通用量子计算机：理论、组成与实现

吴 楠[1),2)] 宋方敏[1),2)] Xiangdong Li[3)]

[1)] (南京大学计算机科学与技术系 南京 210093)

[2)] (计算机软件新技术国家重点实验室 南京大学 南京 210093)

[3)] (纽约城市大学计算机科学系 纽约市 美国 10016)

摘 要 通用量子计算机是指可以在不改变量子计算机物理组成和基本体系结构的条件下针对所有可计算问题进行量子计算及其它量子信息处理的设备. 通用量子计算机的研究和制造均具有重要的理论和实际意义. 要达成制造通用量子计算机的目标，需要在底层量子物理设备、量子计算机体系结构、量子资源调度和上层量子程序设计语言、量子算法及量子应用软件等多方面进行努力. 本文结合国内外在上述各方面研究的最新进展以及作者自身的研究结果，从计算机系统的角度尝试为通用量子计算机的研究和研制绘制一幅蓝图，并详细阐述其中的困难与努力方向.

关键词 量子计算；量子计算机；体系结构；量子信息；量子程序设计语言；量子算法；物理实现

中图法分类号 TP302

论文引用格式

吴 楠，宋方敏，Xiangdong Li，通用量子计算机：理论、组成与实现，计算机学报，2015，Vol.38：在线出版号 No.14

WU Nan，SONG Fang-Min，LI Xiang-Dong，Universal Quantum Computer: Theory, Organization and Implementation，Chinese Journal of Computers,2015, Vol.38: Online Publishing No.14

Universal Quantum Computer: Theory, Organization and Implementation

WU Nan[1),2)] SONG Fang-Min[1),2)] LI Xiang-Dong[3)]

[1)](Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

[2)](State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

[3)](Department of Computer Science, Graduate School, the City University of New York, NY 10016, USA)

Abstract A Universal quantum computer is a quantum computation device which can proceed quantum computation and other quantum information processing without changing the architecture of the physical device. Research and development of universal quantum computers raise important theoretical and practical significance. To reach the goal of building universal quantum computers, we should pay efforts on several aspects such as low-level quantum physical device, quantum instruction set system, quantum computer architecture, quantum resource schedule, high-level quantum programming languages, quantum algorithms and quantum applications. Combined with the studies of our own research results and the latest progresses of domestic and foreign researchers, this paper proposes a tentative study of the blueprint of building universal quantum computers in the

---

本课题得到国家自然科学基金重大研究计划（No. 91321312），国家自然科学基金（No. 61300050），国家自然科学基金创新研究群体科学基金（No. 61321491）和教育部基础研究项目博士点基金（No. 20120091120008）等项目的资助. 吴楠(**通讯作者**)，男，1981年生，博士，副教授，中国计算机学会（CCF）高级会员（No. E200015686S），主要研究方向为量子计算与新型软件设计. Email: nwu@nju.edu.cn. 宋方敏，男，1961年生，博士，教授，主要研究领域为量子计算、数理逻辑、软件方法学. Xiangdong Li，男，1966年生，博士，副教授，主要研究领域为量子力学、量子计算、信息安全

computer system point of view, and makes a detailed description of the difficulties and further works of universal quantum computers.

Key words quantum computation; quantum computer; architecture; quantum information; quantum programming language; quantum algorithm; physical implementation

1 概述

翻开二十世纪的物理学史册，正如仰望深邃浩渺的夜空，群星璀璨，其中最闪亮的几颗新星之一当属量子力学的提出与发展. 而同在二十世纪，电子计算机的诞生同样深刻地改变了世界. 量子力学和计算机科学与技术在二十世纪八十年代开始结合并诞生了一个多学科交叉与融合的新型学科方向：量子计算，它在备受关注中快速发展，并有可能由此产生足以改变传统计算模式从而对人类文明产生巨大推动的计算工具——量子计算机.

研究和制造量子计算机的主要动因是经典电子计算机的局限性. 电子计算机是一种以超大规模集成电路为物理基础的计算设备. 从数学角度看，电子计算机所进行的是依赖于布尔代数逻辑的通用计算. 计算的通用性在电子计算机诞生之前的1930 年代由 Turing 等人研究[1]，其中最著名的结论之一当属 Church-Turing 论题：

论题 1：一切直觉可计算的问题均可由图灵机模拟[1,2].

所谓“直觉可计算”在当时有许多描述方式，如递归函数或者可计算函数等，随着后来计算机科学的发展，这个提法逐渐被计算理论中的“算法”所取代，即：图灵机是可以实现任何算法的通用“计算机”. 有了 Turing 和 Church 所绘制的蓝图，在1940 年代，随着布尔逻辑和数字逻辑电路的发展，可以实现任何算法的通用电子计算机诞生了. 计算机的研究与发展逐渐成为现代科学、技术和工业的重要目标，并无所不在地改变着社会面貌和人类生活.

然而，在经历了电子计算机沿着“摩尔定律[3]”的轨迹快速发展的近 40 年后，人们逐渐认识到电子计算机的发展终究无法突破物理极限，这包括集成电路的密度、电子的量子效应、元器件的热效应等；另一方面，一些现有算法的时间复杂度过高，使得在输入规模较大时很难在可被接受的时间范围内解出. 为解决上述两个问题的探索直接导致了一种新的计算模式——量子计算和一种新的计算设备——量子计算机的提出.

1980 年，Benioff 提出用量子力学中哈密顿量模型来表示图灵机[4]；1982 年，Feynman 在 Benioff 的工作基础上提出，要解决传统计算机面对大规模量子系统模拟的难题的有效途径应该是用量子系统本身模拟另一个量子系统[5]，甚至可以用量子系统模拟其它复杂的物理系统——因为量子力学被认为是当前支配宇宙中所有物理现象的最基本的力学系统——这首次提出了用量子设备处理信息的想法. 1985 年，Deutsch 将 Feynman 的思想具体化并证明用量子计算机有效模拟在经典计算机尚不能有效模拟的系统是可能的[6].

对比基于经典图灵机和布尔逻辑系统的经典计算，量子计算具有一些前者难以比拟的优势. 这可归纳为以下 2 个方面：

第一，信息的表示与存储. 在经典计算中，信息和数据是以二进制数值的形式表示和储存的，比特是经典信息的基本度量单位，它可视为一个取值为$\{0,1\}$的随机变量，这便意味着一个比特在任一确定的时刻必为“0”或“1”，且不可得兼. 不论在数值的表示还是存储方面，布尔逻辑的表示空间都是实线性空间，该空间中的 1 个比特可视为模 2 整数加群$\mathbb{Z}_2$上的一维向量. 当希望通过扩展二进制数值的位数来增加信息和数据的表示和存储能力时，带来的效果只能是线性增长. 与之不同，由于量子态的叠加性（superposition，见于量子力学第一公设[7]），任一量子态可表示和储存为“0”（即量子基态$|0\rangle$）和“1”（即量子基态$|1\rangle$）的线性叠加（即量子态$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$，$\alpha,\beta \in \mathbb{C}$）. 这样，量子信息和量子数据便可表示和储存为上述量子态的形式，显然这不是实线性空间，而是 Hilbert 空间[8]. 故在量子计算中，量子信息的度量单位是以上述单量子态为存储单位的量子位（qubit），它在表示和储存方面带来的优势是随着量子位的增加，量子信息的表示空间和存储空间将以指数规模拓展.

第二，信息的处理. 在计算理论中，信息处理的能力涉及两方面：一是数据的表示能力，二是问题的求解空间. 上文已提及，基于图灵机的经典计算是通用的，任何经典算法都可看做一个接受不同

规模参数输入的可计算函数，函数的输出就是该算法对应的问题的解. 由于数据的表示空间是线性的，所以当求解问题的代价与输入数据的规模呈指数关系时，用确定性函数计算输入数据就需要指数规模的代价（时间或空间）. 而在 Hilbert 空间中由于量子态的叠加性，量子算法处理的问题表示和求解空间均为输入的指数规模，所以某些特定的函数便可以在线性时间内处理指数规模的输入，这种特性现已被诸多量子算法应用.

另外，量子计算作为量子信息的一种基本处理方式，在量子通信、量子网络等领域也起到重要作用.

虽然量子计算的概念最初是为了解决物理问题而由物理学家提出的，但随着量子计算的理论和实验技术的发展，量子计算和量子计算机受到了计算机科学与技术领域的广泛关注.

根据目前人们的认识，量子计算是一种基于量子力学基本原理的计算. 从计算机科学的研究任何计算都需探讨两个要素：一是“算什么”，即计算的对象；二是“如何算”，即计算的规则. 对于量子计算而言，计算的对象是服从量子力学基本原理的，由量子态表示的量子信息；计算的规则是封闭物理环境下的酉变换和测量.

量子计算的步骤一般可归纳为以下三步：

1、 初化（“入”）：信息“进入”量子计算的过程，指将经典信息通过某种形式的编码转化相应的量子态，即初态制备的过程；

2、 演化（“算”）：量子信息以量子态的形式在量子力学基本原理的框架下进行酉演化的过程. 在此过程中，初化后的信息逐步依照量子算法的步骤转化为的计算结果；

3、 测量（“出”）：将上述酉演化结束后的计算结果通过测量转化为经典结果的过程.

根据上述量子计算的物理特性，量子计算的优越性主要体现在以下 2 个方面：

1、量子并行性（quantum parallelism）：对处于叠加态的量子态$|x\rangle = \sum_i^n \alpha_i\ |\psi_i\rangle$进行函数$f$所对应的酉变换$U_f$，得到的结果为$U_f|x\rangle$，根据量子力学的公设，其为所有构成态$|x\rangle$的分量$|\psi_i\rangle$经函数$f$变换后所有分量的概率线性叠加，因此通过一次量子酉变换便可产生所有关于变量x的分量态$|f(x_i)\rangle$. 这种重要特性称为量子并行性. 在经典计算机中，上述计算需要n次计算或需n个处理器并行工作，而在量子计算机中，只要1次计算（变换）即可完成对所有叠加值的计算. 值得说明的是，量子计算虽然可以通过一次操作产生n个（隐含的）结果，但只能读出其中的一个结果. 另外，由于量子的不可克隆性原理（no-cloning theorem），使得一个未知的量子态不可能被精确复制，这意味着量子计算的结果不可能通过复制而进行保存.

2、量子纠缠态（entanglement state）：量子纠缠是量子力学中特有的现象，其物理表述和物理过程目前尚不清楚，它可能与非定域性等更深层次的物理原理有关[7]. 量子的纠缠态的数学表述为：若一个量子计算系统中多个量子位的态不能表示成子状态的张量积的形式，则称这多个量子位处于纠缠态，纠缠态首先在 1935 年由 Einstein 等提出[9]. 如 Bell 态：$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$即为一个典型的纠缠态，其中的两个量子位无法写成张量积形式；当多个量子位处于纠缠态时，对部分量子位的态的测量将影响其它量子位的态的测量，即便这些量子位分处于不同的空间位置. 例如测量 Bell 态$|\psi\rangle$时，测量一个量子位的态将使另一个量子位的态必然与之相反. 量子纠缠是一种重要的量子信息资源，在量子隐态传输、量子通信、量子超密编码、量子密钥分配以及在量子计算的加速、量子纠错、防错等方面都起着重要作用.

量子纠缠态使量子计算机具有更大的优越性，相隔很远的两个处于纠缠态的量子态具有瞬时相关性，改变其中一个的状态另一个状态则立即随之变化，这种关联跨越了空间和时间，与之相关的量子隐态传输通信试验已经取得成功. 在量子纠缠的情况下，n个量子位最多可同时处于2^n个不同状态，这些不同的状态在这些量子位载体上可同时进行计算，产生了量子计算相对于经典计算的指数级加速.

此外，量子傅里叶变换（quantum Fourier transformation，QFT）、量子隐形传态（quantum

teleportation）等都可能是量子计算可能超越经典计算的根源.

许多计算复杂度很高的问题（如 NP 问题）目前尚不能在经典计算机上有效地进行计算，由量子并行性，可使某些量子算法对比经典算法具有指数级加速.

2 量子计算的通用性

与经典计算一样，量子计算也有不同的计算模型，常用的量子计算模型有：量子图灵机模型[4,6,10]（即量子门电路模型，1989 年）、量子线路模型[11]（1993 年）、量子随机访问机模型[12]（QRAM，1996 年）、绝热量子计算模型[13]（2000 年）、拓扑量子计算模型[14]（2003 年）等. 这些模型彼此在计算能力上的等价性已得到证明[7,15-17].

本节只讨论量子计算在量子线路模型上的通用性，其余模型的通用性皆可由其计算能力等价性得到.

量子计算的通用性研究始于对量子图灵机模型和量子线路模型的通用性研究. 类比经典计算中的通用布尔逻辑门（如与非门等）的通用性，Deutsch 在 1989 年提出通用量子逻辑门的概念[10]. 量子逻辑门是在量子环境中执行量子酉演化的逻辑部件，在量子线路模型中，量子逻辑门接受有限个量子态的输入并输出有限个经该逻辑门变换（酉变换）的量子态. Deutsch 指出，存在一个有限量子门集合，其元素的有限组合可以进行任意有限精度的量子计算. 随后，DiVincenzo 和 Lloyd 又证明了任意 2 量子位量子逻辑门的通用性[18,19]. 1997 年， Kitaev 等提出，在容许有界出错概率ϵ的情形下，可以用有穷多（$O(\log^2 1/\epsilon)$）个量子门进行通用量子计算[20].

通用量子逻辑门集合的提出为设计量子计算机指令系统提供了理论依据. 常用的通用量子逻辑门集合有：{CNOT，1-qubit 逻辑门}、{CNOT，Hadamard 门，相位门}、{Toffoli 门，Hadamard 门}等，其中 Toffoli 门即受控 CNOT 门[21]. 在实际设计中，可以根据不同的量子计算机体系结构和量子计算机的物理实现方式确定具体的通用量子门集合和量子计算的指令集.

3 通用量子计算设备

通用量子计算设备是通用量子计算机系统的“硬件”部分. 从目前的发展趋势分析，通用量子计算设备面临的主要问题为：1、逻辑层面，如如何调度平衡用于计算和用于存储的量子资源，如何有效进行容错与纠错等；2、物理层面，如量子位的规模化储存和操控技术等.

一般而言，通用量子计算设备具备以下基本要素：

1、 结构化的量子存储设备. 与经典计算机的硬件体系结构类似，目前认为量子存储设备是通用量子计算设备必须具备的. 而同经典计算机的层次化存储部件相比，量子存储设备还需要一种特殊的容错和纠错结构：这是因为经典信息中的错误仅为比特翻转，即“0”与“1”的互置，而量子信息中除比特翻转外还存在相位偏移，后者导致对量子信息的容错和纠错过程远比经典信息复杂和低效. 故需要在量子存储设备中设计专门的容错和纠错硬件单元和逻辑结构以便进行存储设备中的容错与纠错[22]. 具体内容将在第 6 节中阐述.

2、 混成式运算器及运算控制设备. 通用量子计算设备的核心任务是要进行通用量子计算和其它相关的量子信息处理，由于量子算法不可缺少地要有经典的预处理（将经典数据转换为量子数据）和后处理（将测量后的量子计算结果进行正确性判断和后续处理）的过程，因此应在量子计算设备中引入经典计算能力. 虽然已证明量子计算模式也能进行经典计算，但考虑到量子计算的代价和量子位资源的限制，目前较好的方式是在量子计算设备中同时引入量子运算部件与经典运算部件以构成混成式运算器. 在理论上，邱道文等曾提出半量子自动机计算模型（单向量子有穷自动机）[23]和半量子密钥分配协议[24]，刻画了混成式通用量子计算和量子通信中量子部分与经典部分的交互作用. 要引入运算控制设备以控制两种运算各自的过程及计算中的协同[25]，在量子运算部件中也需要加入容错与纠错机制，为了提高容错用资源的复用度，可考虑将量子运算操作直接置于量子存储设备中并通过量子远程传态（teleportation）与量子运算部件（含少量量子寄存器）进行通信. 此时量子运算部件只需负责远程传态过程中和量子寄存器内的容错与纠错. 混成式运算器的主要功能在第 6 节中详述.

3、　经典信息与量子信息的转换设备. 由于人与量子计算设备的交互是通过经典信息完成的，所有量子程序接受的输入和给出的输出也都是经典信息的形式，因此需要引入经典信息与量子信息的转换设备. 在目前技术条件下，经典信息向量子信息的转换（量子信息引入）一般通过初态制备方式完成，即将经典形式的输入信息转换为量子编码并在量子设备中制备成相应的量子初态；在量子计算完成后，通过量子测量设备将量子计算的结果投影到系统的基向量中并将其转换为经典变量的值[22]. 在理论设计阶段，一般假设系统的演化在封闭环境中完成，因此测量一般采取正交投影测量；在实际应用中，由于系统处于混态，也可采用其它合适的测量方式.

4、　量子资源调度设备. 量子资源调度系统在量子计算设备中以硬件形式工作，同时也在软件层面工作. 具体见第 7 节.

上述各设备的级联方式以及组成与结构均会影响量子计算设备的性能和容错度，因此必须在设计阶段审慎地考虑量子计算设备的组成与结构，并采用模拟方式进行性能分析与测试.

4　通用量子计算机的物理需求

要制造出通用量子计算机，工作需分物理和逻辑两个层面，前者是后者的物质基础. 关于量子计算机的物理实现技术，Loss 和 DiVincenzo 于 1998 年和 2000 年提出了著名的 DiVincenzo 判据[26,27]，认为要制造通用量子计算机在物理实现技术上必须提供：1、一个易扩展的、具有良好性质的双能级物理系统[1]以构建量子位；2、一套可以将量子位初化为任意已知量子初态的方法；3、具有远长于单个量子门演化时间的系统相干时间；4、为量子位提供一组精确度可控的通用量子门集合；5、可控的测量技术；6、使活动量子位与静止量子位可互相转化的能力；7、使活动量子位可在特定的物理位置之间自由传送的能力. 其中前 5 个判据主要针对量子计算机的设计和实现，后 2 个判据主要针对量子通信的设计与实现. 在实际量子计算机系统的理论和实现设计中，DiVincenzo 判据（特别是前 5 个判据）是最基础也是事实上的最低设计准则.

最近，随着量子操控技术的发展，多种新型的量子操控平台和物理实现设备被提出和研制. 表 1 按照 DiVincenzo 判据对其进行分析[28].

表 1. 目前常见的量子计算物理实现平台（或技术）对 DiVincenzo 判据的支持程度[2]（☆代表理论和实验均支持本判据；〇代表理论支持本判据且相关实验正在进行；×代表目前尚无理论和实验技术支持本判据）

量子计算物理实现	DiVincenzo 量子计算实现判据						
平台（或技术）	判据 1	判据 2	判据 3	判据 4	判据 5	判据 6	判据 7
液相核磁共振	〇	〇	〇	☆	〇	×	×
固态核磁共振	〇	〇	〇	〇	〇	×	×
离 子 阱	☆	☆	〇	☆	☆	〇	〇
中性原子	〇	☆	〇	〇	〇	〇	〇
光学（光子偏振）	☆	〇	☆	〇	〇	☆	☆
光学（波导）	〇	〇	☆	〇	〇	×	☆
超导宏观量子效应	☆	☆	〇	〇	☆	×	×
腔量子电动力学	〇	☆	〇	☆	☆	〇	×
Kane 硅基半导体	〇	〇	〇	〇	〇	×	×
Fullerene 球	〇	〇	〇	〇	〇	×	×
量 子 点	〇	〇	〇	〇	☆	×	×
电子自旋共振	☆	☆	〇	〇	☆	×	×

目前在实验中最常见的实现途径分别为核磁共振（液相或固态）、光学、离子阱、量子点和超导宏观量子效应. 其中离子阱操控平台目前最多可支持对 14 个物理量子位的纠缠态进行相干操作及测量[29]，这也是目前公开发表的可操控最多量子位的实验实现，其它物理实现平台已报道的最新成果分别为 8 个物理量子位（基于绝热量子效应的超导磁通宏观量子效应[30]、线性光学系统[31]）、3 个物理量子位（量子点[30]）和 2 个物理量子位（基于量子逻辑门的超导磁通宏观量子效应[32]）. 根据相关文献[7,30]的报道，图 1 给出上述 5 种实现途径对于前 5 个判据的 5 分制评价，评分越高表明对判据的支持度越高.

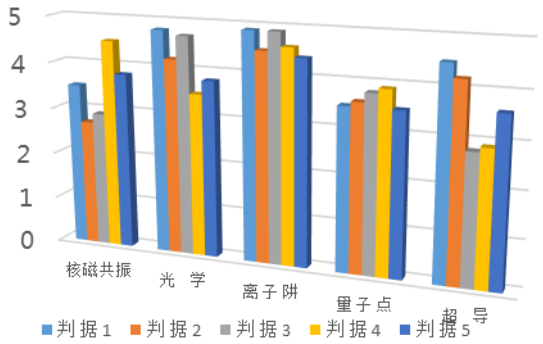

图 1. 最常见的 5 种物理实现途径对 DiVincenzo 判据 1 到判

[1] 在后来的研究中，三能级量子系统（qutrit）和多能级量子系统（qudit）作为基本量子位也已实现.

[2] 表 1 中所有量子计算物理实现平台（或技术）均考虑基于量子线路模型的实现，不包括绝热量子计算、拓扑量子计算和几何量子计算的物理实现途径.

据 5 的支持度

5 量子计算机指令系统

如同经典计算机一样，量子计算机的指令系统处于量子计算设备与量子计算软件之间的界面.

量子计算机指令系统的设计主要考虑并平衡两方面的因素：在“硬”的方面考虑指令实现的物理代价，在“软”的方面考虑指令的计算通用性以及支持程序编写的方便程度.

量子计算的通用性既约束了量子计算机的指令系统，又为指令系统中指令的选择提供了丰富的空间. 如同传统计算机即便最少只用 2 类基本电路门（如与门、非门）便可实现通用 Turing 计算一样，可以最少只用 2 种运算指令（如受控非门、任意旋转门）和其它几种数据转移和测量指令就可实现通用量子计算（参见第 2 节）. 但在实际的指令系统设计中，考虑到指令实现的物理复杂度以及指令系统对程序人员的易用性，实际设计指令条数一般都多于上述最低需求.

从计算角度考虑，混成式量子计算机的指令系统由如下 5 类指令构成：

1、 纯经典指令（与经典计算机的指令集相同）；
2、 量子初化指令；
3、 量子演化指令；
4、 调度指令；
5、 测量指令.

量子初化指令可细分为两种：1、指定量子位的零基态制备指令；2、指定量子位的指定态（即输入态）置入指令. 前者通过制备指定物理平台下的零基态（ground state），使量子存储设备中某个或某些量子位成为零纯态：$|00\cdots0\rangle$；后者则是完成指定非基态纯态置入过程，即在零纯态基础上通过特定的量子门生成指定的量子输入态.

演化指令又可分为两类：1、基本量子门指令，包括单量子位基本指令和双量子位基本指令；2、量子宏指令，包括 3 量子位指令和n量子位指令（三个以上的量子位指令）. 第一类中基本量子门指令已经可以构成通用量子门集合，第二类演化指令主要是为了提高运算功效和降低编译后的机器指令数量而设计. 基本量子门指令的选取遵循完备、简明、易用三原则[28]，通过分析和模拟实验，单量子位基本指令选取为：Nop（空指令）、H（Hadamard）、I（恒同）、X（NOT）、Z（Pauli-Z）、T（$\pi/8$）、S（相位）、Phase（移相）、RotX（绕 X 轴旋转）、RotY（绕 Y 轴旋转）、RotZ（绕 Z 轴旋转）、Measure（$\hat{z}$-基投影测量）. 双量子位基本指令选取为：CNot（受控非）、Swap（对换）、CZ（受控 Pauli-Z）、CS（受控相位）、CPhase（受控移相）、CRotX（受控绕 X 轴旋转）、CRotZ（受控绕 Z 轴旋转）. 第二类演化指令包括：Toff（Toffoli，3 量子位）、Fred（Fredkin，受控对换，3 量子位）、U （任意n-量子位量子门，n量子位）、C-U（受控任意n-量子位酉门，n量子位）、CC-U（受控-受控任意n-量子位酉门，n量子位）、CCC-U（受控-受控-受控任意n-量子位酉门，n量子位）.

量子计算的指令集由上述诸量子指令对应的指令名和操作对象构成：

单量子位基本指令集：$\mathrm{Nop}(\mathbf{void})$，$\mathrm{H}(\mathbf{qubit}\ q_{\mathrm{op}})$，$\mathrm{I}(\mathbf{qubit}\ q_{\mathrm{op}})$，$\mathrm{X}(\mathbf{qubit}\ q_{\mathrm{op}})$，$\mathrm{Z}(\mathbf{qubit}\ q_{\mathrm{op}})$，$\mathrm{T}(\mathbf{qubit}\ q_{\mathrm{op}})$，$\mathrm{S}(\mathbf{qubit}\ q_{\mathrm{op}})$，$\mathrm{Phase}(\mathbf{qubit}\ q_{\mathrm{op}},\ \mathbf{real}\ c_\theta)$，$\mathrm{RotX}(\mathbf{qubit}\ q_{\mathrm{op}},\ \mathbf{real}\ c_\theta)$，$\mathrm{RotY}(\mathbf{qubit}\ q_{\mathrm{op}},\ \mathbf{real}\ c_\theta)$，$\mathrm{RotZ}(\mathbf{qubit}\ q_{\mathrm{op}},\ \mathbf{real}\ c_\theta)$，$\mathrm{Measure}(\mathbf{qubit}\ q_{\mathrm{op}},\ \mathbf{int}\ c_{\mathrm{m}},\ \mathbf{real}\ c_{\mathrm{pre}})$；双量子位基本指令集：$\mathrm{CNot}(\mathbf{qubit}\ q_c,\ \mathbf{qubit}\ q_{\mathrm{op}})$，$\mathrm{Swap}(\mathbf{qubit}\ q_{\mathrm{op}_1},\ \mathbf{qubit}\ q_{\mathrm{op}_2})$，$\mathrm{CZ}(\mathbf{qubit}\ q_c,\ \mathbf{qubit}\ q_{\mathrm{op}})$，

$\mathrm{CS}(\mathbf{qubit}\ q_c,\ \mathbf{qubit}\ q_{op})$ ，

$\mathrm{CPhase}(\mathbf{qubit}\ q_c,\ \mathbf{qubit}\ q_{op},\ \mathbf{real}\ c_\theta)$ ，

$\mathrm{CRotX}(\mathbf{qubit}\ q_c,\ \mathbf{qubit}\ q_{op},\ \mathbf{real}\ c_\theta)$ ，

$\mathrm{CRotZ}(\mathbf{qubit}\ q_c,\ \mathbf{qubit}\ q_{op},\ \mathbf{real}\ c_\theta)$.

量子宏指令集：

$\mathrm{Toff}(\mathbf{qubit}\ q_{c_1},\ \mathbf{qubit}\ q_{c_2},\ \mathbf{qubit}\ q_{op})$ ，

$\mathrm{Fred}(\mathbf{qubit}\ q_c,\ \mathbf{qubit}\ q_{op_1},\ \mathbf{qubit}\ q_{op_2})$ ，

$\mathrm{U}(\mathbf{qureg}\ q_{op}, \mathbf{int}\ N, \mathbf{matrix}\ q_{mat})$ ，

$\mathrm{C-U}(\mathbf{qubit}\ q_c, \mathbf{qureg}\ q_{op}, \mathbf{int}\ N, \mathbf{matrix}\ q_{mat})$ ，

$\mathrm{CC-U}(\mathbf{qubit}\ q_{c_1}, \mathbf{qubit}\ q_{c_2}, \mathbf{qureg}\ q_{op}, \mathbf{int}\ N, \mathbf{matrix}\ q_{mat})$ ，

$\mathrm{CCC-U}(\mathbf{qubit}\ q_{c_1}, \mathbf{qubit}\ q_{c_2}, \mathbf{qubit}\ q_{c_3}, \mathbf{qureg}\ q_{op}, \mathbf{int}\ N, \mathbf{matrix}\ q_{mat})$.

其中 **void** 代表空类型，**qubit** 代表量子位类型，**real** 代表实数类型，**int** 代表整数类型，**qureg** 代表量子寄存器（量子位串）类型，**matrix** 代表矩阵类型（表示操作酉矩阵），所有操作对象均为有类型变量且均对应于上述诸变量类型. 这组量子指令集既考虑了量子计算在不同物理实现平台上执行的功效又考虑了代码生成长度的均衡性，同时还兼顾了指令使用的灵活性和易扩展性.

由于量子算法的复杂性，直接通过底层量子逻辑门实现量子算法或通过量子语言的处理程序将高级语言直接编译为底层逻辑门序列是比较复杂的. 故此可在指令集中加入量子原始操作（primitive）以提高量子指令的层次. 量子原始操作是由若干量子指令（或若干量子指令和若干经典指令）组成并用于完成一个特定量子操作的量子过程. 可将诸如量子傅里叶变换（QFT）、控制酉门等作为量子原始操作，也可针对算法中常用的步骤将模指数运算、特征值估计、相位估计、相位态制备等作为高层量子原始操作.

图 2 显示一个求函数$f(x)$的阶的量子计算过程，及其由低层或高层量子原始操作实现的框图.

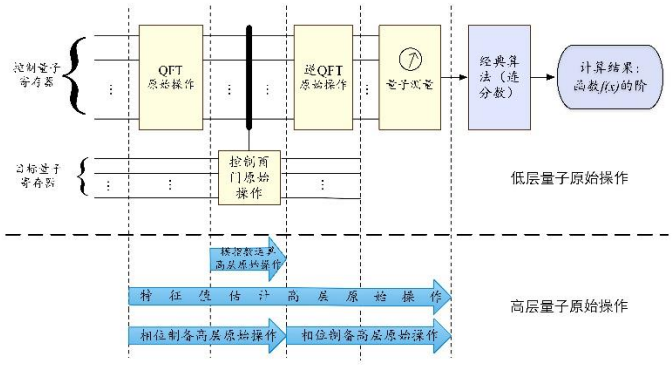

图 2. 用底层或高层量子原始操作实现函数求阶的量子计算过程

6　通用量子计算机体系结构

目前一种经典的计算机体系结构的定义是由 Amdahl 在 1964 年介绍 IBM 360 电子计算机系统时提出的，其具体描述为“程序人员所看到的计算机的属性，即概念性结构与功能特性”[33]. 关于通用量子计算机的体系结构也可进行类似的定义，即通用量子计算机的硬件对程序人员可视的属性与功能.

通用量子计算机的设计关键在于其体系结构的设计. 量子计算机的体系结构不仅影响硬件的组成与结构，同时也影响量子计算机的软件设计.

要构建具有实用意义的通用量子计算机系统，一些基础构架以及理论和技术壁垒必须得到一定程度的发展和突破. 图 3 给出在整个量子计算领域中体系结构所处的位置和与其它领域的依赖关系.

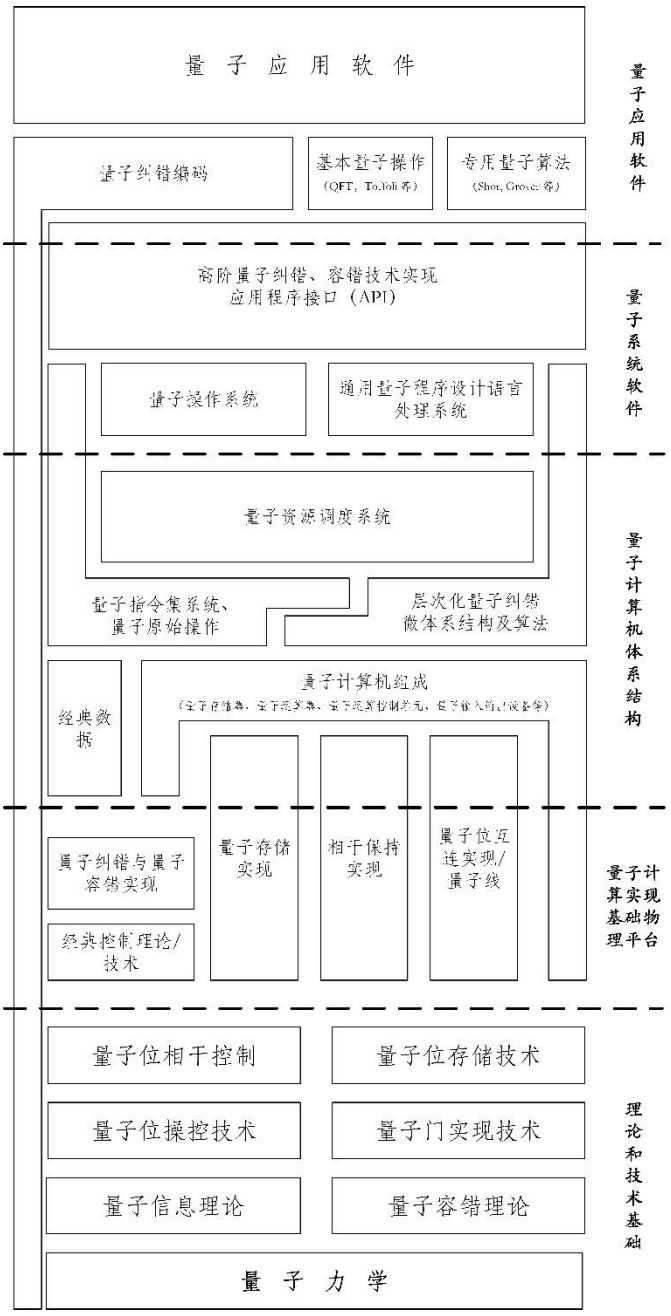

图 3. 量子计算机体系结构与量子计算中其它成分的关系图

在目前的理论与技术条件下，量子计算机的体系结构设计较经典计算机远为复杂，具体主要表现在以下几个方面：

1、 量子计算的数学模型. 量子计算与经典计算的重大区别在于其基础数学模型的差异. 经典计算机软件与硬件的数学模型分别是图灵机模型（对应经典算法）和布尔逻辑模型（对应数字逻辑电路）. 以数据和程序的存储为中心、以运算过程为命令驱动的 von Neumann 体系结构的提出是基于上述两种数学模型的考虑. 而量子计算的数学模型与上述两种模型有明显区别，这就要考虑量子计算机的体系结构是否依然遵循以往的经典模式. 理论研究和模拟实验认为，量子计算机体系结构宜在经典计算机体系结构的基础上依照其特有的模型重新设计.

2、 量子计算的资源. 随着计算机科学的发展与技术的进步，经典计算资源的可用性和扩展性等问题目前已经得到了很好地解决，无论主要用于计算的内存设备还是主要用于存储的外存设备，其稳定性与易扩展性都已达到或超过了应用软件的需求. 与之相比，作为量子计算核心资源的量子位目前还是非常稀缺的，更重要的是由于退相干等原因，储存于量子位的量子信息的保真度和稳定性都比较差，这就要求在原本稀缺的量子位资源中，大部分物理量子位要作为辅助量子位参与到容错量子计算中. 理论表明，在能较好进行容错和纠错量子计算的系统中，物理量子位与逻辑量子位的比例应达到6:1（CSS [[7,1,3]] 量子纠错码）甚至更高[7]. 量子计算资源的有限性对体系结构的可靠性、易扩展性以及量子资源的易复用性提出了严峻挑战.

3、 经典信息与量子信息的交互. 考虑第 1 节中提出的量子计算的主要步骤，经典信息与量子信息的转化一般在于“入”和“出”两个环节中. 同时，量子计算机体系结构还必须考虑经典计算的的问题，因此一般考虑设计成经典部分与量子部分混成式的体系结构，经典部分主要处理经典计算和整体控制，量子部分主要处理量子计算和量子测量等.

4、 要考虑到量子计算和量子信息的特有性质. 量子信息中有许多特有的性质与经典世界完全不同，如量子纠缠、远程传态、不可克隆原理、不可删除原理等. 要考虑从体系结构的层面将量子信息中不允许的操作予以禁止或采用其它方式代替，这样便可避免在上层系统软件（如语言处理程序）中以额外的方式进行处理. 同时可将量子纠缠、远程传态等性质加以利用，生成诸如量子线、代码远程传态等全新的量子信息处理方式.

5、 考虑易用性. 计算机的易用性与其体系结构有很大关联. 与经典计算相比，量子计算由于本身的物理背景，其对应的软硬件的易用性便不如前者，这就要求在设计体系结构时尽可能增加相关设备或部件对程序人员的透明性，避免对程序人员的量子计算或量子信息知识有过高要求.

量子计算机的体系结构还主要与量子计算的物

理实现平台以及量子计算设备的物理构成有密切关联. 这便导致目前无法像经典计算机一样给出非常具体的性能指标.

目前国际上一些研究机构对量子计算机体系结构问题进行了研究，并提出了若干体系结构模型. 两种比较重要的基于物理实现平台而设计的容错量子计算机体系结构模型分别为：由美国加州大学戴维斯分校 Metodi 和麻省理工学院 Chuang 等提出的量子逻辑阵列体系结构（quantum logic array architecture，以下简称 QLA）[34]以及在 QLA 基础上由加州大学戴维斯分校 D. Darshan 和 Chong 等提出的压缩量子逻辑阵列体系结构（compressed quantum logic array architecture，以下简称 CQLA）[35].

这两种模型在数学上是同态的：其作用均为将高层量子操作映射为基于线路的量子计算模型中. 两种模型都可以将一个复杂的量子计算过程描述为一系列“计算片状单元（computational tiles）”的串行操作，每个计算片状单元皆由一个被称之为“逻辑量子位”的具有容错功能的特殊量子位构成，根据容错方式的不同，一个逻辑量子位由多个物理量子位编码而成. CQLA 和 QLA 的主要区别在于 CQLA 更注重了系统层的平衡——主要是速度与容错能力的平衡以及量子计算机的可靠性与物理实现难度的平衡.

上述两种量子计算机体系结构的提出引起了学术界和工业界的广泛关注，人们认为若要让量子计算机支撑未来的关键应用，必须很好地平衡量子位规模化、运算速度、可靠性以及物理实现难度等多种性能. 故此又有多位学者相继提出对此体系结构的改进设想，这要集中于：

1、选择哪种容错方式可以更好地将可容错规模化量子数据与量子存储结构、计算结构以及规模化量子通信机制相结合[36]？

2、运行于经典计算机部分的量子调度程序如何合理地调度量子程序的运行[36,37]？

3、如何对规模化量子计算过程中各种资源的使用进行负载平衡[36,37]？

4、面对快速发展的量子信息物理载体和平台技术，现有的体系结构如何适应未来的量子设备的发展[38]？

遗憾的是，在目前的技术条件下希望完美地解决上述问题仍很困难，主要原因是：第一，量子信息在目前的物理实现中皆是以自然界实体粒子、光子、势阱或者宏观微电子器件为载体，通过编程后的人工干预行为（如脉冲、磁场、微波、光学器件等）进行操控的，这事实上破坏了量子系统的封闭性，使量子计算系统与环境产生了耦合从而引起量子信息向环境泄漏，这将导致系统可靠性的降低[39,40]. 第二，在具备一定规模的量子计算机系统中，大量的量子位会不可避免地与环境形成耦合，并展示出其经典物理效应——经典物理效应无法像量子叠加和量子纠缠等量子效应那样可帮助超越经典计算——因此，在实施量子门作用时应当对相关的量子位动态去耦，对量子效应的观测还应该能区分出其经典成分和量子成分. 而在这些方面，理论与实际仍有较大的距离.

纵观量子计算机体系结构研究的发展，我们可以发现，不论量子体系结构模型的提出还是具体的细节设计，绝大多数工作都是由物理学家完成的，这种现状的原因是量子计算机本由物理学家提出，故对体系结构的考虑和设计一般都从可实现的物理条件出发去考虑容错、易扩展性等问题；但上述情况的缺点也很明显：量子计算机是一种用于计算的系统，最终目的是运行量子程序——它不仅由硬件逻辑组成，同时还应由建立在物理设备之上的控制逻辑和软件逻辑共同构成，而目前对后者的研究却较少，当然，这还与计算机领域的科学工作者较少参与量子计算机的设计等原因有关. 在硬件逻辑之上，量子计算机系统还应包括支撑量子计算正常进行的容错（主要指逻辑容错）系统、指令系统、量子编译或解释程序以及未来的量子应用程序等.

事实上，计算机体系结构研究的重要目标之一就是确定一台计算机硬件的组成部分以及与软件之间的衔接[41]. 因此我们认为，要建立合理的可靠、易扩展的规模化通用量子计算机体系结构除了要有物理学家参与以外，还需要在计算机系统的角度上考虑和确定一台量子计算机的部件、功能、部件之间的接口以及软硬件之间的接口. 在基本组成建立之后，还应从计算机科学的角度考虑体系结构中各系统资源之间的均衡. 我们认为主要的均衡有：

1、量子计算可靠性与量子通信负载的均衡；

2、量子传输总线负载与量子信息对换（quantum swap）操作之间的均衡；

3、量子应用程序层需求与通用量子逻辑门实现复杂度的均衡；

4、计算可靠性与容错系统复杂性的均衡；

5、量子存储时间开销（因量子位无法同经典比

特一样长久地保持信息）与内存的层次化纠错结构之间的均衡.

我们对量子计算机体系结构研究方向进行了多年的跟踪，发现上述研究过去彼此相对独立，鲜有将其有机结合并与量子算法、量子软件和量子资源调度统一考虑与均衡的体系结构提出. 在此研究的基础上，我们提出了一个可对计算空间中的量子相干性进行保持的易扩展量子计算机体系结构，称为ECoPA（environment coherence persisting architecture）.

ECoPA 在组成上延续了经典计算机的几大主要部件的名称和主要功能，包括：1、量子存储单元；2、量子计算与逻辑处理单元；3、量子运算控制与资源调度单元；4、量子初态置入单元；5、量子测量与经典输出单元. ECoPA 是以存储为中心的量子计算机体系结构[22,42-44].

ECoPA 中量子存储单元的组成与结构图见图 4.

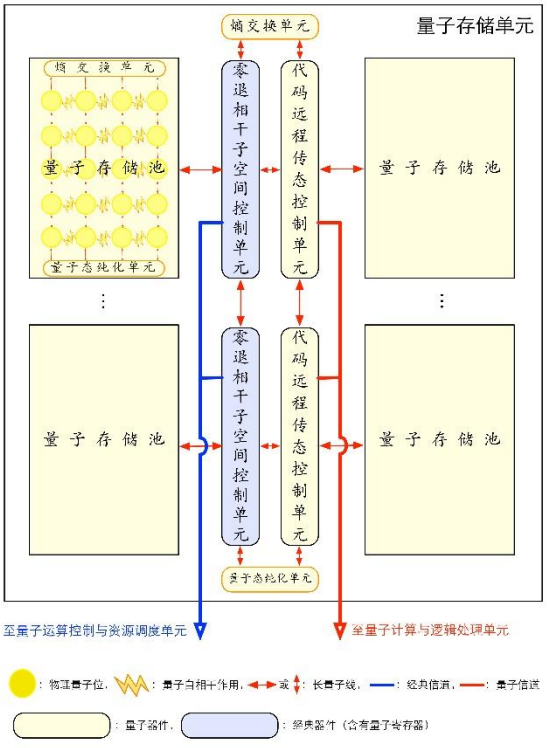

图 4. 环境相干性保持体系结构（ECoPA）的量子存储单元框图

在 ECoPA 体系结构中，量子存储器是具有核心地位的组成部件，它不仅是存储量子信息（不存储经典信息）的部件，而且是量子计算（量子逻辑门操作）发生的部件，这与经典计算机是不同的. 在量子存储器中，为了对存储的量子信息和量子逻辑门操作进行相干性保护，避免环境退相干的影响，我们将负责存储和计算的量子位划分为一些容量不等的区块，称为量子存储池（quantum memory bank），针对每个存储池采用两种途径保护量子信息：即零退相干子空间和层次化量子纠错编码. 通过这样的设计，在有效避免环境退相干作用影响的同时，降低了使用统一量子纠错码带来的存储空间开销，使得逻辑量子位与物理量子位之比最低可达到 1:4，具体理论分析和实验见[22,28]. 在各存储池之间，进行量子通信的信道是称为量子线（quantum wire）[45]的逻辑信道，量子线通过量子计算机指令集提供的 Swap 指令和 Bell 态$|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + |11\rangle$来实现，它是一种基于量子隐形传态的逻辑信道，其通信容量理论上可达到量子信道容量极限. 1999 年，Gottesman 等提出在基于 Bell 态量子通信的同时还可进行单量子逻辑门计算，而且量子计算与量子通信可同时发生，这种现象被称为“量子门隐形传态（gate teleportation）”[46]，2001 年 Raussendorf 等证明用隐形传态进行传输时计算并不影响量子计算的计算能力[47]，2003 年 Neilsen 证明任意有界误差酉变换均可用此种方式实现[48]，量子门隐形传态的这种过去不被人所重视的通信与计算的并行性在 ECoPA 体系结构中引入的量子线中可在很大程度上提高量子计算的效率. 为了有效进行量子存储器内的量子门隐形传态，我们在 ECoPA 体系结构的量子存储器中引入了长量子线（根据数值模拟，用离子阱实现的长量子线在无中继器时最长可达 1.8mm[3]），量子门隐形传态单元（GTU），量子中继器（QRPT，根据数值模拟，使用量子中继器可将量子线长度提高至 10mm，条件与来源同上），量子态纯化单元（SPU），熵交换单元（EEU）等专用设备，同时上述设备受零退相干子空间和层次化量子纠错编码的保护[28].

基于 CNOT 门和单比特量子门的通用性，图 5 展示了一个基于量子门隐形传态和 Bell 测量的量子

[3] 见M. Oskin 2003年的技术报告“Quantum Architecture: More Unknown than Known”.（http://www.cs.washington.edu/homes/oskin/quantum-tutorial/QuantumTutorial.ppt）

信息传递—计算通路，并实现了“远程 CNOT 门[49]”的并行通信与计算，未来可考虑在量子计算机体系结构中将远程 CNOT 门作为一种基本器件（或基本逻辑门）来设计和实现.

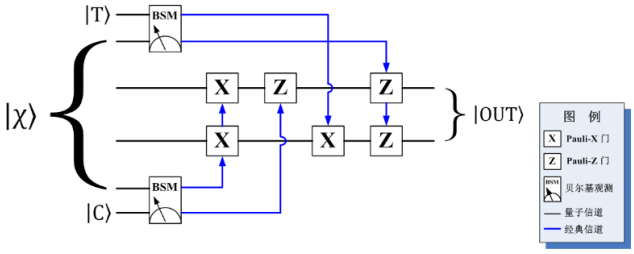

图 5. 通过基于隐形传态和 Bell 测量的量子信息传递—计算通路及用其实现的“远程 CNOT 门” [45,47,49]**，即在进行量子通信的同时，上述过程同时完成了量子计算** $|\mathbf{OUT}\rangle = U_{\mathbf{CNOT}}|\mathbf{T}\rangle|\mathbf{C}\rangle$**. 图中的态** $|\chi\rangle$ **是一个 4 量子位纠缠态：** $\frac{1}{2}((|0\rangle|0\rangle + |1\rangle|1\rangle)|0\rangle|0\rangle + (|0\rangle|1\rangle + |1\rangle|0\rangle)|1\rangle|1\rangle)$**，** $|\chi\rangle$ **可通过标准 Bell 态** $|\Phi^+\rangle$ **快速制备**[50].

图 6 以性能金字塔[35]的形式展示和对比了 QLA、CQLA 和 ECoPA 三种量子计算机体系结构之间在速度、可靠性和物理实现后所占体积（即用一般主流技术实现该体系结构时量子计算机的物理体积大小）之间的比较.

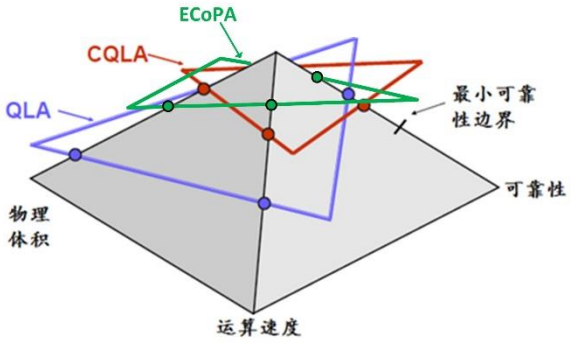

图 6. QLA、CQLA 与 ECoPA 体系结构在运算速度、可靠性和物理体积方面的对比（越靠近金字塔顶相关性能越高，各点的高低表示比较关系，不存在比例关系），最小可靠性边界指在现有的各种物理实现条件下若可靠性低于此边界则在理论上无法保证运行复杂量子程序的正确性

7　量子资源调度系统

如图 3 所示，量子计算机体系结构中包含一个重要的部分即量子资源调度系统. 从计算机体系结构的角度，量子资源的调度，主要包括底层（物理层）指令的调度以及量子逻辑门的调度. 这两者均涉及指令集体系结构（instruction set architecture，以下简称 ISA）.

ISA 是同时关联硬件与软件平台的中间体系结构，上层关联量子程序设计语言的处理系统，下层关联指令集系统，ISA 是与物理实现技术独立、并可被量子语言处理系统直接调用的一种结构性资源.

在应用程序层的量子指令均为施加在逻辑量子位上的逻辑指令，对应于量子程序的量子逻辑指令序列直观地反映量子算法本身而不含任何显式的容错或纠错命令. 这些逻辑指令在向体系结构的最底端——物理层进行映射时将会被自动分解为一系列具有容错或避错能力的基本操作序列（sequence of elementary operations，以下简称 SEO），SEO 是一种类似于汇编语言的低级量子语言，它根据当前量子计算物理实现平台的不同而变化——这保证了所提的体系结构可不经修改地在任何实际的量子计算设备上运行，只需给出设备对应的 SEO 即可.

关于物理层指令的调度，目前绝大多数已实现的量子计算物理平台均以量子线路模型为基础，故 SEO 在物理层对应的一般为基本的量子逻辑门（如 H，CNOT，X，Z，S 等），这些逻辑门的执行功效各不相同，某些逻辑门还对执行顺序有要求，执行后对系统产生的影响也各异. 那么如何在物理层通过调度逻辑门的执行次序，在保证执行结果正确的前提下尽可能减少对通信、容错等物理资源的消耗并最大化逻辑门执行的并行性也是值得研究的问题.

指令级并行（instruction-level parallelism，以下简称 ILP）的理论和技术在经典计算机体系结构中已非常成熟，但在量子计算中尚有一些问题. 这部分研究旨在提出一个运行于经典计算机的物理层量子指令调度系统，通过分析量子程序来精确预测运行中的通信、容错和计算负载，从而在保证程序运行正确性的前提下在容错阈值内合理调度物理层指令，尽可能最大化 ILP 程度并尽可能平衡用于通信和容错的系统资源.

8 量子程序设计语言及其处理系统

在经典计算机系统中，程序设计语言及其处理系统是系统软件的重要组成部分. 量子程序设计语言在量子计算机系统中有着同等的地位.

量子程序设计语言具有以下 3 个重要作用：有利于量子算法的理论性验证和复杂度分析、有利于量子（通信）协议的描述和验证、有利于量子算法和量子协议的实验实现.

截至 2014 年 6 月，已正式发表的量子程序设计语言约有 14 种，其中完成了处理程序设计的约有 8 种，见表 2.

表 2. 已正式发表的量子程序设计语言及相关信息[4]

语言名称	作 者	语言风范	扩展基础	最后更新	处理程序实现
Q-gol[5]	G. Baker	函数式	CaML	1998 年 8 月	是
Quantum C[51]	S. Blaha	命令式	ISOC	2000 年 6 月	否
qGCL[52]	J. Sanders 等	函数式	pGCL	2000 年 9 月	是
QML[52]	T. Altenkirch	函数式	ML	2001 年 11 月	否
Quantum Entanglement[6]	A. Gough	函数式	Perl	2002 年 6 月	是
The Q Language[53]	S. Bettelli	命令式	ISO C/C++	2003 年 5 月	是
QPL[54]	P. Selinger	函数式	—	2004 年 9 月	否
Quantum Predicative Programming[55]	A. Tafliovich	函数式	—	2006 年 12 月	否
NDQJava[56,57]	徐家福 等	命令式	Java	2006 年 7 月	是
QCL[58]	B. Ömer	命令式	C	2006 年 12 月	是
LanQ[59]	H. Mlnarik	命令式	C	2007 年 10 月	是
NDQFP[60]	徐家福 等	函数式	FP	2009 年 6 月	否
Quantum Flowchart Language[61]	应明生 等	命令式	Flowchart Language	2010 年 10 月	否
NDQJava-2[62]	徐家福 等	命令式	Java，NDQJava	2011 年 1 月	是
Quipper[63]	A. Green 等	函数式	Haskell	2013 年 6 月	是

量子程序设计语言用于书写量子程序以实现量子算法、量子协议和通用量子计算. 从程序理论、软件工程和程序设计人员的角度考虑，设计一个适用的量子程序设计语言及其处理系统应遵守如下的准则：

1、**正确性**. 所设计语言能正确完备描述量子计算和经典计算的元素集合、量子/经典数据结构和已知的所有量子算法. 语法和语义精确，程序无歧义性，语言的处理系统应能准确地将良定义的程序翻

[4] 表 2 中所列语言不包括针对专用绝热量子计算设备（如 D-Wave™）所设计的语言及处理程序以及量子伪码[12]（quantum pseudocode）.

[5] 见 http://www.rp.csiro.au/~gbaker/q-gol/.

[6] 见 http://search.cpan.org/dist/Quantum-Entanglement/.

译为目标代码.

2、**实用性**. 通用量子程序设计语言的实用性指所设计的语言可在一定的设备（经典设备、经典与量子的混成设备、纯量子设备）上实现，并能书写多种量子算法，所写程序均能在相应设备上运行.

3、**简明性**. 量子程序设计语言的简明性指语言的语法、结构、接口、文档应简单明了，以提高程序的易写、易读和易维护性. 徐家福等人提出关于量子语言简明性的四点意见：①区分内外性态；②减少接口信息；③记号通用易读；④文档扼要流畅.

4、**设备无关性**. 设备无关性指的是量子程序的书写和实现应与运行量子程序的目标物理设备无关，即硬件设备的实现方法（核磁共振、光学、离子阱等）、体系结构（纯量子、混成结构）和控制方式等与实现有关的部分均不影响量子程序的书写和处理. 量子程序设计语言的处理系统应可以将源程序自动翻译到平台相关的目标语言指令集.

5、**高层抽象**. 好的量子程序设计语言应具有较高的抽象层次，能够在语义层完全描述经典计算尤其是量子计算的特性.

6、**透明性**. 一个好的量子程序设计语言应当使一些量子计算中特有的技术和问题对用户（程序人员）保持透明，这些问题包括：①与物理实现相关的优化：包括代码优化、量子门优化和 SEO 顺序优化，由于这部分与设备平台和物理实现环境关系密切，用户无法也无需对非实时性的程序运行环境进行了解和控制；②测量顺序与再评估（re-evaluation）流程：量子程序中可能不只涉及一次测量，如果后续计算依赖中间测量的结果，则目标设备需进行多重测量. 另外由于多数量子算法是概率算法，量子计算在某些特定平台上的物理实现必须依赖多次评估和重做. 这两种过程都无需用户参与，量子程序设计语言的处理系统应当能自动处理上述两类情况；③程序执行过程中的容错与纠错：容错与纠错是量子计算中两个重要的问题，特别是在目前的技术条件下，在较大规模的量子计算中容错与纠错如被忽略则很难得到正确的计算结果. 然而容错与纠错不仅与算法的复杂程度、程序的规模和输入数据的规模有关，还与量子计算机的体系结构、物理实现方法和运行时的环境密切相关，很多相关参数都要在程序运行时加以确定，显然它们应当对用户透明.

7、**可经典模拟**. 量子程序设计语言虽然面向的最终目标是量子计算机但它应当首先支持在经典计算机上进行模拟. 经典模拟的优点是：①有利于量子程序的预验证和测试，经典计算机执行量子程序仅存在计算时间的差异而不会带来结果的差别，在量子设备执行之前先由经典计算机模拟（可适当缩小输入规模）不失为一个经济实用的选择；②有利于量子程序的调试和量子资源的规划：由于量子力学的理论限制，在量子设备中运行的量子程序在测量之前是不允许探视其各变量的状态的，这很不利于程序的调试. 在目前的技术情况下，实验中尚无法动态增减可用量子位个数，因此，量子程序在量子设备执行之前很有必要在经典计算机上模拟执行以确定需要的量子位数上限——虽然算法中可以直接看到所用的量子位数目，但执行时与系统状态动态相关的容错、纠错等所需的附加量子位数目并不易直接确定. 基于上述原因，即便有可用的量子计算机，依然有必要保留通用量子程序设计语言及其处理系统在可经典模拟上的兼容性.

相对于量子程序设计语言本身，量子语言处理系统的设计则更加综合：量子处理程序位于 ISA 及量子容错微体系结构（FTMA）之上的软件逻辑层中的最底层（图 7），对体系结构接口层和硬件逻辑层的依赖较少；它对整个体系结构的作用是：可通过量子线路优化方式将具有高层语义的量子语言转化为接口层的指令语言. 一个考虑整体可靠、易扩展的容错量子计算机体系结构的量子编译程序的设计流程如图 8 所示.

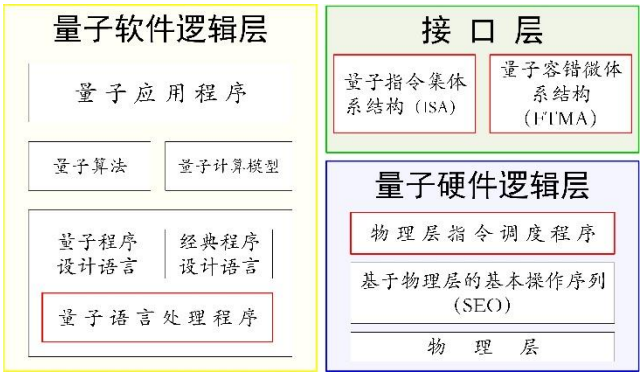

图 7. 与量子语言处理程序相关的量子计算机体系结构的抽象层次

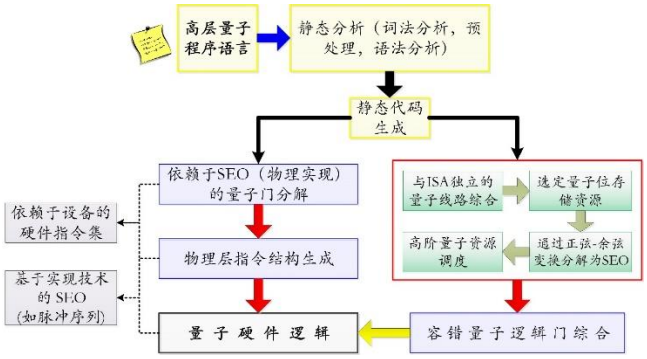

图 8. ECoPA 体系结构下量子编译程序的基本框图

9 量子软件系统

量子软件系统位于整个量子计算机系统的上层，虽然目前没有关于量子软件系统的明确定义和详细说明，但我们认为，量子软件系统应与经典计算机上的软件系统类似，可以分为量子系统软件和量子应用软件两个大类.

图 9 是从软件角度看量子计算机系统的抽象层次图，从下至上抽象度逐层加强.

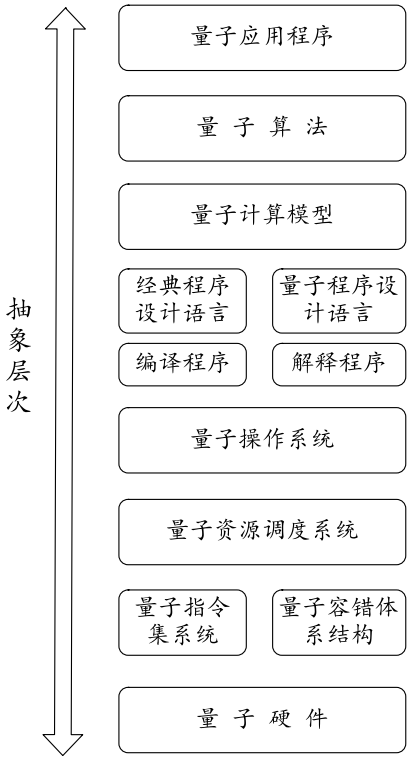

图 9. 从软件系统的角度看量子计算机系统

关于量子软件系统各层次的详述请参见文献[44]，关于量子应用程序的详述请见文献[64]. 此不赘述.

10 通用量子计算机的物理实现

通用量子计算机的物理实现是研究量子计算机的最重要的目标之一，同时也是最困难的部分之一.

自 1982 年量子计算机的设想被提出以来，已经提出了多种途径的物理实现方案，但截至目前仍没有任何一种方案可实现具有现实意义的通用量子计算. 由于 Feynman 提出的方案是基于用量子系统模拟的，因此早期的通用量子计算机的事先设想主要集中于通过某些物理量子体系（如光、粒子波动等）和一些特定的算法来仿真量子计算系统.

因为量子位的物理载体可分为易于实现量子纠缠和不易实现量子纠缠两类，因此从量子计算的物理实现机制上量子计算机可分为利用量子纠缠态和不利用量子纠缠态来进行量子计算两种. 2000 年 Lloyd 证明利用量子力学中粒子波动性所产生的相干效应可代替体现量子计算优越性的量子纠缠进行相关的量子计算[65]. 但是不利用量子纠缠的代价是使用的资源（如经典计算机、分束器、探测器等）呈指数级增加. 2001 年 Laflamme 等只用简单的透镜、反射镜等线形光学器件去处理一束没有纠缠的光，形成了一个不需要纠缠态的光量子计算机的理论模型[66]. 该思想在 2005 年由 Walther 等改进后在单向光量子计算实验中成功实现了小规模 Grover 算法[67]. 2005 年以来，离子阱、硅基半导体、超导约瑟夫森结、量子点等新的量子计算实现方式也相继在实验室成功实现. 2001 年 IBM Almaden 研究中心 Vandersypen 等采用 NMR 系综量子计算实现了 Shor 算法的小规模演示[68]，2007 年哈佛大学等单位利用钻石中氮—空穴色心（N-V color center）实现了固态量子位态[69]; 2009 年耶鲁大学首先实现了超导量子计算中的 Grover 和 Deutsch-Jozsa 算法[70]，超导量子计算被认为是“最自然”的量子计算物理实现；2011 年，奥地利 Monz 等报道了基于离子阱的 14 量子位相干和纠缠量子系统[29]，是目前公开发表的最多量子位的可控相干系统.

关于通用量子计算机的物理实现的细节请参考文献[28,57].

同时，量子计算的物理实现也是十分困难的. 其主要障碍依然在于现实系统对环境封闭效果不佳，从而导致退相干的影响显著增强. 另外目前一些测量仪器（如核磁共振谱仪）的精确度还不足以

探测单个或几个量子位载体表达的量子态，只能用系综方式模拟单量子位或若干量子位的态，这种方式虽然被证明不影响测量结果但会在探测信号中引入噪音，加大了信号解读的难度.

研究可靠的通用量子计算模型及量子计算机体系结构可以促进量子计算机物理实现的发展. 一方面，量子计算模型和量子计算机体系结构的研究离不开量子计算的物理实现和量子计算机的物理基础，后者的研究和进展可为前者的研究提供更高层次的视角和诠释；另一方面量子计算模型和量子计算机体系结构的研究也可指导物理层的实现. 近年来，绝热量子计算、单向量子计算等新型量子计算模型均在相应物理平台上成功实现，这些实现都是在理论模型的基础上研究成功的. 这表明量子计算的模型和理论与物理实现两者可以相互促进，共同发展.

11 展 望

从目前的研究现状看来，通用量子计算机的实现之路并不平坦；从未来相当长的时间来看，量子计算机的研究依然是希望与挑战并存.

从理论上看，量子计算机实现的最大挑战依然是退相干带来的一系列困难，如难以提高量子体系的可靠性与易扩展性、用于量子存储和计算的容错和纠错代价过大、层次化容错微体系结构过于复杂等等. 这些问题虽然在实际设计和实现技术中有一些规避和松弛的途径，但就人们对量子计算机的计算能力需求来看，这些方案依然有很多缺陷.

从技术上看，是先研制通用量子计算机还是先研制专用量子计算设备？是先研制数字量子计算机还是先研制模拟量子计算机？量子指令系统应该采用精简指令系统还是复杂量子指令系统？对这些问题的争论一直持续，且没有最终的答案.

未来或许可以通过以下一些努力去改变上述困难的现状：

1、 提高量子体系的可靠性：可考虑采用分级容错体系结构和分级量子纠错码的方法来平衡容错带来的时间和物理开销，同时考虑在存储器中引入零退相干子空间等方法来提高系统本身对退相干的免疫性.

2、 提高量子体系的可扩展性：可考虑采用量子长距作用的效应或设备（如量子隐形传态、量子长线等）作为不同量子存储或计算设备的扩展媒介，同时应考虑可扩展性与可靠性的平衡. 在最近的报道中，已实现利用单个量子位载体（如光子）的多个自由度进行多自由度隐形传态和超密传输[71]，由于各自由度之间存在一定程度的物理隔离，这可能也有助于提高量子体系的可扩展性。

3、 加强量子算法的研究：与经典算法相比，目前提出的量子算法无论在解决问题的类型还是解决问题的广度方面都远不及前者，特别是近 15 年来量子算法的研究事实上滞后于量子计算物理设备的研究，这是值得具有计算机科学背景的量子计算机工作者值得努力的方向.

4、 兼顾专用量子计算设备的发展：虽然通用量子计算机的研制是量子计算机研究的最终目标之一，但在实际需求中，专用量子计算设备相比通用量子计算机在体系结构、控制方式和易用性方面均有优势. 在通用量子计算机的研制尚处于起步阶段或面临困难之时，兼顾发展专用量子计算机也具有较高的理论或实际价值.

5、 寻求量子计算机的发展点：推动量子计算机发展的原动力应该是当前经典计算机处理代价巨大的复杂问题. 相比于二十世纪八十年代，大规模计算平台有了较大的发展，过去许多难以用经典计算机处理的问题如蛋白质结构预测、人工地震波处理等目前均可在大型机中处理. 因此寻找适用于量子计算机求解的问题也是未来工作之一. 最近的研究已提出利用基于量子计算机的机器学习算法提高模式识别的效果[72]和利用量子搜索提高软件工程中某些依赖搜索的算法的效率[73]，这些实际需求在未来都可能成为推动量子计算机研究和发展的力量。

Van Meter 等人 2013 年曾在《美国计算机协会通信（CACM）》中发出疑问：量子计算机何时可用来服务科学而不是仅仅成为一种科学[74]？这个问题的答案或许能很快揭晓，或许还需要几代人的不懈努力. 但无论如何，量子计算机的研制一定不是单一学科的努力就可以完成的，它需要数学、物理学、计算机科学、电子学、材料科学和工程技术等众多领域共同协作下才能得到发展.

伴随着人们对量子计算、量子通信和量子信息处理的需求越来越迫切，许多国家已经启动关于量子计算和量子计算机研究的中长期规划，越来越多的大型企业开始成立致力于量子信息科学和技术的研究机构. 中国也已启动了数个关于量子计算、

量子通信和量子调控等领域的重大研究计划和协同创新中心.

迎接大力的支持和热情的期盼的是否是通向胜利之路上的曙光？“即将实现”、“仍需 20 到 30 年”、“永远无法成功”这三个截然不同的回答时刻萦绕在量子计算机研究者的耳边.

致 谢：作者感谢南京大学计算机科学与技术系徐家福教授对本文研究内容的指导和对文字的精心修改. 本文研究内容受国家自然科学基金委重大研究计划（集成项目）、创新研究群体科学基金、青年基金项目和教育部基础研究项目博士点基金的资助. 本文在写作过程中还得到了胡海星和王琨的帮助，以在此一并深表谢忱！

参 考 文 献

[1] Turing A M. On computable numbers, with an application to the Entscheidungsproblem. Journal of Mathematics, 1936, 58(345-363): 5.

[2] Church A. A note on the Entscheidungsproblem. The journal of symbolic logic, 1936, 1(01): 40-41.

[3] Moore G E. Cramming more components onto integrated circuits. Electronics Magazine, 1965, 38(8): 114-117.

[4] Benioff P. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. Journal of Statistical Physics, 1980, 22(5): 563-591.

[5] Feynman R P. Simulating physics with computers. International journal of theoretical physics, 1982, 21(6): 467-488.

[6] Deutsch D. Quantum theory, the Church-Turing principle and the universal quantum computer//Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. The Royal Society, 1985, 400(1818): 97-117.

[7] Nielsen M A, Chuang I L. Quantum computation and quantum information. Cambridge, UK: Cambridge university press, 2010.

[8] Feynman R P, Leighton R B, Sands M. The Feynman Lectures on Physics, Desktop Edition Volume I. New York, USA: Basic Books, 2013.

[9] Einstein A, Podolsky B, Rosen N. Can quantum-mechanical description of physical reality be considered complete?. Physical review, 1935, 47(10): 777.

[10] Deutsch D. Quantum computational networks. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 1989, 425(1868): 73-90.

[11] Yao A C C. Quantum circuit complexity. //Proceedings of 34th IEEE Annual Symposium on Foundations of Computer Science, Palo Alto, USA, 1993: 352-361.

[12] Knill E. Conventions for quantum pseudocode. Technical Report LAUR-96-2724, Los Alamos National Laboratory, 1996.

[13] Farhi E, Goldstone J, Gutmann S, et al. Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106, 2000.

[14] Freedman M, Kitaev A, Larsen M, et al. Topological quantum computation. Bulletin of the American Mathematical Society, 2003, 40(1): 31-38.

[15] Van Dam W, Mosca M, Vazirani U. How powerful is adiabatic quantum computation? //Proceedings of 42nd IEEE Symposium on Foundations of Computer Science, Las Vegas, USA, 2001: 279-287.

[16] Sarandy M S, Lidar D A. Adiabatic quantum computation in open systems. Physical review letters, 2005, 95(25): 250503.

[17] Kitaev A Y. Fault-tolerant quantum computation by anyons. Annals of Physics, 2003, 303(1): 2-30.

[18] DiVincenzo D P. Two-bit gates are universal for quantum computation. Physical Review A, 1995, 51(2): 1015.

[19] Lloyd S. Almost any quantum logic gate is universal. Physical Review Letters, 1995, 75(2): 346.

[20] Kitaev A Y. Quantum computations: algorithms and error correction. Russian Mathematical Surveys, 1997, 52(6): 1191-1249.

[21] Toffoli T. Reversible computing. Heidelberg, Germany: Springer Berlin Heidelberg, 1980.

[22] Wu N, Song F. A novel kind of architecture with high-efficiency and error-tolerance of universal quantum computer. Chinese Journal of Computers, 2009, 32(1): 161-168.

[23] Qiu D, Li L, Mateus P, et al. Exponentially more concise quantum recognition of non-RMM regular languages. Journal of Computer and System Sciences, 2015, 81(2): 359-375.

[24] Zou X, Qiu D, Li L, et al. Semiquantum-key distribution using less than four quantum states. Physical Review A, 2009, 79(5): 052312.

[25] Wu N, Song F, Li X. Study of improved semantics on quantum elements and programmable architecture. //Proceedings of SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, Baltimore, USA, 2012: 84000K-84000K-8.

[26] Loss D, DiVincenzo D P. Quantum computation with quantum dots. Physical Review A, 1998, 57(1): 120.

[27] Divincenzo D P. The Physical Implementation of Quantum Computation. Fortschritte der Physik, 2000, 48: 771-783.

[28] Wu N. Research on the Models and Architecture of Reliable Quantum Computers[博士学位论文]. Nanjing: Nanjing University, 2009. (吴楠. 可

靠量子计算机模型及体系结构的研究[博士学位论文]. 南京：南京大学, 2009)

[29] Monz T, Schindler P, Barreiro J T, et al. 14-qubit entanglement: Creation and coherence. Physical Review Letters, 2011, 106(13): 130506.

[30] Ladd T D, Jelezko F, Laflamme R, et al. Quantum computers. Nature, 2010, 464(7285): 45-53.

[31] Yao X C, Wang T X, Chen H Z, et al. Experimental demonstration of topological error correction. Nature, 2012, 482(7386): 489-494..

[32] Ansmann M, Wang H, Bialczak R C, et al. Violation of Bell's inequality in Josephson phase qubits. Nature, 2009, 461(7263): 504-506.

[33] Amdahl G M, Blaauw G A, Brooks F P. Architecture of the IBM System/360. IBM Journal of Research and Development, 1964, 8(2): 87-101.

[34] Metodi T S, Thaker D D, Cross A W, et al. A quantum logic array microarchitecture: Scalable quantum data movement and computation. //Proceedings of 38th Annual IEEE/ACM International Symposium on Microarchitecture, Barcelona, Spain, 2005: 305-318.

[35] Meter R, Oskin M. Architectural implications of quantum computing technologies. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2006, 2(1): 31-63.

[36] Isailovic N, Whitney M, Patel Y, et al. Running a quantum circuit at the speed of data. ACM SIGARCH Computer Architecture News, 2008, 36(3): 177-188.

[37] Isailovic N, Patel Y, Whitney M, et al. Interconnection networks for scalable quantum computers. ACM SIGARCH Computer Architecture News, 2006, 34(2): 366-377.

[38] Politi A, Matthews J C F, O'Brien J L. Shor's quantum factoring algorithm on a photonic chip. Science, 2009, 325(5945): 1221-1221.

[39] Schumacher B. Quantum coding. Physical Review A, 1995, 51(4): 2738.

[40] Santos M. Short-time critical dynamics for the transverse Ising model. Physical Review E, 2000, 61(6): 7204.

[41] Hennessy J L, Patterson D A. Computer architecture: a quantitative approach. Amsterdam, Netherlands: Elsevier, 2012.

[42] Wu N, Song F, Li X. An improved architecture of a realizable quantum computer for quantum programming languages. //Proceedings of SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, Orlando, USA, 2009: 73420J-73420J-10.

[43] Wu N, Song F, Li X. A new software-based architecture for quantum computer. //Proceedings of SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, Orlando, USA, 2010: 77020T-77020T-7.

[44] Wu N, Hu H, Song F, et al. Quantum software framework: a tentative study. Frontiers of Computer Science, 2013, 7(3): 341-349.

[45] Oskin M, Chong F T, Chuang I L, et al. Building quantum wires: the long and the short of it. //Proceedings of 30th IEEE Annual International Symposium on Computer Architecture, San Diego, USA, 2003: 374-385.

[46] Gottesman D, Chuang I L. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. Nature, 1999, 402(6760): 390-393.

[47] Raussendorf R, Briegel H J. A one-way quantum computer. Physical Review Letters, 2001, 86(22): 5188.

[48] Nielsen M A. Quantum computation by measurement and quantum memory. Physics Letters A, 2003, 308(2): 96-100.

[49] Metodi T S, Faruque A I, Chong F T. Quantum Computing for Computer Architects. Synthesis Lectures on Computer Architecture, 2011, 6(1): 1-203.

[50] Goebel A M, Wagenknecht C, Zhang Q, et al. Teleportation-Based Controlled-NOT Gate for Fault-Tolerant Quantum Computation. arXiv preprint arXiv:0809.3583, 2008.

[51] Goebel A M, Wagenknecht C, Zhang Q, et al. Teleportation-Based Controlled-NOT Gate for Fault-Tolerant Quantum Computation. arXiv preprint arXiv:0809.3583, 2008.

[52] Sanders J W, Zuliani P. Quantum programming. Mathematics of Program Construction. Heidelberg, Germany: Springer Berlin Heidelberg, 2000: 80-99.

[53] Bettelli S, Calarco T, Serafini L. Toward an architecture for quantum programming. The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics, 2003, 25(2): 181-200.

[54] Selinger P. Towards a quantum programming language. Mathematical Structures in Computer Science, 2004, 14(04): 527-586.

[55] Tafliovich A, Hehner E C R. Quantum predicative programming. Mathematics of Program Construction. Heidelberg, Germany: Springer Berlin Heidelberg, 2006: 433-454.

[56] Xu J, Song F, Qian S, et al. Quantum programming language NDQJava. Journal of Software, 2008, 19(1): 1-8. (徐家福，宋方敏，钱士钧，等. 量子程序设计语言 NDQJava. 软件学报, 2008, 19(1): 1-8)

[57] Jiao Y, Wu N, Song F. Quantum assembler and interpreter of NDQJava processing system. Journal of Nanjing University: Nat Sci Ed, 2008, 44(2): 107-115. (焦阳，吴楠，宋方敏. NDQJava 语言处理系统量子汇编及解释程序. 南京大学学报：自然科学版, 2008, 44(2): 107-115)

[58] Ömer B. A procedural formalism for quantum computing [Master Dissertation]. Vienna: Technical University of Vienna, 1998.

[59] Mlnarik H. Operational semantics and type soundness of quantum programming language LanQ. arXiv preprint arXiv:0708.0890, 2007.

[60] Xu J F, Song F M. Quantum programming languages: A tentative study. Science in China Series F: Information Sciences, 2008, 51(6): 623-637.

[61] Ying M, Feng Y. A flowchart language for quantum programming. Software Engineering, IEEE Transactions on, 2011, 37(4): 466-485.
[62] Liu L, Xu J. Quantum programming language NDQJava-2. Journal of Software, 2011, 22(5): 877-886. (刘玲, 徐家福. 量子程序设计语言 NDQJava-2. Journal of Software, 2011, 22(5): 877-886)
[63] Green A S, Lumsdaine P L F, Ross N J, et al. Quipper: a scalable quantum programming language. ACM SIGPLAN Notices, 2013, 48(6): 333-342.
[64] Wu N, Song F, Li X. Study of a Quantum Framework for Search Based Software Engineering. International Journal of Theoretical Physics, 2013, 52(6): 2181-2186.
[65] Lloyd S. Quantum search without entanglement. Physical Review A, 1999, 61(1): 010301.
[66] Knill E, Laflamme R, Milburn G J. A scheme for efficient quantum computation with linear optics. nature, 2001, 409(6816): 46-52.
[67] Walther P, Resch K J, Rudolph T, et al. Experimental one-way quantum computing. Nature, 2005, 434(7030): 169-176.
[68] Vandersypen L M K, Steffen M, Breyta G, et al. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature, 2001, 414(6866): 883-887.
[69] Dutt M V G, Childress L, Jiang L, et al. Quantum register based on individual electronic and nuclear spin qubits in diamond. Science, 2007, 316(5829): 1312-1316.
[70] DiCarlo L, Chow J M, Gambetta J M, et al. Demonstration of two-qubit algorithms with a superconducting quantum processor. Nature, 2009, 460(7252): 240-244.
[71] Wang X L, Cai X D, Su Z E, et al. Quantum teleportation of multiple degrees of freedom in a single photon. Nature, 2015, 518: 516-519.
[72] Cai X D, Wu D, Su Z E, et al. Entanglement-Based Machine Learning on a Quantum Computer. Physical review letters, 2015, 114(11): 110504.
[73] Wu N, Song F, Li X. Quantum search-based software engineering: An exploratory study. Scientia Sinica Informationis, 2015, 45: 623-633. (吴楠, 宋方敏, LI XiangDong. 基于量子搜索的软件工程. 中国科学: 信息科学, 2015, 45: 623-633)
[74] Van Meter R, Horsman C. A blueprint for building a quantum computer. Communications of the ACM, 2013, 56(10): 84-93.

WU Nan was born in 1981. He received his BSc and Ph.D. from the Department of Computer Science, Nanjing University, China in 2004 and 2009, respectively. Currently, he is an assistant professor at the department of computer science and technology in Nanjing University. His research interest includes quantum computing, quantum computer architecture and quantum information. He was a research scholar at the Graduate School of the City University of New York between 2008 and 2009. He is a member of ACM, IEEE, SPIE and CCF.

SONG Fang-Min received his BSc, MSc and Ph.D. from the Department of Mathematics and Department of Computer Science at Nanjing University, China in 1982, 1985 and 1988, respectively. He took the position of postdoc research fellow at ETH and CTH in 1993 and 1994, respectively. Currently, he is a full professor at the department of computer science and technology in Nanjing University. His research mainly focus on the symbolic logic, quantum algorithm and quantum computation. He is a member of CCF.

LI Xiang-Dong received his M.S. in Computer Information Science from CUNY Brooklyn College in 1997, and Ph.D. in physics from the CUNY Graduate School in 2000. He is an associate professor at the Department of Computer Systems Technology in New York City College of Technology, CUNY. He is a faculty member of both Ph.D. programs in Computer Science and Physics at the CUNY Graduate School. His research fields include information security and quantum information. He is a member of APS.

Background

Quantum computation is a field of science and technology, combining and drawing on the disciplines of physics, mathematics, computer science and engineering. Indeed, scientists predicate that within next 20 to 40 years there will be quantum computers. The universal quantum computer is therefore worthy of study. Today, most of the physically implemented quantum devices can deal with only some domain-specific problems. People have been trying to build a universal quantum computer, which would be able to solve all kinds of problems on a single architecture. In recent years, quantum algorithms, quantum computational models and physical implementation of universal quantum computers have been studied. The results of these studies provide the necessary theoretical and experimental basis to design the scalable universal quantum computer with new features such as fault tolerance, high performance and high-fidelity teleportation, etc. These features can also provide quantum operating system, quantum programming languages and even quantum application software.