Factorizing 15 Using Shor's Algorithm and My Investigation of Error

Hao-Chien Wang

Department of Physics, National Taiwan University

June 1, 2020

Outline

Review

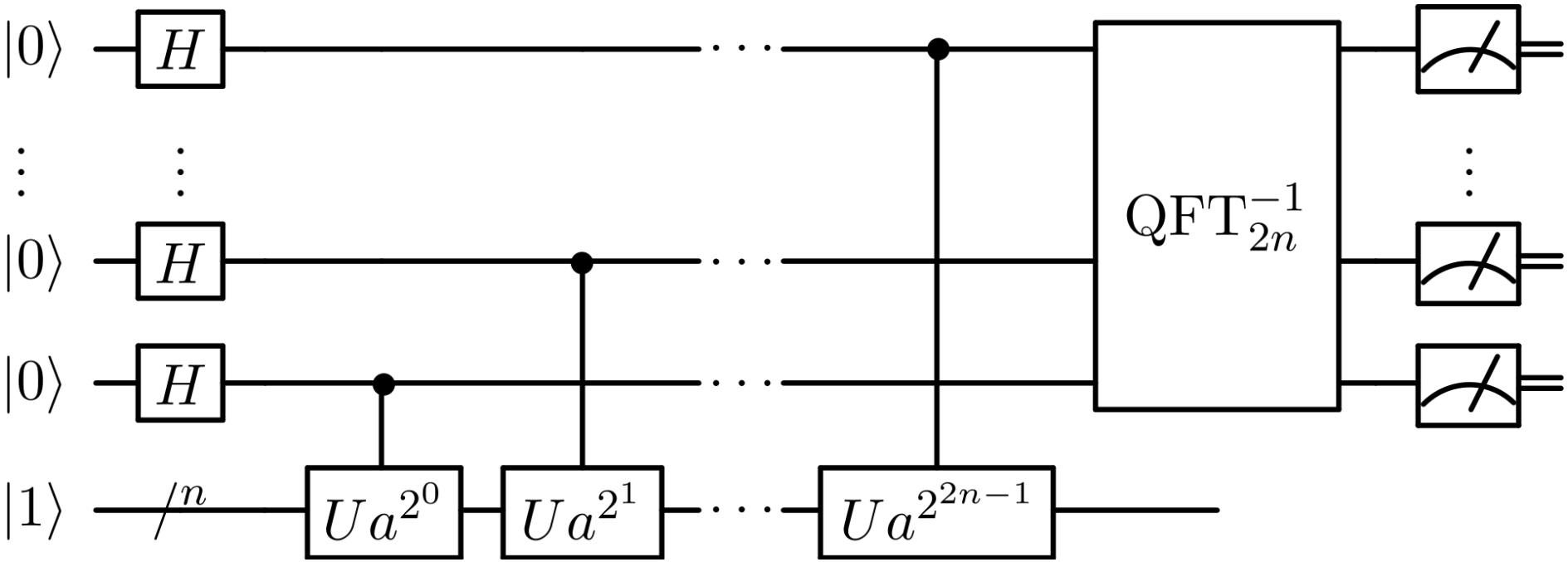


Figure: The circuit of Shor's algorithm.

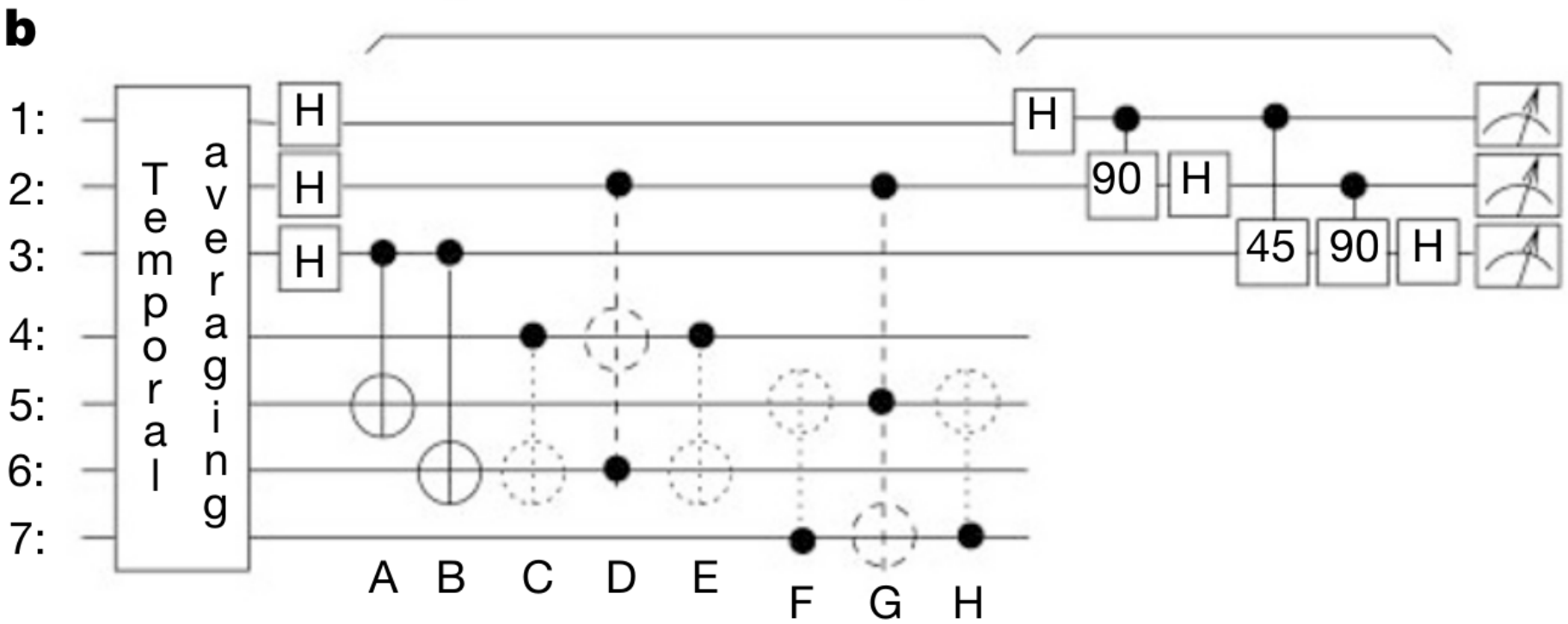
Method

Number of qubits: at least 6.

The options of a for $N = 15$:

- Easy case: 4, 11, 14 ($a^{2^k} \mod N = 1 \; for \; k \geq 1$)
- Difficult case: 2, 7, 8, 13 ($a^{2^k} \mod N = 1 \; for \; k \geq 2$)

Difficult case: Circuit ($a = 7$)

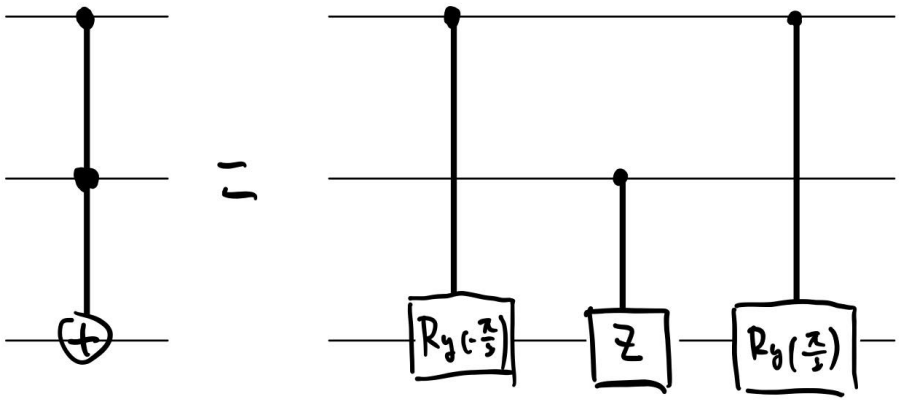


The $4n \mod 15$ gate

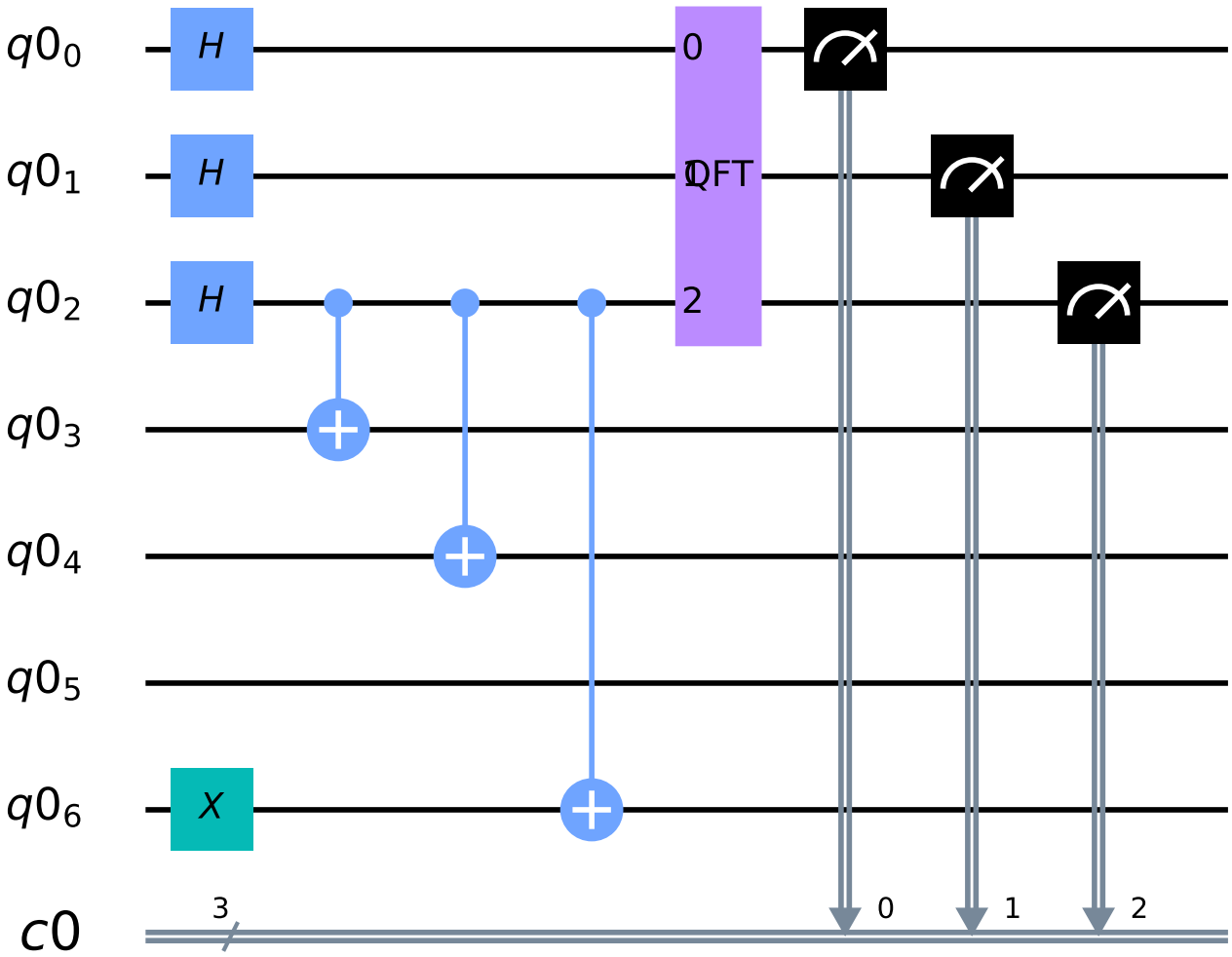
N		4N%15	
0	0000	0	0000
1	0001	4	0100
2	0010	8	1000
3	0011	12	1100
4	0100	1	0001
5	0101	5	0101
6	0110	9	1001
7	0111	13	1101
8	1000	2	0010
9	1001	6	0110
10	1010	10	1010
11	1011	14	1110
12	1100	3	0011
13	1101	7	0111
14	1110	11	1011

Gate simplification

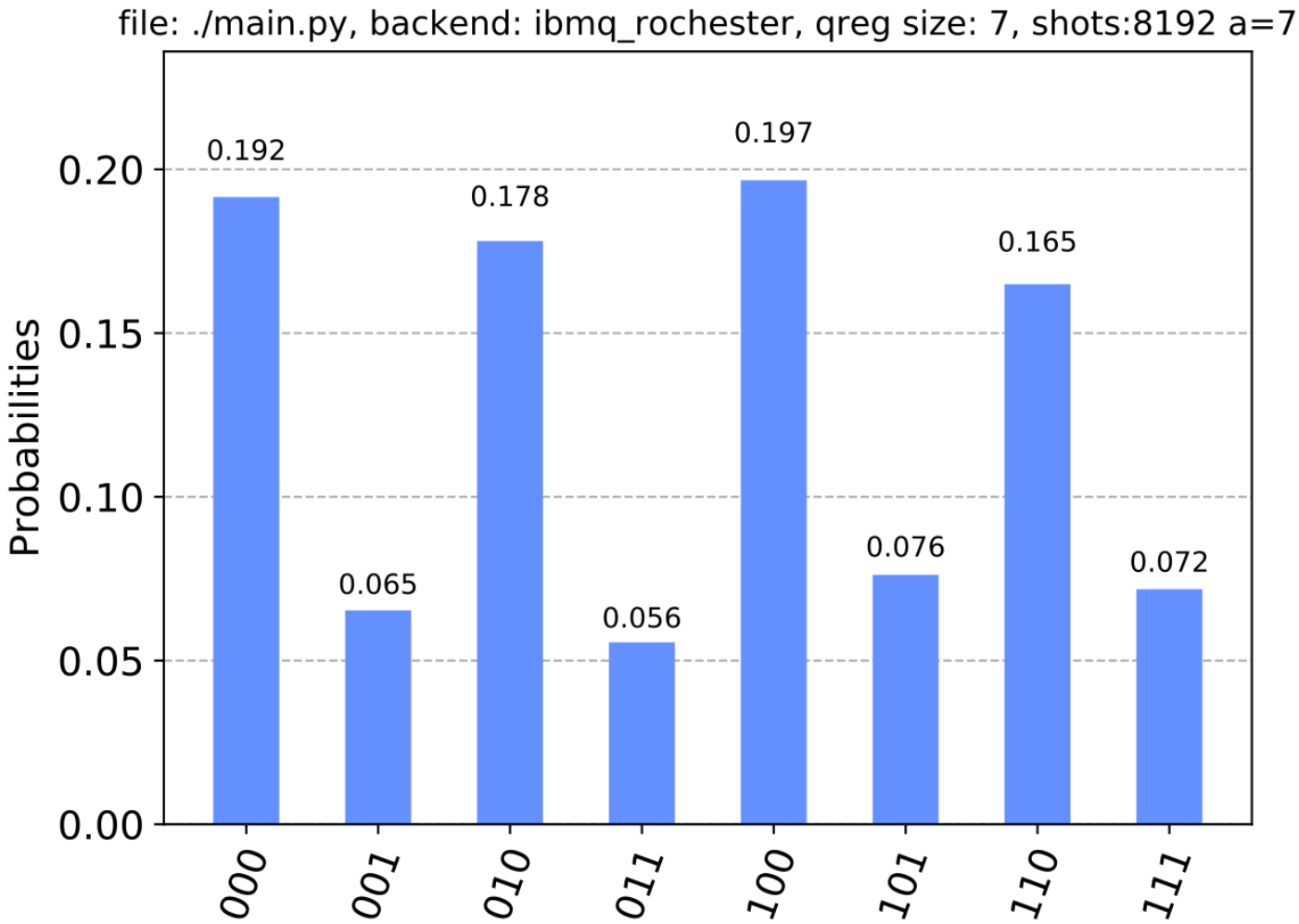
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



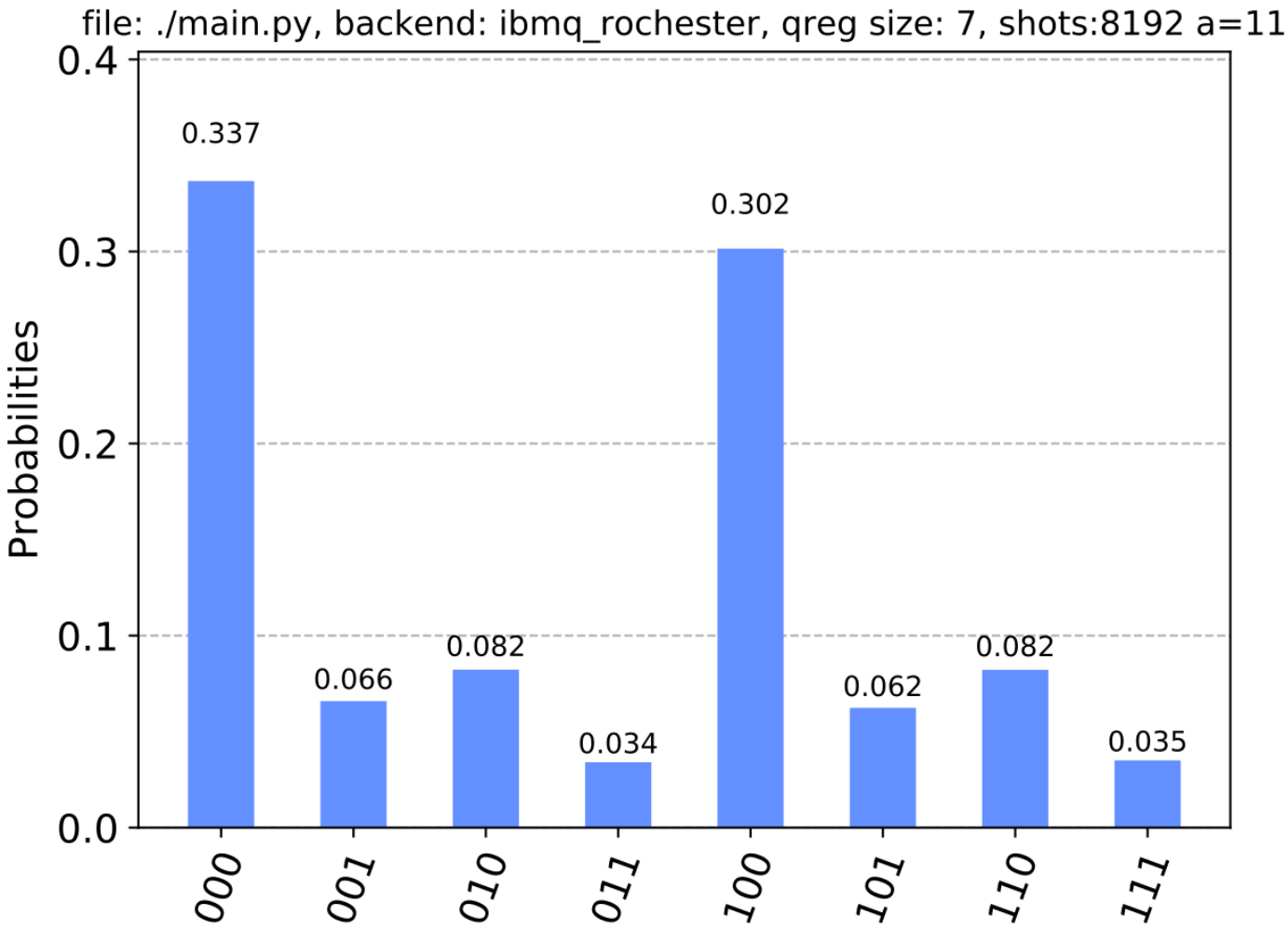
Easy case: Circuit



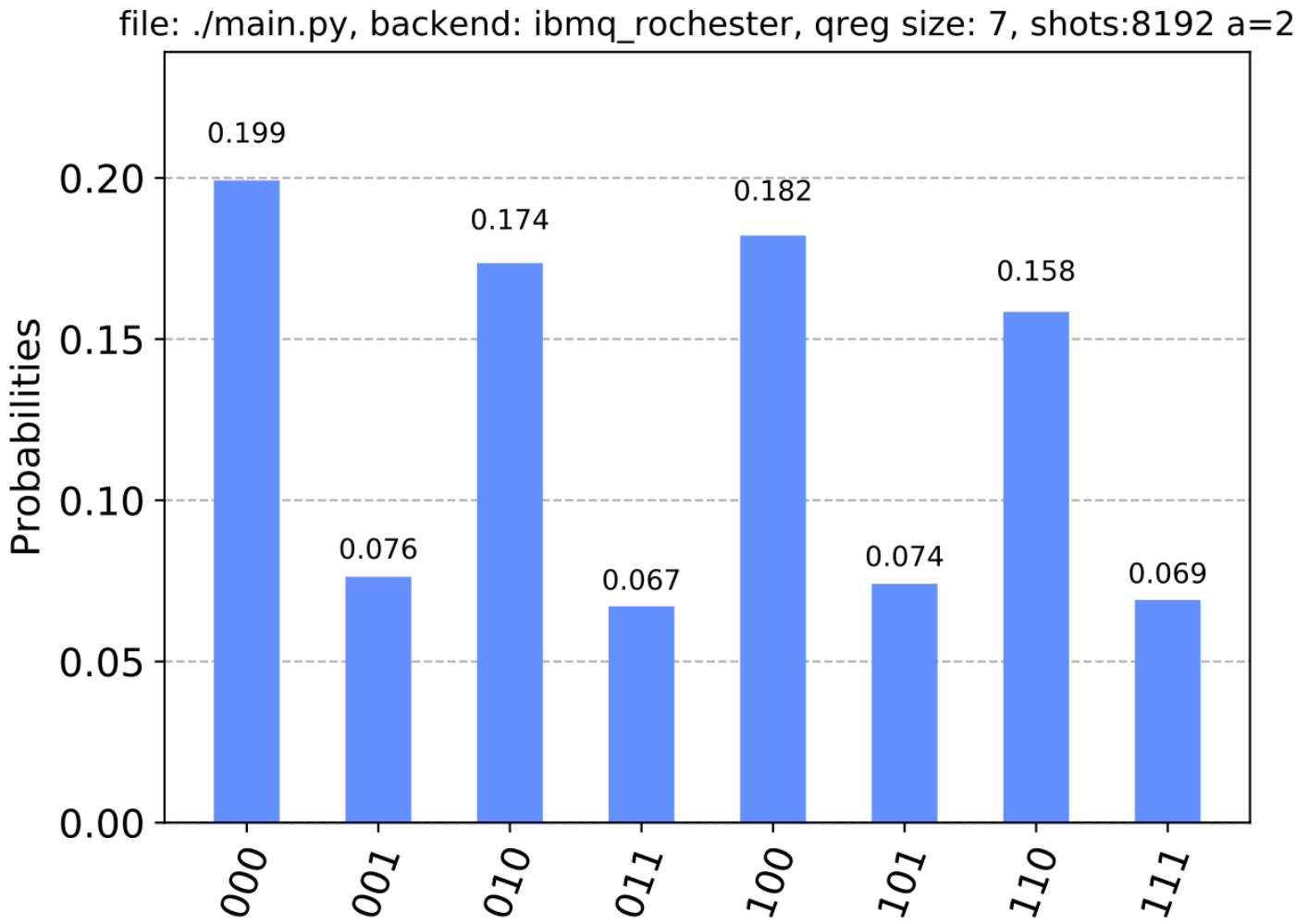
Result: $a = 7$


file: ./main.py, backend: ibmq_rochester, qreg size: 7, shots:8192 a=7
Probabilities
0.00
0.05
0.10
0.15
0.20
0.192
0.065
0.178
0.056
0.197
0.076
0.165
0.072
000
001
010
011
100
101
110
111

Result: $a = 11$


file: ./main.py, backend: ibmq_rochester, qreg size: 7, shots:8192 a=11
Probabilities
0.0
0.1
0.2
0.3
0.4
0.337
0.066
0.082
0.034
0.302
0.062
0.082
0.035
000
001
010
011
100
101
110
111

Result: $a = 2$


file: ./main.py, backend: ibmq_rochester, qreg size: 7, shots:8192 a=2
Probabilities
0.00
0.05
0.10
0.15
0.20
0.199
0.076
0.174
0.067
0.182
0.074
0.158
0.069
000
001
010
011
100
101
110
111

Discussion

- How many qubits do we actually need in the first register?
- Do we need to know the answer in advance? Yes. While deciding the number of qubits of the first register.
- What do we need in order to factorize larger numbers? A powerful compiler is required (peephole compiler)

Can we factorize 21?

2	4	16	4	16	4	16	4	16	4
4	16	4	16	4	16	4	16	4	16
5	4	16	4	16	4	16	4	16	4
8	1	1	1	1	1	1	1	1	1
10	16	4	16	4	16	4	16	4	16
11	16	4	16	4	16	4	16	4	16
13	1	1	1	1	1	1	1	1	1
16	4	16	4	16	4	16	4	16	4
17	16	4	16	4	16	4	16	4	16
19	4	16	4	16	4	16	4	16	4
20	1	1	1	1	1	1	1	1	1

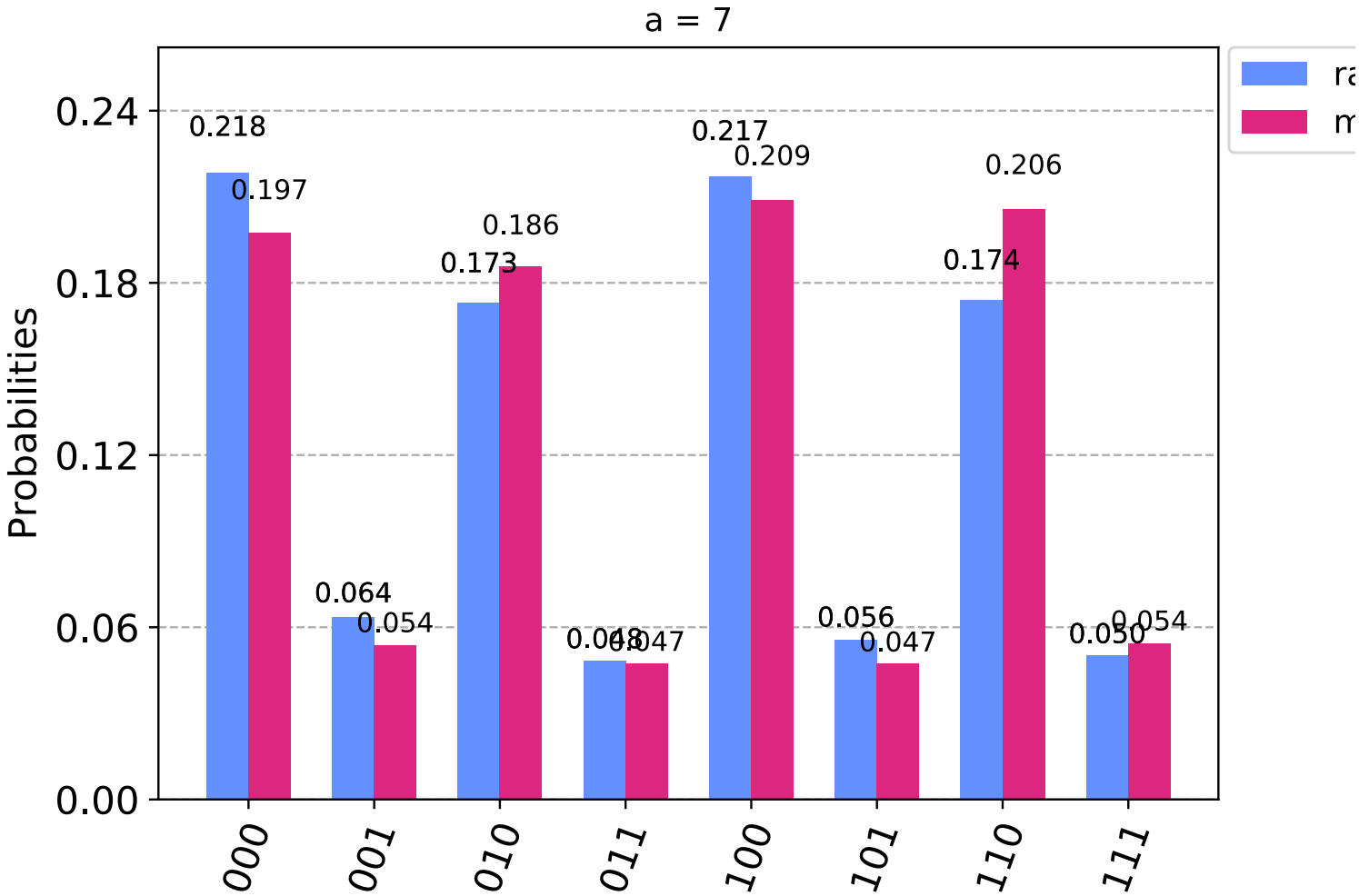
Measurement error mitigation

Assuming that there's some probability for each bitstring to flip to another while measured.

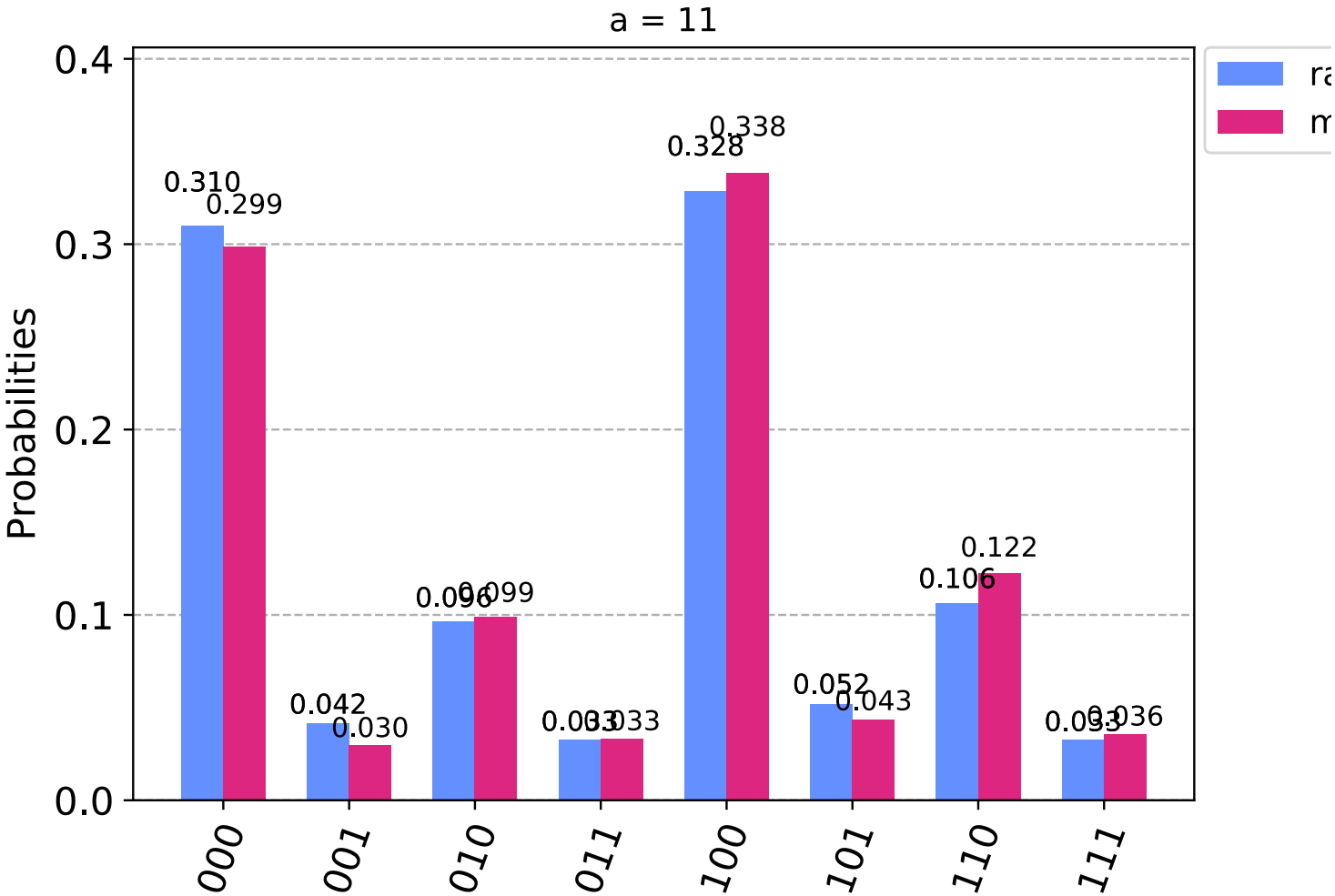
$$C_{noisy} = MC_{ideal}$$

$\Rightarrow$Try each bitstring to construct the matrix!

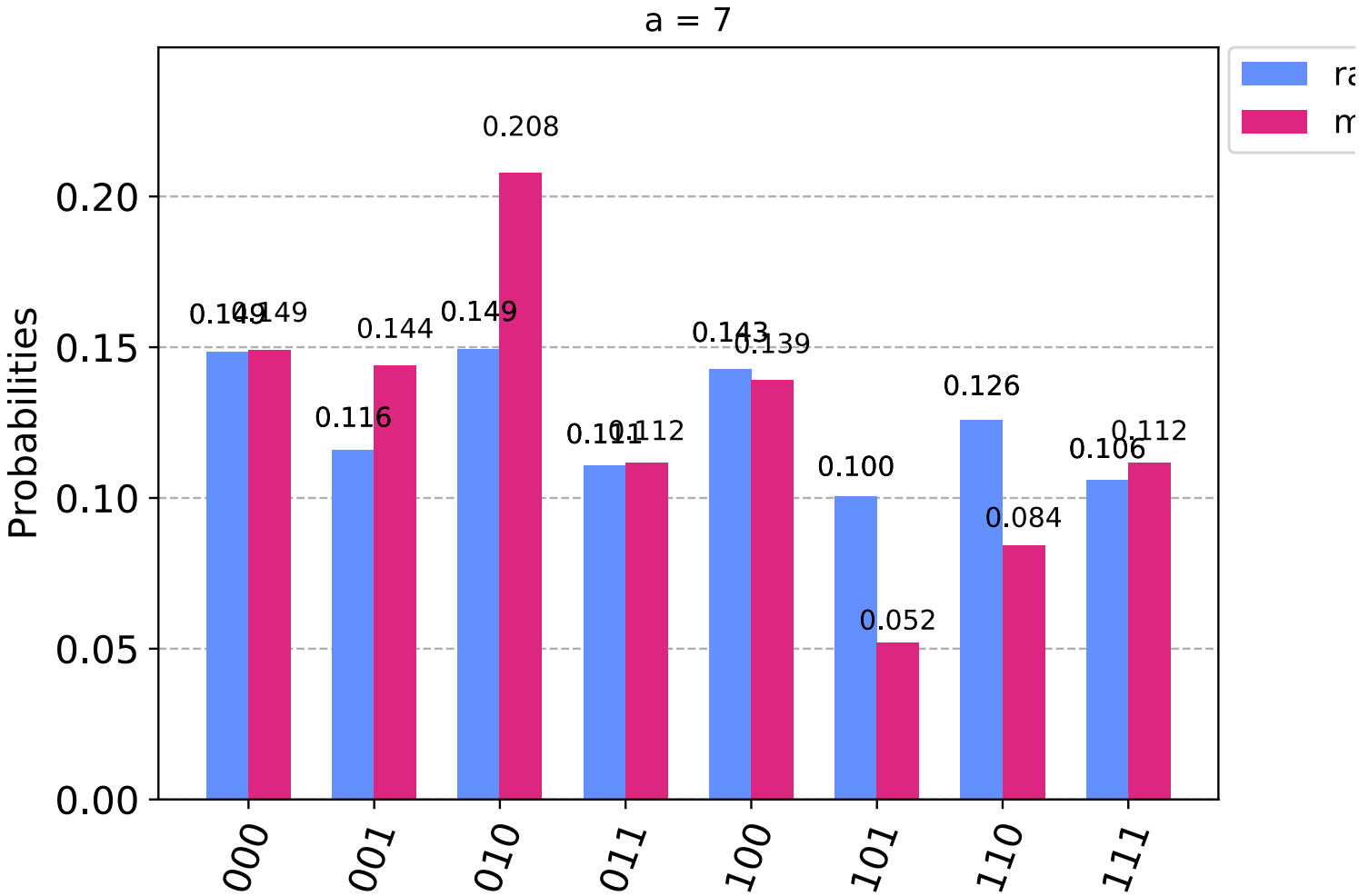
Results


a = 7
Probabilities
0.24
0.18
0.12
0.06
0.00
0.218
0.197
0.064
0.054
0.173
0.186
0.048
0.047
0.217
0.209
0.056
0.047
0.174
0.206
0.050
0.054
000
001
010
011
100
101
110
111

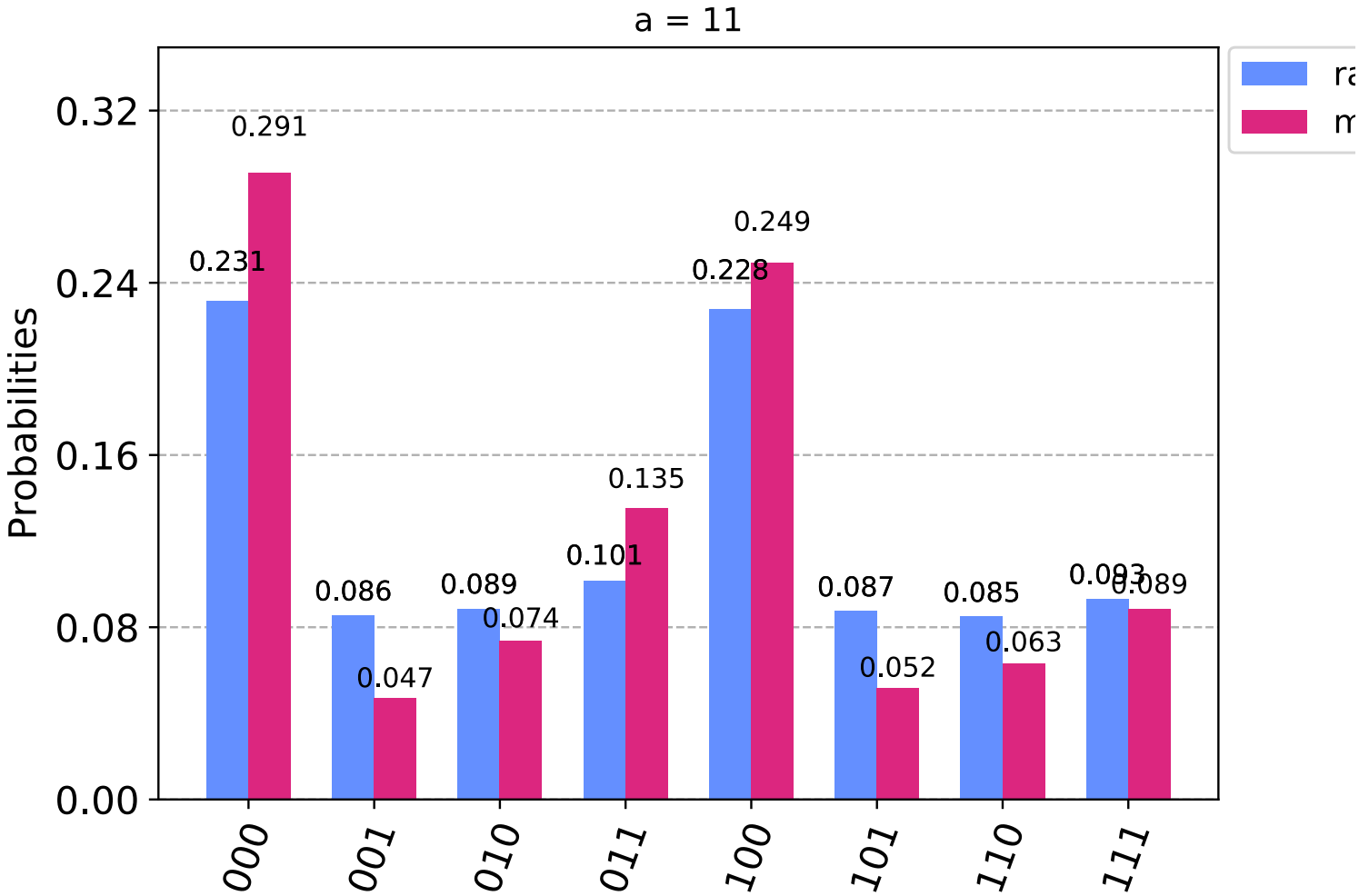
Results


a = 11
Probabilities
0.0
0.1
0.2
0.3
0.4
0.310
0.299
0.042
0.030
0.096
0.099
0.033
0.328
0.338
0.052
0.043
0.106
0.122
000
001
010
011
100
101
110
111

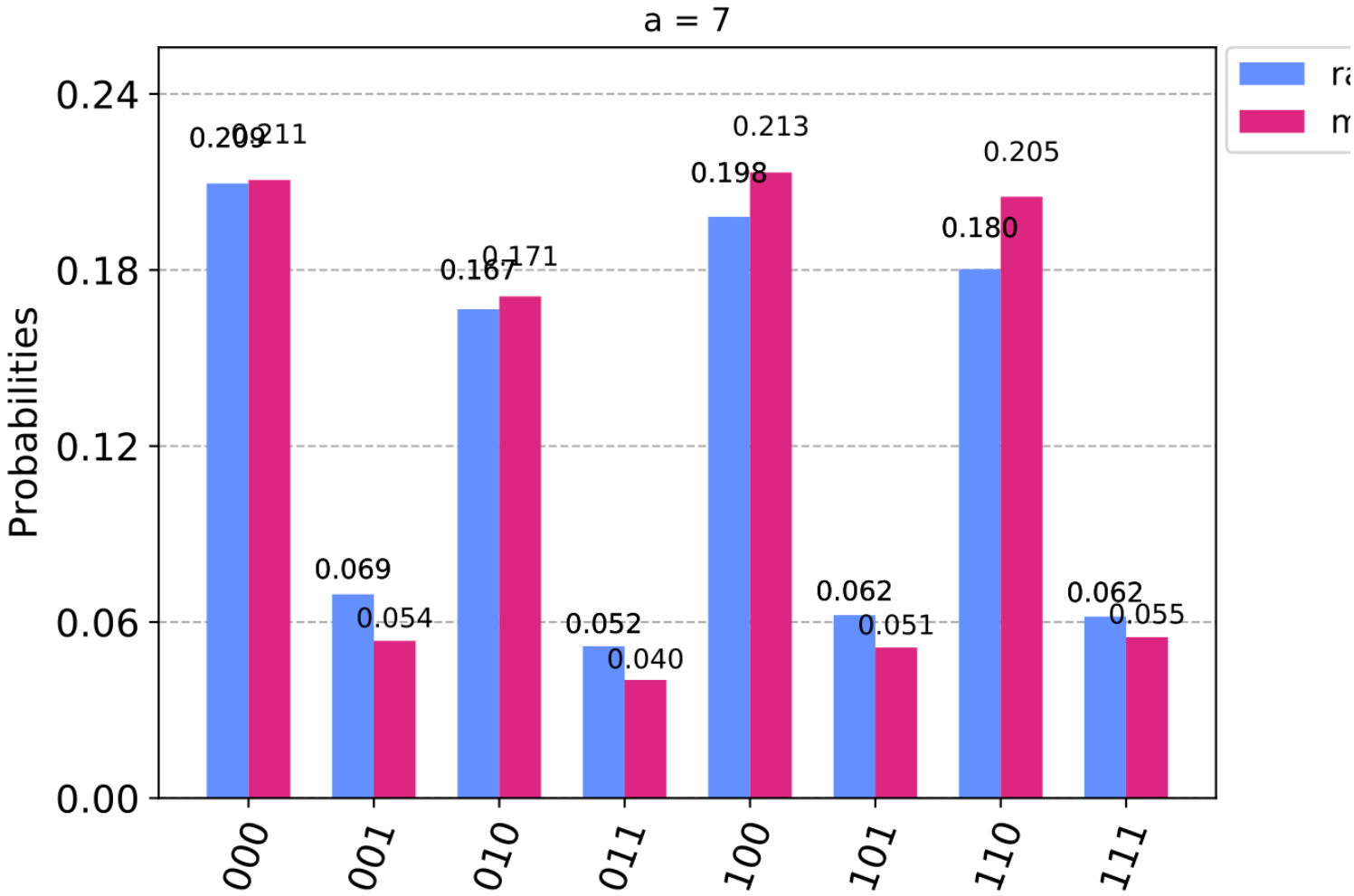
Results


a = 7
Probabilities
0.00
0.05
0.10
0.15
0.20
000
001
010
011
100
101
110
111
0.149
0.144
0.208
0.112
0.139
0.100
0.052
0.126
0.084
0.106
0.112

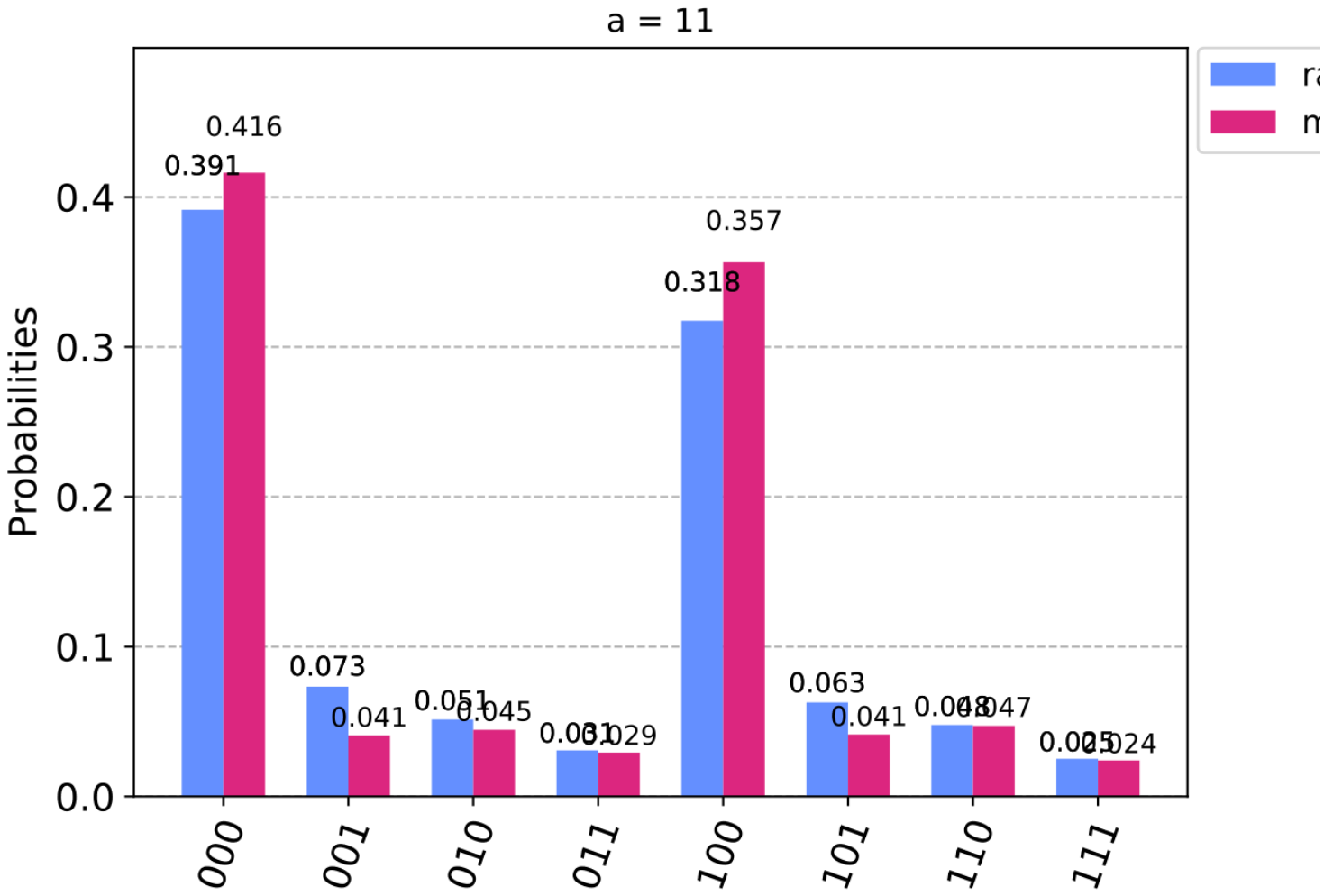
Results


a = 11
Probabilities
0.00
0.08
0.16
0.24
0.32
0.291
0.231
0.086
0.047
0.089
0.074
0.101
0.135
0.249
0.228
0.087
0.052
0.085
0.063
0.093
0.089
000
001
010
011
100
101
110
111

Results


a = 7
Probabilities
0.24
0.18
0.12
0.06
0.00
0.209
0.211
0.069
0.054
0.167
0.171
0.052
0.040
0.198
0.213
0.062
0.051
0.180
0.205
0.062
0.055
000
001
010
011
100
101
110
111

Results


a = 11
Probabilities
0.0
0.1
0.2
0.3
0.4
0.391
0.416
0.073
0.041
0.051
0.045
0.031
0.029
0.318
0.357
0.063
0.041
0.047
0.024
000
001
010
011
100
101
110
111

Discussion

The result was not well. What was causing the error? Depth? What determines the depth?

- Hardware
- Transpiler/compiler

Hardware

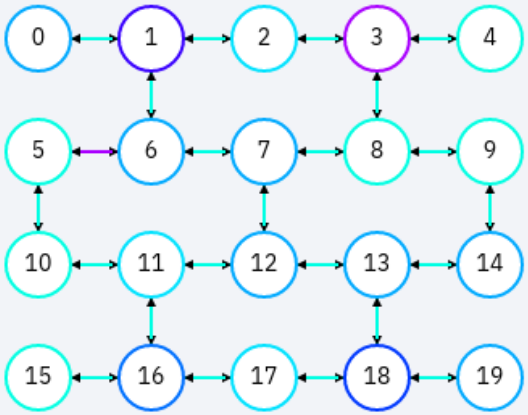
ibmq_almaden v1.4.6

online

Queue: 9 jobs

Providers with access:

ibm-q-hub-ntu/ntu-internal/default



Single-qubit U2 error rate

3.361e-4 1.893e-3

CNOT error rate

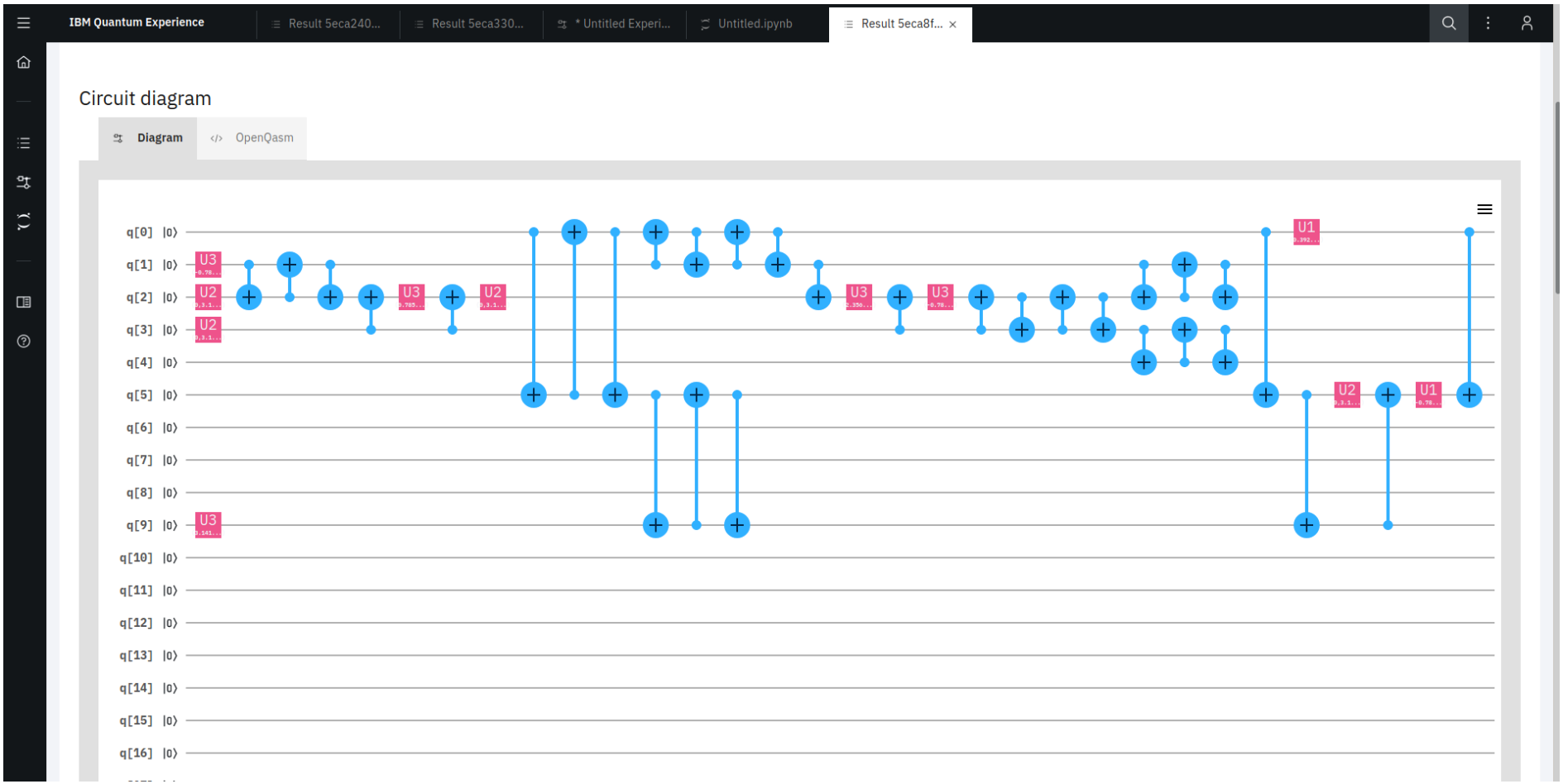
7.887e-3 1.000e+0

Download Calibrations

Qubits	Online since
20	Sep 13, 2019
Basis gates	Last calibration update
u1, u2, u3, cx, id	May 23, 2020 2:42 PM
Maximum shots	Maximum experiments
8192	900

Transpiling/compiling

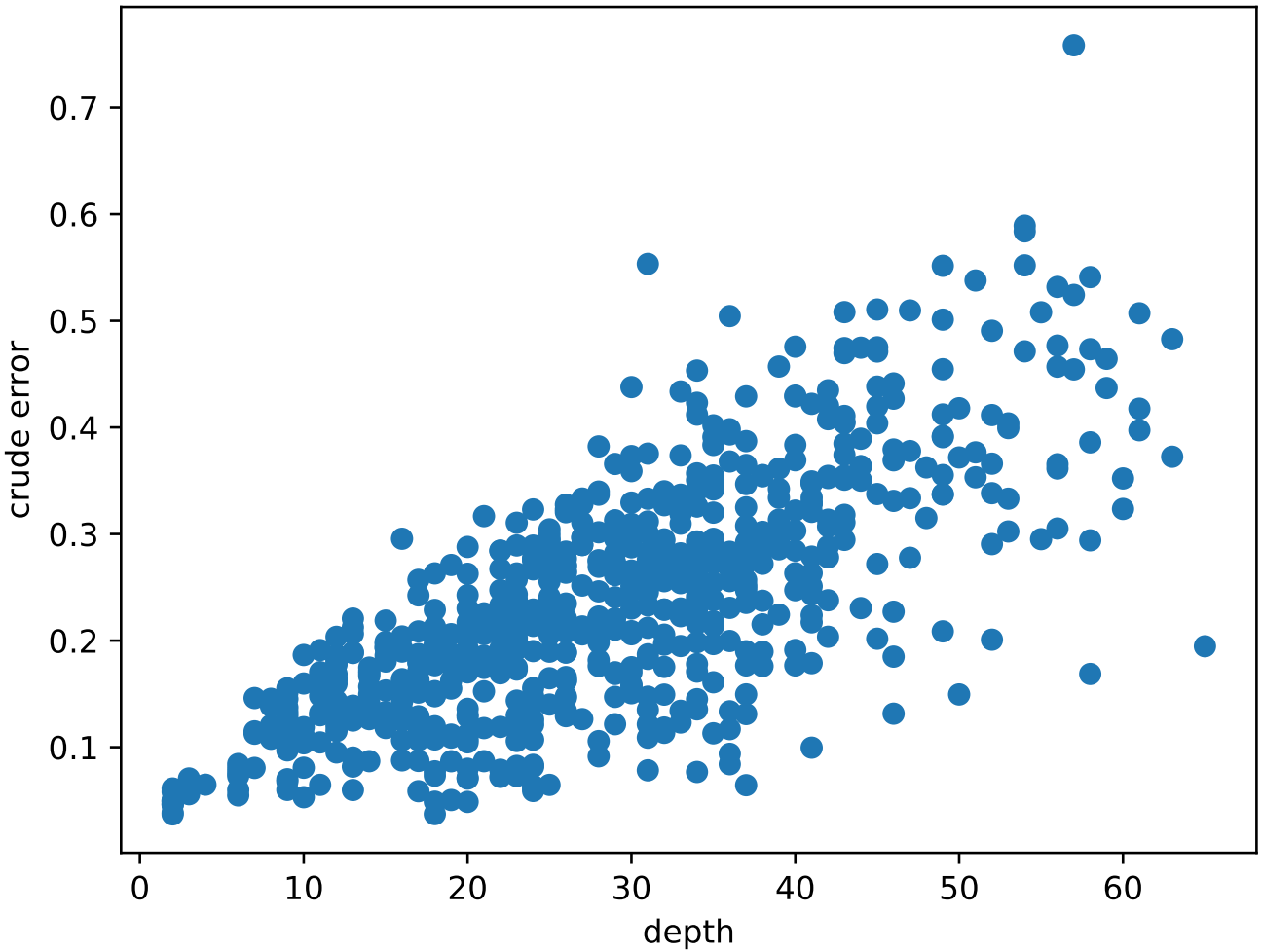
Use `qk.compiler.transpile(qc, backend=be, initial_layout=layout, optimization_level=1)` to assign manually.



Depth vs. error

- Obviously depth $\uparrow \Rightarrow$ error $\uparrow$
  - But quantitatively?
- Create random circuits to test errors!.
- What to expect? Exponential decay of accuracy.

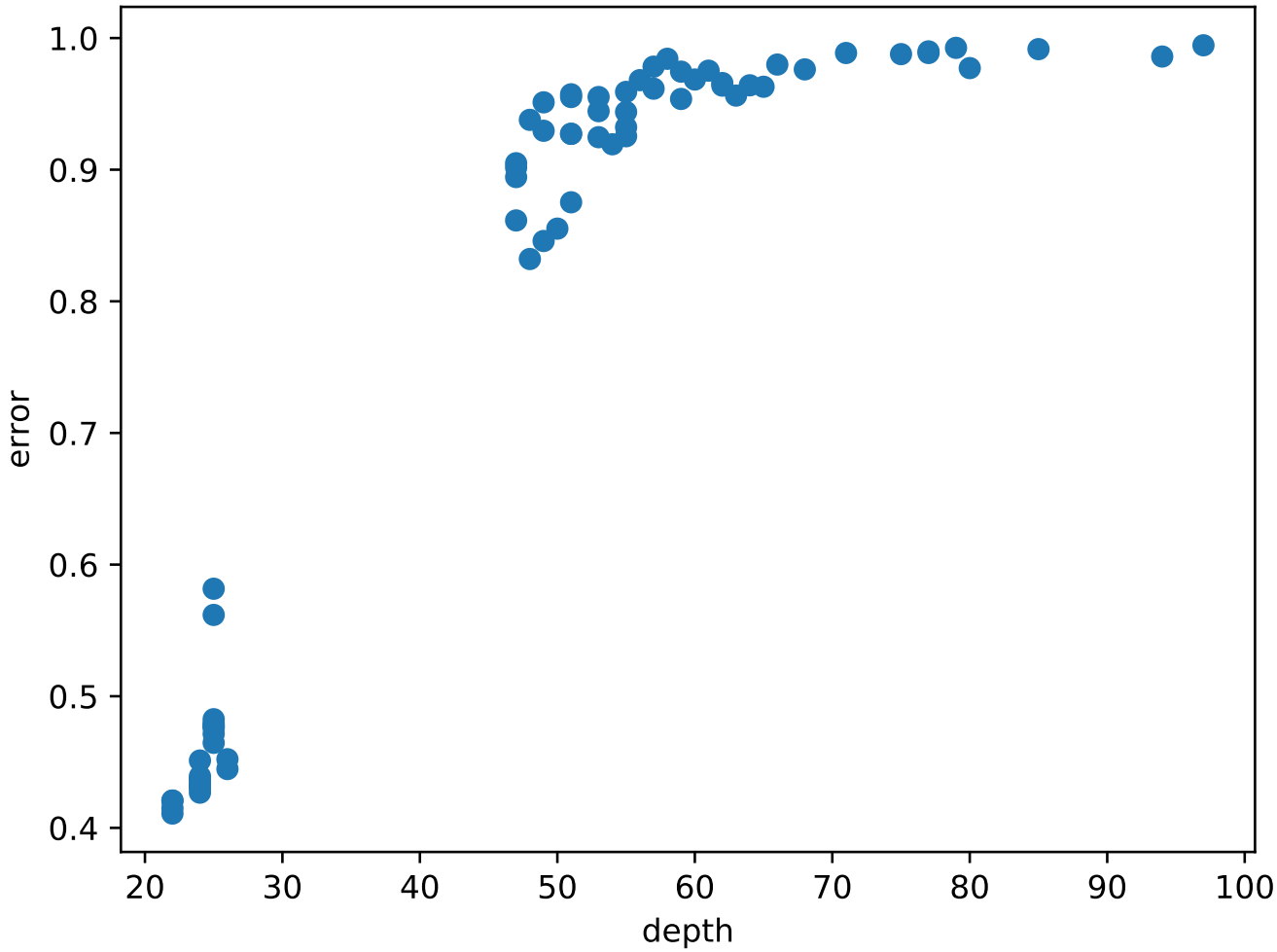
Depth vs. error: random circuits



Discussion

- Bad definition of error!
- Use a circuit with definite results.

Depth vs. error: adder



Depth vs. error: adder