量子计算
—算法篇

Quantum Computer

网址：www.qubits.top

作者：Calvin Tang

邮箱：179209347@qq.com

介绍

教程简介：

- 面向对象：量子计算初学者
- 依赖课程：线性代数，解析几何，量子力学（非必需）

知乎专栏：
https://www.zhihu.com/column/c_1501138176371011584

Github & Gitee 地址：
https://github.com/mymagicpower/qubits
https://gitee.com/mymagicpower/qubits

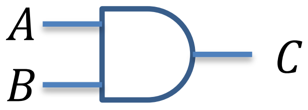
Calvin, QQ: 179209347 Mail: 179209347@qq.com

逻辑门

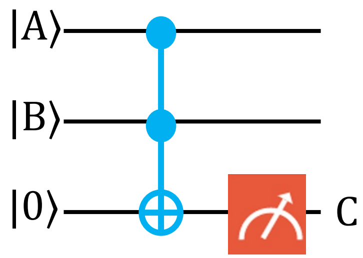
名称	真值表	经典逻辑电路	量子逻辑电路

AND
与门

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1



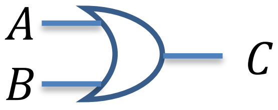
$$C = A \bullet B$$



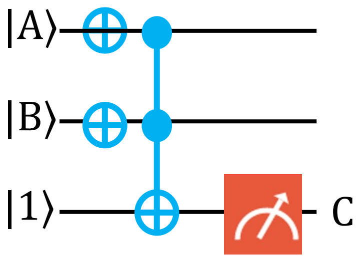
Toffoli 门

OR
或门

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0



$$C = A + B$$



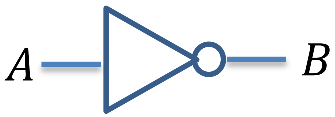
Calvin，QQ: 179209347 Mail: 179209347@qq.com

逻辑门

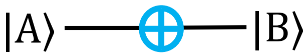
名称 真值表 经典逻辑电路 量子逻辑电路

NOT
非门

A	B
0	1
1	0

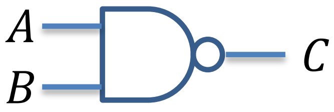


$$B = \bar{A}$$

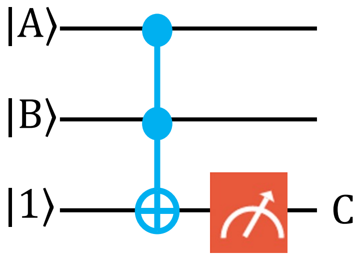


NAND
与非门

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0



$$C = \overline{A \bullet B}$$



Toffoli 门

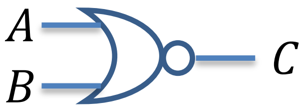
Calvin，QQ: 179209347 Mail: 179209347@qq.com

逻辑门

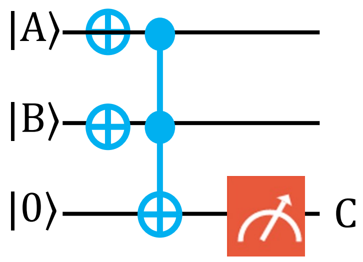
名称	真值表	经典逻辑电路	量子逻辑电路

NOR
或非门

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

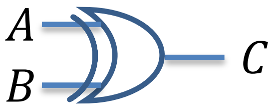


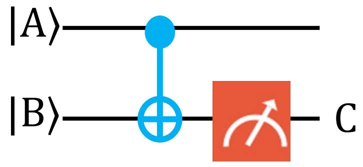
$$C = \overline{A + B}$$



XOR
异或门

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0





CNOT门

$$C = A \oplus B$$

$$= (A + B)\ mod\ 2 = (A + B)\ \%\ 2$$

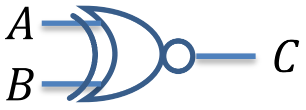
Calvin, QQ: 179209347 Mail: 179209347@qq.com

逻辑门

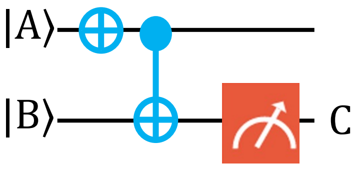
名称 真值表 经典逻辑电路 量子逻辑电路

XNOR
同或门

A	B	C
0	0	1
0	1	0
1	0	0
1	1	1



$$C = \overline{A \oplus B}$$

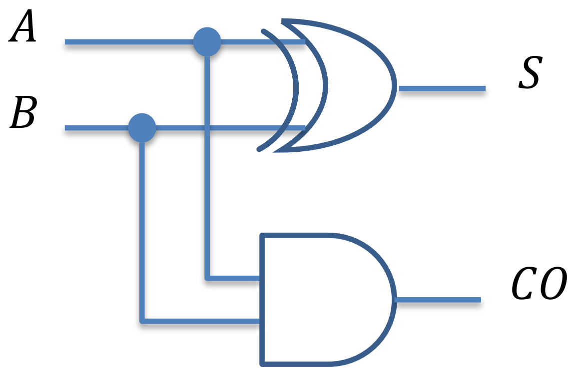


Calvin，QQ: 179209347 Mail: 179209347@qq.com

一位加法器 - 半加器

半加器可以实现两个 1 位的二进制数字相加，并且输出结果和进位。它的真值表根据二进制加法就可以得到。

经典逻辑电路



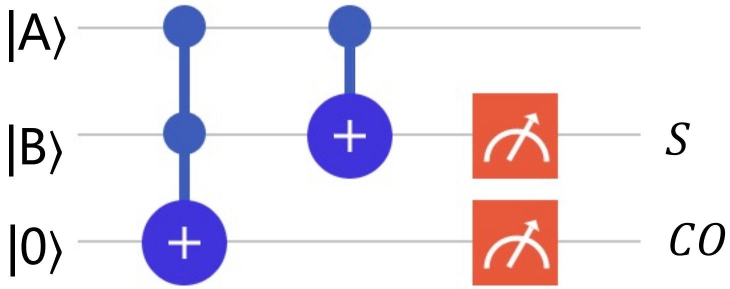
$S = A \oplus B$

$(A + B)\ mod\ 2$

进位

$CO = AB$

量子逻辑电路



半加器真值表

输入		输出	
A	**B**	**CO**	**S**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Calvin, QQ: 179209347 Mail: 179209347@qq.com

一位加法器 - 全加器

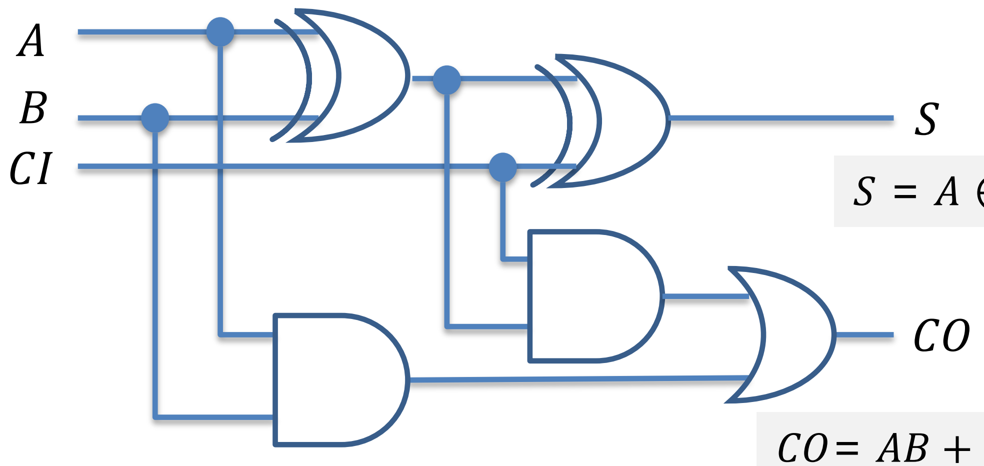
全加器和半加器的区别在于全加器考虑来自低位来的进位数，实质上是一个三个数的加法器。

全加器真值表

输入			输出	
A	B	CI	CO	S
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

一位加法器 - 全加器线路

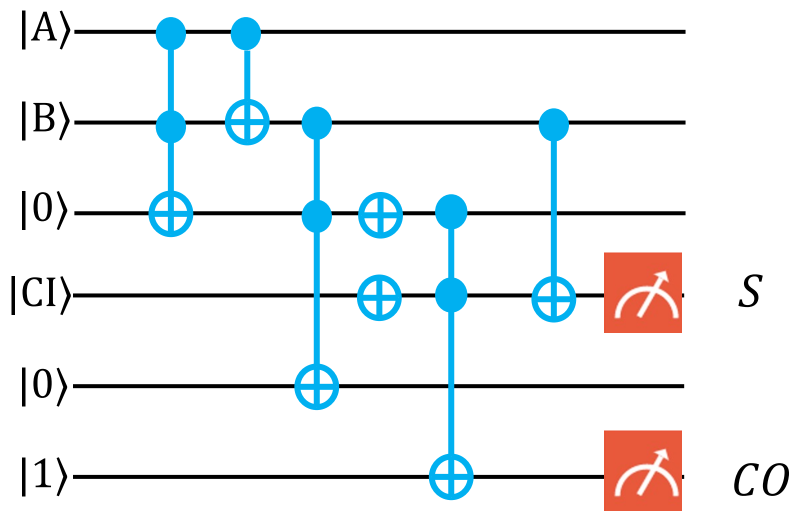
全加器可以由两个半加器和一个或门构成：



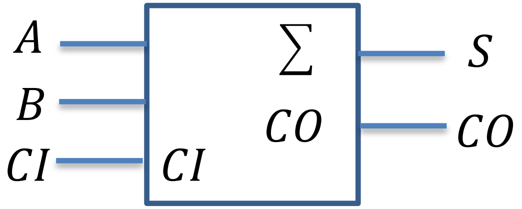
$S = A \oplus B \oplus CI$

$CO = AB + CI(A \oplus B)$

全加器量子线路为：



图形符号

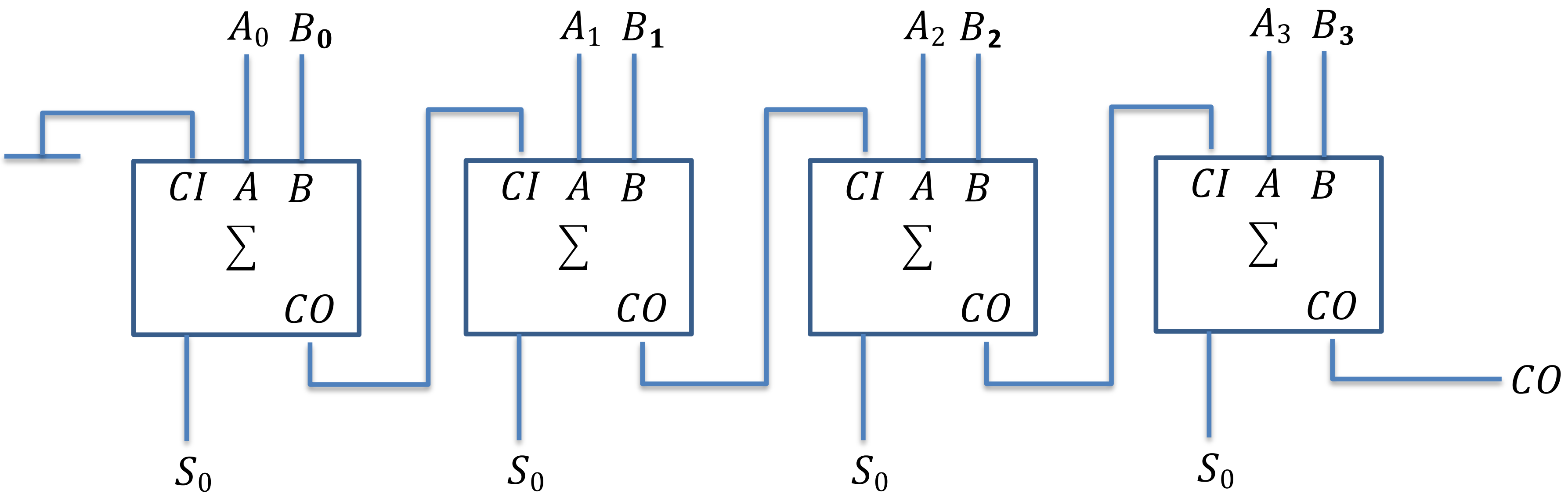


在数字逻辑电路中A、B为 0 或 1，但是在量子电路中A、B 还可以是 0 和 1 的叠加态，得到的结果也可以是叠加态。

Calvin, QQ: 179209347 Mail: 179209347@qq.com

多位加法器

在经典数字电路中，多位加法器是由多个全加器集成而来，如下图：
4 位二进制数相加需要使用 4 个全加器。

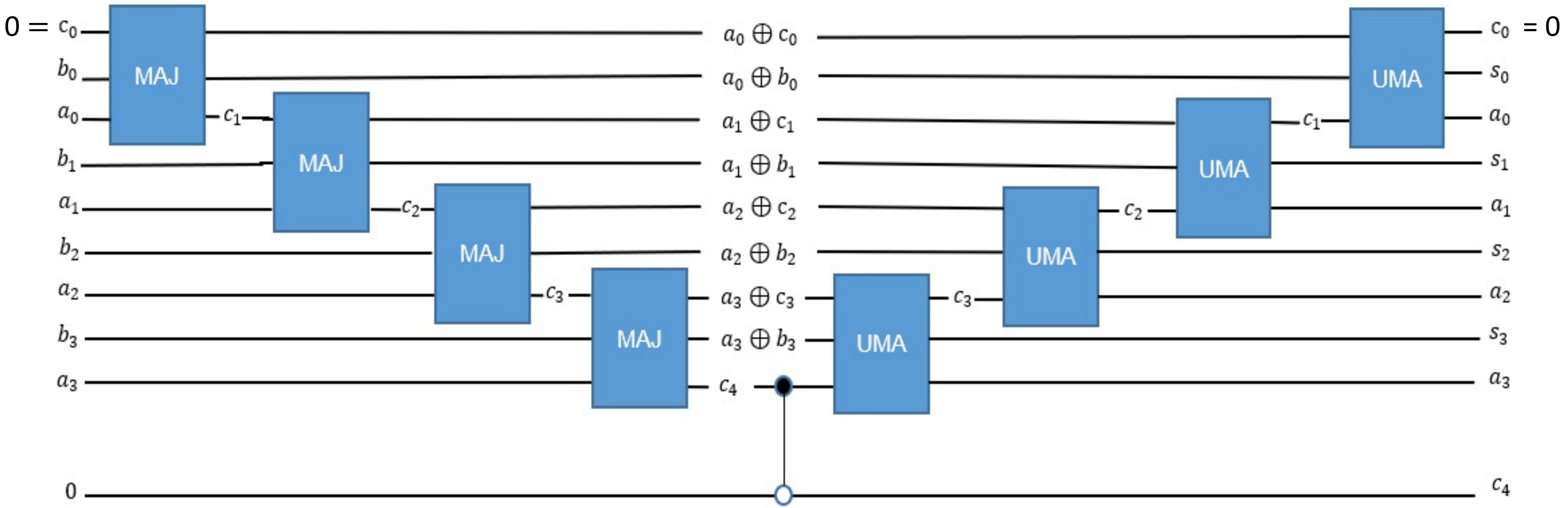


一种更高效的量子电路多位加法器

在某些情况下，量子计算机中需要实现基本的四则运算，而量子加法器是量子四则运算的实现基础。

加法器算法背景

除了测量之外的所有量子门操作都是幺正变换，所以不含测量的量子线路整体是可逆的。
量子加法器的量子线路也应当可逆，因而输入输出是数量相等的量子比特，量子线路图如下所示。

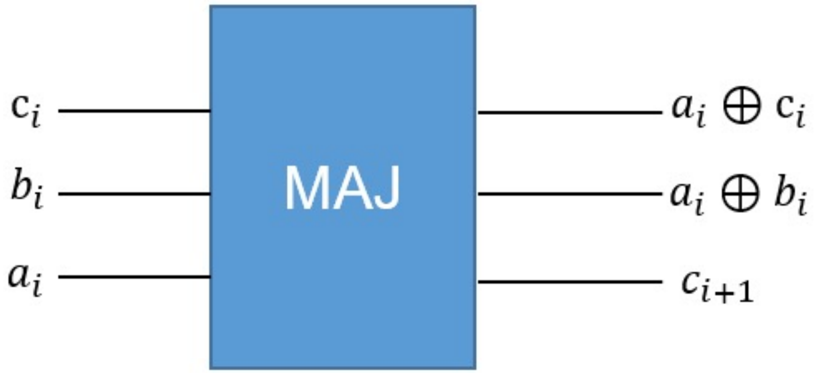


参考来源：https://pyqpanda-toturial.readthedocs.io/zh/latest/QArithmetic.html

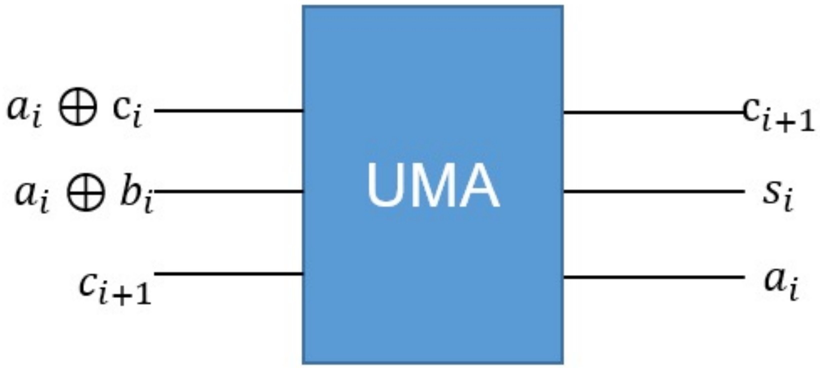
一种更高效的量子电路多位加法器

两个电路用于**求进位**和**求当前位**：in-place majority gate (MAJ) 和UnMajority and Add gate(UMA)。

MAJ的量子线路
MAJ模块是为了获得进位结果



UMA 的量子线路
UMA模块是为了获得当前位结果



输入为：
前一位的进位值 c_i、当前位的两个待加值 a_i, b_i，
输出为：
$a_i + c_i \ mod\ 2$ ， $a_i + b_i \ mod\ 2$ 和当前位进位值 c_{i+1} 。

输入为：
$a_i + c_i \ mod\ 2$ ， $a_i + b_i \ mod\ 2$ 和当前位进位值 $c_i + 1$
输出为：
c_i ， $a_i + b_i + c_i \ mod\ 2 := s_i$ 和 a_i。

参考来源：https://pyqpanda-toturial.readthedocs.io/zh/latest/QArithmetic.html

Calvin，QQ: 179209347 Mail: 179209347@qq.com

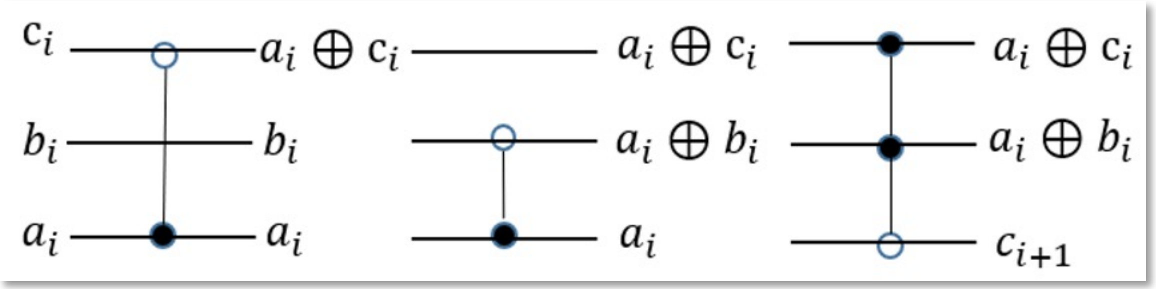
MAJ量子线路组件

MAJ模块是为了实现获得进位：

我们想要得到进位 c_{i+1} ，等价于判断 $(a_i+b_i+c_i)$ / 2 。

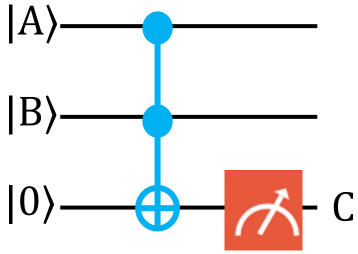
MAJ表达式：

$$|c_i\rangle|b_i\rangle|a_i\rangle \xrightarrow{MAJ} |a_i \oplus c_i\rangle\ |a_i \oplus b_i\rangle\ |c_{i+1}\rangle$$



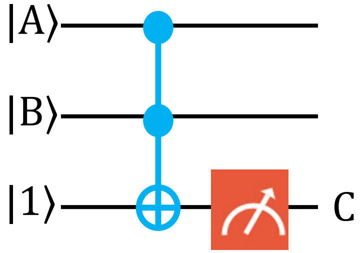
$$A \oplus B = (A + B) \bmod 2 = (A + B)\ \%\ 2$$
$$A \bullet B = A * B$$
$$\bar{A} = (A + 1) \bmod 2 = (A + 1)\ \%\ 2$$

1. 当 a_i=0 时，满足与门：



$$c_{i+1} = A \bullet B = [(a_i+c_i)\ \%\ 2] * [(a_i+b_i)\ \%\ 2]$$

2. 当 a_i=1 时，满足与非门：



$$c_{i+1} = \overline{A \bullet B} = ([(a_i+b_i)\ \%\ 2] * [(a_i+c_i)\ \%\ 2]+1)\ \%\ 2$$

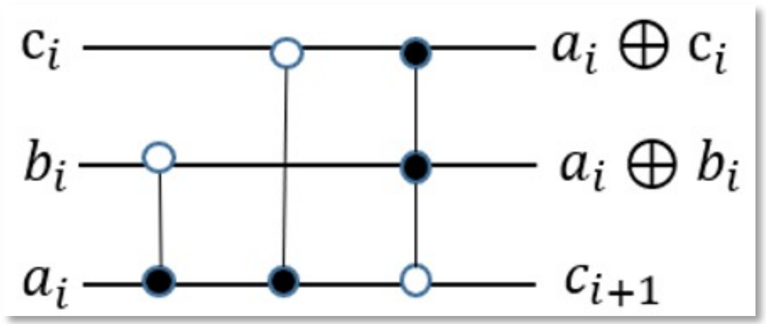
MAJ量子线路组件

通过枚举分析，我们知道只需要考察 a_i，$[(a_i+b_i)\ \%\ 2] * [(a_i+c_i)\ \%\ 2]$ 就可以判断进位情况。

从现有的量子逻辑门出发，制备量子态：

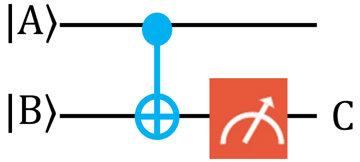
$$a_i，(a_i+b_i)\ \%\ 2，(a_i+c_i)\ \%\ 2$$

即可以准确判断出进位的情况。
此处选取的考察对象并不唯一，其他方案会衍生出相应的量子线路。

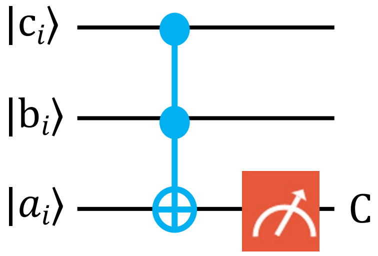


制备三个量子态的方案如图中所示：
使用 CNOT 门来完成模 2 加法得到 $(a_i+b_i)\ \%\ 2$，$(a_i+c_i)\ \%\ 2$
使用 Toffoli 门完成 a 与 $[[(a_i+b_i)\ \%\ 2]*[(a_i+c_i)\ \%\ 2]$ 的运算。

CNOT门



Toffoli 门



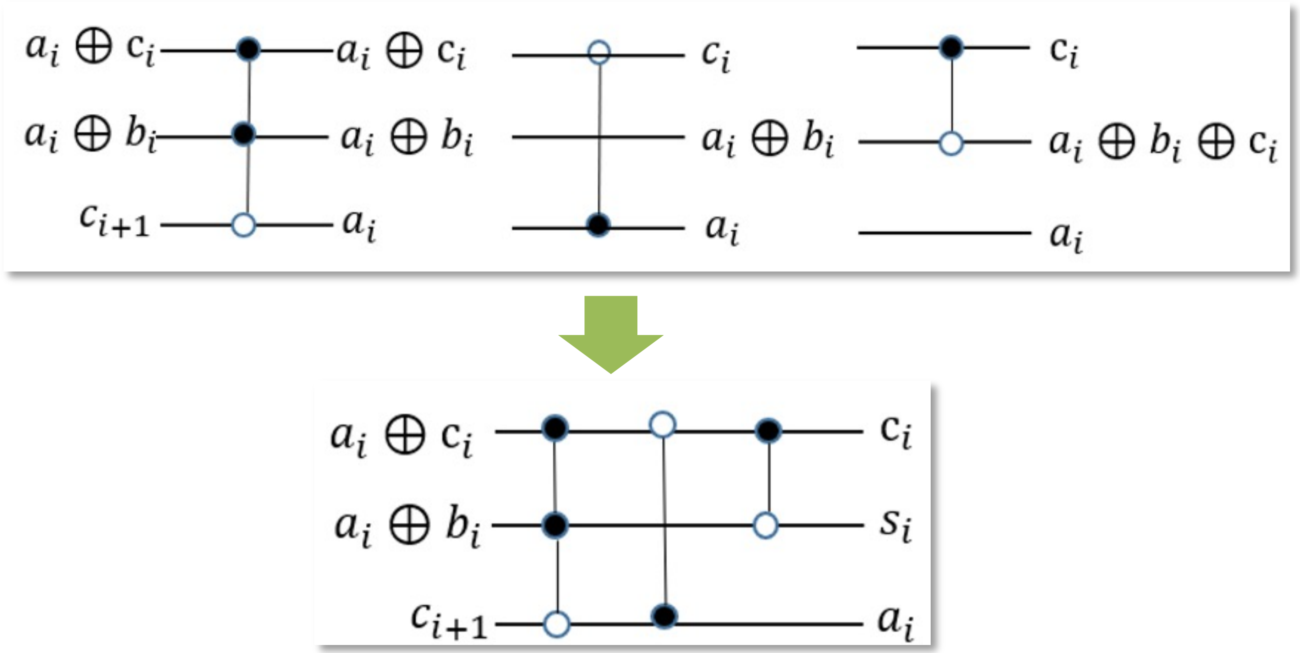
参考来源：https://pyqpanda-toturial.readthedocs.io/zh/latest/QArithmetic.html

Calvin，QQ: 179209347 Mail: 179209347@qq.com

UMA量子线路组件

UMA模块是为了获得当前位：
我们想要得到当前位 s_i ,等价于判断 $(a_i+b_i+c_i)$ % 2。

参考MAJ模块：
- 首先通过与MAJ所用的完全相反的Toffoli门由 c_{i+1} 得到 a_i
- 然后利用与MAJ所用的相反的CNOT变换得到 c_i
- 综合已有的 $a_i+b_i \ mod \ 2$ ，于是可以通过简单的CNOT门得到 $(a_i+b_i+c_i)$ % 2。



从图中的UMA电路中可以看出，开始的前两步Toffoli 和 CNOT 恰好为 MAJ 中的后两个门的逆变换，即互相抵消。所以MAJ和UMA合起来的表达式为：

$$|c_i\rangle|b_i\rangle|a_i\rangle \xrightarrow{MAJ\ \ UMA} |c_i\rangle|s_i\rangle|a_i\rangle$$

求和位： $s_i = a_i \oplus b_i \oplus c_i$

进位位：$c_{i+1} = a_i b_i \oplus b_i c_i \oplus c_i a_i$

参考来源：https://pyqpanda-toturial.readthedocs.io/zh/latest/QArithmetic.html

量子四则运算 - 量子减法器

量子减法器实质上就是量子加法器的带符号版本。
对于带符号变换的量子加法，需要追加辅助比特用于记录符号位。
任给两个目标量子态 A,B，对第二个量子态 B 进行特定的补码操作，然后转换为$A-B=A+(-B)$，此处的 $-B$ 并不以符号位取反的方式实现。

该特定的补码操作为：

- 符号位为正则不变
- 符号位为负需要按位取反后再加1

因此需要一个额外的辅助比特来控制是否进行求补码的操作。

参考来源：https://pyqpanda-toturial.readthedocs.io/zh/latest/QArithmetic.html

Calvin, QQ: 179209347 Mail: 179209347@qq.com

量子四则运算 - 量子乘法器

量子乘法器是基于加法器完成的。
选择乘数 A 作为受控比特，选择乘数 B 以二进制展开逐位作为控制比特，将受控加法器的运算结果累加到辅助比特中。每完成一次 B 控制的受控加法就将乘数 A 左移一位并在末位补零。
于是把通过受控加法输出的数值在辅助比特中累加起来，得到乘法结果。

从竖式中可以看出：

- 乘法相当于做与运算和加法运算。
- 并且 n 位二进制数的乘法最多有 2n 位，其中 c_i 为进位。

			a_2	a_1	a_0
$\times$			b_2	b_1	b_0
	0	0	a_2b_0	a_1b_0	a_0b_0
	0	a_2b_1	a_1b_1	a_0b_1	0
$+$	a_2b_2	a_1b_2	a_0b_2	0	0
c_i	s_4	s_3	s_2	s_1	s_0

Calvin, QQ: 179209347 Mail: 179209347@qq.com

量子四则运算 - 量子除法器

量子除法器是基于量子减法器完成的。
通过执行减法后被除数的符号位是否改变来完成大小比较，并决定除法是否终止。
除数减去被除数时，商结果加 1。每完成一次减法后，重新进行被除数与除数的大小比较，直至除尽或者达到预设精度。因此还需要额外追加一个存储精度参数的辅助比特。

与乘法器类似，除法器也是分为带符号运算和仅限正数两类。

Qubits
qubits.top

Thank You