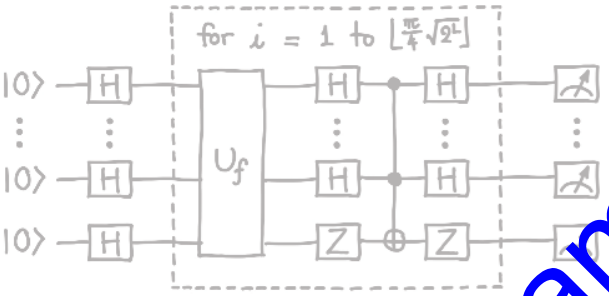
THAMMASAT UNIVERSITY PRESS

QUANTUM COMPUTING

A GENTLE INTRODUCTION FOR AI ENGINEERS



example

PRACHYA BOONKWAN

Sirindhorn International Institute of Technology,
Thammasat University, Thailand

Forewords

example

In modern technology, artificial intelligence (AI) has advanced from theoretical algorithms into practical systems that process complex multimodal data, model natural language, and evaluate high-dimensional visual inputs. As deep learning models scale, traditional silicon-based architectures face severe physical and thermodynamic constraints, including heat dissipation limits, interconnect latency, and the memory wall. To overcome these bottlenecks, the technology industry is pursuing specialized solutions. While novel silicon architectures, chiplet designs, and photonic computing address immediate AI scaling demands, quantum computing is emerging in parallel to tackle fundamentally intractable optimization and simulation problems that lie beyond classical reach.

However, bridging the gap between classical computer science and quantum information science remains a significant hurdle for engineers and researchers. The challenge lies less in mastering abstract physics and more in translating familiar algorithmic intuition into linear algebra over complex vector spaces, Hilbert spaces, and non-intuitive principles like superposition and entanglement. *Quantum Computing: A Gentle Introduction for AI Engineers*, written by our esteemed faculty member Dr. Prachya Boonkwan, serves as a vital and timely bridge across this disciplinary divide.

Dr. Boonkwan's inspiration for this text stemmed from his own journey into quantum computing. Recognizing that existing literature was often either mathematically fragmented or lacked intuitive, code-level metaphors, he set out to build a gentle, hands-on entry point that ensures a smoother learning curve for prospective students and practitioners.

Written specifically from the perspective of an AI practitioner, this book assumes only a foundational knowledge of linear algebra, probability theory, and

basic machine learning concepts. Meticulously organized into four parts across sixteen chapters, it guides the reader systematically from mathematical fundamentals to applied quantum machine learning:

- **Part I: Mathematical Foundations** demystifies the essential mathematics of complex numbers, vectors, matrices, and Hilbert spaces, providing a solid mathematical language for quantum states.
- **Part II: Quantum Logic and Circuits** equips readers with the practical tools to design and simulate quantum circuits using the PennyLane Python library, utilizing both Clifford and non-Clifford gate sets.
- **Part III: Quantum Algorithms and Applications** covers core quantum algorithms such as Grover's search, Quantum Fourier Transform (QFT), and Quantum Phase Estimation (QPE), alongside secure communication protocols like quantum teleportation and key distribution.
- **Part IV: Quantum AI and Optimization** bridges quantum computing directly with artificial intelligence, exploring variational quantum algorithms, Quantum Support Vector Machines (QSVM), Quantum Neural Networks (QNN), and the Quantum Approximate Optimization Algorithm (QAOA).

This curriculum is designed to future-proof the skills of AI engineers, empower researchers exploring quantum-classical hybrids, and offer undergraduate students a structured entry point into advanced computational paradigms.

As the Director of Sirindhorn International Institute of Technology (SIIT), I am confident that this textbook will serve as an indispensable resource for anyone seeking to master quantum principles and shape the future of intelligent systems.

example

Assoc. Prof. Kriengsak Panuwatwanich, PhD
Director
Sirindhorn International Institute of Technology (SIIT)
Thammasat University, Thailand

Preface

example

Artificial intelligence plays a crucial role in modern-day technology. It has transformed from theoretical frameworks into practical systems that understand complex multimodal data, process natural language, analyze images and videos, and make predictions from underlying patterns with deep learning and the Transformer models. As AI engineers like us endeavor to optimize massive neural networks, search model architectures, and leverage pretrained models in every feasible way, we also approach the physical limits of conventional silicon-based hardware. Meanwhile, a next-wave paradigm is silently maturing: Quantum Computing. For many working on machine learning, quantum mechanics is a major hindrance to technology adoption due to seemingly dense and difficult physics equipped with complex algebra.

In spite of the difficulty, I eagerly accepted this challenge two years ago, and I must admit it was quite a rough journey. First, since quantum computing is a niche research area when compared to AI, there are much less programming-based textbooks dedicated to the field. Second, the majority of these books assume the readers to have some background knowledge in quantum mechanics, which seems impenetrable in the eyes of AI engineers. Third, although online resources for quantum computing are available and free of charge, they are scattered across the Internet and inconsistent in terms of mathematical notations. Fourth, even with a handful of high-quality resources, we still need to revise some mathematical background, particularly on complex algebra, which is an extensive principle not frequently used in standard AI applications. Fifth and finally, in spite of adequate backgrounds, some quantum algorithms are notoriously counter-intuitive and they lack self-explanatory metaphors. At one point I simply realized: it does *not* have to be this tough for any prospective student!

The book "*Quantum Computing: A Gentle Introduction for AI Engineers*" is written specifically for the following target groups:

- **AI engineers:** Looking to future-proof their skill sets and understand how quantum circuits operate compared to classical neural networks.
- **AI researchers:** Exploring the theoretical overlaps between quantum algorithms, model interpretability, and machine learning architectures.
- **Undergraduates:** Transitioning into advanced computational paradigms and seeking a clear, intuitive entry point.

I wrote this book from the perspective of an AI engineer who is somewhat familiar with linear algebra, the probability theory, and various machine learning techniques, and I divided it into four parts. In the first part, I will motivate the readers with some basic ideas of quantum computing and then lay mathematical foundations of complex numbers, vectors, and matrices as well as their metaphors. In the second part, I will introduce two sets of quantum gates and output measurement, as well as how to assemble them into a quantum algorithm implementable in Python. In the third part, I will introduce a wide array of quantum algorithms and their applications using self-explanatory metaphors. Finally, in the fourth part, I will emphasize how to leverage quantum computing in the field of AI, ranging from support vector machines, neural networks, to optimization algorithms. Notwithstanding, this book is limited to the field of AI and it may not cover all aspects of quantum computing and relevant topics.

Of course, transitioning from classical to quantum AI may require some efforts, but all mathematical and algorithmic foundations are definitely within your capability. I hope that this book helps demystify the knowledge of quantum computing for you, so that you can focus on what you do best: developing AI systems.

example

Per aspera ad astrà



Prachya Boonkwan



example

Pandóra mala mundi liberávit,
sed spem clausam tenuit in pyxide.
Physica quantica autem docet
spem ipsam simul esse et vivam et mortuam.

Pandora released the evils of the World but kept hope locked inside the box. But quantum physics teaches us that hope itself is simultaneously alive and dead.[1]

Author:

Prachya Boonkwan

Advisory committee:

Akkharawoot Takhom, Apivadee Piyatumrong, and Jessada Karnjana

Proofreader:

Dhanon Leenoi

Copyright © 2026 by Prachya Boonkwan

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, without the prior written permission of the publisher. Exceptions are granted for photocopying of designated student worksheets, which may be reproduced for classroom use only. Equations and illustrations featured on the book's cover are drawn by the author. Some pictures and icons are taken from Wikipedia and `www.flaticon.com`, respectively, and their attributions are provided therein.

example

Disclaimer:

The information in this textbook is intended for educational and instructional purposes only. The author and editors have made every effort to ensure the accuracy of the information; however, no guarantee is made or implied regarding the correctness or completeness of the content. Generative AI is used very sparingly only for the purposes of grammar checking, paraphrasing, and fact checking.

Acknowledgement:

I am greatly indebted to the following people. To my mom and dad, for your never-ending unconditional love and support through good and bad times. To Prof. Thanarak Theeramunkong, Dr. Thepchai Supnithi, Dr. Nongnuch Ketui, and colleagues at the Artificial Intelligence Association Thailand, for your guidance, patience, and support. To Dr. Waree Kongprawechnon, Dr. Sasiporn Usanavasin, Dr. Ekawit Nantajeewarawat, and Dr. Rachada Kongkachandra, for your guidance and mental support. Most importantly, to Prof. Prabhas Chongstitvatana, Prof. Pairash Thajchayapong, and Dr. Sarun Sumriddetchkajorn, for your immense inspiration in quantum computing.

Financial support:

Publication of this textbook is financially supported under the Thammasat University's research fund, Center of Excellence in Intelligent Informatics, Speech and Language Technology and Service Innovation (CILS), and Intelligent Informatics and Service Innovation (IISI) Research Center, the Thailand Research Fund under grant number RTA6080013.

Contents

example

I Mathematical Foundations **1**

1 Introduction **3**

1.1 Classical vs. Quantum Computing . . . **5**
1.2 Foundations of Classical Computing . . . **7**
Bitstring . . . 7
Logical operation . . . 7
1.3 Fundamentals of Quantum Computing . . . **8**
Superposition . . . 8
Entanglement . . . 9
Interference . . . 10
1.4 Quantum-Suitable Applications . . . **11**
Quantum class of problems . . . 11
Milestones of quantum computing . . . 12
Disclaimer . . . 13
1.5 Outline of This Book . . . **13**

2 Complex Mathematics **17**

2.1 Complex Numbers . . . **19**
Definition . . . 19
Wave representation . . . 20
2.2 Complex Algebra . . . **21**
Addition and subtraction . . . 22
Multiplication . . . 22
Conjugate . . . 22
Reciprocal . . . 22

2.3 Trigonometry and Complex Numbers . . . **23**
Polar form . . . 23
Polarity . . . 23
Trigonometric values . . . 23
Relationship between sine and cosine . . . 23
Sum and difference of phases . . . 24
2.4 Complex Vectors . . . **25**
Wave multiplex . . . 25
Column and row vectors . . . 25
Inner product . . . 26
Norm . . . 27
Distance and Hilbert space . . . 27
Phase synchronicity and similarity . . . 27

3 Complex Matrices **31**

3.1 Complex Algebra . . . **34**
Series of wave multiplexes . . . 34
Adjoint . . . 34
Hermitian matrix . . . 35
Affine transformation of vector . . . 36
Affine transformation of matrix . . . 36
3.2 Determinant . . . **37**
Scaling factor . . . 37
Special properties of determinants . . . 37
3.3 Projector Matrix . . . **38**
Outer product . . . 38
Orthogonal projector . . . 39
3.4 Unitary Operators . . . **39**
Orthonormal matrix . . . 39
Unitary operator . . . 40
3.5 Eigen-Decomposition . . . **41**
Dimensionality reduction . . . 41
Trace . . . 42
Analytical solution . . . 42
Factorization of unitary matrices . . . 44
Singular value decomposition . . . 45
3.6 Non-Stretching Basis Rotation . . . **48**
Matrix exponentials . . . 48
Basis rotation . . . 48
3.7 Measurement and Reflection . . . **49**
Projection and rejection . . . 49
Reflection . . . 50
Quantum speedup with reflection . . . 51

4 Quantum Logic Gates **59**

example

4.1 Quantum Information **60**
Qubit 60
Dirac's notation 61
Multi-qubit states 61
4.2 Quantum Operators **62**
State manipulation 62
Quantum operators 62
Construction of a quantum operator 64
4.3 Universal Rotation **65**
Bloch sphere 65
Spinor 67
Universal rotation 68
Conversion to universal rotation 69
4.4 Multi-Qubit States **71**
Qubit string 71
Multi-qubit state 72
4.5 Tensor Product **72**
Concatenation of qubit strings 72
Factorization of multi-qubit states 73
Factorization formulae 74
4.6 Quantum Entanglement **75**
Entangled states 75
Types of quantum entanglement 76
Quantum robustness 76
4.7 Multi-Qubit Quantum Operators **78**
Quantum operators 78
Design of quantum operators 79
4.8 Lossy State Compression **79**
Data marshaling 79
Tensor networks 80

II Quantum Logic and Circuits 91

5 Clifford Gates 93

5.1 Quantum Algorithms and Implementation **94**
Quantum algorithms 94
Software library 95
Installation 95
Example code 96
Code skeleton 96
Workaround for Apple Silicon 97
5.2 Clifford Set 1 . . . . . **98**
Simulatable gates 98
Stabilizer circuit 98

example

Levels of gates . . . 99
NOT gate . . . 99
Hadamard gate . . . 100
Phase shift gate . . . 101
Controlled NOT . . . 102
Special unitary groups . . . 103
Reading a quantum circuit . . . 103
Implementation in PennyLane . . . 104
Exponential parallelization . . . 105
5.3 Superscalar Processing . . . 107
Tensor product of unitary operators . . . 107
Tensor product of quantum logic gates . . . 108
Operator augmentation . . . 110
Superscalar processing . . . 111
Data dependency . . . 112
Preparing the GHZ state . . . 114
5.4 Clifford Set 2 . . . 114
Pauli gates . . . 114
Pauli-X gate . . . 115
Pauli-Z gate . . . 115
Pauli-Y gate . . . 116
Pauli gates in quantum circuits . . . 117
Operator augmentation with qubit swapping . . . 118
5.5 Clifford Set 3 . . . 122
SWAP gate . . . 122
Controlled Z gate . . . 123
Control qubit . . . 123
Circuit visualization . . . 125

6 Non-Clifford Gates 131

6.1 Quantum Tomography . . . 132
Gottesman-Knill Theorem . . . 132
Quantum tomography . . . 133
6.2 Multi-Qubit Control Gates . . . 136
Clifford test . . . 136
Non-Clifford set . . . 137
Toffoli gate . . . 138
Boolean universality . . . 139
Generalized controlled gate . . . 140
6.3 Generalized Rotation Gates . . . 142
Generalized phase shift gate . . . 142
T gate . . . 143
Toffoli gate and T gate . . . 143
Generalized rotation gates . . . 144
Generalized Pauli gates . . . 145

example

Universality of non-Clifford gates . . . 146
Faulty rotation gates . . . 147
6.4 Quantum State Preparation . . . 147
Problem statement . . . 147
Preparation of uniform quantum states . . . 147
Preparation of arbitrary quantum states . . . 153

7 Computing with Light 163

7.1 Photonic Quantum Computing . . . 164
KLM Protocol . . . 164
Dual-rail encoding . . . 165
Optical elements . . . 165
Mach-Zehnder interferometer . . . 168
Quantum logic gates . . . 169
Manufacturable platform for PQC . . . 170
Quantum algorithm as a statistical process . . . 171
7.2 State Measurement . . . 173
Analytical solution . . . 173
Information extraction . . . 174
State expectation . . . 175
Variance . . . 178
Simulation . . . 179
7.3 Quantum Decoherence . . . 181
Density matrix . . . 181
Purity . . . 182
Expectation of a mixed state . . . 182
7.4 Dynamic Circuits . . . 184
Mid-circuit measurement . . . 184
Conditional branching . . . 184
Post-selection . . . 185
Looping . . . 187

8 Basic Core Algorithms 195

8.1 Bernstein-Vazirani Algorithm . . . 196
Oracle separation . . . 196
Quantum parallelism . . . 197
Oracle recovery . . . 198
Quantum solution . . . 198
Phase kickback technique . . . 199
Oracle extraction . . . 201
Implementation . . . 203
8.2 Deutsch-Jozsa Algorithm . . . 204
Hash function . . . 204
Balance testing . . . 205
Quantum solution . . . 205

example

Implementation . . . 206
8.3 Quantum Teleportation . . . 208
Secure communication . . . 208
Teleportation of one qubit . . . 210
Teleportation of two entangled qubits . . . 213
8.4 Quantum Key Distribution . . . 216
Key distribution . . . 216
Quantum alternative . . . 217
Implementation . . . 219
Security concerns . . . 221
8.5 Superdense Coding . . . 222
Data compression . . . 222
Teleportation with compression . . . 222

III Quantum Algorithms and Applications 231

9 Quantum Noisy Channels 233

9.1 Quantum Noisy Channels . . . 234
Density matrix of single-qubit states . . . 234
Density matrix of multi-qubit states . . . 235
Common types of decoherence . . . 236
State purification . . . 238
Fidelity . . . 239
9.2 Error Correction . . . 243
Classical error correction . . . 243
Hamming code . . . 243
Quantum error correction . . . 246
Bias quantum code . . . 246
Shor code . . . 250
Surface code . . . 252
NISQ computers . . . 254

10 Advanced Core Algorithms 259

10.1 Grover's Search Algorithm . . . 260
Information retrieval . . . 260
Quantum information retrieval . . . 261
Amplitude amplification . . . 262
Oracle function . . . 262
Mean shifting by diffusion . . . 264
Optimal number of iterations . . . 266
Actual implementation . . . 267
Searching with multiple solutions . . . 268
10.2 Quantum Fourier Transform . . . 270
Spectral density . . . 270

example

Classical Fourier transform . . . 271
Fast Fourier transform . . . 273
Quantum Fourier transform . . . 274
Inverse quantum Fourier transform . . . 278
Comparison between DFT and QFT . . . 281
10.3 Quantum Phase Estimation . . . 283
Intuition . . . 283
Phase estimation by rotation . . . 283
Uniform angular rotations . . . 284
Phase estimation . . . 286
Implementation . . . 287

11 Quantum Cryptography 293

11.1 Classical Cryptography . . . 295
Cryptosystem . . . 295
Prime factorization . . . 296
Coprimality . . . 298
Totient functions . . . 300
11.2 RSA Cryptosystem . . . 303
Fermat-Euler theorem . . . 303
Asymmetric cryptography . . . 304
Rivest-Shamir-Adleman cryptosystem . . . 305
Key generation . . . 305
Proof of correctness . . . 307
11.3 Shor's Algorithm . . . 308
Factorization as weakness . . . 308
Hybrid algorithm . . . 309
Continued fraction method . . . 312
Implementation . . . 313
11.4 Post-Quantum Cryptography . . . 316
Quantum resistance . . . 316
AES-256 Cryptosystem . . . 316
11.5 Simon's Algorithm . . . 318
Hash function . . . 318
Simon's algorithm . . . 319
Implementation . . . 321
Finding the solution . . . 321

12 Quantum Simulation 327

12.1 Quantum Finite-State Automata . . . 328
Systems dynamics . . . 328
Deterministic finite-state automaton . . . 329
Probabilistic finite-state automaton . . . 331
Quantum finite-state automaton . . . 333
Szegedy's algorithm . . . 334

example

Implementation . . . 337
12.2 Quantum Speedup in Stochastic Simulation . . . 339
Hamiltonian matrix . . . 339
Conversion to Hamiltonian . . . 340
Commutator . . . 342
Trotter-Suzuki decomposition . . . 344
Qubitization algorithm . . . 346
12.3 Simulating Non-Linear Functions . . . 351
Chaos simulation . . . 351
Quantum signal processing . . . 352
Generalized quantum signal processing . . . 353
Quantum singular value transformation . . . 355

IV Quantum AI and Optimization 361

13 Variational Quantum Algorithms 363

13.1 Parameterized Quantum Circuits . . . 365
Variational quantum algorithm . . . 365
Quantum embedding . . . 366
Ansatz . . . 367
Entangler layers . . . 368
Expressivity and trainability . . . 371
Ansatz architectures . . . 371
13.2 Classical Parameter Estimation . . . 373
Prediction loss . . . 373
Parameter optimization . . . 373
Projection-based loss function . . . 374
Classical ML models . . . 376
Popular loss functions . . . 378
13.3 Gradient-Based Parameter Optimization . . . 380
Hermitian generator . . . 380
Parameter-shift rule . . . 381
Implementation . . . 383
Limitations . . . 384
13.4 Gradient-Free Parameter Optimization . . . 385
Bayesian inference . . . 385
Nakanishi-Fujii-Todo algorithm . . . 388
Implementation with Rotoselect . . . 390

14 Quantum Machine Learning 395

14.1 Quantum Kernel Methods . . . 396
Class separation problem . . . 396
Support vector machines . . . 397
Quantum SVM . . . 398

example

Implementation . . . 401
14.2 Quantum Neural Networks . . . 403
Perceptrons . . . 403
Multilayer perceptrons . . . 405
Quantum deep neural networks . . . 409
Training quantum DNNs . . . 410
Implementation . . . 411
14.3 Quantum Convolutional Neural Networks . . . 414
Convolution . . . 414
Signal similarity . . . 414
Pattern matching . . . 415
Classical CNNs . . . 415
Quantum CNNs . . . 417
14.4 Quantum Natural Language Processing . . . 418
Quantum word embedding . . . 418
Sequential modeling . . . 420
Structure prediction . . . 421
14.5 Challenges of Quantum Deep Learning . . . 425
Gradient divergence . . . 425
Barren plateau . . . 427
Scalability . . . 427

15 Quantum Optimization 431

15.1 Classical Simulated Annealing . . . 432
Optimization problem . . . 432
Simulated annealing . . . 433
Implementation . . . 434
15.2 Variational Quantum Eigensolver . . . 439
Coulomb matrix . . . 439
Ground state . . . 440
Variational quantum eigensolver . . . 442
15.3 Adiabatic quantum computing . . . 445
Adiabatic theorem . . . 445
Adiabatic loop . . . 446
Procedure . . . 447
Implementation . . . 449
15.4 Optimization on Graph Structures . . . 450
QUBO problems . . . 450
Quantum QUBO . . . 453

16 Quantum Approximate Optimization Algorithm 461

16.1 Problem Formulation . . . 462
Quantum annealing . . . 462
Approximate annealing with QAOA . . . 465
Design patterns of QAOA . . . 467

example

16.2 Notable QUBO Problems **470**
Max-cut problem 470
Max-clique problem 473
Max-weight cycle problem 474
Max-independent set problem 476
Min-vertex cover problem 478
Traveling salesman problem 479

V Appendices 483

A List of Symbols **485**

A.1 Quantum Information **485**
Logical operations 485
Complex numbers 485
Complex vectors 486
Complex matrix 486
Quantum logic gates 487
A.2 Quantum Circuits **487**
Clifford set 487
Non-Clifford set 488
Photonic quantum computing 488
State measurement 489
Quantum decoherence 489
A.3 Foundational Algorithms **489**
Leveraging superposition and entanglement 489
Quantum noisy channel 489
Grover's search algorithm 490
Quantum Fourier transform 490
Quantum cryptography 491
Quantum simulation 491
A.4 Applications in AI and Optimization **492**
Variational quantum algorithms 492
Quantum machine learning 493
Quantum optimization 493
QAOA 494

B Glossary **495**

C Acronyms **525**

D Pedagogical Materials **529**

D.1 Short Course Description **529**
D.2 Long Course Description **529**

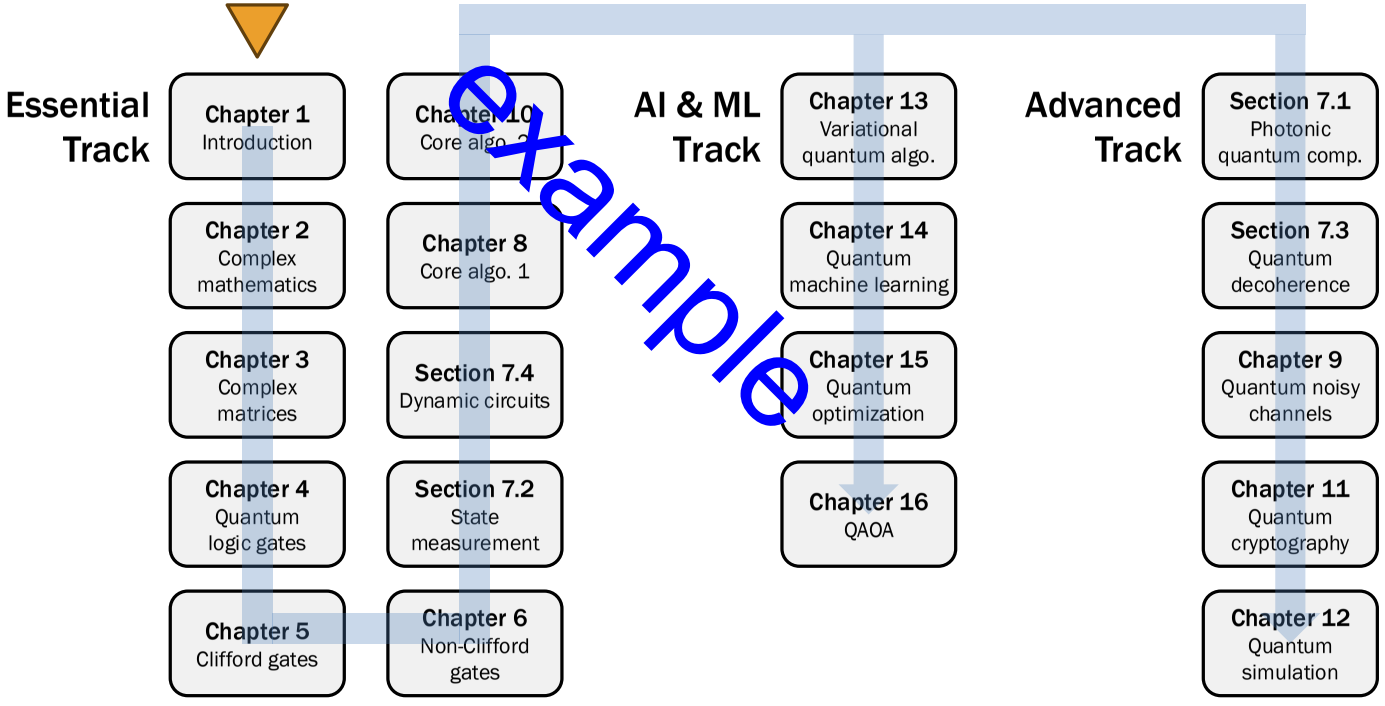
example

D.3 Course Learning Outcomes **530**
D.4 Course Syllabus **530**
D.5 Target Learning Objectives **531**
D.6 Supplementary Materials **534**

E Bibliography **535**

example

Content Map


Essential Track
Chapter 1 Introduction
Chapter 2 Complex mathematics
Chapter 3 Complex matrices
Chapter 4 Quantum logic gates
Chapter 5 Clifford gates
Chapter 10 Core algo. 2
Chapter 8 Core algo. 1
Section 7.4 Dynamic circuits
Section 7.2 State measurement
Chapter 6 Non-Clifford gates
AI & ML Track
Chapter 13 Variational quantum algo.
Chapter 14 Quantum machine learning
Chapter 15 Quantum optimization
Chapter 16 QAOA
Advanced Track
Section 7.1 Photonic quantum comp.
Section 7.3 Quantum decoherence
Chapter 9 Quantum noisy channels
Chapter 11 Quantum cryptography
Chapter 12 Quantum simulation
example

Motivation

In the modern age of artificial intelligence, computer systems become ubiquitous, smaller, and more intelligent. We are equipped with smart phones that talk back to us, autonomous vehicles with driving assistance capability, smart-home devices connected throughout the house, and adaptive wearable gadgets that keep track of our health status. Various tasks, once thought to be arduously and tediously manual, become fully automated by machine learning, whereby natural intelligences of pattern recognition are imitated from living creatures.

Despite these recent advancements, diminishing returns of classical hardware scaling start to emerge from the ever-increasing usage of *graphical processing units* (GPUs). As a rule of thumb, large-scaled models with trillions of parameters can be trained only on high-performance computing facilities. The more GPUs we employ, the more parameters we can accommodate and process in parallel.

However, the energy cost in this training practice significantly accrues due to several factors. First, we bump into the issue of *power wall*.[2] As we add more GPUs, the performance gain gradually saturates due to the issue of thermal ceiling in transistor-based CPU designs. Second, we encounter the issue of *memory wall*.[3] Training a large-scaled model requires high-bandwidth memory for data movement, while the growth of memory bandwidth does not catch up with that of CPU power. Third, the communication overhead on data synchronization in the interconnects is exorbitant owing to the issue of *network bottleneck*.[4,5] Fourth and finally, the state-of-the-art semiconductor design operates on the scale of 1.8nm-3nm.[6] Electrons start to leak through insulation barriers from one transistor to another via quantum tunneling, causing errors and heat as well as limiting the size of transistors. The cost to build semiconductors is no longer decreasing, and even worse, the cost of most advanced computing nodes starts to rise due to this limitation. This issue defies the traditional *Moore's Law*, where the number of transistors on an integrated circuit roughly doubles every two years with minimal increase in cost.[7]

example

Powered by classical hardware, modern AI technology is built on three fundamental pillars, and all of which are reaching the point of brute force:

***Representation learning*:** Deep neural networks learn from the dataset to transform raw data into high-dimensional meaningful vector representation, a.k.a. *vector embedding*. The main objective is to find a geometry in which data of different classes is linearly separable.

***Optimization problem*:** Current techniques for finding the optimal model parameters are the *stochastic gradient descent* (SGD) algorithm and its variants. Its limitation is that it is guaranteed to find a *local optimum* (locally best point) that minimizes the prediction error. The *global optimum* is still not guaranteed by this algorithm.

Hardware constraints: We have now hit the diminishing returns of classical hardware scaling as aforementioned. Training AI models on high-performance

computing facilities may not be an economically viable option due to the issue of energy inefficiency.

All of the hardware constraints can be directly addressed by quantum computing with superposition and entanglement. Quantum computing does not only offer faster AI, but it is also a paradigm shift in AI altogether! It changes the way we learn vector embeddings as wave functions and approximate the optimal model parameters with quantum-mechanical phenomena. We will develop basic ideas of the solution in the following sections.

✍ **Practice:** Question 1

1.1 Classical vs. Quantum Computing

You may have heard about the hype that quantum computers are immensely faster than classical computers. This statement is *theoretically* true, but we have to be cautious of its precise definition. For example, Shor's Algorithm can be used to factorize an integer N in $O(L^2 \log L)$ time complexity instead of the classical $O(L^2 \times 2^L)$, where $L = \log N$.[8] This makes RSA code decryption,[9] which is based on a pair of two large prime numbers, feasible in polynomial time.

Before we take a deep dive to the realm of quantum computing, it is customary to discern quantum computers from our modern-day ones. The term *classical computing* refers to deterministic information processing in the *von Neumann architecture*.[10] Hungarian-American mathematician John von Neumann proposed this architecture as the foundation for the EDVAC computer in 1945. As illustrated in Figure 1.1, a single data item is entered from the input device, loaded to the memory, fetched to and processed by the *central processing unit* (CPU), and passed to the output device.

example

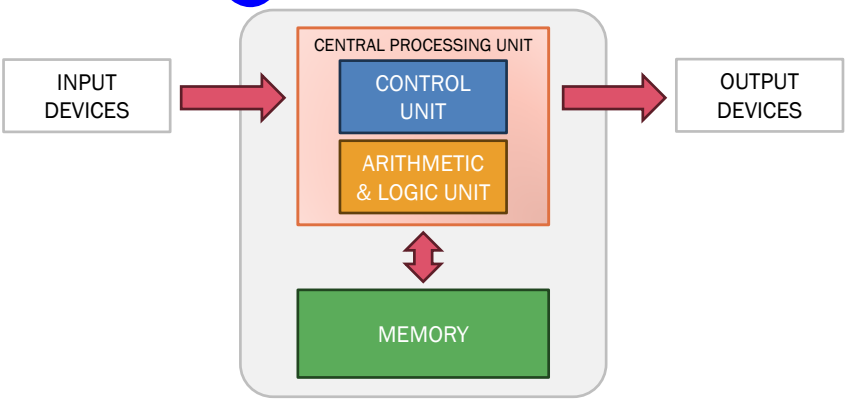


Figure 1.1: von Neumann architecture of classical computers

To speed up this computing sequence, we employ multiple CPU cores to process multiple data items at the same time. This technique is called *parallelism*. If we

compute 2^N data items with M CPU cores, the time complexity is $O(2^N/M)$, which is exponential. This makes classical computing not linearly scalable.

On the other hand, the term *quantum computing* refers to probabilistic information processing via the quantum phenomena of superposition, entanglement, and interference. As illustrated in Figure 1.2, multiple classical data items are transformed into a *superposition* of states, manipulated right in the memory with *interference* in quantum logic gates, and back-transformed into a classical output by statistical sampling (a.k.a *measurement*), destroying the original quantum state in the process. If there is a structure of data items, they are represented as an *entanglement* of states. If one of them is measured, the remaining entangled states will also collapse to their corresponding outputs at the same time, although they are astronomically far from each other. Frequently, the entanglement is compared to a data representation that allows information to flow in a quantum program.

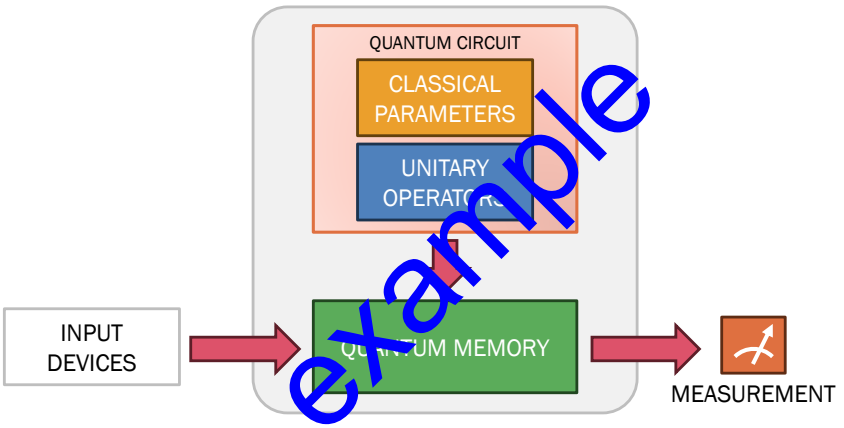


Figure 1.2: von Neumann-styled architecture of quantum computers

With superposition and entanglement, we can speed up some classical algorithms with an exponential time complexity to a linear one. This technique is called *quantum parallelism*. Each quantum bit, or *qubit*, is a data representation for *probability amplitude*s of 0 and 1, respectively. 2^N data items can be stored in only $\log_2 2^N = N$ quantum bits. Then, a system manipulates the probability distribution of all solution candidates with *interference*. This differs from classical parallelism, in which each solution candidate is independently computed on a finite number of computing cores. In this regard, processing these data items via quantum parallelism is equivalent to manipulating only N qubits in the memory, making quantum computing linear scalable despite an exponential input size.

✍ **Practice:** Question 2

$$\underline{\mathrm{QPE}}(\underline{U}) = \underline{\mathrm{QFT}}^{*}_{1:N}\left[\prod_{k=0}^{N-1} \underline{\mathrm{CU}}^{(2^k)}_{k+1,N+1}\right] \qquad \underline{U}_{\mathrm{QAOA}}(\underline{p},\underline{m},\underline{\Theta}) \approx \prod_{k=1}^{N} \underline{U}_L(p_k,\underline{\Theta})\,\underline{U}_M(m_k)$$

$$\mathrm{proj}_{\underline{u}}(\underline{v}) = \frac{\underline{v}\underline{V}\underline{u}}{\underline{u}\underline{V}\underline{u}}\,\underline{u} \qquad \underline{U}^{N} \approx \lim_{n\to\infty}\left[\mathrm{cis}\left(-\frac{\hat{N}\hat{H}_U}{n}\right)\right]^{n}$$

$$\mathrm{rej}_{\underline{u}}(\underline{v}) = \underline{v} - \mathrm{proj}_{\underline{u}}(\underline{v}) \qquad \hat{\underline{H}}^{(t)} = \left(1-\tfrac{t}{T}\right)\hat{\underline{H}}_B + \tfrac{t}{T}\hat{\underline{H}}_P$$

WHAT IS THIS BOOK?

Quantum computing offers new possibilities for AI, bringing the promise of faster and more powerful breakthroughs. This book is designed as an approachable bridge from machine learning into the quantum realm. If you are an AI engineer eager to discover what's next, this book is perfect for you.

ISBN 978-616-602-312-1
9 786166 023121
Price **550**.-Baht/**17.99** USD
Science
http://thammasatpress.tu.ac.th

In this book, we will explore:

- Computing with quantum mechanics
- Mathematical background (i.e. complex algebra) explained with intuitions, metaphors, and visualizations
- Quantum information and logic gates
- Reading and designing quantum circuits
- Foundational quantum algorithms e.g. Deutsch algorithm, Grover's search, Shor's algorithm, etc.
- Quantum simulation
- Quantum machine learning
- Quantum approximate optimization algorithms (QAOA)

example

$$\underline{U}^{\otimes N} = \underbrace{\underline{U}\otimes\underline{U}\otimes\underline{U}\otimes\ldots\otimes\underline{U}}_{N\text{ copies}} \qquad \leq \frac{\langle\psi|\hat{\underline{H}}|\psi\rangle}{\langle\psi|\psi\rangle} \qquad F(\rho,\delta) = \left[\mathrm{tr}\sqrt{\sqrt{\rho}\,\delta\sqrt{\rho}}\right]^2$$

$$\theta_j^{(t+1)} = -\frac{\pi}{2} - \arctan\frac{y_j^{(t)} - z_j^{(t)}}{2x_j^{(t)} - y_j^{(t)} - z_j^{(t)}} \qquad |\underline{u}| = \sqrt{\underline{u}\underline{V}\underline{u}} \qquad \underline{u}\underline{V}\underline{v} = \underline{u}^{\mathsf{T}}\bar{\underline{v}}$$

$$\underline{u}\wedge\underline{v} = \underline{u}\underline{v}^{*} \qquad \underline{u}\cdot\underline{v} = \underline{u}^{\mathsf{T}}\underline{v}$$

ABOUT THE AUTHOR



Prachya BOONKWAN (nickname: **Arm**) is a faculty member at the School of Information, Computer, and Communication Technology (ICT), Sirindhorn International Institute of Technology (SIIT), Thammasat University. He completed his high-school education at Triam Udom Suksa School in Thailand, before earning both his Bachelor's and Master's degrees in Computer Engineering from the Faculty of Engineering, Kasetsart University. He subsequently received his PhD in Informatics from the University of Edinburgh, UK. Driven by his lifelong passion, he is dedicated to breaking down complex technical concepts into accessible bite-sized knowledge. For more information, please visit his Github at `https://github.com/kaamanita`