omega_B)/2 \equiv \omega_{\text{ave}}$. Interestingly, with a bit more manipulation and expressing the time dependence in terms of the normalized time κt , we see that the coupled currents can be written in terms of just two parameters: ω_B/ω_A and $C_g/\sqrt{C_A C_B}$. The first describes the degree of detuning between the two resonators, and the second describes the coupling. The result is

$$\hat{I}_{L_A}(\kappa t) = \cos\left(\frac{\omega_{\text{ave}}}{\kappa} \kappa t\right) \cos\left(\frac{\Omega}{2\kappa} \kappa t\right) + \frac{\Delta}{\Omega} \sin\left(\frac{\omega_{\text{ave}}}{\kappa} \kappa t\right) \sin\left(\frac{\Omega}{2\kappa} \kappa t\right), \quad (5.76)$$

$$\hat{I}_{L_B}(\kappa t) = \frac{2\kappa}{\Omega} \sqrt{\frac{\omega_B}{\omega_A}} \cos\left(\frac{\omega_{\text{ave}}}{\kappa} \kappa t\right) \sin\left(\frac{\Omega}{2\kappa} \kappa t\right), \quad (5.77)$$

where

$$\frac{\omega_{\text{ave}}}{\kappa} = \left(\frac{1 + \omega_B/\omega_A}{\sqrt{\omega_B/\omega_A}} \right) \frac{\sqrt{C_A C_B}}{C_g}, \quad (5.78)$$

$$\frac{\Omega}{\kappa} = \sqrt{(\Delta/\kappa)^2 + 4}, \quad (5.79)$$

$$\frac{\Delta}{\kappa} = 2 \left(\frac{\omega_B/\omega_A - 1}{\sqrt{\omega_B/\omega_A}} \right) \frac{\sqrt{C_A C_B}}{C_g}, \quad (5.80)$$

$$\frac{\Delta}{\Omega} = \frac{1}{\sqrt{1 + 4(\kappa/\Delta)^2}}. \quad (5.81)$$

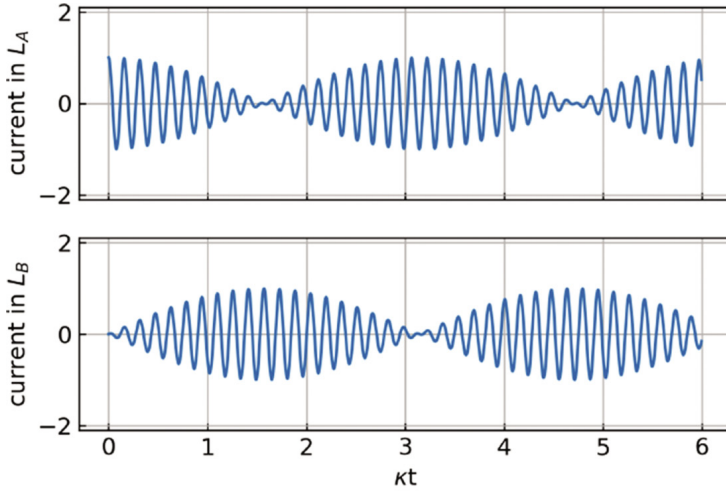
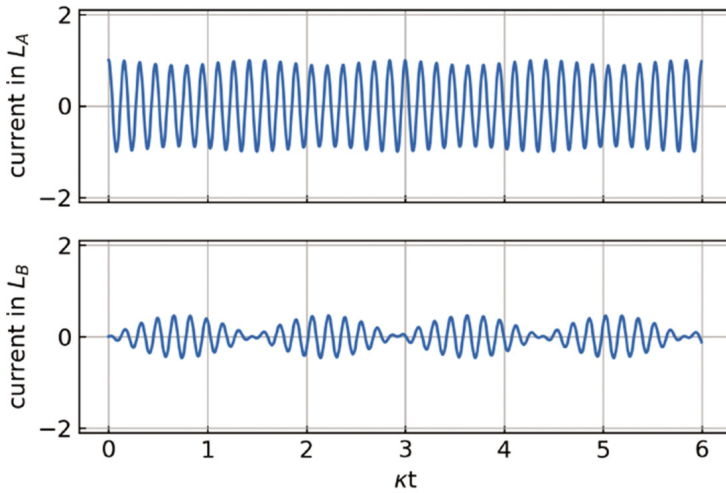
The normalized inductor currents are plotted in Figure 5.8 for the case $C_g/\sqrt{C_A C_B} = 0.05$. When the frequencies of the two circuits are the same, the energy slowly shifts completely from one circuit to the other at a rate determined by $C_g/\sqrt{C_A C_B}$ (Figure 5.8(a)). In contrast, when the frequencies are not the same, there is some periodic energy exchange, but the exchange is never complete. If the detuning is large enough, the energy exchanged can be small enough to be neglected. Consequently, if the frequency of one of the circuits can be controlled, the coupling can be effectively turned on and off. The larger the value of $C_g/\sqrt{C_A C_B}$, the larger the detuning required to turn off the coupling.

Similar behavior can be observed with identical swings on a playground if the swings are not isolated from one another. For example, consider two swings hung from the same horizontal rope instead of a rigid frame. As one swing moves, the support rope is slightly twisted, and this twist affects the motion of the other swing. Imagine pulling back and releasing one swing with the other left at rest. Initially the swing that was pulled back will go back and forth at its natural frequency. However, over time the coupling through the support rope will start the second swing moving, and the maximum excursion of the first swing will gradually get smaller until only the second swing is moving at very nearly the initial maximum excursion of the first swing. The coupling will then start the first swing moving again until all of the motion is once again with the first swing. If the swings are not identical, e.g., if the length of the swing supports or the weight of the seats are different, then the energy will never be completely transferred to the second swing, although the back and forth transfer will still be apparent.

5.6 Classical Model of Qubit Readout

The coupled-resonator model can be used to describe a transmission line resonator coupled to a qubit, provided we only allow the resonator representing the qubit to have two states. As we will see in Chapter 8, this can be realized to a good approximation in practice by using a nonlinear inductor so that the energy separation between higher-order states is different than between the ground state and first excited state. In such a case we can limit our consideration to the two lowest states, provided the excitation signals do not have frequency components corresponding to the higher-level transitions.

The geometry of a transmon qubit coupled to a transmission line resonator is depicted in Figure 5.10(a). The figure shows the top view of the chip, with the insulating substrate shown as white, and the superconducting regions shown as shaded. The capacitor associated with the qubit resonance is formed by the two large parallel

(a) $\omega_A = \omega_B$.(b) $\omega_B/\omega_A = 1.1$.**Figure 5.8** Coupling between lossless LC resonators for $C_g/\sqrt{C_A C_B} = 0.05$.

rectangular conductor patterns in the center of the dielectric region. The two conductors are connected at the center by a Josephson junction (see Chapter 8) which forms a nonlinear inductor. The capacitance in parallel with the inductance forms the resonant circuit for the qubit. Small gaps in the superconducting patterns provide capacitive coupling to input and output transmission lines, as well as ground. On the left side, the transmission line forms a resonator, with the length between the qubit and the gap on the left forming an electrical half wavelength at resonance. Note that Figure 5.10(a)

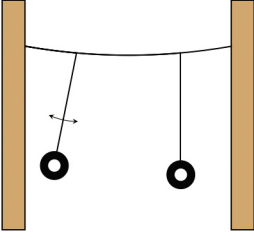
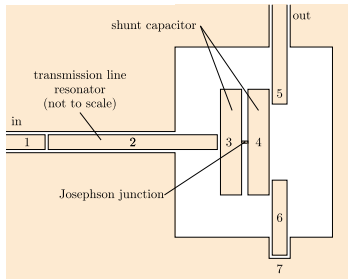


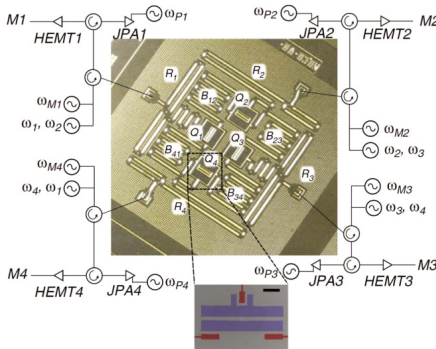
Figure 5.9 Tire swings suspended from a common support rope have similar behavior to coupled LC resonators.

does not show the resonator and qubit at proportional scales, to make it easier to depict the key elements of the geometry. A photograph of an actual four qubit chip is shown in 5.10(b) [37]. The length of the resonator is much longer than the size of the qubit, and the resonator is meandered to reduce the area required on the chip. Additional details regarding qubit fabrication can be found in [36, 38].

As we will see in Section 6.4.9, when a qubit is coupled to a resonator and the frequencies of the qubit ω_q and resonator ω_0 are such that $\omega_0, \omega_q \gg |\omega_0 - \omega_q| \gg \kappa$, the energy does not “slosh” back and forth between the resonator and the qubit as we saw with the coupled resonators of the last section, but instead the frequency of each is shifted by the state of the other. This is a purely quantum phenomenon. In particular,



(a) Top view of capacitively-coupled coplanar transmission line resonator.



(b) Photograph of a four qubit circuit showing the relative sizes of the qubits and transmission line resonators. Note that the transmission line resonators are meandered to reduce their size footprint on the chip. ([37]. Córcoles et al., (2015), Figure 07, p. 6 / Springer Nature / CC BY 4.0.)

Figure 5.10 Geometry of a transmon qubit coupled to a transmission line resonator.

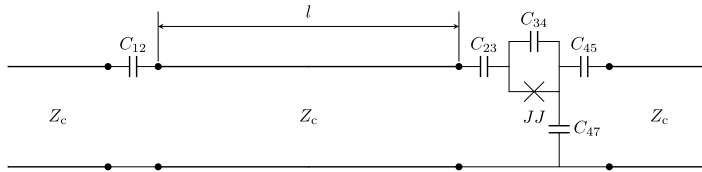
the frequency of the resonator is:

$$\omega_r = \omega_0 \pm \frac{\kappa^2}{|\Delta|}. \quad (5.82)$$

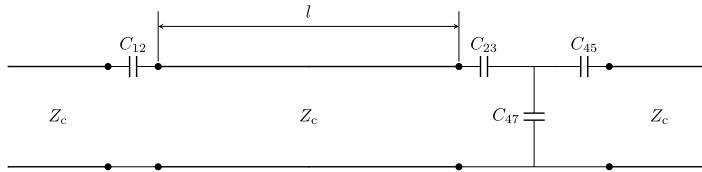
This situation is referred to as the *dispersive limit*. Under this circumstance, a signal near the resonator frequency applied to the left input in Figure 5.10(a) will show features in the reflected power that indicate the frequency shift of the resonator. Since this shift will depend on the qubit state, it provides a practical way to read the state of a qubit.

More details of the measurement configuration are shown in Figure 5.10(b). Referring to the upper left-hand connections as an example, a signal at the frequency of resonator R_1 , ω_{M1} is applied to R_1 . A circulator is used to separate the reflected signal from the input signal. After amplification, the amplitude and phase of the reflected signal are then extracted using a circuit similar to that shown in Figure 4.19. An approximate equivalent circuit of the qubit coupled to the transmission line resonator is shown in Figure 5.11(a). The Josephson junction is shown as the “ $\times$ ” in parallel with the shunt capacitance C_s . At frequencies well above the resonant frequency of the qubit, the impedance of the shunt capacitor will effectively short-out the Josephson junction, leading to the approximate model shown in Figure 5.11(b). This circuit can be analyzed using the ABCD matrix method described in Chapter 4. In particular, the system ABCD matrix is given by

$$M_{\text{sys}} = M_s(C_p)M_t(l)M_s(C'_p)M_p(C''_p)M_s(C'_p) \quad (5.83)$$



(a) Equivalent circuit of the transmission line resonator coupled to a transmon qubit shown in Figure 5.10(a). The capacitance C_{47} is the series combination of C_{46} and C_{67} ; i.e., $1/C_{47} = 1/C_{46} + 1/C_{67}$.



(b) Simplified equivalent circuit in the dispersive limit where the resonator frequency is well above the qubit frequency.

Figure 5.11 Equivalent circuits for transmission line resonator coupled to a transmon qubit.

where

$$M_{\text{sys}} = \begin{bmatrix} A_{\text{sys}} & B_{\text{sys}} \\ C_{\text{sys}} & D_{\text{sys}} \end{bmatrix}, \quad (5.84)$$

$$M_s(C) = \begin{bmatrix} 1 & \frac{1}{j\omega C} \\ 0 & 1 \end{bmatrix}, \quad (5.85)$$

$$M_p(C) = \begin{bmatrix} 1 & 0 \\ j\omega C & 1 \end{bmatrix}, \quad (5.86)$$

$$M_t(l) = \begin{bmatrix} \cosh \gamma l & Z_c \sinh \gamma l \\ Y_c \sinh \gamma l & \cosh \gamma l \end{bmatrix}, \quad (5.87)$$

and $\gamma l = \alpha l + j\beta l$. The reflected signal S_{11} can be obtained from the system ABCD matrix M_{sys} using (4.106).

The slight shift in the resonant frequency described by (5.82) can be modeled by small shifts in the length l . Technically, this would change both αl and βl . For sufficiently small changes we can neglect the effects on amplitude from αl but keep the effects on phase from βl . In this case the product γl can be written (see (5.29) and (5.37))

$$\gamma l = \frac{\pi}{2Q_0} + j\pi \frac{\omega}{\omega_r}, \quad (5.88)$$

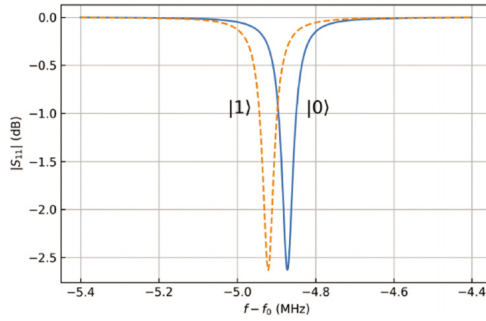
where the fundamental resonance is assumed ($l = \lambda/2$), and Q_0 and ω_r are the intrinsic Q and resonant frequency, respectively. Using (5.82) for ω_r enables us to model the frequency shift caused by the qubit state.

The magnitude of the reflection coefficient for an example calculation is shown in Figure 5.12(a). The shift in frequency of the dip in the reflected signal allows the state of the qubit to be determined. The qubit state can also be read from the phase of the reflected signal. The phase of the reflected signal is shown in Figure 5.12(b), but note that the size of the phase shift is relatively small ($\ll 1$ radian). The reason for this small shift is that the reflected signal is dominated by the simple reflection from an open-circuited transmission line, with a small perturbation caused by coupling to the resonator. Since the reflection coefficient from an open-circuited line is just +1, the reflected component from the resonator can be made more apparent by subtracting 1 from the reflected signal to form $\tilde{S}_{11} = S_{11} - 1$. The phase of the offset quantity $\tilde{S}_{11}$ is shown in Figure 5.12(c).¹ The qubit state can be obtained, for example, by measuring the phase at a single frequency midway between the $|0\rangle$ and $|1\rangle$ peaks.

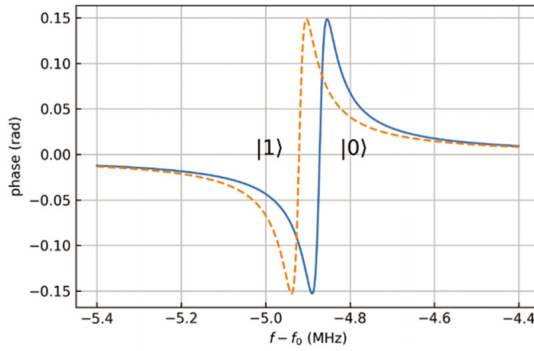
5.7 Exercises

5.1 Consider a lumped-element resonant circuit. The full-width at half maximum of power dissipation versus frequency is measured to be 50 kHz. If the center frequency of the resonance is 5 GHz, find the Q of the resonator.

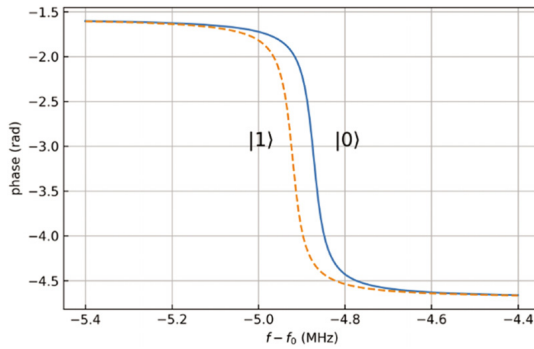
¹ The phase shown has also been “unwrapped” to recover a continuous curve by adding or subtracting an appropriate multiple of 2π when the phase variation exceeds $\pm\pi$.



(a) Magnitude of the reflection coefficient from the transmission line resonator for both qubit states. The frequency of the resonator is a few MHz below the intrinsic resonator frequency f_0 owing to the loading effects of the coupling capacitors.



(b) Phase of the raw reflected signal. The phase shift is relatively small since the effect of the resonator is masked by the large reflection from the nominally open-circuited line.



(c) Phase of the reflected signal after subtracting off the constant reflection coefficient vector and normalizing.

Figure 5.12 Magnitude and phase of the reflected signals from the resonator in the dispersive limit. Parameters used in the calculations are: $Z_c = 50$ Ohms, $f_q = \omega_q/2\pi = 5$ GHz, $f_0 = \omega_0/2\pi = 7$ GHz, $Q_0 = 2.3 \times 10^5$, $C_{12} = 0.5$ fF, $C_{23} = C_{45} = C_{47} = C_{12}/2$, $\kappa = 7$ MHz.

- 5.2** Consider a lumped-element resonator coupled to an external circuit through a capacitor with value C_1 . The measured Q is found to be 5×10^4 . However, measurements have also been made with increasingly smaller coupling capacitors, and it has been determined that as the coupling capacitor gets smaller, the measured Q approaches 2×10^5 . What is Q_{ext} when the resonator is coupled through C_1 ?
- 5.3** A coplanar waveguide transmission line resonator has a fundamental resonance at 6 GHz. If the characteristic impedance of the transmission line is $50 \, \Omega$, what are the equivalent lumped values of inductance and capacitance?
- 5.4** Consider a transmission line resonator capacitively coupled to input and output transmission lines as shown in Figure 5.3. If the resonator is critically coupled, what is the insertion loss in dB?
- 5.5** Consider two identical coupled LC resonators, with $C_A = C_B = 1 \, \text{pF}$, $L_A = L_B = 1 \, \text{nH}$, and $C_g = 50 \, \text{fF}$. Find the frequency Ω characterizing the oscillation of energy back and forth between the resonators.
- 5.6** A qubit with resonant frequency 5 GHz is coupled to a transmission line resonator with resonant frequency 7 GHz. The coupling strength is $\kappa = 7 \, \text{MHz}$. What is the total shift in frequency of the transmission line resonator for the qubit being in the states $|0\rangle$ and $|1\rangle$?

6

Resonators: Quantum Treatment

The basic requirement for a quantum mechanical system to be used as a qubit is for it to have two distinct states that can be unambiguously controlled and measured. In this chapter we will discuss how qubits can be realized from superconducting microwave circuits, drawing on our understanding of resonators from Chapter 5.

However, before we can jump right into understanding superconducting circuits, we need to lay a foundation with some concepts in mathematics and physics.

Several formulations of classical physics have been developed over the past several centuries. Newton's Laws are probably the most widely known, but alternatives of particular interest in making the transition from classical to quantum formulations are *Lagrange's equations* and *Hamilton's equations*. More detailed discussions of these can be found in texts on classical mechanics such as [39]. We begin with Lagrangian Mechanics and Harmonic Oscillators.

6.1 Lagrangian Mechanics

6.1.1 Hamilton's Principle

Of particular interest to us for our discussion of circuits are Lagrange's equations, which are based on *Hamilton's principle*.

According to Hamilton's Principle, the motion of a system from time t_1 to time t_2 is the path along which the quantity

$$I = \int_{t_1}^{t_2} \mathcal{L}(t) dt \quad (6.1)$$

is stationary, where $\mathcal{L}$ is called the *Lagrangian* and is given by the difference between the kinetic and potential energies of the system: $\mathcal{L} = \mathcal{T} - \mathcal{V}$. The quantity I is referred to as the *action*, so Hamilton's Principle is also sometimes referred to as the Principle of Stationary Action and can be compactly expressed

$$\delta \int_{t_1}^{t_2} \mathcal{L}(t) dt = 0. \quad (6.2)$$

Hamilton's Principle is entirely equivalent to Newton's Laws.

To state a bit more precisely what we mean by being "stationary," we need to make a small side trip into the calculus of variations.

6.1.2 Calculus of Variations

Consider the integral over an expression that in general depends on a coordinate x , a function $\phi(x)$, and its derivative $\dot{\phi}(x)$. Specifically, we are interested in the expression

$$I[\phi] = \int_a^b \mathcal{L}(\phi, \dot{\phi}, x) dx. \quad (6.3)$$

Note that I will simply be a number that depends on the choice of ϕ , i.e., it maps a function onto a number. Such an expression is referred to as a *functional*, in contrast with a *function* that maps one number onto another number. The functional I is stationary for a given ϕ if it has the property

$$I[\phi(x) + \xi(x)] = I[\phi(x)] + \text{Order}(\xi^2), \quad (6.4)$$

where ξ represents a small perturbation of the function ϕ . It will be convenient if we write ξ as the product of a small coefficient α and a function $\eta(x)$ that is arbitrary except that its value is zero at the endpoints, i.e., $\eta(a) = \eta(b) = 0$. We can then expand the functional (6.3) as a Taylor series in α :

$$\begin{aligned} I(\alpha) &= \int_a^b \mathcal{L}(\phi + \alpha\eta, \dot{\phi} + \alpha\dot{\eta}, x) dx \\ &= \mathcal{L}(0) + \alpha \int_a^b \left(\frac{\partial \mathcal{L}}{\partial \phi} \eta + \frac{\partial \mathcal{L}}{\partial \dot{\phi}} \dot{\eta} \right) dx + \text{Order}(\alpha^2). \end{aligned} \quad (6.5)$$

This will be stationary according to Eq. (6.4) if the coefficient of α is zero, i.e.,

$$\int_a^b \left(\frac{\partial \mathcal{L}}{\partial \phi} \eta + \frac{\partial \mathcal{L}}{\partial \dot{\phi}} \dot{\eta} \right) dx = 0. \quad (6.6)$$

We can integrate the second term by parts to obtain

$$\int_a^b \left(\frac{\partial \mathcal{L}}{\partial \phi} - \frac{d}{dx} \frac{\partial \mathcal{L}}{\partial \dot{\phi}} \right) \eta dx + \left. \frac{\partial \mathcal{L}}{\partial \dot{\phi}} \eta \right|_a^b = 0. \quad (6.7)$$

The last term vanishes since $\eta(a) = \eta(b) = 0$. If the remaining integral must vanish for an arbitrary η , then we conclude that the coefficient of η must identically vanish over the path from a to b , i.e., we require

$$\frac{\partial \mathcal{L}}{\partial \phi} - \frac{d}{dx} \frac{\partial \mathcal{L}}{\partial \dot{\phi}} = 0. \quad (6.8)$$

This is called the *Euler-Lagrange* equation, and describes the dynamics of the system.

This can be easily extended to functionals that depend on multiple functions. For example, in the two-function case $I[\phi_1, \phi_2]$, Eq. (6.5) becomes

$$\begin{aligned} I(\alpha_1, \alpha_2) &= \mathcal{L}(0) + \alpha_1 \int_a^b \left(\frac{\partial \mathcal{L}}{\partial \phi} \eta_1 + \frac{\partial \mathcal{L}}{\partial \dot{\phi}} \dot{\eta}_1 \right) dx \\ &\quad + \alpha_2 \int_a^b \left(\frac{\partial \mathcal{L}}{\partial \phi} \eta_2 + \frac{\partial \mathcal{L}}{\partial \dot{\phi}} \dot{\eta}_2 \right) dx \\ &\quad + \text{Order}(\alpha_1^2) + \text{Order}(\alpha_2^2). \end{aligned} \quad (6.9)$$

It is clear that the coefficients of both α_1 and α_2 must vanish separately to eliminate first-order dependence on $\alpha_{1,2}$. We conclude that in general

$$\frac{\partial \mathcal{L}}{\partial \phi_i} - \frac{d}{dx} \frac{\partial \mathcal{L}}{\partial \dot{\phi}_i} = 0. \quad (6.10)$$

We recover Hamilton's Principle by simply making the identification $x \rightarrow t$.

More detailed discussions of the calculus of variations can be found in [39] and [40].

6.1.3 Lagrangian Equation of Motion

In the Lagrangian formulation of mechanics, the functions ϕ_i are usually taken to be the coordinates of particles, q_i . Making this identification along with $x \rightarrow t$ gives the common form of the Lagrangian equation of motion

$$\frac{\partial \mathcal{L}}{\partial q_i} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} = 0, \quad i = 1, 2, \dots, N. \quad (6.11)$$

For a particle in the presence of a magnetic field the Lagrangian is given by [39]

$$\mathcal{L} = \frac{1}{2} m \dot{\mathbf{r}}^2 - e\mathcal{V} + e\mathbf{A} \cdot \dot{\mathbf{r}}. \quad (6.12)$$

As an example, let us evaluate the equation of motion for the coordinate x of the particle. First consider the kinetic energy term:

$$KE = \frac{1}{2} m \dot{\mathbf{r}}^2 = \frac{1}{2} m (\dot{x}^2 + \dot{y}^2 + \dot{z}^2). \quad (6.13)$$

When substituted into the equation of motion (6.11) with $q_i = x$ we obtain

$$\frac{\partial(KE)}{\partial x} - \frac{d}{dt} \frac{\partial(KE)}{\partial \dot{x}} = -m\ddot{x}. \quad (6.14)$$

Next, consider the electrostatic potential term:

$$\frac{\partial(-e\mathcal{V})}{\partial x} - \frac{d}{dt} \frac{\partial(-e\mathcal{V})}{\partial \dot{x}} = -e \frac{\partial \mathcal{V}}{\partial x}. \quad (6.15)$$

Finally, consider the last term with the magnetic vector potential $\mathbf{A}$:

$$e\mathbf{A} \cdot \dot{\mathbf{r}} = e(A_x \dot{x} + A_y \dot{y} + A_z \dot{z}). \quad (6.16)$$

The contribution to the equation of motion is

$$\begin{aligned} \frac{\partial(e\mathbf{A} \cdot \dot{\mathbf{r}})}{\partial x} - \frac{d}{dt} \frac{\partial(e\mathbf{A} \cdot \dot{\mathbf{r}})}{\partial \dot{x}} = \\ e \left(\frac{\partial A_x}{\partial x} v_x + \frac{\partial A_y}{\partial x} v_y + \frac{\partial A_z}{\partial x} v_z \right) - e \frac{dA_x}{dt}. \end{aligned} \quad (6.17)$$

The total time derivative of A_x is obtained from the chain rule:

$$\frac{dA_x}{dt} = \frac{\partial A_x}{\partial t} + \left(v_x \frac{\partial A_x}{\partial x} + v_y \frac{\partial A_x}{\partial y} + v_z \frac{\partial A_x}{\partial z} \right). \quad (6.18)$$

Here the first term results from the explicit dependence on time, and the terms in parenthesis are from the motion of the particle, which causes the location at which A_x is

evaluated to change with time. Substituting this expression into (6.17) and simplifying gives

$$\begin{aligned}
 \frac{\partial(e\mathbf{A} \cdot \dot{\mathbf{r}})}{\partial x} - \frac{d}{dt} \frac{\partial(e\mathbf{A} \cdot \dot{\mathbf{r}})}{\partial \dot{x}} &= \\
 e v_y \left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y} \right) + e v_z \left(\frac{\partial A_z}{\partial x} - \frac{\partial A_x}{\partial z} \right) & \\
 = e (v_y (\nabla \times \mathbf{A})_z - v_z (\nabla \times \mathbf{A})_y) & \\
 = e (v_y B_z - v_z B_y) & \\
 = e (\mathbf{v} \times \mathbf{B})_x. &
 \end{aligned} \tag{6.19}$$

Adding all the pieces together gives

$$m\ddot{x} = e \left(-\frac{\partial \mathcal{V}}{\partial x} + (\mathbf{v} \times \mathbf{B})_x \right) \tag{6.20}$$

which we recognize as the x component of the Lorentz force law.

6.2 Hamiltonian Mechanics

In the Lagrangian formalism, a system is described in terms of the variables $(q, \dot{q}, t)$. In the Hamiltonian formalism, the system is described in terms of the variables (q, p, t) , where q, p are referred to as *canonical variables*, and p is called the *conjugate momentum*. The conjugate momentum is obtained from the Lagrangian using

$$p_i = \frac{\partial \mathcal{L}(q_i, \dot{q}_i, t)}{\partial \dot{q}_i}. \tag{6.21}$$

Further, instead of the Lagrangian, the system dynamics are formulated in terms of the new function

$$\mathcal{H}(q, p, t) = \sum_i \dot{q}_i p_i - \mathcal{L}(q, p, t). \tag{6.22}$$

The function $\mathcal{H}$ is referred to as the *Hamiltonian*. The dynamics of the system are then described by Hamilton's equations:

$$\dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \tag{6.23}$$

$$\dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i}. \tag{6.24}$$

For the case of a particle in an electromagnetic field, the conjugate momentum can be found from (6.12) and (6.21):

$$\mathbf{p} = m\mathbf{v} + e\mathbf{A}. \tag{6.25}$$

The Hamiltonian is then constructed from (6.22):

$$\begin{aligned}
 \mathcal{H} &= \dot{\mathbf{r}} \cdot (m\dot{\mathbf{r}} + e\mathbf{A}) - \frac{1}{2}m\dot{\mathbf{r}}^2 + e\mathcal{V} - e\mathbf{A} \cdot \dot{\mathbf{r}} \\
 &= \frac{1}{2}m\dot{\mathbf{r}}^2 + e\mathcal{V}.
 \end{aligned} \tag{6.26}$$

We see that in this case, the Hamiltonian is simply the sum of the kinetic and potential energies.¹ Although not true in general,² the observation that the Hamiltonian is simply the total energy (sum of kinetic and potential energies) is true for all of the systems that we will consider.

6.3 Harmonic Oscillators

The harmonic oscillator plays an important role in both classical and quantum physics. In classical physics, a simple harmonic oscillator is formed by a mass attached to an anchored spring. If the mass is displaced from its equilibrium position, the mass will be pulled back toward the equilibrium by the spring, but the inertia of the moving mass will cause it to move past equilibrium. As soon as it passes the equilibrium position, however, the spring will start pushing it back until the motion stops, and the mass starts back toward the equilibrium position. In the absence of any loss or damping, this back and forth motion, or oscillation, will continue indefinitely. The oscillation is also referred to as harmonic motion. This system is important in its own right (e.g., when used as part of the shock absorbing system of a vehicle), but it is also important because many other physical systems exhibit the same or very similar behavior, and so can be modeled with the harmonic oscillator. An excellent example is an electrical circuit consisting of an inductor in parallel with a capacitor. The equations describing the behavior of the voltage and current are of exactly the same form as the equation of motion for the spring-mass system.

The harmonic oscillator is significant in the quantum theory, as well, since every classical physical system that can be modeled with the harmonic oscillator can also be modeled as a quantum mechanical harmonic oscillator for very small excitations. In particular, although the voltage and current in the LC resonant circuit mentioned above can classically have any value, in the quantum theory they can have only discrete values. This will be important to us in that we will use two of these levels to form a qubit. Consequently we start by considering the classical and quantum harmonic oscillators to form the foundation for our discussion of transmon qubits. More detailed treatments of the harmonic oscillator can be found in [41–43] and in most introductory quantum mechanics texts.

6.3.1 Classical Harmonic Oscillator

Consider the mechanical system shown in Figure 6.1. A spring with force constant K is anchored on the left, and a mass m is attached to the other end of the spring. We will take the position of the mass to be x , with $x = 0$ corresponding to the equilibrium position as shown. We will first analyze this system with Newton's laws, and then with the Lagrangian formulation.

If the mass is displaced from equilibrium by x , the force exerted by the spring is $F = -Kx$, i.e., in a direction trying to return the mass to equilibrium. At the same

¹ Note that the circular motion caused by the magnetic field does not contribute to the kinetic energy of the particle.

² For a discussion of the necessary conditions, see [39] Sections 2–6.

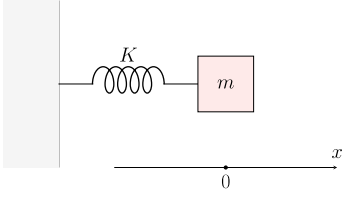


Figure 6.1 Classical harmonic oscillator consisting of a mass m attached to an anchored spring with force constant K .

time, Newton's law $F = dp/dt$ must apply. Setting these two forces equal gives us the equation of motion for the mass:

$$\frac{dp}{dt} = -Kx. \quad (6.27)$$

This can be expressed entirely in terms of x by noting that $p = mv = m dx/dt$, so $dp/dt = m d^2x/dt^2$:

$$m \frac{d^2x}{dt^2} = -Kx. \quad (6.28)$$

It is straightforward to verify that $x(t) = A \cos \omega t$ is a solution to this equation, provided

$$\omega = \omega_0 = \sqrt{\frac{K}{m}}. \quad (6.29)$$

This is the frequency at which the mass will naturally oscillate. Often it is convenient to use complex phasor notation as used in circuit theory. In this case we take $x = A \exp(-i\omega t)$, and the actual function of time is obtained by taking the real part of this expression.

Next, let us consider the system with the Lagrangian formulation. The Lagrangian is given by

$$\mathcal{L} = \frac{1}{2} m \dot{x}^2 - \frac{1}{2} K x^2. \quad (6.30)$$

The Euler-Lagrange equation of motion is then

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}} = 0 \quad (6.31)$$

$$-Kx - m\ddot{x} = 0,$$

which is the same as (6.28).

The canonical momentum is given by

$$p = \frac{\partial \mathcal{L}(x, \dot{x}, t)}{\partial \dot{x}} = m\dot{x}, \quad (6.32)$$

and the Hamiltonian is given by

$$\begin{aligned} \mathcal{H} &= m\dot{x}^2 - \frac{1}{2} m \dot{x}^2 + \frac{1}{2} K x^2 \\ &= \frac{p^2}{2m} + \frac{1}{2} K x^2 \\ &= \frac{p^2}{2m} + \frac{1}{2} m \omega_0^2 x^2. \end{aligned} \quad (6.33)$$

We note that the total energy of the system sloshes back and forth between kinetic (motion of the mass) and potential (displacement of the spring) forms. At any given time, the total energy is given by the sum of the potential and kinetic energies.

6.3.2 Quantum Mechanical Harmonic Oscillator

In Chapter 2 we saw that to convert the classical Hamiltonian representing the Zeeman energy into its quantum counterpart, the classical vector representing the angular momentum had to be replaced by an operator. In the case of spin angular momentum, the corresponding quantum operator is the vector whose x , y , and z components are the corresponding Pauli matrices. We then represented the quantum system with a state vector. This was convenient and sufficient because spin only takes on discrete values.

More generally, a system with continuous variables (such as position) is represented by a *wave function* that is constructed by the product of a continuous function and a vector representing spin (sometimes referred to as a *spinor*). As is the case with spin, classically continuous variables must be replaced with corresponding operators to obtain the quantum Hamiltonian. Some common quantities and their corresponding quantum mechanical operators are shown in Table 6.1.

Note that in classical physics, p and x are simply scalars and so commute: $xp = px$. However, the corresponding quantum mechanical operators do not commute:

$$\begin{aligned} px |\psi\rangle &= \frac{\hbar}{i} \frac{\partial}{\partial x} (x |\psi\rangle) \\ &= \left(\frac{\hbar}{i} + x \frac{\hbar}{i} \frac{\partial}{\partial x} \right) |\psi\rangle \\ &= \left(\frac{\hbar}{i} + xp \right) |\psi\rangle \end{aligned} \quad (6.34)$$

or

$$(xp - px) |\psi\rangle = i\hbar |\psi\rangle. \quad (6.35)$$

This relationship is compactly represented by introducing the *commutator*, defined as $[x, p] \equiv xp - px$. The specific value of the commutator indicates whether or not two operators commute. In this case we have

$$[x, p] = i\hbar. \quad (6.36)$$

The fact that x , p do not commute underlies the *uncertainty principle*, i.e., $\Delta x \Delta p \geq \hbar/2$. The quantities $\mathbf{p}$ and $\mathbf{r}$ are related through a Fourier transform,³ and are referred to as *conjugate variables*.

The quantum mechanical version of the Hamiltonian (6.33) can be constructed by replacing x and p with the corresponding operators from Table 6.1. The time-independent Schrödinger equation (see (2.53)) for the harmonic oscillator is therefore

³ The Fourier transform relation also expresses the uncertainty relation, since a narrow waveform in one domain requires a wide waveform in the other.

Table 6.1 Classical parameters and the corresponding quantum operators.

Quantity	Operator
position $\mathbf{r}$	$\mathbf{r}$
momentum $\mathbf{p}$	$\frac{\hbar}{i} \nabla$
spin $\mathbf{S}$	$\frac{\hbar}{2} \boldsymbol{\sigma}$

$$\mathcal{H} |\psi\rangle = \mathcal{E} |\psi\rangle,$$

$$\left[-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{1}{2} m \omega_0^2 x^2 \right] |\psi\rangle = \mathcal{E} |\psi\rangle. \quad (6.37)$$

Here the second term on the left represents the potential energy $\mathcal{V} = m\omega_0^2 x^2/2$. The solutions to this second-order differential equation are the harmonic oscillator wave functions

$$|\psi_n\rangle = \frac{e^{-x^2/(2\alpha^2)} H_n(x/\alpha)}{\sqrt{\alpha 2^n n! \sqrt{\pi}}}, \quad (6.38)$$

where n is a positive integer, $H_n(y)$ is the n th *Hermite polynomial*, and $\alpha = \sqrt{\hbar/(m\omega_0)}$ is a characteristic length for the system. The first three polynomials are

$$\begin{aligned} H_0(y) &= 1, \\ H_1(y) &= 2y, \\ H_2(y) &= -2 + 4y^2. \end{aligned} \quad (6.39)$$

Hermite polynomials have the following useful properties, referred to as recurrence relations:

$$\frac{dH_n}{dy} = 2nH_{n-1}, \quad (6.40)$$

$$H_{n+1} = 2yH_n - 2nH_{n-1}. \quad (6.41)$$

These recurrence relations lead to the following properties of harmonic oscillator wave functions:

$$\frac{d}{dx} |\psi_n\rangle = \frac{1}{\alpha} \left(\sqrt{\frac{n}{2}} |\psi_{n-1}\rangle - \sqrt{\frac{n+1}{2}} |\psi_{n+1}\rangle \right), \quad (6.42)$$

$$\frac{x}{\alpha} |\psi_n\rangle - \sqrt{\frac{n+1}{2}} |\psi_{n+1}\rangle = \sqrt{\frac{n}{2}} |\psi_{n-1}\rangle. \quad (6.43)$$

The values of n correspond to discrete energy values

$$\mathcal{E}_n = \hbar\omega_0 \left(n + \frac{1}{2} \right). \quad (6.44)$$

The first three harmonic oscillator wave functions are shown in Figure 6.2(a). The wave functions are plotted in arbitrary amplitude units at their respective energy levels in this

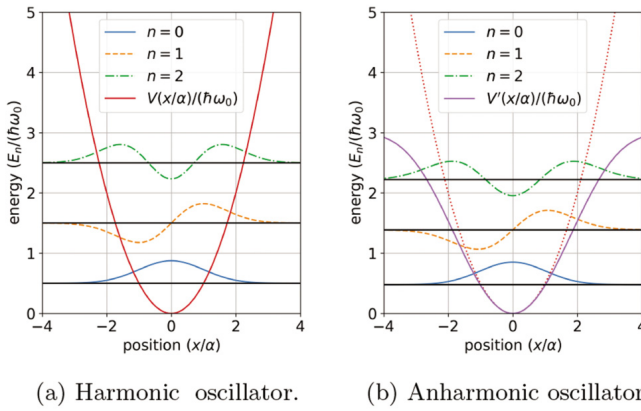


Figure 6.2 Quantum mechanical harmonic oscillator (a) and perturbed harmonic oscillator (b). The wave functions are plotted in arbitrary amplitude units at the associated energy level.

diagram. Note that when $V(x) < E_n$ the solutions are oscillatory, while they decay for $V(x) > E_n$.

6.3.3 Raising and Lowering Operators

The operators x and p are useful for describing the physical properties of a state, but for some applications it is more convenient to have operators that more directly relate to the energy level of a state instead of its physical properties. Consider the operator

$$a = \sqrt{\frac{m\omega_0}{2\hbar}} \left(x + i \frac{p}{m\omega_0} \right). \quad (6.45)$$

It is convenient to normalize the position to the characteristic length α . The operator a becomes

$$\begin{aligned} a &= \frac{1}{\sqrt{2}} \left(\frac{x}{\alpha} + i \frac{1}{\alpha m \omega_0} \frac{\hbar}{i} \frac{d}{dx} \right) \\ &= \frac{1}{\sqrt{2}} \left(\frac{x}{\alpha} + \alpha \frac{d}{dx} \right). \end{aligned} \quad (6.46)$$

To see why this operator is interesting, let us apply it to the wave function:

$$\begin{aligned} a |\psi_n\rangle &= \frac{1}{\sqrt{2}} \left(\frac{x}{\alpha} |\psi_n\rangle + \sqrt{\frac{n}{2}} |\psi_{n-1}\rangle - \sqrt{\frac{n+1}{2}} |\psi_{n+1}\rangle \right) \\ &= \frac{1}{\sqrt{2}} \left(2\sqrt{\frac{n}{2}} |\psi_{n-1}\rangle \right) \\ &= \sqrt{n} |\psi_{n-1}\rangle. \end{aligned} \quad (6.47)$$

Here we used the recurrence relation (6.43) in the second line. Since the operator a changes the state n to $n - 1$, it is called a *lowering* operator.

Next, consider the operator formed by taking the Hermitian conjugate (i.e., transpose⁴ complex conjugate) of a :

$$a^\dagger = \sqrt{\frac{m\omega_0}{2\hbar}} \left(x - i \frac{p}{m\omega_0} \right). \quad (6.48)$$

Normalizing the position to the characteristic length α as before and applying $a^\dagger$ to the wave function lead to

$$a^\dagger |\psi_n\rangle = \sqrt{n+1} |\psi_{n+1}\rangle. \quad (6.49)$$

Here, since $a^\dagger$ takes the state n to the state $n+1$, it is referred to as a *raising* operator.

What happens if both operators are applied to the wave function? Since one lowers and one raises, we expect to stay in the same state:

$$a^\dagger a |\psi_n\rangle = a^\dagger \sqrt{n} |\psi_{n-1}\rangle = n |\psi_n\rangle. \quad (6.50)$$

If the operators are applied in the reverse order, we would still expect to stay in the same state, but let us see if the pre-factor is the same:

$$a a^\dagger |\psi_n\rangle = a \sqrt{n+1} |\psi_{n+1}\rangle = (n+1) |\psi_n\rangle. \quad (6.51)$$

Since the pre-factors are different when the order is reversed, we conclude that the operators *do not commute*. In particular,

$$(a a^\dagger - a^\dagger a) |\psi_n\rangle = |\psi_n\rangle. \quad (6.52)$$

As in the case of position and momentum, this relationship is compactly represented by the commutator $[a, a^\dagger] \equiv a a^\dagger - a^\dagger a$. In this case we have

$$[a, a^\dagger] = 1. \quad (6.53)$$

We can easily invert the pair of Eqs. (6.45) and (6.48) to express x and p in terms of a and $a^\dagger$. The result is

$$x = \sqrt{\frac{\hbar}{2m\omega_0}} (a + a^\dagger), \quad (6.54)$$

$$p = -i\sqrt{\frac{\hbar m\omega_0}{2}} (a - a^\dagger). \quad (6.55)$$

Let's see what the harmonic oscillator Hamiltonian (6.33) looks like when expressed in terms of the new raising and lowering operators. The Hamiltonian becomes

$$\begin{aligned} \mathcal{H} &= -\frac{1}{2m} \frac{1}{2} \left(\frac{\hbar}{\alpha} \right)^2 (a - a^\dagger)^2 + \frac{m\omega_0^2}{2} \frac{\alpha^2}{2} (a + a^\dagger)^2 \\ &= \frac{\hbar\omega_0}{4} (a^\dagger a - a a^\dagger - a^\dagger a^\dagger + a a^\dagger + a^\dagger a + a a^\dagger + a^\dagger a^\dagger + a a^\dagger) \\ &= \frac{\hbar\omega_0}{2} (a^\dagger a + a a^\dagger) \\ &= \hbar\omega_0 \left(a^\dagger a + \frac{1}{2} \right). \end{aligned} \quad (6.56)$$

⁴ In this case the operator is a scalar, so the transpose has no effect. However the dagger notation is commonly used as it does make a difference in cases where, e.g., operators are represented by matrices.

Here we substituted the definition of α and simplified between the first and second line, and used the commutator (6.53) between last two lines. Expressing the Hamiltonian in terms of raising and lowering operators yields a form that only contains the natural frequency and the number operator $a^\dagger a$. In this representation, the wave function can be specified simply by the excited state number: $|\psi_n\rangle \rightarrow |n\rangle$.

6.3.4 Can a Harmonic Oscillator Be Used as a Qubit?

After getting some understanding of the quantum mechanical harmonic oscillator, let us consider for a moment if there is a way to use one as a qubit. Recall that the key requirement of a system to be used as a qubit is for it to have only two states. In contrast, the harmonic oscillator has an unlimited number of states with identical spacing. Consider what would happen if we tried to use one as a qubit. Suppose the harmonic oscillator was prepared in the ground state, and then a π -pulse is used to put it in the first excited state. Now suppose we wanted to return the system to the ground state. With a two-level system, application of another π -pulse will do the trick, but in the case of the harmonic oscillator, the second π -pulse will be just as likely to move the system into the $n = 2$ state as to return it to the ground state!

The spin system we considered in Chapter 2 has exactly two excited states, and so works well as a qubit. However, most other physical systems have more than two states. Having more than two states does not prevent a system from being used as a qubit, provided that the energy spacings of the states are sufficiently different that you can use a pulse frequency that will only couple two of the states. The problem with the harmonic oscillator is that all of the states are equally spaced!

Is there some way to perturb the harmonic oscillator so that all of the states do not have the same spacing? It turns out that the equal spacing of the states is a special property of the quadratic potential function. So if we perturb the potential so that it is no longer exactly quadratic, then the spacing between the levels will no longer be the same.

Before considering perturbations specifically, note that from Eq. (2.53), the energy of any system can be written

$$\mathcal{E} = \langle \psi | \mathcal{H} | \psi \rangle, \quad (6.57)$$

where we have assumed the wave function is normalized, i.e., $\langle \psi | \psi \rangle = 1$. Now suppose that the Hamiltonian has two parts: one part for which one has the exact solution, plus a small perturbation $\mathcal{H} = \mathcal{H}_0 + V'$:

$$\begin{aligned} \mathcal{E} &= \langle \psi | (\mathcal{H}_0 + V') | \psi \rangle \\ &= \langle \psi | \mathcal{H}_0 | \psi \rangle + \langle \psi | V' | \psi \rangle. \end{aligned} \quad (6.58)$$

For sufficiently small perturbations, we can approximate the wave function with that of the unperturbed problem:

$$\begin{aligned} \mathcal{E}'_n &\approx \langle \psi_n | \mathcal{H}_0 | \psi_n \rangle + \langle \psi_n | V' | \psi_n \rangle \\ &\approx \mathcal{E}_n + \langle \psi_n | V' | \psi_n \rangle. \end{aligned} \quad (6.59)$$

Here $|\psi_n\rangle$ is the n th exact solution for the Hamiltonian $\mathcal{H}_0$. As an example, consider Figure 6.2(b). Here a perturbation is applied to the potential that increases with x/α

but is negligible for $x/\alpha \ll 1$. Because the spatial extent of the wave functions $|\psi_n\rangle$ increases with n , the effect of the perturbation is larger as n increases. As a result, $\mathcal{E}'_{n+1} - \mathcal{E}'_n \neq \mathcal{E}'_n - \mathcal{E}'_{n-1}$. If this difference is large enough to be experimentally distinguished, then we can just address transitions between, say, the ground and first excited state, and ignore the rest of the states (at least as an initial approximation).

In the case of our mechanical harmonic oscillator, such a perturbation would result from any departure from the ideal spring force law $F = -Kx^2/2$. We will see shortly that a linear LC circuit behaves like a harmonic oscillator. In this case, a Josephson junction is used as a nonlinear inductor to introduce the perturbation needed to realize a qubit.

6.4 Circuit Quantum Electrodynamics

In this section we explore circuit analogs of the classical and quantum harmonic oscillators discussed previously, and how they can be used to implement qubits.

The most common method of analysis for electrical circuits with lumped elements is the systematic application of *Kirchhoff's laws*: the sum of the currents leaving a node is zero, and the sum of voltages around a loop is zero (assuming no time-changing flux links the circuit). However, similar to what we observed for formulations of mechanics, an alternate formulation based on Lagrange's equations is useful for the transition from classical circuit theory to quantum circuit theory, generally referred to as *circuit quantum electrodynamics* [44–49].

6.4.1 Classical LC Resonant Circuit

Consider the LC parallel circuit shown in Figure 6.3. The classical Hamiltonian for the circuit is just the sum of the energy stored in the inductor and the energy stored in the capacitor:

$$\mathcal{H} = \frac{1}{2}CV^2 + \frac{1}{2}LI_L^2. \quad (6.60)$$

Recognizing that the charge on the capacitor is given by $Q = CV$ and the total flux linkage in the inductor is $\Phi = LI_L$, we can re-write this as

$$\mathcal{H} = \frac{Q^2}{2C} + \frac{\Phi^2}{2L}. \quad (6.61)$$

Comparing with the harmonic oscillator Hamiltonian (6.33), we see that this is of the same form if we make the identifications $m \rightarrow C$ and $L = 1/(C\omega_0^2)$, or

$$\omega_0 = \frac{1}{\sqrt{LC}}. \quad (6.62)$$

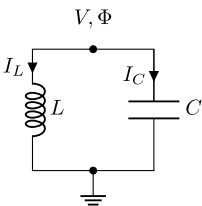


Figure 6.3 LC resonant circuit.

This is the familiar resonant frequency of an LC circuit.

It will be useful for subsequent analyses to discuss the conjugate variables Φ and Q in a bit more detail, before moving on to a quantum analysis of the LC circuit. Note that the current through the inductor I_L and the node voltage V are related by $V = LdI_L/dt$, or assuming a time invariant inductor:

$$V = \frac{d\Phi}{dt}. \quad (6.63)$$

Integrating this equation gives the following expression for the flux Φ :

$$\Phi(t) = \int_{-\infty}^t V(t')dt'. \quad (6.64)$$

Here we assume the voltage vanishes as $t \rightarrow -\infty$. Note that since there is a voltage associated with every node in general, it is also possible to associate a flux with each node. In a similar way, noting that $I_C = dQ/dt$, the charge on the capacitor can be written

$$Q(t) = \int_{-\infty}^t I_C(t')dt', \quad (6.65)$$

where we also assume the current vanishes in the limit $t \rightarrow -\infty$.

We can obtain the “equation of motion” using the Lagrangian formulation discussed in Section 6.1. To do this, we note that the “kinetic” energy term can alternatively be expressed in terms of the time derivative of the flux. In terms of the flux and its derivative, we can construct the Lagrangian as the difference between the “kinetic” and “potential” energies, where the energy in the capacitance is the analog of kinetic energy, and the energy in the inductance is the analog of potential energy:

$$\mathcal{L} = \frac{1}{2}C\dot{\Phi}^2 - \frac{\Phi^2}{2L}. \quad (6.66)$$

The corresponding equation of motion is

$$-\frac{\Phi}{L} - C\ddot{\Phi} = 0, \quad (6.67)$$

which we recognize as having the same form as the equation of motion of the classical mechanical harmonic oscillator (6.28) with $\omega_0 = 1/\sqrt{LC}$ as before.

6.4.2 Quantization of the LC Circuit

Following a similar approach to how we quantized the mechanical harmonic oscillator, we treat the LC circuit quantum mechanically by replacing Q and Φ with corresponding operators satisfying the commutation relation

$$[\Phi, Q] = i\hbar. \quad (6.68)$$

As with the mechanical harmonic oscillator, we can also define raising and lowering operators such that

$$\Phi = \sqrt{\frac{\hbar}{2\omega_0 C}} (a + a^\dagger), \quad (6.69)$$

$$Q = -i\sqrt{\frac{\hbar\omega_0 C}{2}} (a - a^\dagger), \quad (6.70)$$

where the raising and lowering operators satisfy the commutation relation $[a, a^\dagger] = 1$ as before. Substituting these expressions into the Hamiltonian for the LC circuit (6.61) and simplifying lead to

$$\mathcal{H} = \hbar\omega_0 \left(a^\dagger a + \frac{1}{2} \right). \quad (6.71)$$

Since the energy that we are quantizing is actually stored in the electromagnetic fields of the capacitor and the inductor, we have effectively quantized the electromagnetic field. Application of the raising operator $a^\dagger$ corresponds to adding the energy of a photon $\hbar\omega_0$ to the circuit, and application of the lowering operator a subtracts the energy of a photon $\hbar\omega$. Consequently, $a^\dagger$ and a are also referred to as *creation* and *annihilation* operators. Also, the energy level n indicates the number of photons stored in the circuit, and so $a^\dagger a$ gives the photon occupancy of the circuit.

6.4.3 Circuit Electrodynamic Approach for General Circuits

The following procedure is valid for lossless circuits that do not contain mutual inductances, and that are not linked by external magnetic fields.

1. Choose one node as the ground, or reference node. The remaining nodes are referred to as active nodes.
2. For each of the active nodes, assign a node voltage v_i and generalized node flux ϕ_i , where $v_i = \dot{\phi}_i$.
3. Define branch fluxes as $\Phi_b = \phi_n - \phi_{n'}$, where n and n' are adjacent connected nodes.
4. Define the branch voltage as the time derivative of the branch flux, $v_b = \dot{\Phi}_b$.
5. Calculate the energy stored in the capacitors in terms of the branch voltage across each capacitor, and sum to find the total energy stored in capacitors. This stored energy is the analog of the kinetic energy $\mathcal{T}$:

$$\mathcal{T} = \sum_b \frac{1}{2} C_b v_b^2 = \sum_b \frac{1}{2} C_b \dot{\Phi}_b^2. \quad (6.72)$$

6. Calculate the energy stored in the inductors in terms of the branch flux through each inductor, and sum to find the total energy stored in inductance. This stored energy is the analog of the potential energy $\mathcal{V}$:

$$\mathcal{V} = \sum_b \frac{1}{2} L_b i_b^2 = \sum_b \frac{\Phi_b^2}{2L_b}. \quad (6.73)$$

7. Construct the Lagrangian as the difference between the capacitive and inductive stored energies:

$$\mathcal{L}(\phi_1, \dot{\phi}_1, \dots, \phi_N, \dot{\phi}_N) = \mathcal{T} - \mathcal{V}. \quad (6.74)$$

8. The Euler-Lagrange equation for this Lagrangian can be used along with appropriate boundary/initial conditions to describe the modes of the circuit.
 9. The conjugate variables to the node fluxes are the node charges, defined as

$$q_n = \frac{\partial \mathcal{L}}{\partial \dot{\phi}_n}. \quad (6.75)$$

10. Finally, the Hamiltonian is constructed from

$$\mathcal{H}(\phi_1, \dot{\phi}_1, \dots, \phi_N, \dot{\phi}_N) = \sum_{i=1}^N \dot{\phi}_i q_i - \mathcal{L}. \quad (6.76)$$

6.4.4 Circuit Model for Transmission Line Resonator

Let us now apply this procedure to the transmission line circuit shown in Figure 6.4. As shown in the figure, let us choose the bottom node as our reference, and assign node voltages and fluxes as indicated. If the entire circuit was small compared to a wavelength, we could consider the reference node having a single potential. However, since we assume the length of the line could be large, we interpret the node voltages as measured between the node and the reference node at the same location, in other words, the node voltages are the voltages across the corresponding capacitors. In this case, the node voltages and fluxes are the same as the branch voltages and fluxes, so the total kinetic energy is

$$\mathcal{T} = \sum_{i=1}^N \frac{1}{2} C_i v_i^2 = \sum_{i=1}^N \frac{1}{2} C' \Delta z \dot{\phi}_i^2, \quad (6.77)$$

where C' is the capacitance per unit length of the line. Similarly, the total potential energy term is

$$\mathcal{V} = \sum_{i=1}^{N-1} \frac{1}{2} L_i i_i^2 = \sum_{i=1}^{N-1} \frac{(\phi_{i+1} - \phi_i)^2}{2L' \Delta z} \quad (6.78)$$

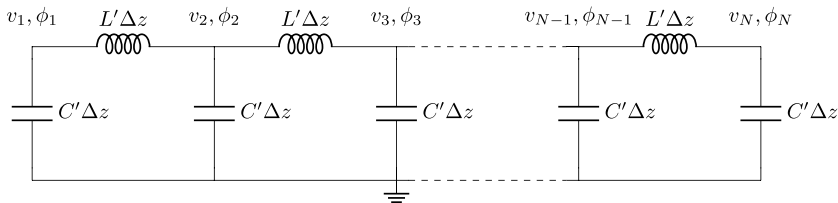


Figure 6.4 Lumped circuit model for an open-circuited section of transmission line of length $N\Delta z = \ell$.

where L' is the inductance per unit length of the line. We can now write the Lagrangian as

$$\mathcal{L}(\phi_1, \dot{\phi}_1, \dots, \phi_N, \dot{\phi}_N) = \sum_{i=1}^N \frac{1}{2} C' \Delta z \dot{\phi}_i^2 - \sum_{i=1}^{N-1} \frac{1}{2L'} \left(\frac{\phi_{i+1} - \phi_i}{\Delta z} \right)^2 \Delta z. \quad (6.79)$$

Taking the limit $N \rightarrow \infty$ this becomes

$$\mathcal{L}(\phi, \dot{\phi}, \phi_z) = \int_0^\ell \left(\frac{1}{2} C' \dot{\phi}^2 - \frac{1}{2L'} \left(\frac{\partial \phi}{\partial z} \right)^2 \right) dz, \quad (6.80)$$

where ℓ is the length of the transmission line. Note that in this case the function ϕ depends on z as well as t . The Euler-Lagrange equation can be easily generalized to the case of two variables by adding a term containing z derivatives in addition to the term containing time derivatives:

$$\frac{\partial \mathcal{L}}{\partial \phi} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\phi}} - \frac{d}{dz} \frac{\partial \mathcal{L}}{\partial \phi_z} = 0. \quad (6.81)$$

(Note that we have made the change of notation $\phi_t \equiv \dot{\phi}$.) Observing that the integrand does not explicitly depend on ϕ , we have $\partial \mathcal{L} / \partial \phi = 0$. For the remaining two terms we have

$$\frac{\partial \mathcal{L}}{\partial \dot{\phi}} = C' \dot{\phi}, \quad (6.82)$$

and

$$\frac{\partial \mathcal{L}}{\partial \phi_z} = -\frac{1}{L'} \frac{\partial \phi}{\partial z}. \quad (6.83)$$

The Euler-Lagrange equation becomes, finally:

$$\frac{\partial^2 \phi}{\partial t^2} - \frac{1}{L'C'} \frac{\partial^2 \phi}{\partial z^2} = 0. \quad (6.84)$$

We recognize this as the wave Equation (4.10), where the wave velocity is given by $v = 1/\sqrt{L'C'}$. For sinusoidal excitations, the general solution to this equation is a superposition of waves traveling to the left and to the right along the line. The precise mix depends on the boundary conditions. In phasor notation,⁵ the general solution is

$$\phi = \phi_0^- e^{j\beta z + j\omega t} + \phi_0^+ e^{-j\beta z + j\omega t}. \quad (6.85)$$

In our case, the transmission line is open-circuited, so the branch currents are zero at each end. Since the flux $\phi(z)$ is proportional to current, we require the flux to also vanish at each end. For ϕ to vanish at $z = 0$ we require $\phi_0^+ = -\phi_0^- \equiv \phi_0$. Making this substitution and multiplying and dividing by $2j$ gives

⁵ Here we have used the time-dependence convention used in Chapter 4 to facilitate reference to that chapter, and to be consistent with phasor notation in Electrical Engineering. Remember that we are using the convention $j = -i$.

$$\begin{aligned}
\phi &= 2j\phi_0 \sin \beta z e^{j\omega t} \\
&= 2|\phi_0| \sin \beta z e^{j\omega t + j\theta}.
\end{aligned} \tag{6.86}$$

Here we have introduced the phase θ associated with the amplitude such that $j\phi_0 = |\phi_0| \exp(j\omega t + j\theta)$. The time-domain expression can be obtained by taking the real part of (6.86):

$$\phi(z, t) = 2|\phi_0| \sin(\beta z) \cos(\omega t + \theta). \tag{6.87}$$

Requiring the flux to vanish at $z = \ell$ leads to the condition

$$\beta_n = \frac{n\pi}{\ell}, \quad \text{or} \quad \ell = \frac{n\lambda_n}{2}. \tag{6.88}$$

In writing the second expression above we have used the relation $\beta = 2\pi/\lambda$. We conclude that modes exist whenever the length of the line is equal to a multiple of a half wavelength where

$$\lambda_n = \frac{v}{f_n} = \frac{1}{f_n \sqrt{L'C'}}. \tag{6.89}$$

In general, the flux can be expressed as the sum of all possible modes:

$$\begin{aligned}
\phi(z, t) &= \sum_{n=1}^{\infty} A_n \sin(n\pi z/\ell) \cos(\omega_n t + \theta_n) \\
&= \sum_{n=1}^{\infty} \Phi_n \sin(n\pi z/\ell)
\end{aligned} \tag{6.90}$$

where we have defined

$$\Phi_n(t) = A_n \cos(\omega_n t + \theta_n). \tag{6.91}$$

Substituting this into the Lagrangian (6.80) gives

$$\begin{aligned}
\mathcal{L} &= \sum_{n=1}^{\infty} \int_0^{\ell} \left(\frac{1}{2} C' \dot{\Phi}_n^2 \sin^2 \left(\frac{n\pi z}{\ell} \right) - \frac{\Phi_n^2}{2L'} \left(\frac{n\pi}{\ell} \right)^2 \cos^2 \left(\frac{n\pi z}{\ell} \right) \right) dz \\
&= \sum_{n=1}^{\infty} \int_0^{n\pi} \left(\frac{1}{2} C' \dot{\Phi}_n^2 \sin^2(\alpha) - \frac{\Phi_n^2}{2L'} \left(\frac{n\pi}{\ell} \right)^2 \cos^2(\alpha) \right) \frac{\ell}{n\pi} d\alpha,
\end{aligned} \tag{6.92}$$

where we have made the change of variable $\alpha = n\pi z/\ell$. Recognizing that

$$\int_0^{n\pi} \cos^2 \alpha d\alpha = \int_0^{n\pi} \sin^2 \alpha d\alpha = \frac{n\pi}{2}, \tag{6.93}$$

the Lagrangian becomes

$$\mathcal{L} = \sum_{n=1}^{\infty} \left(\frac{C'\ell}{4} \dot{\Phi}_n^2 - \frac{(n\pi)^2}{4L'\ell} \Phi_n^2 \right). \tag{6.94}$$

The variables conjugate to the node fluxes can now be found:

$$Q_n = \frac{\partial \mathcal{L}}{\partial \dot{\Phi}_n} = \frac{C'\ell}{2} \dot{\Phi}_n, \tag{6.95}$$

suggesting the effective capacitances $C_n = C'\ell/2$. The Lagrangian now takes the form of a resonant LC circuit for each n :

$$\mathcal{L} = \sum_{n=1}^{\infty} \left(\frac{Q_n^2}{2C_n} - \frac{\Phi_n^2}{2L_n} \right), \quad (6.96)$$

where we have also defined an effective inductance $L_n = 2L'\ell/(n\pi)^2$.

Finally, we can construct the Hamiltonian as

$$\begin{aligned} \mathcal{H} &= \sum_{n=1}^{\infty} Q_n \dot{\Phi}_n - \mathcal{L} \\ &= \sum_{n=1}^{\infty} \left(\frac{C'\ell}{2} \dot{\Phi}_n^2 - \frac{Q_n^2}{2C_n} + \frac{\Phi_n^2}{2L_n} \right) \\ &= \sum_{n=1}^{\infty} \left(\frac{Q_n^2}{C_n} - \frac{Q_n^2}{2C_n} + \frac{\Phi_n^2}{2L_n} \right) \\ &= \sum_{n=1}^{\infty} \left(\frac{Q_n^2}{2C_n} + \frac{\Phi_n^2}{2L_n} \right). \end{aligned} \quad (6.97)$$

6.4.5 Quantizing a Transmission Line Resonator

As with the LC circuit, we can quantize the transmission line circuit by interpreting Q_n and Φ_n as operators with the commutation relation $[\Phi_n, Q_n] = i\hbar$ and introduce raising and lowering operators $a_n^\dagger, a_n$ in a manner analogous to (6.69) and (6.70), so that the Hamiltonian can be written

$$\mathcal{H} = \sum_{n=1}^{\infty} \hbar\omega_n \left(a_n^\dagger a_n + \frac{1}{2} \right), \quad (6.98)$$

with $\omega_n = n\pi v/\ell$. Here each value of n represents a resonant mode of the transmission line characterized by n half wavelengths along the line, and $a_n^\dagger a_n$ gives the number of photons with frequency ω_n , i.e., the photon occupancy of mode n , for the transmission line. Usually we are concerned with exciting a single mode on the transmission line, and in this case the Hamiltonian can be simplified to

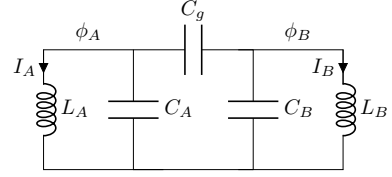
$$\mathcal{H} = \hbar\omega_r \left(a^\dagger a + \frac{1}{2} \right), \quad (6.99)$$

where ω_r is resonant frequency of the mode of interest, and $a^\dagger, a$ are the creation and annihilation operators for that mode. Often the mode of interest is simply the fundamental mode with $n = 1$.

6.4.6 Quantized Coupled LC Resonant Circuits

Consider two lossless LC resonant circuits coupled by a capacitor as shown in Figure 6.5. Note that since a transmission line resonator can be modeled as a lumped-element circuit near one of its resonant frequencies, this could just as well represent two coupled transmission line resonators, or a lumped-element resonator coupled to a transmission line resonator. The conventional classical analysis of this circuit using coupled-mode

Figure 6.5 Capacitively-coupled LC resonant circuits.



theory was presented in Section 5.5. In this section we analyze the circuit using the Lagrangian formulation described in Section 6.4.3, and proceed to construct a quantized Hamiltonian for the circuit using the principles of circuit QED discussed in Section 6.4.2.

The analog to the kinetic energy $\mathcal{T}$ is the sum of the energy stored in the capacitors:

$$\mathcal{T} = \frac{1}{2}C_A\dot{\phi}_A^2 + \frac{1}{2}C_B\dot{\phi}_B^2 + \frac{1}{2}C_g(\dot{\phi}_A - \dot{\phi}_B)^2. \quad (6.100)$$

Similarly, the analog of the potential energy $\mathcal{V}$ is the energy stored in the inductors:

$$\begin{aligned} \mathcal{V} &= \frac{1}{2}L_AI_A^2 + \frac{1}{2}L_BI_B^2 \\ &= \frac{\phi_A^2}{2L_A} + \frac{\phi_B^2}{2L_B}. \end{aligned} \quad (6.101)$$

Here we have used the definition of the fluxes $\phi_j = L_j I_j$. Taking the difference $\mathcal{T} - \mathcal{V}$ forms the Lagrangian

$$\mathcal{L} = \frac{1}{2} \left(C_A\dot{\phi}_A^2 + C_B\dot{\phi}_B^2 + C_g(\dot{\phi}_A - \dot{\phi}_B)^2 - \frac{\phi_A^2}{L_A} - \frac{\phi_B^2}{L_B} \right). \quad (6.102)$$

From step 9 in Section 6.4.3, the node charges are given by

$$q_A = \frac{\partial \mathcal{L}}{\partial \dot{\phi}_A} = (C_A + C_g)\dot{\phi}_A - C_g\dot{\phi}_B \quad (6.103)$$

and

$$q_B = \frac{\partial \mathcal{L}}{\partial \dot{\phi}_B} = (C_B + C_g)\dot{\phi}_B - C_g\dot{\phi}_A. \quad (6.104)$$

These equations can be solved simultaneously to find $\dot{\phi}_{A,B}$ in terms of $q_{A,B}$. Using a matrix approach we can write (6.103) and (6.104) as

$$\begin{bmatrix} q_A \\ q_B \end{bmatrix} = \begin{bmatrix} (C_A + C_g) & -C_g \\ -C_g & (C_B + C_g) \end{bmatrix} \begin{bmatrix} \dot{\phi}_A \\ \dot{\phi}_B \end{bmatrix}. \quad (6.105)$$

Inverting this equation gives

$$\begin{bmatrix} \dot{\phi}_A \\ \dot{\phi}_B \end{bmatrix} = \frac{1}{D} \begin{bmatrix} (C_B + C_g) & C_g \\ C_g & (C_A + C_g) \end{bmatrix} \begin{bmatrix} q_A \\ q_B \end{bmatrix}, \quad (6.106)$$

where D is the matrix determinant

$$\begin{aligned} D &= (C_A + C_g)(C_B + C_g) - C_g^2 \\ &= C_AC_B + C_g(C_A + C_B). \end{aligned} \quad (6.107)$$

The expressions for $\dot{\phi}_{A,B}$ can now be substituted back into the kinetic energy expression (6.100) to express the kinetic energy in terms of $q_{A,B}$. After some rather tedious algebra, we can construct a Hamiltonian for the coupled LC circuit entirely in terms of the conjugate variables $q_{A,B}$ and $\phi_{A,B}$. The result is

$$\begin{aligned}\mathcal{H} &= \mathcal{T} + \mathcal{V} \\ &= \frac{q_A^2}{2C'_A} + \frac{\phi_A^2}{2L_A} + \frac{q_B^2}{2C'_B} + \frac{\phi_B^2}{2L_B} + \frac{q_A q_B}{2C_{AB}},\end{aligned}\quad (6.108)$$

where

$$C'_A = \frac{(C_A C_B + C_g(C_A + C_B))^2}{C_A(C_B + C_g)^2 + C_B C_g^2 + C_B^2 C_g}, \quad (6.109)$$

$$C'_B = \frac{(C_A C_B + C_g(C_A + C_B))^2}{C_B(C_A + C_g)^2 + C_A C_g^2 + C_A^2 C_g}, \quad (6.110)$$

$$C_{AB} = \frac{1}{2C_g} (C_A C_B + C_g(C_A + C_B)). \quad (6.111)$$

If the coupling between the two resonant circuits is weak so that $C_g \ll C_A, C_B$, these expressions reduce to

$$C'_A \approx C_A \left(1 + \frac{C_g}{C_A}\right), \quad (6.112)$$

$$C'_B \approx C_B \left(1 + \frac{C_g}{C_B}\right), \quad (6.113)$$

$$C_{AB} \approx \frac{C_A C_B}{2C_g}. \quad (6.114)$$

Up until this point, our treatment has been entirely classical. To quantize the circuit, we reinterpret the charges and fluxes as operators, and introduce raising and lowering operators just as we did for the harmonic oscillator:

$$\phi_A = \sqrt{\frac{\hbar}{2\omega'_A C'_A}} (a_A + a_A^\dagger), \quad (6.115)$$

$$\phi_B = \sqrt{\frac{\hbar}{2\omega'_B C'_B}} (a_B + a_B^\dagger), \quad (6.116)$$

$$q_A = -i\sqrt{\frac{\hbar\omega'_A C'_A}{2}} (a_A - a_A^\dagger), \quad (6.117)$$

$$q_B = -i\sqrt{\frac{\hbar\omega'_B C'_B}{2}} (a_B - a_B^\dagger), \quad (6.118)$$

where

$$\omega'_A = \frac{1}{\sqrt{L_A C'_A}}, \quad \omega'_B = \frac{1}{\sqrt{L_B C'_B}}. \quad (6.119)$$

Substituting these expressions into the Hamiltonian (6.108) and simplifying leads to

$$\begin{aligned} \mathcal{H} = & \hbar\omega'_A \left(a_A^\dagger a_A + \frac{1}{2} \right) + \hbar\omega'_B \left(a_B^\dagger a_B + \frac{1}{2} \right) \\ & - \hbar g \left(a_A - a_A^\dagger \right) \left(a_B - a_B^\dagger \right), \end{aligned} \quad (6.120)$$

where

$$g = \frac{1}{4} \sqrt{\omega'_A \omega'_B} \frac{\sqrt{C'_A C'_B}}{C_{AB}}. \quad (6.121)$$

6.4.7 Schrödinger, Heisenberg, and Interaction Pictures

To further understand and analyze the quantized Hamiltonian of the coupled LC resonator circuit (6.120), it is useful to make a brief side-trip into three different formulations of quantum mechanics.

We have been using a formulation of quantum mechanics referred to as the *Schrödinger picture*. In this formulation, the state vector is time dependent, but the operators are not. In the *Heisenberg picture*, the state vectors are constant and the operators and observables are time dependent, while in the *Interaction picture* (also called the *Dirac picture*) both the operators and the state vectors are time dependent. While the Interaction picture may sound more complicated, we will see that it provides helpful simplifications in cases where we can divide a Hamiltonian up into two pieces where one piece describes a problem that can be exactly solved, while the other describes a small perturbation to this solvable problem. We will concentrate on the Schrödinger and Interaction pictures here.

Let us first consider the Schrödinger picture that has been the basis of our analysis up to this point. The state vectors satisfy the Schrödinger equation

$$i\hbar\partial_t |\psi(t)\rangle_S = \mathcal{H} |\psi(t)\rangle_S, \quad (6.122)$$

where the time evolution of the state vector is given by

$$|\psi(t)\rangle_S = e^{i\mathcal{H}t/\hbar} |\psi(0)\rangle_S. \quad (6.123)$$

Now suppose that the Hamiltonian can be separated into two parts as described above: $\mathcal{H} = \mathcal{H}_0 + \mathcal{V}$, where $\mathcal{V}$ is a perturbation on a Hamiltonian for which the state vector and energies (eigenvectors and eigenvalues) are known. In the Interaction picture, the state vector is obtained from the Schrödinger state vector by a transformation that only involves the unperturbed component of the Hamiltonian:

$$|\psi(t)\rangle_I = e^{i\mathcal{H}_0 t/\hbar} |\psi(t)\rangle_S. \quad (6.124)$$

To appreciate the advantage of the Interaction picture, consider the time derivative of $|\psi(t)\rangle_I$:

$$\begin{aligned}
 i\hbar\partial_t |\psi(t)\rangle_I &= i\hbar\partial_t (e^{i\mathcal{H}_0 t/\hbar} |\psi(t)\rangle_S) \\
 &= e^{i\mathcal{H}_0 t/\hbar} (i\hbar\partial_t - \mathcal{H}_0) |\psi(t)\rangle_S \\
 &= e^{i\mathcal{H}_0 t/\hbar} (\mathcal{H}_0 + \mathcal{V} - \mathcal{H}_0) |\psi(t)\rangle_S \\
 &= e^{i\mathcal{H}_0 t/\hbar} \mathcal{V} |\psi(t)\rangle_S \\
 &= \underbrace{e^{i\mathcal{H}_0 t/\hbar} \mathcal{V} e^{-i\mathcal{H}_0 t/\hbar}}_{\mathcal{V}_I(t)} \underbrace{e^{i\mathcal{H}_0 t/\hbar} |\psi(t)\rangle_S}_{|\psi(t)\rangle_I} \\
 &= \mathcal{V}_I |\psi(t)\rangle_I.
 \end{aligned} \tag{6.125}$$

We see, then, that the state vector in the Interaction picture obeys a Schrödinger-like equation, but which only involves the perturbation $\mathcal{V}_I$.

With this background, let us transform the Hamiltonian (6.120) into the Interaction picture. The perturbation part of the Hamiltonian becomes

$$\begin{aligned}
 \mathcal{V}_I &= -\hbar g e^{i\mathcal{H}_0 t/\hbar} (a_A - a_A^\dagger) (a_B - a_B^\dagger) e^{-i\mathcal{H}_0 t/\hbar} \\
 &= -\hbar g e^{i\mathcal{H}_0 t/\hbar} (a_A a_B - a_A^\dagger a_B - a_A a_B^\dagger + a_A^\dagger a_B^\dagger) e^{-i\mathcal{H}_0 t/\hbar},
 \end{aligned} \tag{6.126}$$

where $\mathcal{H}_0 = \mathcal{H}_A + \mathcal{H}_B$ and

$$\mathcal{H}_A = \hbar\omega'_A \left(a_A^\dagger a_A + \frac{1}{2} \right), \tag{6.127}$$

$$\mathcal{H}_B = \hbar\omega'_B \left(a_B^\dagger a_B + \frac{1}{2} \right). \tag{6.128}$$

Let us consider the first term in (6.126) in a bit more detail. Note that we can separate the transformation of the product of $a_A a_B$ into separate transformations of a_A and a_B by inserting an equivalent identity operator $\exp(-i\mathcal{H}_0 t/\hbar) \exp(+i\mathcal{H}_0 t/\hbar)$ between the two lowering operators:

$$(e^{i\mathcal{H}_0 t/\hbar} a_A e^{-i\mathcal{H}_0 t/\hbar}) (e^{i\mathcal{H}_0 t/\hbar} a_B e^{-i\mathcal{H}_0 t/\hbar}). \tag{6.129}$$

Note that if $\mathcal{H}_0$ commutes with $a_{A,B}$, then the exponentials before and after the lowering operators would simply cancel. For example, since $\mathcal{H}_A$ and $\mathcal{H}_B$ operate on different subsystems, i.e., resonator A vs. resonator B , $\exp(i\mathcal{H}_B t/\hbar)$ does commute with a_A , so this component cancels in the transformation of a_A :

$$\begin{aligned}
 e^{i\mathcal{H}_0 t/\hbar} a_A e^{-i\mathcal{H}_0 t/\hbar} &= e^{i\mathcal{H}_A t/\hbar} a_A e^{-i\mathcal{H}_A t/\hbar} (e^{i(\mathcal{H}_B - \mathcal{H}_B) t/\hbar}) \\
 &= e^{i\mathcal{H}_A t/\hbar} a_A e^{-i\mathcal{H}_A t/\hbar}.
 \end{aligned} \tag{6.130}$$

Similarly, $\exp(i\omega'_A t/2)$ commutes with both lowering operators, since it is simply a scalar number:

$$\begin{aligned}
 e^{i\mathcal{H}_A t/\hbar} a_A e^{-i\mathcal{H}_A t/\hbar} &= e^{i\omega'_A a_A^\dagger a_A t} a_A e^{-i\omega'_A a_A^\dagger a_A t} (e^{i(\omega'_A - \omega'_A) t/2}) \\
 &= e^{i\omega'_A a_A^\dagger a_A t} a_A e^{-i\omega'_A a_A^\dagger a_A t}.
 \end{aligned} \tag{6.131}$$

To simplify this expression let us define $\alpha = i\omega'_A t$, and

$$f(\alpha) = e^{\alpha a^\dagger a} a e^{-\alpha a^\dagger a}, \quad (6.132)$$

where for the moment we have omitted the subscripts for clarity. Taking the derivative with respect to α gives

$$\begin{aligned} \frac{df}{d\alpha} &= e^{\alpha a^\dagger a} (a^\dagger a a - a a^\dagger a) e^{-\alpha a^\dagger a} \\ &= e^{\alpha a^\dagger a} (a^\dagger a - a a^\dagger) a e^{-\alpha a^\dagger a}. \end{aligned} \quad (6.133)$$

However, we know that $a a^\dagger - a^\dagger a = 1$, so

$$\begin{aligned} \frac{df}{d\alpha} &= e^{\alpha a^\dagger a} (a^\dagger a - (1 + a^\dagger a)) a e^{-\alpha a^\dagger a} \\ &= -e^{\alpha a^\dagger a} a e^{-\alpha a^\dagger a} \\ &= -f. \end{aligned} \quad (6.134)$$

The solution of $df/d\alpha = -f$ is simply

$$f(\alpha) = A e^{-\alpha}, \quad (6.135)$$

where A is a constant. We can find the value of A by substituting $\alpha = 0$ into (6.132). This gives $f(0) = A = a$, from which we obtain an equivalent expression for f :

$$f = a e^{-\alpha} = a e^{-i\omega'_A t}. \quad (6.136)$$

Following a similar procedure for transforming $a^\dagger$ leads to

$$e^{\alpha a^\dagger a} a^\dagger e^{-\alpha a^\dagger a} = a^\dagger e^\alpha = a^\dagger e^{i\omega'_A t}. \quad (6.137)$$

The Hamiltonian in the Interaction picture (6.126) can therefore be written

$$\begin{aligned} \mathcal{V}_I &= -\hbar g \left(a_A a_B e^{-i(\omega'_A + \omega'_B)t} - a_A^\dagger a_B e^{i(\omega'_A - \omega'_B)t} \right. \\ &\quad \left. - a_A a_B^\dagger e^{-i(\omega'_A - \omega'_B)t} + a_A^\dagger a_B^\dagger e^{i(\omega'_A + \omega'_B)t} \right). \end{aligned} \quad (6.138)$$

Note that if $\omega'_A \approx \omega'_B$, the first and last terms are rapidly oscillating in time while the second and third terms are slowly oscillating. We encountered a situation similar to this in Chapter 2 in the context of the rotating wave approximation. Consistent with this approximation, we assume that on time scales of interest (e.g., time intervals needed to measure or manipulate the weakly-coupled circuits) the rapidly-oscillating terms average to zero, leaving us only with the slowly-oscillating terms. We have finally

$$\mathcal{V}_I \approx \hbar g \left(a_A^\dagger a_B e^{i\Delta t} + a_A a_B^\dagger e^{-i\Delta t} \right), \quad (6.139)$$

where $\Delta = \omega'_B - \omega'_A$.

A moment's thought shows that this result is consistent with energy conservation. The surviving terms $a_A^\dagger a_B$ and $a_A a_B^\dagger$ represent destroying a photon on resonator B and creating a photon on resonator A or vice versa. This simply represents energy being transferred from one resonator to the other, which is permitted by energy conservation (provided the time interval over which this transfer occurs is small compared to

$1/|\omega'_B - \omega'_A|$). In contrast, the terms $a_A a_B$ and $a_A^\dagger a_B^\dagger$ represent either destroying or creating photons simultaneously on both resonators—processes that would violate energy conservation.

Consider the lowest excited states of the coupled-resonator system: the states with a single photon either in resonator A or in resonator B. Let us attempt to construct an interaction picture solution by adding up eigenstates of the unperturbed Hamiltonian, but with time-dependent coefficients:

$$|\psi(t)\rangle_I = c_{01}(t) |01\rangle + c_{10}(t) |10\rangle. \quad (6.140)$$

Substituting this trial solution into the interaction picture equation of motion (6.125) gives

$$\begin{aligned} i\hbar\partial_t |\psi(t)\rangle_I &= \mathcal{V}_I |\psi(t)\rangle_I \\ i\hbar\partial_t (c_{01} |01\rangle + c_{10} |10\rangle) &= \hbar g \left(a_A^\dagger a_B e^{i-\Delta t} + a_A a_B^\dagger e^{i\Delta t} \right) \\ &\quad \times (c_{01} |01\rangle + c_{10} |10\rangle) \\ &= \hbar g (c_{01} e^{-i\Delta t} |10\rangle + c_{10} e^{i\Delta t} |01\rangle). \end{aligned} \quad (6.141)$$

Here we have used the facts that $a_A b_B^\dagger |10\rangle = |01\rangle$, $a_A b_B^\dagger |01\rangle = 0$, $a_A^\dagger a_B |01\rangle = |10\rangle$, and $a_A^\dagger a_B |10\rangle = 0$ (see (6.47) and (6.49)). This can be separated into two simultaneous equations by making use of the orthogonality of the states $|01\rangle$ and $|10\rangle$. Specifically, $\langle 01|01\rangle = \langle 10|10\rangle = 1$ and $\langle 01|10\rangle = \langle 10|01\rangle = 0$. Multiplying from the left by $\langle 01|$ gives

$$\partial_t c_{01} = -igc_{10} e^{i\Delta t}, \quad (6.142)$$

while multiplying from the left by $\langle 10|$ gives

$$\partial_t c_{10} = -igc_{01} e^{-i\Delta t}. \quad (6.143)$$

We recognize these as the familiar coupled-mode equations that we first encountered in Chapter 2 (see (2.103) and (2.104)). These are also the same equations that we obtained from our classical analysis of coupled resonators in Section 5.5 (see (5.65) and (5.68)).⁶

From our studies of the solutions to the coupled-mode equations, we see that if the frequencies of the two resonators are sufficiently close, adding the coupling capacitor will allow the probability of finding the photon on a particular resonator to vary sinusoidally with time. When there is a finite probability for the photon to be on either resonator, the states $|01\rangle$ and $|10\rangle$ are entangled. Maximum entanglement is obtained when the photon is equally likely to be found on either resonator (i.e., states $|01\rangle$ and $|10\rangle$ are equally likely).

6.4.8 Resonant Circuits and Qubits

Although our analysis in the previous section was for two coupled resonant circuits, since only the states $|01\rangle$ and $|10\rangle$ were involved, the result would have been the same for two coupled qubits.

⁶ Note that $\kappa = g$ to first order in C_g , and recall that $j = -i$.

Another structure of interest for quantum computers is a transmission line coupled to a qubit. The transmission line can be modeled as an LC resonant circuit near one of its resonant modes, so the Hamiltonian terms from the previous section are appropriate. However, to model a qubit, we need to impose the constraint that there can be only two states (as opposed to the infinite ladder of states of the linear harmonic oscillator). As we will see, this situation can be approximated by using a nonlinear inductor in the resonant circuit. In such a case, the photons coupled to the nonlinear resonator could at most excite the state $|1\rangle$ from the ground state $|0\rangle$, since the energy difference between state $|1\rangle$ and higher-level states would not match the energy of available photons. Consequently the energies of the accessible states would correspond to having either zero or one photons (i.e., $n = 0, 1$, where n is the eigenvalue of $a^\dagger a$). Taking resonant circuit B to be nonlinear, the corresponding energies would be $E_0 = \hbar\omega'_B/2$ and $E_1 = 3\hbar\omega''_B/2$. Since the value of the reference energy does not affect the system dynamics, we can subtract off the average value $(E_0 + E_1)/2$, resulting in energy values of $\pm\hbar\Omega/2$, where $\Omega = E_1 - E_0$. We encountered a system with energy eigenvalues of this form in Section 2.2.2.1: a single spin in a magnetic field with Hamiltonian $-\hbar\Omega\sigma_z/2$. Consequently, to model a qubit, we can replace the harmonic oscillator Hamiltonian $\hbar\omega'_B(a_B^\dagger a_B + 1/2)$ with the two state qubit Hamiltonian $-\hbar\Omega\sigma_z/2$.

To model the qubit in the interaction term, we need two-state operators that raise and lower the state similar to $a, a^\dagger$ for the harmonic oscillator. With this in mind, consider the following operators:

$$\sigma_\pm = \frac{1}{2}(\sigma_x \pm i\sigma_y). \quad (6.144)$$

Using the definitions of the Pauli matrices (2.21), we have

$$\begin{aligned} \sigma_+ &= \frac{1}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \frac{i}{2} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \end{aligned} \quad (6.145)$$

and

$$\begin{aligned} \sigma_- &= \frac{1}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} - \frac{i}{2} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}. \end{aligned} \quad (6.146)$$

With our usual definitions of ground and excited states, it immediately follows that

$$\sigma_+ |0\rangle = 0 \quad (6.147)$$

$$\sigma_+ |1\rangle = |0\rangle \quad (6.148)$$

$$\sigma_- |0\rangle = |1\rangle \quad (6.149)$$

$$\sigma_- |1\rangle = 0. \quad (6.150)$$

Recall from the Bloch sphere representation that by convention $|0\rangle$ is represented by a vector pointing up, while $|1\rangle$ is represented by a vector pointing down. Since σ_+ converts a vector pointing down into a vector pointing up, we can interpret it as an operator that

increases the value of the z component of the state vector. Alternatively, if our two-state system consisted of a spin in a magnetic field as we discussed in Section 2.2.2.1, then σ_+ raises the z component of the spin. It then makes sense if $\sigma_+ |0\rangle = 0$, since in a two-state system the z component cannot be raised any higher than the ground state. Similarly, σ_- converts a vector pointing up to a vector pointing down, and so lowers the value of the z projection of the state vector.

We can therefore consider σ_+ and σ_- to be spin raising and lowering operators, respectively. However, the terminology here can become a bit confusing; we previously used the terms “raising” and “lowering” to refer to operators that raised or lowered the *energy* of a system. In the present case, *raising* the z component of the state vector *lowers* the energy! Since it is intuitive for “+” to indicate raising and “−” to indicate lowering, we define *qubit* raising and lowering operators as

$$\hat{\sigma}_{\pm} = \frac{1}{2} (\sigma_x \mp i\sigma_y). \quad (6.151)$$

With this definition, the two-state system operators that have the same effects as the harmonic oscillator operators $a, a^\dagger$ are

$$a \rightarrow \hat{\sigma}_-, \quad a^\dagger \rightarrow \hat{\sigma}_+. \quad (6.152)$$

Putting these pieces and observations together, we can write down the Hamiltonian corresponding to a linear resonator or resonant circuit coupled to a qubit. Starting from the Schrödinger picture expression for the Hamiltonian for two coupled linear resonant circuits (6.120), we first replace the harmonic oscillator Hamiltonian for circuit B with the qubit Hamiltonian, $\hbar\omega'_B(a_B^\dagger a_B + 1/2) \rightarrow -\hbar\Omega\sigma_z/2$, neglect the coupling terms $a_A a_B$ and $a_A^\dagger a_B^\dagger$ since they do not satisfy energy conservation under our assumptions, and finally make the substitutions $a_B \rightarrow \hat{\sigma}_-$ and $a_B^\dagger \rightarrow \hat{\sigma}_+$ to convert from harmonic oscillator raising and lowering operators to spin raising and lowering operators. The result is

$$\mathcal{H}_{JC} = \hbar\omega \left(a^\dagger a + \frac{1}{2} \right) - \frac{\hbar\Omega}{2} \sigma_z + \hbar g (a \hat{\sigma}_+ + a^\dagger \hat{\sigma}_-). \quad (6.153)$$

Here we have also dropped the subscripts from the operators $a, a^\dagger$ since there is no longer the need to distinguish between two resonators, and re-named the qubit frequencies $\omega'_A \equiv \omega$ and $\Omega \equiv \omega'_B - \omega'_B$ to simplify the notation. This Hamiltonian is known as the *Jaynes-Cummings* Hamiltonian, and has been studied extensively in quantum optics where it also applies to the excitation of atoms in an optical cavity (the study in the quantum optics case is known as *cavity quantum electrodynamics*).

If we limit both resonators to the ground and first excited states, then we can also convert (6.120) to represent the case of two capacitively-coupled qubits by making the substitutions $\hbar\omega'_{A,B}(a_{A,B}^\dagger a_{A,B} + 1/2) \rightarrow -\hbar\omega_{2,1}\sigma_{2,1}^z/2$, along with $a_{A,B} \rightarrow \hat{\sigma}_{2,1}^-$ and $a_{A,B}^\dagger \rightarrow \hat{\sigma}_{2,1}^+$. The result is

$$\begin{aligned} \mathcal{H}_{2q} &= -\frac{\hbar\omega_2}{2} \sigma_2^z - \frac{\hbar\omega_1}{2} \sigma_1^z - \hbar g (\hat{\sigma}_2^- - \hat{\sigma}_2^+) (\hat{\sigma}_1^- - \hat{\sigma}_1^+) \\ &= -\frac{\hbar\omega_2}{2} \sigma_2^z - \frac{\hbar\omega_1}{2} \sigma_1^z + \hbar g \sigma_2^y \sigma_1^y. \end{aligned} \quad (6.154)$$

Note that we have made the identification $A, B \rightarrow 2, 1$ so that σ_1 is on the right as the least significant qubit, according to our convention. In the literature the interaction will

often be written $\sigma_x \sigma_x$ instead of $\sigma_y \sigma_y$. In this context the choice is arbitrary and does not impact the dynamics of the system [48]. Consequently, one often sees the coupled 2-qubit Hamiltonian written

$$\mathcal{H}_{2q} = -\frac{\hbar\omega_2}{2}\sigma_2^z - \frac{\hbar\omega_1}{2}\sigma_1^z + \hbar g \sigma_2^x \sigma_1^x. \quad (6.155)$$

Note that we are using a shorthand by using a subscript to indicate which qubit the operator affects. Making this more explicit using the tensor product notation leads to

$$\mathcal{H}_{2q} = -\frac{\hbar\omega_2}{2}\sigma_z \otimes I - \frac{\hbar\omega_1}{2}I \otimes \sigma_z + \hbar g \sigma_x \otimes \sigma_x. \quad (6.156)$$

6.4.9 The Dispersive Regime

In our previous analyses of coupled resonators, we have taken the two resonant frequencies to be close together. This is the condition that most strongly couples the resonators and facilitates entanglement.

However, it is also of interest to consider the regime where the resonators are detuned so that $|\omega - \Omega| \gg g$, where $\hbar g$ is the coupling energy. This is referred to as the *dispersive regime*, and is useful for reading out the state of a qubit as we will see. To consider this regime, we must return to the original coupled-resonator Hamiltonian (6.120). Making the same changes in notation consistent with interpreting one resonator as a qubit that we discussed in the last section, we have

$$\mathcal{H} = \hbar\omega \left(a^\dagger a + \frac{1}{2} \right) - \frac{\hbar\Omega}{2}\sigma_z - \bar{g}(a - a^\dagger)(\hat{\sigma}_- - \hat{\sigma}_+). \quad (6.157)$$

The approach that we will use for the dispersive regime is to make a unitary transformation on the Hamiltonian that eliminates linear terms in the coupling constant, g . This is called the *Schrieffer-Wolff* transformation. Specifically, we want to consider a transformation of the form

$$\widetilde{\mathcal{H}} = e^S \mathcal{H} e^{-S} = e^S (\mathcal{H}_0 + \mathcal{H}_{\text{int}}) e^{-S}, \quad (6.158)$$

where

$$\mathcal{H}_0 = \hbar\omega \left(a^\dagger a + \frac{1}{2} \right) - \frac{\hbar\Omega}{2}\sigma_z \quad (6.159)$$

and

$$\begin{aligned} \mathcal{H}_{\text{int}} &= -\hbar g (a - a^\dagger)(\hat{\sigma}_- - \hat{\sigma}_+) \\ &= -i\hbar g (a - a^\dagger)\sigma_y. \end{aligned} \quad (6.160)$$

Our task is to find S so that the linear terms in g are eliminated. Expanding $\exp(S)$ and $\exp(-S)$ in Taylor series gives

$$\begin{aligned} \widetilde{\mathcal{H}} &= \left[1 + S + \frac{S^2}{2} \dots \right] \mathcal{H} \left[1 - S + \frac{S^2}{2} \dots \right] \\ &= \mathcal{H} + S\mathcal{H} - \mathcal{H}S + \frac{1}{2}S^2\mathcal{H} + \frac{1}{2}\mathcal{H}S^2 - S\mathcal{H}S + \dots \\ &= \mathcal{H}_0 + \mathcal{H}_{\text{int}} + [S, \mathcal{H}_0] + [S, \mathcal{H}_{\text{int}}] + \frac{1}{2}\{S^2, \mathcal{H}_0\} - S\mathcal{H}_0S + \dots \end{aligned} \quad (6.161)$$

Here we have used the *anticommutator* $\{a, b\} \equiv ab + ba$ to write the expansion more compactly. To eliminate the terms linear in g we want to find S such that

$$[S, \mathcal{H}_0] = -\mathcal{H}_{\text{int}}. \quad (6.162)$$

Imposing this condition means that $S \sim g$. With this in mind, note that the last three terms in (6.161) are second order in g , and all additional terms are higher-order in g . The transformed Hamiltonian becomes

$$\widetilde{\mathcal{H}} = \mathcal{H}_0 + [S, \mathcal{H}_{\text{int}}] + \frac{1}{2}\{S^2, \mathcal{H}_0\} - S\mathcal{H}_0S + \dots \quad (6.163)$$

The second-order terms can be simplified as follows:

$$\begin{aligned} \widetilde{\mathcal{H}} - \mathcal{H}_0 &= [S, \mathcal{H}_{\text{int}}] + \frac{1}{2}\{S^2, \mathcal{H}_0\} - S\mathcal{H}_0S + \dots \\ &= [S, \mathcal{H}_{\text{int}}] + \frac{1}{2}S(S\mathcal{H}_0 - \mathcal{H}_0S) + \frac{1}{2}(\mathcal{H}_0S - S\mathcal{H}_0)S + \dots \\ &= [S, \mathcal{H}_{\text{int}}] + \frac{1}{2}S[S, \mathcal{H}_0] - \frac{1}{2}[S, \mathcal{H}_0]S + \dots \end{aligned} \quad (6.164)$$

Using (6.162) this becomes

$$\begin{aligned} \widetilde{\mathcal{H}} - \mathcal{H}_0 &= [S, \mathcal{H}_{\text{int}}] - \frac{1}{2}(S\mathcal{H}_{\text{int}} - \mathcal{H}_{\text{int}}S) + \dots \\ &= [S, \mathcal{H}_{\text{int}}] - \frac{1}{2}[S, \mathcal{H}_{\text{int}}] + \dots \\ &= \frac{1}{2}[S, \mathcal{H}_{\text{int}}] + \dots \end{aligned} \quad (6.165)$$

So our task of finding the second-order terms reduces to evaluating $[S, \mathcal{H}_{\text{int}}]$. We can do this using the trial form

$$S = (u\sigma_x + v\sigma_y)a + (\tilde{u}\sigma_x + \tilde{v}\sigma_y)a^\dagger, \quad (6.166)$$

where $u, v, \tilde{u}$, and $\tilde{v}$ are constants to be determined so that (6.162) is satisfied. The two parts of $\mathcal{H}_0$ that do not commute with S are σ_z and $a^\dagger a$. Using the commutation relations $[a, a^\dagger a] = a$, $[a^\dagger, a^\dagger a] = -a^\dagger$, $[\sigma_x, \sigma_z] = -2i\sigma_y$, and $[\sigma_y, \sigma_z] = 2i\sigma_x$, we have

$$\begin{aligned} [S, \sigma_z] &= 2i[(v\sigma_x - u\sigma_y)a + (\tilde{v}\sigma_x - \tilde{u}\sigma_y)a^\dagger] \\ &= 2i[(va + \tilde{v}a^\dagger)\sigma_x - (ua + \tilde{u}a^\dagger)\sigma_y] \end{aligned} \quad (6.167)$$

and

$$\begin{aligned} [S, a^\dagger a] &= (u\sigma_x + v\sigma_y)a - (\tilde{u}\sigma_x + \tilde{v}\sigma_y)a^\dagger \\ &= (ua - \tilde{u}a^\dagger)\sigma_x + (va - \tilde{v}a^\dagger)\sigma_y. \end{aligned} \quad (6.168)$$

Putting the pieces together we have

$$\begin{aligned}
[S, \mathcal{H}_0] &= \hbar\omega[S, a^\dagger a] - \frac{\hbar\Omega}{2}[S, \sigma_z] \\
&= [\hbar\omega(ua - \tilde{u}a^\dagger) - i\hbar\Omega(v a + \tilde{v}a^\dagger)]\sigma_x \\
&\quad + [\hbar\omega(v a - \tilde{v}a^\dagger) + i\hbar\Omega(ua + \tilde{u}a^\dagger)]\sigma_y \\
&= [(\hbar\omega u - i\hbar\Omega v)a - (\hbar\omega\tilde{u} + i\hbar\Omega\tilde{v})a^\dagger]\sigma_x \\
&\quad + [(\hbar\omega v + i\hbar\Omega u)a - (\hbar\omega\tilde{v} - i\hbar\Omega\tilde{u})a^\dagger]\sigma_y.
\end{aligned} \tag{6.169}$$

For this expression to be equal to $-\mathcal{H}_{\text{int}}$, the coefficient of σ_x must vanish. This will be satisfied if

$$u = i\frac{\Omega}{\omega}v, \quad \text{and} \quad \tilde{u} = -i\frac{\Omega}{\omega}\tilde{v}. \tag{6.170}$$

We must next compare the coefficients of $a, a^\dagger$ in the σ_y terms in $[S, \mathcal{H}_0]$ and $-\mathcal{H}_{\text{int}}$. Requiring the coefficients to be the same in both expressions gives

$$\omega v + i\Omega u = ig, \quad \text{and} \quad \omega\tilde{v} - i\Omega\tilde{u} = ig. \tag{6.171}$$

Solving (6.170) and (6.171) simultaneously for $u, v, \tilde{u}, \tilde{v}$ gives

$$\tilde{u} = -u = \frac{\Omega g}{\omega^2 - \Omega^2}, \tag{6.172}$$

and

$$\tilde{v} = v = \frac{i\omega g}{\omega^2 - \Omega^2}. \tag{6.173}$$

Substituting these expressions into (6.166) gives

$$S = \frac{\Omega g}{\Omega^2 - \omega^2} \left[\left(\sigma_x - i\frac{\omega}{\Omega}\sigma_y \right) a - \left(\sigma_x + i\frac{\omega}{\Omega}\sigma_y \right) a^\dagger \right]. \tag{6.174}$$

Now if the frequency difference is large compared with g but small compared with ω and Ω , i.e., if $\Omega, \omega \gg |\Omega - \omega| \gg g$, then

$$\begin{aligned}
S &\approx \frac{\Omega g}{2\Omega(\Omega - \omega)} [(\sigma_x - i\sigma_y)a - (\sigma_x + i\sigma_y)a^\dagger] \\
&= \frac{g}{\Omega - \omega} [\hat{\sigma}_+ a - \hat{\sigma}_- a^\dagger].
\end{aligned} \tag{6.175}$$

We can now evaluate the second-order terms in $\widetilde{\mathcal{H}}$. The needed commutator is

$$\begin{aligned}
[S, \mathcal{H}_{\text{int}}] &= \frac{-i\hbar g^2}{\Omega - \omega} [\hat{\sigma}_+ a - \hat{\sigma}_- a^\dagger, (a - a^\dagger)\sigma_y] \\
&= \frac{-i\hbar g^2}{\Omega - \omega} (-[\hat{\sigma}_- a^\dagger, a\sigma_y] - [\hat{\sigma}_+ a, a^\dagger\sigma_y]).
\end{aligned} \tag{6.176}$$

Note that there are four terms when the commutator is expanded, but only two survive. The reason is that the expectation value of terms with aa and $a^\dagger a^\dagger$ vanishes, since $\langle n|aa|n\rangle \sim \langle n|n-2\rangle = 0$ and $\langle n|a^\dagger a^\dagger|n\rangle \sim \langle n|n+2\rangle = 0$, where $|n\rangle$ is the state with n photons in the resonator.

Continuing the calculation and simplifying using the commutation relations $[a, a^\dagger] = 1$, $[\sigma_x, \sigma_y] = 2i\sigma_z$, and $[\hat{\sigma}_-, \sigma_y] = [\hat{\sigma}_+, \sigma_y] = i\sigma_z$, we have

$$\begin{aligned}
 [S, \mathcal{H}_{\text{int}}] &= \frac{-i\hbar g^2}{\Omega - \omega} (a\sigma_y\hat{\sigma}_-a^\dagger - \hat{\sigma}_-a^\dagger a\sigma_y + a^\dagger\sigma_y\hat{\sigma}_+a - \hat{\sigma}_+aa^\dagger\sigma_y) \\
 &= \frac{-i\hbar g^2}{\Omega - \omega} (\sigma_y\hat{\sigma}_- - \hat{\sigma}_+\sigma_y + (\sigma_y\hat{\sigma}_- - \hat{\sigma}_-\sigma_y + \sigma_y\hat{\sigma}_+ - \hat{\sigma}_+\sigma_y)a^\dagger a) \\
 &= \frac{-i\hbar g^2}{\Omega - \omega} [(\sigma_y\sigma_x + i - \sigma_x\sigma_y + i)/2 - 2i\sigma_z a^\dagger a] \\
 &= \frac{-i\hbar g^2}{\Omega - \omega} (i - i\sigma_z - 2i\sigma_z a^\dagger a) \\
 &= \frac{\hbar g^2}{\omega - \Omega} (-1 + (1 + 2a^\dagger a)\sigma_z) \\
 &= \frac{2\hbar g^2}{\omega - \Omega} \left(a^\dagger a + \frac{1}{2}\right) \sigma_z + \text{constant}.
 \end{aligned} \tag{6.177}$$

We can ignore the constant, since it does not affect the dynamics of the system.

The Hamiltonian $\mathcal{H}$ is finally

$$\begin{aligned}
 \tilde{H} &= \mathcal{H}_0 + \frac{1}{2}[S, \mathcal{H}_{\text{int}}] + \dots \\
 &= \hbar\omega \left(a^\dagger a + \frac{1}{2}\right) - \frac{\hbar\Omega}{2}\sigma_z + \frac{\hbar g^2}{\Delta} \left(a^\dagger a + \frac{1}{2}\right) \sigma_z + \dots
 \end{aligned} \tag{6.178}$$

where $\Delta = \omega - \Omega$. If the last term is combined with the first, then the frequency of the resonator will be shifted by $\pm g^2/\Delta$ depending on the state of the qubit. Alternatively, grouping the terms with σ_z instead shows that the frequency of the qubit is shifted by an amount dependent on the the number of photons in the resonator: $(g^2/\Delta)(n + 1/2)$. Since frequency can be controlled and measured very accurately, measuring the frequency of the coupled resonator provides a way to read out the qubit state.

6.5 Exercises

- 6.1** Consider the classical harmonic oscillator Hamiltonian given by (6.33).
- (a) Using Hamilton's equations (6.23) and (6.24), find $\dot{x}$ and $\dot{p}$.
 - (b) Combine the expressions for $\dot{x}$ and $\dot{p}$ to obtain a single second-order differential equation for x .
 - (c) Show that the solution to the differential equation for x is of the form $x(t) = x_0 \exp j\omega_0 t$.
- 6.2** Write a Python program to calculate and plot the first three quantum mechanical harmonic oscillator wave functions.
- 6.3** Invert the pair of equations for a and $a^\dagger$ (6.45) and (6.48) to obtain the expressions for x , p in terms of a , $a^\dagger$, (6.54) and (6.55).

- 6.4** Substitute the operators for Φ, Q given by (6.69) and (6.70) into the Hamiltonian for the LC circuit (6.61) and simplify to obtain the Hamiltonian in terms of raising and lowering operators (6.71).
- 6.5** Explicitly construct the 4×4 matrix representing the Hamilton (6.156).
- (a)** Find the eigenvalues of this Hamiltonian to find the modified energies of the coupled system.
 - (b)** Find the eigenvectors associated with each of the eigenvalues for the case $\omega_1 = \omega_2 \equiv \omega_0 \gg g$.

7

Theory of Superconductivity

When we think of a “good” conductor, we normally think of a metal like copper that is commonly used in electrical circuits. The quality that makes it “good” is a relatively low resistance to the flow of electrical current. It is natural, then to think of a “perfect” conductor as a material that has zero electrical resistance. What then is a “superconductor”? Superconductors do indeed have zero resistance to DC currents, but they have a number of other remarkable properties that distinguish them from a material that simply has zero resistance. These properties appear below a certain critical temperature. Above the critical temperature the behavior is that of an ordinary conducting material. The unusual properties that appear below the critical temperature include:

- Perfect DC conductivity
- Tendency to expel magnetic fields
- A gap for energy absorption, somewhat like a semiconductor
- An effective wave function that maintains coherence over macroscopic dimensions
- Remarkable behavior of weakly-coupled junctions between two superconductors.

A detailed study of these effects and the theory of superconductivity is beyond the scope of our discussion, but we will introduce some fundamental notions about the theory of solids with the goal of giving some insight into the differences between normal conductors—metals in particular—and superconductors, and the origins and descriptions of the properties of superconductors that are most useful for understanding superconducting qubits. A number of texts treat the theory of superconductivity more thoroughly; our discussion draws heavily from Van Duzer and Turner [50] and Tinkham [51].

7.1 Bosons and Fermions

All particles in physics are classified as either *Bosons* or *Fermions*. It will be helpful in the discussions in this chapter to compare and contrast the properties of these two classes of particles. Fermions are particles with half-integer spin, and obey the Pauli-exclusion principle: i.e., no two Fermions can be in the same state at the same time. This has important consequences for the symmetry of the wave function of a multi-Fermion system. For example, suppose the wave function was symmetric such that interchanging two particles left the wave function unchanged. If exchanging the particles does not

change the wave function, then the two particles must be in the same state—which is not allowed by the Pauli-exclusion principle. Consequently we can ensure the Pauli-exclusion principle is satisfied by constructing wave functions that are anti-symmetric with respect to the exchange of any two particles.

At a finite temperature T , the probability that a Fermion state with energy $\mathcal{E}$ is occupied is given by the *Fermi-Dirac Distribution function*:

$$f_{\text{FD}}(\mathcal{E}) = \frac{1}{e^{(\mathcal{E}-\mu)/(k_B T)} + 1}, \quad (7.1)$$

where μ is called the *chemical potential*. If the number of particles N in the system is conserved, then the chemical potential is determined by requiring

$$\sum_s \frac{1}{e^{(\mathcal{E}_s - \mu)/(k_B T)} + 1} = N. \quad (7.2)$$

At low temperatures typical of those encountered in superconductivity, the chemical potential is well-approximated by the *Fermi energy*—the energy of the highest energy state that is occupied at 0 K. The behavior of the Fermi-Dirac distribution function is illustrated in Figure 7.1.

Similar to what we did for the harmonic oscillator in Section 6.3.2, we can also define creation and annihilation operators for Fermions, $c_j^\dagger$ and c_j , respectively. However, unlike the harmonic oscillator creation and annihilation operators, the Fermion operators obey the *anti-commutation relations*:

$$\{c_j, c_k^\dagger\} = c_j c_k^\dagger + c_k^\dagger c_j = \delta_{j,k}. \quad (7.3)$$

The Fermion that we will be most concerned with is the electron.

Bosons are particles with integer spin, and do not obey the Pauli-exclusion principle. As a result, any number of Bosons can be in the same state, and the wave function for a multi-particle Boson system must be symmetric with respect to the interchange of any two particles.

At a finite temperature T , the expected number of Bosons in a state with energy $\mathcal{E}$ is given by the *Bose-Einstein Distribution function*:

$$f_{\text{BE}}(\mathcal{E}) = \frac{1}{e^{(\mathcal{E}-\mu)/(k_B T)} - 1}. \quad (7.4)$$

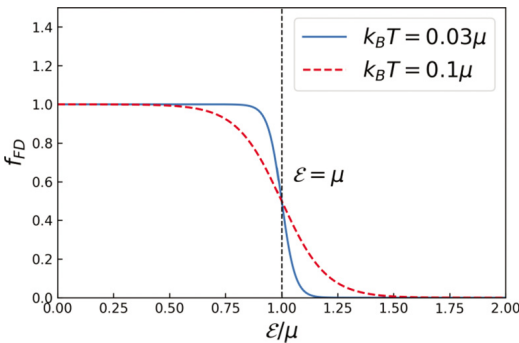
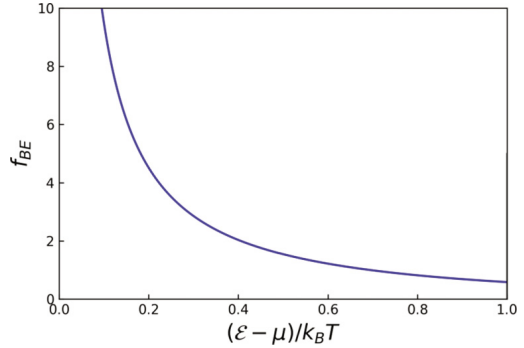


Figure 7.1 The Fermi-Dirac distribution gives the probability that a Fermion state with a given energy is occupied at a specified temperature. The lower the temperature, the sharper the transition from occupied to not-occupied at $\mathcal{E} = \mu$.

Figure 7.2 The Bose-Einstein distribution gives the expected number of Bosons in a state with energy $\mathcal{E}$ at the temperature T . When the number of particles is not conserved (e.g., with photons and phonons), the chemical potential μ is set to zero.



As with the Fermi-Dirac distribution, if the number N of Bosons in the system is fixed, the chemical potential is determined by requiring

$$\sum_s \frac{1}{e^{(\mathcal{E}_s - \mu)/(k_B T)} - 1} = N. \quad (7.5)$$

An example of a case with conserved particles would be a system of ^4He atoms. (^4He has two protons, two neutrons, and two electrons. Each of these has spin $1/2$. In the ground state each pair of spins is anti-parallel, resulting in a net spin of zero. Consequently, ^4He is a *composite Boson*.) If the number of particles is not conserved, for example with photons and phonons, then the chemical potential μ is set to zero. In this case, the Bose-Einstein distribution gives the average number of particles with a particular energy at a given temperature (see Figure 7.2):

$$\frac{1}{e^{\mathcal{E}_s/(k_B T)} - 1} = n_{\mathcal{E},T}. \quad (7.6)$$

The Boson creation and annihilation operators satisfy the commutation relation

$$[a_j, a_k^\dagger] = \delta_{j,k}. \quad (7.7)$$

The quantum resonators discussed in Chapter 6 are examples of Bosonic systems, since any number of photons with energy $\hbar\omega_0$ can be stored in the resonator.

Details of the derivation of the Fermi-Dirac and Bose-Einstein distributions are beyond the scope of our present discussion, but can be found in [52] and other introductory texts in solid state physics and statistical mechanics.

7.2 Bloch Theorem

The metals we encounter in our everyday experience are not perfect crystals; i.e., the atoms in the metal are not organized as a perfect 3-dimensional pattern characteristic of the material. Instead, they are usually composed of randomly-oriented grains of crystals within which the atoms are ordered in the pattern characteristic of the material. Essentially all of the properties of the metal depend on the details of this characteristic order as well as the specific properties of an atom of the metal. It is useful, then, to consider the somewhat idealized case of a perfect crystal. This 3-dimensional pattern is referred to as the *crystal lattice*. Each lattice point has a volume associated with it

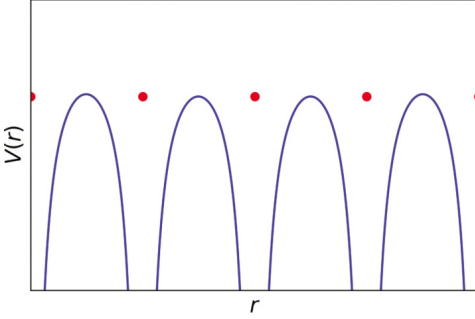


Figure 7.3 Depiction of a periodic potential created by the sum of Coulomb potentials from an array of ions (represented by dots).

called a *primitive cell*. The shape of the primitive cell depends on the type of lattice, but is constructed so that the volume of a sample can be constructed without gaps by stacking primitive cells.

The mobile electrons leave behind positive ions at the lattice sites, and so the electrons are constantly moving in the presence of this periodic potential. A depiction of a periodic potential from ions is shown in Figure 7.3. If electron–electron interactions are neglected, the energy states available to electrons in such a lattice can be obtained by solving Schrödinger’s equation in the presence of the periodic potential from the ions:

$$-\frac{\hbar^2}{2m_e} \nabla^2 \psi + V(\mathbf{r})\psi = \varepsilon\psi. \quad (7.8)$$

Note that $V(\mathbf{r})$ has the same translational symmetry as the lattice; i.e., $V(\mathbf{r} + \mathbf{R}) = V(\mathbf{r})$ where $\mathbf{R}$ is a vector connecting any two ions in the lattice.

According to the *Bloch Theorem*, the solution to Schrödinger’s equation in such a periodic potential is of the form

$$\psi(\mathbf{r}) = U_{\mathbf{k}}(\mathbf{r})e^{i\mathbf{k} \cdot \mathbf{r}} \quad (7.9)$$

where $U_{\mathbf{k}}(\mathbf{r})$ also has the translational symmetry of the lattice: $U_{\mathbf{k}}(\mathbf{r} + \mathbf{R}) = U_{\mathbf{k}}(\mathbf{r})$.

As a plausibility argument for Bloch’s Theorem, consider the following. We can always write $\psi(\mathbf{r})$ in the form

$$\psi(\mathbf{r}) = f(\mathbf{r})U_{\mathbf{k}}(\mathbf{r}) \quad (7.10)$$

where $f(\mathbf{r})$ is a function to be determined. Since the potential is periodic, one would also expect all of the observables to be periodic. In particular

$$\begin{aligned} |\psi(\mathbf{r} + \mathbf{R})|^2 &= |\psi(\mathbf{r})|^2 \\ |f(\mathbf{r} + \mathbf{R})U_{\mathbf{k}}(\mathbf{r} + \mathbf{R})|^2 &= |f(\mathbf{r})U_{\mathbf{k}}(\mathbf{r})|^2 \\ |f(\mathbf{r} + \mathbf{R})|^2 &= |f(\mathbf{r})|^2. \end{aligned} \quad (7.11)$$

The last step follows from the fact that $U_{\mathbf{k}}(\mathbf{r} + \mathbf{R}) = U_{\mathbf{k}}(\mathbf{r})$. It would appear, then, that $|f|^2$ is either periodic in $\mathbf{R}$ or independent of $\mathbf{R}$. However, a function periodic in $\mathbf{R}$ would be better treated by simply redefining $U_{\mathbf{k}}(\mathbf{r})$. A more general possibility is

$$f(\mathbf{r}) = e^{i\mathbf{k} \cdot \mathbf{r}}. \quad (7.12)$$

This clearly satisfies (7.11) since the magnitude squared is unity for any value of $\mathbf{r}$. We conclude that the wave function in the presence of a periodic potential can always be written in the form

$$\psi(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}} U_{\mathbf{k}}(\mathbf{r}). \quad (7.13)$$

Because of this form, ψ has the following important properties:

1. ψ has the form of a traveling plane wave with wave vector $\mathbf{k}$, modulated by $U_{\mathbf{k}}(\mathbf{r})$.
2. Because of the plane wave character, the electron has a de Broglie wavelength $\lambda = 2\pi/k$ and a *crystal momentum* $\mathbf{p} = \hbar\mathbf{k}$.
3. The function ψ represents a state in which the electron is shared throughout the crystal. Such a delocalized state is consistent with the plane wave picture in item 1. It is as if the electron is able to move through the crystal without colliding with or scattering from the ions giving rise to the potential.

7.3 Free Electron Model for Metals

Experimentally, electrons in a metal are found to behave like free particles, except for the fact that they are confined to the finite volume of the sample. Recall the Schrödinger equation for a 1-dimensional free electron:

$$-\frac{\hbar^2}{2m_e} \frac{d^2\psi}{dx^2} = \mathcal{E}\psi. \quad (7.14)$$

It can be easily verified that the solution is of the form

$$\psi(x) \sim e^{ikx}, \quad (7.15)$$

provided

$$\mathcal{E} = \frac{\hbar^2 k^2}{2m_e}. \quad (7.16)$$

Comparison with the classical expression for the energy of a free electron $p^2/(2m_e)$ enables us to identify the momentum as $p = \hbar k$.

7.3.1 Discrete States in Finite Samples

If the particle is confined to a 1-dimensional sample of length L , then we must give some thought to the boundary conditions at the sample edges. One common assumption is the use of periodic¹ boundary conditions, i.e., we impose the condition that $\psi(0) = \psi(L)$:

$$\begin{aligned} \psi(0) &= \psi(L) \\ 1 &= e^{ikL}. \end{aligned} \quad (7.17)$$

This implies $kL = 2\pi n$, $n = 0, \pm 1, \pm 2, \dots$ or

$$k_n = \frac{2\pi n}{L}. \quad (7.18)$$

¹ It turns out that the key insights come from the fact that imposing *any* boundary conditions results in only discrete values of k being allowed, so exactly what boundary conditions we choose is not critical.

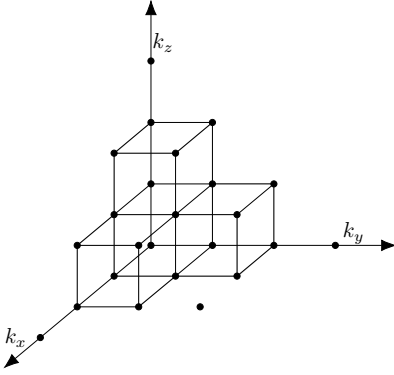


Figure 7.4 Cubic lattice in k -space resulting from periodic boundary conditions.

Following a similar argument, for a 3-dimensional sample the shape of a cube of edge L we require

$$k_x = \frac{2\pi n_x}{L}, \quad k_y = \frac{2\pi n_y}{L}, \quad k_z = \frac{2\pi n_z}{L} \quad (7.19)$$

where n_x, n_y, n_z are integers. The discrete values of $\mathbf{k}$ can be visualized as comprising a cubic lattice of points in “ k -space” as shown in Figure 7.4. Close examination shows that each point is shared by 8 cubes, and there are 8 points associated with each cube, so there is one allowed state for each cube. Alternatively, the volume per state is $(2\pi/L)^3$. According to the Pauli exclusion principle from quantum mechanics, no two electrons can be in the same state. Consequently, if we have N_0 electrons in the sample, we can start with the lowest energy state and then fill the states until we run out of electrons. In doing so, we note that each point in k -space can accommodate two electrons with opposite spin. Owing to the symmetry of the sample, there is no preference for positive or negative values of the components of $\mathbf{k}$. As a result, when we have finished assigning all available electrons to a state, the shape of the volume of filled states in k -space will be approximated by a sphere.

The number of states inside a sphere of radius k is therefore

$$N_0(k) = 2 \frac{\frac{4}{3}\pi k^3}{(2\pi/L)^3} = \frac{8\pi}{3} \left(\frac{L}{2\pi} \right)^3 k^3 \quad (7.20)$$

where we have included spin with an overall factor of 2. It is useful to normalize to the volume of the sample, in which case the number of states per unit volume of real space is

$$N(k) = \frac{8\pi}{3} \left(\frac{k}{2\pi} \right)^3. \quad (7.21)$$

We can estimate the number of states in the sphere of radius $k + dk$ using a Taylor series:

$$N(k + dk) \approx N(k) + \frac{\partial N}{\partial k} dk. \quad (7.22)$$

The number of states in the shell between k and $k + dk$ is therefore

$$\begin{aligned} N(k + dk) - N(k) &= \frac{\partial N}{\partial k} dk \\ &= 8\pi \left(\frac{1}{2\pi} \right)^3 k^2 dk \\ &= \left(\frac{k}{\pi} \right)^2 dk. \end{aligned} \quad (7.23)$$

In the free electron model, a spherical surface of constant $|\mathbf{k}|$ also implies a spherical constant energy surface, in view of (7.16). With this observation we can also write

$$N(\mathcal{E})d\mathcal{E} = \left(\frac{k}{\pi} \right)^2 dk \quad (7.24)$$

where $N(\mathcal{E})$ is the number of states per unit volume between $\mathcal{E}$ and $\mathcal{E} + d\mathcal{E}$ and is called the *density of states*. Using the expression for the energy of a free particle (7.16) the density of states can be expressed explicitly in terms of the energy:

$$N(\mathcal{E}) = \frac{1}{2\pi^2} \left(\frac{2m_e}{\hbar^2} \right)^{3/2} \mathcal{E}^{1/2}. \quad (7.25)$$

Note that $N(\mathcal{E})$ is the volume density of *states* between $\mathcal{E}$ and $\mathcal{E} + d\mathcal{E}$. To obtain the number of *electrons* in this range, we must multiply by the probability that a state of energy $\mathcal{E}$ is occupied:

$$dn(\mathcal{E}) = f(\mathcal{E})N(\mathcal{E})d\mathcal{E}, \quad (7.26)$$

where $f(\mathcal{E})$ is the Fermi-Dirac distribution function (7.1).² To find the number of electrons per unit volume with energy less than $\mathcal{E}$, we simply integrate:

$$n(\mathcal{E}) = \int_0^{\mathcal{E}} f(\mathcal{E}')N(\mathcal{E}')d\mathcal{E}'. \quad (7.27)$$

The chemical potential is a weak function of temperature and is determined by integrating over all energies and setting the integral equal to the density of available electrons. At $T = 0$ the chemical potential is equal to the *Fermi energy* $\mathcal{E}_F$ defined as the energy of the highest occupied states at $T = 0$. For our discussion of superconductors, the temperatures will generally be sufficiently low that $\mu \approx \mathcal{E}_F$ to a good approximation. At $T = 0$, $f(\mathcal{E}) = 1$ for $\mathcal{E} < \mathcal{E}_F$ and is zero otherwise, so the Fermi level is related to the total number of electrons per unit volume by

$$n = \int_0^{\mathcal{E}_F} N(\mathcal{E}')d\mathcal{E}'. \quad (7.28)$$

Substituting the density of states (7.25) and performing the integration gives

$$\mathcal{E}_F = \frac{\hbar^2}{2m_e} (3\pi^2 n)^{2/3}. \quad (7.29)$$

Comparison with the free electron energy (7.16) gives immediately

$$k_F = (3\pi^2 n)^{1/3}. \quad (7.30)$$

² Here we omit the subscript FD for clarity.

The velocity at the Fermi surface is given by $p/m_e = \hbar k_F/m_e$. As an example, the conduction band electron density of aluminum is $18.1 \times 10^{28} \text{ m}^{-3}$, leading to a Fermi velocity of about $2 \times 10^6 \text{ m/s}$ and a Fermi energy³ of $1.87 \times 10^{-18} \text{ J}$, or 11.7 eV.

7.3.2 Phonons

Interactions between electrons and lattice vibrations are important for understanding conductivity in both normal and superconducting metals.

Consider a simple 1-dimensional model of a lattice consisting of masses connected by springs as shown in Figure 7.5. In equilibrium, the interatomic spacing is a . However, if the atoms are displaced, the springs provide restoring forces. The force on the m th atom is

$$F_m = \underbrace{\beta(u_{m+1} - u_m)}_{\text{force from } m+1} - \underbrace{\beta(u_m - u_{m-1})}_{\text{force from } m-1} \quad (7.31)$$

$$= \beta(u_{m+1} + u_{m-1}) - 2\beta u_m.$$

Equating this to the force from Newton's law gives the difference equation

$$M \frac{d^2 u_m}{dt^2} = \beta(u_{m+1} + u_{m-1}) - 2\beta u_m. \quad (7.32)$$

Let us seek a plane wave solution of the form

$$u_m = A e^{iqx_m - i\omega t} \quad (7.33)$$

where $x_m = ma$ is the equilibrium position of the n th atom. Substituting and simplifying gives

$$\begin{aligned} -M\omega^2 A e^{iqma} &= \beta A (e^{iq(m+1)a} + e^{iq(m-1)a} - 2e^{iqma}) \\ -M\omega^2 &= \beta (e^{iqa} + e^{-iqa} - 2) \\ &= 2\beta (\cos qa - 1). \end{aligned} \quad (7.34)$$

Using the identity $\cos \theta - 1 = -2 \sin^2 \theta/2$ and solving for ω gives

$$\omega_q = \sqrt{\frac{4\beta}{M}} |\sin(qa/2)|. \quad (7.35)$$

Note that this relation is periodic with period $2\pi/a$. All of the unique values are represented in the range $-\pi/a < q \leq \pi/2$, and this range is referred to as the first *Brillouin*

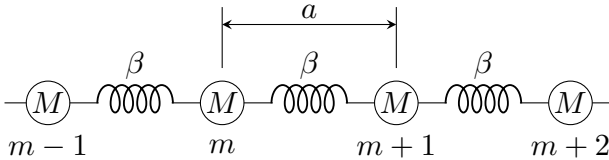


Figure 7.5 Spring-mass model for phonons on a 1-D lattice.

³ Energy in electron volts (eV) is obtained by dividing the energy in Joules by the electronic charge e .

zone. Similar to our discussion of the free electron confined to a sample, suppose the linear chain has a length $L = Ka$, where $K + 1$ is the number of atoms in the chain. Imposing period boundary conditions as before leads to the quantized values of q :

$$q_m = \frac{2\pi m}{Ka}. \quad (7.36)$$

For definiteness, consider the case when K is even. In this case the range of m in the first Brillouin zone is $-K/2 < m \leq K/2$, giving a total of K unique values.

In the modes that we have considered, the displacements of adjacent atoms are similar. Modes of this type are referred to as *acoustic modes*. When there are two or more types of atom per primitive cell, there are also modes in which the displacements of adjacent atoms are in opposite directions. These are called *optical modes*. The term “optical mode” is appropriate for two reasons. First, the branch occurs in the infrared for most materials. Second, if the material is polar, the motion will create alternating electric dipoles that couple strongly to electromagnetic waves at the same frequency (i.e., infrared light).

Note that there are other modes of vibration in addition to the longitudinal vibration modes that we have considered. The vibrations could be transverse to the chain, either in the plane of the paper, or perpendicular to the plane. Since the restoring forces will be different for the transverse modes, there will generally be three branches to the dispersion relation: one longitudinal and two transverse.

Classically, the masses can vibrate with any amplitude. However, as we saw in Section 6.3.2, when considered as a quantum mechanical harmonic oscillator the vibrational energy can take on only discrete values. These quantized modes are called *phonons*. The energy of the quantized system is given by

$$\mathcal{E}_q = \hbar\omega_q (n_q + 1/2) \quad (7.37)$$

where n_q is the number of excited phonons with wave number q .

7.3.3 Debye Model

Many calculations involving phonons at low temperature can be performed satisfactorily using the *Debye model*. In this model, the temperature is considered to be sufficiently low that the excitation of the optical branches is negligible. The first Brillouin zone is replaced with a sphere containing the same number of modes, and no distinction is made between the acoustic branches. If there are N_c primitive cells in the specimen, then there are N_c values of q . The radius of the equivalent sphere is therefore determined by

$$N_c = \left(\frac{L}{2\pi}\right)^3 \frac{4}{3}\pi q_D^3 \quad (7.38)$$

or

$$q_D = (6\pi^2 N)^{1/3} \quad (7.39)$$

where q_D is the *Debye wave number* and N is the number of primitive cells per unit volume.

The Debye model also assumes a simple linear relation between ω and q . This enables us to define the *Debye frequency* as

$$\omega_D = v_s q_D = v_s (6\pi^2 N)^{1/3}, \quad (7.40)$$

where v_s is the speed of sound. The Debye frequency can be thought of as the highest possible phonon frequency. We also define the *Debye temperature* θ_D as $k_B \theta_D = \hbar \omega_D$.

7.3.4 Electron–Phonon Scattering and Electrical Conductivity

Recall from the Bloch theorem that electrons can move without collisions with the atoms in the lattice if the placement of the atoms is perfectly periodic. At finite temperature, however, there will be vibrations of the atoms away from their equilibrium positions. These vibrations interrupt the perfect periodicity, and electron collisions with the atoms in the lattice are possible.⁴

Since the energy of the lattice excitations is quantized, the interaction between electrons and phonons can be modeled as collisions between particles, as depicted in Figure 7.6. Single scattering events must satisfy conservation of both energy and momentum. The conservation equations for scattering by phonon absorption and emission are given in Table 7.1.

A free electron in an electric field would experience a constant force resulting in a constant acceleration, from Newton's law. This constant acceleration would result in a linear increase of velocity with time. However, if the electron periodically scatters from lattice vibrations, there will be time intervals in which the velocity increases punctuated by scattering events that suddenly reduce the component of velocity parallel to the electric field. The resulting average velocity is independent of time, but proportional to

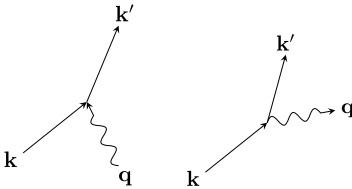


Figure 7.6 Graphical depiction of electron scattering by absorption of a phonon (left), and by emission of a phonon (right).

Table 7.1 Energy and momentum conservation in electron-phonon single scattering events.

Quantity conserved	Phonon absorption	Phonon emission
Energy	$\frac{\hbar^2 k^2}{2m_e} + \hbar \omega_q = \frac{\hbar^2 k'^2}{2m_e}$	$\frac{\hbar^2 k^2}{2m_e} = \hbar \omega_q + \frac{\hbar^2 k'^2}{2m_e}$
Momentum	$\hbar \mathbf{k} + \hbar \mathbf{q} = \hbar \mathbf{k}'$	$\hbar \mathbf{k} = \hbar \mathbf{q} + \hbar \mathbf{k}'$

⁴ Scattering can also occur from defects in the lattice, independent of temperature.

the strength of the electric field. This average velocity is also referred to as the *drift velocity*.

To be specific, consider the motion of an electron in the presence of a constant electric field along the x direction. The behavior of the average velocity can be modeled by the classical equation of motion

$$m_e \frac{d\langle v_x \rangle}{dt} + m_e \frac{\langle v_x \rangle}{\tau} = -eE_x. \quad (7.41)$$

The second term represents damping caused by scattering of the electrons. The scattering is characterized by the *momentum relaxation time* τ . At low temperatures ($T \ll \theta_D$), only phonons with small values of q are excited, while the scattered electrons will be near the Fermi surface with relatively large momenta $k \approx k_F$. Consequently, at low temperatures, only small angle scattering is possible and many collisions are required to significantly alter the momentum. If the average time between collisions is τ_c , we conclude that $\tau_c \ll \tau$.

For static fields, the steady-state is obtained by setting the time derivative to zero. This gives

$$m_e \frac{\langle v_x \rangle}{\tau} = -eE_x \quad (7.42)$$

or

$$\langle v_x \rangle = -\frac{e\tau}{m_e} E_x. \quad (7.43)$$

To get the current density, we multiply by the charge and the density of electrons:

$$J_x = -ne \langle v_x \rangle = \frac{ne^2\tau}{m_e} E_x \equiv \sigma E_x \quad (7.44)$$

where σ is the electrical conductivity in the free electron model

$$\sigma = \frac{ne^2\tau}{m_e}. \quad (7.45)$$

We can interpret this result in either of two equivalent ways. The most straightforward interpretation is that *each* electron is moving with an average velocity of $\langle v_x \rangle$ so that the total current is given by (7.44). Alternatively, consider the following picture, as illustrated in Figure 7.7. When $E_x = 0$, the velocities of the electrons in the “Fermi sea” cancel in pairs so that no net current exists (left part of Figure 7.7). Now consider what happens when an electric field is applied. According to our previous arguments, each electron now acquires a drift velocity $\langle v_x \rangle$. Consequently, if $\mathbf{E}$ is along the $+\hat{x}$ direction, the Fermi sphere is displaced along the $-\hat{x}$ direction as depicted in the right portion of Figure 7.7. Although the velocities of most of the electrons still cancel in pairs, the small number in the shaded crescent now contribute to conduction. Using this picture we can estimate the fraction of total electrons that are uncompensated as $\langle v_x \rangle / v_F$. Since these uncompensated electrons are very near the Fermi surface, their velocities are approximately $-v_F$. The current density can then be estimated as

$$J_x \approx -en \left| \frac{\langle v_x \rangle}{v_F} \right| (-v_F) = en |\langle v_x \rangle|. \quad (7.46)$$

In this interpretation, we can view the conduction as only involving electrons near the Fermi surface.

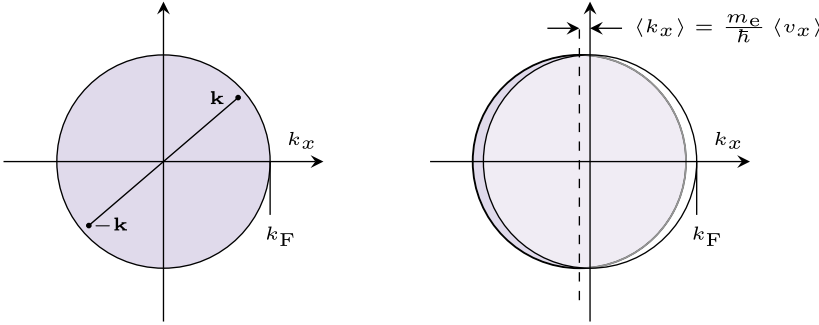


Figure 7.7 Visualizing conduction as the displacement of the Fermi sea. Left: in the absence of an electric field, the k -vectors cancel in pairs. Right: the application of an electric field shifts the Fermi sphere, resulting in a small sliver of volume near the Fermi surface in which the k -vectors are not canceled.

7.3.5 Perfect Conductor vs. Superconductor

Let's return to the equation of motion for the electrons (7.41), but this time consider a time-varying electric field $E_x = E_0 e^{j\omega t}$. Substituting this into (7.41) and solving for $\langle v_x \rangle$ gives

$$\langle v_x \rangle = \frac{-(e\tau/m_e)}{1 + j\omega\tau} E_x. \quad (7.47)$$

For a perfect conductor there is no scattering, so we let $\tau \rightarrow \infty$. In this limit the current density is

$$J_x = -en \langle v_x \rangle = -j \frac{e^2 n}{\omega m_e} E_x. \quad (7.48)$$

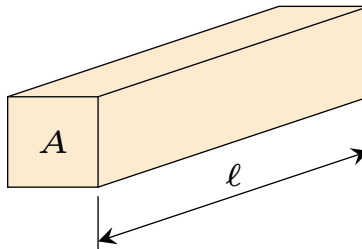
We note two interesting things:

- although the electric field must vanish at DC ($\omega = 0$) for finite J_x , a time-varying electric field is possible inside of a perfect conductor
- a perfect conductor exhibits an inductance resulting from the inertia of the electrons.

To explore this second observation a bit more, consider a rectangular perfectly conducting wire as shown in Figure 7.8. Recognizing that the current is $I = J_x A$ and voltage is $V = E_x \ell$, the relation between the voltage and the current is

$$I = -j \frac{e^2 n A}{\omega m_e \ell} V. \quad (7.49)$$

Figure 7.8 A rectangular wire used to relate the voltage to the current in normal and perfect conductors.



Comparing this to the relation between current and voltage in an inductor $I = V/(j\omega L)$, we identify a *kinetic inductance* given by

$$L_k = \frac{m_e \ell}{e^2 n A}. \quad (7.50)$$

This kinetic inductance is in addition to the magnetic inductance resulting from magnetic fields linking the current. Superconductors also exhibit a kinetic inductance, except that the electrons form pairs so that the charge and mass of the particles are twice those of the electron.

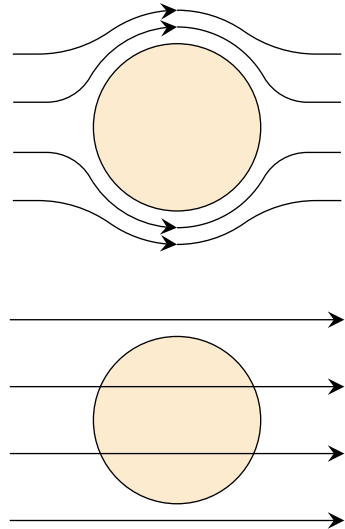
Next let's consider the magnetic properties of a perfect conductor. From Maxwell's equation from Faraday's law

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}. \quad (7.51)$$

If the electric field is zero, then $\partial \mathbf{B} / \partial t = 0$ and the magnetic field $\mathbf{B}$ must be constant. We conclude that in a perfect conductor the magnetic field cannot *change*, but it does not have to be zero. Suppose that as we cooled the metal there was a transition from a normal conductor to a perfect conductor. If we applied a magnetic field, then cooled it so that it became a perfect conductor, the magnetic field would remain constant—it would be “frozen in.”

Experimentally, however, Meissner and Ochsenfeld[53] found that the magnetic field in a superconductor is not only constant—it must be zero. Instead of the field being frozen in, the lines of magnetic flux are expelled from the sample (see Figure 7.9). We conclude that a superconductor must be fundamentally different from a perfect conductor.

Figure 7.9 A superconductor expels magnetic flux (top), while static magnetic flux can penetrate a perfect conductor (bottom).



7.4 Bardeen, Cooper, and Schrieffer Theory of Superconductivity

A microscopic theory of superconductivity was presented by Bardeen, Cooper, and Schrieffer[54] that has had significant quantitative success, despite being based on a very simplified free electron model without any reference to the underlying crystal structure. Central to this theory is the formation of bound states consisting of electron pairs. Since particles with the same charge would normally repel each other, the ability of electrons to form bound states is remarkable. The key ingredient to creating a net attractive interaction turns out to be scattering between electrons and phonons. We begin our discussion by first assuming that an attractive potential exists, and showing that this leads to bound pair states. We then consider how the electron–phonon interaction can give rise to a net attractive potential leading to electron binding in pairs. Next we use this foundation to consider the superconducting ground state, and discuss how this leads to the ability to describe the superconductor with a macroscopic effective wave function.

7.4.1 Cooper Pair Model

We now consider a simple model first proposed by Cooper[55] that shows that pairing of electrons as just discussed can actually lower the energy of the system. Consider a filled Fermi sea with all states filled for $k \leq k_F$. Two electrons are now added with $k > k_F$. We assume that there is an interaction between the electrons that can be described by a potential energy term in the form $\mathcal{V}(\mathbf{r}_1 - \mathbf{r}_2)$. The Hamiltonian of the added pair can be written

$$\mathcal{H} = \mathcal{H}_0 + \mathcal{V}(\mathbf{r}_1 - \mathbf{r}_2), \quad (7.52)$$

where $\mathcal{H}_0$ is the two particle Hamiltonian in the absence of interactions, using the Fermi energy $\mathcal{E}_F$ as the reference energy. The two-particle wave function $\psi(\mathbf{r}_1 - \mathbf{r}_2)$ must, of course, satisfy the Schrödinger equation

$$\mathcal{H}\psi = \mathcal{E}_p\psi \quad (7.53)$$

where $\mathcal{E}_p$ is the energy of the two-electron state.

To construct a reasonable guess for ψ , we note that there are two parts to the wave function—one that describes the spatial distribution, and one that describes the spin orientations of the electrons. We can use what we know about the required symmetry of the wave function along with the fact that we are particularly interested in the lowest energy state to see if a bound state is formed to arrive at forms for the two parts of the wave function. First, let's consider the spatial part of the wave function. Since we are looking for a bound state, it is reasonable to look for a state with zero net momentum, since net momentum would be associated with increased kinetic energy. If we approximate the non-interacting electron wave functions as plane waves, this suggests that a key building block would be terms of the form $\exp(i\mathbf{k} \cdot \mathbf{r}_1) \exp(-i\mathbf{k} \cdot \mathbf{r}_1) = \exp(i\mathbf{k} \cdot (\mathbf{r}_1 - \mathbf{r}_2))$, or electrons with opposite momenta. We could construct even or odd combinations of these building blocks—e.g., $\cos(\mathbf{k} \cdot (\mathbf{r}_1 - \mathbf{r}_2))$ or $\sin(\mathbf{k} \cdot (\mathbf{r}_1 - \mathbf{r}_2))$. However, anticipating that we are looking for bound states with the electrons attracted to one

another, the cosine seems to be the best choice, since it suggests a large probability amplitude for finding the electrons close together.

Turning now to the spin part of the wave function (sometimes referred to as a *spinor*), we know from Section 7.1 that the two-electron wave function must be anti-symmetric with respect to the interchange of the particles, so if the spatial part of the wave function is even, we should choose an odd form for the spinor portion. Putting these observations together leads to a trial wave function of the form

$$\psi(\mathbf{r}_1 - \mathbf{r}_2) = \left[\sum_{k > k_F} g_{\mathbf{k}} \cos(\mathbf{k} \cdot (\mathbf{r}_1 - \mathbf{r}_2)) \right] (\alpha_1 \beta_2 - \beta_1 \alpha_2), \quad (7.54)$$

where α_i and β_i represent the spin-up and spin-down states, respectively, for electron i . A key take-away from these arguments is that the most likely combinations of electronic states to lead to a bound pair are electrons with opposite momenta, and opposite spin. Having made this observation, we will concentrate on the spatial part of the wave function and omit the spinor part in the remainder of our discussion. Further, it will be mathematically convenient to proceed with a somewhat more general and compact form for the spatial part of the wave function:

$$\begin{aligned} \psi(\mathbf{r}_1 - \mathbf{r}_2) &= \sum_{k' > k_F} a_{\mathbf{k}'} e^{i\mathbf{k}' \cdot (\mathbf{r}_1 - \mathbf{r}_2)} \\ &= \sum_{k' > k_F} a_{\mathbf{k}'} e^{i\mathbf{k}' \cdot \mathbf{r}}. \end{aligned} \quad (7.55)$$

Here we have also made the substitution $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$. Note that this form is slightly more general than (7.54) in that it no longer explicitly imposes even symmetry in $\mathbf{r}$, but it does preserve the choice of electrons with opposite momenta. Substituting this new form for the wave function into the Schrödinger equation (7.53) leads to

$$(\mathcal{E}_p - \mathcal{H}_0) \sum_{k' > k_F} a_{\mathbf{k}'} e^{i\mathbf{k}' \cdot \mathbf{r}} = \mathcal{V}(\mathbf{r}) \sum_{k' > k_F} a_{\mathbf{k}'} e^{i\mathbf{k}' \cdot \mathbf{r}}. \quad (7.56)$$

To find the Fourier coefficients, we multiply both sides by $\exp(-i\mathbf{k} \cdot \mathbf{r})$ and integrate over all space. Let's first consider the right-hand-side of (7.56):

$$\begin{aligned} \frac{1}{\Omega} \int_{\Omega} d^3r e^{-i\mathbf{k} \cdot \mathbf{r}} \mathcal{V}(\mathbf{r}) \sum_{k' > k_F} a_{\mathbf{k}'} e^{i\mathbf{k}' \cdot \mathbf{r}} &= \sum_{k' > k_F} a_{\mathbf{k}'} \frac{1}{\Omega} \int_{\Omega} d^3r e^{-i\mathbf{k} \cdot \mathbf{r}} \mathcal{V}(\mathbf{r}) e^{i\mathbf{k}' \cdot \mathbf{r}} \\ &= \sum_{k' > k_F} a_{\mathbf{k}'} \mathcal{V}_{\mathbf{k}, \mathbf{k}'}, \end{aligned} \quad (7.57)$$

where Ω is the volume of the sample and $\mathcal{V}_{\mathbf{k}, \mathbf{k}'}$ is given by

$$\begin{aligned} \mathcal{V}_{\mathbf{k}, \mathbf{k}'} &= \frac{1}{\Omega} \int_{\Omega} d^3r e^{-i\mathbf{k} \cdot \mathbf{r}} \mathcal{V}(\mathbf{r}) e^{i\mathbf{k}' \cdot \mathbf{r}} \\ &\equiv \langle \mathbf{k}, -\mathbf{k} | \mathcal{V} | \mathbf{k}', -\mathbf{k}' \rangle. \end{aligned} \quad (7.58)$$

$\mathcal{V}_{\mathbf{k}, \mathbf{k}'}$ represents the probability of making a transition between the states $|\mathbf{k}', -\mathbf{k}'\rangle$ and $|\mathbf{k}, -\mathbf{k}\rangle$ because of scattering induced by the potential $\mathcal{V}$.

Next, consider the left-hand-side of (7.56):

$$\begin{aligned}
& \frac{1}{\Omega} \int_{\Omega} d^3r e^{-i\mathbf{k}\cdot\mathbf{r}} (\mathcal{E}_p - \mathcal{H}_0) \sum_{k' > k_F} a_{\mathbf{k}'} e^{i\mathbf{k}'\cdot\mathbf{r}} \\
&= \mathcal{E}_p \sum_{k' > k_F} a_{\mathbf{k}'} \frac{1}{\Omega} \int_{\Omega} d^3r e^{i(\mathbf{k}' - \mathbf{k})\cdot\mathbf{r}} \\
&\quad - \frac{1}{\Omega} \int_{\Omega} d^3r e^{-i\mathbf{k}\cdot\mathbf{r}} \sum_{k' > k_F} a_{\mathbf{k}'} \mathcal{H}_0 e^{i\mathbf{k}'\cdot\mathbf{r}} \\
&= \mathcal{E}_p a_{\mathbf{k}} - \sum_{k' > k_F} a_{\mathbf{k}'} \frac{1}{\Omega} \int_{\Omega} d^3r e^{-i\mathbf{k}\cdot\mathbf{r}} \mathcal{H}_0 e^{i\mathbf{k}'\cdot\mathbf{r}} \\
&= (\mathcal{E}_p - 2\varepsilon_{\mathbf{k}}) a_{\mathbf{k}}.
\end{aligned} \tag{7.59}$$

Here we have made use of the fact that

$$\frac{1}{\Omega} \int_{\Omega} d^3r e^{i(\mathbf{k}' - \mathbf{k})\cdot\mathbf{r}} = \delta_{\mathbf{k}', \mathbf{k}} \tag{7.60}$$

as well as recognizing that

$$\frac{1}{\Omega} \int_{\Omega} d^3r e^{-i\mathbf{k}\cdot\mathbf{r}} \mathcal{H}_0 e^{i\mathbf{k}'\cdot\mathbf{r}} = \langle \mathbf{k}, -\mathbf{k} | \mathcal{H}_0 | \mathbf{k}', -\mathbf{k}' \rangle = 2\varepsilon_{\mathbf{k}} \delta_{\mathbf{k}, \mathbf{k}'} \tag{7.61}$$

where $\varepsilon_{\mathbf{k}}$ is the energy of a single free particle. In other words, the expected value of the two-particle Hamiltonian without interactions is simply twice the energy of a single free particle.

Putting together the results from integrating over the left and right sides of (7.56) gives

$$a_{\mathbf{k}} = \frac{1}{\mathcal{E}_p - 2\varepsilon_{\mathbf{k}}} \sum_{k' > k_F} a_{\mathbf{k}'} \mathcal{V}_{\mathbf{k}, \mathbf{k}'}. \tag{7.62}$$

We now make an important simplifying assumption: we take $\mathcal{V}_{\mathbf{k}, \mathbf{k}'} = -\mathcal{V}$ (with $\mathcal{V}$ positive) when both $\hbar^2 k^2/(2m)$ and $\hbar^2 (k')^2/(2m)$ lie within the energy shell between $\mathcal{E}_F$ and $\mathcal{E}_F + \hbar\omega_D$, and is zero otherwise. In other words, we assume that only electrons near the Fermi surface scatter with phonons (since the available empty states are just above the Fermi surface), and the maximum energy that can be gained from phonon scattering corresponds to the maximum phonon frequency, approximated by the energy at the Debye frequency $\hbar\omega_D$. The expression for the Fourier amplitudes then becomes

$$a_{\mathbf{k}} = \frac{-\mathcal{V}}{\mathcal{E}_p - 2\varepsilon_{\mathbf{k}}} \sum_{k' > k_F} a_{\mathbf{k}'}. \tag{7.63}$$

Summing both sides allows us to cancel terms as follows:

$$\begin{aligned}
\sum_{k > k_F} a_{\mathbf{k}} &= \sum_{k > k_F} \frac{-\mathcal{V}}{\mathcal{E}_p - 2\varepsilon_{\mathbf{k}}} \sum_{k' > k_F} a_{\mathbf{k}'} \\
1 &= -\mathcal{V} \sum_{k > k_F} \frac{1}{\mathcal{E}_p - 2\varepsilon_{\mathbf{k}}}.
\end{aligned} \tag{7.64}$$

The sum over $\mathbf{k}$ can be replaced by an integral over the corresponding range of energies provided we add a factor of the density of states $N(\varepsilon)$:

$$1 = -\mathcal{V} \int_0^{\hbar\omega_D} \frac{N(\varepsilon)}{\mathcal{E}_F - 2\varepsilon} d\varepsilon. \quad (7.65)$$

Because the Debye energy is small compared to the energy at the Fermi surface, we can approximate the density of states as constant and equal to that at the Fermi surface. Since we have taken our zero of energy at the Fermi surface, we represent this constant as $N(0)$. This simplification allows us to evaluate the integral:

$$\begin{aligned} 1 &= -\mathcal{V}N(0) \int_0^{\hbar\omega_D} \frac{1}{\mathcal{E}_F - 2\varepsilon} d\varepsilon \\ &= -\frac{\mathcal{V}N(0)}{2} \ln(\mathcal{E}_F - 2\varepsilon) \Big|_0^{\hbar\omega_D} \\ &= -\frac{\mathcal{V}N(0)}{2} \ln\left(1 - \frac{2\hbar\omega_D}{\mathcal{E}_F}\right). \end{aligned} \quad (7.66)$$

Solving this for the pair energy gives

$$\mathcal{E}_F = \frac{2\hbar\omega_D}{1 - e^{2/\mathcal{V}N(0)}}. \quad (7.67)$$

In most superconductors it is found that the interaction is weak so that $\mathcal{V}N(0) \ll 1$, in which case the energy is given approximately by

$$\mathcal{E}_F = -2\hbar\omega_D e^{-2/\mathcal{V}N(0)}. \quad (7.68)$$

The fact that this is negative indicates that the increase in kinetic energy associated with scattering into states slightly above the Fermi surface is more than compensated by the decrease in potential energy associated with the attractive interaction. The Fermi surface is therefore unstable; slight thermal excitations of electrons to states just above the Fermi surface will lead to reductions in energy as the excited electrons form bound pair states. Presumably this process would continue until a new equilibrium state is reached in which the formation of additional bound pairs no longer reduces the total energy. We will return to what this ground state looks like after considering the origin of the attractive interaction between the electrons.

7.4.2 Dielectric Function

When an electric field is applied to a material, charges are redistributed and/or induced in such a way that the electric field generated by the new charge distribution opposes the applied electric field. As a result, the effect of the applied field is diminished or “screened.” The *dielectric function*, or *permittivity*, describes the screened response of the material. As an example, let us take the interaction energy between the Cooper pair electrons to be the Coulomb interaction

$$\mathcal{V}(\mathbf{r}_1 - \mathbf{r}_2) = \frac{e^2}{4\pi\epsilon|\mathbf{r}_1 - \mathbf{r}_2|}. \quad (7.69)$$

For two isolated electrons in a vacuum, the dielectric function is simply the permittivity of free space, i.e., $\epsilon = \epsilon_0$. However, in a physical material, the electrons and ions can

redistribute themselves in such a way as to screen the two electrons from one another. In this case, $\epsilon > \epsilon_0$. However, depending on the dynamics of the particles in the material, the dielectric function can be less than that of free space, or even negative. The case of a negative dielectric function is of particular interest to us, since this would turn the repulsive Coulomb potential (7.69) into an attractive potential—just what we need to form Cooper pairs.

We therefore need to develop a model for the dielectric function. To do this it will be helpful to obtain a relation between the dielectric function and the imposed and induced charges. Let us assume that there is no net charge in the medium in the absence of an imposed perturbation. From Gauss' law we have

$$\nabla \cdot \mathbf{D} = \delta\rho \quad (7.70)$$

where $\delta\rho$ is the imposed charge fluctuation. However, we also know that $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}$. Substituting this expression for $\mathbf{D}$ gives

$$\nabla \cdot \mathbf{E} = \frac{\rho_T}{\epsilon_0} = \frac{\delta\rho + \rho_s}{\epsilon_0} \quad (7.71)$$

where the total charge density ρ_T is equal to the imposed charge density $\delta\rho$ plus the screening response of the medium $\rho_s = -\nabla \cdot \mathbf{P}$.

Since we are interested in phonon scattering, let us consider the case where $\delta\rho$ and ρ_s are proportional to $\exp(i\mathbf{q} \cdot \mathbf{r})$:

$$i\mathbf{q} \cdot \mathbf{E} = \frac{\delta\rho + \rho_s}{\epsilon_0}. \quad (7.72)$$

We define the dielectric function $\epsilon(\omega, q)$ by the constitutive relation $\mathbf{D} = \epsilon \mathbf{E}$. This enables us to write

$$\frac{i\mathbf{q} \cdot \mathbf{D}}{\epsilon} = \frac{\delta\rho + \rho_s}{\epsilon_0}. \quad (7.73)$$

However, $i\mathbf{q} \cdot \mathbf{D} = \delta\rho$ from (7.70). Making this substitution and solving for ϵ gives the relation we are looking for between the imposed and induced charge densities and the dielectric function:

$$\epsilon = \epsilon_0 \frac{\delta\rho}{\delta\rho + \rho_s}. \quad (7.74)$$

The screening charge density ρ_s is composed of contributions from both the movement of the electrons as well as the background ions

$$\rho_s = \rho_e + \rho_i. \quad (7.75)$$

In the next section we will examine simple models for both contributions.

7.4.3 Jellium

“Jellium” is a simple material model that gives insight into important features of metals. In this model, the solid is viewed much like a plasma. The electrons form a negatively-charged gas that interpenetrates a positively-charged gas of ions. The electron gas is just the same as in our free electron model. On the other hand, there is no periodicity in the ion gas so the jellium model is just the continuum model for longitudinal phonon

modes in the lattice. Transverse phonon modes do not exist in a gas and are neglected in the jellium model.

Recall that the electron density is related to the Fermi energy by

$$n = \frac{1}{3\pi^2} \left(\frac{2m_e}{\hbar^2} \right)^{3/2} \mathcal{E}_F^{3/2}. \quad (7.76)$$

In the presence of a charge perturbation, there will also be a potential perturbation given by Poisson's equation:

$$\nabla^2 \phi = -\frac{\rho_T}{\epsilon_0}, \quad (7.77)$$

with $\rho_T = \delta\rho + \rho_s$ representing the sum of the charge perturbation and the induced screening charge. This potential perturbation will also cause local fluctuations in the electron density:

$$\begin{aligned} n + \delta n &= \frac{1}{3\pi^2} \left(\frac{2m_e}{\hbar^2} \right)^{3/2} (\mathcal{E}_F + e\phi)^{3/2} \\ &= \frac{1}{3\pi^2} \left(\frac{2m_e}{\hbar^2} \right)^{3/2} \mathcal{E}_F^{3/2} \left(1 + \frac{e\phi}{\mathcal{E}_F} \right)^{3/2} \\ &\approx n \left(1 + \frac{3e\phi}{2\mathcal{E}_F} \right). \end{aligned} \quad (7.78)$$

It follows that the perturbation in electron density is

$$\delta n \approx \frac{3ne}{2\mathcal{E}_F} \phi. \quad (7.79)$$

The induced screening charge from the electrons is therefore

$$\rho_e = -e\delta n = -\frac{3ne^2}{2\mathcal{E}_F} \phi. \quad (7.80)$$

We are interested in particular in perturbations caused by lattice vibrations, or phonons. With this in mind, let $\phi \sim \exp(i\mathbf{q} \cdot \mathbf{r})$. Poisson's equation becomes

$$-q^2 \phi = -\frac{\rho_T}{\epsilon_0} \quad (7.81)$$

or

$$\phi = \frac{\rho_T}{q^2 \epsilon_0}. \quad (7.82)$$

Substituting this expression for ϕ into (7.80) enables us to express the induced electronic charge as

$$\rho_e = -\frac{k_s^2}{q^2} (\delta\rho + \rho_s), \quad (7.83)$$

where

$$k_s^2 = \frac{3ne^2}{2\epsilon_0 \mathcal{E}_F}. \quad (7.84)$$

We can also define a length λ_s such that

$$k_s = \frac{2\pi}{\lambda_s}. \quad (7.85)$$

The length λ_s is called the *Thomas-Fermi scattering length*.

We now need an expression for the induced screening charge from the ions. The induced movement of the ions is much smaller owing to their much larger mass, so we use a simple model of the dynamics based on Newton's law

$$M \frac{d\mathbf{v}_i}{dt} = Ze\mathbf{E} \quad (7.86)$$

where M and $\mathbf{v}_i$ are the mass and velocity of an ion, respectively; Z is the atomic number; and the force on a single charge from an imposed electric field is $e\mathbf{E}$.

The current density resulting from the motion of the ions is

$$\mathbf{J}_i = n_i Z e \mathbf{v}_i = n e \mathbf{v}_i. \quad (7.87)$$

Here we have noted that because the jellium model is electrically neutral, we know that $n = n_i Z$. Solving this expression for $\mathbf{v}_i$ and substituting into (7.86) gives

$$M \frac{d\mathbf{J}_i}{dt} = -Zne^2 \nabla \phi, \quad (7.88)$$

where we have also used the relation between the potential and the electric field $\mathbf{E} = -\nabla \phi$.

Note that from the chain rule we have

$$\frac{dJ_x(\mathbf{r}, t)}{dt} = \frac{\partial J_x}{\partial t} + \frac{\partial J_x}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial J_x}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial J_x}{\partial z} \frac{\partial z}{\partial t}. \quad (7.89)$$

The analogous equations for the three components of the current density can be combined into the single vector equation

$$\begin{aligned} \frac{d\mathbf{J}_i}{dt} &= \frac{\partial \mathbf{J}_i}{\partial t} + \left(\frac{\partial \mathbf{r}}{\partial t} \cdot \nabla \right) \mathbf{J}_i \\ &= \frac{\partial \mathbf{J}_i}{\partial t} + (\mathbf{v}_i \cdot \nabla) \mathbf{J}_i \\ &= -i\omega \left(1 - \frac{\mathbf{v}_i \cdot \mathbf{q}}{\omega} \right) \mathbf{J}_i \end{aligned} \quad (7.90)$$

where in the last step we have taken $\mathbf{J}_i \sim \exp(i\mathbf{q} \cdot \mathbf{r} - i\omega t)$. If the motion of the ions is sufficiently small so that $v_i \ll \omega/q$, then

$$\frac{d\mathbf{J}_i}{dt} \approx \frac{\partial \mathbf{J}_i}{\partial t}. \quad (7.91)$$

We can then write (7.88) as

$$M \frac{\partial \mathbf{J}_i}{\partial t} = -Zne^2 \nabla \phi. \quad (7.92)$$

The current density can now be related to the charge density using the continuity equation

$$\frac{\partial \rho_i}{\partial t} = -\nabla \cdot \mathbf{J}_i. \quad (7.93)$$

Taking the partial derivative with respect to time on both sides, assuming time and position dependence of the form $\exp(i\mathbf{q} \cdot \mathbf{r} - i\omega t)$, and combining with (7.92) leads to

$$\rho_i = \frac{\omega_{\text{Pi}}^2}{\omega^2} (\delta\rho + \rho_s), \quad (7.94)$$

where ω_{Pi} is the ion plasma frequency given by

$$\omega_{\text{Pi}}^2 = \frac{Zne^2}{\epsilon_0 M}. \quad (7.95)$$

Noting that $\rho_s = \rho_e + \rho_i$, from (7.83) and (7.94) we now have two equations in the two unknowns ρ_e and ρ_i , expressed in terms of the imposed charge perturbation $\delta\rho$. However, we would really like the total screening charge ρ_s to substitute into (7.74) to find the dielectric function. We can obtain this more directly by adding (7.83) and (7.94) and solving for ρ_s :

$$\rho_e + \rho_i = \rho_s = (\rho_s + \delta\rho) \frac{\omega_{\text{Pi}}^2 q^2 - \omega^2 k_s^2}{\omega^2 q^2}. \quad (7.96)$$

Solving for ρ_s gives

$$\rho_s = \frac{\omega_{\text{Pi}}^2 q^2 - k_s^2 \omega^2}{\omega^2 (k_s^2 + q^2) - q^2 \omega_{\text{Pi}}^2} \delta\rho. \quad (7.97)$$

Substituting this result into (7.74) gives the dielectric function

$$\epsilon(\omega, q) = \epsilon_0 \frac{\omega^2 (k_s^2 + q^2) - q^2 \omega_{\text{Pi}}^2}{\omega^2 q^2}. \quad (7.98)$$

It will be useful to express this in terms of the natural vibration frequency of the ions in the presence of the electron screening. We can obtain this frequency by setting $\delta\rho = 0$ in (7.96) and solving for ω^2 . The result is

$$\omega^2 = \frac{\omega_{\text{Pi}}^2 q^2}{k_s^2 + q^2} \equiv \omega_q^2, \quad (7.99)$$

where ω_q is the screened natural frequency for an excitation with wave number q . Using this expression to eliminate ω_{Pi}^2 from (7.98) gives the following form for the dielectric function:

$$\epsilon(\omega, q) = \epsilon_0 (k_s^2 + q^2) \frac{\omega^2 - \omega_q^2}{\omega^2 q^2}. \quad (7.100)$$

7.4.4 Scattering Amplitude and Attractive Electron–Electron Interaction

Armed with a model for the dielectric function, let us return to the discussion of Section 7.4.1 in which we discussed a scattering amplitude for making the transition from state $\mathbf{k}'$ to state $\mathbf{k}$. From (7.58) and using our screened Coulomb potential, the scattering amplitude is

$$\begin{aligned} \mathcal{V}_{\mathbf{k}, \mathbf{k}'} &= \frac{1}{\Omega} \int_{\Omega} d^3 r e^{-i\mathbf{k} \cdot \mathbf{r}} \mathcal{V}(\mathbf{r}) e^{i\mathbf{k}' \cdot \mathbf{r}} \\ &= \frac{1}{\Omega} \int_{\Omega} d^3 r \frac{e^2}{4\pi\epsilon r} e^{i(\mathbf{k}' - \mathbf{k}) \cdot \mathbf{r}} \\ &= \frac{e^2}{4\pi\epsilon} \frac{1}{\Omega} \int_{\Omega} d^3 r \frac{1}{r} e^{i\mathbf{q} \cdot \mathbf{r}} \\ &= \frac{e^2}{\Omega\epsilon q^2}. \end{aligned} \quad (7.101)$$

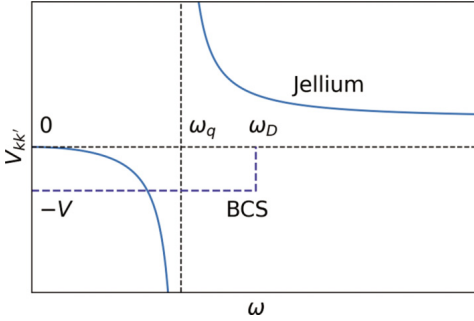


Figure 7.10 The potential is negative (attractive) for $\omega < \omega_q$ in the Jellium model. The BCS theory simply assumes that the potential is a negative constant up to the maximum phonon frequency ω_D .

For the details of evaluating the integral in the last step, see Exercise 7.5. For simplicity, in the following we will simplify our notation by taking a unit volume $\Omega = 1$ without loss of generality. Substituting explicitly for ϵ gives

$$\mathcal{V}_{\mathbf{k},\mathbf{k}'} = \frac{e^2 \omega^2}{\epsilon_0(k_s^2 + q^2)(\omega^2 - \omega_q^2)}. \quad (7.102)$$

This shows that an attractive (i.e., negative) potential is possible for $\omega < \omega_q$. In the Debye model, the maximum phonon frequency is ω_D . This suggests a simple approximation that is used in the BCS theory: take the amplitude $\mathcal{V}_{\mathbf{k},\mathbf{k}'}$ to be a negative constant out to ω_D , as suggested in Figure 7.10.

Note that $\mathcal{V}_{\mathbf{k},\mathbf{k}'}$ can also be written as

$$\begin{aligned} \mathcal{V}_{\mathbf{k},\mathbf{k}'} &= \frac{e^2 \omega^2 + e^2 \omega_q^2 - e^2 \omega_q^2}{\epsilon_0(k_s^2 + q^2)(\omega^2 - \omega_q^2)} \\ &= \frac{e^2}{\epsilon_0(k_s^2 + q^2)} + \frac{e^2 \omega_q^2}{\epsilon_0(k_s^2 + q^2)(\omega^2 - \omega_q^2)}. \end{aligned} \quad (7.103)$$

The first term is just the screened Coulomb interaction, while the second term describes the interaction with the lattice that gives rise to an attractive potential for $\omega < \omega_q$.

7.4.5 Interpretation of Attractive Interaction

The origin of the attractive electron–electron interaction can be pictured in the following way. Imagine for a moment the artificial picture of a stationary electron among the ions. The ions in the lattice will be attracted toward the electron creating a local increase in the positive charge that opposes the charge of the electron. As a result, the electronic charge is partially shielded and other electrons do not see the entire bare charge. However, in reality the electron is moving very rapidly (on the order of the Fermi velocity) while the ions respond relatively slowly because of their larger mass. Consequently, by the time the ions have fully responded, the electron has left the region! As a result, the delayed ion displacement creates a local increase in positive charge that can then attract another electron. This interaction can be viewed as electrons interacting by the emission and absorption of a phonon, as depicted diagrammatically in Figure 7.11. Figure 7.12 suggests how electrons with opposite momenta are particularly subject to this effective mutual attraction.

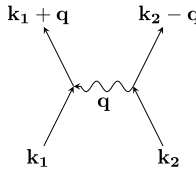


Figure 7.11 Diagrammatic representation of two electrons interacting via the emission and absorption of a phonon.

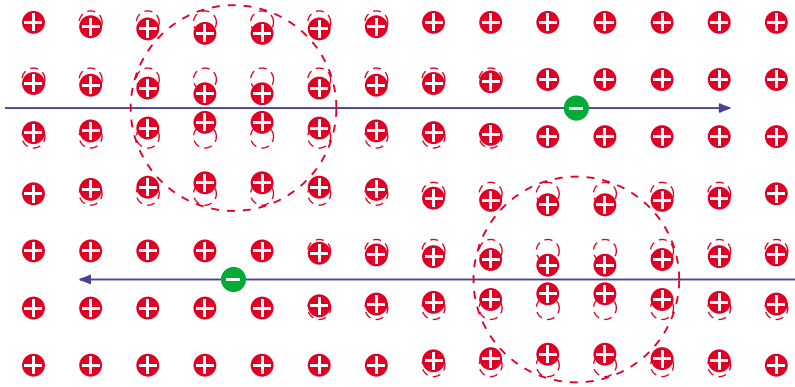


Figure 7.12 The delayed response to the presence of an electron creates a local concentration of positive charge that can attract another electron.

7.4.6 Superconductor Hamiltonian

We now have a sense of what the form of the superconducting Hamiltonian should be: the kinetic energy associated with the electrons in the Fermi sea must be combined with an attractive potential interaction between electrons with opposite momenta and opposite spins. To construct such a Hamiltonian, let us introduce appropriate Fermion creation and annihilation operators similar to those described in Section 7.1. In particular, we define $c_{\mathbf{k}\uparrow}^*$ as a creation operator⁵ for an electron with spin up in state $\mathbf{k}$, and $c_{\mathbf{k}\uparrow}$ as an annihilation operator for an electron with spin up in state $\mathbf{k}$. If we denote the state with no particles as $|\phi_0\rangle$ and the state with one spin up particle in state $\mathbf{k}$ as $|1_{\mathbf{k}\uparrow}\rangle$, then

$$\begin{aligned} c_{\mathbf{k}\uparrow}^* |\phi_0\rangle &= |1_{\mathbf{k}\uparrow}\rangle, \\ c_{\mathbf{k}\uparrow} |1_{\mathbf{k}\uparrow}\rangle &= |\phi_0\rangle, \\ c_{\mathbf{k}\uparrow} |\phi_0\rangle &= 0, \\ c_{\mathbf{k}\uparrow}^* |1_{\mathbf{k}\uparrow}\rangle &= 0. \end{aligned} \tag{7.104}$$

The last two relations follow from the facts that we cannot remove a particle from a state that is already empty, and two Fermions cannot occupy the same state. The electron–phonon–electron interaction depicted in Figure 7.11 can then be represented as

$$c_{(\mathbf{k}_2-\mathbf{q})\downarrow}^* c_{(\mathbf{k}_1+\mathbf{q})\uparrow}^* c_{\mathbf{k}_2\downarrow} c_{\mathbf{k}_1\uparrow},$$

⁵ In the superconductivity literature, creation operators are usually denoted by an asterisk instead of a dagger. We adopt the asterisk notation here to agree with the common notation in the literature.

or, with $\mathbf{k}_2 = -\mathbf{k}_1 = -\mathbf{k}$:

$$c_{-(\mathbf{k}+\mathbf{q})\downarrow}^* c_{(\mathbf{k}+\mathbf{q})\uparrow}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow}.$$

The potential energy associated with this interaction is therefore represented by the operator

$$\mathcal{V} = \sum_{\mathbf{k}, \mathbf{k}'} \mathcal{V}_{\mathbf{k}, \mathbf{k}'} c_{-\mathbf{k}'\downarrow}^* c_{\mathbf{k}'\uparrow}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow} \quad (7.105)$$

where $\mathbf{k}' = \mathbf{k} + \mathbf{q}$.

The kinetic energy of a pair is $2\varepsilon_{\mathbf{k}}$, where $\varepsilon_{\mathbf{k}}$ is the energy measured with respect to the Fermi surface:

$$\varepsilon_{\mathbf{k}} = \frac{\hbar^2 k^2}{2m_e} - \mathcal{E}_F. \quad (7.106)$$

Note that $c_{-\mathbf{k}\downarrow}^* c_{\mathbf{k}\uparrow}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow} |\psi\rangle = |\psi\rangle$ if the state $|\psi\rangle$ contains a particle pair corresponding to $\mathbf{k}$ and is zero otherwise. Consequently, we define the pair number operator as

$$\hat{n}_{\mathbf{k}} = c_{-\mathbf{k}\downarrow}^* c_{\mathbf{k}\uparrow}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow}. \quad (7.107)$$

Combining the kinetic and potential energy expressions gives the Hamiltonian used in the BCS theory:

$$\mathcal{H}_{\text{BCS}} = 2 \sum_{\mathbf{k}} \varepsilon_{\mathbf{k}} \hat{n}_{\mathbf{k}} + \sum_{\mathbf{k}, \mathbf{k}'} \mathcal{V}_{\mathbf{k}, \mathbf{k}'} c_{-\mathbf{k}'\downarrow}^* c_{\mathbf{k}'\uparrow}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow}. \quad (7.108)$$

7.4.7 Superconducting Ground State

We would now like to find the ground state and the ground state energy of the BCS Hamiltonian (7.108). The approach we will take is to construct a trial wave function with adjustable parameters, and vary these parameters to minimize the expectation value of the BCS Hamiltonian. Ideally, the trial wave function would have features similar to what we expect for the true wave function. An educated guess of this sort is sometimes referred to as an *ansatz*.

To construct the ground state ansatz used in the BCS theory, we start with the vacuum state, or state with no particles, and apply operators that create particle pairs with opposite spin and opposite momenta. The state with one Cooper pair is constructed as

$$\left[u_{\mathbf{k}} + v_{\mathbf{k}} c_{-\mathbf{k}\downarrow}^* c_{\mathbf{k}\uparrow}^* \right] |\phi_0\rangle$$

where $u_{\mathbf{k}}$ is the probability amplitude that the pair state is empty, and $v_{\mathbf{k}}$ is the probability amplitude that the pair state is occupied. It follows that $|u_{\mathbf{k}}|^2 + |v_{\mathbf{k}}|^2 = 1$. Similarly, the state with two Cooper pairs is

$$\left[u_{\mathbf{k}'} + v_{\mathbf{k}'} c_{-\mathbf{k}'\downarrow}^* c_{\mathbf{k}'\uparrow}^* \right] \left[u_{\mathbf{k}} + v_{\mathbf{k}} c_{-\mathbf{k}\downarrow}^* c_{\mathbf{k}\uparrow}^* \right] |\phi_0\rangle.$$

Continuing this process leads to ground state

$$|\psi\rangle = \prod_{\mathbf{k}} \left[u_{\mathbf{k}} + v_{\mathbf{k}} c_{-\mathbf{k}\downarrow}^* c_{\mathbf{k}\uparrow}^* \right] |\phi_0\rangle. \quad (7.109)$$

We want to vary $v_{\mathbf{k}}$ in the trial ground state so as to minimize $\langle \psi | \mathcal{H}_{\text{BCS}} | \psi \rangle$. This will give us the best estimate for the ground state energy. The $|v_{\mathbf{k}}|^2$ will then give us the ground state pair occupancy.

Let us begin by considering the expectation of the first term (the kinetic energy term) in (7.108). For simplicity, we will consider a single term in the sum. Recognizing that the pair number operator will only be non-zero when the pair state $\mathbf{k}$ is occupied, it is sufficient to construct a simplified trial state with only this pair. With these assumptions we can calculate the expectation as follows:

$$\begin{aligned}
 \langle \psi | \hat{n}_{\mathbf{k}} | \psi \rangle &= \langle \psi | \hat{n}_{\mathbf{k}} [u_{\mathbf{k}} + v_{\mathbf{k}} c_{-\mathbf{k}\downarrow}^* c_{\mathbf{k}\uparrow}^*] | \phi_0 \rangle \\
 &= \langle \psi | \hat{n}_{\mathbf{k}} [u_{\mathbf{k}} | \phi_0 \rangle + v_{\mathbf{k}} | 1_{-\mathbf{k}}, 1_{\mathbf{k}} \rangle] \\
 &= \langle \psi | v_{\mathbf{k}} | 1_{-\mathbf{k}}, 1_{\mathbf{k}} \rangle \\
 &= \langle \phi_0 | [u_{\mathbf{k}}^* + v_{\mathbf{k}}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow}] v_{\mathbf{k}} | 1_{-\mathbf{k}}, 1_{\mathbf{k}} \rangle \\
 &= |v_{\mathbf{k}}|^2 \langle \phi_0 | \phi_0 \rangle \\
 &= |v_{\mathbf{k}}|^2
 \end{aligned} \tag{7.110}$$

where we have used the orthogonality property $\langle \phi_0 | 1_{-\mathbf{k}}, 1_{\mathbf{k}} \rangle = 0$ and the normalization $\langle \phi_0 | \phi_0 \rangle = 1$. This is a very reasonable result, stating that the expectation value of the pair number operator for state $\mathbf{k}$ is simply the probability that the state is occupied.

Following a similar argument, let us consider a single term in the second sum (the potential energy) in (7.108). Since a single term only involves pairs at $\mathbf{k}$ and $\mathbf{k}'$, it suffices to consider a two pair state:

$$\langle \psi | c_{-\mathbf{k}'\downarrow}^* c_{\mathbf{k}'\uparrow}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow} | \psi \rangle = \langle \psi_{\mathbf{k}'\mathbf{k}} | c_{-\mathbf{k}'\downarrow}^* c_{\mathbf{k}'\uparrow}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow} | \psi_{\mathbf{k}'\mathbf{k}} \rangle \tag{7.111}$$

where

$$|\psi_{\mathbf{k}'\mathbf{k}} \rangle = [u_{\mathbf{k}'} + v_{\mathbf{k}'} c_{-\mathbf{k}'\downarrow}^* c_{\mathbf{k}'\uparrow}^*] [u_{\mathbf{k}} + v_{\mathbf{k}} c_{-\mathbf{k}\downarrow}^* c_{\mathbf{k}\uparrow}^*] | \phi_0 \rangle. \tag{7.112}$$

Completing this calculation using a similar approach to that used for the kinetic energy term leads to

$$\langle \psi | c_{-\mathbf{k}'\downarrow}^* c_{\mathbf{k}'\uparrow}^* c_{-\mathbf{k}\downarrow} c_{\mathbf{k}\uparrow} | \psi \rangle = u_{\mathbf{k}}^* u_{\mathbf{k}'}^* u_{\mathbf{k}'} v_{\mathbf{k}}. \tag{7.113}$$

The interpretation of this term is also intuitive. The scattering event takes the pair at $(\mathbf{k}, -\mathbf{k})$ and scatters it into the state $(\mathbf{k}', -\mathbf{k}')$. Thus the initial state is characterized by pair $\mathbf{k}$ being occupied, and the pair $\mathbf{k}'$ being empty. The probability amplitude of this state is $u_{\mathbf{k}} v_{\mathbf{k}}$. Similarly, the term $u_{\mathbf{k}}^* u_{\mathbf{k}'}^*$ is the conjugate of the probability amplitude of the final state with the pair $\mathbf{k}'$ occupied and $\mathbf{k}$ being empty.

Summing terms for all $\mathbf{k}$ leads to the expectation value of the BCS Hamiltonian

$$W = \langle \psi | \mathcal{H}_{\text{BCS}} | \psi \rangle = 2 \sum_{\mathbf{k}} \varepsilon_{\mathbf{k}} |v_{\mathbf{k}}|^2 + \sum_{\mathbf{k}, \mathbf{k}'} \mathcal{V}_{\mathbf{k}, \mathbf{k}'} u_{\mathbf{k}}^* u_{\mathbf{k}'}^* u_{\mathbf{k}'} v_{\mathbf{k}}. \tag{7.114}$$

Similar to our discussion about Cooper pairs in Section 7.4.1, the BCS theory simplifies $\mathcal{V}_{\mathbf{k}, \mathbf{k}'}$ as

$$\mathcal{V}_{\mathbf{k}, \mathbf{k}'} = \begin{cases} -\mathcal{V} & \text{if } |\varepsilon_{\mathbf{k}}| \text{ and } |\varepsilon_{\mathbf{k}'}| \leq \hbar\omega_{\text{D}} \\ 0 & \text{otherwise} \end{cases} \tag{7.115}$$

where $\mathcal{V}$ is a positive constant.

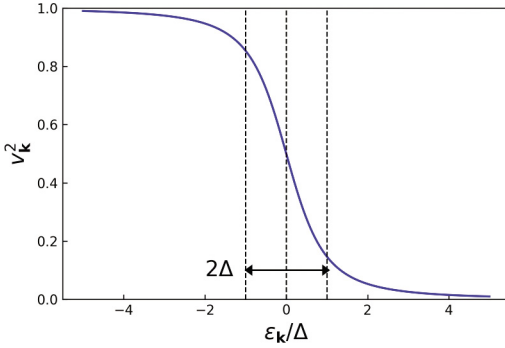


Figure 7.13 The probability that a pair state is occupied as a function of energy near the Fermi surface. The width of the transition is approximately 2Δ .

The next step is to minimize W by varying $v_{\mathbf{k}}$ subject to the constraint $|u_{\mathbf{k}}|^2 + |v_{\mathbf{k}}|^2 = 1$. The details of this calculation are beyond the scope of our discussion, but can be found in Tinkham [51] and Van Duzer and Turner [50]. The result is found to be

$$v_{\mathbf{k}}^2 = \frac{1}{2} \left[1 - \frac{\epsilon_{\mathbf{k}}}{\mathcal{E}_{\mathbf{k}}} \right], \quad (7.116)$$

where

$$\mathcal{E}_{\mathbf{k}} = \sqrt{\epsilon_{\mathbf{k}}^2 + \Delta^2} \quad (7.117)$$

is the energy of an electron that has broken free from a bound pair, referred to as a *quasi-particle*, and

$$\Delta \approx 2\hbar\omega_{\text{D}} e^{-1/\mathcal{V}N(0)}. \quad (7.118)$$

The behavior of $v_{\mathbf{k}}^2$ is shown in Figure 7.13.

For an electron at the Fermi surface, $\epsilon_{\mathbf{k}} = 0$, so $\mathcal{E}_{\mathbf{k}} = \Delta$. In other words, the minimum amount of energy a quasi-particle can have after escaping a Cooper pair is Δ . However, to break a Cooper pair requires an energy of 2Δ , since there are two electrons. Thus there is an *energy gap* of 2Δ that must be overcome to break a Cooper pair.

The previous analysis is valid at a temperature of 0 K. The model can be extended to finite temperature by incorporating the Fermi-Dirac distribution function to describe the probability that energy states are occupied. The resulting calculation shows that the superconducting gap Δ decreases as the temperature increases, and goes to zero at a critical temperature T_{c} . The expression for the critical temperature from the BCS theory is

$$k_{\text{B}}T_{\text{c}} \approx 1.13\hbar\omega_{\text{D}} e^{-1/\mathcal{V}N(0)}. \quad (7.119)$$

One of the remarkable results of the BCS theory is that the ratio of the energy gap to the critical temperature is a constant, independent of model parameters such as $\mathcal{V}$ and ω_{D} :

$$\frac{2\Delta(0)}{k_{\text{B}}T_{\text{c}}} \approx 3.5. \quad (7.120)$$

This result is compared with the experimental values for several technologically important materials in Table 7.2. The fact that this result is independent of the precise details

Table 7.2 Comparison of $2\Delta(0)/k_B T_c$ with the BCS theory prediction for some technologically important superconductors.

Element	$2\Delta(0)/k_B T_c$
BCS	3.5
Al	3.3
Nb	3.8
Ta	3.6

of the model for the potential and phonon interactions helps to justify the simple approximations associated with the jellium model.

7.5 Electrodynamics of Superconductors

7.5.1 Cooper Pairs and the Macroscopic Wave Function

Recall that the two-particle wave function could be written (see (7.55))

$$\psi = \sum_{\mathbf{k}} a_{\mathbf{k}} e^{i\mathbf{k} \cdot (\mathbf{r}_1 - \mathbf{r}_2)} \quad (7.121)$$

where the sum over $\mathbf{k}$ includes only values in a range Δk about k_F . The range Δk can be estimated from the associated range of energies $\Delta \mathcal{E}$ using

$$\mathcal{E} = \frac{\hbar^2 k^2}{2m_e}.$$

For $\Delta \mathcal{E}$ near the Fermi surface, we have

$$\Delta \mathcal{E} = \frac{\hbar^2 k_F}{m_e} \Delta k = \hbar v_F \Delta k \quad (7.122)$$

where v_F is the velocity at the Fermi surface. We can then estimate the size of the pair using the uncertainty relation

$$\Delta r \Delta k \approx 1$$

or

$$\Delta r \approx \frac{1}{\Delta k} \approx \frac{\hbar v_F}{\Delta \mathcal{E}}. \quad (7.123)$$

If the states making up a pair state are within a range $2\Delta(0)$ about the Fermi energy, then from (7.120)

$$\Delta r \sim \frac{\hbar v_F}{3.5 k_B T_c}. \quad (7.124)$$

The size of a Cooper pair is also referred to as the *coherence length* ξ_0 . The actual value from the BCS theory is

$$\Delta r \sim \xi_0 = 0.18 \frac{\hbar v_F}{k_B T_c}. \quad (7.125)$$

As an example, consider aluminum, which has a critical temperature of 1.18 K and a Fermi velocity of 2.03×10^6 m/s in the free electron theory [56]. The size estimate is then

$$\Delta r \approx \xi_0 = 2.4 \mu\text{m}. \quad (7.126)$$

Since the range of k 's is centered about k_F , there are rapid oscillations within this wave packet with period of order

$$\lambda_F = \frac{2\pi}{k_F} = \frac{2\pi\hbar}{m_e v_F} \approx 3.6 \times 10^{-4} \mu\text{m}.$$

Taking Δr to be the diameter of a spherical volume associated with a Cooper pair, the volume is found to be of order $6.8 \times 10^{-18} \text{ m}^3$. The electron density in aluminum is $n_e = 18.1 \times 10^{28} \text{ m}^{-3}$, so the volume associated with a pair is of order $2/n_e = 1.1 \times 10^{-29} \text{ m}^3$. We conclude that a Cooper pair overlaps with billions of other Cooper pairs. Under these circumstances it is energetically favorable for the pairs to “lock phases.” In the absence of net pair momentum, the fluid of Cooper pairs can be represented by an ensemble average function

$$\psi = |\psi(\mathbf{r})|. \quad (7.127)$$

This is not an actual wave function, but rather it represents an ensemble average. If net pair momentum does exist, then the ensemble average takes the form

$$\psi = |\psi(\mathbf{r})| e^{i\mathbf{K} \cdot \mathbf{r}}, \quad (7.128)$$

or more generally

$$\psi = |\psi(\mathbf{r})| e^{i\theta(\mathbf{r})}. \quad (7.129)$$

A fundamental assumption in our treatment of the electrodynamics of superconductors is that the superconducting fluid can be represented by such a macroscopic many body wave function with a well-defined amplitude and phase.

7.5.2 Potential Functions

It is possible to define auxiliary functions Φ and $\mathbf{A}$ from which the electromagnetic fields $\mathbf{E}$ and $\mathbf{B}$ can be obtained:

$$\mathbf{E} = -\nabla\Phi - \frac{\partial\mathbf{A}}{\partial t} \quad (7.130)$$

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (7.131)$$

where Φ is the electric scalar potential and $\mathbf{A}$ is the magnetic vector potential. Note that some computational economy is apparent since $\mathbf{E}, \mathbf{B}$ (6 scalar components) can be derived from $\Phi, \mathbf{A}$ (4 scalar components). These functions are constructed to satisfy two of Maxwell's equations:

$$\nabla \cdot \mathbf{B} = \nabla \cdot (\nabla \times \mathbf{A}) = 0, \quad (7.132)$$

and

$$\begin{aligned}\nabla \times \mathbf{E} &= -\nabla \times (\nabla \Phi) - \frac{\partial}{\partial t} \nabla \times \mathbf{A} \\ \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t}.\end{aligned}\tag{7.133}$$

Here we have used the vector identities $\nabla \cdot (\nabla \times \mathbf{Q}) = 0$ and $\nabla \times (\nabla \phi) = 0$, where ϕ and $\mathbf{Q}$ are arbitrary differentiable scalar and vector functions, respectively.

By requiring that the other two Maxwell equations be satisfied we obtain differential equations for $\Phi, \mathbf{A}$. Assuming the scalar constitutive relations $\mathbf{D} = \epsilon \mathbf{E}$ and $\mathbf{B} = \mu \mathbf{H}$ we have

$$\begin{aligned}\nabla \cdot \mathbf{D} &= \rho \\ \nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon} \\ -\left[\nabla \cdot \nabla \Phi + \frac{\partial}{\partial t} (\nabla \cdot \mathbf{A}) \right] &= \frac{\rho}{\epsilon} \\ \nabla^2 \Phi + \frac{\partial}{\partial t} (\nabla \cdot \mathbf{A}) &= \frac{\rho}{\epsilon}\end{aligned}\tag{7.134}$$

and

$$\begin{aligned}\nabla \times \mathbf{H} &= \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J} \\ \nabla \times \mathbf{B} &= \mu \epsilon \frac{\partial \mathbf{E}}{\partial t} + \mu \mathbf{J} \\ \nabla \times (\nabla \times \mathbf{A}) &= \mu \epsilon \left[-\nabla \frac{\partial \Phi}{\partial t} - \frac{\partial^2 \mathbf{A}}{\partial t^2} \right] + \mu \mathbf{J} \\ \nabla^2 \mathbf{A} - \nabla \left[\nabla \cdot \mathbf{A} + \mu \epsilon \frac{\partial \Phi}{\partial t} \right] - \mu \epsilon \frac{\partial^2 \mathbf{A}}{\partial t^2} &= -\mu \mathbf{J}\end{aligned}\tag{7.135}$$

where in the last line we have used the vector identity $\nabla \times (\nabla \times \mathbf{A}) = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A}$.

So far we have only specified the *curl* of $\mathbf{A}$. However, according to the Helmholtz theorem, both the *divergence* and the *curl* must be specified to uniquely define a vector field. Two common choices are the *Coulomb gauge* and the *Lorentz gauge*.

The Coulomb gauge is $\nabla \cdot \mathbf{A} = 0$. This is particularly useful for static problems where time derivatives vanish. In this case the two differential equations become decoupled.

The Lorentz gauge is $\nabla \cdot \mathbf{A} = -\mu \epsilon \partial \Phi / \partial t$. This choice decouples the equations even in the presence of time-varying fields.

We will return to the discussion of the choice of gauge for superconductors after the next section.

7.5.3 London Equations

Recall from Section 6.2 that the canonical momentum for a particle in an electromagnetic field is $\mathbf{p} = m\mathbf{v} + e\mathbf{A}$. It follows that the momentum for a Cooper pair is

$$\mathbf{p} = m^* \mathbf{v}_s + e^* \mathbf{A}\tag{7.136}$$

where $m^* = 2m_e$ and $e^* = -2e$. If the pair density is n_s^* and all pairs have the same momentum, then the momentum density is $n_s^* \mathbf{p}$. To make the transition to a quantum description, we make the identification

$$n_s^* \mathbf{p} = \langle \psi | \frac{\hbar}{i} \nabla | \psi \rangle \quad (7.137)$$

where $\psi(\mathbf{r}) = \sqrt{n_s^*(\mathbf{r})} \exp i\theta(\mathbf{r})$ is the pair-fluid wave function. If $n_s^*(\mathbf{r})$ is constant, then

$$\frac{\hbar}{i} \nabla \psi = \hbar \nabla \theta \psi. \quad (7.138)$$

It follows that⁶

$$\langle \psi | \frac{\hbar}{i} \nabla | \psi \rangle = \hbar \nabla \theta \langle \psi | \psi \rangle = n_s^* \hbar \nabla \theta \quad (7.139)$$

and

$$\mathbf{p} = \hbar \nabla \theta = m^* \mathbf{v}_s + e^* \mathbf{A}. \quad (7.140)$$

The pair current density is

$$\mathbf{J}_s = n_s^* e^* \mathbf{v}_s. \quad (7.141)$$

Solving for $\mathbf{v}_s$ and substituting into the expression for $\mathbf{p}$ gives

$$\mathbf{p} = \hbar \nabla \theta = e^* \Lambda \mathbf{J}_s + e^* \mathbf{A} \quad (7.142)$$

where

$$\Lambda = \frac{m^*}{n_s^* (e^*)^2}. \quad (7.143)$$

Taking the curl of (7.142) gives

$$\nabla \times \mathbf{p} = \hbar \nabla \times \nabla \theta = e^* \Lambda \nabla \times \mathbf{J}_s + e^* \nabla \times \mathbf{A}. \quad (7.144)$$

However, $\nabla \times \nabla \theta \equiv 0$, so we have

$$\Lambda \nabla \times \mathbf{J}_s + \mathbf{B} = 0. \quad (7.145)$$

This is known as the *second London equation*. We will see in Section 7.5.5 that when combined with Maxwell's equation from Faraday's law, this equation implies that fields decay exponentially from a surface into a superconductor.

⁶ Note that here the wave function has been normalized so that $\langle \psi | \psi \rangle = n_s^*$ instead of the usual case $\langle \psi | \psi \rangle = 1$.

Taking the time derivative of (7.145) gives

$$\Lambda \nabla \times \frac{\partial \mathbf{J}_s}{\partial t} = -\frac{\partial \mathbf{B}}{\partial t}. \quad (7.146)$$

Comparing this to Maxwell's equation from Faraday's law

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (7.147)$$

suggests that

$$\Lambda \frac{\partial \mathbf{J}_s}{\partial t} = \mathbf{E}. \quad (7.148)$$

This is known as the *first London equation*. It states that particles accelerate when a field is applied, as opposed to sustaining a current against resistance as is the case in normal conductors.

7.5.4 London Gauge

F. London argued that $\langle \mathbf{p} \rangle = 0$ for the ground state of a superconductor with no magnetic fields applied. He then pointed out that if the wave function was unchanged to lowest order when a field was applied, then $\langle \mathbf{p} \rangle = 0$ would still be true. This “rigidity” of the state can be thought of as resulting from the energy gap; i.e., application of sufficiently small fields cannot change the state of the system. Setting $\mathbf{p} = 0$ in (7.142) gives

$$\mathbf{J}_s = -\frac{1}{\Lambda} \mathbf{A}. \quad (7.149)$$

Since this depends on $\mathbf{A}$ directly, it can only be true if the gauge of $\mathbf{A}$ is specified properly. To do this, consider the simply-connected superconductor shown in Figure 7.14. Integrating the second London equation over a closed surface S just inside the superconductor gives

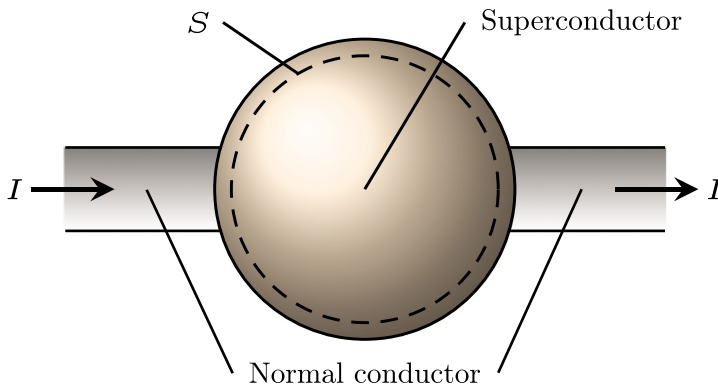


Figure 7.14 Geometry used to understand the London gauge.

$$\int_S \mathbf{J}_s \cdot d\mathbf{s} = -\frac{1}{\Lambda} \int_S \mathbf{A} \cdot d\mathbf{s}. \quad (7.150)$$

If the current leaving the superconductor is the same as that entering so that there is no net charge accumulation, then

$$\int_S \mathbf{J}_s \cdot d\mathbf{s} = \int_S \mathbf{A} \cdot d\mathbf{s} = 0. \quad (7.151)$$

Using the divergence theorem to relate the integral over a closed surface to one over the enclosed volume, we have

$$\int_S \mathbf{A} \cdot d\mathbf{s} = \int_V \nabla \cdot \mathbf{A} dv = 0. \quad (7.152)$$

If this is true for *any* volume inside the superconductor, then it must be true that

$$\nabla \cdot \mathbf{A} = 0. \quad (7.153)$$

This along with (7.149) is known as the *London gauge*. It is also generally true that inside a superconductor $\Phi = 0$.

7.5.5 Penetration Depth

For static fields, we have from Maxwell's equation from Ampere's law:

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}_s. \quad (7.154)$$

Taking the curl of both sides and using the identity $\nabla \times (\nabla \times \mathbf{B}) = \nabla(\nabla \cdot \mathbf{B}) - \nabla^2 \mathbf{B}$ along with the Maxwell equation $\nabla \cdot \mathbf{B} = 0$ gives

$$\nabla^2 \mathbf{B} = -\frac{1}{\lambda_L^2} \mathbf{B}, \quad (7.155)$$

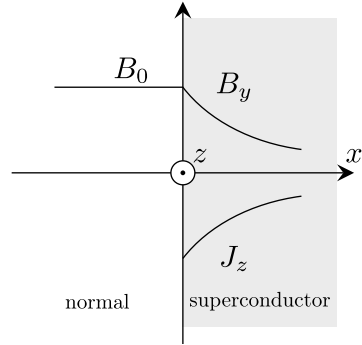
where

$$\lambda_L = \sqrt{\frac{\Lambda}{\mu_0}} = \sqrt{\frac{m^*}{\mu_0 n_s^* (e^*)^2}}. \quad (7.156)$$

Note that (7.149) implies that the current at a particular point depends only on the field at that point. However, if the coherence length is large compared to the variations in the field, then the current has a non-local relationship to the field and the point relation (7.149) is not valid. In such cases it turns out that (7.155) is still useful, though the effective characteristic length will be different from λ_L . Consequently in the following discussion we use the more generic λ instead of λ_L .

As an example, consider the case of a magnetic field applied parallel to the surface of a semi-infinite superconducting region as illustrated in Figure 7.15. In this case (7.155) becomes

Figure 7.15 Behavior of the magnetic field and superconducting current at the interface between a superconductor and a normal region. (Note that the field and current have different scales.)



$$\frac{\partial^2 B_y}{\partial x^2} = \frac{1}{\lambda} B_y. \quad (7.157)$$

The general solution to this equation is

$$B_y = B_+ e^{x/\lambda} + B_- e^{-x/\lambda}. \quad (7.158)$$

If B_y is to remain finite, we conclude $B_+ = 0$. Further, matching the boundary condition at $x = 0$ gives $B_- = B_0$. The solution inside the superconductor is therefore

$$B_y(x) = B_0 e^{-x/\lambda}, \quad x \geq 0. \quad (7.159)$$

The current is then obtained from (7.154):

$$J_{sz} = -\frac{B_0}{\mu_0 \lambda} e^{-x/\lambda}. \quad (7.160)$$

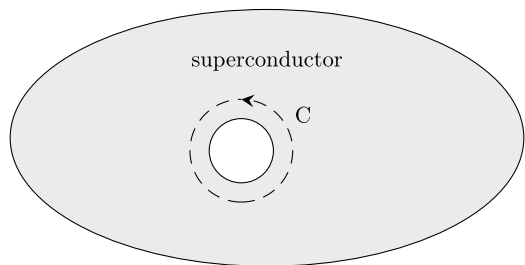
We see that the field is indeed excluded from the bulk, but it does penetrate a distance λ . The field deep within the superconductor vanishes because of shielding currents near the surface that also extend over a depth λ .

7.5.6 Flux Quantization

In this section we will see that although the magnetic field must be zero inside a superconductor, a field can exist in a hole through a superconductor, but it can only take on discrete values.

Consider a superconducting sample with a hole in it as shown in Figure 7.16. Recall that the canonical momentum in a superconductor in the presence of a magnetic field is $\mathbf{p} = \hbar \nabla \theta = e^* \Lambda \mathbf{J}_s + e^* \mathbf{A}$. Let us integrate this around the closed contour C shown in Figure 7.16:

Figure 7.16 Geometry of a superconducting sample containing a hole used to calculate the magnetic flux quantum.



$$\oint \mathbf{p} \cdot d\mathbf{l} = \hbar \oint \nabla \theta \cdot d\mathbf{l} = \oint \mathbf{A} \cdot d\mathbf{l}. \quad (7.161)$$

Here we have used the fact that $\mathbf{J}_s = 0$ if we choose the contour to be deep within the superconductor (i.e., deep with respect to the decay length λ).

Note that in general

$$\int_a^b \nabla \theta \cdot d\mathbf{l} = \theta_b - \theta_a. \quad (7.162)$$

Since the wave function must be single-valued and the phase is determined to within a multiple of 2π at each point, we have $\theta_b = \theta_a + 2n\pi$ if the starting and ending points are the same (i.e., if the contour is closed). We conclude that

$$\hbar \oint \nabla \theta \cdot d\mathbf{l} = 2n\pi\hbar = \oint \mathbf{A} \cdot d\mathbf{l}. \quad (7.163)$$

Stokes's theorem enables us to relate a contour integral to an integral over the surface enclosed by the contour:

$$\begin{aligned} \oint \mathbf{A} \cdot d\mathbf{l} &= e^* \int_S (\nabla \times \mathbf{A}) \cdot d\mathbf{s} \\ &= e^* \int_S \mathbf{B} \cdot d\mathbf{s} \\ &\equiv e^* \Phi_s \end{aligned} \quad (7.164)$$

where Φ_s is the total flux enclosed by the contour C . Since the flux density is zero in the bulk of the superconductor, Φ_s is the flux through the hole. We conclude that the flux is *quantized*:

$$\Phi_s = \frac{2n\pi\hbar}{e^*} = \frac{nh}{e^*}. \quad (7.165)$$

The basic unit of flux, called the *flux quantum*, is

$$\Phi_0 = \frac{h}{2e} = 2.07 \times 10^{-15} \text{ Wb}. \quad (7.166)$$

The trapped flux is accompanied by shielding currents required to keep $\mathbf{B}$ zero in the bulk of the superconductor. These *persistent currents* do not exhibit any measurable decay.

7.6 Chapter Summary

In this chapter we have provided an introduction to the microscopic theory of superconductivity, along with the needed basic concepts from solid state physics. The main points for our purposes can be summarized briefly as follows:

- Below a certain critical temperature, conduction electrons in certain materials experience a net attractive interaction caused by their interaction with the ions in the material.
- As a result of this attractive interaction, electrons with opposite spin and opposite momenta form bound states called Cooper pairs.

- The size of a Cooper pair is very large compared to the distance between electrons so that there is significant overlap between Cooper pairs.
- Owing to this overlap, it is energetically favorable for the Cooper pairs to “lock phases” so that a macroscopic sample can be described with a single ensemble average wave function of the form $\psi = |\psi(\mathbf{r})| \exp(i\theta(\mathbf{r}))$.
- The formation of Cooper pairs lowers the energy of the ground state from that of the normal metal. The amount of energy necessary to break a Cooper pair—and thus increase the system energy—is 2Δ , where Δ is the superconducting energy gap and represents the energy needed to remove a single electron from a Cooper pair.
- Because of the energy gap, application of small perturbations to the system that do not break Cooper pairs will not change the ground state. As a result, application of a magnetic field will result in supercurrents near the surface that cancel the applied field interior to the sample, provided the energy stored in this induced magnetic field is insufficient to break Cooper pairs.
- The fact that the magnetic field is zero inside a superconductor along with the effective macroscopic wave function lead to the conclusion that the magnetic flux through a hole in a superconductor must be quantized. The value of the flux quantum is $\Phi_0 = h/2e$.

7.7 Exercises

- 7.1** Using the free electron model of a metal, find the Fermi energy of copper if the density of conduction electrons is $8.47 \times 10^{28} \text{ m}^{-3}$. Express your answer in both Joules and electron volts (eV).
- 7.2** Aluminum has an atomic density of $6.03 \times 10^{28} \text{ m}^{-3}$ and an acoustic phonon velocity (longitudinal sound velocity) of 6,420 m/s. Find the Debye frequency and Debye temperature.
- 7.3** The conductivity of copper is $5.96 \times 10^7 \text{ S/m}$. If the electron density is $8.47 \times 10^{28} \text{ m}^{-3}$, estimate the momentum relaxation time τ using the free electron model.
- 7.4** Estimate the kinetic inductance of a copper wire 1 cm long and 0.1 mm in diameter. The electron density in copper is $8.47 \times 10^{28} \text{ m}^{-3}$.
- 7.5** In this problem we will find the 3D Fourier transform of the Coulomb potential, that is we will show that

$$\mathcal{F}\left\{\frac{1}{r}\right\} = \int d^3r \frac{1}{r} e^{i\mathbf{q}\cdot\mathbf{r}} = \frac{1}{q^2}. \quad (7.167)$$

Our approach will be to take the Fourier transform of a similar integral with exponential damping, then taking the limit as the damping goes to zero. A suitable form is the *Yukawa* potential $\exp(-\lambda r)/r$.

- (a)** Choose the coordinate system so that $\mathbf{q}$ is along the z axis, and write the integral explicitly in spherical coordinates, replacing the dot product with $\cos \theta$. Complete the integration over the azimuthal coordinate ϕ .

- (b) Perform a change of variable from θ to $x = \cos \theta$, also changing the limits of integration as appropriate. Perform the integration over x .
- (c) Finally, perform the integration over r and take the limit as $\lambda \rightarrow 0$ to recover the desired result.

7.6 Calculate the London penetration depth for aluminum if the single-particle electron density is $18.1 \times 10^{28} \text{ m}^{-3}$.

8

Josephson Junctions

The device that is the workhorse of most superconducting computers is the Josephson Junction, which consists of two superconducting regions separated by a non-superconducting layer—usually an insulator. A typical junction is fabricated from aluminum films with thicknesses in the range of 35–85 nm. The native oxide of the aluminum is used to form the insulating layer. This structure exhibits a non-linear inductance, which results in different spacings between the excited energy levels and enables us to uniquely address only two states—a key requirement for a qubit. By biasing the Josephson junction with different currents, it can also be used to realize a time-varying inductance that is used in low noise parametric amplifiers. Finally, when two junctions are placed in parallel, the effective inductance can be varied by the application of a magnetic field. This makes it possible to realize tunable qubits.

The useful dynamics result from Cooper pairs that “tunnel” across the non-superconducting layer in the junction. This tunneling phenomenon is purely quantum mechanical and very unintuitive. For this reason we begin the chapter with a basic discussion of tunneling.

8.1 Tunneling

Suppose that we have a tennis ball in the bottom of a box. The only way to get the ball out of the box classically is to lift it up and over the side. In doing this, we are giving the ball additional gravitational potential energy so that its energy exceeds the potential energy at the top of the wall of the box.

In contrast, if we consider instead an electron trapped in a “box,” we must analyze the problem quantum mechanically. It turns out that because the wave function of the electron does not go to zero inside the wall of the box but rather decays exponentially, there is a finite possibility that the electron will be found outside the box if a measurement is made. It is as if the electron can “tunnel” through the barrier represented by the wall of the box.

The amount of energy it takes to remove an electron from a metal is called the *work function*, and is usually of order several electron volts—enough to keep the electrons comfortably inside the metal under normal circumstances. We can consider the electrons trapped inside a box with walls whose height is represented by the work function.

However, if two superconductors are placed sufficiently close together, the “wall” of the box becomes sufficiently thin that the wave function representing each superconductor still has non-negligible amplitude at the other side of the barrier, making tunneling possible.

The details of tunneling between two superconductors are not critical for our purposes, but it is helpful to have an intuitive notion about tunneling. Toward this end, let us consider the 1-dimensional problem of a particle incident on a potential barrier that is higher than the energy of the particle.¹ We first consider the case of a step in potential so that the barrier thickness extends indefinitely, then we consider the case of a finite thickness barrier.

8.1.1 Reflection from a Barrier

Consider an electron represented by the plane wave $\exp(ikx)$ incident from the left on a potential barrier $U(x)$ of the form

$$U(x) = \begin{cases} 0 & x \leq 0 \\ V & x > 0 \end{cases} \quad (8.1)$$

as shown in Figure 8.1(a). To understand the behavior of the electron in this case, we need to solve the Schrödinger equation

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} + U(x)\psi = \mathcal{E}\psi. \quad (8.2)$$

It is easy to verify that when U is a constant, the general solution of Schrödinger’s equation is

$$\psi = A e^{ikx} + B e^{-ikx}, \quad (8.3)$$

where A, B are constants determined by the boundary conditions, and

$$k = \frac{1}{\hbar} \sqrt{2m(\mathcal{E} - U)}. \quad (8.4)$$

With these definitions we can construct the general solution for the wave function in the two regions as

$$\psi(x) = \begin{cases} e^{ikx} + R e^{-ikx} & x \leq 0 \\ T e^{-\kappa x} & x > 0. \end{cases} \quad (8.5)$$

Here

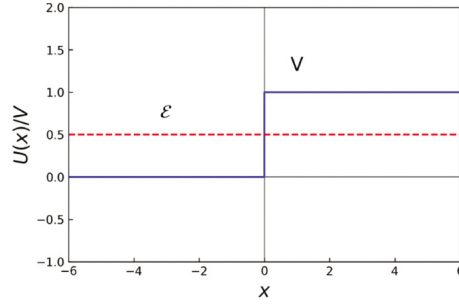
$$k = \frac{1}{\hbar} \sqrt{2m\mathcal{E}} \quad (8.6)$$

and

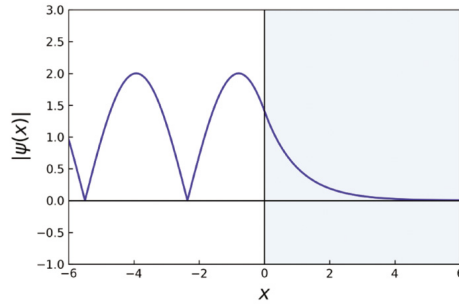
$$\kappa = \frac{1}{\hbar} \sqrt{2m(V - \mathcal{E})}. \quad (8.7)$$

¹ This problem is discussed in most introductory texts on quantum mechanics. Our discussion draws from [41].

Figure 8.1 Geometry for a particle incident from the left on a potential barrier filling the half-space for $x > 0$. The wave function plot is for the case $\mathcal{E} = V/2$.



(a) Step potential barrier.



(b) Magnitude of the wave function of a particle incident from the left on a step potential barrier.

The quantity $1/\kappa$ can be interpreted as a characteristic decay length in the barrier; it represents the distance over which the amplitude decays to $1/e$ of its value at the interface.

The two simultaneous equations needed to solve for R and T are obtained by requiring the wave function and its derivative to be continuous at $x = 0$:

$$1 + R = T \quad (8.8)$$

$$ik(1 - R) = -\kappa T. \quad (8.9)$$

Eliminating T and solving for R gives

$$R = \frac{ik + \kappa}{ik - \kappa} \equiv e^{i\alpha}. \quad (8.10)$$

The introduction of the phase angle α is motivated by the observation that the numerator and denominator have the same magnitude, so R is characterized by unity magnitude but with a phase that depends on the barrier height. $|R|^2$ simply gives the probability that the electron will be reflected, and the fact that its magnitude is unity means, of course, that it will always be reflected since its energy is less than the barrier height. It follows from (8.8) that

$$T = 1 + e^{i\alpha}. \quad (8.11)$$

Substituting these values for R, T into (8.5) gives

$$\psi = \begin{cases} 2e^{i\alpha/2} \cos(kx - \alpha/2) & x \leq 0 \\ 2e^{i\alpha/2} \cos(\alpha/2)e^{-\kappa x} & x > 0. \end{cases} \quad (8.12)$$

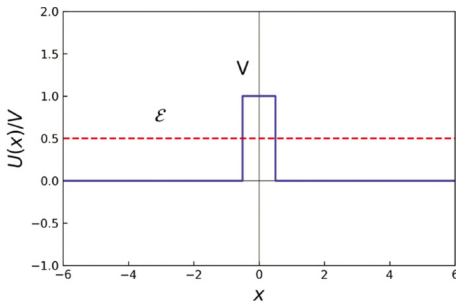
This wave function is plotted in Figure 8.1(b) for the case $V = 2\mathcal{E}$. The pattern to the left of the boundary is simply the envelope of a standing wave pattern, similar to that shown in Figure 4.4.

Note that although the particle is reflected with certainty, the wave function does penetrate into the barrier with an exponential decay. It is easy to imagine what would happen if the barrier had a finite depth: satisfying the boundary conditions at the second interface would require a finite amplitude beyond the barrier. In other words there would be some non-zero probability that the electron would be found beyond the barrier, even though its energy was less than the barrier height!

8.1.2 Finite Thickness Barrier

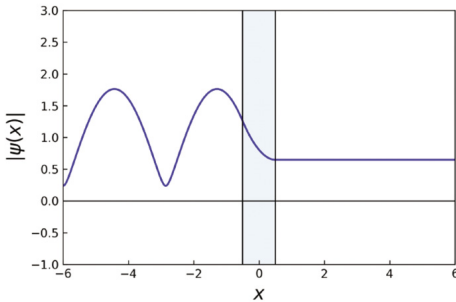
To verify our intuition, the problem can be generalized to the case of a barrier with finite width as shown in Figure 8.2(a) using a similar approach. In this case the potential is given by

$$U(x) = \begin{cases} V & |x| \leq a/2 \\ 0 & \text{otherwise.} \end{cases} \quad (8.13)$$



(a) Finite width potential barrier.

Figure 8.2 Geometry for a particle incident from the left on a potential barrier with finite width. The wave function plot is for the case $\mathcal{E} = V/2$.



(b) Magnitude of the wave function of a particle incident from the left on a finite width potential barrier.

With these definitions we can construct the general solution for the wave function in the three regions as

$$\psi(x) = \begin{cases} e^{ikx} + R e^{-ikx} & x \leq -a/2 \\ A e^{-\kappa x} + B e^{\kappa x} & |x| < a/2 \\ T e^{ikx} & x \geq a/2. \end{cases} \quad (8.14)$$

The four equations needed to find R, A, B, T are provided by satisfying the boundary conditions that both ψ and $d\psi/dx$ are continuous at $x = \pm a/2$. The resulting equations for $x = -a/2$ are

$$e^{-ika/2} + R e^{ika/2} = A e^{\kappa a/2} + B e^{-\kappa a/2} \quad (8.15)$$

$$ik e^{-ika/2} - ik R e^{ika/2} = -\kappa A e^{\kappa a/2} + \kappa B e^{-\kappa a/2}. \quad (8.16)$$

Adding and subtracting these two equations and expressing the result in matrix form gives

$$\begin{bmatrix} 1 \\ R \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \left(1 + i\frac{\kappa}{k}\right) e^{(\kappa+ik)a/2} & \left(1 - i\frac{\kappa}{k}\right) e^{(-\kappa+ik)a/2} \\ \left(1 - i\frac{\kappa}{k}\right) e^{(\kappa-ik)a/2} & \left(1 + i\frac{\kappa}{k}\right) e^{(-\kappa-ik)a/2} \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix}. \quad (8.17)$$

Similarly, applying the boundary conditions at $x = a/2$ gives

$$A e^{-\kappa a/2} + B e^{\kappa a/2} = T e^{ika/2} \quad (8.18)$$

$$-\kappa A e^{-\kappa a/2} + \kappa B e^{\kappa a/2} = ik T e^{ika/2}. \quad (8.19)$$

Adding and subtracting these equations as before leads to the matrix equation

$$\begin{bmatrix} A \\ B \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \left(1 - i\frac{k}{\kappa}\right) e^{(\kappa+ik)a/2} \\ \left(1 + i\frac{k}{\kappa}\right) e^{(-\kappa+ik)a/2} \end{bmatrix} T. \quad (8.20)$$

Substituting (8.20) into (8.17) gives

$$T = \frac{e^{-ika}}{\cosh \kappa a + i(\varepsilon/2) \sinh \kappa a}, \quad (8.21)$$

and

$$R = \frac{-i(\eta/2) \sinh \kappa a e^{-ika}}{\cosh \kappa a + i(\varepsilon/2) \sinh \kappa a}, \quad (8.22)$$

where

$$\varepsilon = \frac{\kappa}{k} - \frac{k}{\kappa}, \quad \text{and} \quad \eta = \frac{\kappa}{k} + \frac{k}{\kappa}. \quad (8.23)$$

The coefficients A, B are then given by (8.20). The magnitude of the resulting wave function for the case of $\mathcal{E} = V/2$ is shown in Figure 8.2(b). Note that the finite amplitude in the transmitted region means that there is some probability that the electron will somehow “tunnel” through the barrier even though its energy is less than the barrier height. Note also that the cancellation in the standing wave pattern on the left side is not complete, since the amplitude of the reflected wave is now diminished, and conservation of probability requires that $|R|^2 = 1 - |T|^2$ (see 8.1).

In this section we have discussed the tunneling of a single electron. In the case of the Josephson junction, Cooper pairs are tunneling as opposed to single electrons (which in this context we would refer to as quasi-particle tunneling). The main notions of the wave function decaying exponentially across a thin barrier allowing coupling between excitations on the two sides remains applicable, but there are important differences. In particular, pair tunneling can occur even without a bias voltage applied across the junction, and consequently currents can flow without a voltage drop under certain circumstances.

8.2 Josephson Junctions

In this section we use a phenomenological approach to model the coupling between the superconductors in a Josephson junction resulting from the tunneling of Cooper pairs.

8.2.1 Current and Voltage Relations

As we have discussed, the state of a superconductor can be represented by a macroscopic wave function of the form

$$\psi(\mathbf{r}) = \sqrt{n^*} e^{i\theta}. \quad (8.24)$$

For an isolated superconductor, this wave function must satisfy the time-dependent Schrödinger equation

$$i\hbar \frac{\partial \psi}{\partial t} = U\psi, \quad (8.25)$$

where U is the energy of the superconducting state. The magnitude of this wave function will be uniform inside the superconductor, but will decay exponentially away from the surface. Consequently, if two superconductors are placed close enough to one another, it is possible for the wave functions to overlap, resulting in tunneling of Cooper pairs from one to the other. As previously discussed, a *Josephson Junction* is formed when two superconductors are separated by a thin insulating layer as shown in Figure 8.3. The interaction causes shifts in the energies of the two wave functions. The interaction can be described phenomenologically by the coupled equations²

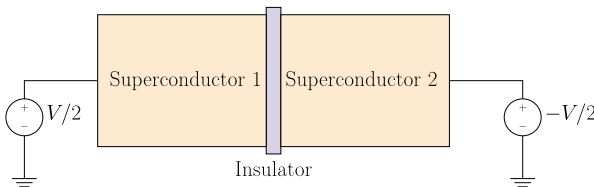


Figure 8.3 A Josephson junction is formed by two superconductors separated by a thin insulating layer.

² Our analysis follows Feynman [57].

$$i\hbar \frac{\partial \psi_1}{\partial t} = U_1 \psi_1 + \kappa \psi_2, \quad (8.26)$$

$$i\hbar \frac{\partial \psi_2}{\partial t} = U_2 \psi_2 + \kappa \psi_1, \quad (8.27)$$

where κ is a coupling constant describing the strength of the interaction, and determined by the details of the geometry. This approach has the advantage of enabling us to understand the behavior of the junction without detailed modeling of the tunneling process.

Now suppose the voltage $+V/2$ is applied to superconductor 1, and $-V/2$ to superconductor 2, as shown in Figure 8.3. If both superconductors are identical, then $U_1 = U_0 - e^*V/2$ and $U_2 = U_0 + e^*V/2$, where e^* is the magnitude of the charge of a Cooper pair, i.e., $e^* = 2|e|$, and U_0 is the energy of the uncoupled superconductors. Choosing U_0 as our reference energy, the coupled equations become

$$i\hbar \frac{\partial \psi_1}{\partial t} = -\frac{e^*V}{2} \psi_1 + \kappa \psi_2, \quad (8.28)$$

$$i\hbar \frac{\partial \psi_2}{\partial t} = +\frac{e^*V}{2} \psi_2 + \kappa \psi_1. \quad (8.29)$$

Substituting $\psi_1 = \sqrt{n_1^*} \exp i\theta_1$ and $\psi_2 = \sqrt{n_2^*} \exp i\theta_2$ into (8.28) leads to

$$i\hbar \frac{\partial n_1^*}{\partial t} - 2\hbar n_1^* \frac{\partial \theta_1}{\partial t} = -e^*V n_1^* + 2\kappa \sqrt{n_1^* n_2^*} e^{i\phi}, \quad (8.30)$$

where $\phi = \theta_2 - \theta_1$ is the difference in phase between the wave functions in the two superconductors. Taking the real and imaginary parts of (8.30) gives

$$\frac{\partial \theta_1}{\partial t} = \frac{e^*V}{2\hbar} - \frac{\kappa}{\hbar} \sqrt{\frac{n_2^*}{n_1^*}} \cos \phi, \quad (8.31)$$

and

$$\frac{\partial n_1^*}{\partial t} = \frac{2\kappa}{\hbar} \sqrt{n_1^* n_2^*} \sin \phi. \quad (8.32)$$

Following a similar process with (8.29) leads to

$$\frac{\partial \theta_2}{\partial t} = -\frac{e^*V}{2\hbar} - \frac{\kappa}{\hbar} \sqrt{\frac{n_1^*}{n_2^*}} \cos \phi, \quad (8.33)$$

$$\frac{\partial n_2^*}{\partial t} = -\frac{2\kappa}{\hbar} \sqrt{n_1^* n_2^*} \sin \phi. \quad (8.34)$$

Note that $\partial n_1^*/\partial t = -\partial n_2^*/\partial t$; i.e., when pairs flow across, the tendency to increase on one side is equal to the tendency to decrease on the other. We say *tendency*, since a charge imbalance is prevented by allowing the electrons to return through the external circuit.

To relate this to the current density across the junction, let us consider what this tendency would mean if there was no external circuit. Consider a volume that encloses superconductor 1 and passes through the insulator. The continuity equation relates the

current across the junction to the change in charge in the volume:

$$\begin{aligned}
 \oint \mathbf{J}_1 \cdot d\mathbf{s} &= -\frac{\partial}{\partial t} \int_v \rho_1 dv \\
 J_1 A &= -\frac{\partial}{\partial t} \int_v (-e^* n_1^*) dv \\
 &= e^* \frac{\partial n_1^*}{\partial t} AW \\
 &= e^* AW \frac{2\kappa}{\hbar} \sqrt{n_1^* n_2^*} \sin \phi,
 \end{aligned} \tag{8.35}$$

or

$$I_1 = I_c \sin \phi. \tag{8.36}$$

Here J_1 , I_1 are the current density and current directed from superconductor 1 to superconductor 2, A is the junction area, and W is the dimension of superconductor 1 perpendicular to the junction (we assume the superconductor to be a rectangular solid for simplicity). The quantity I_c represents the maximum current that can flow through the superconducting junction, also referred to as the *critical current*.

The behavior of ϕ is given by

$$\frac{\partial \phi}{\partial t} = \frac{\partial \theta_2}{\partial t} - \frac{\partial \theta_1}{\partial t}. \tag{8.37}$$

If the superconductors are identical so that $n_1^* = n_2^*$, this reduces to

$$\frac{\partial \phi}{\partial t} = \frac{e^* V}{\hbar} = \frac{2\pi}{\hbar} 2eV \equiv \frac{2\pi}{\Phi_0} V, \tag{8.38}$$

where $\Phi_0 = h/(2e)$ is the flux quantum. The voltage across the junction can therefore be written

$$V = \frac{\Phi_0}{2\pi} \frac{\partial \phi}{\partial t}. \tag{8.39}$$

Equations (8.36) and (8.39) give the current and voltage across a Josephson junction in terms of the phase difference ϕ .

To get some understanding about why the Josephson junction is of interest for making qubits, let's take the derivative of the Josephson current (8.36), then express this in terms of the Josephson voltage using (8.39):

$$\begin{aligned}
 \frac{\partial I}{\partial t} &= I_c \cos \phi \frac{\partial \phi}{\partial t} \\
 &= I_c \cos \phi \frac{2\pi V}{\Phi_0}.
 \end{aligned} \tag{8.40}$$

Solving for the voltage, we have

$$\begin{aligned}
 V &= \frac{\Phi_0}{2\pi I_c \cos \phi} \frac{\partial I}{\partial t} \\
 &= \frac{\Phi_0}{2\pi I_c \sqrt{1 - \sin^2 \phi}} \frac{\partial I}{\partial t} \\
 &= \frac{\Phi_0}{2\pi I_c \sqrt{1 - (I/I_c)^2}} \frac{\partial I}{\partial t} \\
 &\equiv L_{\text{eff}}(I) \frac{\partial I}{\partial t},
 \end{aligned} \tag{8.41}$$

where

$$L_{\text{eff}} = \frac{\Phi_0}{2\pi I_c \sqrt{1 - (I/I_c)^2}}. \tag{8.42}$$

The Josephson junction therefore looks like a non-linear inductor; i.e., an inductor whose value depends on the current through it. The consequence of this is that if this inductor was used to form a circuit resonator, the energy levels would be unequally spaced—unlike a circuit with a linear inductor that can be modeled as a quantum harmonic oscillator. This enables individual transitions to be specifically addressed so that an approximation to a two-level system can be obtained and used to represent a qubit.

It is important to note that the inductance described by (8.42) does not arise from magnetic flux linkage as in the case of familiar magnetic inductors. Instead, it is a kinetic phenomena arising from the inertia of the Cooper pairs.

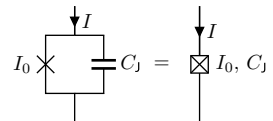
From the geometry (Figure 8.3), it is also clear that there is a junction capacitance in parallel with the Josephson tunneling junction. Consequently, the equivalent circuit for the Josephson junction is shown in Figure 8.4.

8.2.2 Josephson Junction Hamiltonian

In the previous section, we treated the current and voltage through the Josephson junction as classical circuit quantities. Recall that our procedure for transitioning to a quantum analysis of a system is to obtain the Hamiltonian, then replace the conjugate variables in the Hamiltonian with operators. We therefore need to obtain the Hamiltonian for the Josephson Junction. Our discussion follows the treatment in the *Qiskit Textbook* [58].

Consider the circuit shown in Figure 8.5 consisting of a Josephson junction in parallel with a capacitance. In general the capacitance includes the junction capacitance as well as additional parallel capacitance from the circuit. Since we have observed that the Josephson junction has a kinetic inductance (i.e., an inductance arising from the momentum of the Cooper pairs as opposed to magnetic induction), this represents a resonant circuit.

Figure 8.4 Circuit symbols for a Josephson junction. Left: equivalent circuit showing the tunneling junction in parallel with the junction capacitance. Right: compact symbol including both the tunneling junction and the capacitance.



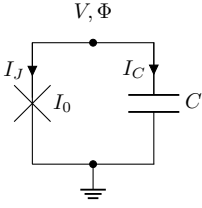


Figure 8.5 Non-linear resonant circuit formed by shunting a Josephson tunnel junction with a capacitor. Here C is the total parallel capacitance including the intrinsic junction capacitance.

Applying Kirchhoff's current law to the top node gives

$$\begin{aligned}
 I_J + I_C &= 0 \\
 I_c \sin \phi + C \frac{dV}{dt} &= 0 \\
 I_c \sin \phi + \frac{C\Phi_0}{2\pi} \frac{d^2\phi}{dt^2} &= 0
 \end{aligned} \tag{8.43}$$

where use has been made of the relation between the phase and voltage given by (8.39). Comparing (8.39) with (6.63), we see that the phase is related to the node flux by $\phi = 2\pi\Phi/\Phi_0$. The equation from Kirchhoff's law can then be written in terms of the node flux:

$$I_c \sin(2\pi\Phi/\Phi_0) + C\ddot{\Phi} = 0. \tag{8.44}$$

Since we are interested in the two lowest energy states, we are interested in solutions when Φ is small. In this case the sine function can be approximated by its argument. If we further take the solution of the form $\exp(j\omega t)$, this gives the ground state resonant frequency as

$$\omega_0 = \sqrt{\frac{2\pi I_c}{\Phi_0 C}}. \tag{8.45}$$

We will return to this frequency in a moment.

In Section 6.1 we obtained the “equation of motion” from the Lagrangian using the Euler-Lagrange equation. In this case, we have the equation of motion and we need to work backward to find the Lagrangian. Specifically, we need to find $\mathcal{L}$ that satisfies

$$\frac{\partial \mathcal{L}}{\partial \Phi} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\Phi}} \right) = I_c \sin(2\pi\Phi/\Phi_0) + C\ddot{\Phi} = 0. \tag{8.46}$$

Noting that Φ only appears in the first term on the RHS of (8.46), and $\dot{\Phi}$ only appears in the second term, we can guess

$$\frac{\partial \mathcal{L}}{\partial \Phi} = I_c \sin(2\pi\Phi/\Phi_0), \tag{8.47}$$

and

$$\frac{\partial \mathcal{L}}{\partial \dot{\Phi}} = -C\dot{\Phi}. \tag{8.48}$$

Integrating both of these expressions and noting that the overall sign of the Euler-Lagrange equation does not affect the dynamics suggests that a suitable Lagrangian

is given by³

$$\mathcal{L} = \frac{I_c \Phi_0}{2\pi} \cos(2\pi\Phi/\Phi_0) + \frac{C}{2} (\dot{\Phi})^2. \quad (8.49)$$

Following the procedure outlined in Section 6.4.3, the conjugate variable to Φ is given by

$$Q = \frac{\partial \mathcal{L}}{\partial \dot{\Phi}} = C\dot{\Phi}, \quad (8.50)$$

and the Hamiltonian is given by

$$\begin{aligned} \mathcal{H} &= \dot{\Phi}Q - \mathcal{L} \\ &= \frac{Q^2}{2C} - \frac{I_c \Phi_0}{2\pi} \cos(2\pi\Phi/\Phi_0) \\ &= 4E_C \left(\frac{Q}{e^*} \right)^2 - E_J \cos(2\pi\Phi/\Phi_0), \end{aligned} \quad (8.51)$$

where we have introduced the charging energy⁴ for a single electron $E_C = e^2/(2C)$ and the Josephson energy $E_J = I_c \Phi_0/(2\pi)$. The charging energy of a Cooper pair is then $4E_C$. We see that the capacitive term is the same form as in the linear resonator model, but the inductive term is now replaced by the Josephson cosine term. In terms of the charging and Josephson energies, the resonant frequency (8.45) becomes

$$\omega_0 = \frac{\sqrt{8E_C E_J}}{\hbar}. \quad (8.52)$$

8.2.3 Quantized Josephson Junction Analysis

Our general procedure for quantizing the Hamiltonian is to re-interpret the conjugate variables as non-commuting operators. To transform into the second-quantized or occupation number representation, we introduce the operators

$$Q = -i \frac{e^*}{2} \left(\frac{E_J}{2E_C} \right)^{1/4} (c - c^\dagger), \quad (8.53)$$

$$\Phi = \frac{\Phi_0}{2\pi} \left(\frac{2E_C}{E_J} \right)^{1/4} (c + c^\dagger), \quad (8.54)$$

where $[c, c^\dagger] = 1$, and $c^\dagger, c$ are raising and lowering operators for the anharmonic system and have the usual properties

$$a|j\rangle = \sqrt{j}|j-1\rangle, \quad a^\dagger|j\rangle = \sqrt{(j+1)}|j+1\rangle, \quad a^\dagger a|j\rangle = j|j\rangle, \quad (8.55)$$

and $|j\rangle$ is the j th state of the anharmonic system.

³ Note that we have not rigorously integrated the equation of motion, but rather have made some “educated guesses” to obtain a satisfactory form for the Lagrangian.

⁴ Recall that the energy stored in a capacitor is $CV^2/2$, and the charge is related to the voltage through $Q = CV$. Combining these to eliminate V gives the energy $Q^2/(2C)$. Thus $e^2/(2C)$ is the energy contained in a capacitor charged with a single unit of charge.

One of the most commonly-used superconducting qubits is the *transmon* (short for *transmission-line shunted plasma oscillation*) qubit. This qubit is realized for our Josephson junction resonant circuit if $E_J \gg E_C$. This is realized by adding a large shunt capacitance. This condition minimizes sensitivity to fluctuations in charge. In this limit, we see from (8.54) that Φ becomes small, enabling us to expand the cosine term in (8.51) in a Taylor series

$$\begin{aligned}
 -E_J \cos\left(\frac{2\pi\Phi}{\Phi_0}\right) &= -E_J \left(1 - \frac{1}{2!} \left(\frac{2E_C}{E_J}\right)^{1/2} (c + c^\dagger)^2 \right. \\
 &\quad \left. + \frac{1}{4!} \left(\frac{2E_C}{E_J}\right) (c + c^\dagger)^4 + \dots\right) \\
 &= -E_J + \frac{\sqrt{8E_C E_J}}{4} (c + c^\dagger)^2 - \frac{E_C}{12} (c + c^\dagger)^4 + \dots \\
 &= -E_J + \frac{\hbar\omega_0}{4} (c + c^\dagger)^2 - \frac{E_C}{12} (c + c^\dagger)^4 + \dots
 \end{aligned} \tag{8.56}$$

Next let's consider the first term in the Hamiltonian (8.51):

$$\begin{aligned}
 4E_C \left(\frac{Q}{e^*}\right)^2 &= -E_C \left(\frac{E_J}{2E_C}\right)^{1/2} (c - c^\dagger)^2 \\
 &= -\frac{\sqrt{8E_J E_C}}{4} (c - c^\dagger)^2 \\
 &= -\frac{\hbar\omega_0}{4} (c - c^\dagger)^2.
 \end{aligned} \tag{8.57}$$

Adding (8.56) and (8.57) and simplifying using the commutator $[c, c^\dagger] = 1$ allows us to write the complete Hamiltonian as

$$\mathcal{H} = -E_J + \hbar\omega_0 (c^\dagger c + 1/2) - \frac{E_C}{12} (c + c^\dagger)^4 + \dots \tag{8.58}$$

Let's consider taking the expectation value of the Hamiltonian $\langle j | \mathcal{H} | j \rangle$. From the third term in (8.58) we need to evaluate $\langle j | (c + c^\dagger)^4 | j \rangle$. Because $\langle j | k \rangle = \delta_{j,k}$, the terms in this expectation will vanish unless there are the same number of c 's and $c^\dagger$'s in the term. The surviving terms are therefore

$$(c + c^\dagger)^4 \rightarrow c c c^\dagger c^\dagger + (c c^\dagger)^2 + c c^\dagger c^\dagger c + c^\dagger c c c^\dagger + (c^\dagger c)^2 + c^\dagger c^\dagger c c. \tag{8.59}$$

Using the commutator for c and $c^\dagger$ this can be simplified to

$$(c + c^\dagger)^4 \rightarrow 3 + 6c^\dagger c + 6(c^\dagger c)^2. \tag{8.60}$$

The Hamiltonian finally becomes

$$\mathcal{H} = -E_J - \frac{E_C}{4} + \frac{\hbar\omega_0}{2} + \left(\hbar\omega_0 - \frac{E_C}{2}\right) c^\dagger c - \frac{E_C}{2} (c^\dagger c)^2. \tag{8.61}$$

The result of a measurement of the energy will be the expectation value

$$\begin{aligned}
 \langle j | \mathcal{H} | j \rangle &= -E_J - \frac{E_C}{4} + \frac{\hbar\omega_0}{2} + \left(\hbar\omega_0 - \frac{E_C}{2}\right) j - \frac{E_C}{2} j^2 \\
 &\equiv \hbar\omega_j
 \end{aligned} \tag{8.62}$$

where $\hbar\omega_j$ is the energy of the j th transmon state. We are particularly interested in the difference between adjacent states, since that is the energy that must be supplied or given up for the transmon to change states. The energy gap between the state $(j + 1)$ and j is

$$\hbar(\omega_{j+1} - \omega_j) = \hbar\omega_0 - E_C(1 + j). \quad (8.63)$$

The energy needed for the transition $|0\rangle \rightarrow |1\rangle$ is therefore $\hbar\omega_0 - E_C$, while the energy for the transition $|1\rangle \rightarrow |2\rangle$ is $\hbar\omega_0 - 2E_C$. This makes it possible to just address the $|0\rangle \rightarrow |1\rangle$ transition, and therefore model the transmon as a two-state system. The difference in energy between adjacent transitions, E_C , is referred to as the *anharmonicity* and sometimes denoted δ .

8.3 Superconducting Quantum Interference Devices (SQUIDs)

A qubit can also be formed by placing two Josephson junctions in parallel as shown in Figure 8.6. This configuration is referred to as a Superconducting Quantum Interference Device, or SQUID. The name comes from the observation that when an applied current splits along the two branches, the currents will acquire different phase shifts if magnetic flux links the loop. The currents will then interfere when they recombine in a manner that is determined by the magnitude of the flux linking the loop. This configuration is of interest in the present context since it allows a mechanism for tuning the frequency of the qubit by applying a magnetic field.

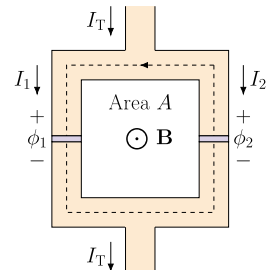
Recall from Section 7.5.3 that the momentum of a Cooper pair is given by

$$\mathbf{p} = \hbar\nabla\theta = e^*(\Lambda\mathbf{J}_S + \mathbf{A}). \quad (8.64)$$

Consider the line integral of the momentum around the dashed line shown in Figure 8.6. Assuming that the thickness of the superconductor is large compared with the London penetration depth, we can choose the location of the path interior to the superconductor where the current density is zero. The line integral becomes

$$\begin{aligned} \oint \nabla\theta \cdot d\ell &= \frac{e^*}{\hbar} \oint \mathbf{A} \cdot d\ell \\ \phi_1 - \phi_2 &= \frac{e^*}{\hbar} \int \mathbf{B} \cdot d\mathbf{s} \\ \phi_1 - \phi_2 &= \frac{e^*}{\hbar} BA \\ &= -\frac{2\pi\Phi_B}{\Phi_0} \end{aligned} \quad (8.65)$$

Figure 8.6 Two Josephson junctions in parallel forming a Superconducting Quantum Interference Device, or SQUID.



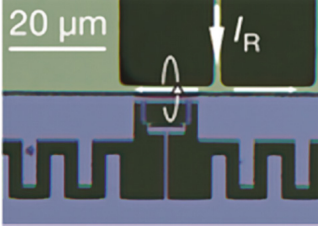


Figure 8.7 Example geometry for applying a current to tune a SQUID-type transmon qubit. The interdigital pattern is a capacitance in parallel with a SQUID. A current is applied to create a magnetic field linking the SQUID, appearing as the small “Y”-shaped structure near the center of the interdigital structure. *Source:* [59].

or

$$\phi_2 = \phi_1 + \frac{2\pi\Phi_B}{\Phi_0}. \quad (8.66)$$

Here we have used Stoke’s Theorem to relate the closed line integral to an integral over the surface enclosed by the line contour, the total flux through the loop from the uniform magnetic field is $\Phi_B = BA$ (neglecting the flux contribution from the induced supercurrent), $\Phi_0 = h/(2e)$ is the flux quantum, and $e^* = -2e$. Assuming the junctions are identical, the total current through the SQUID is

$$\begin{aligned} I_T &= I_c (\sin \phi_1 + \sin \phi_2) \\ &= I_c \left(\sin \phi_1 + \sin \left(\phi_1 + \frac{2\pi\Phi_B}{\Phi_0} \right) \right). \end{aligned} \quad (8.67)$$

For a given value of Φ_B , there is a value of ϕ_1 that maximizes I_T . This maximum value is the effective critical current of the SQUID and is given by (see Exercise 8.5)

$$I_{Tc}(\Phi_B) = 2I_c \left| \cos \left(\frac{\pi\Phi_B}{\Phi_0} \right) \right|. \quad (8.68)$$

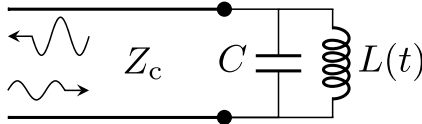
Since the natural frequency of a Josephson junction resonant circuit depends on the critical current (see (8.52)), the frequency of the qubit can be tuned by varying the strength of an applied magnetic field. This provides one way to control the coupling between qubits, since we have seen that weakly-coupled qubits interact most strongly when they have the same frequency.

An example of a tunable SQUID qubit is shown in Figure 8.7 [59]. The current I entering the structure at the top is split along two paths. The left of the two paths passes near a SQUID structure so that magnetic flux generated by the current links the SQUID allowing it to be tuned.

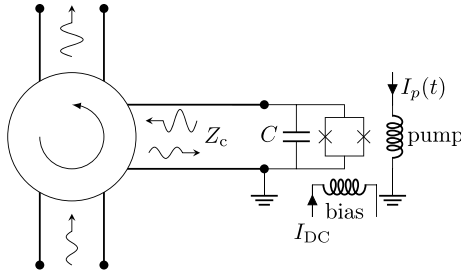
8.4 Josephson Junction Parametric Amplifiers

As the name suggests, a parametric amplifier is a device that amplifies a particular quantity by varying the value of a system parameter in just the right way. Perhaps the most common example is “pumping” your legs to get height on a swing. The body center-of-mass is shifted in such a way that it adds energy to the motion when the swing is at its highest positions. The person swinging supplies the energy to do this.

As a circuit example, consider the LC resonant circuit shown in Figure 8.8(a). As we saw in Chapters 5 and 6, at the resonance frequency, the energy oscillates back and forth between being stored in the capacitor and the inductor. Suppose that whenever the magnetic flux in the inductor is maximum, we increased the inductance. Since the



(a) General scheme of a parametric amplifier in which the inductance terminating a transmission line is varied.



(b) Scheme for implementing a variable inductance parametric amplifier using a flux biased Josephson junction as the variable inductor, based on [186]. Since the amplified signal returns to the input port, a circulator is used to separate the output from the input.

Figure 8.8 Parametric reflection amplifiers using variable inductors.

stored energy is $LI^2/2$, if we increase L for a given I , we will add energy to the system. Note that the energy is stored in the inductor twice each cycle: once with the current in either direction. As a result, if we want to get the maximum amplification effect, we want to increase the inductance twice per cycle of the resonance frequency. Said differently, we want the pump frequency to be twice the resonance frequency. Since an ideal device has zero loss, the only noise introduced by the parametric amplifier is noise associated with the pump signal source. Consequently it is possible to obtain noise performance very near the quantum limit, and since there is no dissipation in the device, it can be kept cold at the lowest temperatures in a dilution refrigerator.

An implementation using a flux-biased SQUID is shown in Figure 8.8(b) [60]. The bias point for the SQUID is set with a DC current in the bias coil, while a variation in the inductance is induced with an AC current at twice the resonance frequency in the pump coil. As observed above, we can model the SQUID as a tunable Josephson junction, and will refer to it as such. An input signal at the resonant frequency of the circuit creates currents circulating between the Josephson junction and the capacitor. Because energy is added by changing the inductance value at the proper time, the reflected signal can be larger than the input signal, and amplification is achieved. A circulator is used to separate the output signal from the input signal since the amplified signal is reflected to the same transmission line used for the input. Additional detail about the operation of Josephson junction parametric amplifiers can be found in [61] and [35].

8.5 Exercises

8.1 Starting with Eqs. (8.21) and (8.22), show that

$$|R|^2 + |T|^2 = 1, \quad (8.69)$$

i.e., the probability that the particle is reflected plus the probability that the particle is transmitted must equal one.

- 8.2** A certain Josephson junction has a critical current of $I_c = 20$ nA and is shunted by a capacitance of $C = 60$ fF, which is much greater than the junction capacitance.
- (a) Calculate the single electron charging energy, E_C .
 - (b) Calculate the Josephson energy, E_J .
 - (c) What is the resonant frequency of the circuit?
 - (d) Calculate the anharmonicity expressed as a frequency, $\delta/(2\pi\hbar)$.
- 8.3** A certain Josephson junction has a critical current of $I_c = 15$ nA. What capacitance is needed in parallel with the junction to create a resonant circuit at 6 GHz?
- 8.4** A SQUID is formed using two identical Josephson junctions with critical current $I_c = 20$ nA.
- (a) What is the effective Josephson energy E_J if no magnetic flux links the circuit?
 - (b) What magnetic flux Φ_B should be applied to reduce the effective Josephson energy to 0?
- 8.5** Starting from the expression for the total current through a SQUID (8.67), show that for a given magnetic flux Φ_B , the maximum value of the current I_T is given by (8.68).

9

Errors and Error Mitigation

9.1 NISQ Processors

As we have mentioned before, the first generation of quantum computers is known as **NISQ** (Noisy Intermediate-Scale Quantum), a term coined by John Preskill [62]. *Intermediate-scale* refers to the number of qubits available in fully-programmable systems over the next several years (i.e., the 2020s), in the range of fifty to a few thousand. The number 50 is important, because it is generally considered the point at which classical computers¹ cannot use brute force simulation to replicate the behavior of a quantum system. Therefore, these systems are large enough to do something interesting, potentially demonstrating a computational advantage over classical computing, but are not large enough to employ error correction (see Chapter 10) for long-running, fault-tolerant quantum computing.

In this chapter, we concentrate on the N in NISQ: *Noisy*. The ability to manipulate matter at a quantum scale is nothing short of remarkable. But the qubits and our ability to control them are not perfect. Theory tells us what will happen in a *closed* system, with no interaction with the surrounding environment, but this can never be achieved in practice. *Noise* is a generic term that refers to anything that leads to non-ideal behavior of the system under consideration, including: environmental interactions, imprecise control of qubits, non-ideal measurements, and undesired interactions between qubits. We describe various types of noise that lead to errors in superconducting quantum computers, and measures that can be taken to mitigate the effect of errors, leading to higher-fidelity results.

Error mitigation is different from the quantum error *correction* techniques described in Chapter 10. Error correction actively detects and corrects errors during the operation of the quantum circuit, allowing quantum programs to run for long times in a fault-tolerant manner. Error mitigation, on the other hand, does not extend the depth of the circuit beyond the limits imposed by decoherence times. It tries to either suppress the effects of noise in the physical implementation of a circuit, or to improve the interpretation of the measured results, making them as close as possible to what would have been observed in an ideal, error-free quantum system. The former case (suppressing the effects of noise) is also known as *noise mitigation*.

¹ Again, using 2020s-era systems.

Mitigation techniques are critical for getting useful results from NISQ computers, and noise suppression is an important component of the path toward the development of highly-reliable, scalable systems required for fault-tolerant quantum computing.

9.2 Decoherence

The term *decoherence* refers to any loss of quantum information due to interaction with the environment surrounding a qubit (or quantum system). It is synonymous with *quantum noise*. There are many sources of decoherence, and it cannot be avoided for any physical quantum system, because it is impossible to completely isolate the system from the environment. For superconducting qubits, common sources of noise include fluctuations in the Josephson energy of the junctions, gate charge fluctuations, magnetic field fluctuations, and interaction with photons and phonons [63, 64].

The decoherence of a single qubit is quantified by two “decoherence times,” known as T_1 and T_2 . T_1 , known as the *relaxation time*, is the average time it takes for a qubit in the excited state $|1\rangle$ to spontaneously relax to the ground state $|0\rangle$. This process is also known as depolarization or decay. T_2 is the *dephasing time*, the time it takes for a superposition state to flip the relative phase between its $|0\rangle$ and $|1\rangle$ components.

Figure 9.1 illustrates an experimental method to measure T_1 . First, we prepare the qubit in state $|1\rangle$, using an X gate, and then we delay for a certain time (Δt) before measuring. By increasing Δt , we increase the probability that the state will relax from $|1\rangle$ to $|0\rangle$. The experiment is repeated many times for each value of Δt , to get a good average of measurements, and we report the percentage of measurements that return “1” for each delay time.

The right side of the figure shows the results for one qubit of a superconducting quantum computer.² As the delay increases, the percentage of ones drops exponentially from 100% (no decay) to 0% (complete decay). The data points are fitted to a curve of the following form:

$$f(t) = Ae^{-t/T_1} + B \quad (9.1)$$

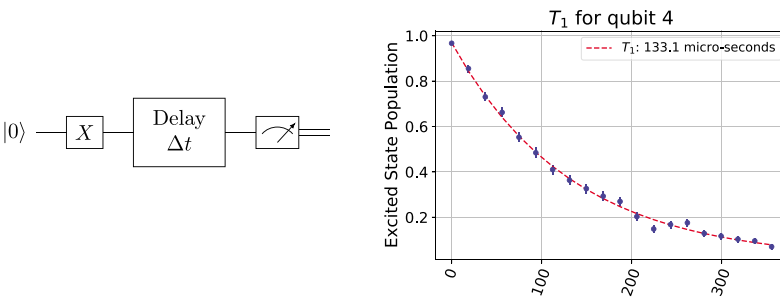


Figure 9.1 Measurement of T_1 relaxation time. The horizontal axis in the graph is delay in microseconds.

² All experimental results in this chapter were run on IBM Quantum hardware. Hardware: IBMQ Jakarta, seven qubits, Falcon r5.11H processor. Software: `qiskit` 0.28.0, `qiskit-terra` 0.18.0, `qiskit-ignis` 0.6.0. Experiments were run in July 2021.

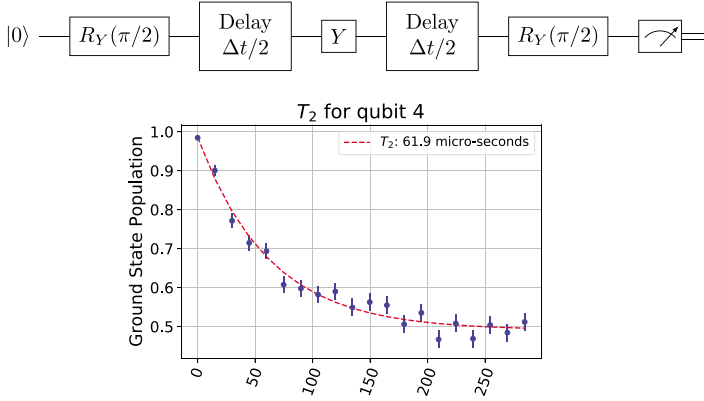


Figure 9.2 Measurement of T_2 dephasing time. The horizontal axis in the graph is delay in microseconds.

The time T_1 represents the average time to decay to 50%, meaning that the initial $|1\rangle$ has an equal chance of being measured as 1 or 0. The current generation of IBM hardware exhibits T_1 times around 50–150 microseconds.

Figure 9.2 illustrates an experimental method to measure T_2 . There are a few different measurement techniques, and this is the one used by the Qiskit Ignis software for qubit characterization on IBM’s hardware. This particular approach is known as the CPMG (Carr-Purcell-Meiboom-Gill) method [65, 66].

Starting with state $|0\rangle$, we first rotate by $\pi/2$ around the Y axis, resulting in state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. We then delay for an arbitrary time $\Delta t/2$, followed by a π rotation around the Y axis. In the absence of dephasing noise, this would take the state to $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. After another delay of $\Delta t/2$, we apply another $R_y(\pi/2)$ rotation. If there is no dephasing, this would bring the state back to $|0\rangle$. (The delay is denoted as $\Delta t/2$ so that the total delay is Δt , comparable to the T_1 experiment described above.)

In the presence of dephasing noise, the state precesses around the Z axis during the delays. Consider the case where the state moves clockwise around the axis. The second $\pi/2$ rotation then brings the state back toward $|0\rangle$, but it is not lined up directly with the Z axis, which introduces a probability of measuring 1.

As before, the experimental data is fitted to an exponential curve, this time with T_2 as the time constant:

$$f(t) = Ae^{-t/T_2} + B \quad (9.2)$$

Why does the curve decay to 50% and not zero? During the delay, the superposition state is also subject to T_1 relaxation. If the state decays to $|0\rangle$, then the second $\pi/2$ rotation takes the state to $|+\rangle$, resulting in an equal chance of measuring 1 or 0. In contrast, the states $|0\rangle$ and $|1\rangle$ are not sensitive to phase change, so dephasing does not impact the results of the T_1 experiment.

Decoherence times are a key metric for a quantum system, because they directly impact the *depth* of quantum circuits that are reasonable—that is, the number of quantum gates that can be performed before decoherence makes the results too random to be useful. For the hardware used in this chapter, the latency of a single-qubit gate is

around 35ns, and the latency of a two-qubit CNOT ranges from 350ns to 550ns, depending on the qubit pair. With a decoherence time in the 100 μ s range, one could expect to execute a few hundred gates with a reasonable chance of meaningful results.

9.3 State Preparation and Measurement Errors

In addition to decoherence, caused by environmental interactions, quantum systems experience errors during state preparation, gates, and measurements. Gate errors will be discussed in the next section, so here we will concentrate on state preparation and measurement errors, also known as *SPAM* errors. A state preparation error occurs when a qubit is initialized to $|0\rangle$, but ends up in the $|1\rangle$ state instead. A measurement error occurs when a $|0\rangle$ state is measured as 1, or vice versa. Measurement errors are typically larger than state preparation or gate errors, partially because the duration of a measurement is relatively long, and decoherence can occur during the process. In practice, it is usually not possible to separate state preparation from measurement errors [67]. For example, suppose we prepare a state $|0\rangle$ and measure 1. Is this evidence of a state preparation error, or a measurement error? Since measurement errors are more likely, we will ignore state preparation and assume all SPAM errors are actually measurement errors.

We can calibrate SPAM errors simply, by preparing each possible state for n qubits and then measuring it many times. We store the outcomes—the percentage of occurrences of each possible output—in a matrix M_e , shown in Figure 9.3. Each column j shows the measured outputs after preparing state $|j\rangle$. In other words, matrix element (i, j) is the probability of measuring i after preparing $|j\rangle$. There is no delay between preparation and measurement, and there is at most one X gate for each qubit, so these errors can be attributed to SPAM.

The matrix M_e can be used to adjust the measured outputs of other circuits to compensate for measurement errors. Let v be the vector of measurements from the execution of a circuit C : each element v_i is the percentage of occurrences of measurement i . Let x be the vector of *mitigated* results, which is intended to correct the measured outputs to be closer to the *exact* results that would have been measured in the absence of any SPAM errors.

0.804	0.024	0.044	0.001	0.054	0.002	0.001	0.
0.010	0.777	0.002	0.035	0.	0.044	0.	0.001
0.053	0.	0.832	0.019	0.002	0.	0.05	0.
0.001	0.062	0.010	0.840	0.	0.001	0.001	0.037
0.122	0.004	0.004	0.	0.872	0.019	0.059	0.001
0.	0.123	0.	0.007	0.009	0.859	0.	0.056
0.009	0.001	0.106	0.002	0.061	0.002	0.878	0.020
0.001	0.009	0.002	0.096	0.002	0.073	0.011	0.885

Figure 9.3 Measurement error calibration matrix M_e for three qubits. Element (i, j) shows the probability of measuring i after preparing state $|j\rangle$. Thus, each column j corresponds to the outcomes when preparing state $|j\rangle$. Diagonal elements show probabilities of correct measurement, while off-diagonals represent probabilities of errors.

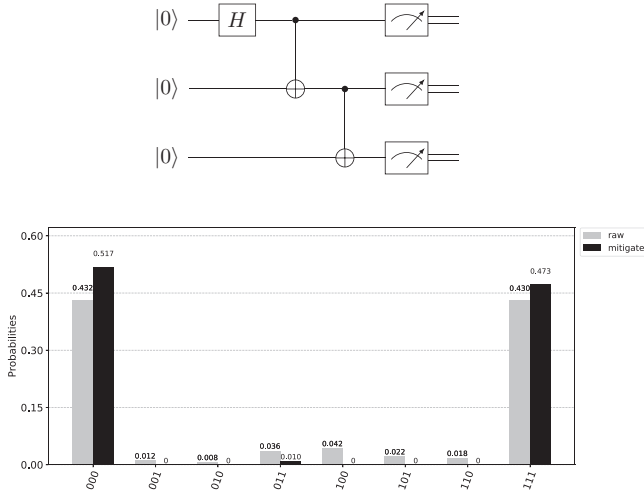


Figure 9.4 Mitigating measurement errors using the calibration matrix from Figure 9.3. Prepared state is $(|000\rangle + |111\rangle)/\sqrt{2}$.

We can think of the matrix M_e as an error operator that takes the desired output vector x to the measured output vector v : $M_e x = v$. However, if we invert M_e to solve for x , we may end up with values outside of the range $0 \leq x \leq 1$ [68]. To avoid this problem, we use the *least squares* method. Change the problem to finding the minimum of the following function:

$$f(x) = \sum_{i=0}^{2^N-1} (v_i - (M_e x)_i)^2, \quad (9.3)$$

given the constraints $0 \leq x_i \leq 1$ and $\sum x_i = 1$.

To demonstrate the use of the error matrix, consider the experiment in Figure 9.4, in which we prepare the three-qubit state $(|000\rangle + |111\rangle)/\sqrt{2}$. We ran the circuit on quantum hardware and obtained the results in the light gray bars; this is vector v in the equation above. The M_e for this system comes from the calibration experiment shown in Figure 9.3. Solving for x gives the mitigated results, shown by the dark gray bars. There are still errors in the mitigated results, but the frequency of correct measurements has increased and many of the errors have been eliminated.

9.4 Characterizing Gate Errors

Errors are also created by imperfect quantum gates. Recall that a gate is a unitary operation performed on some number of qubits; for our purposes, we will restrict our attention to one- and two-qubit gates, because those are currently the only gates directly implemented by quantum hardware. For transmon qubits, gates are performed using control pulses that interact with the circuit to move it to the desired state. Due to imprecise control and noise in the system, the pulse can be slightly off in terms of frequency or amplitude, or the pulse signal may contain undesired frequency components. Any

of these factors, as well as manufacturing defects, can result in incorrect behavior of the gate operation.

Full characterization of the behavior of a gate can be done through a technique known as *quantum process tomography*. A known set of quantum states are used to observe the dynamics of the system. (For a full description, see Chapter 8 of Nielsen and Chuang [9].) However, the cost of process tomography scales exponentially with the number of qubits, so it is prohibitively expensive for systems with more than a few qubits.

Randomized benchmarking (RB) is proposed as a scalable method to partially characterize the error rate of applying quantum gates [69]. The general idea is to apply a sequence of randomly-chosen gates that combine to perform the identity operation, and to measure the fidelity of the result as the number of gates in the sequence increases. If the starting state is $|0\rangle^{\otimes n}$, and the gates are perfect, then the ending state will also be $|0\rangle^{\otimes n}$. If the gates have errors, then the percentage of results that are measured in the desired state decays exponentially with the length of the sequence. The rate of decay is related to the gate error rate.

Figure 9.5 shows the structure of an RB experiment. Each operator C_i is selected at random from a well-defined group of operators. In this case, we use the *Clifford* group, generated by the set $\{H, S, \text{CNOT}\}$, where H is the Hadamard gate and S is the phase gate ($\pi/2$ rotation around the Z axis). The Pauli gates can all be generated using H and S , so they are also included in the set of Clifford operators. (For simplicity, we will call these Clifford “gates,” even though each operator may contain multiple basic gates of the type discussed in Chapter 1.) The advantage of using the Clifford group is that it is easy to calculate the final operator, C^{-1} , that performs the inverse of all the preceding operators.

For each value m , a number of random circuits are generated with m Clifford gates plus the final inverting gate that should bring the state back to all zeroes. Each circuit is run multiple times to determine the fidelity, which in this case is the percentage of measurements that yield all zero. Then the fidelity of each random circuit of size m is averaged.

There is a fair amount of math involved, but the end result is that the experimental data is then fit to a curve of the following form:

$$f(x) = A_0\alpha^m + B_0. \quad (9.4)$$

The error rate r per Clifford gate, also known as *Errors per Clifford (EPC)*, is related to α as follows:

$$r = 1 - \alpha - \frac{1 - \alpha}{2^n} = \frac{2^n - 1}{2^n}(1 - \alpha), \quad (9.5)$$

where n is the number of qubits. The constants A_0 and B_0 account for SPAM errors.

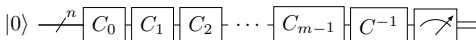


Figure 9.5 Randomized benchmarking circuit. A random selection of m operators is followed by a single operator representing the inverse of the previous m operators. With no errors, the output will be $|0\rangle^{\otimes n}$.

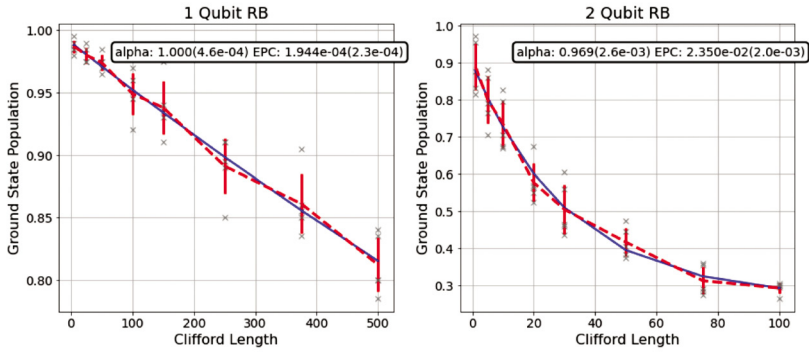


Figure 9.6 Randomized benchmarking experiment on IBM Q hardware.

Figure 9.6 shows the results of an RB experiment using the same IBM quantum system as the other experiments in this chapter. On the left is the one-qubit experiment. The number of Clifford gates (m) varies from 1 to 500. For each number of Clifford operators, five different random circuits were generated. The red (dashed) line connects the average percentage of zeroes in the results, and the blue (solid) line shows the exponential fit. The graph shows $\alpha = 1.000$, but that’s a rounding issue. The corresponding EPC is 1.994×10^{-4} . The numbers in parentheses are the calculated errors for the fit.

On the right-hand side is the two-qubit RB experiment. In this case, the Clifford gate includes both single-qubit and CNOT gates. Because two-qubit gates have both higher latency and higher error rates, we only vary the number m from 1 to 100. The two-qubit EPC is reported as 2.350×10^{-2} , two orders of magnitude higher than the single-qubit EPC.

Note that each Clifford gate is implemented using one or more *native* gates provided by the quantum hardware. How does EPC relate to the error rate of those native gates? The hardware used for these experiments provides the following native gates: $R_Z(\theta)$, X , CNOT, and $\sqrt{X}$, defined as:

$$\sqrt{X} = \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}. \quad (9.6)$$

In other words, all gates are decomposed by the transpiler into sequences of these basic operations.

In the single-qubit case, we can analyze the circuits used in the experiment to figure out how many of each gate are used for each Clifford gate. For the experiment in Figure 9.6, the values are shown in the table below.

Gate	Avg. per Clifford
R_Z	3.885
X	0.246
$\sqrt{X}$	1.866

In IBM’s hardware, Z -rotations are performed “virtually” by adding a phase offset to subsequent control pulses. These are essentially zero-latency and zero-error, so we will simply ignore them in the following calculation. The X and $\sqrt{X}$ are both implemented with a single pulse of the same duration but different amplitudes. In terms of errors,

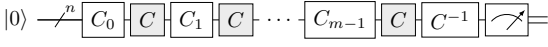


Figure 9.7 Interleaved Randomized Benchmarking (IRB) adds a specific Clifford gate, C , after each randomized gate. The relationship between the exponential curves with and without C characterize the error rate for C .

they are expected to be essentially the same, so we will bundle them together. Therefore, we have $g = 2.112$ gates per Clifford.

Defining EPC in terms of errors per gate (EPG), where $EPG \ll 1$:

$$\begin{aligned} EPC &= 1 - (1 - EPG)^g \\ &\simeq EPG \cdot g, \end{aligned} \tag{9.7}$$

which means $EPG = EPC/2.112 = 9.21 \times 10^{-5}$.

For the two-qubit experiment, we want to find the EPG rate for the CNOT gate. As before, we analyze the circuits to count the single-qubit and two-qubit gates on both qubits.

Gate	Avg. per Clifford	
R_Z	7.727	7.531
X	0.258	0.251
$\sqrt{X}$	3.795	3.691
CNOT	1.484	1.484

Because the error of the CNOT dominates, we can approximate its error rate by $EPC/1.484 = 1.584 \times 10^{-2}$. Using a more accurate method described by McKay *et al.* [70] (see supplemental material), which takes the one-qubit EPG into account, we get an estimate of 1.555×10^{-2} errors per CNOT gate.

Suppose we are interested in the error rate of a specific gate that is not a native gate. For example, what is the EPG for a Hadamard gate? One approach is to extract information about how this gate is implemented using native gates, and estimate based on the native gate EPG. Another method is called *Interleaved Randomized Benchmarking*, shown in Figure 9.7.

For IRB, we interleave the gate that we want to characterize (C) in between the random gates chosen for the RB sequence. First we run RB to find the parameter α , as above. Then we run the IRB sequence and find a new parameter for the decay of that curve, α_C . The estimated error rate for Clifford gate C is:

$$r_C^{\text{est}} = \frac{(d-1)(1-\alpha_C/\alpha)}{d}, \tag{9.8}$$

where d is 2^n and n is the number of qubits. The error of the estimate is bounded, depending on the assumptions about the quantum error characteristics [71].

9.5 State Leakage and Suppression Using Pulse Shaping

As discussed in Chapter 8, transmons can be modeled as qubits owing to the fact that the nonlinear inductance shifts the frequencies of the excited states so that the energy difference between the ground and first excited state does not match any other level difference. Consequently a monochromatic tone tuned to the energy difference between the ground and excited states will cause Rabi oscillations only between those two states, the higher-energy states can be ignored, and the transmon can be treated as a two-state system.

However, as discussed in Chapter 2, we need to apply pulses of finite width to control the Rabi oscillations. These pulses are not monochromatic, but have spectral width that is inversely proportional to the pulse duration. As a result, it is possible that higher states can be inadvertently excited if the bandwidth of the control pulse also includes frequencies matching transitions to higher transmon states. This phenomenon is known as *state leakage*.

As an example, consider a Gaussian pulse in time with half-width τ :

$$v_G(t) = V_0 e^{-t^2/2\tau^2}. \quad (9.9)$$

The frequency spectrum is obtained by taking the Fourier transform:

$$\begin{aligned} V_G(\omega) &= \int_{-\infty}^{\infty} v_G(t) e^{-j\omega t} dt \\ &= \tau \sqrt{2\pi} V_0 e^{-\omega^2 \tau^2 / 2}, \end{aligned} \quad (9.10)$$

which has the half-width in frequency of $1/\tau$.

However, as discussed in Section 4.8, we are normally interested in modulating a signal whose frequency matches a qubit transition frequency with a shaped pulse. In other words, we are interested in the spectrum of a pulse of the form

$$v_m(t) = v_G(t) \cos \omega_0 t. \quad (9.11)$$

The spectrum of the modulated pulse is easily obtained, given the spectrum of the Gaussian pulse:

$$\begin{aligned} V_m(\omega) &= \int_{-\infty}^{\infty} v_G(t) \cos \omega_0 t e^{-j\omega t} dt \\ &= \int_{-\infty}^{\infty} v_G(t) \frac{1}{2} (e^{-j(\omega-\omega_0)t} + e^{-j(\omega+\omega_0)t}) dt \\ &= \frac{1}{2} V_G(\omega - \omega_0) + \frac{1}{2} V_G(\omega + \omega_0). \end{aligned} \quad (9.12)$$

We see then that the energy is spread about the qubit frequency of ω_0 with half-width $1/\tau$. If the transition frequency from $|1\rangle \rightarrow |2\rangle$ lies within this range, then there is some probability of exciting this third state, and the system will expand beyond the qubit states, causing errors.

This suggests putting a notch in the spectrum of the pulse at this particular frequency, to minimize exciting this transition. One way to do this is to apply a pulse with Gaussian modulation to a cosine signal, the derivative of the Gaussian pulse modulated on a sine signal, and sum the two waveforms [72–76]. This approach is referred to as the

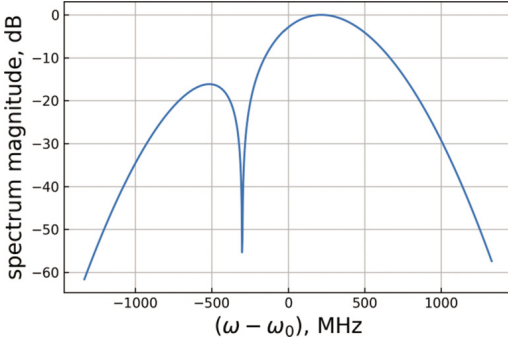


Figure 9.8 Optimized spectrum with a notch at the anharmonicity value of -300 MHz.

Derivative Removal by Adiabatic Gate protocol, or DRAG. Specifically, we are interested in the spectrum of the pulse waveform

$$v_D(t) = v_G(t) \cos \omega_0 t + \alpha \frac{dv_G}{dt} \sin \omega_0 t, \quad (9.13)$$

where α is a parameter controlling the location of the notch in the spectrum. Following a procedure similar to that of (9.12), we find that

$$\mathcal{F}\{v'_G\} = \frac{1}{2j} V'_G(\omega - \omega_0) - \frac{1}{2j} V'_G(\omega + \omega_0) \quad (9.14)$$

where $v'_G = dv_G/dt$ and $V'_G(\omega) = \mathcal{F}\{v'_G(t)\}$. From the properties of the Fourier transform³ we have

$$\mathcal{F}\{v'_G\} = j\omega V_G(\omega). \quad (9.15)$$

The positive-frequency portion of the spectrum of the optimized waveform (9.13) is therefore

$$V_D(\omega)|_{\omega>0} = V_0 \tau \sqrt{\frac{\pi}{2}} (1 + \alpha(\omega - \omega_0)) e^{-(\omega - \omega_0)^2 \tau^2 / 2}. \quad (9.16)$$

From Section 8.2.3, for a transmon with anharmonicity δ , the frequency of the transition $|1\rangle \rightarrow |2\rangle$ is $\omega_0 - \delta/\hbar$ if the frequency of the transition $|0\rangle \rightarrow |1\rangle$ is ω_0 . Requiring $V_D(\omega_0 - \delta/\hbar) = 0$ gives

$$\alpha = \frac{\hbar}{\delta}. \quad (9.17)$$

This spectrum is plotted in normalized dB in Figure 9.8 (i.e., $V_D(\text{dB}) = 20 \log_{10}(|V_D|/\max V_D)$) for the case of anharmonicity expressed as a frequency $(\delta/(2\pi\hbar))$ of 300 MHz. The notch at the desired frequency is readily apparent.

9.6 Zero-Noise Extrapolation

In the event that noise cannot be removed from the quantum system, it would be nice if there were a way to remove the effects of the noise from the results of the computation. A technique known as *zero-noise extrapolation* [78–80] does this by artificially

³ See, for example, [77].

increasing the impact of noise in a controlled way, and then extrapolating back to the expected outcome if there were zero noise. The following description is largely adapted from Kandala *et al.* [81].

Any quantum circuit can be described by evolution under a *drive Hamiltonian*:

$$K(t) = \sum_{\alpha} J_{\alpha}(t) P_{\alpha}, \quad (9.18)$$

for a time T , where each P_{α} represents an N -qubit Pauli operator and J_{α} is the strength of the associated interaction. At time T , suppose that the expectation value of an observable is E^* . An estimate of that observable in the presence of noise $\lambda \ll 1$ can be written as a power series

$$E_K(\lambda) = E^* + \sum_{k=1}^n a_k \lambda^k + O(\lambda^{n+1}), \quad (9.19)$$

where coefficients a_k depend on the specific details of the noise model.

We are interested in $E^* = \lim_{\lambda \rightarrow 0} E_K(\lambda)$, but we can only observe the noisy estimate. If we can scale λ in a controlled way to make additional observations, Richardson's deferred approach to the limit [82] can be used to suppress higher-order noise terms in (9.19). For this reason, the technique is also known as *Richardson's extrapolation*.

If we can obtain n additional estimates $\hat{E}_K(c_i \lambda)$ by precisely amplifying the noise rate by factors c_i for $i = 1, \dots, n$, we can improve the estimate with reduced error of $O(\lambda^{n+1})$:

$$\hat{E}_K^n(\lambda) = \sum_{i=0}^n \gamma_i \hat{E}_K(c_i \lambda). \quad (9.20)$$

For a given choice of c_i , coefficients γ_i are solutions to $\sum_{i=0}^n \gamma_i = 1$ and $\sum_{i=0}^n \gamma_i c_i^k = 0$ for $k = 1, \dots, n$.

How do we precisely increase noise λ by c_i , also known as *stretch factors*? We can add redundant gates to the circuit [79], but that requires that the noise characteristics of the gates be precisely understood. Instead, we can focus on the Hamiltonian description in Eq. (9.18). If the noise is time-invariant, we can “stretch” the time to $c_i T$ and also scale the interaction parameters to retain the original evolution. For stretch factor c_i , the scaled Hamiltonian $K^I(t)$ is derived as follows:

$$K^I(t) = \sum_{\alpha} J_{\alpha}^I(t) P_{\alpha} \quad (9.21)$$

$$J_{\alpha}^I(t) = \frac{J_{\alpha}(t/c_i)}{c_i}. \quad (9.22)$$

In practice this means that we scale the duration of the control pulse for P_{α} by a factor of c_i , and we scale the amplitude by $1/c_i$. The term “time-invariant” in this context means that the noise can be described by a dynamical equation that is not sensitive to time scaling, such as constant-amplitude relaxation and dephasing noise (Section 9.2) or other noise caused by constant coupling with the environment. This is a reasonable approximation for quantum circuits that are well within the system's decoherence limits.

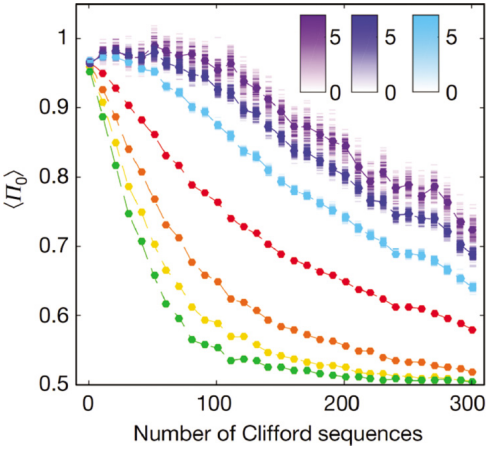


Figure 9.9 Error-mitigated results for a single-qubit randomized benchmarking circuit. The four lower curves are the measured results with stretch factors $c = 1$ (red), $c = 2$ (orange), $c = 3$ (yellow), and $c = 4$ (green). The upper curves are the Richardson extrapolations to the first (light blue), second (dark blue), and third order (violet). The color bars represent histogram frequencies of 100 data points for each sequence size. *Source:* [81].

Figures 9.9–9.11 show experimental evidence provided by Kandala *et al.* on the effectiveness of the technique. Experiments were performed on a five-qubit IBM system. It should be noted that the two-qubit CNOT gates in these experiments were implemented with significantly longer latency than usual, in order to avoid nonlinear interactions between qubits and to remain in range of time-invariant noise required by the technique [81].

Figure 9.9 shows results from a randomized benchmarking experiment, as described in Figure 9.5. The middle curve (red) shows the unscaled outcome ($c = 1$), while the lower curves show the effects of increasing noise with scaling factors 2, 3, and 4. The upper curves show the mitigated results using varying numbers of the scaled results for the extrapolation.

A single-qubit experiment is shown in Figure 9.10. In this case a sequence of unitaries U_j are applied to a starting state of $|0\rangle$. In the absence of errors, the state moves

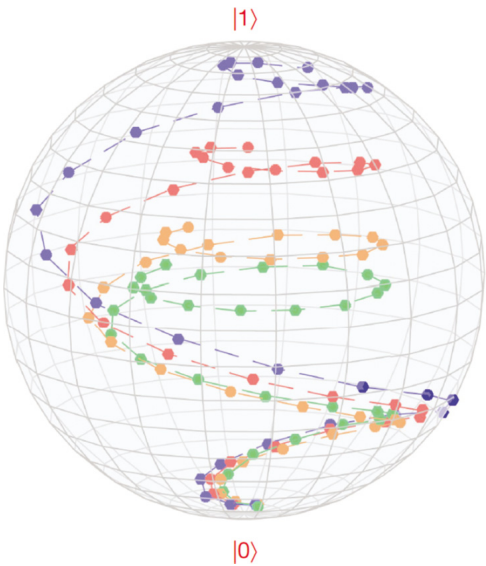


Figure 9.10 Error-mitigated results for traversal of a path along the Bloch sphere. The first-order mitigated result (blue) reaches the target state of $|1\rangle$, while the experimental results show the effects of decoherence. Stretch factors are $c = 1$ (red), $c = 2$ (orange), and $c = 3$ (green). *Source:* [81].

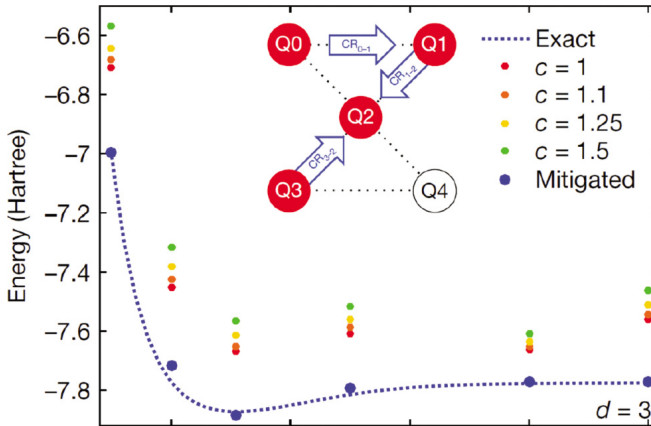


Figure 9.11 Error-mitigated results for finding the ground state energy of a lithium-hydride (LiH) molecule using VQE on a four-qubit system. Stretch factors are 1, 1.1, 1.25, and 1.5, and the mitigated result uses first-order Richardson extrapolation. The inset shows the topology of the qubits involved in the experiment. *Source:* [81].

along the surface of the Bloch sphere and should go to $|1\rangle$ in the last step. The position on the Bloch sphere is computed by measuring expectation values $\langle X \rangle$, $\langle Y \rangle$, and $\langle Z \rangle$ for each unitary. Stretch factors $c = 1, 2, 3$ show increasing effects of decoherence, while the mitigated result (dark blue) does reach the $|1\rangle$ state.

Figure 9.11 shows the results from a quantum chemistry calculation, finding the ground state energy of a lithium hydride (LiH) molecule at different interatomic distances using the variational quantum eigensolver (Section 12.6.1). The VQE circuit includes both single-qubit and two-qubit gates. The mitigated results are much closer to the classically-computed known exact values for the ground state energy.

9.7 Optimized Control Using Deep Learning

In this final section, we briefly introduce recent work on using Deep Reinforcement Learning (DRL) to optimize the control pulses for one- and two-qubit gates on a superconducting quantum computer [83], led by Q-CTRL. The DRL agent is able to optimize gates without knowing the detailed Hamiltonian of the system, its controls, or the underlying error processes. This is a promising step toward automatic calibration and control of large-scale quantum systems.

Quantum control is a discipline that seeks to manipulate systems that obey the laws of quantum mechanics to create desired behaviors [84]. Growing from early work in nuclear magnetic resonance (NMR) and other technologies, current efforts involve control of experimental quantum systems in computing, sensing, and communication. In the context of quantum computing, the emphasis is to enhance the reliability and robustness of qubits and gates.

Traditional methods of quantum optimal control require precise knowledge of the Hamiltonian of the system and the noise processes that act on the system. As we've

seen in this chapter, it becomes more and more difficult to fully characterize systems as they scale, and it is even more challenging to deal with noise sources that we have not discussed: known and unknown transients, control signal distortion in transmission, cross-talk among qubits, and so forth.

Instead of beginning with a full and robust mathematical model, we can use the standard machine learning approach: (1) observe the state of the system through measurement, (2) choose an action to perform, in this case a control pulse to be applied, and (3) measure the new state and determine the *reward* resulting from the previous action. The DRL agent uses a deep neural network to maximize the cumulative reward over many trials.

The control system is represented by a piecewise constant (PWC) Hamiltonian, divided into N_{seg} segments of width Δt :

$$H_{ctrl}(t) = \sum_{k=0}^{N_{seg}} H_k \mathbb{1}_k(t), \quad (9.23)$$

where $\mathbb{1}_k = 1$ if $(k-1)\Delta t \leq t < k\Delta t$ and zero otherwise. We observe the state after step k and have the agent decide what action to take in step $k+1$. Because measurement is destructive, the system has to perform steps 0 through k each time.

Observation is based on quantum state tomography. States are prepared in each of the Pauli bases for single-qubit gates, and all two-qubit Pauli strings for two-qubit gates,

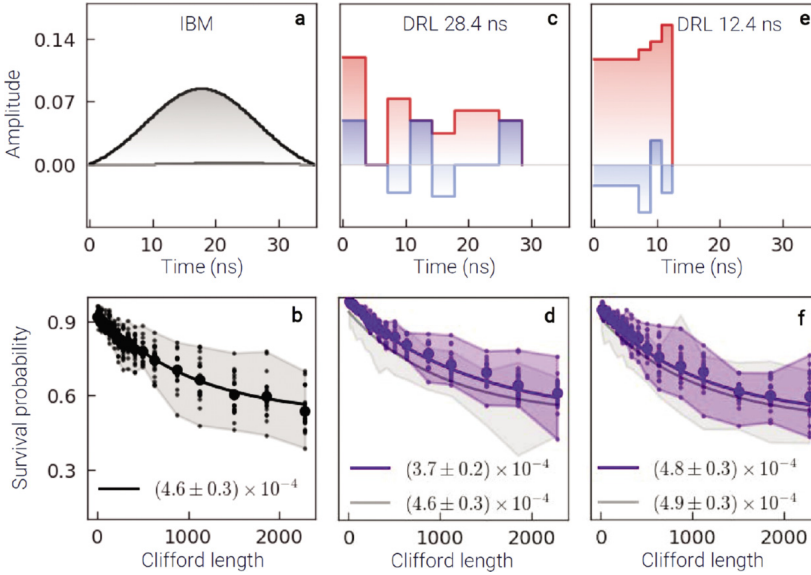


Figure 9.12 Sample control pulses generated by deep reinforcement learning (DRL) agent for the $R_X(\pi/2)$ gate. (a) Standard IBM DRAG pulse with a duration of 35ns. (b) Result of randomized benchmark characterization. The gray area shows the spread of data points and the black line is the exponential fit. The number is the errors per gate (EPG). (c)–(d) Stepwise DRL pulse (28.4ns) and RB results. Different colors represent the real (in-phase) and imaginary (out-of-phase) components of the waveform. RB fidelity is slightly better and spread of data is slightly smaller. (e)–(f) Stepwise DRL pulse with a 12.4ns duration, with similar fidelity. *Source:* [83].

and expectation values are calculated. The reward is computed as the fidelity once the full waveform is composed.

Figure 9.12 shows two example pulses generated by the DRL agent for the $R_X(\pi/2)$ gate, compared to the standard IBM DRAG pulse on the left. The stepwise nature of the DRL pulses is evident. The 28.4ns pulse (c) has a slightly better EPG than the IBM pulse, with a somewhat tighter range of results. The pulse on the right (e) is nearly one-third the latency of the standard IBM pulse, with essentially the same fidelity.

In addition, the DRL-generated pulses were found to be robust across variations in system parameters. IBM gates are recalibrated and can show variances in amplitude on the order of several percent from day to day. The DRL pulses were evaluated on the same machines over a period of two weeks, with no recalibration, and showed consistent performance relative to the default gates, with measured EPG tracking fluctuations in hardware T_1 measurements taken over that time period.

9.8 Exercises

- 9.1 Consider the T_2 measurement experiment in Figure 9.2. If noise causes a z -rotation of $\pi/16$ in a counter-clockwise direction during the delay of Δt , what is the probability of measuring zero?
- 9.2 Assume a fixed delay of 50 nanoseconds for any one- or two-qubit gate. Assuming no gate errors and ignoring dephasing, what should T_1 be in order to execute a quantum circuit with a depth of 100, with an 85% probability of measuring the correct result?
- 9.3 Consider a quantum circuit with four qubits, where each layer of the circuit contains at least one CNOT gate. If the CNOT error rate is 0.015, what is the depth of the circuit that can be executed with an overall 85% success rate? (Assume that gate error is the only source of noise, ignoring decoherence time.)
- 9.4 Consider the implementation of a three-qubit Toffoli gate, shown in Figure 9.13. Given the per-gate error rates from Section 9.4, what is the expected error per Toffoli gate?

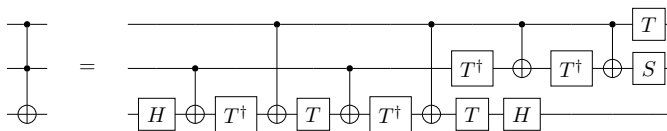


Figure 9.13 Implementation of a three-qubit Toffoli gate (see Section 11.1.1).

10

Quantum Error Correction

In Chapter 9, we discussed various sources of noise in quantum systems, and how to reduce and compensate for noise. However, noise cannot be completely eliminated. Some noise will cause *errors* in the quantum information used in our computations. In this chapter, we discuss how qubit errors can be detected and corrected, leading to the development of *fault-tolerant quantum computing*.

To deal with errors, we encode information using additional qubits. For example, we might use 12 qubits to represent 2^8 basis states, rather than 8. This allows us to distinguish *valid* encodings, representing the 2^8 “good” states, from *invalid* encodings, representing error states. With enough information in the extra qubits, we can determine which data qubit has an error and how to correct it.

Such *error correction codes* are used extensively in the classical computing realm, particularly for communicating over noisy channels or dealing with leakage errors in memory. Quantum Error Correction (QEC) codes build on these concepts, but additional features are required to deal with the nature of quantum information and errors.

10.1 Review of Classical Error Correction

Errors may occur when transmitting information across a *noisy channel*. A channel is any medium through which information is communicated, either through space (e.g., transmission of bits over a wire) or time (e.g., bits stored in memory). The channel sits between the transmitter of information and the receiver of information, as shown in Figure 10.1.

In classical computing, the only error that matters is a *bit flip*, where 0 is transmitted but 1 is received, or vice versa. In this chapter, we will assume a simple error model, in which each transmitted bit will experience a bit flip error with probability p , as shown in Figure 10.1. Bits are independent of each other, so that an error in one bit has no influence on whether an error occurs in another bit.

In order to protect against errors, we transmit redundant information, which can be used by the receiver to better estimate the information that was transmitted over the noisy channel. We use an *encoder* to transform the intended message by adding information (bits), and a *decoder* to extract the intended message from the received bits, as shown in Figure 10.2.

$$\Pr(0 \rightarrow 1) = \Pr(1 \rightarrow 0) = p$$

$$\Pr(0 \rightarrow 0) = \Pr(1 \rightarrow 1) = 1 - p$$

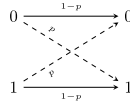


Figure 10.1 Classical noisy channel.



Figure 10.2 Encoding data for error detection/correction: k -bit data is encoded using n bits to transmit over the channel. Decoding detects/corrects errors and transforms back to k bits.

10.1.1 Error Detection

An error detection code (EDC) allows the receiver to determine that an error has occurred, but it cannot determine which bit has flipped. For example, a single *parity* bit can be added to a group of n bits, so that the total number of 1's in the encoded data is always odd. By counting the 1's in the received bits, the decoder can detect a single bit flip. While it knows that an error occurred, it cannot know which bit flipped, since flipping any bit will cause the number of 1's to go from odd to even. Likewise, it will not detect the case where two bits have flipped, since the total number of 1's will still be odd. (A parity bit will detect any odd number of errors.)

Note that it is just as likely for the parity bit to flip as any other bit. This generates a false positive, since there is no error in any of the data bits that we actually care about. It's important to understand that errors may occur within the error detection and correction process itself.

10.1.2 Error Correction: Repetition Code

An error correction code (ECC) provides enough redundant information for the decoder to know which bit has flipped. This allows the error to be corrected, rather than simply detected.

The simplest code that allows correction of a single bit flip is the *repetition code*, in which each message bit is transmitted multiple times. If the error probability p is small enough, we can expect that only a minority of bits will be flipped. Therefore, the decoder can use a “majority vote” to determine the intended value. Any bits that are different than the majority can be flipped to correct the error.

For example, let's choose to repeat each bit three times. To transmit a logical 0, we send 000. To transmit a logical 1, we send 111. If the receiver gets 101, there are two possible reasons: (a) 000 was transmitted, and the first and last bits were flipped, or (b) 111 was transmitted, and the middle bit was flipped.

Which of these scenarios is more likely? Since each bit is independent:

$$\Pr(000 \rightarrow 101) = p \cdot (1 - p) \cdot p = p^2 - p^3 \approx p^2 \quad (10.1)$$

$$\Pr(111 \rightarrow 101) = (1 - p) \cdot p \cdot (1 - p) = p - 2p^2 + p^3 \approx p. \quad (10.2)$$

Since p is small, it's more likely that 111 was transmitted. This is consistent with the majority decoder. The zero bit is flipped to recover the original data 111.

Figure 10.3 Error correction using the 3-bit repetition code.

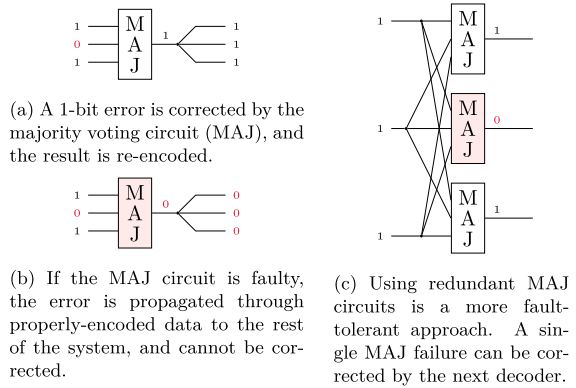


Figure 10.3(a) shows a decoder for the three-bit replication code that uses a majority vote (MAJ) logic circuit. In the figure, the decoded data is re-encoded (replicated) for use in another part of the system. This approach, however, assumes that the MAJ circuit always works correctly. In part (b), the MAJ circuit fails, generating an incorrect output. When encoded, this replicated incorrect output cannot be corrected by the rest of the system because 000 is a legal encoding.

A fault-tolerant approach is shown in Figure 10.3(c). We replicate the MAJ circuit, and each part of the encoded input is replicated and sent to three decoders. If all of the MAJ circuits function correctly, then the circuit will produce 000 or 111. If one MAJ fails, however, the result is correctable by the next stage of the system.

The repetition code can correct for more than a single error if we add more bits. For example, with 5 bits, the code can correct up to 2 errors, because the 3 correct bits form a majority. With 7 bits, we can correct 3 errors. In general, an n -bit repetition code can correct $\frac{n-1}{2}$ errors. On the other hand, the n -bit code requires us to transmit n times more data. This is not very efficient, because we only get one logical bit of information for every n physical bits transmitted. The *capacity* of the communication channel is reduced, even though the correctness of the data being received is improved.

A general notation to describe an ECC is $[n, k, d]$, where n is the number of physical bits, k is the number of logical bits, and d is the minimum *distance* between any two legal (no-error) encodings. For two binary strings, the distance is the number of bits that are different. A code with distance d can correct up to $\frac{d-1}{2}$ errors. Using this notation, the repetition codes are $[n, 1, n]$ codes, because the distance between n zeroes and n ones is n .

10.1.3 Hamming Code

To improve efficiency, we must be more sophisticated in how to generate the encoded bits. As an example, we consider a $[7, 4, 3]$ Hamming code, developed by Richard Hamming at Bell Labs in 1950 [85]. The encoding includes three parity bits, covering different combinations of data bits, which allow the encoder to distinguish among different single-bit errors. Figure 10.4 shows the generation of the parity bits c_i for a

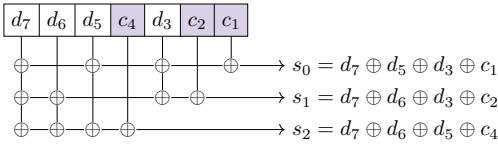


Figure 10.4 Hamming [7,4,3] code. Data bits are d_i and parity bits are c_i . Syndrome bits (s) indicate the position of a single bit flip error.

four-bit data value d . The parity bits are interleaved with the data bits, and the numbering starts at 1 instead of 0 for convenience; when the syndrome bits are calculated, they indicate the position of a single bit flip error.¹

Suppose we start with data value 1010. The parity bits are interleaved to create 101x0xx. The parity bits are calculated to create an even parity with the associated data bits. For example, $c_2 = 1$ because $d_7 \oplus d_6 \oplus d_3 = 1$, so the encoded value is 1010010. Now a bit flip error occurs on d_3 , leading to a received value of 1010110. The syndrome bits are calculated as 011, correctly indicating that bit d_3 is flipped.

Consistent with a distance of 3, this code is not able to detect or correct two-bit errors. For example, suppose we start with 0000000 and flip both d_6 and d_5 . The received value is 0110000. The syndrome bits are 011, which incorrectly indicates that d_3 was flipped.

The efficiency of the $[7, 4, 3]$ code is $4/7 = 0.571$, meaning that 57.1% of the transmitted bits contain data. This is significantly better than the 33.3% efficiency of the $[3, 1, 3]$ code above. We can increase the size of the code to improve efficiency even more. The number of parity bits needed for n data bits is $O(\log n)$, so the ratio of data to parity bits gets better and better. However, we still end up with a minimum distance of 3, meaning that only single-bit errors can be corrected. The more bits we add in the data block, the higher probability that multiple independent bits will be flipped. If this happens too often, more errors go undetected and the quality of the channel degrades.

We will end our discussion of classical ECCs at this point. There has been a lot of progress over the decades since Hamming’s discovery, but that is the topic for another textbook. We now turn our attention to quantum errors and how encoding can play a role in the development of fault-tolerant quantum computing.

10.2 Quantum Errors

Error correction becomes more complicated in the quantum realm because of three major factors.

- The quantum state is more complex than a binary 1 or 0. Information carried by superposition and relative phase is critical to the correct execution of quantum algorithms. We will see that errors in both amplitude and phase are important.
- Measuring the qubit to detect an error is destructive. For example, supposed a bit flip error changes a state from $\alpha |0\rangle + \beta |1\rangle$ to $\beta |0\rangle + \alpha |1\rangle$. If we measure while detecting the error, we end up in state $|0\rangle$ or $|1\rangle$, and we lose the opportunity to correct back to the desired state.

¹ It is traditional to use p_i to denote a parity bit, but we use c_i to avoid confusion with our error probability p .

- The no-cloning theorem prevents us from copying the state to measure it. No-cloning also means that we cannot simply replicate an existing state to encode it.

Figure 10.5 illustrates the more complex nature of quantum errors. Even if we consider only errors that manifest as rotations on the Bloch sphere—so-called *Pauli* errors—we see that the space of possible errors is continuous.

A *bit flip* error is analogous to a bit flip in the classical realm, where a $|0\rangle$ becomes a $|1\rangle$ and vice versa. Note that this is equivalent to an X gate being applied to the state. If we can detect this error, then it can easily be corrected with another X gate.

A *phase flip* is a new type of error, with no classical equivalent. The error changes the relative phase in the quantum state. When measuring in the computational basis, such an error is not visible, but phase differences can be critical in the proper execution of many quantum algorithms. When detected, this error can also be easily corrected by applying a Z gate.

However, an error can cause a fractional rotation along any or all of the axes of the Bloch sphere, as shown in the bottom line of Figure 10.5. These errors can occur due to imprecise control during gate operations or state preparation, or through interaction with other qubits or the environment. It would seem that correcting this sort of error would be impossible, because we'd have to know precisely the amount of rotation in each basis.

The good news, however, is that the technique for detecting (measuring) errors will project the state into another state which exhibits a bit flip and/or phase flip error, which can then be corrected. This is known as *quantization* of quantum errors, and leads to the following maxim:

Errors are continuous. *Measured errors* are discrete.

By detecting and correcting bit and phase flip errors, we also address the more subtle cases of continuous errors. We will see how this works in the following sections.

Of course, there are other types of non-Pauli errors, as described in Chapter 9. After our detailed discussion of Pauli errors, we will consider these other error types in the last section of this chapter.

In the rest of the chapter, logical qubit $|\psi\rangle_L$ is an encoding of state $|\psi\rangle$ using n physical qubits, which we will label $|\psi\rangle_{n-1}, |\psi\rangle_{n-2}, \dots, |\psi\rangle_1, |\psi\rangle_0$. Physical qubit $|\psi\rangle_0$ is the least significant qubit in the multi-qubit state, and will be drawn at the top of a circuit

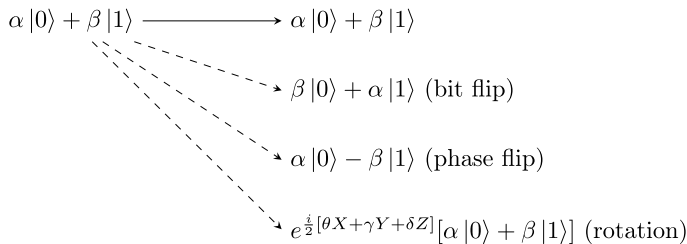


Figure 10.5 Quantum channel with Pauli errors.

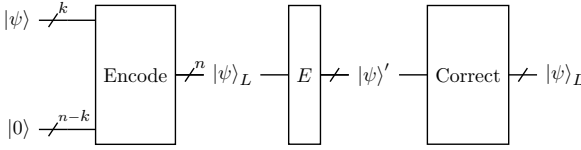


Figure 10.6 Quantum error correction: encoding, error, correction.

diagram. Likewise, $|\psi\rangle_{n-1}$ is the most significant physical qubit, and will be drawn at the bottom.

We represent a noisy quantum channel as an error operator E acting on the logical qubit. We use subscripts to denote single-qubit operators on the physical qubits. For example, X_i is the X operator applied to qubit i .

Figure 10.6 illustrates the error correction process: (1) Data qubits and additional bits are used to encode the desired state. (2) An encoded state $|\psi\rangle_L$ passes through a noisy channel, represented by error operator E , producing a new state $|\psi\rangle'$. (3) The error correction circuit (QEC) then detects and corrects any errors, resulting in the original state $|\psi\rangle_L$. As shown, the QEC circuit typically relies on ancilla qubits; as we'll see later, this avoids the need to directly measure the data qubits.

For now, we will assume that both the encoding circuit and the QEC circuit are error-free.

In general, a quantum error correction code is described as $[[n, k, d]]$, where n is the number of physical qubits, k is the number of logical qubits, and d is the minimum number of errors that will transform one logically-encoded state into another. As in the classical case, correcting t errors requires $d \geq 2t + 1$.

10.3 Detecting and Correcting Quantum Errors

10.3.1 Bit Flip

Let's create a QEC code for a single logical qubit that will detect and correct a single bit flip error. In our classical code, we used replication to create logical bits 000 and 111. We will take a similar approach here and use $|000\rangle = |0\rangle_L$ to represent a logical $|0\rangle$ and $|111\rangle = |1\rangle_L$ to represent a logical $|1\rangle$.

We can't replicate a single qubit to create the logical qubit, but we can create an entangling circuit, shown in left-hand box of Figure 10.7. An initial state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ becomes an entangled superposition of our logical qubits:

$$|\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle = \alpha|0\rangle_L + \beta|1\rangle_L. \quad (10.3)$$

Mathematically, we are embedding the two-dimensional Hilbert space ($\mathcal{H}_2$) representing the single-qubit state into a larger Hilbert space, in this case a 2^3 -dimensional space ($\mathcal{H}_8$) created with three physical qubits. Correct values of the logical qubit fall into a subspace, represented by

$$\mathcal{C} = \text{span}\{|000\rangle, |111\rangle\} \subset \mathcal{H}_8. \quad (10.4)$$

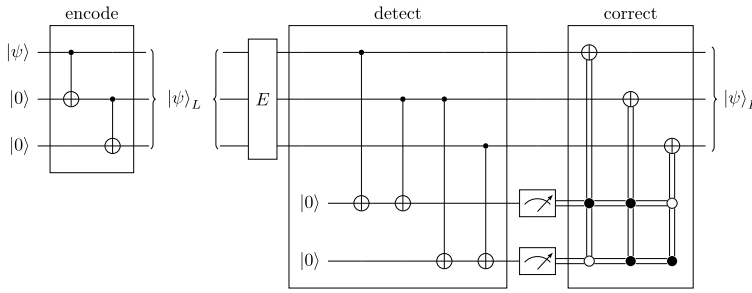


Figure 10.7 Three-qubit error correction code for bit flips only. The double lines are classical signals after measurement. Closed circles are controls, and open circles are “control-if-zero.”

In other words, a legal logical state is any linear combination of $|0\rangle_L$ and $|1\rangle_L$. The subspace $\mathcal{F} = \mathcal{H}_8 - \mathcal{C}$ is the subspace formed by error states. If we can determine whether a state $|\psi\rangle_L$ is in $\mathcal{C}$ or $\mathcal{F}$, we can detect an error.

In this section, our error operator E can only be a bit flip on a single physical qubit (or no flip at all). In other words:

$$E \in \{I, X_0, X_1, X_2\}. \quad (10.5)$$

Table 10.1 shows the effect of each error on an encoded qubit.

To detect a bit flip, we check whether any bit is different than the others. In addition, we need to know which bit is different, so that we can correct the error by applying the X operator to only that bit. In order to avoid direct measurement of the data qubits, we use ancilla qubits. The detection circuit is shown in the middle of Figure 10.7. The first ancilla qubit q_0 will become $|1\rangle$ if $|\psi\rangle_0$ and $|\psi\rangle_1$ are different. Otherwise, it will remain $|0\rangle$. Likewise, q_1 will become $|1\rangle$ if $|\psi\rangle_1$ and $|\psi\rangle_2$ are different. Measuring these ancilla qubits, therefore, gives us two classical bits that determine which data qubit, if any, has been flipped. In error correction terminology, these are known as *syndrome bits*.

Table 10.1 Effects of bit-flip error channel on a 3-qubit replication code.

$ \psi\rangle_L = 0\rangle_L$	E	$ \psi\rangle'$	$ \psi\rangle_L = 1\rangle_L$	E	$ \psi\rangle'$
$ 000\rangle$	I	$ 000\rangle$	$ 111\rangle$	I	$ 111\rangle$
$ 000\rangle$	X_0	$ 001\rangle$	$ 111\rangle$	X_0	$ 110\rangle$
$ 000\rangle$	X_1	$ 010\rangle$	$ 111\rangle$	X_1	$ 101\rangle$
$ 000\rangle$	X_2	$ 100\rangle$	$ 111\rangle$	X_2	$ 011\rangle$

$ \psi\rangle_L$	E	$ \psi\rangle'$
$\alpha 000\rangle + \beta 111\rangle$	I	$\alpha 000\rangle + \beta 111\rangle$
$\alpha 000\rangle + \beta 111\rangle$	X_0	$\alpha 001\rangle + \beta 110\rangle$
$\alpha 000\rangle + \beta 111\rangle$	X_1	$\alpha 010\rangle + \beta 101\rangle$
$\alpha 000\rangle + \beta 111\rangle$	X_2	$\alpha 100\rangle + \beta 011\rangle$

The right-hand side of the figure shows the correction, using the measured syndrome bits (s_1 and s_0) as controls. An X gate is applied to at most one of the data qubits, restoring the original logical state.

What happens if there are two bit flips? Just like in the classical case, the detection circuit will misclassify the error. For example, applying an error of X_1X_0 on an input state of $|000\rangle$ results in $|011\rangle$. This will be interpreted as an error in qubit 2, and the “restored” logical state will be $|111\rangle$. In other words, two physical bit flip errors will result in a logical bit flip error.

This simple encoding, however, is not a complete QEC code, because it fails to detect and correct phase errors. Consider the general logical state $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$. A phase flip (Z gate) on any single bit yields $\alpha|000\rangle - \beta|111\rangle$. The detection circuit will produce syndrome bits 00 and the phase error will be completely missed.

10.3.2 Phase Flip

In order to detect and correct for a phase flip, we change the basis set by applying a Hadamard gate after the encoding circuit, shown in Figure 10.8. This results in an encoding of $|0\rangle_L = |+++\rangle$ and $|1\rangle_L = |--\rangle$. We now consider a quantum channel that only exhibits single-qubit phase flips, namely:

$$E \in \{I, Z_0, Z_1, Z_2\}.$$
 (10.6)

A single-phase flip error changes one of the $|+\rangle$ states into $|-\rangle$ or vice versa, as shown in Table 10.2.

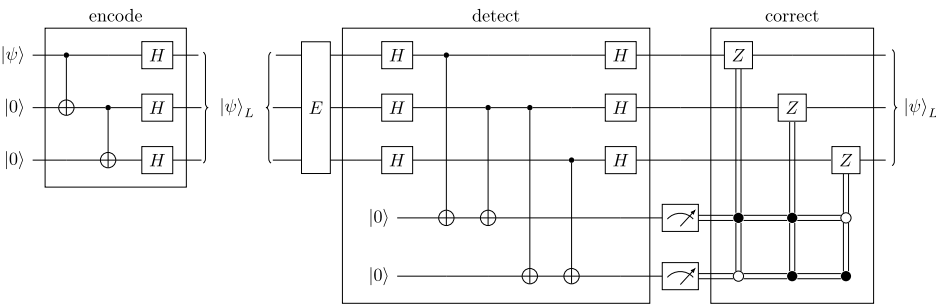


Figure 10.8 Three-qubit error correction code for phase flips only.

Table 10.2 Effects of phase-flip error channel on a 3-qubit phase replication code.

$ \psi\rangle_L$	E	$ \psi'\rangle$
$\alpha +++\rangle + \beta ---\rangle$	I	$\alpha +++\rangle + \beta ---\rangle$
$\alpha +++\rangle + \beta ---\rangle$	Z_0	$\alpha ++-\rangle + \beta --+\rangle$
$\alpha +++\rangle + \beta ---\rangle$	Z_1	$\alpha +-+\rangle + \beta +--\rangle$
$\alpha +++\rangle + \beta ---\rangle$	Z_2	$\alpha -++\rangle + \beta +-\rangle$

Using a similar technique as in Section 10.3.1, the error detection circuit uses ancilla qubits to determine whether the relative phases of two qubits are equal. In the circuit shown in Figure 10.8, Hadamard gates are used to translate the phase-encoded data qubits back into the computational basis. CNOT gates are then used to produce syndrome bits, followed by another set of Hadamards to restore the $+/ -$ basis. Controlled-Z gates are then used to apply a Z operator to the bit that was flipped.

As before, this is not a full QEC code, because it only detects and corrects phase flip errors. Suppose we have logical state $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$. Consider a bit flip error on a single physical qubit, such as X_0 . The resulting state is $\alpha|0\rangle_L - \beta|1\rangle_L$, and the syndrome bits will be calculated as 00 and no correction is made. The physical bit flip error becomes a logical phase flip error that cannot be corrected.

10.3.3 Correcting Bit and Phase Flips: Shor's 9-Qubit Code

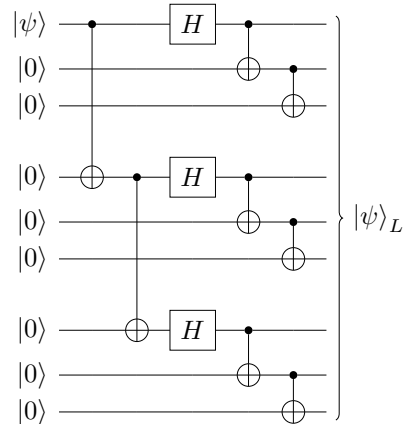
The first-known example of a complete QEC, one that corrects both bit flip and phase flip errors, was published by Peter Shor in 1995 [86]. It uses nine physical qubits to represent one logical qubit, and it can detect and correct any single bit or phase flip. It is a generalization of the three-qubit codes described above.

The encoding is shown in Figure 10.9. First the data qubit is replicated via entanglement, and then the Hadamard gate changes basis. This is reminiscent of the phase flip code described earlier. Now, each of those bits is triplicated by entanglement, allowing a bit flip within each group of three to be corrected. The encoding created by this circuit is:

$$\begin{aligned} |0\rangle_L &= \frac{1}{2\sqrt{2}}[(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)] \\ |1\rangle_L &= \frac{1}{2\sqrt{2}}[(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)]. \end{aligned} \quad (10.7)$$

An error correction circuit for the Shor code is shown in Figure 10.10. In the left-hand section, a bit flip error is detected for each group of three, using the technique in Section 10.3.1. The middle section detects a phase flip error. As in Section 10.3.2, we change to the x -axis basis to compare the phase of the physical qubits. This might not be obvious, but a single phase flip would cause an odd number of the CNOT gates on

Figure 10.9 Shor's $[[9,1,3]]$ code that corrects a single bit flip or phase flip.



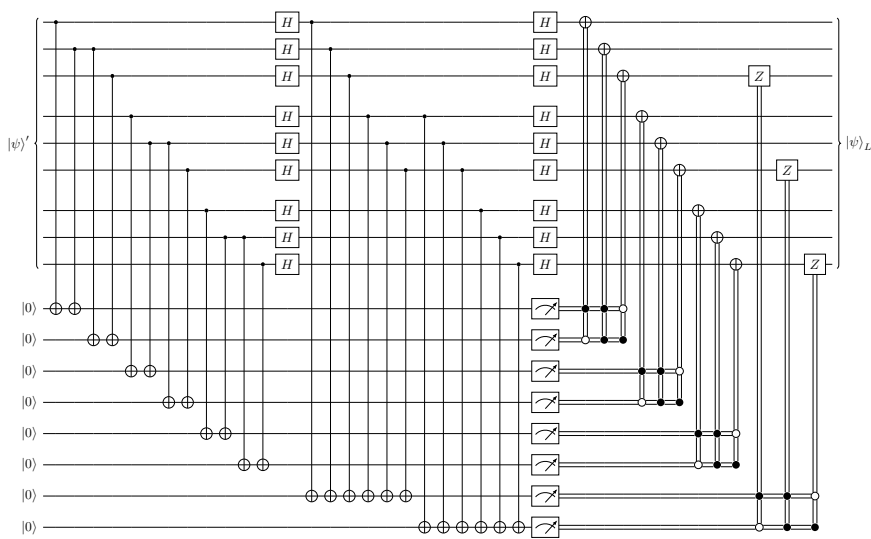


Figure 10.10 Correction circuit for Shor's $[[9,1,3]]$ code.

the ancilla qubit to flip. The first ancilla checks for a phase flip in either the first group or the second group; the second ancilla checks for a phase flip in the second or third groups.

Finally, the appropriate correction gate— X or Z , or both—will be applied to the appropriate qubit, based on the measured syndrome bits. The phase correction is a little different than before. A single phase flip in a group can be corrected by performing a Z gate on *any* of the three qubits.

This code is classified as $[[9, 1, 3]]$ because it requires nine physical qubits to encode a single logical qubit. The minimum number of errors that can result in a valid state is three. For example, a single phase flip in each group of three would transform $|0\rangle_L$ to $|1\rangle_L$ or vice versa. There are some multi-qubit errors that can be corrected by this code, such as a single bit flip in each of the three groups.

10.3.4 Arbitrary Rotations

So far, we have only considered bit flips and phase flips, each representing a rotation of π around one of the axes of the Bloch sphere. At the beginning of this section, however, we describe a quantum channel in which arbitrary rotations may occur. How does our QEC code perform on such errors?

For purposes of illustration, we again consider the bit flip code from Section 10.3.1. Consider applying an error operator $R_x(\pi/4) = X^{1/4}$ to a initial state of $|0\rangle$. The resulting state is $\cos^2 \frac{\pi}{8} |0\rangle + i \sin^2 \frac{\pi}{8} |1\rangle$. Measuring this state yields 0 with 85% probability and 1 with 15% probability,² as shown in Figure 10.11.

² The math: $\cos^2 \frac{\pi}{8} \approx (0.924)^2 \approx 0.85$; $\sin^2 \frac{\pi}{8} \approx (0.383)^2 \approx 0.15$.

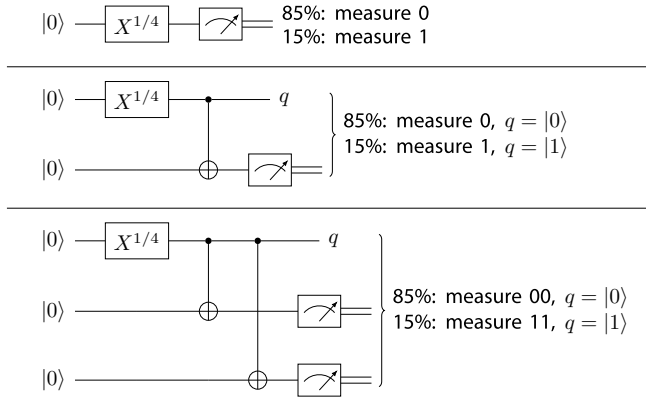


Figure 10.11 Partial rotation around the x -axis.

If this state is used as the control of a CNOT with a target of $|0\rangle$, the result is an entangled state: $\cos \frac{\pi}{8} |00\rangle + i \sin \frac{\pi}{8} |11\rangle$. If we measure the target qubit and get 0 (85% chance), then the control bit is projected to $|0\rangle$, because that is the only state compatible with a measurement of 0. Likewise, if we measure 1 (15% chance), the control qubit is projected to $|1\rangle$.

Finally, consider the error correction circuit from Figure 10.7. Assume that the $R_x(\delta)$ error occurs on physical qubit $|\psi\rangle_0$. The first ancilla qubit becomes entangled with $|\psi\rangle_0$ as described, and the syndrome bit could be measured as 0 or 1. Does this mean that sometimes the error is corrected and sometimes not? No. If the measurement is 0, then $|\psi\rangle_0 = |0\rangle$ and no correction is needed. If the measurement is 1, then $|\psi\rangle_0 = |1\rangle$; the correcting X gate corrects this to $|0\rangle$.

This is interesting—there’s an 85% chance that no error will be detected and no correction will be made. We “know” that a rotation occurred, because we explicitly performed it, but the measurement corrected the rotation without any further action. In other words, we cannot reliably *observe* whether the error occurred or not. If we don’t observe an error, then for all practical purposes, it never happened! This is captured by a saying attributed to Alexander Korotkov:

In quantum mechanics, you don’t see what you get, you get what you see.

When the rotation occurs on the second qubit, $|\psi\rangle_1$, there are two ancilla qubits involved. Is it possible for one of them to measure 0 and the other measure 1? No, because the two CNOT gates create a three-way entangled state, illustrated in Figure 10.11. If one measurement is 0, then the other measurement is certainly also 0. Therefore the error is consistently detected or not by the two ancilla qubits, and the data qubit is projected to either the valid state or the error state. The appropriate correction is performed.

While we have considered only an initial state of $|0\rangle$ and only a portion of the correction circuit, the same math applies to all the other qubits, and also to the phase correction portion. Therefore, the 9-qubit Shor code will correct any Pauli error on a single physical qubit.

10.4 Stabilizer Codes

Many QEC codes can be conveniently and compactly described through the use of *stabilizers*. This description allows codes to be analyzed and developed in a more formal and systematic way. In this section, we introduce the stabilizer approach and describe additional QEC codes.

10.4.1 Stabilizers

An operator M is said to *stabilize* a state $|\psi\rangle$ if it leaves the state unchanged.

$$M|\psi\rangle = |\psi\rangle = (+1)|\psi\rangle. \quad (10.8)$$

The state $|\psi\rangle$ is an eigenstate of the matrix representation of operator M , with an eigenvalue of $+1$.

In Section 10.3.1, we described a code as a partition of an extended Hilbert space, where valid states occupy one subspace $\mathcal{C}$, which we call the *codespace*, and all other states are members of an orthogonal subspace $\mathcal{F}$. We can use the stabilizer formalism to determine whether a state is in $\mathcal{C}$ or $\mathcal{F}$.

Consider a two-bit code that detects a single bit flip. (This is not a correction code, because we won't be able to tell which bit was flipped, only that a flip occurred.) We choose $|0\rangle_L = |00\rangle$ and $|1\rangle_L = |11\rangle$. It is clear that a single bit flip will produce a non-valid state. Another way to express this:

$$\mathcal{C} = \text{span}\{|00\rangle, |11\rangle\}, \quad \mathcal{F} = \text{span}\{|01\rangle, |10\rangle\}. \quad (10.9)$$

There is an operator $M = Z_1 Z_0$ that stabilizes all states in $\mathcal{C}$ and does not stabilize states in $\mathcal{F}$. Stated another way, $\mathcal{C}$ is in the $+1$ -eigenspace of M . Recall that the notation Z_i means that Z is applied to qubit i of the multi-qubit state, and qubit 0 is the least significant qubit.

$$\begin{aligned} M|00\rangle &= Z_1 Z_0 |00\rangle = |00\rangle \\ M|11\rangle &= Z_1 Z_0 |11\rangle = |11\rangle \\ M|01\rangle &= Z_1 Z_0 |01\rangle = -|01\rangle \\ M|10\rangle &= Z_1 Z_0 |10\rangle = -|10\rangle. \end{aligned} \quad (10.10)$$

Because Z is a Pauli operator, we know that it is Hermitian with eigenvalues ± 1 , and that means that $Z_1 Z_0$ is an observable. We can perform a *projective measurement* of the observable $Z_1 Z_0$ using a controlled- $Z_1 Z_0$ gate, as shown in Figure 10.12. An ancilla qubit is used to perform the measurement. Measuring in the computational basis will result in 0 for eigenvalue $+1$ and 1 for eigenvalue -1 . Note that we are not measuring

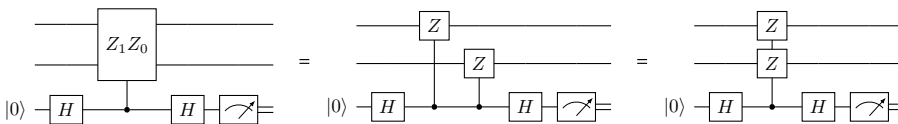


Figure 10.12 Measuring $Z_1 Z_0$ for the two-qubit bit flip detection code. The top two qubits are the state to be measured, and the bottom qubit is an ancilla. The three circuits are equivalent, just drawn in different ways.

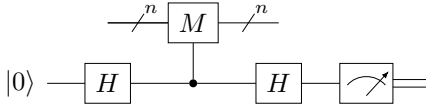


Figure 10.13 Projective measurement for an n -bit observable M with eigenvalues ± 1 .

Z_0 and then Z_1 and then multiplying the results. We are measuring the state using the combined operator $Z_1 Z_0$.

In general, a multi-qubit observable M with eigenvalues ± 1 can be measured with the circuit shown in Figure 10.13. If a stabilizer M is chosen such that it anti-commutes³ with an error operator E , then measuring M after the error operator will yield an eigenvalue of -1 :

$$ME|\psi\rangle = -EM|\psi\rangle = -E|\psi\rangle. \quad (10.11)$$

Therefore, the projective measurement reliably distinguishes between the error state and the non-error state. If we choose M to be a tensor product of Pauli operators, then any error that is also a combination of Pauli operators will either commute or anti-commute with the stabilizer. If all error operators anti-commute, then we can perform error detection.

10.4.2 Stabilizers for Error Correction

In order to correct an error, we need for two different errors E and F to be distinguished from one another, so that we know which to correct. We do this by choosing multiple stabilizers— M_a, M_b , etc.—such that $\mathcal{C}$ is in the $+1$ -eigenspace of each M . In addition, we choose the M 's to commute with each other, so that $\mathcal{C}$ is stabilized by the product of any combination of M 's. To be a bit more mathematically precise, the stabilizer S for a code is an Abelian group of operators, and $\{M_a, M_b, \dots\}$ is a set of generators for that group. (Abelian means that the group elements all commute with one another.) The set of generators is not unique. For a more rigorous mathematical description, refer to Daniel Gottesman's thesis [87].

Each error operator E must anti-commute with at least one of the M 's. This means that the error can be detected by at least one of the stabilizers. By finding stabilizers that interact differently with errors, we can uniquely identify the specific error.

For example, consider a 3-qubit code with two stabilizer generators:

$$M_a = Z_1 Z_0, \quad M_b = Z_2 Z_1. \quad (10.12)$$

This code can correct any single bit-flip error. X_0 anti-commutes with M_a , so it will measure -1 for that stabilizer; it commutes with M_b , which means it will measure $+1$. X_2 anti-commutes with M_b , and X_1 anti-commutes with both M_a and M_b . By measuring both stabilizers, we can distinguish among these three errors. Also, because M_a and M_b commute, then $M_a M_b$ is also a stabilizer: $M_b M_a |\psi\rangle = M_b |\psi\rangle = |\psi\rangle$, leaving the original state undisturbed by the measurement.

In Figure 10.14, we show the measurement outcomes for each possible error in the single-bit-flip channel. We use a common notation for the stabilizer and error operators in which the Pauli operation is shown in each bit position: $Z_1 Z_0$ is IZZ . The figure also

³ An operator A commutes with operator B if $AB - BA = 0$. A anti-commutes with B if $AB + BA = 0$.

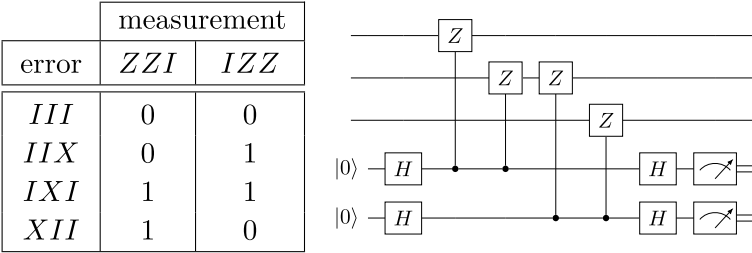


Figure 10.14 Stabilizer version of 3-qubit bit flip error detection.

shows the error detection circuit based on the stabilizer formalism. This is equivalent to the 3-qubit circuit shown in Figure 10.7.

The $[[9,1,3]]$ Shor code from Section 10.3.3 can also be described using the stabilizer formalism. There are eight stabilizers, shown below. The Z operators are used to detect bit flips, while the X operators are used to detect phase flips. (The Z error anti-commutes with the X stabilizer. Note that Y anti-commutes with both X and Z , so the same stabilizers detect a bit and phase flip on the same physical qubit.)

$$\begin{array}{l} I I I I I I Z Z \\ I I I I I I Z Z I \\ I I I I Z Z I I I \\ I I I Z Z I I I I \\ I Z Z I I I I I I \\ Z Z I I I I I I I \\ I I I X X X X X X \\ X X X X X X I I I \end{array} \tag{10.13}$$

In general, an n -qubit code that represents k logical qubits requires $n-k$ independent stabilizers. Each stabilizer reduces the size of the codespace by a factor of 2.

In 1996, Andrew Steane [88] proposed a quantum error correction code that is related to the classical $[7, 4, 3]$ code discussed in Section 10.1.3. It uses seven physical qubits to encode one logical qubit, and it corrects any single qubit bit or phase flip. This is an example of a more general class of codes known as Calderbank-Shor-Steane (CSS) codes [89]. CSS codes combine classical linear codes to create quantum codes that correct bit and phase flips. Though these codes were not originally described using the stabilizer formalism, they are a useful subclass of stabilizer codes.

Table 10.3 shows the stabilizer generators for the Steane code. These operators follow the same pattern as the Hamming code, though the order of qubits is different. The first group of X operators work to detect phase flip errors, and the group of Z operators is used to detect bit flips. Table 10.4 shows the syndrome bits that result from measuring the ancilla qubits in the error correction circuit in Figure 10.15. Three of the syndrome bits indicate the location of a bit flip, and the other three syndrome bits indicate the location of a single phase flip. This code can actually correct two errors, if both a bit and phase flip occur on different physical qubits.

Table 10.3 Stabilizer generators for Steane $[[7, 1, 3]]$ code.

Name	Operator						
M_0	I	I	I	X	X	X	X
M_1	I	X	X	I	I	X	X
M_2	X	I	X	I	X	I	X
M_3	I	I	I	Z	Z	Z	Z
M_4	I	Z	Z	I	I	Z	Z
M_5	Z	I	Z	I	Z	I	Z

Table 10.4 Syndrome measurements for all single-qubit errors in the Steane code. Each syndrome bit corresponds to the measurement of the corresponding operator from Table 10.3—e.g., bit 0 corresponds to measuring M_0 .

Error	Syndrome	Error	Syndrome	Error	Syndrome
Z_0	000111	X_0	111000	Y_0	111111
Z_1	000011	X_1	011000	Y_1	011011
Z_2	000101	X_2	101000	Y_2	101101
Z_3	000001	X_3	001000	Y_3	001001
Z_4	000110	X_4	110000	Y_4	110110
Z_5	000010	X_5	010000	Y_5	010010
Z_6	000100	X_6	100000	Y_6	100100

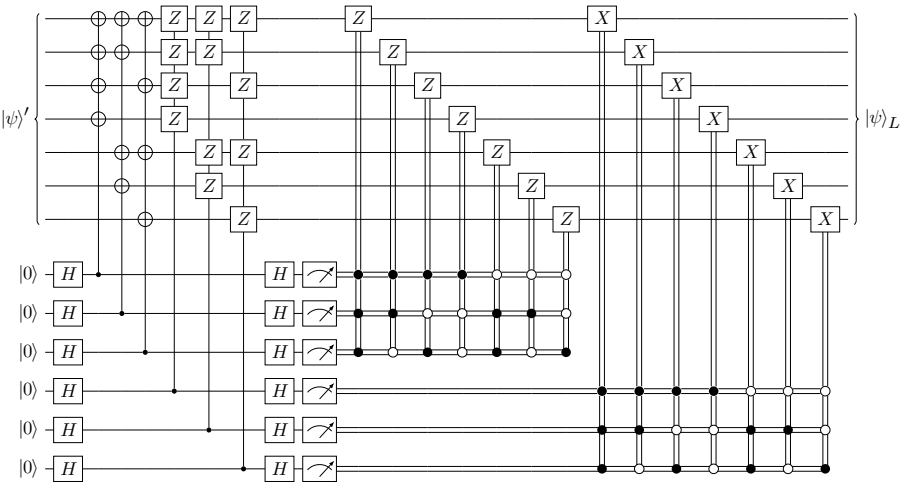


Figure 10.15 Error correction circuit for the $[[7, 1, 3]]$ Steane code.

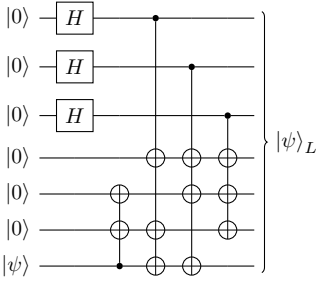


Figure 10.16 Encoding circuit for the $[[7, 1, 3]]$ Steane code.

The encoding of the logical $|0\rangle_L$ and $|1\rangle_L$ states for the Steane code is less obvious than for the Shor code.

$$\begin{aligned}
 |0\rangle_L &= \frac{1}{2\sqrt{2}} (|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\
 &\quad + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle) \\
 |1\rangle_L &= \frac{1}{2\sqrt{2}} (|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\
 &\quad + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle).
 \end{aligned} \tag{10.14}$$

Each component of the state is stabilized by each of the generator operators, and there are a couple of notable features. The components of $|1\rangle_L$ are derived by flipping all qubits in the $|0\rangle_L$ representation—in other words, $X^{\otimes 7} |0\rangle_L = |1\rangle_L$. Also, all components of $|0\rangle_L$ have an even number of bits, so that $Z^{\otimes 7} |0\rangle_L = |0\rangle_L$, while all components of $|1\rangle_L$ have an odd number of bits, resulting in $Z^{\otimes 7} |1\rangle_L = -|1\rangle_L$.

An encoding circuit for the Steane code is in Figure 10.16. While this version is adapted from Preskill [90], Cleve and Gottesman [91] describe a general method for encoding stabilizer codes.

10.5 Operating on Logical Qubits

In the previous sections, we have considered how to encode logical qubit states $|0\rangle_L$, $|1\rangle_L$, and general superposition states. Of course, we also need to perform quantum operations on these states. If quantum gates get significantly more complex, we will pay a cost in both circuit width and circuit depth for fault tolerance.

From the examples in the previous section, it should be pretty clear how we can perform a logical X , which we denote as $\bar{X}$. Each encoding of $|0\rangle_L$ has a corresponding encoding in $|1\rangle_L$ with all of the bits flipped. Therefore, just performing an X gate on each physical qubit, as shown in Figure 10.17 has the result of flipping the logical state.

Likewise, all encodings of $|0\rangle_L$ have an even number of ones, so performing a Z on all qubits has no net effect. On the other hand, all encodings of $|1\rangle_L$ have an odd number of ones, and performing a Z on all qubits will multiply the state by -1 . Therefore, $\bar{Z}$ is equivalent to performing Z on each physical qubit.

This property is known as *transversality*, and such gates are known as *transversal* gates. The CNOT gate is also transversal, achieved by choosing the control and target

Figure 10.17 The logical $\bar{X}$ operator is transversal for the $[[7,1,3]]$ code, consisting of a single X gate on each physical qubit. Logical operators for Z , H , S , and $CNOT$ are also transversal.

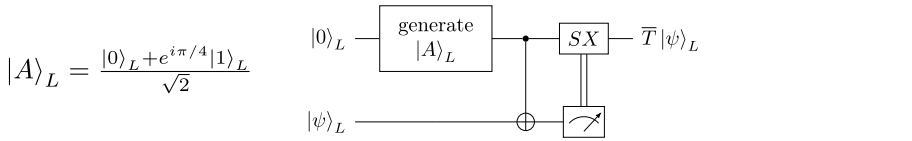


Figure 10.18 Performing the T operation on encoded data.

qubits from corresponding positions in the encoded states. In addition, H and S are also transversal; in the case of S , the $S^\dagger$ operator must be applied to each qubit. (The proof is left as an exercise for the reader.)

Transversality is a useful property. Not only is it trivially easy to implement, it also means that qubits do not need to interact with each other. Therefore, errors do not propagate from one qubit to another during the logical operation.

However, not all encoded gates can be implemented transversally. For example, $\bar{T}$ is not transversal for the Steane code. While the Clifford gates are all transversal, the T gate is required for universal quantum computation. In fact, there can be no code for which a transversally-encoded gate set is universal [92].

The logical $\bar{T}$ gate can be performed with the help of a so-called *magic state*. For the $\bar{T}$ gate, we can use a magic state $|A\rangle_L$ with the circuit shown in Figure 10.18 to perform the T operator on an encoded state $|\psi\rangle_L$. State $|A\rangle_L$ is equivalent to a T operation on encoded state $|+\rangle_L$. We can create this state by encoding a single-qubit state $TH|0\rangle$ using the circuit in Figure 10.16. This is not a fault-tolerant preparation, however. The process of creating a high-quality magic state is known as *distillation*, in which several copies of the magic state are prepared and refined into higher-fidelity magic states.

10.6 Error Thresholds

In this section, we describe what is known as the *threshold theorem*, which claims that arbitrarily accurate quantum computing can be performed, provided that the error per operation is below a threshold value [93]. It is this result that gives hope that large-scale quantum computation is possible, even with less-than-perfect qubits and gates. The threshold gives a target for a “reasonable” amount of noise in our physical system.

10.6.1 Concatenation of Error Codes

The QEC codes we have discussed all have a distance of 3, meaning that they can correct only a single physical error. One way to tolerate more physical errors is to *concatenate* codes into higher-level encodings. Concatenation means that we take multiple logical qubits and further encode them, using either the same or a different code.

For example, suppose we take 49 logical qubits and encode them in groups of seven, using the Steane $[[7, 1, 3]]$ code. This yields seven *level-1* logical qubits. It’s not strictly necessary that each of these level-1 qubits use the same code, but it’s easier to assume so.

Then we perform the $[[7, 1, 3]]$ encoding operation again on the seven level-1 logical qubits, using transversal logical operators, creating a *level-2* logical qubit. We now have $7^2 = 49$ physical qubits, one logical qubit, and a distance of $3^2 = 9$, meaning we can detect and correct up to four physical errors on the 49 qubits: $t = (d-1)/2 = (9-1)/2 = 4$.

We can continue this indefinitely, choosing the same or different encodings at each level. Using the Steane code, a g -level encoding requires 7^g physical qubits and with a distance of 3^g . Both the distance and the number of qubits scale exponentially with the number of levels; unfortunately the number of qubits scales faster. In addition, the encoding and error correction circuits require additional quantum gates, providing more opportunity for errors. Does concatenation result in a lower overall logical error rate? It depends on the physical error rate, as we discuss in the next section.

10.6.2 Threshold Theorem

The threshold theorem allows us to understand whether using more physical qubits will reduce the errors that are observed at the logical qubit level.

First, we assume that qubit errors are independent of each other, and that each qubit experiences an X and/or Z error with probability p for each gate operation. Second, we assume that quantum circuits are composed according to fault-tolerant principles, such that any single gate failure will result in at most one error at the input of a logical qubit block. Finally, we assume that a round of error correction is performed after each logical operation, using a code that can correct a single-qubit error. This means that the logical operation will fail only if two or more errors occur.

Therefore, the failure rate of each operation is

$$p_{L1} = cp^2, \quad (10.15)$$

where p_{L1} is the rate of level-1 logical error and c is an upper bound for the number of possible two-error combinations that can occur while performing the logical and error correction operations. The value of c is considered constant for a given correction code, and it depends on the implementation of the code and the details of the quantum computer.

If we now concatenate using the same code, the logical failure rate of a level-2 operation is

$$p_{L2} = c(p_{L1})^2 = c^3 p^4. \quad (10.16)$$

Extending this to a g -level concatenated code yields

$$p_{Lg} = \frac{(cp)^{2^g}}{c}. \quad (10.17)$$

This implies that for a given *physical* error rate of p , the failure rate of a level- g logical qubit can be made arbitrarily small, as long as $cp < 1$. This inequality defines a *threshold* physical error rate, such that application of additional error correction will

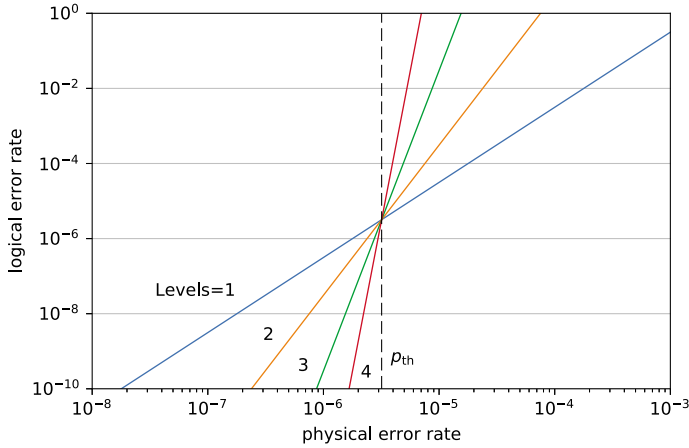


Figure 10.19 Estimated logical error rates for a concatenated Steane $[[7, 1, 3]]$ code for various physical error rates. For error rates below the threshold (p_{th}), additional levels of concatenation can reduce the logical error rate.

reduce the failure rate of logical gates.

$$p_{th} < 1/c. \quad (10.18)$$

As a concrete example, Knill *et al.* [93] estimated $c \leq 313,894$ for a system with arbitrary qubit interconnect, using the Steane $[[7, 1, 3]]$ code. This estimate was based on the preparation of the $|\pi/8\rangle$ state, and it leads to a threshold $p_{th} = 3 \times 10^{-6}$. The value of c would be higher for realistic systems with limited interconnect.

Figure 10.19 shows the calculated logical error rates of a concatenated Steane code for various physical error rates. If the physical error rate is above p_{th} then adding layers of error correction only makes things worse. However, if the physical error rate is below the threshold, then the logical error rate can be driven arbitrarily low by using more qubits.

Given current gate and readout errors on the order of 10^{-2} , it appears that the Steane code is not a feasible path to fault-tolerant quantum computing in the near term. However, a different family of error correction codes, known as *surface codes*, is known to have a higher threshold than the Steane code.

10.7 Surface Codes

The desire for scalable error correction led to an investigation of *topological codes*. These codes are constructed on a lattice (of dimension ≥ 2), and the protection of information relies on the unlikelihood of error chains forming non-trivial paths over the code surface.

The first such code was defined by Kitaev [94] for a 2D lattice on the surface of a torus, with data qubits representing the edges of the lattice and stabilizers located at the vertices and on the faces. It is therefore known as the *toric code*. Others have shown

that the toric topology is not a requirement, which led to the development of planar *surface codes* [95, 96].

The surface code exploits localized interactions between neighboring qubits, so it is well-suited for the types of planar connection topologies that are common in superconducting quantum systems. It is also insensitive to local errors and can be scaled to improve the logical error rate. The threshold for the surface code is estimated in the 10^{-2} to 10^{-3} range, orders of magnitude better than the stabilizer codes discussed earlier.

However, a large number of physical qubits are required to implement a single logical qubit. In addition, logical operations require operations that move logical qubits around on the planar surface. Just like the stabilizer codes, non-Clifford gates (such as T) require the preparation (distillation) of special states, which incurs significant overhead.

In spite of these overheads, the surface code is a promising candidate for achieving fault-tolerant quantum computing for technologies that favor nearest-neighbor interaction among physical qubits. Our description of the 2D planar surface code is largely based on the survey paper by Fowler *et al.* [97].

10.7.1 Stabilizers

The operation of the surface code is based on X and Z stabilizers. The basic construct is a square mesh of physical qubits, as shown in Figure 10.20. This square mesh represents a single logical qubit, though we will see that larger collections will be needed for realistic error correction. The distance of the code is determined by the size of the square, and the distance determines how many physical errors can be reliably detected and corrected. If errors are rare and not physically-correlated, a larger array can detect and correct more physical errors.

Physical qubits are designated as data qubits and stabilizer (measurement) qubits. Data qubits are shown as white circles in the figure. Each data qubit is surrounded by *measure-Z* (yellow/light gray) qubits and *measure-X* (teal/dark gray) qubits. Each measurement qubit performs a stabilizer measurement on four data qubits; the direction

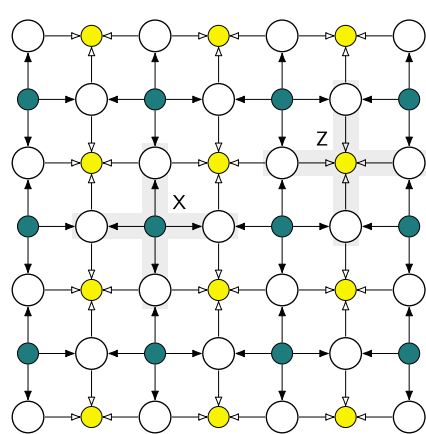


Figure 10.20 Surface code, composed by a square mesh of physical qubits. Data qubits (white) are interleaved with *measure-X* qubits (teal/dark gray) and *measure-Z* qubits (yellow/light gray). The direction of arrows show the direction of CNOT operations for projective measurements.

of the arrows indicate the direction of the CNOT operation performed during the projective measurement. Each data qubit is associated with two measure-X qubits and two measure-Z qubits, except on the boundaries of the square.

The left and right edges of the square are known as *X boundaries*, because the edge only contains data and measure-X qubits. This is also known as a “smooth” boundary in the surface code literature. The top and bottom edges are *Z boundaries*, also known as “rough” boundaries.

The distance of the code is determined by the minimum number of data qubits in a path that connects one boundary to the other boundary of the same type. In Figure 10.20, $d = 4$. It’s not a requirement that the mesh be square, but since distance is determined by the minimum path between left/right or top/bottom edges, there is no advantage to adding more qubits in only one dimension.

There are more data qubits than measurement qubits in the square. In this example, there are 25 data qubits and 24 measurement qubits. Therefore, there are 2×25 degrees of freedom in the measured outcomes of data qubits, with 2×24 constraints imposed by the stabilizers. The stabilizers are linearly independent, as no stabilizer can be written as a product of the other stabilizers. The two unconstrained degrees of freedom can be used to define a single logical qubit, as we will show later.

As a side note, the description above does not apply to arrays that have strictly X boundaries or strictly Z boundaries. In this case one of the stabilizers can be written as a product of the others, so there are $2 \times (N - 1)$ constraints. In such arrays, there are only $N - 1$ data qubits, so the array is completely constrained by the stabilizers, and there are no additional degrees of freedom. In order to create a logical qubit, the array must have both X and Z boundaries.

The error correction cycle is performed with XXXX and ZZZZ stabilizer measurements performed on all of the data qubits in parallel. Figure 10.21 shows a zig-zag pattern (a, b, c, d) of access to neighboring data qubits. Each measurement qubit is reset to the $|0\rangle$ state, and then CNOT gates are used to perform projective measurements on the data qubits. The idle (*I*) gates in the ZZZZ circuit are used to compensate for the Hadamard (*H*) gates in the XXXX circuit, so that the timing of each cycle is the same. Each physical CNOT gate is performed on all data qubits in parallel, before the next step is performed.

Suppose all of the data and measurement qubits are initialized to $|0\rangle$, and then an error correction cycle is performed. As a result of the measurements, the data qubits will be projected to a state $|\psi\rangle$ that is a simultaneous eigenstate of all of the *X* and *Z* stabilizers. This is known as a *quiescent state*; repeated measurement cycles will leave the state as $|\psi\rangle$ and will result in the same measurement outcomes for all of the measure-X and measure-Z qubits. Each measurement will result in a +1 or -1 outcome. For N measurement qubits, there are 2^N such quiescent states, and any of them can be used as the logical $|0\rangle_L$ state. We’ll discuss the manipulation of this logical state in a later section, but first let’s consider how this code allows a physical qubit error to be detected and corrected.

10.7.2 Error Detection and Correction

Figure 10.22 shows what happens when a bit-flip error occurs on a data qubit. The two *Z* stabilizers on either side will report a different measurement than before the error.

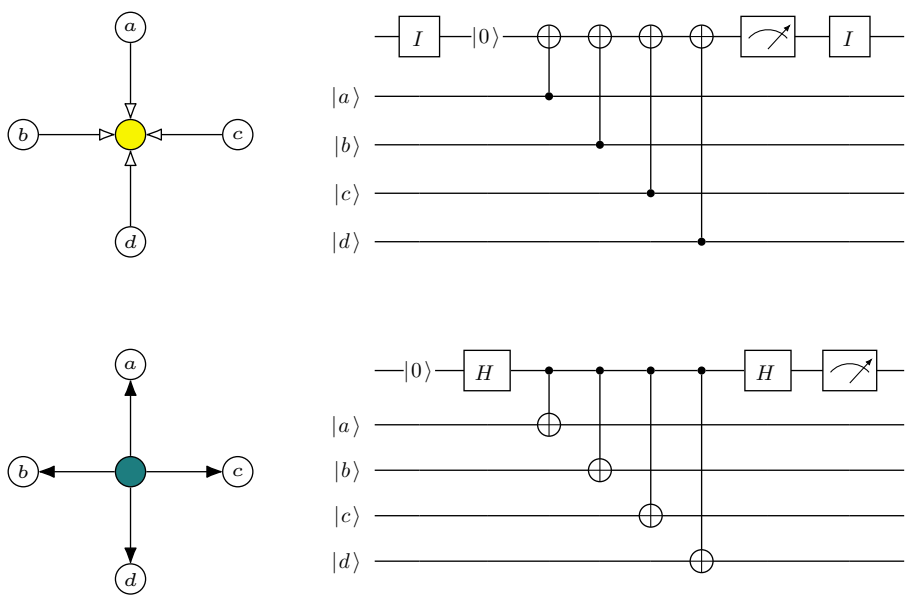
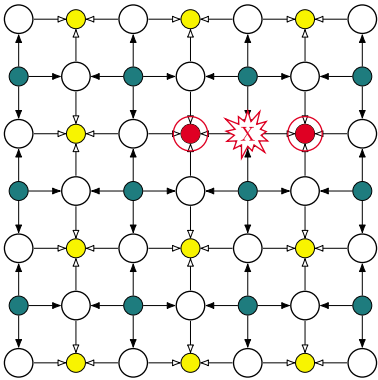


Figure 10.21 Surface code error cycle. The top row is a measure-Z circuit (ZZZZ stabilizer); the second row is a measure-X circuit (XXXX stabilizer). The sequence of gates allows for all measurements to happen in parallel across the entire surface.

Figure 10.22 When a bit flip (X) error occurs on a data qubit, the two measure-Z qubits will flip their measurement outcomes, indicated by circles and a change in color. This can be used as a *syndrome* to detect a bit flip error on a specific data qubit.



For instance, if the prior measurement was +1, the measurement in this cycle will be -1, because an odd number of flips occurred.

Recall the measure-Z circuit in Figure 10.21. To simplify the discussion, assume that all of the neighboring qubits, $a-d$, are in state $|0\rangle$.⁴ The measurement outcome will therefore be +1, the eigenvalue corresponding to measuring $|0000\rangle$.

Now assume that a bit flip error occurs on qubit b . In other words, this measure-Z qubit is the one to the right of the flipped data qubit in Figure 10.22. This change affects one of the CNOT gates in the circuit, causing the measurement outcome to change to -1.

⁴ They won't be $|0\rangle$, because each data qubit is projected into an eigenstate of all of its X and Z stabilizers, but this makes for an easier explanation.

For the measure-Z qubit to the left of the error, the same thing happens, only this time because its data qubit c has flipped. Note that the X stabilizer measurements above and below the data qubit are unaffected, because they detect only phase flips, not bit flips.

Therefore, the change of two neighboring measure-Z outcomes can be used as a *syndrome* to indicate a bit flip error on the data qubit shared by those two measure-Z qubits. Similarly, a phase flip error causes outcomes to change for the two adjacent measure- X qubits, as shown in Figure 10.23. If both a bit flip and a phase flip error occurs (a Y error), all four of the neighboring measurement qubits will have flipped outcomes.

In the case of a partial bit or phase flip, the stabilizer measurement performs as discussed in Section 10.4.1. The state will be projected to an eigenstate of the stabilizers. This will either correct the error, or it will project to a “full” bit/phase flip that can be corrected.

Figure 10.24 shows the effect on measurements when two, three, and four bit flip errors occur in the same area of the array. With two errors, a different (diagonal) pattern of Z stabilizers changes. However, this is not a unique syndrome—the red errors (bottom plus left) and the gray errors (top and right) result in the same outcome. The code cannot reliably detect the difference between these two scenarios.

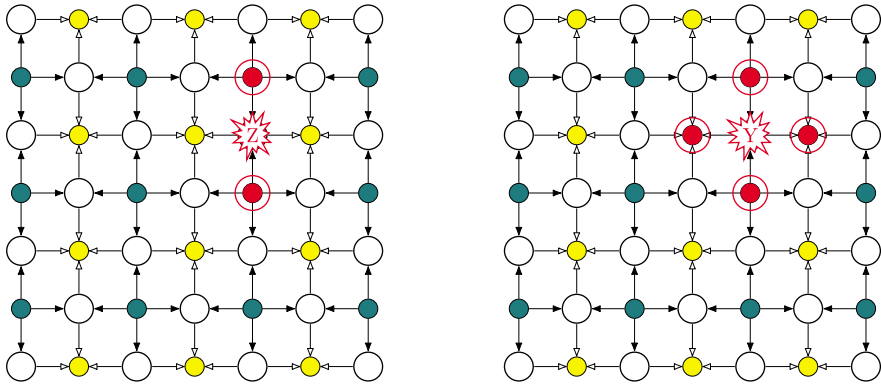


Figure 10.23 A phase flip error affects the outcome of its two adjacent X stabilizers. If both bit and phase flips occur, all four neighboring stabilizers will flip their outcomes.

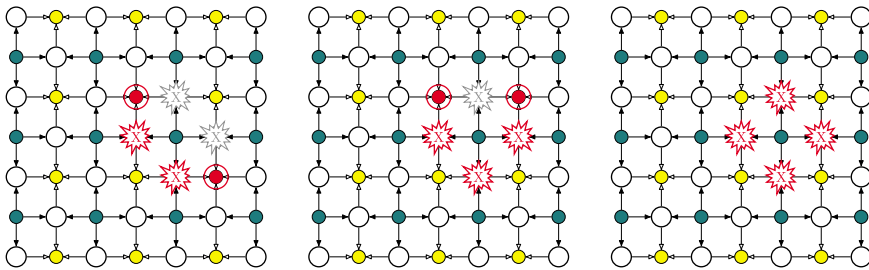


Figure 10.24 When multiple errors occur in the same vicinity, they may not be uniquely distinguishable. In general, many different chains of bit flip errors can result in the same syndrome measurement. In the case of four errors (right), no change in the measurement outcome occurs. This actually not an error at all, because the resulting state is trivially related to the original quiescent state $|\psi\rangle$.

The case of three errors, in the middle, has the same syndrome as the one-error scenario in Figure 10.22. In fact, many different chains of X errors can result in this same measurement outcome. However, if errors are rare and not physically correlated, the probability of a single bit flip error is much higher. Therefore, the error detection software will generally interpret this as a single-qubit error, even though that would be an incorrect interpretation in this case.

The rightmost part of Figure 10.24 shows a ring of four X errors. This is interesting, because none of the stabilizer measurements is affected. Each measure- Z stabilizer sees two bit flips, which maintains the sign of the previous measurement. This seems like an undetected error. In fact, it's effectively not an error at all: it's another version of the quiescent state $|\psi\rangle$. The key observation is that the four X operators are equivalent to the measure- X stabilizer in the middle of the four data qubits. Therefore, $|\psi\rangle$ already includes the superposition of both the flipped and unflipped versions of these data qubits, and the overall state does not change.

Going back to the middle picture, consider what happens if these three errors occur, but the error correction software decides to correct by applying an X operator to the gray data qubit instead. This correction creates a loop of X operators, just like the rightmost picture, which produces the original quiescent state. In fact, any path that connects the two flipped measure- Z qubits will either undo the actual errors, or will create a loop that will neutralize the errors. (The chosen path must include only data qubits and measure- Z qubits.)

At each surface code cycle, the measurement results are compared to the previous cycle's results. Classical control software must track and interpret changes in the measurements. Since the results can be ambiguous, the software must *decode* any changes to determine the most likely error(s) that caused the result. A variety of decoding algorithms have been proposed, such as minimum weight perfect matching [98, 99] or approaches based on neural networks [100].

Of course, errors can also occur on the measurement qubits. A measurement error can be distinguished from a data qubit error in two ways. First, only one measurement outcome changes. Second, a measurement error is not likely to happen on the next measurement cycle, so it will tend to be ephemeral. A data qubit error, on the other hand, will create a new quiescent state which will be stable over multiple measurement cycles. Typically, each error detection cycle involves d measurements, to reduce the probability and impact of measurement errors.

When a physical error is detected, we could apply an X or Z operation to correct it. However, such operations are also error-prone, so we are not guaranteed that the error will be corrected. Instead of manipulating the data qubit, the software can track the error and compensate when logical measurements are performed. For example, if a bit flip error occurs on data qubit a , if that qubit is later measured as 0, the software can correct the measured value to 1. This approach of software-corrected errors is more reliable than using faulty hardware operations to restore data qubits to their original states.

10.7.3 Logical X and Z Operators

In Section 10.7.1, we noted that the data qubits in the array outnumber the measurement qubits, which creates two unconstrained degrees of freedom. We can manipulate those degrees of freedom to realize a two-state system equivalent to a logical qubit.

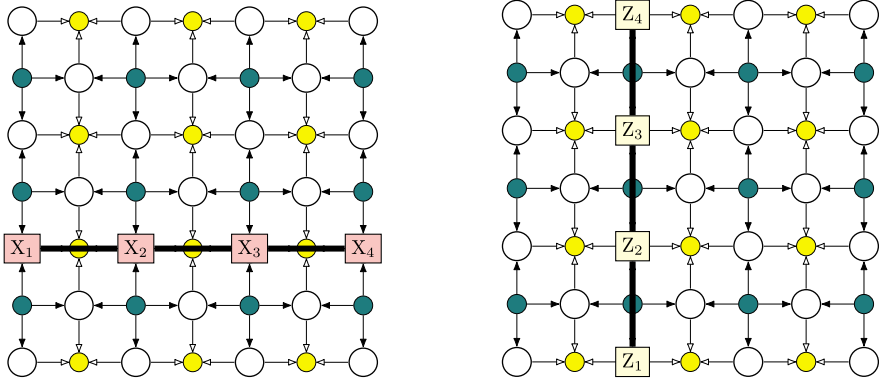


Figure 10.25 Logical X and Y operations on the surface code array.

Consider what happens when we apply an X operator to each of the data qubits in a single row, as shown in the left-hand part of Figure 10.25. Since each measure- Z qubit in that row sees two bit flips, their measurement outcomes do not change. We have applied an operator $X_1X_2X_3X_4$ to create a new state $|\psi_X\rangle$ that has exactly the same measurement outcomes as $|\psi\rangle$.

However, unlike the loop of X operations in Figure 10.24, this set of X operations is not equivalent to a measure- X stabilizer. Also, it cannot be written as a product of measure- X stabilizers. Therefore, $|\psi_X\rangle$ is not equivalent to $|\psi\rangle$, but it is a new quiescent state with the same measurement outcomes. We have manipulated one of the two degrees of freedom available in the array, and we can say:

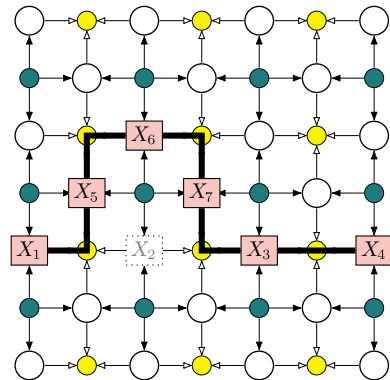
$$|\psi_X\rangle = \bar{X} |\psi\rangle \quad (10.19)$$

where $\bar{X}$ is the notation we used earlier to denote a logical X operator.

The $\bar{X}$ operator uses data qubits from one X -boundary of the array to the other. Does the sequence have to be along a single row? In other words, are there other logical X operators that would produce a state different than $|\psi_X\rangle$?

Consider the operation shown in Figure 10.26, where we choose an alternate path between the two X -boundaries. We'll call this operator $\bar{X}' = X_1X_5X_6X_7X_3X_4$. Using

Figure 10.26 Any path of X operators connecting one X boundary to the other is equivalent to the logical $\bar{X}$ operator, up to a global phase.



the relationship $X^2 = I$, we can rewrite this operator as in terms of $\bar{X}$ as follows:

$$\begin{aligned}\bar{X}' &= X_1 X_5 X_6 X_7 X_3 X_4 \\ &= X_5 X_6 X_7 X_2 (X_1 X_2 X_3 X_4) \\ &= X_5 X_6 X_7 X_2 (\bar{X}).\end{aligned}\tag{10.20}$$

The operator $X_5 X_6 X_7 X_2$ forms a loop that is equivalent to the X stabilizer in the middle of that loop. Therefore we have multiplied $\bar{X}$ by an operator that is stabilized to a ± 1 eigenvalue by the surface code, which means that:

$$\bar{X}' |\psi\rangle = \pm \bar{X} |\psi\rangle = \pm |\psi_X\rangle\tag{10.21}$$

where the $+$ or $-$ depends on the measurement outcome of the X stabilizer.

Similarly, any path that crosses the array from one X boundary to the other can be written as a product of $\bar{X}$ times a product of X stabilizers, and therefore will differ only by the sign of those stabilizer measurements. That means there is only one state $|\psi_X\rangle$ that is linearly independent of $|\psi\rangle$ and only one logical operator $\bar{X}$.

By the same logic, we can define a logical Z operator by performing Z operations on a chain of data qubits connecting one Z boundary to the other, as shown in the right-hand side of Figure 10.25. Beginning with quiescent state $|\psi\rangle$, this creates a state $|\psi_Z\rangle$ that has the same measurement outcomes but cannot be represented as a product of stabilizers.

$$|\psi_Z\rangle = \bar{Z} |\psi\rangle\tag{10.22}$$

Because any $\bar{X}$ operator and any $\bar{Z}$ operator will intersect at exactly one data qubit, and because the X and Z operators on that qubit anti-commute, we can show that the logical X and Z operators also anti-commute [97]:

$$\bar{X} \bar{Z} = -\bar{Z} \bar{X}.\tag{10.23}$$

Therefore the logical X and Z operators have the desired properties and the array can be viewed as a logical qubit.

Fowler *et al.* [97] describe the system in the following way. The quiescent state can be written as the product $|\psi\rangle = |Q\rangle |q_L\rangle$, where $|Q\rangle$ is a vector in the 2^N -dimensional Hilbert space defined by the N stabilizers. $|Q\rangle$ is constrained to a specific state in this Hilbert space, as determined by the measurement outcomes. The remaining degrees of freedom in $|\psi\rangle$ are captured by $|q_L\rangle$. The stabilizers have no effect on $|q_L\rangle$, and the logical operators have no effect on $|Q\rangle$.

In the array described above, there are two degrees of freedom that are unconstrained from the stabilizers, so $|q_L\rangle$ is defined by a two-dimensional Hilbert space. The eigenstates of $\bar{Z}$ are $|0\rangle_L$ and $|1\rangle_L$, such that $\bar{Z} |0\rangle_L = +|0\rangle_L$, and $\bar{Z} |1\rangle_L = -|1\rangle_L$. The eigenstates of $\bar{X}$ are $|+\rangle_L = (|0\rangle_L + |1\rangle_L)/\sqrt{2}$ and $|-\rangle_L = (|0\rangle_L - |1\rangle_L)/\sqrt{2}$. Either pair of logical states can be used as a basis for $|q_L\rangle$.

In order to use this as a logical qubit, we must be able to initialize it into a known state and to measure it. To initialize as $|0\rangle_L$, we initialize all data qubits as $|0\rangle$ and perform d rounds of error correction. Alternatively, we can initialize in the $|+\rangle_L$ state by initializing the physical data qubits to $|+\rangle$. To perform a logical measurement in either basis, we perform a physical measurement of all qubits in the appropriate basis (Z or X) and then calculate the parity of the physical measurement outcomes. The measurement

outcomes may be adjusted due to errors tracked by the control software for specific physical qubits. An even parity indicates $|0\rangle_L (|+\rangle_L)$ and odd parity indicates $|1\rangle_L (|-\rangle_L)$.

To do quantum computation, we also need other single-qubit gates, such as H , S , and T , as well as two-qubit gates, like CNOT. There are several different surface code approaches. In the next section, we will describe a technique known as lattice surgery.

10.7.4 Multiple Qubits: Lattice Surgery

Given the planar surface array described above, there are several alternatives for implementing multiple qubits. We could have a separate array for each qubit. This creates difficulty for multi-qubit operations: a CNOT, for example, can be performed transversally between corresponding physical data qubits, but this requires operations between qubits that are physically distant, eliminating the nearest-neighbor interactions that make the surface code attractive for technologies like superconducting qubits that prefer planar, local interconnections.

Another approach is to introduce *defects* or “holes” in the surface by turning off the stabilizer measurements on some qubits [97]. This creates an additional degree of freedom within the array that can be manipulated as a logical qubit. Typically, two defects are required for each logical qubit, and the distance between the holes is related to the desired distance d of the error correction code. This creates a significant overhead in the number of physical qubits needed for each logical qubit. In this model, two-qubit operations are performed by moving the defect of one logical qubit around the defect in another, an operation known as *braiding*.

A more efficient approach is to use multiple planar arrays, and to perform interactions between the arrays using a technique known as *lattice surgery*. As the name suggests, qubits can be stitched together (merged) and cut apart (split). The effects of the splitting and merging on the logical states of the qubit can be used to perform entangling operations using only nearest-neighbor interactions. In the rest of this section, we show the merge and split operations, and then show how CNOT can be performed.

10.7.4.1 Lattice Merge

The *merge* operation takes two planar logical qubits and merges them into one larger qubit. In this section, we describe how to perform a merge, and what the resulting state will be.

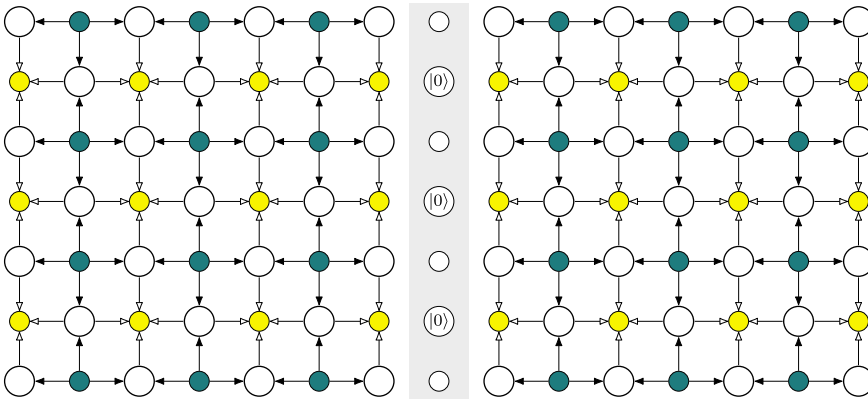
Figure 10.27(a) shows two qubits with distance $d = 4$. We’ll call the left qubit $|\lambda\rangle_L$ and the right qubit $|\rho\rangle_L$.

$$|\lambda\rangle_L = \alpha |0\rangle_L + \beta |1\rangle_L \quad (10.24)$$

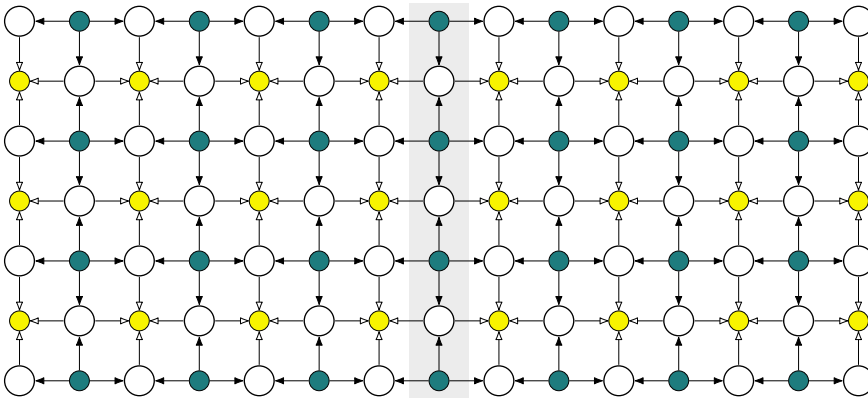
$$|\rho\rangle_L = \alpha' |0\rangle_L + \beta' |1\rangle_L. \quad (10.25)$$

Note that these qubits are rotated by 90 degrees, relative to Figure 10.20: the X (smooth) boundaries are at the top and bottom, and the Y (rough) boundaries are at the left and right.

Between the two qubits is a column of unused physical qubits. Following the same pattern as the logical qubits, we label each as either a data qubit or a measure-X qubit. These qubits are “unconnected,” in the sense that they are not participating in any of the stabilizer measurements for the logical qubits.



(a) Preparation for merging two planar surface qubits along a rough (Y) boundary. Data qubits in the middle are initialized to $|0\rangle$.



(b) Resulting logical qubit.

Figure 10.27 Rough merge of two planar qubits.

Because we are going to perform a merge along the rough boundaries, this action is known as a *rough merge*. To perform a rough merge, we initialize the physical data qubits in the middle column to $|0\rangle$. We then turn on the measure-X qubits and connect the measure-Z qubits along the boundaries to the data qubits, as shown in Figure 10.27(b), and perform d rounds of error correction. The result is a single logical qubit that still has distance $d = 4$, because the distance is determined by the length of the $\bar{X}$ operator, which hasn't changed. The interesting part, however, is how the state of this logical qubit relates to the states of the previous two logical qubits.

After error correction, we can look at the measurement outcomes for the new measure-X qubits that span the two original qubits, and we know the eigenvalue of the product of these X stabilizers. This is equivalent to measuring $\bar{X}\bar{X}$ on the two logical qubits. The initial state with two logical qubits is $|\lambda\rangle_L |\rho\rangle_L$. If we measure $\bar{X}\bar{X}$ without

merging, we end up in this logical state:

$$\frac{1}{\sqrt{2}} \left(|\lambda\rangle_L |\rho\rangle_L + (-1)^M \bar{X} |\lambda\rangle_L \bar{X} |\rho\rangle_L \right) \quad (10.26)$$

where M is the logical outcome of the measurement (0 or 1).

After the merge, however, we have one logical qubit, not two. We can write the resulting logical state as follows, using the symbol $\odot$ to represent the rough merge operation:

$$\begin{aligned} |\lambda\rangle_L \odot |\rho\rangle_L &= \alpha |\rho\rangle_L + \beta \bar{X} |\rho\rangle_L \\ &= \alpha' |\lambda\rangle_L + \beta' \bar{X} |\lambda\rangle_L. \end{aligned} \quad (10.27)$$

In classical terms, when the input qubits are either in the $|0\rangle_L$ or $|1\rangle_L$ state, the result of the merge is an XOR of the two states:

$ \lambda\rangle_L$	$ \rho\rangle_L$	$ \lambda\rangle_L \odot \rho\rangle_L$
$ 0\rangle_L$	$ 0\rangle_L$	$ 0\rangle_L$
$ 0\rangle_L$	$ 1\rangle_L$	$ 1\rangle_L$
$ 1\rangle_L$	$ 0\rangle_L$	$ 1\rangle_L$
$ 1\rangle_L$	$ 1\rangle_L$	$ 0\rangle_L$

A *smooth merge* operation can also be performed, which merges two planar qubits along their X (smooth) boundaries, adding new data and measure- Z qubits. In this operation the data qubits are initialized to $|+\rangle$ before the d rounds of error correction. The resulting logical state is shown below, where $\odot$ represents the smooth merge operation:

$$\begin{aligned} |\lambda\rangle_L \odot |\rho\rangle_L &= (a |+\rangle_L + b |-\rangle_L) \odot (a' |+\rangle_L + b' |-\rangle_L) \\ &= a |\rho\rangle_L + b \bar{Z} |\rho\rangle_L \\ &= a' |\lambda\rangle_L + b' \bar{Z} |\lambda\rangle_L. \end{aligned} \quad (10.28)$$

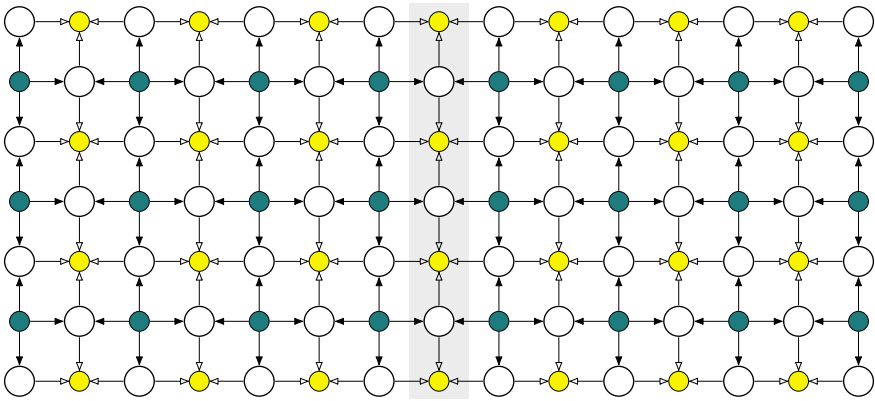
An interesting property of the merge operation is that it is *not unitary*. It starts with two logical qubits and ends with one. However, it is well-defined and fault-tolerant, and only requires nearest-neighbor operations.

10.7.4.2 Lattice Split

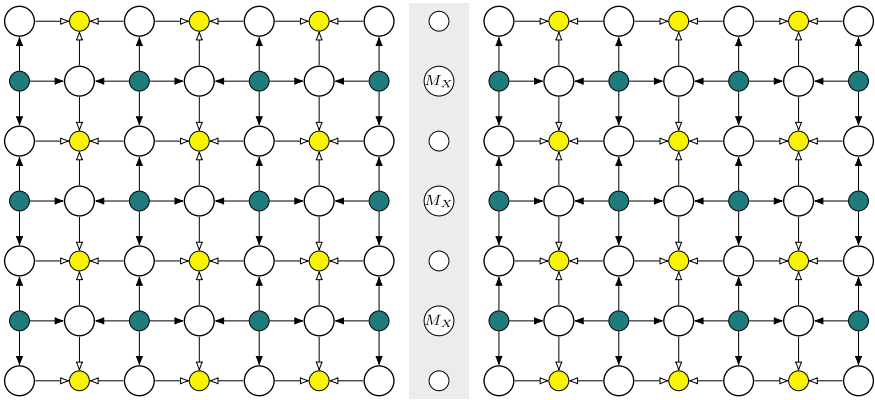
The opposite of a merge is a *split*, which takes one logical qubit and creates two. This is also a *non-unitary* operation, since it changes the number of logical qubits. Unlike merge, however, no information is lost; a split can be logically reversed by performing a merge.

The split operation can change the distance. Splitting a square logical qubit would result in two qubits with distance $d/2$. To preserve the distance d , we must start with a rectangular qubit, with distance d in one direction and $2d$ in the other. Note that this is exactly what we ended up with after the merge in the previous section.

As with the merge, a split can be either rough or smooth, depending on the type of boundary that is created by the split. Figure 10.28 illustrates a *smooth split*, creating two logical qubits with two new smooth (X) boundaries. For a smooth split, the separation is performed by measuring the physical data qubits in the X basis. The measure- Z qubits in the row/column are turned off, and the measure- X qubits attached to the removed data qubits are converted to three-qubit stabilizers.



(a) Preparation for splitting a qubit along a smooth (X) boundary. Data qubits in the middle will be measured in the X basis.



(b) Resulting logical qubits.

Figure 10.28 Smooth split of a planar qubits.

The removal of the measure-Z qubits has no effect on any of the other Z stabilizers, and all of the data qubits in the left and right sides of the split remain in the same Z-basis eigenstates. In other words, the logical $\bar{Z}$ operator on the two resulting logical qubits are unchanged and independent from each other.

The X -measurements of the removed data qubits will each have a result of $+1$ (logical 0) or -1 (logical 1). The measure- X qubits on either side must be multiplied by this result for the purpose of error correction. The parity of the remainder of the surface remains fixed, but now there are two surfaces, so the original logical $\bar{X}$ state is effectively entangled across the two resulting qubits. The result is that the smooth split generates the following transformation:

$$\alpha |0\rangle_L + \beta |1\rangle_L \longrightarrow \alpha |00\rangle_L + \beta |11\rangle_L. \quad (10.29)$$

A *rough split* operation is complementary, splitting along a rough boundary and measuring the data qubits in the Z basis. Again the result is an entanglement:

$$a|+\rangle_L + b|-\rangle_L \longrightarrow a|++\rangle_L + b|--\rangle_L. \quad (10.30)$$

10.7.5 CNOT

As we've seen, the CNOT is a two-qubit entangling gate that is necessary for quantum computation. Any logical qubit scheme must support such a gate, and the motivation of lattice surgery is to enable CNOT without requiring long-distance transversal interactions between physical qubits. This section shows how to perform CNOT using lattice split and merge operations [101].

Figure 10.29 shows a control qubit $|C\rangle$, a target qubit $|T\rangle$, and a helper intermediate qubit $|INT\rangle$. $|C\rangle$ and $|T\rangle$ have arbitrary quantum states, and $|INT\rangle$ is initialized to $|+\rangle_L$. There are also physical qubits set up to perform merge operators, as needed.

$$\begin{aligned} |C\rangle &= \alpha|0\rangle_L + \beta|1\rangle_L \\ &= \frac{1}{\sqrt{2}}(a+b)|0\rangle_L + \frac{1}{\sqrt{2}}(a-b)|1\rangle_L \\ &= a|+\rangle_L + b|-\rangle_L \\ |T\rangle &= \alpha'|0\rangle_L + \beta'|1\rangle_L \\ |INT\rangle &= |+\rangle_L. \end{aligned} \quad (10.31)$$

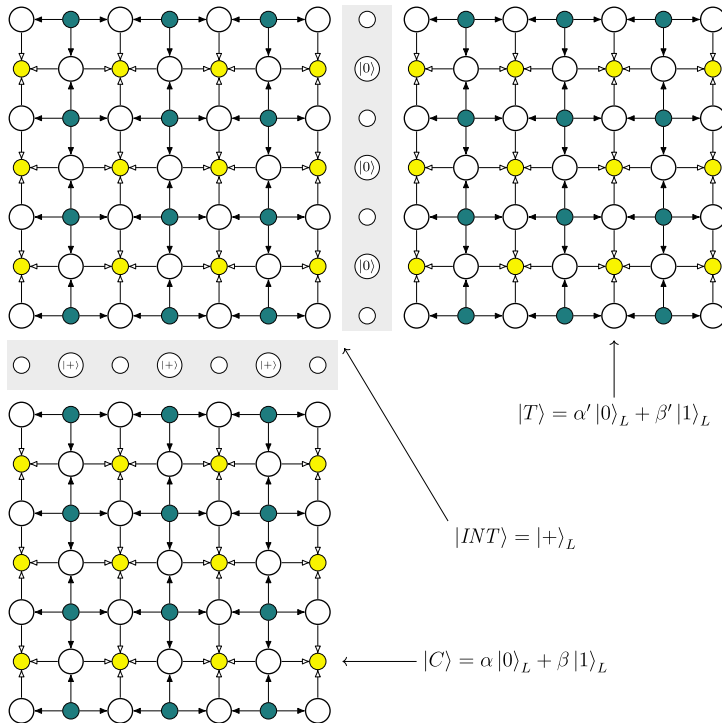


Figure 10.29 Preparation for a CNOT operation on control qubit $|C\rangle$ and target qubit $|T\rangle$, with the help of an intermediate qubit $|INT\rangle$.

First, a smooth merge is performed between $|C\rangle$ and $|INT\rangle$.

$$\begin{aligned} |C\rangle \odot |INT\rangle &= a|+\rangle_L + (-1)^M b|-\rangle_L \\ &= \frac{1}{\sqrt{2}}(a + (-1)^M b)|0\rangle_L + \frac{1}{\sqrt{2}}(a - (-1)^M b)|1\rangle_L \end{aligned} \quad (10.32)$$

where M is the logical measurement of $\bar{Z}\bar{Z}$ performed by the merge. Consider the output for each value of M :

$$\begin{aligned} |C\rangle \odot |INT\rangle &= \frac{1}{\sqrt{2}}(a + b)|0\rangle_L + \frac{1}{\sqrt{2}}(a - b)|1\rangle_L, M = 0 \\ &= \frac{1}{\sqrt{2}}(a - b)|0\rangle_L + \frac{1}{\sqrt{2}}(a + b)|1\rangle_L, M = 1. \end{aligned} \quad (10.33)$$

If M is 1, we do a bit flip ($\bar{X}$), either in software or by flipping the appropriate data qubits, to fix the result as:

$$|C\rangle \odot |INT\rangle = \alpha|0\rangle_L + \beta|1\rangle_L. \quad (10.34)$$

In other words, by doing a smooth merge with $|+\rangle_L$, plus the correction, we have effectively changed the shape of the qubit without changing its state.

Second, we perform a smooth split to separate the qubits back into their original shapes. As described above, this results in an entangled pair of logical qubits:

$$|C' INT'\rangle = \alpha|00\rangle_L + \beta|11\rangle_L. \quad (10.35)$$

Third, we perform a rough merge of $|INT'\rangle$ with $|T\rangle$:

$$\begin{aligned} |C'(INT' \odot T)\rangle &= \alpha|0\rangle_L \otimes (|0\rangle_L \odot |T\rangle) + \beta|1\rangle_L \otimes (|1\rangle_L \odot |T\rangle) \\ &= \alpha|0\rangle_L \otimes |T\rangle + (-1)^{M'}\beta|1\rangle_L \otimes \bar{X}|T\rangle \end{aligned} \quad (10.36)$$

where M' is the logical measurement of $\bar{X}\bar{X}$ performed by the merge. Correcting for $M' = 1$, either in software or with a $\bar{Z}$, we now have two logical qubits with state equivalent to $|C\rangle \oplus |T\rangle$.

Finally, note that the second logical qubit is a different shape than the original $|T\rangle$. It's not necessary, but we may want to restore the target qubit to its original region. If we split this qubit, we end up with entanglement with the helper qubit, which is not what we want. Instead, we can “shrink” the qubit to the original shape by measuring all of the data qubits in the unwanted region in the Z basis.

10.7.6 Single-Qubit Gates

To support universal quantum computing, the surface code must be able to perform the Clifford+T set of gates on logical qubits. We've discussed the Pauli gates (X , Z , Y) and CNOT. In this section we will briefly describe how to perform H , S , and T gates, which complete the required set.

The H and S gates can be performed classically, in software, by simply relabeling the logical operators on the qubit [102]. For example, since $HX = Z$, we “perform” an H gate on the qubit in Figure 10.25 by redefining $\bar{X}$ to be what is normally the $\bar{Z}$ operator:

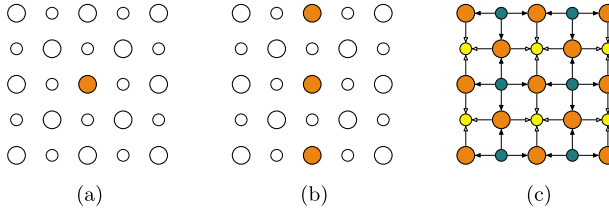


Figure 10.30 State injection. (a) Prepare a single data qubit with the desired state $|\psi\rangle$, and other qubits as $|0\rangle$. (b) Entangle data qubits to create multi-qubit state. (c) Stabilize to create logical state $|\psi\rangle_L$.

a sequence of Z operators between two Z boundaries. The $\bar{Z}$ operator would similarly be redefined as the old $\bar{X}$. The S gate maps:

$$\begin{aligned}\bar{X} &\rightarrow \bar{Y} \\ \bar{Y} &\rightarrow -\bar{X} \\ \bar{Z} &\rightarrow \bar{Z}.\end{aligned}\tag{10.37}$$

Alternatively, a Hadamard can be performed transversally on all physical data qubits, followed by changing the roles of the stabilizer qubits: measure- X qubits become measure- Z , and vice versa. This changes the orientation of the planar qubit, but there are various transformations to rotate the block if needed [101, 103].

The T gate, however, requires the use of a magic state, as described in Section 10.5. A procedure for state injection, as described by Horsman *et al.* [101], is described below and illustrated in Figure 10.30.

First, a distance-3 surface qubit is prepared, with all physical qubits, except one, prepared in the $|0\rangle$ state. The middle qubit (orange/gray) is prepared in the desired state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. For example, $|\psi\rangle$ could be the magic state required for a T gate: $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle)$.

Next, CNOT gates are used to entangle $|\psi\rangle$ with the measurement qubits above and below, and then swaps are performed to move the entangled states into the data qubits, as shown in the figure. This creates the state $\alpha|000\rangle + \beta|111\rangle$. The stabilizer qubits are then turned on to produce a distance-3 logical qubit in the desired state. To enlarge the state, we prepare additional physical qubits in state $|0\rangle$ and merge. This step can be repeated as many times as necessary to create a qubit with the desired distance.

Since the operations used to prepare the state for injection are not fault-tolerant, then the state injection process is not error-free. A process called *magic state distillation* is used to create high-fidelity magic states using fault-tolerant operations. Distillation is beyond the scope of this chapter, but there are a number of good sources for further information [104–106]. The cost of producing and consuming high-quality magic states is expected to be a large fraction of the time and space required for long-running quantum programs.

10.8 Summary and Further Reading

A chapter on quantum error correction will naturally be an incomplete treatment of such a broad and fast-moving field. We have tried to convey the key concepts and

to illustrate some of the practical steps needed to build large, fault-tolerant quantum computing systems.

Quantum errors are continuous, but measured errors are discrete. Though quantum states are quite complex (pardon the pun), the errors we can observe are constrained along the dimensions in which they are measured. This chapter discusses a simplified error model, one in which only rotations on the Bloch sphere are applied. Other types of errors include environmental decoherence, leakage, and so forth. The stabilizer formalism serves to *digitize* these errors into correctable bit and phase flips. Devitt, Monro, and Nemoto [107] provide a good introduction to how QEC addresses a broader range of errors.

Many physical qubits are required to implement a high-fidelity logical qubit. As in classical error correction, extra qubits are needed to provide sufficient redundancy to detect and correct physical errors. The *threshold theorem* [108] shows that we can increase fidelity by increasing the number of physical qubits, but only if the error rate of a physical qubit is better than the threshold. One of the attractive properties of the surface code is that its threshold requirement is much higher than other stabilizer codes, which makes it more likely to apply to qubits that can be reliably deployed in the near term. Even so, it is generally estimated that something on the order of 1000 physical qubits will be needed for each logical qubit, in order to effectively execute long-running algorithms like the ones described in later chapters. With additional needs for logical ancilla qubits, magic states, and qubit relocation, the actual resource needs can be much higher. Resource estimates for integer factoring, for example, are given by Van Meter *et al.* [109], Fowler *et al.* [97], and Gidney and Ekarå [110].

Fault tolerance is a broader topic than error correction. Most of the discussion above assumes that error detection and correction circuits are error-free. Some techniques are relatively simple, such as designing circuits so that a single error is not propagated to multiple qubits. Techniques from classical fault-tolerant circuits can be helpful. Gottesman [111] and Nielsen and Chuang [9, Chapter 10] offer good discussions of fault tolerance for stabilizer codes. For surface codes, fault tolerance largely comes in the form of repeated rounds of error correction, but is also influenced by the ability of control software to correctly interpret error syndromes.

There are several additional types of QEC code that were not discussed here, building on both classical and quantum information theory. Examples include subsystem codes, such as Bacon-Shor [112], color codes [113], and convolutional codes [114] (designed for communication). There are also a number of flavors of surface code, using different versions and combinations of lattice surgery, defects, and twists [97, 101, 115, 116]. While most surface codes assume a square lattice, as shown in the previous sections, some vendors have used less dense topologies (see Figure 11.18 in the next chapter) to minimize frequency collisions, crosstalk, and other factors. Chamberland *et al.* [117] propose hybrid surface/Bacon-Shor code on a (heavy) hexagonal lattice, which is consistent with recent versions of the IBM superconducting systems.

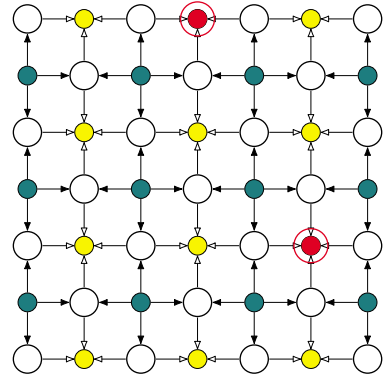
In summary, the surface code approach appears to be the best suited for superconducting quantum computers, because of their 2D, nearest-neighbor topologies. In addition, its relatively high threshold makes it more likely to be practical in the near term. However, systems have not yet scaled to the number of qubits needed for fault-tolerant quantum computing. Quantum error correction has been called the “biggest

flipping challenge” in quantum computing [118], and it is likely to remain an active area of research and development for many years.

10.9 Exercises

- 10.1 Demonstrate that the H gate is transversal for the $[[7, 1, 3]]$ Steane code: $\overline{H} = H^{\otimes 7}$.
- 10.2 Demonstrate that the S gate (rotation of $\pi/2$ around the z -axis) is transversal for the $[[7, 1, 3]]$ Steane code, such that $\overline{S} = (S^\dagger)^{\otimes 7}$.
- 10.3 Demonstrate that the T gate (rotation of $\pi/4$ around the z -axis) is *not* transversal for the $[[7, 1, 3]]$ Steane code.
- 10.4 Suppose the measure- Z qubits shown in Figure 10.31 indicate a change since the last surface code cycle. What physical X operations on data qubits would you perform to correct the error?

Figure 10.31 Errors in surface code for Exercise 10.4.



- 10.5 Using the array-based logical qubit in Figure 10.25, show that the logical $\overline{X}$ and $\overline{Z}$ operators anti-commute: $\overline{X}\overline{Z} = -\overline{Z}\overline{X}$.

11

Quantum Logic: Efficient Implementation of Classical Computations

In this chapter, we describe how to implement classical computations, such as addition and multiplication, using quantum gates. Of course, if your quantum program contains only classical operations on classical data, you're doing it wrong! We already have fast and inexpensive classical logic gates to do this, and a quantum program should exploit uniquely quantum features, such as phase, superposition, and entanglement.

However, there are very good reasons to consider efficient quantum implementations of classical logic. First, these operations are embedded as kernels in important quantum algorithms, such as Shor's algorithm for factoring large integers, and Grover's search algorithm. Second, classical computations can be performed on quantum information, and we will see how logical operations operate on general quantum states. If we had to transfer information from the quantum realm to the classical realm in order to perform arithmetic, we would lose the opportunity to utilize quantum parallelism.

As a motivating example, consider the quantum circuit in Figure 11.1. The function U_{ADD} is a three-bit adder: it takes the 3-qubit quantum state $|x\rangle$ and the 3-qubit quantum state $|y\rangle$ and produces $|(x + y) \bmod 8\rangle$, where x and y are interpreted as unsigned binary numbers. Actually, the output of the circuit is $|x, x + y\rangle$, because an output of $|x + y\rangle$ by itself would not be reversible: given only $x + y$, there's no way to undo the calculation to recover the inputs x and y . However, if we have both x and $x + y$, we can easily recover the other input y .

Therefore, if the inputs are $|3\rangle$ and $|2\rangle$, then the output would be $|3, 5\rangle$. It's tempting to think of $|x\rangle$ and $|x + y\rangle$ as independent states—the input and the output—but in reality they are *entangled*. If we measure $|x\rangle$ and get 3, then we know that measuring the other qubits will yield $3 + y$.

What happens for general quantum states? Suppose $|x\rangle$ is a superposition of 1 and 3: $|x\rangle = \frac{1}{\sqrt{2}}(|1\rangle + |3\rangle)$, and $|y\rangle = |2\rangle$. Then the output will also be a superposition:

$$|x, x + y\rangle = \frac{1}{\sqrt{2}}(|1, 3\rangle + |3, 5\rangle). \quad (11.1)$$

Again, note the entanglement. If we measure $|x\rangle$ and get 3, we know that measuring the other bits will yield 5.

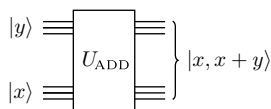


Figure 11.1 Three-qubit quantum adder.

Table 11.1 Boolean algebra operators.

Operator	Notation	Description
NOT	$\bar{a}$	Logical inverse: true if a is false.
AND	$ab, a \cdot b$	Conjunction: true if both a and b are true.
OR	$a + b$	Disjunction: true if either a or b is true.
XOR	$a \oplus b$	Exclusive-OR: true if either a or b is true, but not both.

What if both inputs are superpositions?

$$\begin{aligned} |x\rangle &= \frac{1}{\sqrt{2}}(|3\rangle + |6\rangle) \\ |y\rangle &= \frac{1}{\sqrt{2}}(|1\rangle + |4\rangle) \\ |x, x + y\rangle &= \frac{1}{2}(|3, 4\rangle + |3, 7\rangle + |6, 7\rangle + |6, 2\rangle). \end{aligned}$$

(11.2)

It seems like we’re computing four sums with one execution of the quantum function, and that’s true. That’s quantum parallelism! Unfortunately, we only get to use one of the results. Once we measure the output, the other three states are destroyed.

In the rest of this chapter, we’ll discuss how such circuits are built from quantum gates. We will almost exclusively consider quantum states in the *computational basis*, that is $|0\rangle$ and $|1\rangle$, as well as superpositions of those states. We will also use standard Boolean algebra notation, as shown in Table 11.1. In Section 11.4, we will consider logic operations that affect the *phase* of the quantum states, which will be quite useful for quantum algorithms.

11.1 Reversible Logic

A *reversible* function is one that can be “uncomputed”—given the outputs, it is possible to recreate the inputs. Early interest in reversible computing [119] was driven by a desire to minimize the energy dissipated during computation. For our purposes, however, we note that any operation performed on a qubit must be reversible. Therefore, any computational function must be built from reversible quantum gates, and the function itself must be reversible.

11.1.1 Reversible Logic Gates

In classical logic, some gates are naturally reversible. For example, a NOT gate can be reversed by another NOT gate (Figure 11.2).

For other gates, some or all of the inputs must be carried forward, so that the full set of inputs can be deduced from the output. For example, Figure 11.3 shows a reversible exclusive-OR (XOR) gate. Given the outputs x and $x \oplus y$, the original input can be

Figure 11.2 A NOT gate is reversible. By using another NOT gate, we recover the original input x .

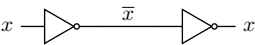
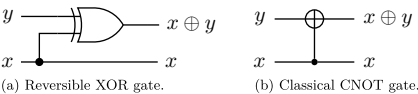


Figure 11.3 The reversible XOR gate requires that one of the inputs be carried over to the output. The result is the classical equivalent of the quantum CNOT gate.



computed as $y = x \oplus (x \oplus y)$. This gate is the classical equivalent of the quantum CNOT gate introduced in Chapter 1.

The NOT and CNOT gates are examples of *in-place* reversible circuits, where some of the inputs are replaced by outputs. In general, however, this may not be possible. Consider the AND gate in Figure 11.4. If we only carry one of the inputs (x or y) through the gate, then we can't reverse the computation. If xy is zero, it could be because $x = 0$ or $y = 0$ or both. Therefore, we have to carry both inputs through to the output, and we must introduce a new input (typically initialized to zero) that will be replaced with the computational output.

The *Toffoli gate*, shown in Figure 11.5, is a universal gate for reversible computing. The Toffoli is essentially a CNOT gate with two control bits: if both control bits are 1, the target bit is flipped. In other words, for control bits x and y and target bit z , the Toffoli gate computes $xy \otimes z$.

We say that the Toffoli gate is “universal” because we can implement a NAND gate, as shown in Figure 11.6(a). Any Boolean combinational circuit can be implemented using only NAND gates. Toffoli-based implementations of other fundamental Boolean gates are shown in the figure. Note that the OR implementation uses *negative control* bits; this is just a shortcut notation for putting a NOT on the control signal before and after the gate.

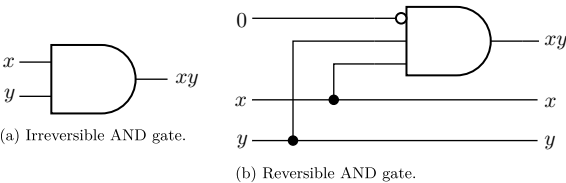


Figure 11.4 To make an AND gate reversible, we must remember both of its input bits. Otherwise, for $xy = 0$, there's no way to know whether $x = 0$, $y = 0$, or both.

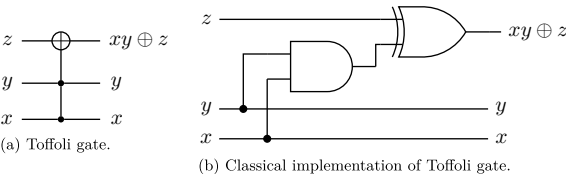


Figure 11.5 The Toffoli gate has two control bits and a target bit. The target is flipped if and only if *both* control bits are 1. The gate symbol for Toffoli is shown in (a), while (b) shows an implementation using classical gates.

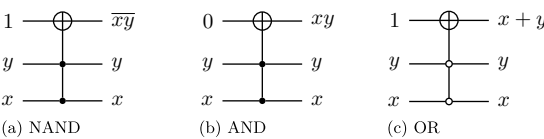


Figure 11.6 The Toffoli gate is universal for Boolean logic, because it can be used to implement the NAND gate. Implementations of several fundamental gates are shown here. An open circle is a *negative* control; the control is active if the input bit is 0.

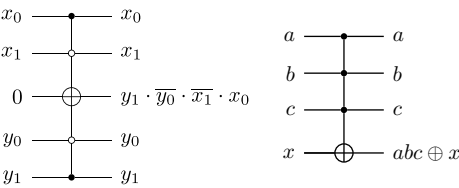


Figure 11.7 Examples of multi-controlled Toffoli (MCT) gates.

The Toffoli is essentially a controlled-CNOT, so it is sometimes called CCNOT or C^2 NOT. The concept can be extended to more than two control bits, known as a multi-controlled Toffoli (MCT) or C^k NOT for k control bits. In general, the control bits can be either positive or negative. Also, the target bit does not have to be on the bottom or top—it can be anywhere. Examples of MCT gates are shown in Figure 11.7. While MCT gates are convenient for expressing complex logic functions, they can be expensive to implement, especially using quantum gates, as we’ll see in Section 11.2.

11.1.2 Reversible Logic Circuits

A reversible combinational logic circuit has the following characteristics: (1) The number of input signals is equal to the number of output signals. (2) There are no loops. (3) There is no explicit fanout; an output of a gate can only be connected as input to one other gate.

With regard to (3), we can simulate fanout using a CNOT gate, as shown in Figure 11.10. Classically, the input value x is replicated to both output bits, and this copying can be reversed using another CNOT gate. In the quantum realm, things are a bit more complicated, but we’ll address that in the next section.

When extra bits are needed to implement a reversible circuit, these are known as *ancilla* bits. Ancilla bits may be initialized to 0 or 1. Sometimes ancilla bits are used to carry the output of a gate, as in the AND gate in Figure 11.4. Other times, ancillae are used to carry intermediate results from one gate to another in a reversible circuit. Ancilla bits that are not needed for the output must be restored to their original values; this restoration must be done by *uncomputing* the bit, because simply resetting it to 0 or 1 is not a reversible operation. If ancilla bits are not uncomputed, the circuit may not be reversible.

The use of ancilla bits is illustrated in Figure 11.8. The circuit implements a four-way AND using only Toffoli gates. Ancilla bits (t_1 and t_0) are needed to carry the output from one gate to the input of the next. After the computation, the ancilla bits have intermediate values, which means they cannot be used again for other parts of a larger circuit. Figure 11.9 shows how the ancilla bits are uncomputed by reversing all gates

Figure 11.8 Reversible four-input AND circuit. The temporary values held in the ancilla bits t_1 and t_0 make this circuit unusable as a general reusable computing block; the ancilla bits cannot be reused in other parts of a larger circuit.

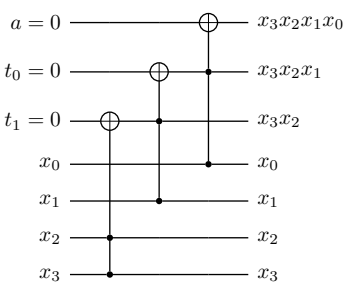
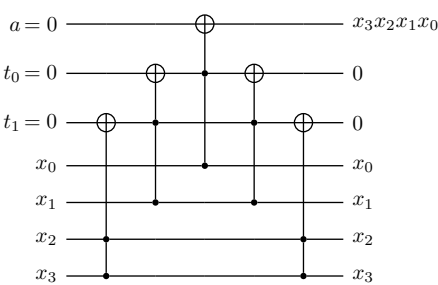


Figure 11.9 Reversible four-input AND circuit; ancilla bits t_1 and t_0 are uncomputed and restored to zero.



except the one generating the desired output. Note that this circuit implements a four-way MCT, with four control bits (x) and one target bit (a).

Toffoli [120] developed a general technique for implementing a reversible circuit for an irreversible function $f(x)$, shown in Figure 11.11. Suppose the function $f(x)$ takes an m -bit input x and produces an n -bit output value. The reversible circuit takes an m -bit input x and an n -bit input y and produces x and $y \oplus f(x)$. This is guaranteed to be reversible, because using the output $y \oplus f(x)$ as the input (along with x), yields $y \oplus f(x) \oplus f(x) = y$. Typically, the y input is initialized to 0, as shown in part (b) of the figure.

Figure 11.12 sketches a generalized procedure for generating the type of circuit shown in Figure 11.11. If we have a circuit F that computes $f(x)$ using only NOT and AND gates, we can replace the AND gates with Toffoli gates, adding ancilla bits as needed. The output bits from F include both the desired result and some unneeded “garbage” bits: temporary values and possibly changed input bits. The desired output

Figure 11.10 Reversible fanout using a CNOT. While this works for classical circuits, copying (cloning) of an arbitrary state is not permitted in quantum circuits (Section 11.2).

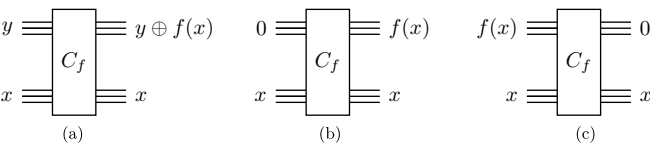
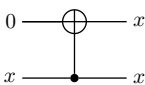


Figure 11.11 General reversible circuit (a) for irreversible function $f(x)$. Using zero inputs (b) generates $f(x)$, and the circuit is its own inverse (c).

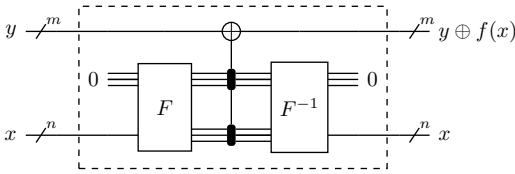


Figure 11.12 Sketch of a generalized reversible circuit. Circuit F replaces AND gates with Toffoli gates, using additional ancilla bits as needed. Some CNOT gates (illustrated by the MCT in the middle) XOR the results with the y bits to produce the output $y \oplus f(x)$. Then circuit F^{-1} uncomputes the ancilla bits and restores the x bits as needed.

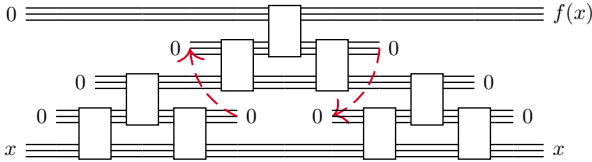


Figure 11.13 Reuse of ancilla bits (dashed lines) reduces the number of bits/qubits required for the circuit. (Adapted from [8].)

bits are copied using CNOT gates (represented by the MCT gate in the middle of the figure), and then garbage bits are uncomputed using F^{-1} , which is the same circuit as F , except the order of gates is reversed. The ancilla bits are part of the implementation of C_f and are therefore inside the dashed box; they are not visible to the rest of the circuit. The outputs are therefore x (the original input) and $y \oplus f(x)$.

As the combinational circuit gets larger, uncomputing temporary bits allows them to be reclaimed and reused, reducing the number of extra bits needed. Using this technique, illustrated in Figure 11.13, Bennett [121] proves that an irreversible Boolean circuit using t steps and s bits can be implemented as a reversible circuit with $O(t^{1+\epsilon})$ steps and $O(s \log t)$ bits. In other words, a reversible version of a Boolean circuit requires roughly the same order of steps (time) and a moderate increase in the number of bits (space).

11.2 Quantum Logic Circuits

With a basic understanding of reversible logic, we can now discuss how to do arithmetic and logic operations on qubits. If we restrict ourselves to qubits in the *computational basis* states— $|0\rangle$ and $|1\rangle$ —then the same principles apply, and we build quantum circuits using the same techniques. However, there are a few considerations worth noting.

- If the inputs are general quantum states, then the action of the gates often results in entangling the states. This creates useful behavior that we can exploit, but it also makes uncomputing (and disentangling) temporary states even more important.
- The implementation of multi-qubit gates, like Toffoli and MCT, is non-trivial using one- and two-qubit gates. This means that even simple-looking circuits from Section 11.1 can require a significant amount of quantum resources.
- Physical qubits must be coupled to each other in order to interact via a two-qubit gate. This often requires the insertion of SWAP gates to rearrange qubits during the execution of a circuit, increasing both time and the probability of error.

11.2.1 Entanglement and Uncomputing

In Figure 11.10, we showed how fanout can be achieved in a reversible circuit using a CNOT gate. However, this appears to violate the *no-cloning* theorem that was introduced in Section 1.1.3: there is no unitary transform that will copy the state of one qubit to another.

If the input state is known to be $|0\rangle$ or $|1\rangle$, cloning works just fine. In Figure 11.14(a), the two-qubit input is $|x0\rangle$ and the two-qubit output state is $|xx\rangle$.

On the other hand, if the input state $|x\rangle$ is a superposition, then its state is not cloned—rather, the output is an *entangled* state. For example, consider Figure 11.14(b), where $|x\rangle = \alpha|0\rangle + \beta|1\rangle$. The two-qubit input state is $\alpha|00\rangle + \beta|10\rangle$, which is not entangled, but the output state is $\alpha|00\rangle + \beta|11\rangle$. This is an entangled state, because it cannot be factored into two independent qubit states.

Figure 11.15 illustrates how general quantum states are affected by a Toffoli gate. Again, the three qubit states are not separable. If the output (top) bit is measured as 1, then the 3-qubit state is $|111\rangle$ —in other words, the control qubits are known to both be $|1\rangle$. Even if we measure 0, we know something about the control bits: at least one of them will be measured as 0.

Since many quantum inputs are actually superpositions, this means that the qubits are often entangled with one another after the circuit is executed.¹ This is generally a good thing, especially for circuits that produce a state like $|x\rangle|f(x)\rangle$. When we measure a certain classical value $f(x)$, we can be sure that the x that we measure is the one that matches the result $f(x)$.

However, the entanglement also applies to temporary values (ancilla qubits) used during the execution of the circuit. This makes uncomputing even more important for quantum circuits than for classical reversible circuits. If an entangled ancilla qubit is measured or experiences a decoherence error, it can affect the state of a qubit in the primary output or in another part of the computation. Uncomputing has the effect of

Figure 11.14 A CNOT can be used to replicate computational basis states (a), but for a superposition, the resulting state is entangled (b).

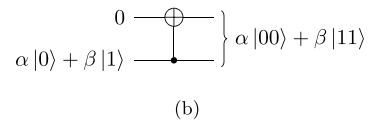
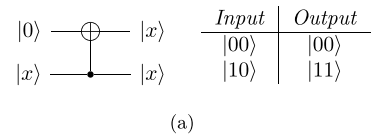
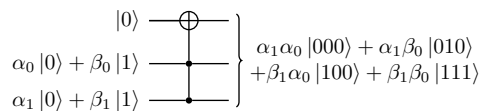


Figure 11.15 Action of Toffoli gate on superposition states.



¹ The notion of *input* and *output* of a quantum circuit is somewhat misleading, as the circuit is executed in time rather than in space. The terms *before* and *after* are more accurate, but the input/output terminology is commonly used.

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \left\{ \begin{array}{c} \text{---} \bullet \text{---} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ | \\ \text{---} \oplus \text{---} |0\rangle \end{array} \right.$$

Figure 11.16 An entangled Bell state is disentangled by a CNOT gate.

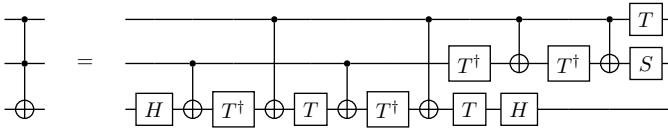


Figure 11.17 Implementation of Toffoli gate [9].

disentangling the qubit from the primary input/output qubits. As an illustration, consider what happens to the entangled Bell state in Figure 11.16; after the CNOT gate, the two qubits are no longer entangled.

11.2.2 Multi-Qubit Gates

While circuits using Toffoli and MCT gates appear simple, like the ones in Figures 11.6 and 11.7, we must understand that quantum computers typically only directly implement one- and two-qubit gates. In fact, each computer provides CNOT and some basic set of one-qubit gates, and all other gates are implemented in terms of these “native” gates.

Figure 11.17 shows an implementation of the Toffoli gate using CNOT, Hadamard, and phase rotations. The S gate is a rotation of $\pi/4$ around the z -axis of the Bloch sphere. The T and $T^\dagger$ gates are z -rotations of $\pi/4$ and $-\pi/4$, respectively. Proof of this implementation is beyond the scope of this chapter,² but the important takeaway is that a single Toffoli gate requires six CNOT and multiple single-qubit gates, with a depth of 13. This has significant implications for NISQ computers, because the higher depth means a higher probability of error.

MCT and other multi-controlled gates require multiple Toffoli gates, as well as multiple ancilla qubits to hold temporary results, as shown in Figure 11.9. Fewer ancilla qubits can be used by uncomputing and reusing them, but the uncomputing introduces more gates and results in a higher circuit depth. Current quantum computers are limited in both the number of bits (width) and the number of gates (depth), so this tradeoff must be carefully managed.

11.2.3 Qubit Topology

In our circuits so far, we have been free to connect any qubit with any other qubit using gates. However, two qubits need to be physically connected in order to directly interact using a two-qubit gate, like CNOT. Because of physical limitations, the qubits are not fully-interconnected, as shown in Figure 11.18. The connections are chosen by the designers of the quantum system, based on physical limitations, noise characteristics, and other considerations.

2 See [9, chapter 4] for a more complete discussion.

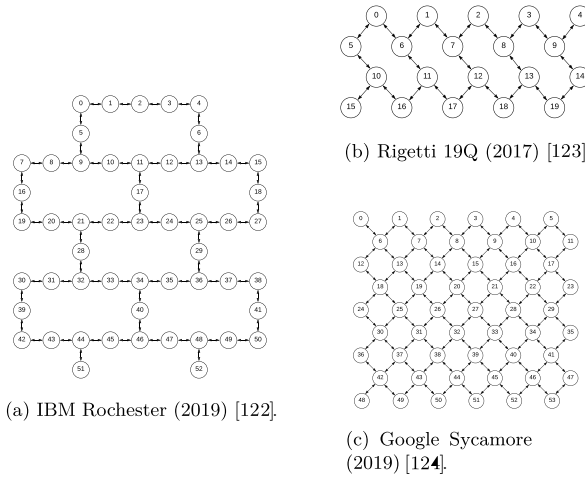


Figure 11.18 Qubit topologies for some superconducting quantum systems.

The connections in the figure are shown as bidirectional, implying that either qubit can be used as the *control* qubit in a CNOT. In fact, this may not be the case. For example, a desired control qubit may not have a control signal that would allow it to drive the connection bus at the target qubit's frequency. In this case, Hadamard gates can be used to reverse the direction of the CNOT as shown in Figure 11.19.

Because of this limited connection topology, when implementing a quantum circuit on an actual quantum system, we have to be aware of which qubits are connected to which, and we can only perform CNOT operations on physically-connected qubits. Fortunately, if we want two non-connected qubits to interact with one another, we can move (not copy!) the quantum state to physical qubits that are connected. This is done using a SWAP operation.

SWAP is a reversible two-qubit operation, and it has its own symbol, shown on the left-hand side of Figure 11.20. It is typically not implemented natively, however, but through the use of three CNOT gates. As shown in the figure, we can use Hadamard gates, if needed, to reverse the direction of one of the CNOT gates.

Figure 11.21 shows an example circuit in which we need to perform a CNOT between the bottom qubit and the top qubit. Suppose that only neighboring qubits are connected to each other. We need to insert SWAP gates to move the desired qubit states next to each other.

Figure 11.19 The direction of a CNOT gate can be reversed using Hadamard gates.

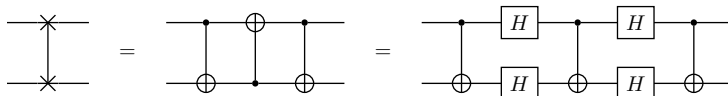
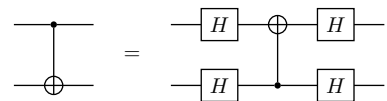
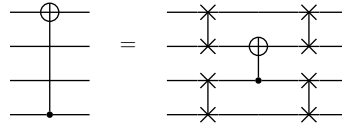


Figure 11.20 The SWAP gate exchanges the state of two qubits. It can be implemented using three CNOT gates.

Figure 11.21 SWAP gates may be needed to move qubit states into a position where they can interact. In the figure, we move the top and bottom qubits to the middle, perform a CNOT, and then move them back.



other, then perform the desired CNOT, and then swap again to move the result back to the original locations.

SWAP insertion increases the depth of a quantum circuit, so it is very important in NISQ systems to match the circuit as closely as possible to the system's topology. In most quantum programming environments, the SWAP insertion is performed by the *compiler* that translates a circuit diagram or high-level language into low-level machine instructions for the quantum system. The programmer can happily assume a fully-connected topology, and part of the compiler's job is to map the *logical* qubits in the program/circuit to *physical* qubits in system, in a way that minimizes the number of SWAP gates that must be inserted. For example, the compiler may decide to leave the qubits in Figure 11.21 where they are, after the CNOT gate, rather than moving them back, transforming the downstream circuit accordingly.

11.3 Efficient Arithmetic Circuits: Adder

In this section, we focus on implementing common arithmetic operations on unsigned binary integers, using a *binary adder* as an example. The circuits also work for two's complement signed integers, but the explanation will be simpler if we only consider unsigned numbers. An integer a is represented by a sequence of bits $a_{n-1}a_{n-2} \dots a_1a_0$, where a_i has a weight of 2^i . Therefore, an n -bit integer represents decimal values between 0 and $2^n - 1$.

For our quantum circuits, we assume that our input qubits carry only classical information—that is, each qubit is $|0\rangle$ or $|1\rangle$. As discussed at the beginning of this chapter, the circuits will also work with any superposition of computational basis states, but we will keep the examples simple by only considering a single-valued state.

For an n -bit binary integer a , the notation $|a\rangle$ represents the n -qubit state

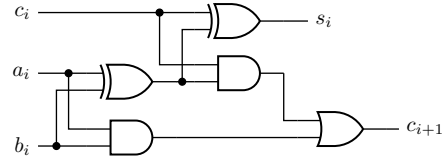
$$|a_{n-2}a_{n-1} \dots a_0\rangle, \quad (11.3)$$

where a_i is the bit in position i , and position 0 is the least significant bit.

For the adder, we take two unsigned n -bit binary numbers a and b and produce an $n+1$ -bit sum: $s = (a+b)$.³ The simplest classical implementation of an adder, shown in Figure 11.22, is a collection of one-bit *full adder* circuits. For each bit position i , input bits a_i and b_i are added with a carry-in bit c_i from the previous stage, producing a sum bit s_i and a carry-out bit c_{i+1} , which serves as the carry-in to the next bit position. Bit c_0 is assumed to be zero, and the carry-out of position $n-1$ (c_n) becomes the most significant sum bit s_n . This is known as a *ripple-carry adder*, because the carry propagates sequentially from one bit position to the next.

³ In many digital implementations, it is common to drop the most significant output bit to keep the number of bits as n . For unsigned integers, this would compute the sum modulo 2^n .

Figure 11.22 Classical non-reversible one-bit full adder.



11.3.1 Quantum Ripple-Carry Adder

Figure 11.23 shows a simple quantum implementation of a one-bit adder. The circuit on the left is a *half adder*, with no carry input. This is suitable for the least significant bit in an adder circuit, because the carry-in is zero and can be ignored. Note that we do not simply implement the classical Boolean logic gates using Toffoli gates, as discussed in Section 11.1.1.

The sum bit is calculated using two CNOT gates: if either $|a_0\rangle$ or $|b_0\rangle$ is $|1\rangle$, then the sum qubit will be flipped to $|1\rangle$. If both are one, then the sum qubit is flipped twice, resulting in the correct output of $|0\rangle$. For the half-adder, the carry-out is $|1\rangle$ only if both inputs are $|1\rangle$. This can be implemented using a single Toffoli gate.

A *full adder*, shown in the right half of the figure, is needed for the higher-order bits in the computation. The carry-out from the previous stage becomes an additional input to this stage. The sum requires only one additional CNOT gate—any odd combination of $|1\rangle$ inputs will result in a $|1\rangle$ output. The carry-out requires two additional Toffoli gates to implement the expression $ab \oplus ac \oplus bc$.

An n -bit adder can be implemented by replicating the full adder circuit, using the carry-out bit from the previous position as the carry-in of the next. A three-bit adder is shown in Figure 11.24. Note that the internal carry bits (except for $|c_n\rangle$) are not part of the desired output; they are ancilla bits. If we want to reclaim them for later use, or if we are concerned about entanglement of carry bits with other bits, we need to uncompute to reclaim them, as shown in Figure 11.25. Note that the carry bits can only be uncomputed in reverse order: since $|c_i\rangle$ is needed to uncompute $|c_{i+1}\rangle$, it cannot be uncomputed until after $|c_{i+1}\rangle$ is uncomputed.

In order to evaluate the efficiency of a quantum circuit, we measure the number of qubits, the number of gates, and the depth of the circuit. Since the input and output qubits of a particular computation are always the same, we focus on the the number of ancilla bits. For size and depth, we will assume that single-qubit, two-qubit, and Toffoli gates will all count as a single gate (even though we know that Toffoli gates require multiple elementary gates). The size is the total number of gates in the circuit, and the depth is the longest sequence of gates that must be applied.

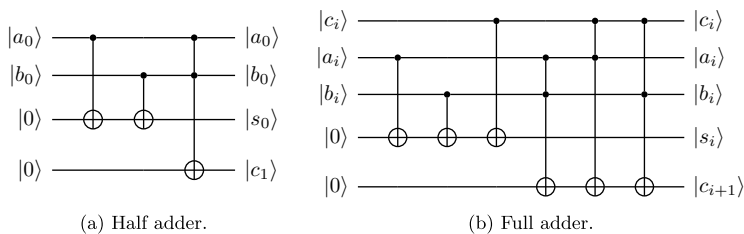


Figure 11.23 Simple quantum adder. Carry output from bit i becomes an input for bit $i + 1$.

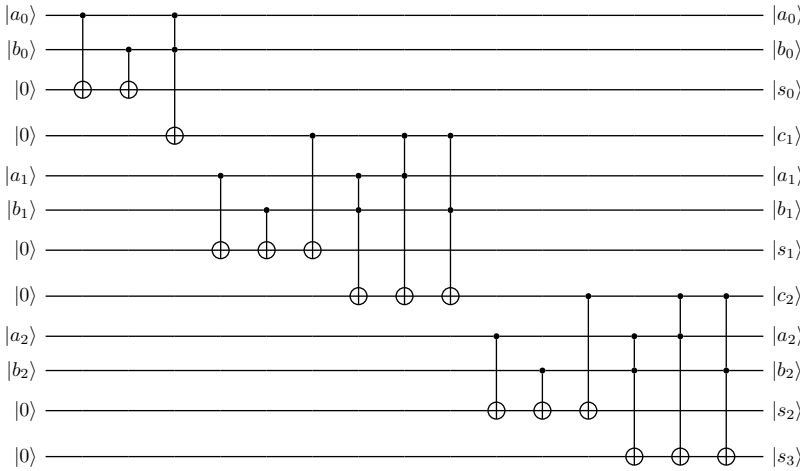


Figure 11.24 Three-bit ripple-carry adder.

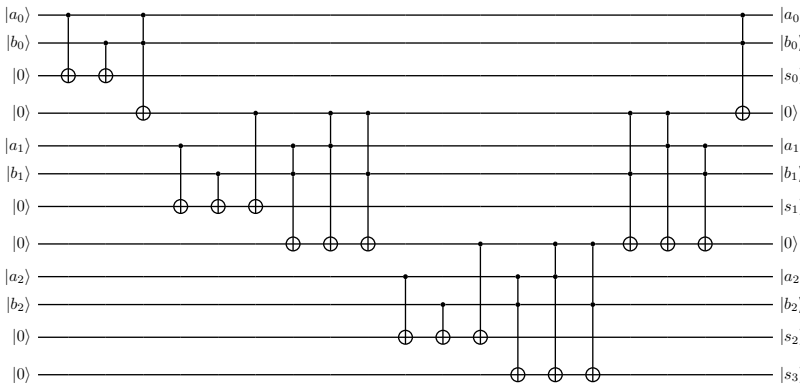


Figure 11.25 Three-bit ripple-carry adder with reclaimed ancillas.

By inspecting Figure 11.25, we can estimate the efficiency of the ripple adder circuit. For the estimate, we'll ignore the half adder and assume that every bit of output requires one full adder. Each full adder requires six gates, plus another three gates to reclaim the ancilla bit. Therefore the total size is roughly $9n$, which is $O(n)$. Each full adder increases the number of ancilla bits by one, so there are also $O(n)$ ancilla bits required for the carries. Finally, gates in a full adder are applied sequentially, so the depth of each adder is 6. Potentially, the last CNOT of the sum and the first Toffoli of the carry-out can be done in parallel, but this has little real effect, since the Toffoli is in reality much more expensive than the CNOT. Also, as shown in Figure 11.24, each full adder depends on an input from the previous one. Therefore, the depth is overall $O(n)$.

We can improve the depth of our simple adder circuit by performing the non-carry part of the sum computations in parallel, as shown in Figure 11.26. In addition, the part of the carry computation that relies only on $|a_i\rangle$ and $|b_i\rangle$ can be done in parallel. However, the depth remains $O(n)$ because of the dependency on the carry bits. This sort

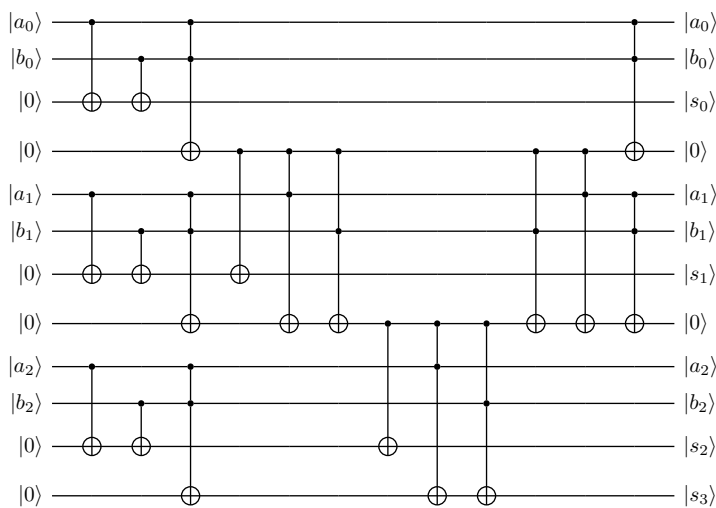


Figure 11.26 Three-bit ripple-carry adder, redrawn to better show circuit depth.

of rearrangement of the gates to extract parallelism is typically done by the compiler in a quantum programming environment.

11.3.2 In-Place Ripple-Carry Adder

In the previous adder, both input arguments are carried through to the output, so that we end up with a , b , and $a + b$. However, this is not necessary for the circuit to be reversible. If we only have a and $a + b$, we can easily reverse the computation to recover b . Cuccaro *et al.* [125] designed a ripple-carry adder in which b is replaced by the sum. In addition, the adder requires only one ancilla bit, rather than $O(n)$.

The adder is built using a pair of three-qubit functions, known as MAJ and UMA, shown in Figure 11.27. MAJ is an in-place “majority” function: given three input qubits, the output (on the bottom) is $|0\rangle$ if two of the input bits are $|0\rangle$, and $|1\rangle$ otherwise.

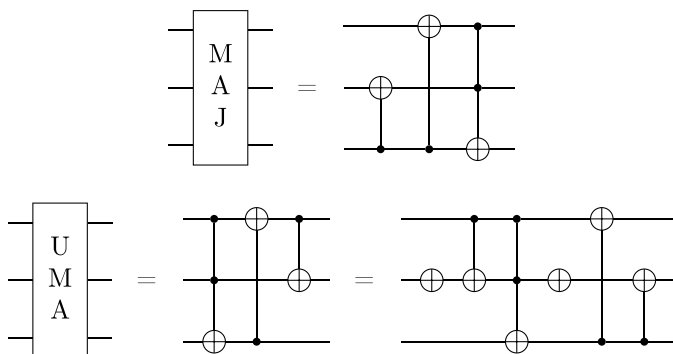


Figure 11.27 MAJ (majority) and UMA (unmajority and add) functions for in-place ripple-carry adder. The three-CNOT version of UMA on the right is used in the optimized adder circuit.

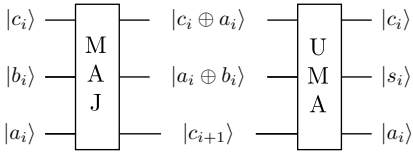


Figure 11.28 Full adder built with MAJ and UMA.

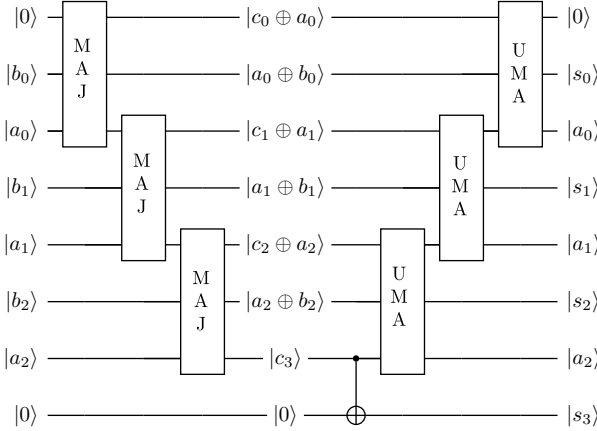


Figure 11.29 Three-bit ripple adder with one ancilla bit.

Logically, for inputs a_i , b_i , and c_i , this computes $a_i b_i \oplus a_i c_i \oplus b_i c_i$, which is the same as c_{i+1} in the adder described above. The other output bits are shown in the figure.

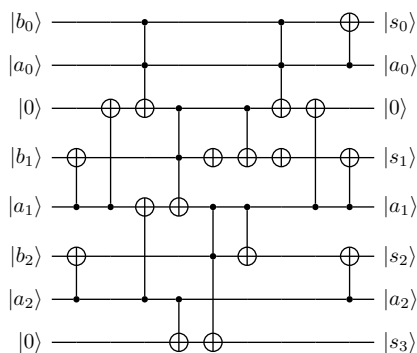
The UMA function stands for “Unmajority and Add.” This function restores the top and bottom inputs to their previous values (c_i and a_i , respectively), and uses c_i to conditionally flip the middle bit, replacing b_i with the sum bit s_i .

To create an n -bit adder, we use the carry-out of one MAJ as the carry-in of the next MAJ. After n repetitions, we have calculated the high-order carry-out c_{i+1} . We make a copy of the carry-out bit to become the high-order sum bit s_n . (Recall that we can copy a qubit in the computational basis without violating the no-cloning rule.) Then we use a series of UMA functions to uncompute the s_i and a_i bits. Figure 11.29 shows the implementation of a three-bit adder using this construction.

We can improve the circuit in Figure 11.29 by performing some reordering and optimizations, described in more detail in [125]: (1) Since we know that c_0 is zero, we can simplify the top adder and move the ancilla bit to hold c_1 instead. (2) Shift and reverse the order of some gates (without affecting the computation) to perform gates in parallel as much as possible. (3) Generate the high-order sum bit (s_3) directly, rather than computing, copying, and uncomputing it. The optimized circuit is shown in Figure 11.30.

Compared to Figure 11.26, the number of qubits required by this adder is significantly reduced, because we don’t need the carry ancilla bits, and we don’t need both b_i and s_i in the output. In addition, the depth is reduced, although it is still $O(n)$. (Takahashi and Kunihiro [126] describe a ripple-carry adder with zero ancilla bits and fewer gates, also with $O(n)$ depth.)

Figure 11.30 Optimized three-bit ripple adder with one ancilla bit.



Could we expect a compiler to take a programmer-friendly description like Figure 11.29 and optimize it into Figure 11.30? It seems feasible. Recognizing that c_0 is zero and removing it is roughly equivalent to *constant propagation*. Directly computing s_3 instead of c_3 is a form of *dead expression elimination*. Finally, changing the order and timing of gates is based on *dependency analysis*. All of these are standard optimizations performed by classical compilers on sequential code, so there's reason to believe they can be performed on quantum gates (instructions), as well.

Could we expect a compiler to take the naive ripple-carry adder from Section 11.3.1 and transform it into Figure 11.30? That seems less likely, because it involves algorithmic changes. At this stage of quantum software maturity, the design of optimized algorithmic building blocks is more similar to digital logic design than high-level programming. We will still rely on talented algorithm designers to creatively optimize gates and map to quantum hardware.

Finally, there is one important concern to emphasize: the organization of qubits in the input and output values. Instead of starting with two n -qubit values, the circuit assumes that a_0 is next to b_0 , a_1 is next to b_1 , and so on. Of course we can redraw the circuit to group all the a and b qubits together, but we also know that in practice these bits do need to be close to each other in order to interact using CNOT gates. The sequence of SWAP gates necessary to perform such a rearrangement is explored in Exercise 11.4, but it's worth noting that both the movement of qubits and the complexity of Toffoli gates will increase the actual cost of performing this fundamental arithmetic computation.

11.3.3 Carry-Lookahead Adder

Several variations of fast adders have been developed for classical logic, typically adding gates to reduce the depth of the circuit. One specific approach is a carry-lookahead adder (CLA), which allows carry bits to propagate more quickly through the circuit. Instead of calculating the carry from each bit position in conjunction with the sum bit, the CLA computes the carry-out for a group of bits as fast as possible. This group carry is passed to the next group of bits, so that the sum bits of both groups can be computed in parallel (using the standard ripple-carry approach).

Figure 11.31 shows the carry-out logic for a four-bit CLA in classical digital logic. Each pair of input bits, a_i and b_i , is used to generate two intermediate bits. The *generate* bit, g_i , is true when the inputs will definitely generate a carry-out: when a_i and b_i are

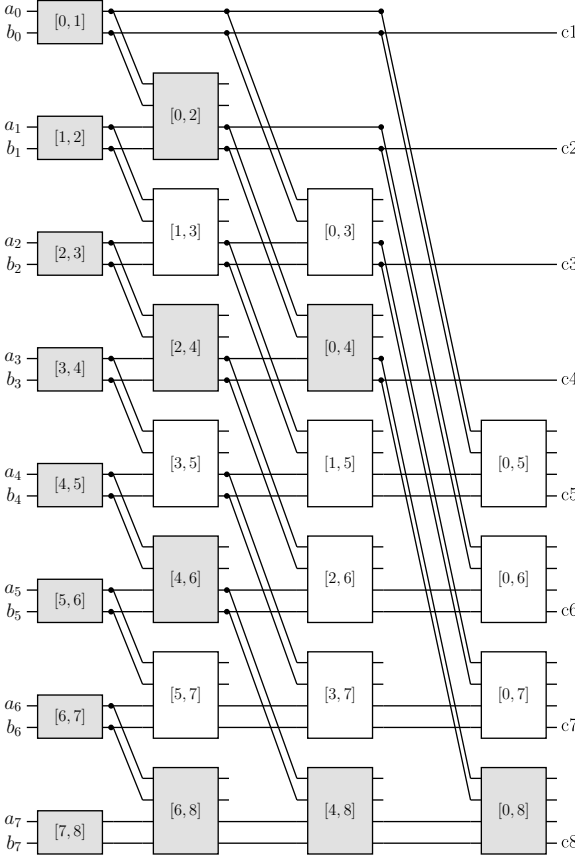


Figure 11.31 Carry-lookahead circuit for 8-bit adder. All carry bits are generated using $O(\log n)$ levels of logic. Block $[i, j]$ calculates both propagate $P[i, j]$ and generate $G[i, j]$ signals. To compute $c_8 = G[0, 8]$, only the shaded blocks are required.

both one, then there will definitely be a carry-out, no matter whether c_i is 0 or 1. The *propagate* bit, p_i indicates whether an incoming carry of 1 would be passed along to the next higher bit; this is true if either a_i or b_i is true.

$$g_i = a_i b_i \quad (11.4)$$

$$p_i = a_i \oplus b_i. \quad (11.5)$$

The p_i and g_i bits can be calculated in parallel, because they only depend on the input bits, not on the output of any bits in a previous stage. Note: We use an XOR gate to compute p_i to clearly distinguish generate from propagate. With this definition, it is never the case that both p_i and g_i will be 1.

Now we consider the notions of generate and propagate over a range of bits $[i, j]$, where $0 \leq i < j < n$. We define $P[i, j]$ to be 1 if a carry is propagated from bit i to bit j . In other words, if the carry-in to position i is 1, and exactly one input bits is 1 in all bits i to j , then that carry would be propagated from i to j .

$$P[i, i + 1] = p_i = a_i \oplus b_i \quad (11.6)$$

$$P[i, j] = P[i, t]P[t, j], \text{ for any } i < t < j. \quad (11.7)$$

We define $G[i, j]$ to be 1 if there is a carry generated between bits i and j . This would be true if the carry is generated at bit j , or is generated at $j - 1$ and propagated at j , or is generated at $j - 2$ and propagated at both $j - 1$ and j , and so on. The carry-out for an n -bit adder would be $G[0, n]$.

$$G[i, i + 1] = g_i = a_i b_i \quad (11.8)$$

$$G[i, j] = G[i, t]P[t, j] \oplus G[t, j], \text{ for any } i < t < j. \quad (11.9)$$

We can also rewrite the definition of the sum (s) bits using only P and G .

$$s_0 = a_0 \oplus b_0 = P[0, 1] \quad (11.10)$$

$$s_i = a_i \oplus b_i \oplus c_i = P[i, i + 1] \oplus G[0, i], 0 < i < n \quad (11.11)$$

$$s_n = G[0, n]. \quad (11.12)$$

It is clear, then, that we need to calculate $G[0, i]$ for all values of i in order to compute the sums. This seems like a linear process, since each G may depend on the g_i and p_i values for all lower-order bits. However, we can compute non-overlapping values of $G[i, j]$ in parallel, and combine them using the equation above. In particular, we can compute in groups of 2, then 4, etc., until all the necessary values are generated.

As shown in Figure 11.31, all carry bits $G[0, i]$ can be calculated with $O(\log n)$ levels of logic. This is much faster than the ripple-carry adder, but it also requires a lot more gates. A compromise is to divide the values into fixed-size chunks—e.g., 8 bits—and use CLA to calculate only the high-order carry-out signal, c_8 , which requires only the shaded blocks in the figure. This can be used as the carry-in to the next group of bits while the carry signals internal to this group are rippled as before. (The first stage $[0, 1]$ would require another input, c_0 , as the input carry from the previous group.) This allows the ripple-carry of each group to be “activated” earlier, and the ripple-carry adders can proceed largely in parallel.

A quantum carry-lookahead adder operates according to the same principles. First, local P and G values are generated, which are then combined to create P and G values for larger and larger groups. The group size is doubled in each step, until all $G[0, j]$ bits (which are the carry-in c_j bits) are generated. Finally, the c bits are combined with a and b to generate the sum.

Figure 11.32 shows an 8-bit adder, as designed by Draper *et al.* [127]. This version is an “out-of-place” adder, in which additional qubits are provided for the sum. For an n bit adder, we have $2n$ input qubits, $n + 1$ output qubits, and $O(n - \log n)$ ancilla bits. An in-place adder is also an option, in which the sum overwrites one of the inputs, but additional ancilla bits are required; the total number of qubits is essentially the same, and the depth of the circuit is higher.

The INIT phase of the circuit calculates the local G and P values:

$$G[i, i + 1] = a_i b_i \quad (11.13)$$

$$P[i, i + 1] = a_i \oplus b_i. \quad (11.14)$$

Note that b_i is replaced by $P[i, i + 1]$.

Next, in the P phase, the local P values are combined into group propagate values. For example, we use $P[2, 3]$ (which was $|b_2\rangle$) and $P[3, 4]$ (which was $|b_3\rangle$) to compute $P[2, 4]$. In the first round, we compute $P[2, 4]$, $P[4, 6]$, and $P[6, 8]$. In the second

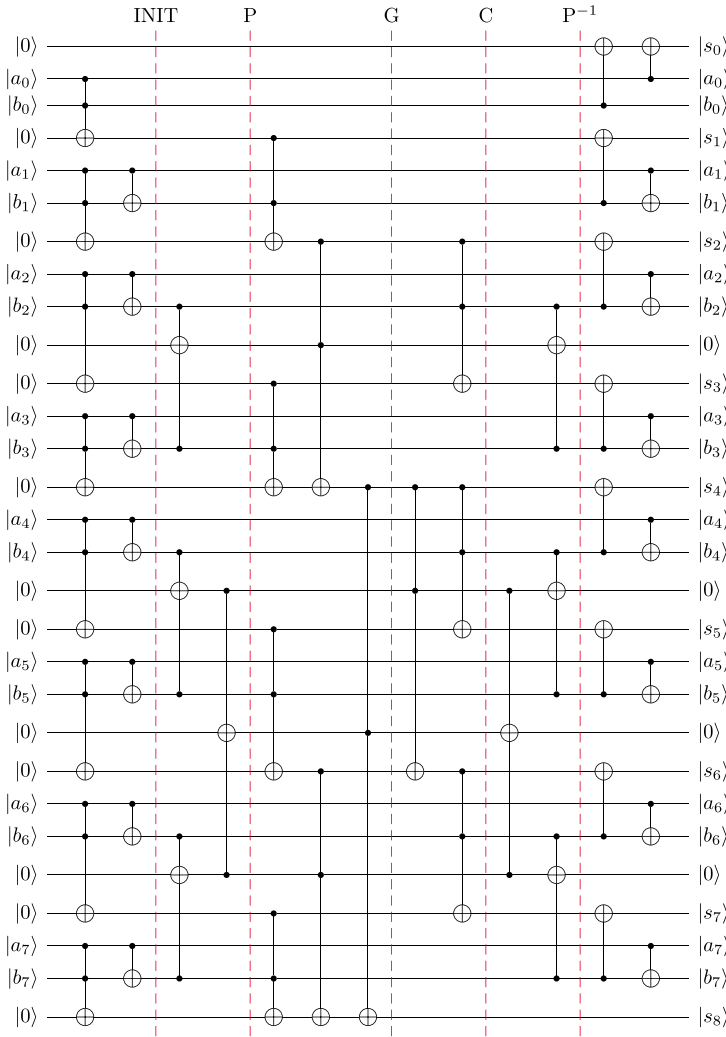


Figure 11.32 Quantum carry-lookahead adder for 8-bit integers [127].

round, we compute $P[4, 8]$. In general, there are $\lceil \log n \rceil - 1$ rounds to generate all of the necessary P values; in each round the size of the group doubles.

In the G phase, we compute selected G values using this relationship:

$$G[i, j] = G[i, t]P[t, j] \oplus G[t, j]. \quad (11.15)$$

This requires $\lceil \log n \rceil$ rounds, in which we again double the size of the group used in each round's calculation. Consider the top Toffoli gate in each round of the G phase:

$$\begin{aligned} G[0, 2] &= G[0, 1]P[1, 2] \oplus G[1, 2] \\ G[0, 4] &= G[0, 2]P[2, 4] \oplus G[2, 4] \\ G[0, 8] &= G[0, 4]P[4, 8] \oplus G[4, 8]. \end{aligned} \quad (11.16)$$

In the C phase, we complete the calculation of all carry-in values for the addition operation. Recall that the carry-in for bit j is $G[0, j]$.

Next, we uncompute the P values by reversing the operations from the P phase. We call this the P^{-1} phase. After this phase, qubit $|b_i\rangle$ again holds the value $P[i, i+1] = a_i b_i$ that was computed in the INIT phase. For each sum qubit $|s_i\rangle$, we compute $s_i = c_i \oplus a_i b_i$ with a CNOT gate. For bit 0, there is no c_0 , so we have only computed $s_0 = b_0$; therefore, we need to XOR with a_0 to compute the sum. Finally, we uncompute $P[i, i+1]$ to restore the original $|b_i\rangle$ values.

11.3.4 Adder Comparison

The differences in the adder designs illustrate basic tradeoffs in reversible logic design. The optimized ripple-carry adder performs addition in-place (replacing one of the inputs) and only requires one extra ancilla qubit, but the depth increases linearly with the number of bits in the input data. The carry-lookahead adder uses a logarithmic structure to significantly reduce the depth, scaling as $O(\log n)$ for an n -bit adder. This adder, however, requires $n + 1$ additional qubits for the sum output, as well as roughly $n - \log n$ ancilla qubits. (There is an in-place version of the adder, but it requires twice the depth and about the same total number of qubits as this version.)

Table 11.2 shows how the number of qubits and the circuit depth grow for each circuit as the input size increases. In this table, only Toffoli gates are counted and each has a depth of one; this is reasonable for comparison, since the cost is dominated by Toffoli gates, especially when expanded into two-qubit gates (which we will explore below). If you think a 1024-bit adder is unreasonable, keep in mind that Shor's factoring algorithm requires operations on large numbers with 1000's of bits.

Table 11.3 gives more insight into what happens when we try to map these algorithms onto an actual superconducting quantum system. The table shows the depth and number of qubits for an 8-bit adder that generates a 9-bit sum. In the ideal case, where any qubit can interact with any other qubit, the in-place ripple-carry adder uses the fewest qubits, but the CLA has the lowest depth.

The second row in the table shows the depth of the circuit when implemented using Toffoli gates. However, current quantum systems do not implement Toffoli or other three-qubit gates directly. They translate the Toffoli gate into a collection of one- and two-qubit gates, as shown in Figure 11.17. The third row shows the depth when that

Table 11.2 Comparison of in-place ripple-carry and carry-lookahead adder circuits. For this table, depth only counts Toffoli gates.

Input size (bits)	Depth		Qubits	
	Ripple	CLA	Ripple	CLA
8	15	8	18	29
64	127	14	130	250
256	511	18	515	1016
1024	2047	22	2050	4086

Table 11.3 Comparison of adder circuits for 8-bit binary numbers. The bottom two rows show the results of compiling the circuit to a specific quantum computer topology. *Sparse* is the 53-qubit IBM Rochester topology, and *Full mesh* is Google Sycamore topology, both shown in Figure 11.18.

	Ripple	In-place ripple	CLA
Qubits	24	18	28
Depth: Toffoli gates	37	20	11
Depth: 2-qubit gates	277	141	83
Depth: Sparse	433	213	204
Depth: Full mesh	456	200	249

transformation is performed by the IBM Qiskit transpiler [2]. The CLA has an advantage, even with only eight bits, and its depth grows as $O(\log n)$, compared to $O(n)$ for the ripple-carry adders.

However, the CLA uses Toffoli gates that require interaction between qubits that are far away from one another. Compare the Toffoli gates in Figure 11.30 to those used in Figure 11.32. With the interleaving of input qubits, the ripple-carry adder uses gates that involve nearby qubits. In the CLA circuit, especially in the G and C stages, Toffoli gates span distances that grow by a factor of 2 in each level. Depending on the system topology, this can require many swap gates to move qubit values close enough to interact, as described in Section 11.2.3. Each swap gate increases the depth of the circuit.

To see the impact of swap gates, we use the IBM Qiskit transpiler to map the circuit to actual machine topologies. For the 8-bit adder, we need at most 29 qubits, so we use the two larger topologies from Figure 11.18. The row labeled *Sparse* is IBM’s Rochester machine with interconnecting rings, and the row labeled *Full mesh* is Google’s Sycamore. For this experiment, the transpiler assumes that all physical qubits and gate operations are error-free.

There are two interesting observations from the data. First, the CLA circuit loses all of its depth advantage compared to in-place ripple-carry. The extra SWAP gates needed to move data around introduces significant overhead. We expect this overhead to increase with adder size, since qubits are further apart, but we also expect that the logarithmic scaling of the CLA would start to dominate with larger circuits.

Second, there is not much difference between the two topologies, even though the mesh has a larger number of connections. In fact, it performs worse for the CLA, when more qubit movement is required. There could be mesh-specific optimizations that the IBM transpiler does not consider, since the IBM machines do not use a mesh topology. The choice of topology has other implications for the system, such as the susceptibility to crosstalk errors between qubits. It is clear that the choice of topology and the mapping of circuits to a physical topology are important considerations in the performance of applications on physical systems.

11.4 Phase Logic

So far, we have only considered logic that operates in the computational basis, the $|0\rangle$ and $|1\rangle$ states that are most familiar to us from Boolean logic. In this section, we consider operations that affect the phase of the quantum state.

Phase adds a quantum twist to the topic of computational logic. It allows qubit states to constructively and destructively interfere with one another, which is an extremely useful tool in the quantum algorithms discussed in Chapter 12. In the context of this chapter, we will be using phase to “mark” certain basis states as interesting. Later, we will see how differences in phase can be converted to differences in amplitude.

11.4.1 Controlled- Z and Controlled-Phase Gates

As a single-qubit gate, recall that the Z performs a rotation of π around the z -axis of the Bloch sphere. Since $e^{i\pi} = -1$, this introduces a factor of -1 , but only for the $|1\rangle$ component of the state in the standard basis.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{aligned} Z|0\rangle &= |0\rangle \\ Z|1\rangle &= -|1\rangle \\ Z|\psi\rangle &= \alpha|0\rangle - \beta|1\rangle. \end{aligned} \quad (11.17)$$

Also recall that we have S and T gates that perform rotations of $\pi/2$ and $\pi/4$, respectively. We can also consider a generalized *phase* gate, often called P , that performs an arbitrary rotation of λ .

$$P(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix} \quad \begin{aligned} P(\lambda)|0\rangle &= |0\rangle \\ P(\lambda)|1\rangle &= e^{i\lambda}|1\rangle \\ P(\lambda)|\psi\rangle &= \alpha|0\rangle + e^{i\lambda}\beta|1\rangle. \end{aligned} \quad (11.18)$$

Now consider the *controlled* version of the Z , called CZ . This is a two-qubit gate, with a control and a target. If the control qubit is $|1\rangle$, the Z operator is applied to the target qubit. The truth table for this operation is shown in Table 11.4, where q_0 is the control bit and q_1 is the target.

Notice that the CZ gate has no effect unless *both* qubits are $|1\rangle$. Also, as shown in the last line, it makes no difference whether we interpret the phase shift as occurring on the target or the control: the affect is the same. There really is no distinction between the control and target, in this case. For this reason, the CZ gate is often drawn as two connected dots, rather than a control bit and a Z gate, emphasizing the symmetric nature of the gate, as shown in Figure 11.33.

The controlled-phase (CP) gate is similar. When both qubits are $|1\rangle$, the rotation of λ is applied.

You may be wondering why this phase rotation even matters, since it appears that the CZ imposes a global phase, and we know that $|11\rangle$ and $-|11\rangle$ are indistinguishable. However, when a state is in superposition, *only* the basis states where both qubits are one will be affected, introducing a *relative* phase shift.

Table 11.4 Controlled-Z (CZ) gate, where q_0 is the control and q_1 is the target.

q_1	q_0	$CZ(q_0, q_1)$
$ 0\rangle$	$ 0\rangle$	$ 00\rangle$
$ 0\rangle$	$ 1\rangle$	$ 01\rangle$
$ 1\rangle$	$ 0\rangle$	$ 10\rangle$
$ 1\rangle$	$ 1\rangle$	$ (-1)1\rangle = 1(-1)\rangle = - 11\rangle$

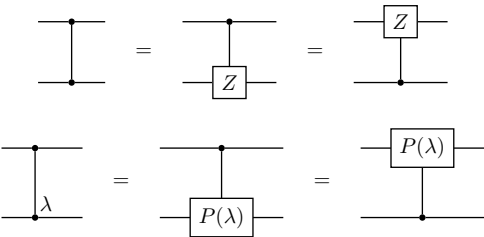


Figure 11.33 Controlled-Z (CZ) and controlled-phase (CP) gates.

For example, consider a two-qubit state $|\psi\rangle$ that is in equal superposition:

$$|\psi\rangle = H^{\otimes 2} |00\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \tag{11.19}$$

Applying CZ to $|\psi\rangle$ will flip the phase of only the $|11\rangle$ basis state, resulting in a relative phase difference:

$$CZ(|\psi\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle). \tag{11.20}$$

See Figure 11.34 for an illustration of this circuit and the output state. This figure also introduces a new *disc notation* to visualize a quantum state, adopted from Johnston *et al.* [128]. Each standard basis state is represented by a disc. The magnitude is represented by a shaded circle whose radius is the probability amplitude. The area of the shaded circle is proportional to the square of the radius, so this is a visual indication of the probability of measuring that state. (An “empty” disc means the basis state has a magnitude of zero.) The phase of each basis state is represented by a line that rotates counter-clockwise; straight up means zero phase. This notation is convenient for quickly visualizing the magnitude and phase of each component of a multi-qubit state, and is often more helpful than the Bloch sphere visualization, especially for entangled states. However, the number of discs scales as 2^n for an n -qubit state, so it is most useful when n is not too large. A similar notation is used by the Quirk quantum circuit simulator [129].

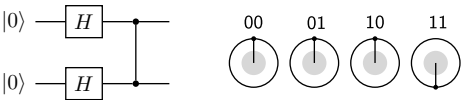


Figure 11.34 CZ gate acting on an equal superposition of basis states. Only the state where both qubits are one is flipped.

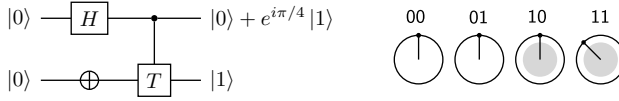


Figure 11.35 Phase kickback using controlled- T gate. The target qubit is apparently unaffected, while the phase rotation occurs on the control qubit.

The symmetric nature of the controlled-phase gates leads to a phenomenon called *phase kickback*, where applying a phase rotation to one qubit causes a phase change on one or more other qubits.

Consider the circuit shown in Figure 11.35, where both qubits are initialized as $|0\rangle$. The state before the CP is $|1+\rangle$. Even though the phase change is shown on the target, the effect is to leave the target qubit alone and to apply the $\pi/4$ phase shift to the control qubit.

$$\begin{aligned}
 \text{CP}(\pi/4) |1+\rangle &= \text{CP}(\pi/4) \left(\frac{1}{\sqrt{2}} |10\rangle + \frac{1}{\sqrt{2}} |11\rangle \right) \\
 &= \frac{1}{\sqrt{2}} (|10\rangle + e^{i\pi/4} |11\rangle) \\
 &= |1\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle + e^{i\pi/4} |1\rangle).
 \end{aligned} \tag{11.21}$$

11.4.2 Selective Phase Change

The CZ and CP gates allow us to change the phase of a specific basis state, but what if we want to be more general, to change the phase of any basis states that satisfy a particular logical condition?

Consider an n -qubit state $|A\rangle$, which is a superposition of basis states:

$$\begin{aligned}
 |A\rangle &= a_0 |0 \dots 00\rangle + a_1 |0 \dots 01\rangle + \dots + a_{n-1} |1 \dots 11\rangle \\
 &= a_0 |0\rangle + a_1 |1\rangle + \dots + a_{n-1} |n-1\rangle \\
 &= \sum_{x=0}^{n-1} a_x |x\rangle.
 \end{aligned} \tag{11.22}$$

Now consider a predicate function $S(x)$ that returns either 1 (true) or false (0) for any n -bit integer value x , and the set of integers G that contains all of the integers for which $S(x) = 1$.

$$G = \{x \in \mathbb{Z} \mid 0 \leq x \leq 2^{n-1} \text{ and } S(x) = 1\}. \tag{11.23}$$

We can write $|A\rangle$ as a combination of basis states that satisfy the predicate S and those that don't. We'll use $|G\rangle$ to represent the "good" basis states that satisfy S and $|B\rangle$ to represent the "bad" basis states that don't.

$$|A\rangle = \sum_{x \notin G} a_x |x\rangle + \sum_{x \in G} a_x |x\rangle = |B\rangle + |G\rangle. \tag{11.24}$$

Now consider a unitary function U_S that takes an n -qubit integer state $|x\rangle$ and a single ancilla qubit, and calculates $S(x)$ on the ancilla. In other words, $U_S |0\rangle |x\rangle = |S(x)\rangle |x\rangle$.

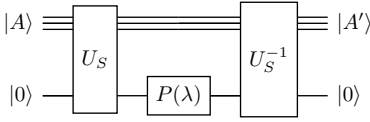


Figure 11.36 Selective phase rotation of basis states in $|A\rangle$ for which the predicate S evaluates as true. U_S is a unitary function that calculates the predicate S using an ancilla qubit.

We put the ancilla bit as the most significant, because we want to draw it at the bottom of the circuit, as shown in Figure 11.36.

What happens if we put our superposition state $|A\rangle$ as the input to the circuit?

$$U_S |0\rangle |A\rangle = U_S |0\rangle (|B\rangle + |G\rangle) = |0\rangle |B\rangle + |1\rangle |G\rangle. \quad (11.25)$$

In general, the output state is entangled. Measuring the integer part of the output state will completely determine the value of the ancilla qubit. Likewise, if we measure the ancilla bit, the integer part will be projected to either $|G\rangle$ (if 1) or $|B\rangle$ (if 0).

Next, we apply a phase gate $P(\lambda)$ to the ancilla qubit. This rotates the part of the entangled state for which the ancilla bit is $|1\rangle$:

$$P(\lambda)U_S |0\rangle |A\rangle = |0\rangle |B\rangle + e^{i\lambda} |1\rangle |G\rangle. \quad (11.26)$$

Note that the phase change applies to the entire multi-qubit state; we can either associate it with the most significant qubit $|1\rangle$ or the rest of the state $|G\rangle$.

Finally, if we uncompute the predicate, we restore the ancilla qubit to $|0\rangle$, but the phase change on $|G\rangle$ remains, because the predicate logic operates on computational basis states without affecting phase.

$$\begin{aligned} U_S^{-1}(|0\rangle |B\rangle + e^{i\lambda} |1\rangle |G\rangle) &= |0\rangle |B\rangle + e^{i\lambda} |0\rangle |G\rangle \\ &= |0\rangle \left(\sum_{x \notin G} a_x |x\rangle + e^{i\lambda} \sum_{x \in G} a_x |x\rangle \right). \end{aligned} \quad (11.27)$$

If we ignore the ancilla bit, we have created a new state $|A'\rangle$, which has the same amplitudes and basis states of the original $|A\rangle$ state with a phase rotation of λ applied to all basis states that satisfy the predicate S .

In the special case of a π -rotation, the circuit can be even simpler, avoiding the need for uncomputing, as shown in Figure 11.37. To understand this circuit, first consider the simplified circuit in Figure 11.38, where $|\psi\rangle$ is a single-qubit state. On the left-hand side, phase kickback will flip the phase of the $|1\rangle$ component of $|\psi\rangle$. On the right-hand side, we use the relationship $Z = HXH$ to redraw the circuit using a CNOT instead of a CZ.

Now, consider U_S to be a sort of “generalized CNOT,” which flips the target (ancilla) qubit whenever the control $|A\rangle$ satisfies the predicate. Then the same phase kickback

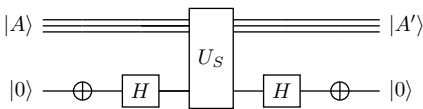


Figure 11.37 Selective phase flip (π -rotation) of basis states in $|A\rangle$ for which the predicate S evaluates as true. U_S is a unitary function that calculates the predicate S using an ancilla qubit.

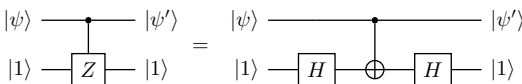


Figure 11.38 Phase kickback using CZ and CNOT gates.

will apply the phase flip to the basis states of $|A\rangle$ that cause the ancilla bit to flip. This is exactly what we want, and leads to the circuit in Figure 11.37. The uncompute part is not needed, because the ancilla bit is returned to $|1\rangle$ after the second Hadamard gate, and it is not entangled with the output state $|A'\rangle$.

11.4.3 Phase Logic Gates

In some cases, we can implement logical phase rotations directly, without the use of ancilla qubits. Johnston *et al.* [128] illustrate a number of fundamental logic operations that encode the result in the phase of the output state, shown in Figure 11.39. The inputs to these circuits are interpreted as logical values encoded in the computational basis. The output amplitudes are unchanged, but the phase rotation has been applied to the basis states according to the desired logical operation.

Since these circuits do not operate on phase-encoded states, they cannot be used together to create complex logical operations. Instead, the phase logic replaces the last stage of a more standard logic circuit, using Toffoli and CNOT gates, as described in Section 11.2. Then the first stages must be uncomputed to restore the amplitudes to their original states, while keeping the phase rotations intact.

To perform a general logical phase rotation using n -stage combinational logic:

1. Perform the first $n - 1$ stages using conventional logic.
2. Substitute the last gate with one of the circuits in Figure 11.39.
3. Uncompute the first $n - 1$ stages.

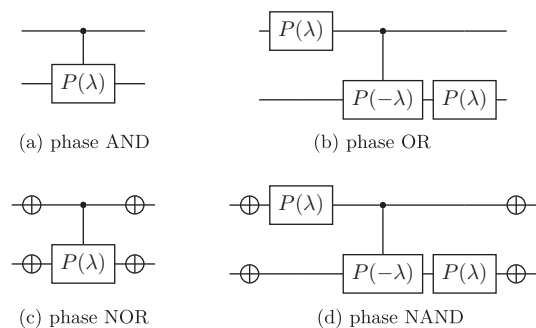
For a concrete example, adopted from [128], consider the following logic operation:

$$(a \text{ OR } \text{NOT } b) \text{ AND } c$$

We first perform the OR using a Toffoli gate and an ancilla qubit, as shown in Figure 11.40. Then we apply a phase-AND with c and the ancilla qubit. Finally, we uncompute the OR. If the input state is an equal superposition, the resulting output state is shown in the figure: the phases of all basis states for which $(a + \bar{b}) \cdot c$ is true have been flipped.

With the exception of phase-AND and phase-NOR, these phase logic circuits do not easily extend to more than two qubits. For phase-AND, the CP gate can easily be extended to a multi-controlled-phase (MCP) or multi-controlled-Z (MCZ) gate. However, while these gates look simple, remember that they must be implemented in terms of two-qubit gates on actual hardware.

Figure 11.39 Phase logic gates: performs phase rotation of λ on computational basis states that satisfy the Boolean operator.



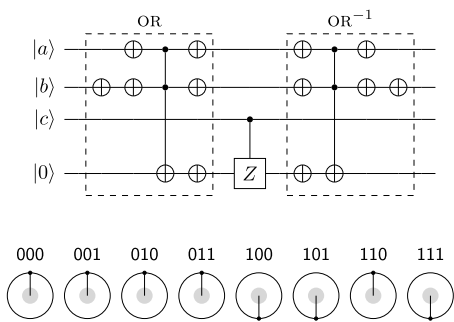


Figure 11.40 Flipping the phase when $(a + b) \cdot c$ is true. If the input starts as an equal superposition, the output state is shown below the circuit.

11.5 Summary and Further Reading

Manipulating quantum data in the computational basis is an important building block for quantum algorithms. As we’ve shown in this chapter, any classical reversible logic can be implemented using quantum gates. But quantum gates are much slower and (currently) more error-prone than classical logic gates, so why should we bother? Shouldn’t we do classical operations on classical hardware and leave the quantum computer to do the hard stuff?

One source of quantum speedup is through quantum parallelism: the effect of logical operations on multiple basis states in a superposition. In other words, a unitary operator that converts $|x, 0\rangle$ to $|x, f(x)\rangle$ will perform that transformation on all components of $|x\rangle$:

$$\sum_i |x_i, f(x_i)\rangle. \tag{11.28}$$

This is different than classical parallelism, in that we do not get to see all of those values of $f(x_i)$. But we know that they are there, encoded in the quantum state. We can extract information about the values using the quantum building blocks described in Chapter 12. For example, the Quantum Fourier Transform (QFT) can reveal information about the sequence $f(0), f(1), f(2), \dots$. Shor’s algorithm combines QFT with modular exponentiation (a classical function) to help find prime factors of a large integer. As another example, if $f(x)$ is a Boolean function, a technique known as Amplitude Amplification can make it easier to discover a value of x that makes $f(x) = \text{true}$.

Both phase logic and regular logic are indispensable components of these algorithms. Ironically, because of the current limitations on the number of qubits and the number of operations that can be performed before decoherence, it is often the classical computations that are the most challenging to perform efficiently. Automated optimization of large quantum circuits is an active area of research, with approaches adopted from classical and reversible logic design and synthesis [130, 131], compiler techniques [132], machine learning and other optimized searching [133, 134], and others [135].

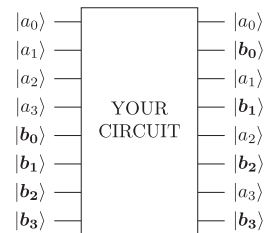
Of course, the circuits discussed in this chapter assume error-free logical qubits. The error correction techniques from Chapter 10 will require orders of magnitude more physical qubits. Gidney and Ekerå [110] give a detailed resource estimate for factoring large integers using Shor’s algorithm. While some of the details rely on understanding

information presented in Chapter 12, a large part of the discussion revolves around efficient adders and multipliers and their implementation on the surface code.

11.6 Exercises

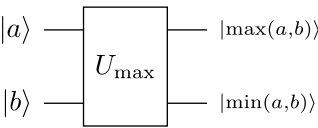
- 11.1** Without using a Toffoli gate, draw an implementation of a reversible classical OR gate. Also draw the inverse OR^{-1} gate, or show that the gate is its own inverse.
- 11.2** Prove that the SWAP gate can be implemented using three CNOT gates, as shown in Figure 11.20. Also show the unitary matrix representation of the operation performed by the SWAP gate.
- 11.3** Using big- O notation, what is the depth of an n -bit quantum ripple-carry adder (Section 11.3.1) if we don't uncompute the carry bits?
- 11.4** Use swap gates to interleave the qubits of two four-qubit values, as shown in Figure 11.41. Assume a linear qubit topology: only qubits that are neighbors and can swap with each other.

Figure 11.41 Interleave function for Exercise 11.4.



- 11.5** Design a quantum circuit to compute the absolute value of a number. Given a four-qubit value $|x\rangle$, produce the four-qubit value $|y\rangle$, where x and y are 2's complement binary integers and $y = |x|$. You may use Toffoli gates and any one- or two-qubit gates. You may need additional qubits to make this reversible, and you may add ancilla qubits, but minimize the total number of qubits in your solution.
- 11.6** Given two three-qubit states $|a\rangle$ and $|b\rangle$, each representing a three-bit unsigned integer, we want a quantum circuit that outputs the larger value on the top and the smaller value on the bottom. See Figure 11.42.
- (a)** The function as drawn is not reversible. Explain why, and describe how adding another qubit to the input and output states can make a reversible function.
- (b)** Design a quantum circuit to implement the reversible function. Ancilla qubits can be added, if needed, as long as they are uncomputed.

Figure 11.42 Max function for Exercise 11.6.



11.7 Given two four-qubit states $|a\rangle$ and $|b\rangle$, each representing a four-bit unsigned integer, design a quantum circuit that flips the phase of $|b\rangle$ if the two integers are equal.

12

Some Quantum Algorithms

In this final chapter, we present a few selected quantum algorithms that illustrate the promise of quantum computers to provide a computational advantage over classical computers. We use the word “advantage” informally here, simply meaning that there is some problem size at which the quantum computer can give a solution faster than any feasible classical computer.

What do we mean by “faster”? In the next section, we briefly discuss computational complexity and how it applies to quantum computing. Then we present the two most famous quantum algorithms which promise quantum advantage: Grover’s search algorithm (12.2) and Shor’s algorithm for factoring (12.5). In the process, we will introduce several fundamental building blocks, including amplitude amplification (12.2), the Quantum Fourier Transform (12.3), and quantum phase estimation (12.4). Finally, we will discuss a class of hybrid classical-quantum algorithms, known as *variational algorithms*, that are designed to operate within the realm of noisy, intermediate-scale quantum (NISQ) systems that are likely to be prevalent in the early years of quantum applications.

12.1 Computational Complexity

In classical computing, we analyze algorithms to determine the resources needed to solve a computational problem. A resource typically means *time* or *space*, but could be energy, circuit gates, or some other measurable quantity. We are especially interested in the algorithm’s behavior as the problem size increases. We use “big-O” notation to abstract away details, concentrating on upper-bound execution time as problem size (n) increases. When we say that the run-time $T(n)$ for a problem size n is $O(g(n))$, we mean that there are positive numbers M and n_0 such that

$$T(n) \leq Mg(n), \text{ for all } n \geq n_0. \quad (12.1)$$

Using this approach, we can state that using the selection sort algorithm on a collection of n items requires $O(n^2)$ time (computational steps), while the quicksort algorithm requires $O(n \log n)$ time. We can’t say whether quicksort will be faster for any particular size n , because $O(\cdot)$ does not explicitly include constant factors or other terms that are dominated by the highest-power terms. However, we can say with confidence that large sorting problems are likely to run much faster using quicksort.

12.1.1 Quantum Program Run-Time

The execution time of a quantum algorithm is estimated by the *depth* of the circuit: the longest sequence of one- and two-qubit gates between the initial state preparation and the final measurement. Even though different gate operations may take different amounts of time, we typically simplify and assume that all one- and two-qubit gates execute in unit time. Some estimates focus on the number of Toffoli gates, since they require multiple simple gates and are very prevalent in quantum logic (Chapter 11). Other approaches focus on the number of sequential T gates, since they require costly magic state distillation for error-corrected computation (Chapter 10).

For the purposes of understanding the run-time complexity of an algorithm, a number of assumptions are made. First, the *width* of the circuit, the number or (logical) qubits required, is generally unbounded; we can use as many qubits as we need, including helper (ancilla) qubits. (The number of qubits, however, may be considered as an additional measure of scalability. There is often a tradeoff between circuit depth (time) and width.) Second, the qubits are usually assumed to be *completely connected*—any qubit can interact with any other qubit, though we may favor algorithms for which nearby connections are sufficient. Finally, we assume *fault-tolerant* computation, using error-free logical qubits with long decoherence times.

Some algorithms utilize *oracles*, which are “black-box” functions that return a specified result. The details of the oracle are not known to the algorithm, and therefore cannot be included in estimates of computational complexity. In these cases, we normally refer to *query complexity* as the number of times the oracle must be queried for a problem size n . There is an implicit assumption that the execution of the oracle is efficient; otherwise, the cost of running the algorithm will be dominated by the cost of computing the oracle function.

12.1.2 Classical Complexity Classes

We use *complexity classes* to group together problems in terms of the resources required to solve them. The best-known classification scheme deals with *decision* problems, those that provide a single yes/no answer. Many computational problems can be transformed into decision problems, so this choice does not limit the generality of the approach. Other problems types have their own complexity class hierarchies. For our purposes, we will be casual about this difference, and we will not distinguish between different types of computational problems. Much of the discussion below is based on Arora and Barak [136].

P is the class of decision problems that run in polynomial time. In other words, the execution time of a problem with input size n is $O(n^k)$ for some positive integer k . Most of the fundamental algorithms we learn early in our computing experience are in **P**: computing the product of two integers, finding a particular value in an array, sorting an array of values, etc.

NP is the class of decision problems for which a *solution can be verified* in polynomial time. The cost of finding a solution may be more than polynomial. For example, finding the prime factors of x is difficult, and the time grows exponentially with the number of bits needed to represent x , but it is easy to verify whether p is a prime factor of x , simply by dividing x by p .

NP does not stand for “nonpolynomial,” as you might expect, but for *nondeterministic polynomial*, because an alternate definition stems from problems that can be solved in polynomial time using a nondeterministic Turing machine. In fact, it is clear that **NP** contains **P** ($\mathbf{P} \subseteq \mathbf{NP}$), since any solution that can be computed in polynomial time can certainly be verified in polynomial time.

On the other hand, there are problems that are known to be in **NP** that are *not* known to be in **P**—that is, there is no known polynomial-time solution. One example is the *satisfiability problem*, known as SAT: Given a Boolean formula with n binary variables, is there an assignment of variables (to 0 or 1) for which the formula is true (satisfied)? The fastest-known algorithm for 3SAT, in which each part of the formula can have up to three variables, is reported as $O(1.30331^n)$, where n is the number of variables in the formula [137]. However, it is very easy to plug in a solution of n variables and verify that the formula is true. In the worst case, an **NP** problem can be solved in exponential time, since a “brute force” check of an n -bit input requires 2^n polynomial verifications.

A problem is known as **NP-hard** if any problem in **NP** is *reducible* to this problem. By reducible, we mean that there is a polynomial-time function that can transform an instance of one problem to an instance of the other. If a problem is **NP-hard** and is also in **NP**, it is said to be **NP-complete**. In some sense, the set of **NP-complete** problems represents the “hardest” problems in **NP**. Thousands of problems have been identified to be **NP-complete**.

The significance of **NP-complete** problems is that *every* **NP** problem can be provably reduced to an **NP-complete** problem. Therefore, if someone finds a polynomial-time algorithm to solve an **NP-complete** problem, it means that *all* problems in **NP** can be solved in polynomial time, which means $\mathbf{P} = \mathbf{NP}$. While it is strongly conjectured that $\mathbf{P} \neq \mathbf{NP}$, it has never been proven, and this is one of the most important outstanding problems in computational computer science.

Before we discuss how quantum processors fit in, we introduce one more classical complexity class. **BPP** is the set of decision problems that can be solved in polynomial time on a *probabilistic* Turing machine (one that can make a random decision at each step) with a *bounded probability* of error. This allows randomness to be incorporated into algorithms. The bounded error usually means that the probability of a correct answer is $\geq 2/3$, though the definition allows for any value greater than one-half.

It should be clear that $\mathbf{P} \subseteq \mathbf{BPP}$, since the random choice could be the same as the deterministic choice. It is an open question whether $\mathbf{P} = \mathbf{BPP}$, or whether there is a way to convert *every* probabilistic algorithm to a deterministic algorithm. Many theorists believe this is true, but it has not been proven. We do not know if $\mathbf{BPP} \subseteq \mathbf{NP}$.

12.1.3 Quantum Complexity

For analyzing problems and algorithms in the quantum domain, Deutsch [138] developed the quantum Turing machine in 1985. This was refined later [139, 140], and Yao [141] formalized the *quantum circuit model* in 1993, establishing that the run-time of a quantum program is based on the depth of its circuit, as described above.

BQP is the class of decision problems that run in polynomial time on a quantum circuit, with bounded error. Because of the probabilistic nature of quantum measurements, quantum programs are more like **BPP** than **P**. In Chapter 11, we showed that

any classical computation can be done efficiently on a quantum computer, so it should be clear that $\mathbf{BPP} \subseteq \mathbf{BQP}$.

What do we know about the relationship between $\mathbf{BQP}$ and $\mathbf{NP}$? Not a lot. There's a popular misconception that quantum computers can solve any exponential problem in polynomial time, but that's not true. There are some $\mathbf{NP}$ problems that are in $\mathbf{BQP}$ —for example, integer factoring and discrete logarithm are solved by Shor's algorithm (Section 12.5). However, there are other $\mathbf{NP}$ (and particularly $\mathbf{NP}$ -complete) algorithms for which there are no known polynomial quantum algorithms. Therefore, there is a strong conjecture that $\mathbf{NP} \not\subseteq \mathbf{BQP}$, but there is no proof. Watrous [142] provides more information about quantum complexity classes.

There is an understanding that problems must have a certain kind of *structure* that can be exploited by quantum algorithms in order to get large speedups. This structure can be exploited by algorithmic building blocks such as amplitude amplification (Section 12.2) and the Quantum Fourier Transform (Section 12.3), which are described in this chapter. For some problems, it is believed that quantum computing will provide no speedup at all. Aaronson [143] provides a discussion of how complexity theory informs our understanding about what is possible for quantum computers.

12.2 Grover's Search Algorithm

Grover's algorithm, published in 1996 [144], identifies a single n -bit value x that satisfies a yes/no predicate. In other words, it finds the value x for which $P(x) = 1$. The predicate is calculated by an oracle, a black-box subroutine for which the algorithm has no information. A classical algorithm must evaluate $P(x)$ for every value $0 \leq x < N$ until it finds the correct answer, where $N = 2^n$. On average, this will require $N/2$ trials. Grover's algorithm, however, will find the value x using only $O(\sqrt{N})$ trials, which has been shown to be optimal for any quantum algorithm [145].

The oracle requirement means that there is no information within the value x or any known *structure* to the search space that would yield an improved classical search. For example, binary search would take $O(\log N)$ evaluations, but requires that the predicate be related to the order of values.

It's a simple matter to create a superposition of all x values, using Hadamard gates, and then to compute $P(x)$ on the superposition. However, a measurement will yield only one output state, $|x_i, P(x_i)\rangle$, and there would only be a $1/N$ chance that $x_i = x$. Therefore, this random sampling approach does not give an advantage over the classical approach: it would require $O(N)$ trials to find a likely match. (In fact, this would be worse than the classical approach, which could try each value sequentially. The random sampling may take more than N trials and in fact is never guaranteed to find the correct result.)

In order to exploit the quantum parallelism enabled by superposition, some other operation is needed to increase the likelihood of measuring the desired state.

12.2.1 Grover Iteration

Starting with an equal superposition of all values, Grover's algorithm iterates the following two steps:

Figure 12.1 Grover search initialization. All states are in equal superposition. The correct answer ($x = 4$) is shown in black.

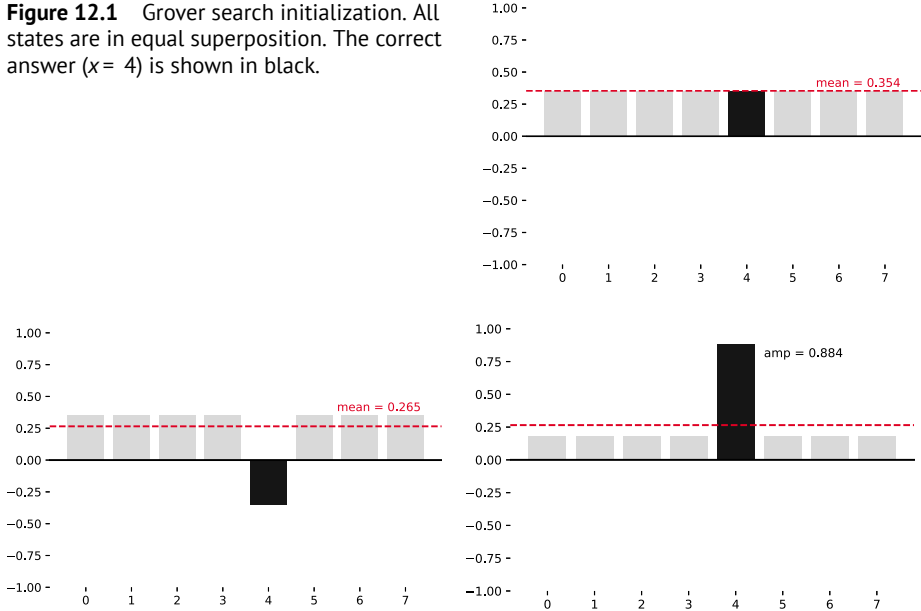


Figure 12.2 First iteration of Grover search. On the left, the oracle changes the sign of the correct answer, reducing the mean amplitude. On the right, all amplitudes are inverted around the mean, increasing the amplitude of the correct answer.

1. Flip the phase of the state $|x_p\rangle$ for which the predicate is true. In other words, if the amplitude of $|x_p\rangle$ is a_p , change it to $-a_p$.
2. For each element of the superposition, change amplitude a_i to $2A - a_i$, where A is the mean amplitude.

The process is illustrated in Figures 12.1 and 12.2. In this example, there are eight candidate values (0–7), encoded in three qubits. The qubits are initialized in an equal superposition, each with amplitude $1/\sqrt{8} = 0.354$. For the purpose of this example, we know that $x = 4$ is the desired solution, and we show that state in a darker color. If we measure the state at this point, the probability of measuring 4 is $1/8$, no better than a random guess.

Figure 12.2 shows the first search iteration. The first step flips the phase of the state x where $P(x) = 1$. This also has the effect of lowering the mean amplitude ($A = 0.265$), since this state now contributes a negative value. Each non-marked (positive) state's amplitude is now greater than A , so the second step reduces these amplitudes to $2A - 0.354 = 0.176$, while the marked (negative) state is significantly increased to $2A + 0.354 = 0.884$.

If we measure the state after this iteration, the probability of measuring 4 is $(0.884)^2 = 0.781$. Performing a second iteration (Figure 12.3) again increases the amplitude of the correct state and raises its probability to $(0.972)^2 = 0.945$.

However, notice that the amplitudes of the incorrect answers have become negative. This is enough to make the mean amplitude negative as well. If we perform a third iteration (Figure 12.4), the amplitude of the correct answer decreases, reducing the probability that it will be measured. The optimal number of iterations, k , depends on the number of qubits involved in the search:

$$k = \left\lceil \frac{\pi}{4} \sqrt{N} \right\rceil \quad (12.2)$$

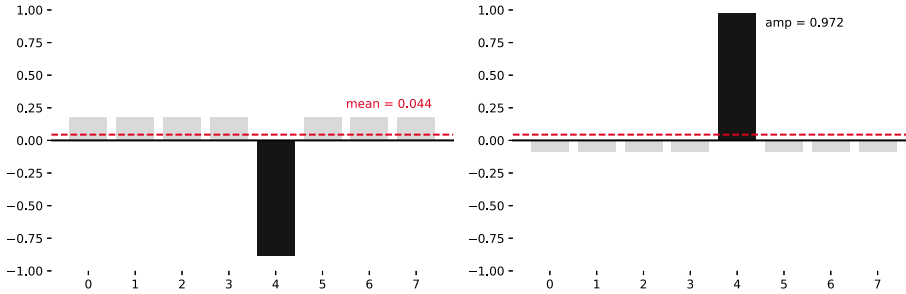


Figure 12.3 Second iteration of Grover search, further increasing the amplitude of the correct answer.

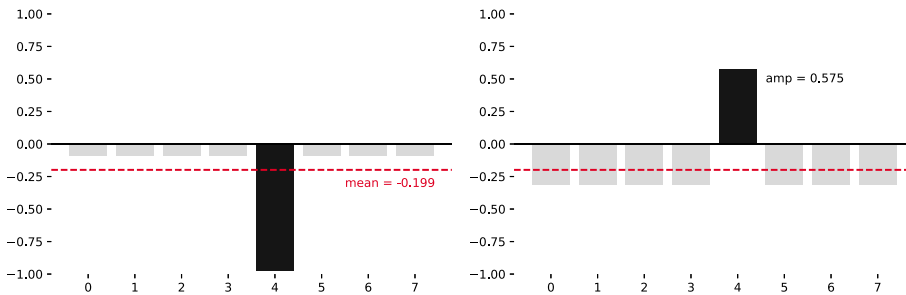


Figure 12.4 Third iteration of Grover search. In this case, the amplitude of the correct answer is reduced from its previous value.

where $N = 2^n$ and n is the number of qubits.¹ In our example, $N = 8$ and $\lfloor (\pi/4)\sqrt{8} \rfloor = 2$, which is exactly what we discovered.

The amplitude of the correct answer will decrease with additional iterations, and then will start to increase again, exhibiting a periodic behavior. With $N = 8$, the amplitude after six iterations is very close to -1.0 , yielding essentially a 100% probability of measuring the correct answer. However, each iteration increases the depth of the quantum circuit, which increases both the time and the chance for error. Just repeating the entire computation (with two iterations) gives the correct answer with a 99.7% probability, and only requires a total of four iterations.

12.2.2 Quantum Implementation

A quantum circuit for Grover search with one iteration is shown in Figure 12.5.

We first need to create an equal superposition. This is easily done by initializing n qubits to $|0\rangle$ and using n Hadamard gates.

The next step is to flip the phase of the basis state for which the predicate is true. This is an instance of *phase logic*, described in Section 11.4. First, we create a unitary U_P that sets an ancilla qubit to 1 when $P(x) = -1$. If we initialize that ancilla qubit to $|-\rangle$, phase kickback will change the phase of the desired basis state. Since the ancilla qubit state does not change, we initialize it at the beginning and leave it in the $|-\rangle$ state through all of the iterations.

The cost of this first step depends on the cost of computing U_P . If the cost of computing U_P is too high (as a function of n), then it may dominate the computation time and essentially eliminate the $\sqrt{N}$ advantage of Grover's algorithm.

¹ The *floor* operator $\lfloor x \rfloor$ returns the largest integer that is less than or equal to x .

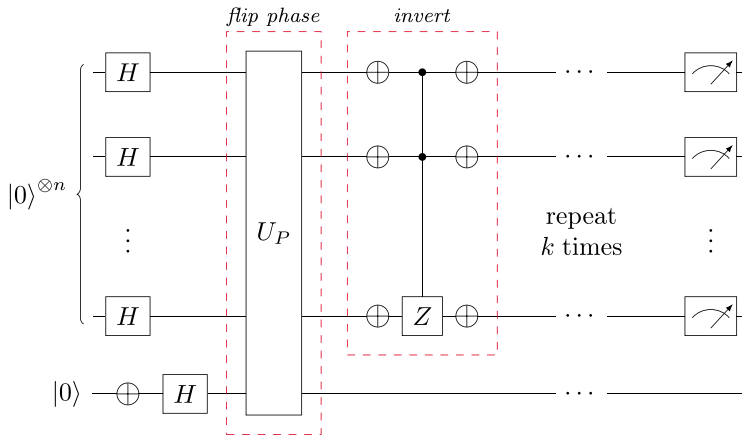


Figure 12.5 Grover search algorithm for n qubits. Boxes *flip phase* and *invert* form one iteration, which is repeated $k = (\pi/4)\sqrt{N}$ times, where $N = 2^n$.

The second step seems more challenging. How do we compute the mean amplitude, and how do we invert all of the amplitudes around the mean? The circuit, shown in the *invert* box in Figure 12.5, is quite simple, so our task is to convince you that the circuit performs the desired computation.

First, the transformation we want is given by the following unitary matrix, remembering that $N = 2^n$:

$$D = \begin{bmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{bmatrix}. \quad (12.3)$$

To convince yourself this is correct, take a column vector with amplitudes a_i and multiply times the j -th row of the matrix to produce the new amplitude a'_j :

$$\begin{aligned} a'_j &= \left(\frac{2}{N} \sum_{i \neq j} a_i \right) + \left(\frac{2}{N} - 1 \right) a_j \\ &= \left(\frac{2}{N} \sum_i a_i \right) - a_j \\ &= 2A - a_j. \end{aligned} \quad (12.4)$$

Now consider the middle part of the *invert* circuit, the controlled- Z surrounded by X gates. Based on our understanding of phase logic, this will flip the phase of $|000 \dots 00\rangle$ and leave the other basis states alone. We'll call this n -qubit operator S_n , and its matrix is:

$$S_n = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}. \quad (12.5)$$

We define a matrix R which will give us the factor of 2 we're looking for, and is related to S as follows:

$$S_n = I - R_n = I - \begin{bmatrix} 2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix}. \quad (12.6)$$

Before and after S , we have a collection of Hadamard gates, $H^{\otimes n}$. This circuit is also known as the *Walsh-Hadamard* transform. We'll use the notation W_n to denote the n -qubit transform. For $n > 1$, the matrix can be defined recursively:

$$W_n = \frac{1}{\sqrt{2^n}} \begin{bmatrix} W_{n-1} & W_{n-1} \\ W_{n-1} & -W_{n-1} \end{bmatrix}. \quad (12.7)$$

Here are a couple of examples:

$$W_2 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (12.8)$$

$$W_3 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix}. \quad (12.9)$$

Element (i, j) of the matrix is defined as

$$(W_n)_{i,j} = (-1)^{i \cdot j}, \quad (12.10)$$

where $i \cdot j$ is the number of bits that are different in the binary representations of i and j , also known as the Hamming distance. For the purposes of this discussion, note that both the top row and the left column always contain 1.

So the *invert* circuit is described as WSW , leaving off the subscript n for convenience, which we rewrite as:

$$WSW = W(I - R)W = I - WRW = -(WRW - I), \quad (12.11)$$

noting that $WW = I$. Each element of WRW is calculated as:

$$WRW_{i,j} = (W_{i,0}R_{0,0}W_{0,j}) = \frac{2}{N}. \quad (12.12)$$

Therefore, it should be relatively clear that $WRW - I = D$, and $WSW = -D$. Note that there is a global phase of -1 applied to the output of the circuit, which makes no difference in the measured outcome, or when this outcome is fed into the next iteration.

To summarize: (1) Phase oracle U_p flips the phase of the basis state which satisfies the predicate. (2) WSW inverts the amplitudes around the mean. (3) These two steps

are repeated $O(\sqrt{N})$ times. (4) The output state is measured, which yields the desired basis state with high probability.

12.2.3 Generalizations

The algorithm assumes that there is only one basis state $|x\rangle$ for which $P(x) = 1$. It will also work with multiple matching states. Fewer iterations are needed to maximize the probability of measuring one of the matching states. For m matching states,

$$k = \frac{\pi}{4} \sqrt{\frac{N}{m}}. \quad (12.13)$$

Of course, the measurement will only yield one of the states; there is no guaranteed way to read them all.

Because the measurement likelihood goes down if too many iterations are performed, it is important for the number of iterations to agree with the number of possible answers. Otherwise, the algorithm can be run assuming $m = 1, 2, \dots$ until a satisfactory answer is measured.

A more generalized version of the algorithm is known as *amplitude amplification*. Instead of starting with $W|0\rangle$ (an equal superposition of all possible states), suppose we have a transform U such that $U|0\rangle$ gives an initial state with a higher probability for correct basis states.

Starting with the state $U|0\rangle$, the algorithm proceeds as above, except that WSW is replaced with USU^{-1} . If the algorithm U succeeds with probability t , then a classical repetition of U requires an average of $1/t$ iterations to find a solution. According to Rieffel and Polak [8, Chapter 9], amplitude amplification improves this bound to $O(\sqrt{1/t})$.

Grover extended his algorithm to solve other non-search problems, such as finding the mean and median of a function [146]. Other applications include finding collisions in functions [147] and string matching [148].

12.3 Quantum Fourier Transform

The Quantum Fourier Transform (QFT) is one of the most important building blocks for quantum algorithms. It was developed in the early 1990s by Peter Shor [149] and Don Coppersmith [150]. On its own, it is of interest because it exhibits an exponential speedup over the Fast Fourier Transform—a classical algorithm that can be used to compute the Discrete Fourier Transform when the number of samples is a power of 2. However, although the complete transform is available in the output state, a measurement will only give one component of the output with a probability proportional to the square of the corresponding Fourier coefficient. With this limitation its benefits may not be obvious, but it is a crucial component in factoring (Section 12.5) and other algorithms that show promise for quantum advantage. In particular, the QFT is useful when the measured value has some utility—some important information about the input state—or when the processed state is passed along to some other stage of quantum processing.

The QFT has much in common with the Discrete Fourier Transform (DFT), familiar to most electrical and computer engineers for its utility in digital signal processing. In the case of the QFT, the “time domain” signal is the set of basis state amplitudes in a quantum state. For an n -qubit state, the sequence of amplitudes for $|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$ is interpreted as a periodic signal that wraps around. The QFT relates to the frequencies associated with that periodic function.

12.3.1 Discrete Fourier Transform

The Discrete Fourier Transform takes a sequence of N complex numbers x_k and transforms them into a new sequence of complex numbers y_l , given by

$$y_l = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{2\pi i \frac{kl}{N}}. \quad (12.14)$$

The value $\omega_N = e^{2\pi i/N}$ is known as an N -th root of unity, and the equation above is often simplified as:

$$y_l = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k \omega_N^{kl}. \quad (12.15)$$

As mentioned, the most common case in electrical engineering is when the sequence x_k represents a sampled signal in time, and the x_k are real numbers.

Consider the case of a single non-zero sample: $x_0 = 1$, with $x_k = 0$ for all $k \neq 0$. In this case we have

$$y_l = \frac{1}{\sqrt{N}}, \quad (12.16)$$

i.e., the DFT consists of all “1s”. This is the discrete expression of the analog notion that an impulse contains all frequencies equally. Suppose instead that $x_1 = 1$ and all other samples are zero. This gives

$$y_l = \frac{1}{\sqrt{N}} e^{2\pi i \frac{l}{N}}. \quad (12.17)$$

Incrementing the index of the nonzero sample by 1 is equivalent to introducing a time delay of one sampling interval, and this results in a “spectrum” that is again uniform in magnitude, but now has a linear phase shift. This phase undergoes a shift of 2π for l ranging from 1 to N . In general, we can see that a time shift of m samples introduces a phase shift in the spectrum that has m multiples of 2π for l ranging from 1 to N .

For real-valued input sequences, the frequency-domain sequence has the property that $y_l = y_{N-l}^*$. The transform of a complex sequence does not have this symmetry. However, occasionally using examples of real sequences can help us connect to intuition and experience with Fourier transforms of time signals.

12.3.2 Inverse Discrete Fourier Transform

The DFT is unitary, so it has an inverse defined as:

$$x_k = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} y_l \omega_N^{-kl}. \quad (12.18)$$

Note how similar this is to the QFT: only the sign in the exponent is different! Let us again consider the case where only a single component of the input sequence is nonzero: $y_0 = 1$. In this case we have

$$x_k = \frac{1}{\sqrt{N}}, \quad (12.19)$$

i.e., all of the “time domain” samples are positive constants, so this represents a “DC” signal. Proceeding as before, suppose instead that $y_1 = 1$ and all other elements of the sequence are zero, this gives

$$x_k = \frac{1}{\sqrt{N}} e^{-2\pi i \frac{k}{N}}. \quad (12.20)$$

We obtain a complex “time domain” signal, because we omitted the “negative frequency” component of the signal, and consequently only included one exponential. The negative frequency component is given by y_{N-1} whose inverse transform is

$$x_k = \frac{1}{\sqrt{N}} e^{+2\pi i \frac{k}{N}}. \quad (12.21)$$

Clearly the sum of these inverse transforms will give a real-time domain function as expected.

Note that although we have given different interpretations of what happens when we take the DFT and IDFT, the effects of the transforms are very similar: a single non-zero element transforms into a sequence with uniform magnitude and linear phase variations for both transforms! In fact, it is common in electrical engineering to reverse the signs of the exponents (remember our convention $i = -j$). This effectively interchanges the definitions of the DFT and the IDFT. However, regardless of the sign convention on the transform, in common usage the DFT transforms the time or signal domain into the frequency or Fourier domain.

12.3.3 Quantum Implementation of the DFT

In the case of the QFT, the sequence is encoded in the input quantum state as the coefficients of the basis states. For example, let's start with an n -qubit quantum state $|\alpha\rangle = \sum_{k=0}^{N-1} x_k |k\rangle$, where $N = 2^n$ and $|k\rangle$ is a computational basis state corresponding to the binary representation of k . The QFT transforms this to a state $|\beta\rangle = \sum_{l=0}^{N-1} y_l |l\rangle$, where the amplitudes y_l are related to the amplitudes x_k by (12.14). After taking the QFT we say that the state is expressed in the *Fourier basis*.

Similar to our discussion of the DFT, suppose $|\alpha\rangle = |2\rangle$, meaning that $x_2 = 1$ and all other $x_k = 0$. To make our example concrete, let $N = 16$. Then,

$$y_l = \frac{1}{4} \omega_{16}^{2l} = \frac{1}{4} e^{2\pi i \frac{l}{8}}. \quad (12.22)$$

Each basis state therefore has an equal magnitude ($1/4$) and has a phase rotation of $l\pi/4$. If we view the states in sequence, as in Figure 12.6, we clearly see that the phase shift has completed two full cycles, as expected from our discussion of the DFT.

Each non-zero basis state $|k\rangle$ of the original state $|\alpha\rangle$ represents a sample of the encoded signal. Each basis state $|l\rangle$ of the resulting state $|\beta\rangle$ will be the sum of the corresponding phase shifted amplitudes resulting from the QFT of the basis states. For

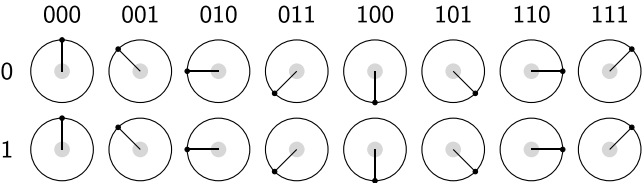


Figure 12.6 QFT of the state $|2\rangle$.

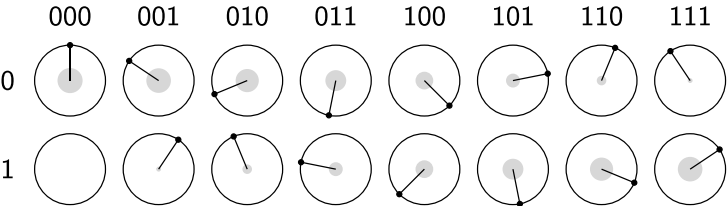


Figure 12.7 QFT of the state $(|2\rangle + |3\rangle)/\sqrt{2}$.

another example, let's consider $|\alpha\rangle = (|2\rangle + |3\rangle)/\sqrt{2}$. The output, shown in Figure 12.7 is the sum of the linear phase amplitudes of $\text{QFT}(|2\rangle)$ and $\text{QFT}(|3\rangle)$. Both the magnitude and phase vary, due to constructive and destructive interference.

Finally, consider that the DFT of a real-valued signal has frequency components such that complex conjugates are mirrored around $N/2$. For example, one period of a sine wave would have frequency peaks at 1 and $N - 1$, where the amplitudes of these peaks are complex conjugates of one another, as discussed previously. It follows that the QFT of the state $|k\rangle$ is the complex conjugate of the QFT of the state $|N - k\rangle$. This means that the QFT of a superposition of these states will yield a real function. As a concrete example, the QFT of the quantum state of $(|1\rangle + |15\rangle)/\sqrt{2}$ is shown in Figure 12.8. We see that the output state amplitudes are all real numbers (with phase equal to zero or π).

12.3.4 Encoding Quantum States

In the examples above, the encoded input signals were single basis states that can be created by starting with the ground state $|0\rangle^{\otimes n}$ and inserting X gates where needed to

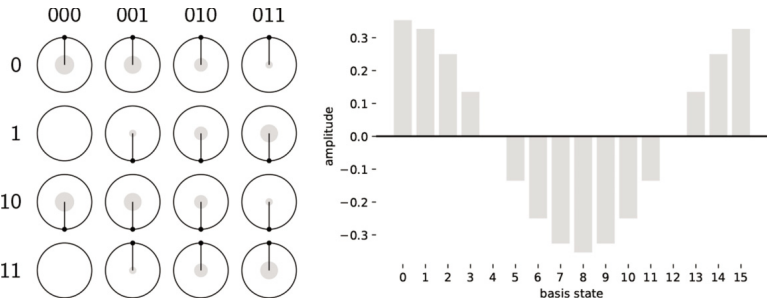


Figure 12.8 Result of QFT applied to $(|1\rangle + |15\rangle)/\sqrt{2}$ yields a real-valued Fourier-basis state. Amplitudes are shown in the plot on the right, clearly showing one period of a sinusoid.

represent the desired computational basis state. However, in general, we would like to create an input state containing multiple computational basis states with different complex amplitudes representing our data.

Creating an arbitrary state like this is one of the challenges of applying quantum computing to useful data (see, for example, Section 12.6.1). However, we can create a real square wave in a relatively straightforward way, as an illustration of a non-trivial input state.

We know that applying a Hadamard gate to each input qubit creates an equal superposition of all computational basis states. To flip the sign on half of these, we only need to add an X gate to the most significant qubit, as shown in Figure 12.9 for four qubits as an example. This creates the state

$$\frac{1}{4}[(|0\rangle - |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)]. \quad (12.23)$$

Multiplying this out shows that every state with “1” in the most significant qubit has a negative sign, so that the sequence of amplitudes creates a square wave. As shown in Figure 12.10, the amplitudes are real, alternating between $+1/N$ and $-1/N$. (The magnitudes are $1/N$, and phase alternates between 0 and π .) The output state of the QFT is shown in Figure 12.11. By plotting the amplitudes only in Figure 12.12, we see the familiar pattern of the frequency components of a square wave: a primary frequency (1) with decreasing odd harmonics, followed by a reflection around frequency $N/2$.

Figure 12.9 Circuit for creating a quantum-encoded square wave state with frequency = 1.

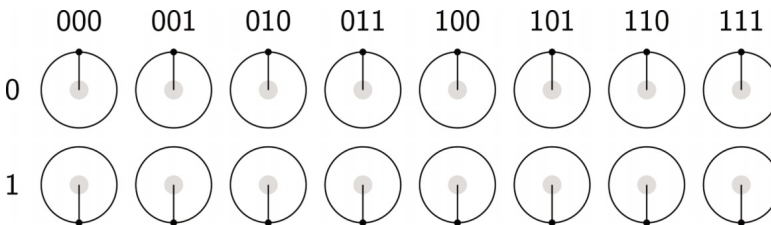
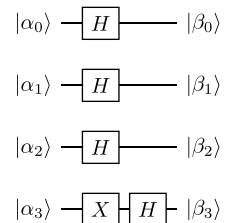
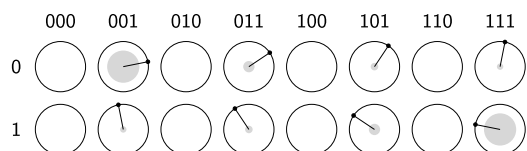


Figure 12.10 Quantum-encoded square wave state with frequency = 1 generated from the circuit in Figure 12.9.

Figure 12.11 Output of QFT applied to square wave.



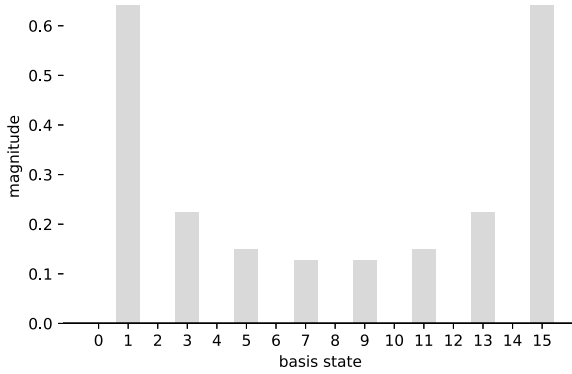


Figure 12.12 Magnitudes of basis states from Figure 12.11. The same pattern is observed when taking the DFT of a time-domain square wave.

12.3.5 Quantum Implementation

A quantum circuit for the 4-qubit QFT is shown in Figure 12.13. The purpose of this section is to explain how this circuit works.

Consider the quantum-encoded signal shown in Figure 12.6. How would we create this state without thinking about the QFT? We want a sequence of basis states that make two complete revolutions ($2\pi \times 2$) in phase. Since there are 16 basis states, each phase rotation should be incremented by $2\pi \times 2/16 = \pi/4$. We could: (a) start with four qubits in the $|0\rangle$ state, (b) apply Hadamard gates to get an equal superposition, and (c) rotate each component state $|k\rangle$ by $k \times \pi/4$.

To do part (c), we note that k can be written in binary as $k_3k_2k_1k_0$, and we can rewrite the multiplication in terms of powers of two:

$$k \times \frac{\pi}{4} = (k_3 \times 8) \frac{\pi}{4} + (k_2 \times 4) \frac{\pi}{4} + (k_1 \times 2) \frac{\pi}{4} + (k_0 \times 1) \frac{\pi}{4}. \quad (12.24)$$

The binary decomposition also applies to the quantum state $|k\rangle$, such that the j -th bit of k corresponds to the j -th qubit of $|k\rangle$. Therefore, if we add a relative phase of $2^j \times \pi/4$ to each qubit, we get our desired sequence. The state can be written as a tensor product of phase-shifted qubits:

$$\frac{1}{4} \left[(|0\rangle + e^{\frac{8\pi}{4}} |1\rangle) \otimes (|0\rangle + e^{\frac{4\pi}{4}} |1\rangle) \otimes (|0\rangle + e^{\frac{2\pi}{4}} |1\rangle) \otimes (|0\rangle + e^{\frac{\pi}{4}} |1\rangle) \right]. \quad (12.25)$$

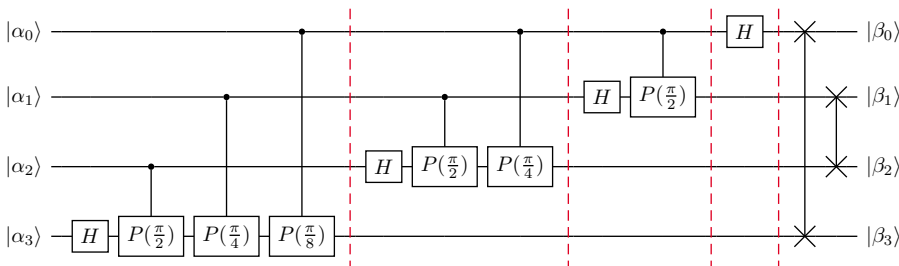


Figure 12.13 QFT for four qubits.

Figure 12.14 Relative phase rotations for each qubit in a 4-qubit QFT. Output state is a tensor product of these qubit states.

		freq = 2	freq = f
Qubit 3		$8 \times \frac{\pi}{4} \equiv 0$	$8 \times \frac{2\pi f}{16} = f\pi$
Qubit 2		$4 \times \frac{\pi}{4} = \pi$	$4 \times \frac{2\pi f}{16} = f\frac{\pi}{2}$
Qubit 1		$2 \times \frac{\pi}{4} = \frac{\pi}{2}$	$2 \times \frac{2\pi f}{16} = f\frac{\pi}{4}$
Qubit 0		$1 \times \frac{\pi}{4} = \frac{\pi}{4}$	$1 \times \frac{2\pi f}{16} = f\frac{\pi}{8}$

Figure 12.15 First stage of 4-qubit QFT.

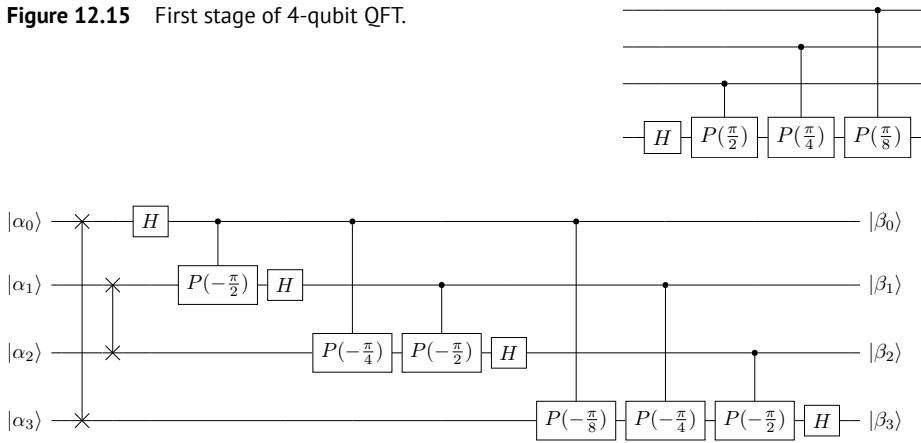


Figure 12.16 Inverse QFT for four qubits.

This rotation is illustrated in Figure 12.14. For the computational basis state $|k\rangle$, the total number of rotations must be k , which means that the incremental rotation is $2\pi k/N$.

Now consider the circuit in Figure 12.15, which is a copy of the first part of the 4-qubit QFT circuit from Figure 12.13. Assume that the input state is a single integer state $|k\rangle$. Because this is a 4-qubit transform, the smallest rotation increment will be $2\pi/2^4 = \pi/8$.

Assume for the moment that qubit 3 is $|0\rangle$, so the Hadamard gate produces $(|0\rangle + |1\rangle)/\sqrt{2}$. The other controlled- P gates perform a conditional rotation on the $|1\rangle$ component, and the amount of rotation is weighted according to the position of the control qubit.

- If qubit 0 is 1, rotate by $2^0 \times \frac{\pi}{8} = \frac{\pi}{8}$.
- If qubit 1 is 1, rotate by $2^1 \times \frac{\pi}{8} = \frac{\pi}{4}$.
- If qubit 2 is 1, rotate by $2^2 \times \frac{\pi}{8} = \frac{\pi}{2}$.

What about qubit 3? The Hadamard gate also performs a conditional rotation. If qubit 3 is 1, the initial state is $(|0\rangle - |1\rangle)/\sqrt{2}$, which adds a relative phase of $2^3 \times (\pi/8) = \pi$. The cumulative effect is that qubit 3 has a relative phase rotation of $k(2\pi/8)$. The next part of the circuit in Figure 12.13 does the same for qubit 2, except only the least

Table 12.1 Result of conditional phase rotations, the first four stages of Figure 12.13. The swap circuits at the end of the circuit reverse the order of the qubits, in order to match the desired rotations in Figure 12.14.

Qubit	Relative phase
3	$k \times \frac{2\pi}{N}$
2	$2k \times \frac{2\pi}{N}$
1	$4k \times \frac{2\pi}{N}$
0	$8k \times \frac{2\pi}{N}$

significant three qubits are used for the conditional rotation, and the rotation increments are doubled. This means that qubit 2 will have a relative phase of $2k(2\pi/8)$. The next two sections result in relative phases of $4k(2\pi/8)$ and $8k(2\pi/8)$ for qubits 1 and 0, respectively. The result is summarized in Table 12.1.

Compared to Figure 12.14, this is *almost* what we want, but the rotations have been applied to the qubits in the wrong order. Therefore, we need to reverse the order, which is accomplished by the swap gates at the end of Figure 12.13.

For a general n -qubit QFT, the number of stages increases to n , and the rotation increment is $2\pi/N = \pi/2^{n-1}$. Of course, additional swap gates are also needed to reverse the rotated qubits. For the inverse QFT circuit, the stages are in reverse order, and the rotation amounts are multiplied by -1 . The 4-qubit inverse QFT is shown in Figure 12.16.

12.3.6 Computational Complexity

The QFT provides an exponential scaling advantage, compared to the classical DFT. For a classical time-domain signal with 2^n samples, the best known algorithm is known as the Fast Fourier Transform (FFT) [151], which takes $O(n2^n)$ computational steps. In comparison, an n -qubit QFT requires n stages of n or fewer gates, resulting in a $O(n^2)$ steps. Rieffel and Polak [8, Chapter 7] show a QFT implementation that more closely follows the FFT.

The QFT should be considered a drop-in replacement for the FFT for most applications. The output is encoded in a quantum state, which means that collection of individual amplitudes of each sample/frequency is not available in a single execution and measurement. However, the scaling benefit does mean that quantum algorithms using the QFT cannot always be efficiently simulated classically using the FFT.

It has been shown that the QFT can be classically simulated for some input states, in the sense that a classical computer can efficiently sample from a distribution identical to the output of the QFT [152]. However, this has not been shown for the output of modular exponentiation, as used in Shor’s algorithm 12.5. Such a result would lead to an efficient classical algorithm for the factoring problem [8].

From a practical standpoint, note that the high-order qubit must interact with all other qubits in the first stage. Interacting with distant qubits can create significant overhead, as we observed in Chapter 11. In addition, as the QFT circuit becomes large, it spends more and more time doing smaller and smaller rotations [58]. Limiting the number of rotations in a stage results in an approximate QFT that may be “good

enough.” Limiting the number of gates is beneficial for near-term machines, and it also limits the interaction distance between qubits.

12.4 Quantum Phase Estimation

Quantum Phase Estimation (QPE) is another building block for quantum algorithms, for which the utility is not immediately obvious. It’s an important component of many interesting algorithms. Combined with Grover’s search algorithm (12.2), it leads to the *quantum counting algorithm*, which estimates the number of solutions to a search problem. Combined with the inverse QFT (12.3), it is the core of Shor’s factoring algorithm (12.5). It is a crucial part of algorithms that sample from the solution of a linear system of equations [153, 154].

The problem solved by QPE is the following: Given a unitary operator U with known eigenstate $|\psi\rangle$, find the eigenphase $0 \leq \theta < 1$ such that

$$U |\psi\rangle = e^{2\pi i\theta} |\psi\rangle. \quad (12.26)$$

More precisely, we will find an n -bit unsigned integer t such that $\theta \approx \frac{t}{2^n}$ with accuracy depending on the number of bits in t . (Another interpretation is that t represents an n -bit binary fraction, with the binary point to the left of the most significant bit: $0.t_{n-1}t_{n-2} \dots t_1t_0$.)

The major steps of the algorithm are:

1. Initialize an n -qubit register with $|0\rangle$ and create an equal superposition in the computational basis.
2. Rotate qubit j by $2^j \times 2\pi\theta$. This creates a quantum-encoded periodic signal, as described in Section 12.3, with frequency $2^n\theta$.
3. Use the inverse QFT to determine t , measure the output state, and compute $\theta \approx \frac{t}{2^n}$.

12.4.1 Quantum Implementation

A circuit for QPE is shown in Figure 12.17. In the first stage, we initialize the top qubits, which will hold the solution, as $|0\rangle$ and then use Hadamard gates to create an equal superposition in the computational basis.

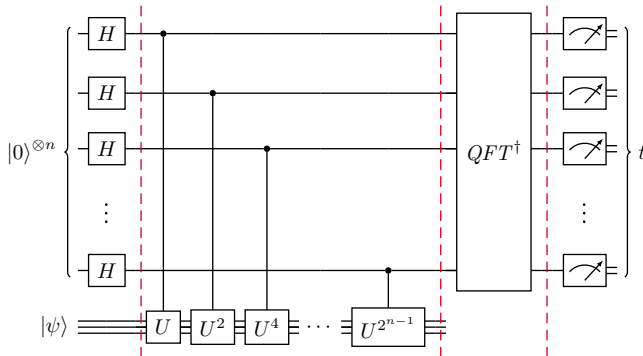


Figure 12.17 Quantum Phase Estimation. Following our normal convention, the least significant bit qubit for the phase is at the top.

For the bottom half of the circuit, the input $|\psi\rangle$ is an eigenstate of U . In particular, it is the eigenstate for which we want to learn the eigenphase. The algorithm still works if $|\psi\rangle$ is not an eigenstate, but the output is different, as we will discuss later.

The next stage of the circuit creates a quantum-encoded periodic signal, with $2\pi\theta$ as the incremental rotation for each basis state. The total rotation over 2^n states is $(2\pi)(2^n\theta) = 2\pi t$, which translates to a frequency of t .

Phase kickback (Section 11.4) is used to rotate the qubits in the top register. Qubit 0 is the control qubit for a controlled- U gate. Since $|\psi\rangle$ is an eigentate, then $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$. The phase is applied to qubit 0, and $|\psi\rangle$ is unchanged.

Qubit 1 has a weight of 2 in the output state, so we need to scale the rotation by a factor of 2. We accomplish this with a controlled- U^2 gate:

$$\begin{aligned} U^2|\psi\rangle &= U(e^{2\pi i\theta}|\psi\rangle) \\ &= (e^{2\pi i\theta})(e^{2\pi i\theta})|\psi\rangle \\ &= e^{2\pi i(2\theta)}|\psi\rangle. \end{aligned} \tag{12.27}$$

Similarly, each qubit j has a weight of 2^j , and we use controlled- U^{2^j} to realize a rotation of $2\pi(2^j\theta)$.

Now we have the encoded signal based on θ , but we don't know what θ is. We use an inverse QFT to change the state to $|t\rangle$ and we measure the state in the computational basis to recover t .

12.4.2 Computational Complexity and Other Issues

If we treat U^{2^x} as a black box taking one computational step, then the computational cost is $O(n^2)$, where n is the number of bits in t and also the number of qubits in the upper part of the circuit. There are n invocations of the black box, and the inverse QFT requires $O(n^2)$ steps.

Of course the time complexity of U may depend on the number of bits in $|\psi\rangle$. Conversion of U to controlled- U is essentially a constant, though non-trivial, overhead. If we naively repeat U to implement U^{2^x} , then this time scales exponentially with n and will dominate the computation time. For QPE to be efficient, there must be an efficient way to compute U^{2^x} .

Because t has limited precision, it may be that θ cannot be represented exactly. In that case, $|t\rangle$ will be a superposition of values that are close to θ . The closest value will have the highest amplitude, so there is a reasonable chance that the best approximate value will be measured. As in many quantum algorithms, the circuit can be executed a number of times, and the most frequently occurring output is very likely to be a good approximation of θ . More qubits provides a more precise answer, at the cost of increased circuit depth.

What if the eigenstates of U are not known? Any state can be expressed as a linear combination of eigenstates of U . Therefore, if a non-eigenstate is used as input instead of $|\psi\rangle$, the output will be a superposition of eigenphases. Therefore, any measured value will be an approximation of some eigenphase, but the algorithm will give no information about the eigenstates.

12.5 Shor's Algorithm

Shor's algorithm is the most famous and arguably most important quantum algorithm discovered to date. Published in 1994 [149], the algorithm solves the integer factoring problem: given a number M , find two prime numbers p and q such that $pq = M$. For large integers, this is an extremely hard problem for classical computers; the best known classical algorithm requires $O(e^{m^{1/3}})$ time, where m is the number of bits required to represent M . In contrast, Shor's algorithm requires $O(m^2 \log m \log \log m)$ steps, reducing a sub-exponential problem to a polynomial problem.

The impact of this algorithm was huge, because it showed a clear quantum advantage on a problem that is important to a lot of people. The widely-used RSA public-key cryptography system [155] is based on the difficulty of factoring a large publicly-known integer. If this problem becomes easy due to quantum computing, then a lot of encrypted secret information would become vulnerable. As we'll see, the problem is certainly not easy, or even feasible, with the small quantum computers that we can build today, but the ability to factor a large integer (e.g., 2048 bits) serves as a milestone and benchmark for practical quantum computing [110].

Incidentally, Shor's algorithm is also applicable to the discrete logarithm problem, which is the basis for a different public key scheme known as Diffie-Hellman, used to exchange secret key information [156, 157]. The algorithm has spawned a new subfield, known as *post-quantum cryptography*, devoted to finding cryptographic schemes based on mathematical problems for which quantum computers are not expected to have an advantage.

12.5.1 Hybrid Classical-Quantum Algorithm

Shor's algorithm is not a "pure" quantum algorithm. It is an example of a hybrid algorithm, with work divided between classical and quantum processors. While quantum processors can perform classical operations (Chapter 11), classical processors are much faster and more efficient if that's all you're doing. The quantum processor is best used for things that are impractical for classical hardware.

A simplified flowchart for factoring M is shown in Figure 12.18. Only the gray box, finding the period of the modular exponentiation function, is performed on a quantum computer. The classical part of the computation includes finding the *greatest common divisor* (GCD) of two integers (Euclidean algorithm), and extracting the period from the measured value returned by the quantum computation (continued fractions).

1. Choose a random integer a between zero and M . Computing the GCD tells us whether a and M have a factor in common. If there is a common factor greater than one, then we have found a factor of M , so we return it. Otherwise, a and M are relatively prime, and we proceed with the algorithm.
2. Use the quantum processor to find the period r of the function $f(x) = a^x \bmod M$. This process is described in the next section.
3. If the period r is even, then we know that both $a^{r/2} + 1$ and $a^{r/2} - 1$ will have a common factor with M , because:

$$(a^{r/2} + 1)(a^{r/2} - 1) = a^r - 1 \equiv 0 \pmod{M}. \quad (12.28)$$

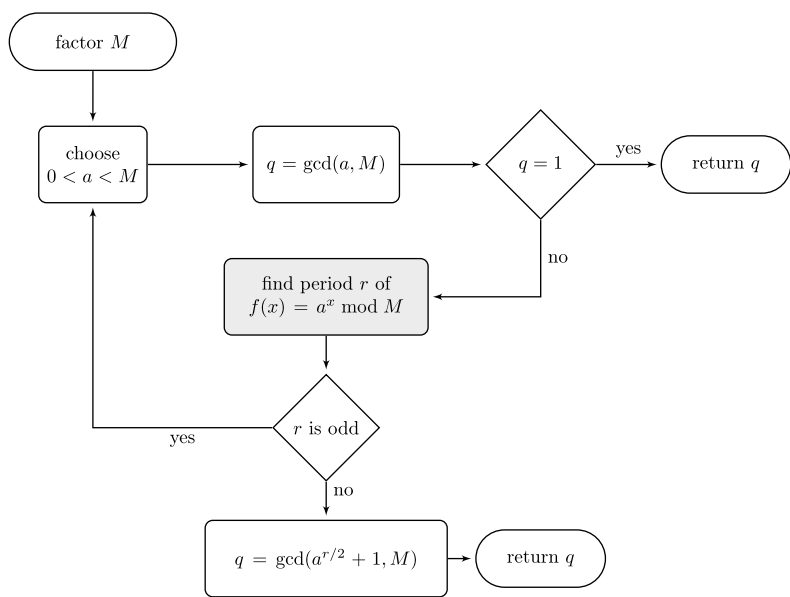


Figure 12.18 Factoring integer M using Shor's algorithm. The gray box is the only part of the algorithm that runs on a quantum computer.

If r is odd, then the rest of the algorithm does not work, so we return to the first step and choose a new value of a . We also choose a new a if either $a^{r/2} + 1$ or $a^{r/2} - 1$ is a multiple of M . (This is not shown in the flowchart.)

4. We can now find a factor of M by computing the GCD using either $a^{r/2} + 1$ or $a^{r/2} - 1$.

12.5.2 Finding the Period

For any relatively prime numbers a and M , the function $f(x) = a^x \bmod M$ is periodic. The period is the smallest integer r such that $f(x) = f(x+r)$. For example, Figure 12.19 plots the values of $f(x) = 3^x \bmod 35$, showing that the period is 12.

For small values of M , we could repeatedly multiply by a until we see a result of 1, indicating the beginning of a new period. Unfortunately, the worst case number of iterations for this method scales exponentially with $m = \log_2 M$, the number of bits required to represent M . On a quantum computer, however, if we can encode the values from Figure 12.19 in the amplitudes of a quantum state, we can use the QFT to extract information about the period.

Consider a unitary transform U that performs modular multiplication with a and M as fixed values²:

$$U |y\rangle = |ay \bmod M\rangle. \tag{12.29}$$

We can compute $a^x \bmod M$ using repeated applications of this transform. Returning to the specific case of $a = 3$ and $M = 35$ from Figure 12.19, we know that the period

² To make this unitary and reversible, we also need to specify that $U |y\rangle = |y\rangle$ for $y \geq M$. It's easier to leave out this detail in the description.

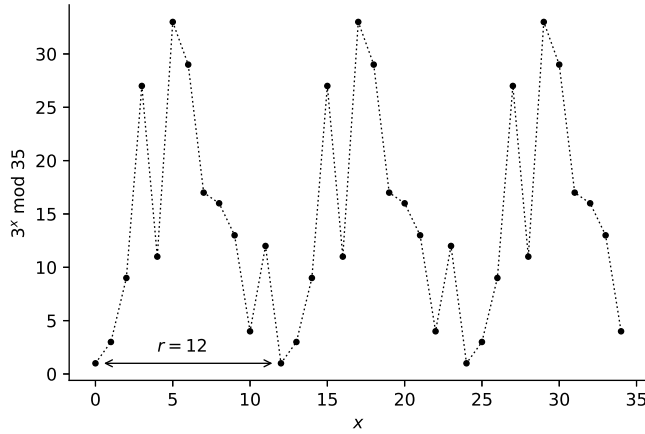


Figure 12.19 For $a = 3$ and $M = 35$, the period of $a^x \bmod M$ is 12. Finding the period by brute force scales exponentially with m , the number of bits in M .

$r = 12$:

$$\begin{aligned}
 U|1\rangle &= |3\rangle \\
 U^2|1\rangle &= U|3\rangle = |9\rangle = |3^2 \bmod 35\rangle \\
 U^3|1\rangle &= U|9\rangle = |27\rangle = |3^3 \bmod 35\rangle \\
 U^4|1\rangle &= U|27\rangle = |11\rangle = |3^4 \bmod 35\rangle \\
 &\dots \\
 U^{r-1}|1\rangle &= U|4\rangle = |12\rangle = |3^{11} \bmod 35\rangle \\
 U^r|1\rangle &= U|12\rangle = |1\rangle = |3^{12} \bmod 35\rangle.
 \end{aligned} \tag{12.30}$$

In more general terms,

$$\begin{aligned}
 U^x|y\rangle &= |a^x y \bmod M\rangle \\
 U^x|1\rangle &= |a^x \bmod M\rangle.
 \end{aligned} \tag{12.31}$$

To compute U^x , is there a more efficient way than having x copies of U ? Yes, if you're willing to do a little classical precomputation. Consider x written as an n -bit binary value:

$$x = x_{n-1} \cdot 2^{n-1} + x_{n-2} \cdot 2^{n-2} + \dots + x_1 \cdot 2^1 + x_0 \cdot 2^0. \tag{12.32}$$

We can rewrite U^x using this binary expansion:

$$\begin{aligned}
 U^x|y\rangle &= U^{x_{n-1} \cdot 2^{n-1} + x_{n-2} \cdot 2^{n-2} + \dots + x_1 \cdot 2^1 + x_0 \cdot 2^0}|y\rangle \\
 &= U^{x_{n-1} \cdot 2^{n-1}} U^{x_{n-2} \cdot 2^{n-2}} \dots U^{x_1 \cdot 2^1} U^{x_0 \cdot 2^0}|y\rangle \\
 &= \prod_{j=0}^{n-1} U^{x_j \cdot 2^j}|y\rangle = \prod_{j=0}^{n-1} |a^{x_j \cdot 2^j} y \bmod M\rangle.
 \end{aligned} \tag{12.33}$$

The values of x_j will be zero or one. If zero, we don't perform the multiplication. We can classically precompute $a^{2^j} \bmod M$, so we need at most n modular multiplications, one for each non-zero bit in x .

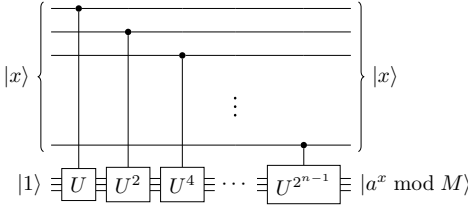


Figure 12.20 Modular exponentiation circuit using repeated modular multiplication, where $U^k |y\rangle = |a^k y \bmod M\rangle$.

Figure 12.20 shows a quantum circuit to perform modular exponentiation in this way. If $|x_j\rangle$ is zero, the multiplication is not performed. If $|x_j\rangle$ is one, the multiplication is performed. According to Shor [149], the most efficient circuit for modular multiplication takes $O(n \log n \log \log n)$ steps for n bits, so this form of exponentiation takes $O(n^2 \log n \log \log n)$ steps.

We can use Hadamard gates to create an equal superposition of x values, and we will end up with a superposition of all $a^x \bmod M$ values. That's quantum parallelism—in the same time it takes to compute one value, we can compute 2^n values. This essentially results in a state that looks like the graph in Figure 12.19.

However, all those values are encoded in one quantum state, and we don't have access to them. If we measure the outcome, we only get one specific randomly-selected value $\tilde{x}$ and its corresponding value $a^{\tilde{x}} \bmod M$. This tells us nothing about the period r , and even multiple measurements will only give us a random sample with no usable information.

Luckily, the inverse Quantum Fourier Transform (12.3) is designed to extract frequency information from a quantum-encoded signal. In fact, the circuit in Figure 12.20 looks a lot like the first part of the circuit for Quantum Phase Estimation (12.4), extracting the eigenphase of U . Let's see if the eigenphase gives us any information about r .

Again, consider the values shown in Figure 12.19 for $a = 3$ and $M = 35$. Consider a superposition state of the first r values of $a^x \bmod M$:

$$\begin{aligned} |\lambda_0\rangle &= \frac{1}{\sqrt{12}} (|1\rangle + |3\rangle + |9\rangle + \cdots + |4\rangle + |12\rangle) \\ &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |a^k \bmod M\rangle. \end{aligned} \quad (12.34)$$

It is easy to see that this is an eigenstate of U , since applying U sends $|1\rangle \rightarrow |3\rangle$, $|3\rangle \rightarrow |9\rangle$, ..., $|12\rangle \rightarrow |1\rangle$. The eigenvalue corresponding to $|\psi_0\rangle$ is 1:

$$U |\lambda_0\rangle = |\lambda_0\rangle. \quad (12.35)$$

For a more interesting eigenstate, consider applying a phase rotation of $-2\pi i(k/r)$ to the ket representing the k -th power of a :

$$\begin{aligned} |\lambda_1\rangle &= \frac{1}{\sqrt{r}} (|1\rangle + e^{\frac{-2\pi i}{r}} |3\rangle + e^{\frac{-4\pi i}{r}} |9\rangle + \cdots \\ &\quad + e^{\frac{-20\pi i}{r}} |4\rangle + e^{\frac{-22\pi i}{r}} |12\rangle) \\ U |\lambda_1\rangle &= \frac{1}{\sqrt{r}} (|3\rangle + e^{\frac{-2\pi i}{r}} |9\rangle + e^{\frac{-4\pi i}{r}} |27\rangle + \cdots \\ &\quad + e^{\frac{-20\pi i}{r}} |12\rangle + e^{\frac{-22\pi i}{r}} |1\rangle) \\ &= e^{2\pi i \frac{1}{r}} |\lambda_1\rangle. \end{aligned} \quad (12.36)$$

This is very interesting. If we could create the eigenstate λ_1 , we could measure the phase to get a close approximation of $2\pi/r$. Of course, we would have to know r in order to create λ_1 , so that's not helpful.

Based on the logic above, there are r eigenstates, each with a phase that is a multiple of $2\pi/r$:

$$\begin{aligned}
 |\lambda_j\rangle &= \frac{1}{\sqrt{r}} \left(|1\rangle + e^{\frac{-2j\pi i}{r}} |3\rangle + e^{\frac{-4j\pi i}{r}} |9\rangle + \dots \right. \\
 &\quad \left. + e^{\frac{-20j\pi i}{r}} |4\rangle + e^{\frac{-22j\pi i}{r}} |12\rangle \right) \\
 U |\lambda_j\rangle &= \frac{1}{\sqrt{r}} \left(|3\rangle + e^{\frac{-2j\pi i}{r}} |9\rangle + e^{\frac{-4j\pi i}{r}} |27\rangle + \dots \right. \\
 &\quad \left. + e^{\frac{-20j\pi i}{r}} |12\rangle + e^{\frac{-22j\pi i}{r}} |1\rangle \right) \\
 &= e^{2\pi i \frac{j}{r}} |\lambda_j\rangle.
 \end{aligned} \tag{12.37}$$

If we can't create any of these eigenstates without knowing r , how does this help? Recall that any quantum state can be written as a superposition of eigenstates of U . So if we use *any* state as the input to QPE, even one that is easy to prepare, the result will be a superposition of eigenphases.

A very interesting observation is that $|1\rangle$ is an *equal* superposition of all of the $|\lambda_j\rangle$ eigenstates.

$$|1\rangle = \frac{1}{r} \sum_{j=0}^{r-1} |\lambda_j\rangle. \tag{12.38}$$

Therefore, if we create a QPE circuit for U with $|1\rangle$ as the input, the state after the inverse QFT will be an *equal* superposition of all of the eigenphases $2\pi(j/r)$. This circuit is shown in Figure 12.21.

When we measure this outcome, we will get one of the eigenphases, which we call $\phi = s/r$ in the figure. But we don't know which one, which means we don't know s

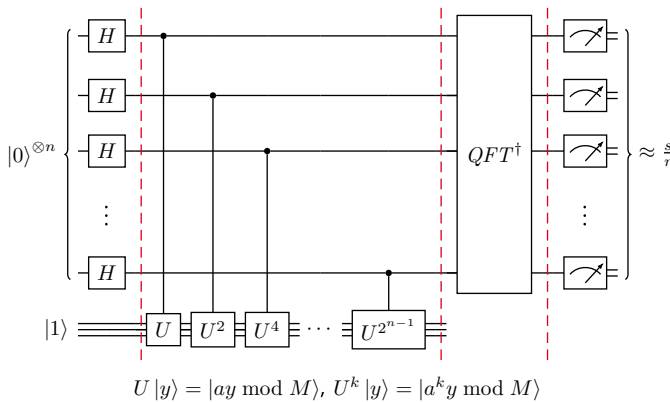


Figure 12.21 Quantum circuit for period-finding in Shor's algorithm. M is the integer to be factored, a is a number that is relatively prime to M , and r is the period of $a^k \bmod M$. The number of qubits n is chosen such that $M^2 \leq 2^n < 2M^2$. The measured eigenphase approximates a multiple of $1/r$, and classical post-processing determines r .

and we don't know r . Therefore, we need to do some classical post-processing on the measured value to estimate r . We use the technique of *continued fraction expansion*, which is well-described by Rieffel and Polak [8, Chapter 8]. In the case where r is a power of 2, we get an exact answer. Otherwise, the measured result will yield a fraction that is close to s/r with high probability. With large enough n , only a few executions of the algorithm are needed to have a high probability of success.

12.5.3 Computational Complexity

The quantum component of the factoring algorithm is dominated by the modular exponentiation and the inverse QFT. As described in Section 12.3, the inverse QFT requires $O(n^2)$ time for n qubits. The discussion above shows that modular exponentiation is $O(m^2 \log m \log \log m)$. Since $n \approx 2m$, the cost of modular exponentiation dominates. The cost of state preparation is trivial, so the overall quantum complexity is $O(m^2 \log m \log \log m)$.

The classical computation time is quite modest. There are n pre-computed values of a^{2^k} , each of which takes $O(m^2 n)$. The GCD calculations use Euclid's algorithm, which is $O(n)$ for n -bit numbers. Continued fraction expansion is likewise linear in n . Recalling that classical processor cycles are much faster than quantum gate times in modern systems, it is clear that the quantum computation dominates the overall execution time.

As shown in the flowchart (Figure 12.18) and discussed above, there are four things that can require the calculation to be repeated [8, Chapter 8]:

- The period r could be odd.
- The calculation could yield M as a factor of M .
- The measured QPE value might not be close enough to a multiple of $1/r$.
- The values s and r in the measured value s/r may have a common factor, in which case the reduced fraction will have a denominator that is a factor of r , rather than r itself.

Classical analysis bounds the probability of the first two cases to at most $1/2$. For the third case, the choice of n and the use of $|1\rangle$ as an equal superposition of eigenstates gives a high probability of measuring a value that is close to a multiple of $1/r$. Shor shows that the last case requires only $O(\log \log r)$ attempts to have a high probability of success [149].

Therefore, if we can build a quantum computer that is sufficiently large, fast, and reliable, it seems clear that there is a value of m for which factoring an m -bit integer M will be significantly faster than classical processing. Gidney and Ekarå [110] give a detailed analysis of the resources needed to factor a 2048-bit number, as well as a review of the literature on implementing Shor's algorithm. There are other candidate algorithms for factoring, such as Variational Quantum Factoring [158], which nicely leads us to our next section.

12.6 Variational Quantum Algorithms

The algorithms described so far in this chapter, as well as Chapter 11, are written for scalable, fault-tolerant quantum computers. While efforts are made to optimize the

number of qubits or the depth of the circuit, they generally assume an unlimited number of error-free qubits, as well as the ability for any qubit to interact with any other. As we have described in this textbook, current quantum systems do not match those characteristics. As a consequence, current hardware can successfully execute only very small instances of algorithms like QPE or Shor's factoring algorithm.

Since quantum hardware has become publically available, a major effort has emerged to find algorithms that are suitable for addressing practical problems on near-term NISQ machines. As noted by Cerezo *et al.* [159], such algorithms must account for:

- Limited number of qubits.
- Limited interconnectivity among qubits.
- Coherent and incoherent errors, including short decoherence times, that limit the depth of quantum circuits.

Variational quantum algorithms (VQA) represent a strategy that combines parameterized quantum circuits with classical optimization and machine learning approaches [159]. The name comes from the *variational principle* in mathematics, which allows a problem to be solved using the *calculus of variations*, finding functions that can be optimized to discover quantities of interest. An example is Hamilton's Principle, used in Section 6.1 to describe the physics of superconducting qubits.

The first such algorithm was the *Variational Quantum Eigensolver (VQE)* [160], along with the closely-related *Quantum Approximate Optimization Algorithm (QAOA)* [161], discovered around the same time. These algorithms will be described in some detail later in this section, but the term is now used for any algorithm based on the VQA strategy.

The structure of a VQA is shown in Figure 12.22. The vector θ is a set of discrete or continuous *parameters* that will be used to adapt the quantum circuit to produce different states. The parameterized quantum circuit is known as an *ansatz*. Starting with a known (and easy to produce) initial state, the ansatz is a unitary transform that is meant to estimate the desired quantum state that will lead to a problem solution. The circuit uses the values in θ to generate different states; for example, parameter θ_i might be the angle of rotation for a particular gate in the circuit. In the figure, the quantum state is shown as $|\psi(\theta)\rangle$.

The next component is a *cost function*, $C(\theta)$, that represents the problem to be solved. The cost function is based on measured observables of the state $|\psi(\theta)\rangle$, and it is calculated by the classical computer. The cost function must be minimal when applied to a

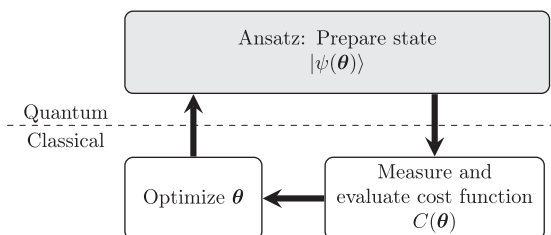


Figure 12.22 Components of a Variational Quantum Algorithm (VQA). Parameterized state preparation and measurement occur on the quantum processor, followed by classical optimization of the cost function.

state that represents a solution. We will discuss cost functions more in the context of VQE and QAOA below.

Finally, a classical *optimizer* is used to find the parameters that minimize the cost function. A new set of parameters produces a new quantum circuit, which generates a new $|\psi(\theta)\rangle$, and the cycle repeats until the cost function is minimized. One type of optimizer is based on the idea of *gradient descent*. By changing the parameters a bit and re-evaluating $C(\theta)$, you can determine the rate of change in various “directions.” You then want to make changes along the gradient that decreases the cost the most. The choice of optimizer is informed by a large body of literature in both optimization and machine learning. A few examples that are often mentioned in the VQA literature include SPSA [162], COBYLA [163], and Adam [164].

The output of the algorithm depends on the problem that is being solved. We could be interested in the expectation value of a particular observable, the bit string represented by the qubit states, or a probability distribution. In addition, the optimized version of θ may be the desired outcome, i.e., the ability to prepare this state, or to use the parameterized ansatz as a quantum operator.

The next two sections describe two fundamental and widely-used VQAs. Then we will wrap up with a discussion of challenges and opportunities for running VQAs on near-term hardware.

12.6.1 Variational Quantum Eigensolver

The Variational Quantum Eigensolver (VQE) [160] is a method for finding the minimum eigenvalue of a known Hamiltonian, which describes the *energy* of a system. The minimum eigenvalue corresponds to the *ground state*, the state with the lowest energy. In quantum chemistry, the ground state energy of a particular molecule gives insight into other useful properties, such as reaction rates, binding energies, and molecular pathways [165]. In general, finding the ground state energy is NP-hard [166, 167].

12.6.1.1 Eigenvalues and Expectations

The energy of a system is described by its Hamiltonian, $\mathcal{H}$. This operator represents a measurable value, and all of its eigenvalues are real. $\mathcal{H}$ can be represented as a linear combination of operations on its eigenstates:

$$\mathcal{H} = \sum_i \lambda_i |u_i\rangle\langle u_i|, \quad (12.39)$$

where $|u_i\rangle$ is an eigenstate and λ_i is the corresponding eigenvalue. Furthermore, we’ve seen before that any quantum state can be written as a superposition of an operator’s eigenstates.

Because $\mathcal{H}$ models a physical system, it has a minimum energy—the ground state energy. The value of the ground state energy (E_0) is given by the minimum eigenvalue, which we will denote as λ_0 , and the corresponding eigenstate $|u_0\rangle$ is the ground state.

$$E_0 = \lambda_0 = \mathcal{H} |u_0\rangle. \quad (12.40)$$

Since VQE’s goal is to find the ground state energy of $\mathcal{H}$, this is equivalent to finding the minimum eigenvalue.

If we apply $\mathcal{H}$ to an arbitrary quantum state $|\psi\rangle$ and measure the outcome, what would we expect to see? If $|\psi\rangle$ is an eigenstate, then we would only see the eigenvalue associated with that state. Otherwise, we will see one of the eigenvalues (λ_i), with a probability $|a_i|^2$ determined by the amplitude a_i of the corresponding eigenstate:

$$|\psi\rangle = \sum_i a_i |u_i\rangle. \quad (12.41)$$

A single measurement of $\mathcal{H}$ $|\psi\rangle$ does not give us much information about $|\psi\rangle$. We need to prepare and measure the state multiple times and collect information about the various measurements.

In statistics, the *expectation* (or *expected value*) of a random variable X is the sum of its possible values times the probability of each value. In quantum mechanics, a measurement outcome is a random variable with a specific probability distribution. The expectation of operator $\mathcal{H}$ is denoted as $\langle \mathcal{H} \rangle$, and the expectation for a given state $|\psi\rangle$ is:

$$\langle \mathcal{H} \rangle_\psi \equiv \langle \psi | \mathcal{H} | \psi \rangle. \quad (12.42)$$

Rewriting $\mathcal{H}$ in terms of its eigenstates and eigenvalues gives us the following result:

$$\begin{aligned} \langle \mathcal{H} \rangle_\psi &= \langle \psi | \mathcal{H} | \psi \rangle = \langle \psi | \left(\sum_i \lambda_i |u_i\rangle \langle u_i| \right) | \psi \rangle \\ &= \sum_i \lambda_i \langle \psi | u_i \rangle \langle u_i | \psi \rangle \\ &= \sum_i \lambda_i |\langle u_i | \psi \rangle|^2. \end{aligned} \quad (12.43)$$

Since each of the weights $|\langle u_i | \psi \rangle|^2 \geq 0$ and the sum of the weights must equal 1, the expected value of any state $|\psi\rangle$ will be greater than or equal to the ground state energy:

$$\langle \mathcal{H} \rangle_\psi \geq \lambda_0 = E_0. \quad (12.44)$$

We can think of it this way: if we start with the correct ground state and add in a little bit of any other eigenstate, the probability of measuring the extra state must be subtracted from the probability of measuring the ground state. This replaces a little bit of the ground state eigenvalue in (12.43) with an equal amount of a larger eigenvalue. Consequently any mixture other than the pure ground state will yield an expectation value larger than that of the ground state.

12.6.1.2 Ansatz

Applying the variational method to the problem involves creating a state $|\psi\rangle$ that is an estimate of the ground state, known as an *ansatz*. As described above, the ansatz is created using a parameterized quantum circuit, with a vector of parameters θ . By adjusting θ , we can produce a state $|\psi(\theta)\rangle$ that gets closer and closer to the ground state.

An ansatz can be *problem-inspired* or *problem-agnostic*. If information is available about the likely range of the ground state, based on the problem being solved, that may be used to restrict the choice of circuit and may lead to faster convergence. Examples

of this approach from the quantum chemistry domain include variants of the Unitary Coupled Cluster method [64, 168].

The problem-agnostic approach, on the other hand, chooses a circuit from a set of *variational forms* that are designed to generate states over a large range of the state space. This property of an ansatz is known as *expressibility*.

For a single qubit, the state space is the Bloch sphere. Figure 12.23 illustrates the expressibility of a few circuits, where R_x , R_z , and U are parameterized. In Circuit A, rotating around the z-axis will only generate states around the equator, while Circuit B can access more of the space by allowing rotations around both the x- and z-axes.

For multi-qubit states, it is important to also introduce entangling gates to explore the Hilbert space. Figure 12.24 shows a few examples of ansatzes considered in the expressibility study by Sim *et al.* [170] The choice of an ansatz may also be driven by its ease of implementation on specific hardware, known as *hardware-efficient* ansatzes. For example, entangling gates between neighboring qubits will be a better fit for superconducting quantum processors with limited interconnectivity.

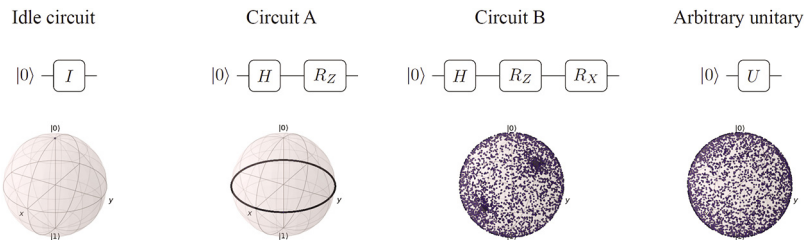


Figure 12.23 Expressibility of various ansatzes, illustrated by 2000 sampled states [170].

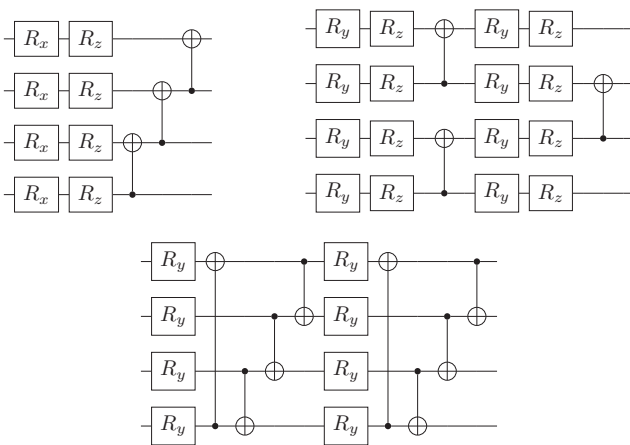


Figure 12.24 Examples of problem-agnostic ansatz circuits from Sim *et al.* (Redrawn from [170]). R_x , R_y , and R_z gates are parameterized. In general, CNOT gates can be replaced with controlled- $\{Z, R_z, R_x, R_y\}$. Each circuit is a *layer* that can be repeated.

12.6.1.3 Measuring the Hamiltonian

As described above, the cost function for VQE is the expectation of the Hamiltonian, $\langle \mathcal{H} \rangle$. But where does the Hamiltonian come from, and how do we measure it? The first question is beyond the scope of this discussion, as it is problem-dependent. For quantum chemistry, the molecular structure determines electron–electron and electron–nuclei interactions, and well-known fermion-to-qubit transformations (such as Jordan-Wigner and Bravyi-Kitaev) are used to generate a Hamiltonian that is a linear combination of Pauli operators [171]. For example, a two-qubit Hamiltonian for the hydrogen molecule is

$$\mathcal{H} = g_0 I + g_1 Z_0 + g_2 Z_1 + g_3 Z_0 Z_1 + g_4 Y_0 Y_1 + g_5 X_0 X_1, \quad (12.45)$$

where the Pauli subscripts represent the qubit, and g_i are efficiently computable constants from the hydrogen–hydrogen bond length [64]. We then measure the various Pauli operators separately, multiply (classically) by the scaling factors, and sum the expectations:

$$\langle \mathcal{H} \rangle = g_0 + g_1 \langle Z_0 \rangle + g_2 \langle Z_1 \rangle + g_3 \langle Z_0 Z_1 \rangle + g_4 \langle Y_0 Y_1 \rangle + g_5 \langle X_0 X_1 \rangle. \quad (12.46)$$

Some of the Pauli operators do not commute, and therefore we cannot measure them all at once. We can group together commuting terms, but we must use separate circuits for non-commuting measurements. Depending on the capabilities of the quantum computer, the circuits can be run concurrently on different sets of qubits, or sequentially.

In addition, most quantum processors only measure in the Z basis. To measure in the Y or X basis, we can perform projective measurements (as shown for stabilizers in Chapter 10) or perform rotations into the Z basis prior to the measurements (see Section 2.3). In other words, to measure in the X basis, first apply a Hadamard gate H and then measure in the Z basis. To measure in the Y basis, first apply the operator $HS^\dagger$ and then measure in the Z basis. In all cases, measuring the $|0\rangle$ state corresponds to an eigenvalue of $+1$, and measuring $|1\rangle$ corresponds to an eigenvalue of -1 .

12.6.1.4 Summary

To summarize, the steps performed for the VQE algorithm are:

1. Determine the Hamiltonian $\mathcal{H}$ that expresses the energy of the system.
2. Choose an ansatz with an initial set of parameters θ .
3. Generate a candidate state $|\psi(\theta)\rangle$.
4. Measure and compute the expectation $\langle \psi(\theta) | \mathcal{H} | \psi(\theta) \rangle$.
5. Pass the expectation and θ to a classical optimizer.
6. Repeat steps 3–5 with new θ to find $\langle \mathcal{H} \rangle_{\min} \approx E_0$.

While the discussion above is in the context of quantum chemistry, VQE and its variants can be used in many domains, given an appropriate Hamiltonian. Example domains include combinatorial optimization [172], financial portfolio optimization [173], quantum simulation [174], and linear algebra [175]. We will further discuss the challenges and opportunities for VQE in the larger context of VQAs (12.6.3), but we first describe QAOA, the other best-known variational algorithm, specifically designed for combinatorial optimization.

12.6.2 Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) [161] was developed around the same time as VQE, and they have many common properties. But its origins are more closely linked to *adiabatic quantum computing* than to variational methods. Its goal is to provide approximate solutions to combinatorial optimization problems.

As described by Farhi *et al.* [161], a combinatorial optimization problem involves n bits and m clauses. Each clause is a constraint on a subset of the bits, which is satisfied for certain assignments (0 or 1) of those bits and unsatisfied for all other assignments. The objective function is

$$C(x) = \sum_{\alpha=1}^m C_{\alpha}(x) \quad (12.47)$$

where x is an n -bit string and $C_{\alpha}(x) = 1$ if clause α is satisfied. In most cases, only a few bits are involved in each C_{α} . Approximate optimization asks for a string whose cost is close to the maximum of $C(x)$.

The number of combinations (potential solutions) often grows exponentially (e^n) or combinatorially ($n!$). Most such problems are NP-hard, meaning there is no known polynomial solution for large problems. Well-known problems in this space are the Travelling Salesperson Problem, the Knapsack Problem, and various graph/network problems, such as MaxCut.

Similar to VQE, we start with a Hamiltonian, H_C , that describes the objective function that we want to minimize or maximize. (We will discuss how to create this Hamiltonian below.) Then we create a parameterized unitary operator:

$$U(H_C, \gamma) = e^{-i\gamma H_C}. \quad (12.48)$$

Second, we create a *mixing operator* out of Pauli X operators on each of N qubits:

$$H_B = \sum_i^N X_i \quad (12.49)$$

$$U(H_B, \beta) = e^{-i\beta H_B}. \quad (12.50)$$

The mixing operator is very easy to implement, as the exponentiation results in a collection of $R_x(2\beta)$ gates.

Now we choose a number of steps, p , and a set of angles γ_i and β_i for each step, and construct the state $|\gamma, \beta\rangle$ as follows:

$$|\gamma, \beta\rangle = U(H_B, \beta_p)U(H_C, \gamma_p)U(H_B, \beta_{p-1})U(H_C, \gamma_{p-1}) \cdots U(H_B, \beta_1)U(H_C, \gamma_1) |s\rangle, \quad (12.51)$$

where $|s\rangle$ is an initial state, usually $|00 \dots 000\rangle$ or $|++ \dots + ++\rangle$.

When we measure $|\gamma, \beta\rangle$, we get a *bit string* that encodes a potential solution to the problem. We evaluate the cost function for the solution, and use a classical optimizer to choose a new set of γ and β , and iterate until we find a good solution.

The algorithm is not guaranteed to find an exact solution to the problem, but the approximation improves as p increases. So there is a clear tradeoff between circuit depth ($O(p)$) and solution quality.

12.6.2.1 Encoding the Objective Function

Consider a specific clause $C_\alpha(x)$ from the objective function above. Suppose the clause is only satisfied when a small subset of bits Q are all set to 1, and another small subset $\bar{Q}$ are set to 0. To count toward the objective function, the term in the objective function related to this clause must look like this:

$$C_\alpha(x) = \prod_{i \in Q} x_i \prod_{j \in \bar{Q}} (1 - x_j). \quad (12.52)$$

To convert to a Hamiltonian, we replace each binary variable x_i with a Pauli Z operator acting on a single qubit, using the following substitution:

$$x_i \mapsto \frac{1}{2}(1 - Z_i). \quad (12.53)$$

This means that $(x_i = 0)$ corresponds to a measured eigenvalue of $+1$ in the computational (Z) basis, and $(x_i = 1)$ corresponds to a measured eigenvalue of -1 .

The Hamiltonian term for that clause then takes on this form:

$$H_\alpha = \frac{1}{2^{|Q|+|\bar{Q}|}} \prod_{i \in Q} (1 - Z_i) \prod_{j \in \bar{Q}} (1 + Z_j). \quad (12.54)$$

The overall Hamiltonian $H_C = \sum H_\alpha$ is diagonal and k -local if at most k qubits are involved in each term. (With some luck, the circuit will not be too deep for NISQ hardware.)

As a concrete example, consider the MaxCut problem for an unweighted graph. This is the most commonly-used example for QAOA, because it's easy to understand, and it's the example used in the original QAOA paper [161].

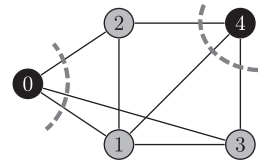
The goal of MaxCut is to partition a graph such that the number of edges between the two partitions is maximized. This can be viewed as assigning one of two colors to each vertex; an edge is considered “cut” if it connects vertices with different colors. Figure 12.25 shows an example, where a black vertex is encoded as 1, and a gray vertex is 0. E is the set of edges. For an edge between vertices i and j , we consider both 01 and 10 as assignments that will contribute toward the objective.

$$\begin{aligned} C(x) &= \sum_{(i,j) \in E} w_i(1 - w_j) + (1 - w_i)w_j \\ H_C &= \sum_{(i,j) \in E} \frac{1}{4}(1 - Z_i)(1 + Z_j) + \frac{1}{4}(1 + Z_i)(1 - Z_j) \\ &= \sum_{(i,j) \in E} \frac{1}{2}(1 - Z_i Z_j). \end{aligned} \quad (12.55)$$

12.6.2.2 Ising and QUBO Formulations

A general technique for mapping combinatorial optimization problems to Hamiltonians is derived from the *Ising model* of ferromagnetic materials. The model consists of

Figure 12.25 Example of MaxCut: Partition a graph such that the number of edges crossing between the partitions is maximized.



discrete variables, each with a spin of +1 (up) or −1 (down). Variables are connected in a graph, usually a lattice, and the energy depends on the relative spin values on neighboring sites.

The Ising model is normally written as a function of N spins:

$$H(s) = - \sum_{i < j} J_{ij} s_i s_j - \sum_{i=1}^N h_i s_i \quad (12.56)$$

where J_{ij} is the strength of interaction between spins i and j , and h_i is an additional magnetic field acting on spin i . This is converted to a quantum Hamiltonian by replacing spins with the Pauli Z operator.

Ising models are very flexible for encoding binary combinatorial problems. Lucas [176] provides mappings for many NP-hard and NP-complete problems, with the number of spins at most cubic in the problem size. In the mathematical literature, similar problems are more commonly known as *Quadratic Unconstrained Binary Optimization (QUBO)* formulations. Glover *et al.* [177] provide a good overview with many examples. Ising/QUBO formulations are the basis for hardware that performs *quantum annealing*, which is less general than universal quantum computing, but may provide quantum speedup on some optimization problems [178].

12.6.2.3 QAOA Ansatz

As shown earlier, the ansatz is built by alternating parameterized unitaries, based on the cost Hamiltonian and the mixing Hamiltonian:

$$U(H_C, \gamma) = e^{-i\gamma H_C}, U(H_B, \beta) = e^{-i\beta H_B}. \quad (12.57)$$

For H_C , the circuit is generally built around phase (R_x) and controlled-phase gates. For example, Figure 12.26 shows one term of $U(H_C, \gamma)$ for the MaxCut problem: phase is rotated by γ_k if qubits i and j are different.

The mixing Hamiltonian, H_B , is composed of X gates, so $U(H_B, \beta)$ is simply a rank of $R_x(2\beta)$ gates acting on each qubit. While other mixing circuits can be used [179], it is important that the mixing operators do not commute with the cost function operators, in order to enhance the expressibility of the ansatz.

Figure 12.27 shows the first part of a QAOA circuit for the MaxCut problem described earlier. Each edge performs a phase rotation if the vertex assignment is different, and the R_x gates perform the mixing operator. For depth of p , the ansatz is repeated p times, and the optimizer chooses a value of γ and β for each instance.

12.6.2.4 Comparison with VQE

While QAOA can be viewed as a special case of VQE, with a specific ansatz, the motivation from combinatorial optimization leads to some distinctions that are helpful to note.

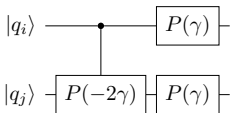


Figure 12.26 Circuit to rotate qubits i and j by $-\gamma$, if they are different.

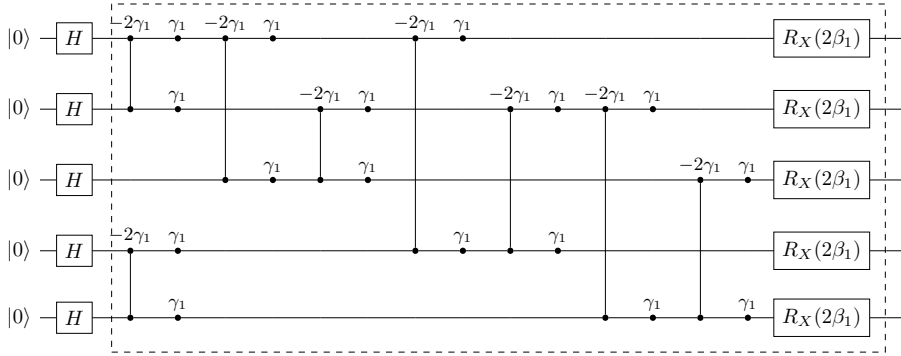


Figure 12.27 QAOA circuit for MaxCut example in Figure 12.25. Dashed box is repeated p times with different parameters γ_i and β_i in each iteration. Dot notation is used for phase rotation gates to save space.

- The ansatz is derived from the problem instance, rather than a variational form designed to explore the Hilbert space. Depending on the problem, this often results in a circuit that is deeper than those often used for VQE.
- Only single-qubit measurements in the computational basis are required, which simplifies the circuit execution.
- The goal is to find a good near-optimal answer, rather than to find a single solution.

12.6.3 Challenges and Opportunities

VQAs are a very active area of research, because they may offer the best opportunity to demonstrate quantum advantage with the first generation of quantum computers. The hybrid strategy allows quantum processors to do what they do best, while exploiting the extensive capabilities provided by classical optimization and machine learning tools.

We don't have a section on computational complexity or run-time advantage, because VQA is a framework, not a specific algorithm. Run-time will depend on the depth of the ansatz, the measurement scheme, the number of parameters, the efficiency of the classical optimizer, and the number of iterations required to converge to a good answer. These issues are difficult to quantify in a general way.

For application areas that are inherently quantum, such as quantum chemistry, nuclear physics, and quantum simulation, the advantage comes from the ability to prepare states that are classically intractable, and also in the ability to model time evolution of the Hamiltonian. For examples, see Cao *et al.* [165] and Kokail *et al.* [180]. For optimization and other problems expressed in the computational basis, QAOA is a preferred option because it provides some guarantee about the quality of the approximation. QAOA tends to be more resource-intensive, but Harrigan *et al.* [181] have demonstrated results on a physical machine using 23 qubits. They have shown cases where performance improves with $p > 1$, despite the increased circuit depth (and increased errors) required for such circuits.

VQAs appear to be fairly robust to noise, particularly for the VQE-style problems, and full error correction is not mandatory. The noise techniques described in Chapter 9 appear to be well-suited to improving the accuracy and reliability of results. This allows

the additional physical qubits provided by hardware scaling to be applied to solving larger problems, rather than implementing logical qubits.

One area of concern, however, comes with the need to optimize over more parameters as problem sizes scale. For some choices of ansatz, large regions of the optimization space appear to be fairly flat, giving little information about how to adjust parameters. Known as a *barren plateau*, the problem has been shown to get worse with more qubits, further restricting the fruitful area of exploration [182]. Potential solutions involve starting with a better initial state, choosing better ansatzes, or optimizing layer-by-layer using adaptive circuits.

While we have described the two best-known VQAs, algorithms have been developed in a variety of areas, including factorization [158], matrix decomposition [183], quantum simulation [184], and compilation of quantum programs [185].

12.7 Summary and Further Reading

In this chapter, we have described only a few major quantum algorithms. These were chosen because of their prevalence as building blocks for quantum applications, and because they illustrate properties that indicate the potential for speedups over classical computation. After a few decades of research, the source of quantum computing's power is still an open question, as eloquently discussed by Rieffel and Polak [8].

Quantum parallelism and the exponential size of the quantum state space are often cited as the primary source of speedup. However, only n bits of state can be extracted from n qubits; to get values of $f(x)$ for all 2^n values of x still requires 2^n operations, the same as classical processing. Algorithms that rely on quantum parallelism for speedup require nontraditional programming techniques, such as amplifying amplitudes of desired states (Grover) or finding properties of the entire set of $f(x)$ values (Shor).

Similarly, the n -qubit quantum state is extremely large, so that the majority of states cannot be approximated by an efficient quantum algorithm. Therefore, quantum parallelism does not come close to exploring the entire state space.

Entanglement is also mentioned as the source of quantum computing's power, and there is no classical analog. Josza and Linden [186] show that any pure state quantum algorithm that shows exponential speedup must make use of entanglement among a number of qubits that increases with the size of the problem. However, similar results can be proved for other properties, and the result does not hold for less dramatic speedups. Other applications of quantum processing do not include any entanglement at all.

One observation that should be made from this chapter is that quantum computers in the early 2020s are far from the scale required for fault-tolerant universal quantum computing, at least at the scale needed to solve classically intractable problems. Variational algorithms and those based on *quantum walks* [187] offer paths that may be more resistant to noise and more useful with short-depth circuits in the near term.

The Quantum Algorithm Zoo [188] is a catalog of known quantum algorithms, and Montanaro [189] provides an excellent overview of past research and promising directions. The availability of commercial systems, software infrastructure, and educational

material have inspired a new generation of quantum enthusiasts, and will continue to accelerate the development of new approaches and applications.

12.8 Exercises

12.1 Consider a quantum state consisting of two four-qubit values $|a\rangle$ and $|b\rangle$. Design a phase oracle that flips the phase when b is the reverse of a . In other words, if $a = 1011$, then flip the phase when $b = 1101$.

12.2 Design a phase oracle that flips the phase of input qubits $|x_j\rangle$ if the following function returns `true`. ($\wedge$ is AND, $\vee$ is OR, $\neg$ is NOT.)

$$f(x_2, x_1, x_0) = (\neg x_2 \vee \neg x_1 \vee \neg x_0) \wedge (x_2 \vee \neg x_1 \vee x_0) \wedge (x_2 \vee x_1 \vee \neg x_0) \\ \wedge (x_2 \vee \neg x_1 \vee \neg x_0) \wedge (\neg x_2 \vee x_1 \vee x_0).$$

How many combinations of $x_2x_1x_0$ will be flipped?

- 12.3** (a) Use the brute force method to determine the period r of $f(x) = 11^x \bmod 21$.
 (b) Write the r eigenstates of U $|y\rangle = |11y \bmod 21\rangle$.
 (c) Show that the normalized sum of all of the eigenstates is equal to $|1\rangle$.
 (d) Show that both $11^{r/2} + 1$ and $11^{r/2} - 1$ have factors in common with 21. You can use Euclid's algorithm to find the GCD, but you also probably know the prime factors of 21, so you can just check those.

Bibliography

- [1] Edward Fredkin and Tommaso Toffoli. “Conservative Logic”. In: *Int J Theor Phys* 21.3 (Apr. 1, 1982), pp. 219–253.
- [2] Gadi Aleksandrowicz et al. *Qiskit: An Open-source Framework for Quantum Computing*. 2019. DOI: 10.5281/zenodo.2562110.
- [3] David P. DiVincenzo. *The Physical Implementation of Quantum Computation*. Apr. 2000. arXiv: 0002077 [quant-ph]
- [4] David C. McKay et al. “Efficient Z-Gates for Quantum Computing”. In: *Physical Review A* 96.2 (Aug. 2017), p. 022330.
- [5] Hermann A. Haus. *Waves and Fields in Optoelectronics*. Prentice-Hall, 1983. ISBN: 978-0-139-46053-1.
- [6] H.A. Haus and W. Huang. “Coupled-Mode Theory”. In: *Proceedings of the IEEE* 79.10 (Oct. 1991), pp. 1505–1518.
- [7] Amnon Yariv. *Optical Electronics in Modern Communications*. 5th ed. Oxford University Press, 1997. ISBN: 978-0-195-10626-8.
- [8] Eleanor Rieffel and Wolfgang Polak. *Quantum Computing: A Gentle Introduction*. The MIT Press, 2014. ISBN: 978-0-262-52667-8.
- [9] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. ISBN: 978-1-107-61919-7.
- [10] Adriano Barenco et al. “Elementary Gates for Quantum Computation”. In: *Phys. Rev. A* 52.5 (Nov. 1995), pp. 3457–3467.
- [11] Michael J. Bremner et al. “Practical Scheme for Quantum Computation with Any Two-Qubit Entangling Gate”. In: *Physical Review Letters* 89.24, 247902 (Nov. 2002).
- [12] R. C. Bialczak et al. “Quantum Process Tomography of a Universal Entangling Gate Implemented with Josephson Phase Qubits”. In: *Nature Phys* 6.6 (June 2010), pp. 409–413.
- [13] David C. McKay et al. “Universal Gate for Fixed-Frequency Qubits via a Tunable Bus”. In: *Phys. Rev. Applied* 6.6, 064007 (Dec. 2016).
- [14] Chad Rigetti and Michel Devoret. “Fully Microwave-Tunable Universal Gates in Superconducting Qubits with Linear Couplings and Fixed Transition Frequencies”. In: *Phys. Rev. B* 81.13, 134507 (Apr. 2010).
- [15] Jerry M. Chow et al. “A Simple All-Microwave Entangling Gate for Fixed-Frequency Superconducting Qubits”. In: *Physical Review Letters* 107.8, 080502 (Aug. 2011).

- [16] Chad Rigetti, Rémy Mosseri, and Michel Devoret. “Geometric Approach to Digital Quantum Information”. In: *Quantum Information Processing* 3.6 (Dec. 1, 2004), pp. 351–380.
- [17] Dagmar Bruss. *Partial Trace*. Lecture notes from Theoretical Quantum Optics and Quantum Information, Heinrich-Heine-Universität Düsseldorf. 2015. URL: http://www.thphy.uni-duesseldorf.de/~ls3/teaching/1515-QOQI/Additional/partial_trace.pdf. Accessed Dec. 29, 2021.
- [18] Kurt Jacobs. *Quantum Measurement Theory and its Applications*. Cambridge University Press, 2014. ISBN: 978-1-107-02548-6.
- [19] Fawwaz Ulaby, Eric Michielssen, and Umberto Ravaioli. *Fundamentals of Applied Electromagnetics*. Pearson College Div, 2010. ISBN: 0-13-213931-6.
- [20] R. P. Owens. “Accurate Analytical Determination of Quasi-Static Microstrip Line Parameters”. In: *Radio and Electronic Engineer* 46.7 (July 1976), pp. 360–364.
- [21] David M. Pozar. *Microwave Engineering*. 4th ed. Wiley, 2012.
- [22] M. V. Schneider. “Microstrip Lines for Microwave Integrated Circuits”. In: *Bell System Technical Journal* 48.5 (May 1969), pp. 1421–1444.
- [23] H. A. Wheeler. “Transmission-Line Properties of Parallel Wide Strips by a Conformal-Mapping Approximation”. In: *IEEE Transactions on Microwave Theory and Techniques* 12.3 (May 1964), pp. 280–289.
- [24] H. A. Wheeler. “Transmission-Line Properties of Parallel Strips Separated by a Dielectric Sheet”. In: *IEEE Transactions on Microwave Theory and Techniques* 13.2 (Mar. 1965), pp. 172–185.
- [25] Erli Chen and S. Y. Chou. “Characteristics of Coplanar Transmission Lines on Multilayer Substrates: Modeling and Experiments”. In: *IEEE Transactions on Microwave Theory and Techniques* 45.6 (June 1997), pp. 939–945.
- [26] M. Göppl et al. “Coplanar Waveguide Resonators for Circuit Quantum Electrodynamics”. In: *Journal of Applied Physics* 104.11, 113904 (Dec. 2008).
- [27] C. P. Wen. “Coplanar Waveguide: A Surface Strip Transmission Line Suitable for Nonreciprocal Gyromagnetic Device Applications”. In: *IEEE Transactions on Microwave Theory and Techniques* 17.12 (Dec. 1969), pp. 1087–1090.
- [28] Koki Watanabe et al. “Kinetic Inductance of Superconducting Coplanar Waveguides”. In: *Japanese Journal of Applied Physics* 33.10R (Oct. 1994), pp. 5708–5712.
- [29] Dean A. Frickey. “Calculation of S Parameters from ABCD Parameters with Complex Normalizing Impedances”. In: *Microwave and Optical Technology Letters* 5.12 (1992), pp. 613–615.
- [30] Joseph Helszajn. *Microwave Polarizers, Power Dividers, Phase Shifters, Circulators, and Switches*. Wiley-IEEE Press, 2018. ISBN: 978-1-119-49005-0.
- [31] A. A. Clerk et al. “Introduction to Quantum Noise, Measurement, and Amplification”. In: *Reviews of Modern Physics* 82.2 (Apr. 2010), pp. 1155–1208.
- [32] “IRE Standards on Methods of Measuring Noise in Linear Twoports, 1959”. In: *Proceedings of the IRE* 48.1 (1960), pp. 60–68.
- [33] H.T. Friis. “Noise Figures of Radio Receivers”. In: *Proceedings of the IRE* 32.7 (July 1944), pp. 419–422.
- [34] Thomas H. Lee. *Planar Microwave Engineering*. Cambridge University Press, 2004. ISBN: 978-0-521-83526-8.

- [35] Jose Aumentado. “Superconducting Parametric Amplifiers: The State of the Art in Josephson Parametric Amplifiers”. In: *IEEE Microwave Magazine* 21.8 (Aug. 2020), pp. 45–59.
- [36] Jerry M. Chow et al. “Implementing a Strand of a Scalable Fault-Tolerant Quantum Computing Fabric”. In: *Nature Communications* 5.1, 4015 (Sept. 2014).
- [37] A. D. Córcoles et al. “Demonstration of a Quantum Error Detection Code Using a Square Lattice of Four Superconducting Qubits”. In: *Nature Communications* 6.1, 6979 (Apr. 2015).
- [38] M. D. Hutchings et al. “Tunable Superconducting Qubits with Flux-Independent Coherence”. In: *Phys. Rev. Applied* 8.4, 044003 (Oct. 2017).
- [39] Herbert Goldstein, Charles Poole, and John Safko. *Classical Mechanics*. Addison-Wesley, 2002. ISBN: 978-0-201-65702-9.
- [40] Jon Mathews and Robert L. Walker. *Mathematical Methods of Physics*. 2nd ed. W.A. Benjamin, 1970. ISBN: 978-0-8053-7002-7.
- [41] Eugen Merzbacher. *Quantum Mechanics*. 3rd ed. Wiley, 1997. ISBN: 978-0-471-88702-7.
- [42] R. Shankar. *Principles of Quantum Mechanics*. 2nd ed. Plenum Press, 1994. ISBN: 978-0-306-44790-7.
- [43] John S. Townsend. *A Modern Approach to Quantum Mechanics*. 2nd ed. University Science Books, 2012. ISBN: 978-1-891389-78-8.
- [44] Lev S. Bishop. *Circuit Quantum Electrodynamics*. 2010. arXiv: 1007 . 3520 [cond-mat , physics:quant-ph]
- [45] Alexandre Blais et al. “Cavity Quantum Electrodynamics for Superconducting Electrical Circuits: An Architecture for Quantum Computation”. In: *Phys. Rev. A* 69.6, 062320 (June 2004).
- [46] Alexandre Blais et al. “Quantum Information Processing with Circuit Quantum Electrodynamics”. In: *Phys. Rev. A* 75.3, 032329 (Mar. 2007).
- [47] M. H. Devoret. “Quantum Fluctuations in Electrical Circuits”. In: *Quantum Fluctuations*. Ed. by S. Reynaud, E. Giacobino, and J. Zinn-Justin. Les Houches, Session LXIII, 1995. Elsevier, 1997. ISBN: 978-0-444-82593-3.
- [48] Philip Krantz et al. “A Quantum Engineer’s Guide to Superconducting Qubits”. In: *Applied Physics Reviews* 6.2, 021318 (June 2019).
- [49] David Isaac Schuster. “Circuit Quantum Electrodynamics”. Ph.D. dissertation. Yale University, 2007.
- [50] T. Van Duzer and C.W. Turner. *Principles of Superconductive Devices and Circuits*. Elsevier, 1981. ISBN: 978-0-444004-11-6.
- [51] Michael Tinkham. *Introduction to Superconductivity*. 2nd ed. Dover, 1996. ISBN: 978-0-486435-03-9.
- [52] R. K. Pathria and Paul D. Beale. *Statistical Mechanics*. 4th ed. Academic Press, 2021. ISBN: 978-0-081026-92-2.
- [53] W. Meissner and R. Ochsenfeld. “Ein Neuer Effekt Bei Eintritt Der Supraleitfähigkeit”. In: *Naturwissenschaften* 21 (1933), pp. 787–788.
- [54] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. “Theory of Superconductivity”. In: *Phys. Rev.* 108.5 (Dec. 1957), pp. 1175–1204.
- [55] Leon N. Cooper. “Bound Electron Pairs in a Degenerate Fermi Gas”. In: *Phys. Rev.* 104.4 (Nov. 1956), pp. 1189–1190.

- [56] Neil W. Ashcroft and N. David Mermin. *Solid State Physics*. Thomson Press, 2003. ISBN: 978-8-131500-52-1.
- [57] Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics, Vol III: Quantum Mechanics*. Pearson, 1970. ISBN: 978-0-201021-15-8.
- [58] Abraham Asfaw et al. *Learn Quantum Computation Using Qiskit*. 2020. URL: <http://community.qiskit.org/textbook>. Accessed Dec. 29, 2021.
- [59] L. DiCarlo et al. “Demonstration of Two-Qubit Algorithms with a Superconducting Quantum Processor”. In: *Nature* 460 (July 2009), pp. 240–244.
- [60] T. Yamamoto et al. “Flux-Driven Josephson Parametric Amplifier”. In: *Applied Physics Letters* 93.4, 042510 (July 2008).
- [61] Ananda Roy and Michel Devoret. “Introduction to Parametric Amplification of Quantum Signals with Josephson Circuits”. In: *Comptes Rendus Physique* 17.7 (Aug. 2016), pp. 740–755.
- [62] John Preskill. “Quantum Computing in the NISQ Era and Beyond”. In: *Quantum* 2 (Aug. 2018), p. 79.
- [63] G. Ithier et al. “Decoherence in a Superconducting Quantum Bit Circuit”. In: *Physical Review B* 72, 134519 (2005).
- [64] Peter James Joyce O’Malley. “Superconducting Qubits: Dephasing and Quantum Chemistry”. PhD thesis. University of California, Santa Barbara, June 2016. URL: <https://web.physics.ucsb.edu/~martinisgroup/theses/OMalley2016.pdf>. Accessed Dec. 29, 2021.
- [65] H. Y. Carr and E. M. Purcell. “Effects of Diffusion on Free Precession in Nuclear Magnetic Resonance Experiments”. In: *Physical Review* 94.3 (May 1954), pp. 630–638.
- [66] S. Meiboom and D. Gill. “Modified Spin-Echo Method for Measuring Nuclear Relaxation Times”. In: *Review of Scientific Instruments* 29.8 (1958), p. 688.
- [67] Benjamin Nachman and Michael R. Geller. *Categorizing Readout Error Correlations on Near Term Quantum Computers*. Apr. 2021. arXiv: 2104.04607 [quant-ph]
- [68] Manpreet Singh Jattana et al. “General Error Mitigation for Quantum Circuits”. In: *Quantum Information Processing* 19.414 (Nov. 2020).
- [69] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. “Characterizing Quantum Gates via Randomized Benchmarking”. In: *Physical Review A* 85.4 (Apr. 2012), p. 042311.
- [70] David C. McKay et al. “Three-Qubit Randomized Benchmarking”. In: *Physical Review Letters* 122.20 (May 2019), p. 200502.
- [71] Easwar Magesan et al. “Efficient Measurement of Quantum Gate Error by Interleaved Randomized Benchmarking”. In: *Physical Review Letters* 109.8 (Aug. 2012), p. 080505.
- [72] Zijun Chen et al. “Measuring and Suppressing Quantum State Leakage in a Superconducting Qubit”. In: *Phys. Rev. Lett.* 116.2, 020501 (Jan. 2016).
- [73] J. M. Chow et al. “Optimized Driving of Superconducting Artificial Atoms for Improved Single-Qubit Gates”. In: *Phys. Rev. A* 82.4, 040305 (Oct. 2010).
- [74] Erik Lucero et al. “Reduced Phase Error through Optimized Control of a Superconducting Qubit”. In: *Phys. Rev. A* 82.4, 042339 (Oct. 2010).

- [75] F. Motzoi and F. K. Wilhelm. “Improving Frequency Selection of Driven Pulses Using Derivative-Based Transition Suppression”. In: *Phys. Rev. A* 88.6, 062318 (Dec. 2013).
- [76] F. Motzoi et al. “Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits”. In: *Phys. Rev. Lett.* 103.11, 110501 (Sept. 2009).
- [77] John G. Proakis and Masoud Salehi. *Fundamentals of Communication Systems*. 2nd ed. Pearson, 2014. ISBN: 978-0-133354-85-0.
- [78] Suguru Endo, Simon C. Benjamin, and Ying Li. “Practical Quantum Error Mitigation for Near-Future Applications”. In: *Physical Review X* 8, 031027 (July 2018).
- [79] Ying Li and Simon C. Benjamin. “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”. In: *Physical Review X* 7.2 (June 2017), p. 8744.
- [80] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. “Error Mitigation for Short-Depth Quantum Circuits”. In: *Physical Review Letters* 119, 180509 (Nov. 2017).
- [81] Abhinav Kandala et al. “Error Mitigation Extends the Computational Reach of a Noisy Quantum Processor”. In: *Nature* 567 (Mar. 2019), pp. 491–495.
- [82] Lewis Fry Richardson and J. Arthur Gaunt. “The Deferred Approach to the Limit”. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 226 (Jan. 1927), pp. 299–361.
- [83] Yuval Baum et al. *Experimental Deep Reinforcement Learning for Error-Robust Gateset Design on a Superconducting Quantum Computer*. 2021. arXiv: 2105.01079 [quant-ph]
- [84] *Quantum Control*. URL: <https://q-ctrl.com/foundations/quantum-control/> (visited on 07/27/2021).
- [85] R. W. Hamming. “Error Detecting and Error Correcting Codes”. In: *Bell System Technical Journal* 29.2 (Apr. 1950), pp. 147–160.
- [86] Peter W. Shor. “Scheme for Reducing Decoherence in Quantum Computer Memory”. In: *Physical Review A* 52.4 (Oct. 1995), R2493–R2496.
- [87] Daniel Gottesman. “Stabilizer Codes and Quantum Error Correction”. PhD Thesis. California Institute of Technology, 1997. arXiv: 9705052 [quant-ph]
- [88] Andrew Steane. “Multiple-Particle Interference and Quantum Error Correction”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (Nov. 1996), pp. 2551–2577.
- [89] A. R. Calderbank and Peter W. Shor. “Good Quantum Error-Correcting Codes Exist”. In: *Physical Review A* 54.2 (Aug. 1996), pp. 1098–1105.
- [90] John Preskill. “Reliable Quantum Computers”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (Jan. 1998), pp. 385–410.
- [91] Richard Cleve and Daniel Gottesman. “Efficient Computations of Encodings for Quantum Error Correction”. In: *Physical Review A* 56.1 (July 1997), pp. 76–82.
- [92] Bryan Eastin and Emanuel Knill. “Restrictions on Transversal Encoded Quantum Gate Sets”. In: *Physical Review Letters* 102.11, 110502 (Mar. 2009).

- [93] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. “Resilient Quantum Computation”. In: *Science* 279.5349 (Jan. 1998), pp. 342–345.
- [94] A. Yu Kitaev. “Fault-Tolerant Quantum Computation by Anyons”. In: *Annals of Physics* 303.1 (Jan. 2003), pp. 2–30.
- [95] S. B. Bravyi and A. Yu Kitaev. *Quantum Codes on a Lattice with Boundary*. Nov. 1998. arXiv: 9811052 [quant-ph]
- [96] Michael H. Freedman and David A. Meyer. “Projective Plane and Planar Quantum Codes”. In: *Foundations of Computational Mathematics* 1.3 (July 2001), pp. 325–332.
- [97] Austin G. Fowler et al. “Surface Codes: Towards Practical Large-Scale Quantum Computation”. In: *Physical Review A* 86.3 (Sept. 2012), p. 032324.
- [98] Jack Edmonds. “Paths, Trees, and Flowers”. In: *Canadian Journal of Mathematics* 17 (1965), pp. 449–467.
- [99] Vladimir Kolmogorov. “Blossom V: A New Implementation of a Minimum Cost Perfect Matching Algorithm”. In: *Mathematical Programming Computation* 1.1 (July 2009), pp. 43–67.
- [100] Savvas Varsamopoulos, Koen Bertels, and Carmen G. Almudever. “Decoding Surface Code with a Distributed Neural Network – Based Decoder”. In: *Quantum Machine Intelligence* 2.1 (Mar. 2020), p. 3.
- [101] Clare Horsman et al. “Surface Code Quantum Computing by Lattice Surgery”. In: *New Journal of Physics* 14.12 (Dec. 2012), p. 123011.
- [102] Daniel Litinski and Felix von Oppen. “Lattice Surgery with a Twist: Simplifying Clifford Gates of Surface Codes”. In: *Quantum* 2 (May 2018), p. 62.
- [103] Austin G. Fowler and Craig Gidney. *Low Overhead Quantum Computation Using Lattice Surgery*. Aug. 2019. arXiv: 1808.06709 [quant-ph]
- [104] Sergey Bravyi and Jeongwan Haah. “Magic-State Distillation with Low Overhead”. In: *Physical Review A* 86.5, 052329 (Nov. 2012).
- [105] Yongshan Ding et al. “Magic-State Functional Units: Mapping and Scheduling Multi-Level Distillation Circuits for Fault-Tolerant Quantum Architectures”. In: *IEEE/ACM Intl. Symp. on Microarchitecture (MICRO)* Oct. 2018, pp. 828–840.
- [106] Daniel Litinski. “Magic State Distillation: Not as Costly as You Think”. In: *Quantum* 3 (Dec. 2019), p. 205.
- [107] Simon J. Devitt, William J. Munro, and Kae Nemoto. “Quantum Error Correction for Beginners”. In: *Reports on Progress in Physics* 76.7 (June 2013), p. 31745.
- [108] E. Knill, R. Laflamme, and W. Zurek. *Threshold Accuracy for Quantum Computation*. Oct. 1996. arXiv: 9610011 [quant-ph]
- [109] Rodney Van Meter et al. “Distributed Quantum Computation Architecture Using Semiconductor Nanophotonics”. In: *International Journal of Quantum Information* 08.01n02 (Feb. 2010), pp. 295–323.
- [110] Craig Gidney and Martin Ekerå. “How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits”. In: *Quantum* 5 (Apr. 2021), p. 433.
- [111] Daniel Gottesman. *An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation*. Apr. 2009. arXiv: 0904.2557 [quant-ph]

- [112] Dave Bacon. “Operator Quantum Error Correcting Subsystems for Self-Correcting Quantum Memories”. In: *Physical Review A* 73.1, 012340 (Jan. 2006).
- [113] Andrew J. Landahl, Jonas T. Anderson, and Patrick R. Rice. *Fault-Tolerant Quantum Computing with Color Codes*. Aug. 2011. arXiv: 1108 . 5738 [quant-ph]
- [114] H. Ollivier and J.-P. Tillich. *Quantum Convolutional Codes: Fundamentals*. Jan. 2004. arXiv: 0401134 [quant-ph]
- [115] Benjamin J. Brown et al. “Poking Holes and Cutting Corners to Achieve Clifford Gates with the Surface Code”. In: *Physical Review X* 7.2, 021029 (May 2017).
- [116] Daniel Litinski. “A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery”. In: *Quantum* 3 (Mar. 2019), p. 128.
- [117] Christopher Chamberland et al. “Topological and Subsystem Codes on Low-Degree Graphs with Flag Qubits”. In: *Physical Review X* 10.1, 011022 (Jan. 2020).
- [118] Adrian Cho. *The Biggest Flipping Challenge in Quantum Computing*. July 9, 2020. URL: <https://www.sciencemag.org/news/2020/07/biggest-flipping-challenge-quantum-computing>. Accessed July 9, 2020.
- [119] C. H. Bennett. “Logical Reversibility of Computation”. In: *IBM Journal of Research and Development* 17.6 (Nov. 1973), pp. 525–532.
- [120] Tommaso Toffoli. *Reversible Computing*. Tech. rep. MIT/LCS/TM-151. Laboratory for Computer Science, Massachusetts Institute of Technology, Feb. 1980.
- [121] Charles H. Bennett. “Time/Space Trade-Offs for Reversible Computation”. In: *SIAM Journal on Computing* 18.4 (Aug. 1989), pp. 766–776.
- [122] Doug McClure and Jay Gambetta. *Quantum Computation Center Opens*. Sept. 19, 2019. URL: <https://www.ibm.com/blogs/research/2019/09/quantum-computation-center/>. Accessed Dec. 29, 2021.
- [123] J. S. Otterbach et al. *Unsupervised Machine Learning on a Hybrid Quantum Computer*. Dec. 2017. arXiv: 1712.05771 [quant-ph]
- [124] Frank Arute et al. “Quantum Supremacy Using a Programmable Superconducting Processor”. In: *Nature* 574.7779 (Oct. 2019), pp. 505–510.
- [125] Steven A. Cuccaro et al. *A New Quantum Ripple-Carry Addition Circuit*. Oct. 2004. arXiv: 0410184 [quant-ph]
- [126] Yasuhiro Takahashi, Seiichiro Tani, and Noboru Kunihiro. *Quantum Addition Circuits and Unbounded Fan-Out*. Oct. 2009. arXiv: 0910.2530 [quant-ph]
- [127] Thomas G. Draper et al. *A Logarithmic-Depth Quantum Carry-Lookahead Adder*. June 2004. arXiv: 0406142 [quant-ph]
- [128] Eric R Johnston, Mercedes Gimeno-Segovia, and Nic Harrigan. *Programming Quantum Computers: Essential Algorithms and Code Samples*. O’Reilly Media, Inc., 2019. ISBN: 978-1-4920-3968-6.
- [129] Craig Gidney. *Quirk: Quantum Circuit Simulator*. URL: <https://algassert.com/quirk>. Accessed Dec. 29, 2021.
- [130] Anupam Chattopadhyay, Nilanjan Pal, and Soumajit Majumder. *Ancilla-Quantum Cost Trade-off during Reversible Logic Synthesis Using Exclusive Sum-of-Products*. May 2014. arXiv: 1405.6073 [cs]

- [131] J.-H. Bae et al. “Quantum Circuit Optimization Using Quantum Karnaugh Map”. In: *Scientific Reports* 10.1 (Sept. 2020), p. 15651.
- [132] Martin Roetteler. “Tools for Quantum and Reversible Circuit Compilation”. In: *Reversible Computation*. Ed. by Iain Phillips and Hafizur Rahaman. Lecture Notes in Computer Science. Springer, 2017, pp. 3–16.
- [133] Marc G. Davis et al. “Towards Optimal Topology Aware Quantum Circuit Synthesis”. In: *IEEE Intl. Conf. on Quantum Computing and Engineering (QCE)* Oct. 2020, pp. 223–234.
- [134] Thomas Fösel et al. *Quantum Circuit Optimization with Deep Reinforcement Learning*. Mar. 2021. arXiv: 2103.07585 [quant-ph]
- [135] Yunseong Nam et al. “Automated Optimization of Large Quantum Circuits with Continuous Parameters”. In: *npj Quantum Information* 4.1 (May 2018), pp. 1–12.
- [136] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. ISBN: 978-0-521-42426-4.
- [137] Thomas Dueholm Hansen et al. “Faster k-SAT Algorithms Using Biased-PPSZ” in: *ACM SIGACT Symposium on Theory of Computing (STOC)* 2019, pp. 578–589.
- [138] David Deutsch. “Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer”. In: *Proc. of the Royal Society of London, Series A, Mathematical and Physical Sciences* 400.1818 (July 1985), pp. 97–117.
- [139] David Deutsch and Richard Jozsa. “Rapid Solution of Problems by Quantum Computation”. In: *Proceedings: Mathematical and Physical Sciences* 439.1907 (Dec. 1992), pp. 553–558.
- [140] Ethan Bernstein and Umesh Vazirani. “Quantum Complexity Theory”. In: *SIAM Journal of Computing* 26.5 (Oct. 1997), pp. 1411–1473. Prelim. version in *STOC* 1993.
- [141] Andrew Chi-Chih Yao. “Quantum Circuit Complexity”. In: *Proc. IEEE Foundations of Computer Science*. Nov. 1993, pp. 352–361.
- [142] John Watrous. “Quantum Computational Complexity”. In: *Computational Complexity: Theory, Techniques, and Applications*. Ed. by Robert A. Meyers. Springer New York, 2012, pp. 2361–2387. ISBN: 978-1-4614-1800-9.
- [143] Scott Aaronson. *When Exactly Do Quantum Computers Provide a Speedup?* Presentation at Yale Quantum Institute Seminar, Yale University. Oct. 10, 2014. URL: <https://scottaaronson.com/talks/speedup.ppt>. Accessed Dec. 29, 2021.
- [144] Lov Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. In: *ACM Symp. on the Theory of Computing (STOC)* May 1996, pp. 212–219. Updated version at arXiv:quant-ph/9605043.
- [145] Christof Zalka. “Grover’s Quantum Searching Algorithm Is Optimal”. In: *Physical Review A* 60.4 (Oct. 1999), pp. 2746–2751.
- [146] Lov Grover. “A Framework for Fast Quantum Mechanical Algorithms”. In: *ACM Symp. on the Theory of Computing (STOC)* 1998, pp. 53–62.
- [147] Gilles Brassard, Peter Høyer, and Alain Tapp. “Quantum Algorithm for the Collision Problem”. In: *ACM SIGACT News* 28 (1998), pp. 14–19.
- [148] H. Ramesh and V. Vinay. “On String Matching in $\tilde{O}(\sqrt{n} + \sqrt{m})$ Quantum Time”. In: *Journal of Discrete Algorithms* 1.1 (2001), pp. 103–110.

- [149] P.W. Shor. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”. In: *IEEE Symposium on Foundations of Computer Science (FOCS)* Nov. 1994, pp. 124–134.
- [150] D. Coppersmith. *An Approximate Fourier Transform Useful for Quantum Factoring*. IBM Research Report RC 19642. July 1994.
- [151] James W. Cooley and John W. Tukey. “An Algorithm for the Machine Calculation of Complex Fourier Series”. In: *Mathematics of Computation* 19.90 (1965), pp. 297–301.
- [152] Daniel E. Browne. “Efficient Classical Simulation of the Quantum Fourier Transform”. In: *New Journal of Physics* 9.146 (2007).
- [153] Danial Dervovic et al. *Quantum Linear Systems Algorithms: A Primer*. 2018. arXiv: 1802.08227 [quant-ph]
- [154] Aram W. Harrow, Avinandan Hassidim, and Seth Lloyd. “Quantum Algorithm for Linear Systems of Equations”. In: *Physical Review Letters* 103.15 (Oct. 2009), p. 150502.
- [155] R. L. Rivest, A. Shamir, and L. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Communications of the ACM* 21.2 (Feb. 1978), pp. 120–126.
- [156] Whitfield Diffie and Martin Hellman. “New Directions in Cryptography”. In: *IEEE Transactions on Information Theory* IT-22.6 (Nov. 1976), pp. 644–654.
- [157] Ralph C. Merkle. “Secure Communications over Insecure Channels”. In: *Communications of the ACM* 21.4 (Apr. 1978), pp. 294–299.
- [158] Eric R. Anschuetz et al. *Variational Quantum Factoring*. 2018. arXiv: 1808.08927 [quant-ph]
- [159] M. Cerezo et al. *Variational Quantum Algorithms*. 2020. arXiv: 2012.09265 [quant-ph]
- [160] Alberto Peruzzo et al. “A Variational Eigenvalue Solver on a Photonic Quantum Processor”. In: *Nature Communications* 5 (2014), p. 4213.
- [161] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411.4028 [quant-ph]
- [162] James C. Spall. “An Overview of the Simultaneous Perturbation Method for Efficient Optimization”. In: *Johns Hopkins APL Technical Digest* 19.4 (1998), pp. 482–492.
- [163] M. Powell. “Direct Search Algorithms for Optimization Calculations”. In: *Acta Numerica* 7 (1998), pp. 287–386.
- [164] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]
- [165] Yudong Cao et al. “Quantum Chemistry in the Age of Quantum Computing”. In: *Chemical Reviews* 119.19 (Aug. 2019), pp. 10856–10915.
- [166] F. Barahona. “On the Computational Complexity of Ising Spin Glass Models”. In: *Journal of Physics A: Mathematical and General* 15.10 (1982), p. 3241.
- [167] L. T. Wille and J. Vennik. “Computational Complexity of the Ground-State Determination of Atomic Cluster”. In: *Journal of Physics A: Mathematical and General* 18.8 (1985), p. L419.
- [168] Panagiotis Kl. Barkoutsos et al. “Quantum Algorithms for Electronic Structure Calculations: Particle-hole Hamiltonian and Optimized Wave-function Expansions”. In: *Physical Review A* 98.2 (Aug. 2018), p. 022322.

- [169] P. J. J. O'Malley et al. "Scalable Quantum Simulation of Molecular Energies". In: *Physical Review X* 6.3 (2016), p. 031007.
- [170] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. "Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms". In: *Advanced Quantum Technologies* 2.12, 1900070 (Oct. 2019).
- [171] Sam McArdle et al. "Quantum Computational Chemistry". In: *Reviews of Modern Physics* 92.1 (Mar. 2020), p. 015003.
- [172] Xiaoyuan Liu et al. *Layer VQE: A Variational Approach for Combinatorial Optimization on Noisy Quantum Computers*. 2021. arXiv: 2102.05566 [quant-ph]
- [173] Samuel Mugel et al. *Dynamic Portfolio Optimization with Real Datasets Using Quantum Processors and Quantum-Inspired Tensor Networks*. 2020. arXiv: 2007.00017 [quant-ph]
- [174] Alba Cervera-Lierta, Jakob S. Kottmann, and Alán Aspuru-Guzik. "Meta-Variational Quantum Eigensolver: Learning Energy Profiles of Parameterized Hamiltonians for Quantum Simulation". In: *PRX Quantum* 2.2 (May 2021).
- [175] Suguru Endo et al. "Variational Quantum Simulation of General Processes". In: *Physical Review Letters* 125.1 (June 2020), p. 4417.
- [176] Andrew Lucas. "Ising Formulations of Many NP Problems". In: *Frontiers in Physics* 2 (Feb. 2014).
- [177] Fred Glover, Gary Kochenberger, and Yu Du. *A Tutorial on Formulating and Using QUBO Models*. 2019. arXiv: 1811.11538 [cs.DS]
- [178] Catherine McGeoch. *Adiabatic Quantum Computing and Quantum Annealing: Theory and Practice*. Morgan and Claypool, 2014. ISBN: 978-1681732121.
- [179] Stuart Hadfield et al. "From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz". In: *Algorithms* 12.2 (Feb. 2019), p. 34.
- [180] C. Kokail et al. "Self-Verifying Variational Quantum Simulation of Lattice Models". In: *Nature* 569 (2019), pp. 355–360.
- [181] Matthew P. Harrigan et al. "Quantum Approximate Optimization of Non-planar Graph Problems on a Planar Superconducting Processor". In: *Nature Physics* 17.3 (Feb. 2021), pp. 332–336.
- [182] Jarrod R. McClean et al. "Barren Plateaus in Quantum Neural Network Training Landscapes". In: *Nature Communications* 9 (2018), p. 4812.
- [183] Xin Wang, Zhixin Song, and Youle Wang. "Variational Quantum Singular Value Decomposition". In: *Quantum* 5 (June 2021), p. 483.
- [184] Jingxiang Wu and Timothy H. Hsieh. "Variational Thermal Quantum Simulation via Thermofield Double States". In: *Physical Review Letters* 123, 220502 (22 Nov. 2019).
- [185] Sumeet Khatri et al. "Quantum-assisted Quantum Compiling". In: *Quantum* 3 (May 2019), p. 140.
- [186] Richard Josza and Noah Linden. "On the Role of Entanglement in Quantum-Computational Speed-up". In: *Proc. of the Royal Society of London* 459 (Aug. 2003), pp. 2011–2032.
- [187] Andrew M. Childs. "Universal Computation by Quantum Walk". In: *Physical Review Letters* 102.18, 180501 (May 2009).

- [188] Steven Jordan. *Quantum Algorithm Zoo*. URL: <https://quantumalgorithmzoo.org/>. Created April 22, 2011. Updated February 1, 2021.
- [189] Ashley Montanaro. “Quantum Algorithms: An Overview”. In: *npj Quantum Information* 2.1 (Jan. 2016), p. 15023.

Index

a

ABCD matrix 81, 113, 123
 Abelian group 239
 acoustic modes 167
 added-noise number 99
 adder, classical 272
 adder, quantum
 carry-lookahead 277
 design tradeoffs 281
 MAJ/UMA functions 275
 ripple-carry 273, 275
 adiabatic quantum
 computing 320
 adjoint 5
 amplifier
 HEMT 21, 104
 parametric 105, 208
 amplitude amplification 299
 ancilla bit 266
 ancilla qubit 17, 269, 270, 292,
 296
 angular momentum
 orbital 34
 spin 34
 anharmonicity 207, 210
 annihilation operator 140, 160,
 181
 ansatz 315, 317, 322
 expressibility 318

anti-commuting operators
 154, 239

attenuation per unit length 69
 attenuator 20, 85, 101
 AXBXC decomposition 57

b

barren plateau 324
 BCS Hamiltonian 182
 Bell state 12, 269
 big-O notation 291
 bit 1
 bit flip error 227, 231, 232, 235,
 238, 248
 black body radiation 97
 Bloch sphere 30, 31, 44, 46
 Bloch Theorem 162
 Boolean logic 264
 Born rule 5
 Bose-Einstein distribution 97,
 160
 Bosons 160
 boundary conditions 198, 199
 periodic 163, 167
BPP complexity class 293
BQP complexity class 293
 bra-ket notation 5
 branch flux 140
 branch voltage 140
 Brillouin zone 166

c

- canonical variables 130
- capacitance
 - average stored energy 108
 - equivalent for transmission line resonator 112, 143
 - junction 203, 210
 - per unit length 63, 141
 - stored energy 140
- carrier frequency 93
- cascaded two-port networks 84
- Central Limit Theorem 45
- channel
 - classical 227
 - quantum 230
- characteristic impedance 66, 68, 70
- charging energy 205
- chemical potential 160
- circuit quantum
 - electrodynamics 138
- circulator 19, 87, 89, 209
- classical optimizer 316
- Clifford gates 216, 243
- CNOT gate 11
 - reverse 271
- coaxial line 78
- coherence length 185
- combinatorial
 - optimization 320
- commutation relations 28, 133, 154, 239
- commuting operators 239
- compiler 272
- computational basis 1, 264, 269
- computational complexity 291
 - BPP** 293
 - BQP** 293
 - classes 292
 - NP** 292
 - NP-complete** 293
 - NP-hard** 293
 - P** 292
 - quantum 293
 - query 292
- concatenated codes 243
- conductance per unit length 63
- conformal mapping 79
- conjugate variables 133, 139
- conservation of probability 37, 199
- constitutive relations 187
- continued fraction
 - expansion 313
- control qubit 57
- controlled-NOT gate 11
- controlled-phase (CP) gate 283
- controlled- U gate 57
- controlled- Z (CZ) gate 235, 283
- Cooper pairs 182, 185, 192
- coplanar waveguide 79
- cost function 315
- Coulomb gauge 187
- coupled qubits
 - fixed-frequency 55
 - tunable 53
- coupled swings 120
- coupled-mode equations 41, 119, 150
- coupling coefficient 114
- coupling field 36
- CPMG experiment 213
- creation operator 140, 160, 181
- critical current 202
- critical temperature 159
- cross-resonance 55
- crystal lattice 161
- crystal momentum 163
- CSS codes 240
- cubic lattice 164
- cutoff frequency, filter 96

d

- de Broglie wave length 163
- Debye frequency 168, 174
- Debye model 167
- Debye temperature 168
- Debye wave number 167
- decision problem 292
- decoherence 212

Deep Reinforcement Learning
 (DRL) 223
 density matrix 45, 59
 density of states 165, 175
 Deutsch algorithm 15
 dielectric function 175
 dilution refrigerator 19, 99
 diode
 current-voltage relation 92
 ideality factor 92
 saturation current 92
 Discrete Fourier Transform
 (DFT) 300
 discrete logarithm 309
 dispersion relation 70
 dispersive coupling 123, 153
 distillation 259
 divergence theorem 190
 DiVincenzo requirements 25
 DRAG pulse 219
 driving field 36

e
 effective permittivity 77
 eigenphase 307, 312
 eigenstate 307, 312, 316
 eigenvalue 34, 316
 eigenvector 34
 electric scalar potential 186
 electron spin 27
 electron-electron
 interaction 180
 energy gap 184
 ensemble of pure states 46
 entangled state 3, 12, 46, 51, 52,
 263, 269, 324
 error correction
 decoder 227
 encoder 227
 error correction code
 (ECC) 228
 distance 230
 Hamming 229
 repetition 228
 error detection code (EDC) 228

 parity 228
 error mitigation 211
 error operator 232
 errors per Clifford (EPC) 216
 errors per gate (EPG) 217
 Euler angle 32
 Euler-Lagrange equation 128,
 142, 204
 expectation 27, 35, 44, 317
 exponential of a matrix 29

f

Faraday's law 188
 Fast Fourier Transform
 (FFT) 306
 fault-tolerant quantum
 computing 227, 245
 Fermi energy 160, 165
 Fermi surface 166, 169
 Fermi velocity 166, 180
 Fermi-Dirac distribution 160
 Fermions 159
 filter
 high-pass 93
 low-pass 93
 flux quantum 192, 202, 208
 Fourier basis 301
 free electron model 163
 functional 128

g

gate errors 215
 Gauss' law 176
 global phase 31
 gradient descent 316
 ground state 7, 19
 ground state energy 316
 Grover's algorithm 294
 gyromagnetic ratio 34

h

Hadamard gate 6, 43
 Hamilton's principle 127
 Hamiltonian 33, 130, 221
 Hamiltonian mechanics 130

Hamming code 229
 harmonic oscillator
 classical 131
 quantum 133
 Heisenberg picture 147
 Helmholtz theorem 187
 Hermite polynomial 134
 recurrence relations 134
 Hermitian conjugate 5
 Hermitian operator 27
 High Electron Mobility
 Transistor (HEMT) 21
 high-pass filter 93
 Hilbert space 232
 hybrid classical-quantum
 algorithm 309, 315, 323

i

I-Q demodulator 95
 I-Q modulator 21
 I-Q plane 95
 impedance
 transmission line 66,
 68, 70
 inductance
 equivalent for transmission
 line resonator 112, 143
 kinetic 80, 171, 203
 non-linear 203
 per unit length 63, 142
 stored energy 140
 inner product 5
 integer factoring 309
 Interaction picture 147
 interleaved randomized
 benchmarking (IRB) 218
 Ising model 321
 isolator 20, 88
 iSWAP gate 52

j

Jaynes-Cummings
 Hamiltonian 152
 jellium 176
 Josephson energy 205

Josephson junction 195, 200
 Hamiltonian 203
 inductance 203
 Josephson Junction Parametric
 Amplifier (JPA) 20

k

ket 5
 Kirchoff's current law 65, 117,
 204
 Kirchoff's voltage law 63

l

Lagrangian 127
 Lagrangian mechanics 127
 Larmor precession 35
 lattice surgery 253
 CNOT 257
 logical operator 258
 merge 253
 split 255
 leakage 219
 least squares method 215
 local oscillator 93
 local phase 31
 logical operator 242
 logical qubit 231
 London equation 188, 189
 London gauge 189
 London penetration depth 207
 longitudinal phonon mode 167
 Lorentz gauge 187
 low-pass filter 93
 lowering operator 135, 205

m

magic state 243, 259
 magic state distillation 259
 magnetic moment 33, 34
 magnetic vector potential 186
 MAJ function 229
 MaxCut 321
 measurement error 214
 calibration matrix 214
 microstrip line 78

mixed state 46, 59
 mixer 21, 92, 219
 mixing operator 320
 modular exponentiation 309,
 312, 314
 modular multiplication 310

n

NbTi 21
 Newton's law 166
 niobium 117
 NISQ 211, 272
 algorithms 314
 no-cloning theorem 3, 13, 269
 node flux 140, 204
 node voltage 140
 noise
 cascaded systems 103
 factor 101
 figure 101
 mitigation 211
 number 99, 105
 power spectral density 99
 quantum 211, 212
 temperature 99
 thermal 97
 non-reciprocal devices 87
 Norton equivalent 110
NP complexity class 292
NP-complete complexity
 class 293
NP-hard complexity class 293

o

objective function 321
 occupation number
 representation 205
 optical modes 167
 oracle 292, 294

p

P complexity class 292
P gate 283
 parameterized quantum
 circuit 315

parametric amplifier 105
 partial trace 60
 passive rotation 38
 Pauli errors 231, 236
 Pauli exclusion principle 160,
 164
 Pauli gates 6
 Pauli matrices 28, 34
 perfect conductor 170
 period finding 309
 permittivity 175
 phase flip error 231, 234, 235
 phase kickback 16, 285, 296,
 308
 phase logic 283, 296, 297, 304
 gates 287
 selective phase 285
 phonon 167
 acoustic 167
 longitudinal 167
 optical 167
 transverse 167
 Planck radiation law 97
 post-quantum
 cryptography 309
 power divider 89
 precession 35, 88
 primitive cell 162
 projective measurement 238
 propagation constant 68
 public-key cryptography 309
 pulse, baseband 93
 pure state 45, 59

q

Qiskit 13, 33, 213
 Quadratic Unconstrained Binary
 Optimization
 (QUBO) 322
 quadrature hybrid 89
 quantization of errors 231
 quantum advantage 291
 Quantum Approximate
 Optimization Algorithm
 (QAOA) 315, 320

- quantum circuit 7, 263
 - quantum circuit depth 213, 270, 273, 292
 - quantum circuit width 270, 292
 - quantum compiler 272
 - quantum computing system 18
 - quantum error correction (QEC)
 - distance 232, 246
 - Quantum Fourier Transform (QFT) 299, 310
 - approximate 306
 - inverse 308, 312
 - quantum gate
 - matrix 6, 9
 - multi-qubit 270
 - single-qubit 6, 28, 283
 - two-qubit 9, 283
 - quantum interference 16
 - quantum measurement 1, 5, 8, 42, 319
 - projective 238
 - quantum noise 211
 - quantum optimal control 223
 - quantum parallelism 15, 288, 294, 312, 324
 - Quantum Phase Estimation (QPE) 307, 312
 - precision 308
 - superposition 308
 - quantum process
 - tomography 216
 - quantum search 294
 - quantum state
 - Bloch sphere 30
 - density matrix 45
 - disc notation 284
 - mixed 46, 59
 - phase 31
 - pure 45, 59
 - state vector 4
 - quantum state tomography 42, 48, 224
 - quantum system topology 270, 281
 - quantum teleportation 3
 - quantum-encoded signal 304, 308
 - quarter wave transformer 76
 - quasi-particle 184
 - qubit 1
 - qubit readout 120
 - query complexity 292
- r**
- Rabi oscillations 36
 - raising operator 136, 205
 - randomized benchmarking (RB) 216, 222
 - Rayleigh-Jeans
 - approximation 98
 - reciprocity 81
 - reflection coefficient 71
 - repetition code 228
 - resistance
 - equivalent for transmission
 - line resonator 112
 - per unit length 63
 - resonant frequency 39, 108, 111, 118, 123
 - resonator
 - bandwidth 109
 - critically-coupled 111
 - impedance 107, 111
 - lumped element 107
 - over-coupled 111
 - power dissipated 109
 - Q, 108
 - transmission line 111
 - under-coupled 111
 - return loss 73
 - reversible logic 3, 263, 264
 - classical 264
 - implementation 267
 - in-place 265
 - Toffoli gate 265
 - uncomputing 266, 268, 269
 - Richardson extrapolation 220
 - rotating frame 38, 40
 - rotating wave
 - approximation 39, 149

rotation matrix 26, 30
 rotations 25, 32
 rough (Z) boundary 247
 rough merge 254
 rough split 257

s

S gate 44, 283
 S parameters 80
 Schrödinger equation 33, 133, 163, 172
 Schrödinger picture 147
 Schrieffer-Wolff transformation 153
 screening 175
 second quantization 205
 Shor's 9-qubit code 235, 240
 Shor's algorithm 309
 silicon substrate 79, 116
 smooth (X) boundary 247
 smooth merge 255
 smooth split 255
 spin 26, 27, 34
 spinor 26, 133, 173
 SQUID 207
 stabilizer 238, 239, 246
 standard basis 1
 standing wave pattern 74, 198
 standing wave ratio (SWR) 73
 state leakage 219
 state preparation and measurement (SPAM) errors 214
 state vector 4, 8, 25, 27, 38
 stationary action principle 127
 Steane code 240
 Stokes' theorem 192, 208
 stretch factor 221
 superposition 1, 11, 48, 263, 269
 surface code 245
 braiding 253
 cycle 247
 distance 247
 error correction 247

lattice surgery 253
 logical operator 250
 quiescent state 247
 SWAP gate 51, 271, 277
 SWR 73
 syndrome 229, 233, 240

t

T gate 283
 T_1 relaxation time 212
 T_2 dephasing time 212
 target qubit 57
 tensor product 8
 Thevenin equivalent 109
 threshold theorem 244, 246, 260
 time evolution 56
 Toffoli gate 265
 implementation 270
 multi-controlled (MCT) 266, 270
 topological codes 245
 topology 270
 toric code 245
 trace 46
 transmission lines 63
 attenuation 69
 characteristic impedance 66, 68, 70
 co-planar 77
 coaxial 77
 ladder line 63
 lossless 65
 microstrip 77
 phase velocity 69
 reflection coefficient 71
 terminated 71
 transmon 120, 206
 transversal 242, 253, 259
 transverse phonon mode 167
 tunneling 195

u

u1, u2, u3 gates 32
 uncertainty 48

unitary 6, 27, 47, 81
 unitary transformation 56
 unwrapped phase 124

v

Variational Quantum Algorithm
 (VQA) 314
 Variational Quantum
 Eigensolver (VQE) 223,
 315, 316

vector operator 27
 velocity
 drift 169
 phase 66, 69
 voltage standing wave ratio
 (VSWR) 73

w

Walsh-Hadamard
 transform 298
 wave equation 65
 wave number 69

wavelength 69
 waves 69
 white noise 98
 Wilkinson combiner/
 divider 21, 89
 work function 195

x

X (NOT) gate 1, 6
 X boundary 247

y

Y gate 6

z

Z boundary 247
 Z gate 6, 283
 z rotation 40
 Zeeman energy 34, 133
 zero-noise extrapolation 220
 zero-point fluctuations 99
 ZY Z convention 32
 ZY Z decomposition 57

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook
EULA.