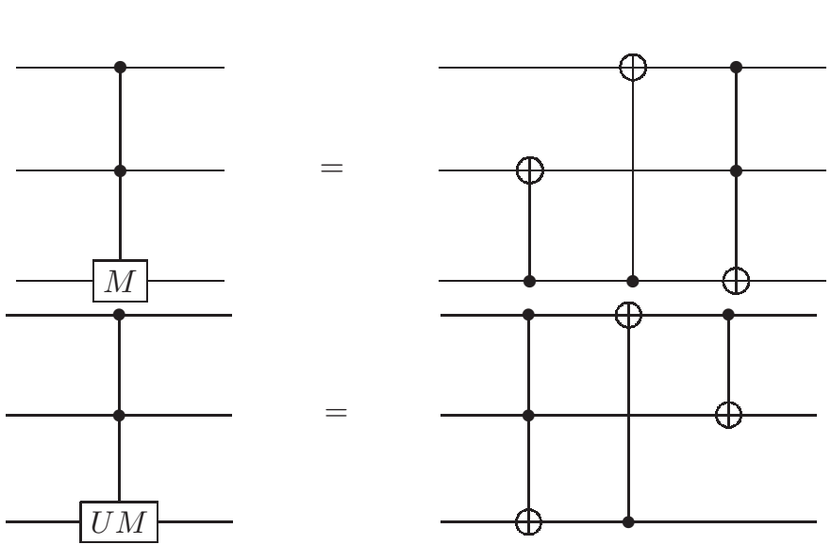
IBMQを用いた
足し算の実装

目的

- CCNOTゲートを用いてIBMQ上で加法を実装すること。
- 参考：https://github.com/QISKit/openqasm/blob/master/examples/generic/adder.qasm

用いたゲート

- CNOT, CCNOT …… 主に他のゲートの定義に用いた。
- M , UM …… 共に３ビットに作用するゲート。CNOT, CCNOTの組み合わせで作られる。



$$
\begin{aligned}
\mathrm{CCN} &= |000\rangle\langle 000| + |001\rangle\langle 001| + |010\rangle\langle 010| + |011\rangle\langle 011| \\
&+ |100\rangle\langle 100| + |101\rangle\langle 101| + |111\rangle\langle 110| + |110\rangle\langle 111| \\
\mathrm{M} &= |000\rangle\langle 000| + |110\rangle\langle 001| + |010\rangle\langle 010| + |101\rangle\langle 011| \\
&+ |100\rangle\langle 100| + |011\rangle\langle 101| + |111\rangle\langle 110| + |001\rangle\langle 111| \\
\mathrm{UM} &= |000\rangle\langle 000| + |111\rangle\langle 001| + |010\rangle\langle 010| + |101\rangle\langle 011| \\
&+ |110\rangle\langle 100| + |001\rangle\langle 101| + |011\rangle\langle 110| + |100\rangle\langle 111|
\end{aligned}
$$

M, UM の性質①

Mの性質：作用する２番目の量子ビットを２・３番目のビットの和の一の位に、3番目のビットを１・２・3番目のビットの和の繰り上がりの有無（有→１、無→０）にする。

　→a+bを計算する際、整数iに対し次々にMをa[i-1],b[i],a[i]（[ ]内は桁の番号）に作用させると、bに繰り上がりを無視した各位の和が蓄えられていき、aの最後の桁a[n]にb[n]での繰り上がりの有無が入る

M, UMの性質②

- ２つのゲートの積UM･Mを作用させると、１・3番目のビットは変化しない

$$\begin{aligned}\mathrm{UM}\cdot\mathrm{M} \quad &= |000\rangle\langle 000| + |011\rangle\langle 001| + |010\rangle\langle 010| + |001\rangle\langle 011| \\ &+ \ |110\rangle\langle 100| + |101\rangle\langle 101| + |100\rangle\langle 110| + |111\rangle\langle 111|\end{aligned}$$

- UMはMのやり残した繰り上がりの処理を完遂する

　→一の位から順にMを作用させていき、その後Mと同じビットに逆の順でUMを作用させて行くと、bにa+bが蓄えられる。

初期値の設定

c_{in}
b_0
a_0
b_1
a_1
b_2
a_2
b_3
a_3
c_{out}

X M UM

この部分でbは正しく繰り上がりの処理をされ、aではMとUMが打ち消しあって次のUMゲートにMで変更される前と同じ状態になる

実装結果

- 15bit+15bitの足し算の実装に成功

- 16bit+16bitで実装したかったがIBM Q QASM Simulatorのqbuitsが32のため断念

- 形式としてはpythonファイルの引数に足し算するものを取るようにし、出力は10進法で返すようにした。

- 30000+40000は

```
python adder.py 30000 40000
```

で実行され、70000と返してくれる。

問題点＆今後の展望

- 実行にめちゃくちゃ時間がかかる
- 30000+30000=60000の計算で441[sec]かかった
- 今回の実装方式はriiple-carry adderと呼ばれるもの(実行時間がかかることが知られている)
- **Carry-lookahead adderでの実装だとどうなる？？**