

基于量子汇编指令的量子线路设计

本章核心知识点：

- ☐ 量子汇编指令语言 OpenQASM
- ☐ OpenQASM 量子线路的设计与调试
- ☐ 量子逻辑门
- ☐ 量子加法器
- ☐ 量子相位反冲

5.1 量子汇编指令语言 OpenQASM

5.1.1 OpenQASM 语言基本语句

不同开发平台中的量子线路编辑器或模拟器支持的 OpenQASM 语句格式会稍有区别。本书以 Quantum Composer 支持的 OpenQASM 2.0 为例,书中所有 OpenQASM 代码实例均可在 Quantum Composer 中测试及运行。

表 5.1 列出了 OpenQASM 的基本语句及其简要说明。

表 5.1 OpenQASM 语言基本语句 (Ver2.0)

语 句	说 明
OPENQASM 2.0;	表明本文件符合 OpenQASM 2.0 版本格式
include "filename";	打开并解析另一个源文件
qreg qregname[size];	声明一个量子寄存器, size 为量子比特个数。 例: qreg q[5];
qregname[i]	引用一个量子寄存器。例: q[0]

续表

语 句	说 明
creg cregname[size];	声明一个经典寄存器,size 为经典比特个数。 例: creg c[5];
cregname[i]	引用一个经典寄存器。例: c[0]
U(theta, phi, lambda) qubit qreg;	U 门。例: U(pi/2, pi/3, 0) q[0];
CX qubit qreg, qubit qreg;	CNOT 门。例: CX q[0], q[1];
measure qubit qreg->bit creg;	在计算基下测量。例: measure q -> c;
reset qubit qreg;	将量子比特或量子寄存器重置为 0>。 例: reset q[0];
gatename(params) qargs;	引用一个酉门。例: crz(pi/2) q[1], q[0];
if(creg==int) qop;	条件满足时,执行量子操作。例: if(c==5) CX q[0], q[1];
barrier qargs;	barrier 操作(qargs 是以逗号分隔的量子比特或量子寄存器列表)。例: barrier q[0], q[1];
//comment text	注释文本

- 单量子比特门和多量子比特门的 OpenQASM 语句请参见表 3.1 和表 4.1。OpenQASM 具有简明易读的语法。
- ① 每条语句以分号结束,大小写敏感,多余的空格会被忽略。
 - ② 第一条语句必须是“OPENQASM M.m;”,其中,M 为主版本号,m 为次版本号。当前版本的 Quantum Composer 支持的 OpenQASM 的版本号为 Version 2.0,故第一条语句必须为“OPENQASM 2.0”。
 - ③ 接下来是 include 语句“include "filename";”。include 语句的用法与 C 语言类似,其功能是将头文件插入该命令所在的位置,从而把头文件和当前源文件连接成一个源文件。Quantum Composer 中必须有“include "qelib1.inc";”语句。
 - ④ 接下来声明量子寄存器和经典寄存器。量子寄存器中的各量子比特初始化为|0>;经典寄存器中的各比特初始化为 0。各类标识符必须以小写字母开头,合法字符为西文半角的数字、字母和下画线。
 - ⑤ 添加量子门,设计量子线路。
 - ⑥ 注释以“//”开头。

5.1.2 OpenQASM 量子线路编程实例

【例 5.1】 请给出如下 OpenQASM 代码对应的量子线路图。

CH5-1.qasm: Bell 态观测

```
//OpenQASM 文件
OPENQASM 2.0;           //指明 OpenQASM 的版本号
include "qelib1.inc";    //包含头文件 qelib1.inc
qreg q[2];               //声明一个名为 q 的量子寄存器,带 2 个量子比特
creg c[2];               //声明一个名为 c 的经典寄存器,带 2 个经典比特
h q[0];                  //在量子比特 q[0]上添加一个 H 门
cx q[0],q[1];            //CX 门,控制量子比特为 q[0],目标量子比特为 q[1]
measure q[0] -> c[0];    //对量子比特 q[0]进行测量,对应经典比特为 c[0]
measure q[1] -> c[1];    //对量子比特 q[1]进行测量,对应经典比特为 c[1]
```

本例程对应的量子线路如图 5.1 所示,其实现了 Bell 态的制备与测量(详见 5.2.1 节)。

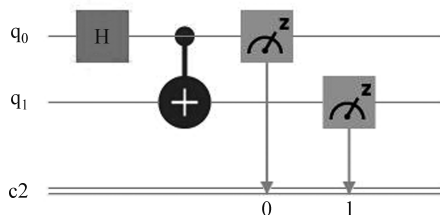


图 5.1 Bell 态的制备与测量

【例 5.2】 请给出如图 5.2 所示的量子线路的 OpenQASM 代码。

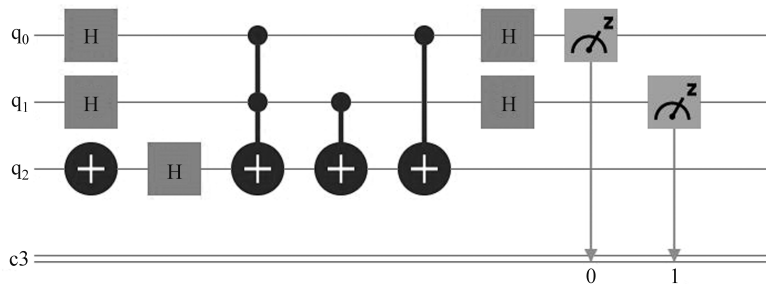


图 5.2 DJ 算法的量子线路($n=2$)

解：图 5.2 所示的量子线路为 Deutsch-Jozsa 算法(简称 DJ 算法,详见 7.1 节)的一种实现。该线路声明了一个 3 量子比特的量子寄存器和一个 3 比特的经典寄存器,对应的 OpenQASM 代码如下所示。

CH5-2.qasm:DJ 算法 ($n=2$)

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[3];
creg c[3];
h q[0];
h q[1];
x q[2];
h q[2];
ccx q[0],q[1],q[2];
cx q[1],q[2];
cx q[0],q[2];
h q[0];
h q[1];
measure q[0]->c[0];
measure q[1]->c[1];
```

5.1.3 图形化量子线路开发工具 Quantum Composer

Quantum Composer 的用户界面如图 5.3 所示。

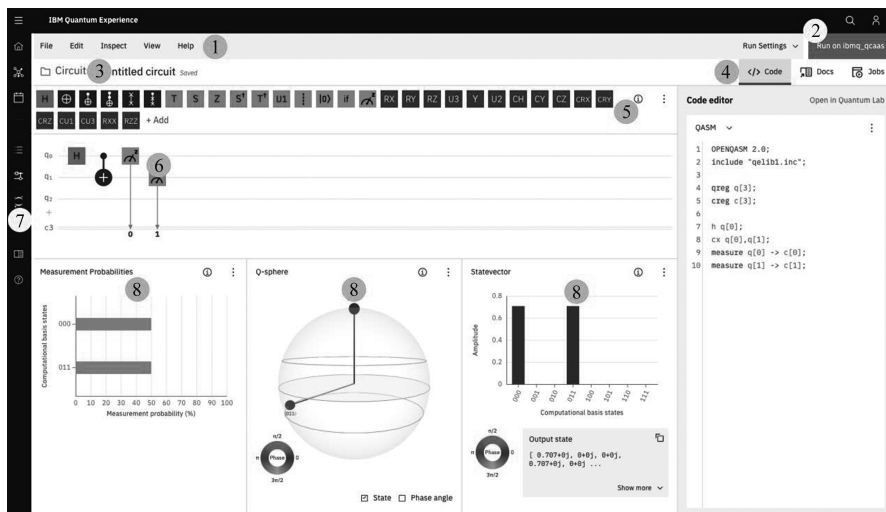


图 5.3 Quantum Composer 的用户界面

表 5.2 给出了用户界面各功能区的简要说明(注:随着版本的更新,界面会有所调整,但其功能结构基本一致)。

表 5.2 Quantum Composer 用户界面功能区说明

标识号	功能区名称	说 明
①	菜单栏	单步调试(inspect)和视图选择(view)等功能
②	线路运行参数设置	指定线路执行的后端(backend)和运行次数(shots)等
③	线路名称设置	命名量子线路并以同名的 OpenQASM 文件存储
④	OpenQASM 代码编辑器	编辑 OpenQASM 代码,生成相应的量子线路
⑤	量子门线路符号列表	拖曳线路符号以在量子线路中插入量子门;可增加自定义门
⑥	量子线路图形化编辑器	通过线路符号创建量子线路;编辑、修改量子线路
⑦	应用切换 (Application Switcher)	在 Quantum Composer、Quantum Lab、Services(后台实体机和模拟器列表等)、Jobs(工作任务)和 Documentation(各类文档)等之间进行切换
⑧	实时可视化窗口 (Live Visualizations)	以多视图的形式展示量子线路的实时结果

5.2 OpenQASM 量子线路设计与调试

5.2.1 Bell 态观测实验

Bell 态(Bell states)指双量子比特系统的四种最大纠缠态,式(5.1)给出了这四种态,它得名于贝尔不等式的提出者——爱尔兰物理学家约翰·斯图尔特·贝尔。

$$\begin{cases} |\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2} \\ |\Phi^-\rangle = (|00\rangle - |11\rangle)/\sqrt{2} \\ |\Psi^+\rangle = (|01\rangle + |10\rangle)/\sqrt{2} \\ |\Psi^-\rangle = (|01\rangle - |10\rangle)/\sqrt{2} \end{cases} \quad (5.1)$$

【实验内容】

- ① 基于自定义门 nG0 构建如图 5.4 所示的 Bell 态观测量子线路(5.2.2 节)。

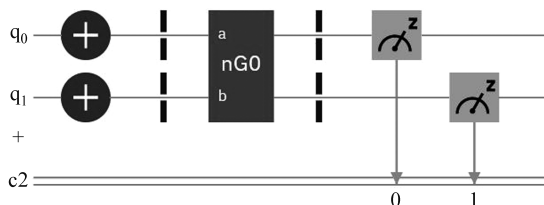


图 5.4 基于 nG0 门的 Bell 态观测量子线路

- ② 在模拟器上运行该量子线路,对结果做可视化分析(5.2.3 节)。
- ③ 在远程实体机上运行该量子线路,对结果做可视化分析(5.2.4 节)。
- ④ 比较以上两步的运行结果。请思考分析:为何用两种不同的运行环境执行同一量子线路时执行结果会有差异?

本节后续内容将以该实验为例介绍 Quantum Composer 量子线路的开发流程。

Quantum Composer 量子线路的开发流程主要包括以下三个步骤:

- ① 量子线路的创建(5.2.2 节和 5.2.3 节);
- ② 选择指定后端执行量子线路(模拟器作为后端见 5.2.6 节;后台真实量子计算机作为后端见 5.2.7 节);
- ③ 结果的可视化与分析(5.2.5 节)。

Quantum Composer 还提供了对量子线路的单步调试功能(5.2.4 节)。

5.2.2 OpenQASM 自定义门的构建

除了代码编辑器提供的基本门汇编指令以外,还可将一条或多条量子汇编指令封装成一个新的门。自定义门用来实现一个可以重复使用的独立功能,类似于汇编语言或高级语言中函数的用法。OpenQASM 使用 `gate` 指令定义一个新的门,其指令格式为

```
gate gatename(params) qargs;
```

其中,`gate` 为指令助记符,`gatename` 是新定义的门的名称;`params` 是自定义门需要指明的参数,为可选项;`qargs` 是自定义门作用的量子比特名称的形式参数列表。

【例 5.3】 请将如图 5.5 所示的量子线路构造为一个自定义门 `nG0`。

解: 实现自定义门 `nG0` 的 OpenQASM 代码如下。

CH5-3.qasm: 自定义门 nG0

```
OPENQASM 2.0;
include "qelib1.inc";
gate nG0 a, b
{
    h a;                //对量子比特 a 进行 H 门操作
    cx a, b;            //CNOT 门, a 为控制量子比特, b 为目标量子比特
}
```

以上代码新定义了一个名为 nG0 的量子门,形式参数 a 和 b 表示自定义门作用的量子比特。

在代码编辑器中输入以上代码后,量子线路图形编辑器 Quantum Composer 的量子门线路符号列表中将新增如图 5.6 所示的 nG0 门的线路符号。

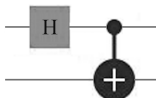


图 5.5 自定义门 nG0 的量子线路

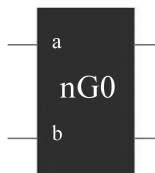


图 5.6 自定义门 nG0 的线路符号

5.2.3 量子线路图的输入与编辑

1. 图形化用户界面方式

按照图 5.4 所示的 Bell 态观测量子线路,可在 Quantum Composer 的量子线路图形化编辑器(图 5.3 的区域⑥)中录入该线路。选中量子线路中的某一量子门线路符号,右击再选择“编辑”选项,即可调整该量子门的外联方式和内部参数等。

2. OpenQASM 代码文本方式

在 OpenQASM 代码编辑器(图 5.3 的区域④)中录入如下代码以生成量子线路。

#CH5-4.qasm: 基于自定义门的 Bell 态观测

```
OPENQASM 2.0;
include "qelib1.inc";
gate nG0 a, b
{
    h a;
    cx a, b;
}
qreg q[2];
creg c[2];
x q[0];                      //初态设定,测试中根据需要修改
x q[1];                      //初态设定,测试中根据需要修改
barrier q[0],q[1];
nG0 q[0], q[1];
barrier q[0],q[1];
measure q[0] -> c[0];
measure q[1] -> c[1];
```

自定义门 nG0 的 a 映射在量子比特 q_0 上, b 映射在量子比特 q_1 上。 q_1 为高位, q_0 为低位, 量子态记为 $|q_1q_0\rangle$ 。当初态为 $|11\rangle$ 时, Quantum Composer 中实测的末态与 $|\Psi^-\rangle$ 相差一个整体全局相位因子 -1 。改变初态设定, 利用构造的自定义门 nG0 可得到式(5.1)所列的四种态, 即

$$\begin{aligned} |00\rangle &\rightarrow |\Phi^+\rangle, |01\rangle \rightarrow |\Phi^-\rangle \\ |10\rangle &\rightarrow |\Psi^+\rangle, |11\rangle \rightarrow |\Psi^-\rangle \end{aligned}$$

特别提醒: 对于 Quantum Composer 中的量子门线路符号、量子寄存器或量子线路的 n 个量子比特 $q_0, q_1, \dots, q_{n-1}$, 其量子态记为 $|q_{n-1} \dots q_0\rangle$, 其中 q_{n-1} 为高位, q_0 为低位。

5.2.4 量子线路的单步调试

对量子线路进行单步调试依赖于模拟器。单步调试功能用来逐步查看量子态的演化过程。

在菜单栏(图 5.3 的区域①)选择 Inspect 选项可对当前显示的量子线路进行单步调试。图 5.7 所示为单步调试的用户界面, 加载的量子线路为例 5.1 中的 CH5-1.qasm。单击控制面板上的左右箭头符号可令量子线路上的蓝色框左右移动。蓝色框标示了当前选中的量子门, 实时可视化分析工具将显示当前量子门作用后的线路状态, 其中的各类视图也会随之发生变化。

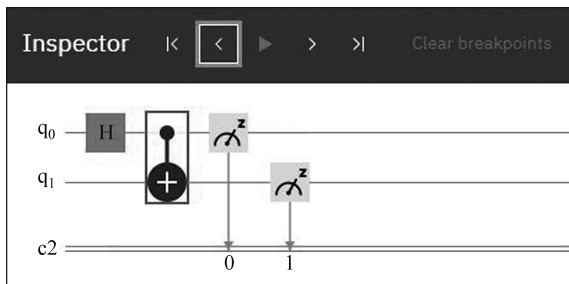


图 5.7 单步调试界面

5.2.5 结果实时可视化与分析

实时可视化区域(Live Visualizations)出现在 Quantum Composer 工作区底部的窗口中(图 5.3 的区域⑧),它显示的是当前量子线路的实时状态,提供了状态向量视图(Statevector View)、测量概率视图(Probabilities View)和 Q 球视图(Q-sphere View)等结果可视化分析工具。在菜单栏的 View 选项中可选择或取消相关的视图。

在真机上运行时,实时可视化分析工具不起作用。下面介绍 Bell 态观测量子线路在 ibmq_qasm_simulator 模拟器上运行后的实时可视化分析视图。实时结果反映的是图 5.7 所示的 Bell 线路。该线路的初态为 $|00\rangle$,生成纠缠态 $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ 。单步调试时,断点位于 CNOT 门上的量子线路的状态。

1. 状态向量视图

图 5.8 所示为 Statevector 视图中的量子态及其振幅的直方图。横轴为计算基态,纵轴表示各计算基态的振幅值。该图表明:结果中仅出现了 $|00\rangle$ 和 $|11\rangle$,各自的振幅约为 0.707。

图 5.9 为状态向量视图中的另一部分,位于图 5.8 所示的直方图的下方。左图用色轮的颜色表示相位角,与图 5.8 所示的直方图配合使用,用颜色区分直方图各状态的相位角。对照色轮,可知图 5.8 所示的直方图中的 $|00\rangle$ 和 $|11\rangle$ 的相位角均为 0。图 5.9 的右图中显示的 Bell 态 $|\Phi^+\rangle$ 的状态向量为 $[0.707+0j, 0+0j, 0+0j, 0.707+0j]$ 。

格式说明: IBM Quantum Composer 的界面或 Quantum Lab 代码运行结果中输出的量子态状态向量会将各分量的格式显示为 $a+bj$ (a, b 为实数),其表示复数 $a+bi$ (a, b 为实数);下文中,在用数学原理描述量子态状态向量时,其分量采用 $a+bi$ (a, b 为实数)的格式,但在说明软件界面输出或代码运

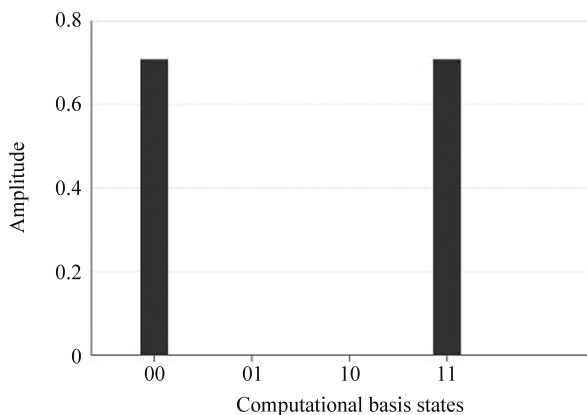


图 5.8 量子态及其对应的幅值直方图

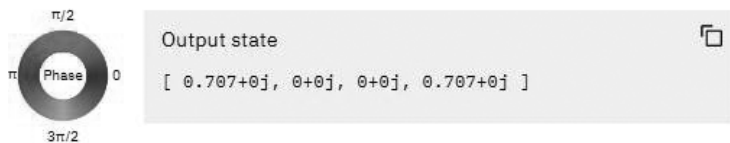


图 5.9 色轮与状态向量

行结果时,将直接采用系统输出结果,以保证书中内容与软件界面的显示或代码输出的运行结果一致,从而便于读者验证代码、调试与运行结果。

当前版本的 Quantum Composer 中的状态向量视图支持显示的最大量子比特数为 6。

2. 测量概率视图

图 5.10 所示的测量概率视图为量子态及其测量概率的直方图。纵轴为计算基态,横轴表示各计算基态下测量得到的概率。根据窗口的大小,横轴与纵轴会变换位置,概率值等于振幅值的平方。该图表明:结果中仅出现了 $|00\rangle$ 和 $|11\rangle$,各自的概率约为 50%。

对 $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ 重复进行测量,计算基态 $|00\rangle$ 和 $|11\rangle$ 的测量概率分别为

$$\begin{cases} p_{00} = |\langle 00 | \psi \rangle|^2 = |1/\sqrt{2}|^2 = 1/2 = 50\% \\ p_{11} = |\langle 11 | \psi \rangle|^2 = |1/\sqrt{2}|^2 = 1/2 = 50\% \end{cases} \quad (5.2)$$

当前版本的 Quantum Composer 中的测量概率视图支持显示的最大量子比特数为 8。