Shor's Algorithm

Boris Varbanov Santiago Sager La Ganga

Delft University of Technology, The Netherlands

February 11, 2018

Shor's Algorithm

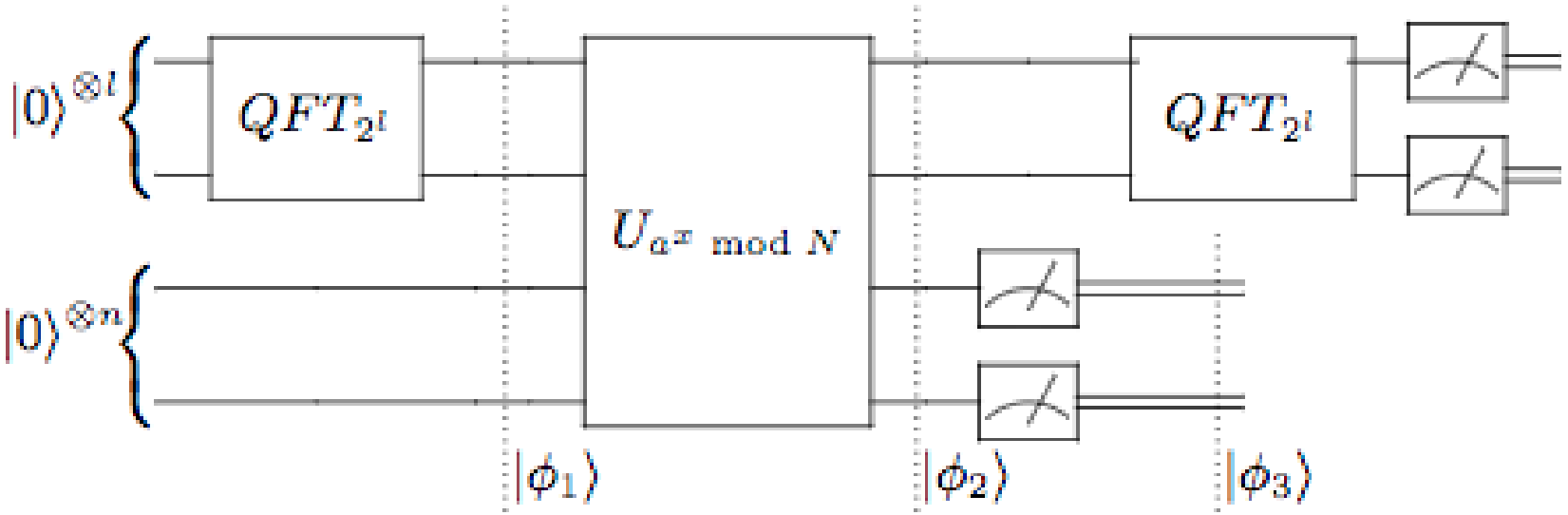


FIG. 2. Shor's quantum circuit for period finding

Binary exponentiation

$$a^x = a^{2^n x_n}\ a^{2^{n-1} x_{n-1}} \ldots a^{2x_1}\ a^{x_0}$$

$$a^{2x_1} a^{x_0} = a^2 a = (a^2 2^n) a_n + \cdots + (a^2 2^1)\ a_1 + (a^2) a_0 + 0$$

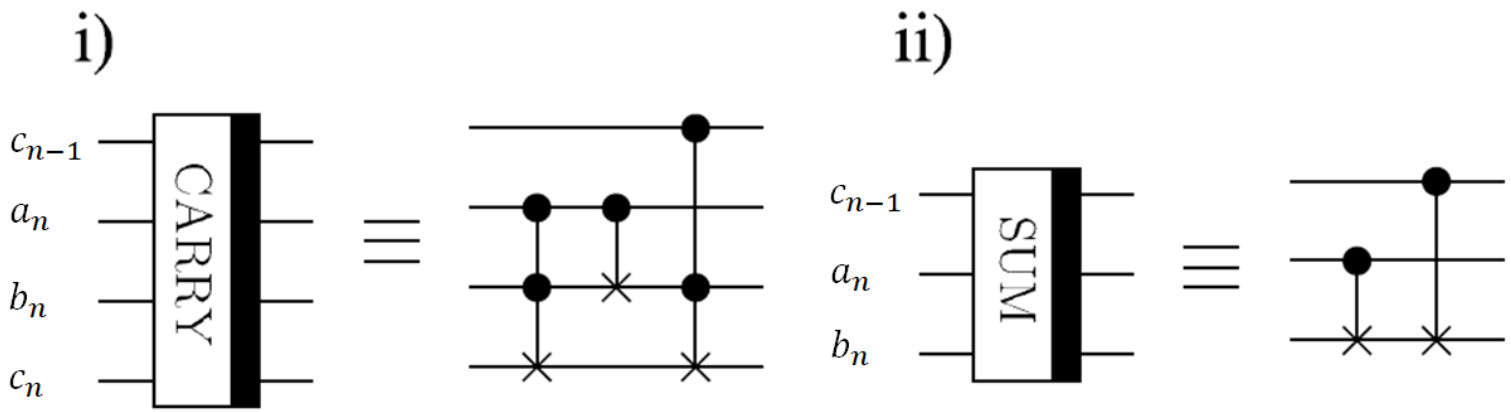
Set

Add

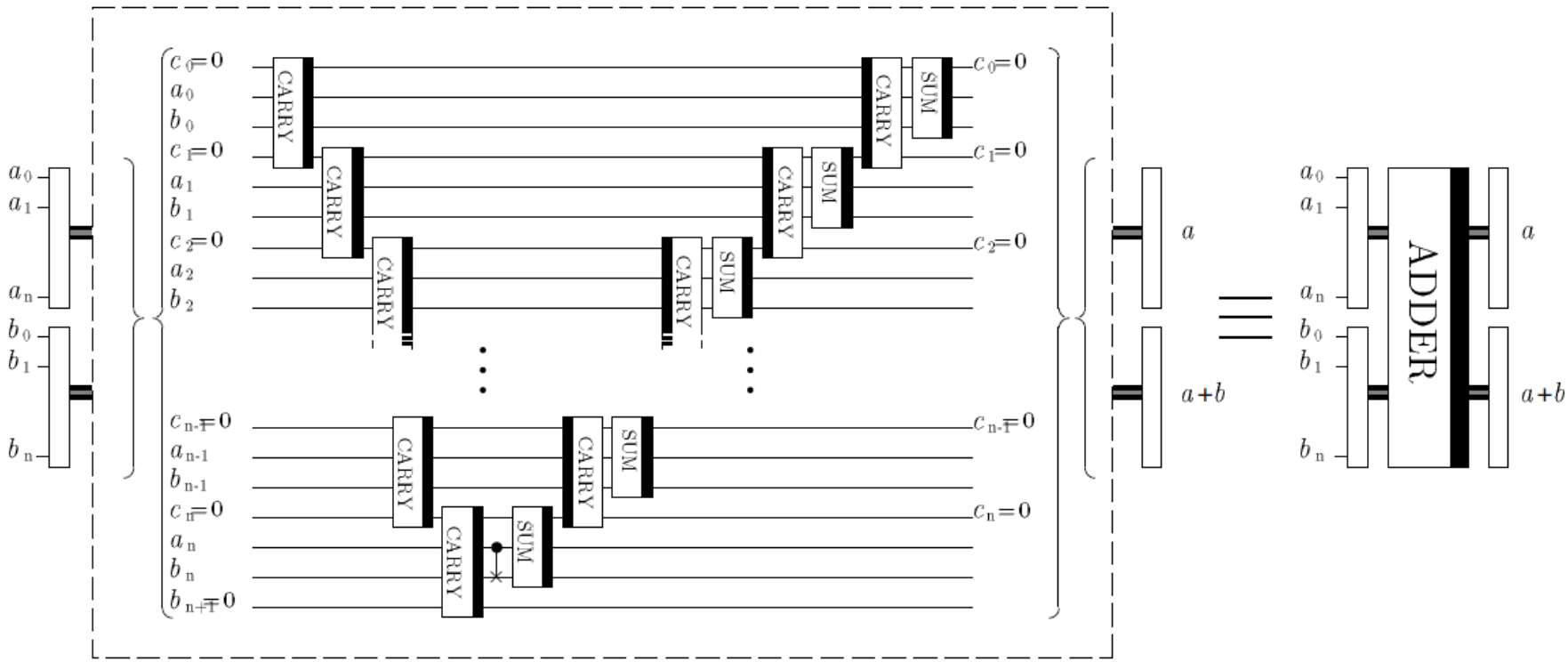
Reset Store

Carry and sum

$$
\begin{array}{rcccc}
 & {\scriptstyle c_3} & {\scriptstyle c_2} & {\scriptstyle c_1} & {\scriptstyle 0} \\
 & 0 & a_2 & a_1 & a_0 \\
+ & 0 & b_2 & b_1 & b_0 \\
\hline
 & c_3 & a_2 + b_2 + c_2 & a_1 + b_1 + c_1 & a_0 + b_0
\end{array}
$$



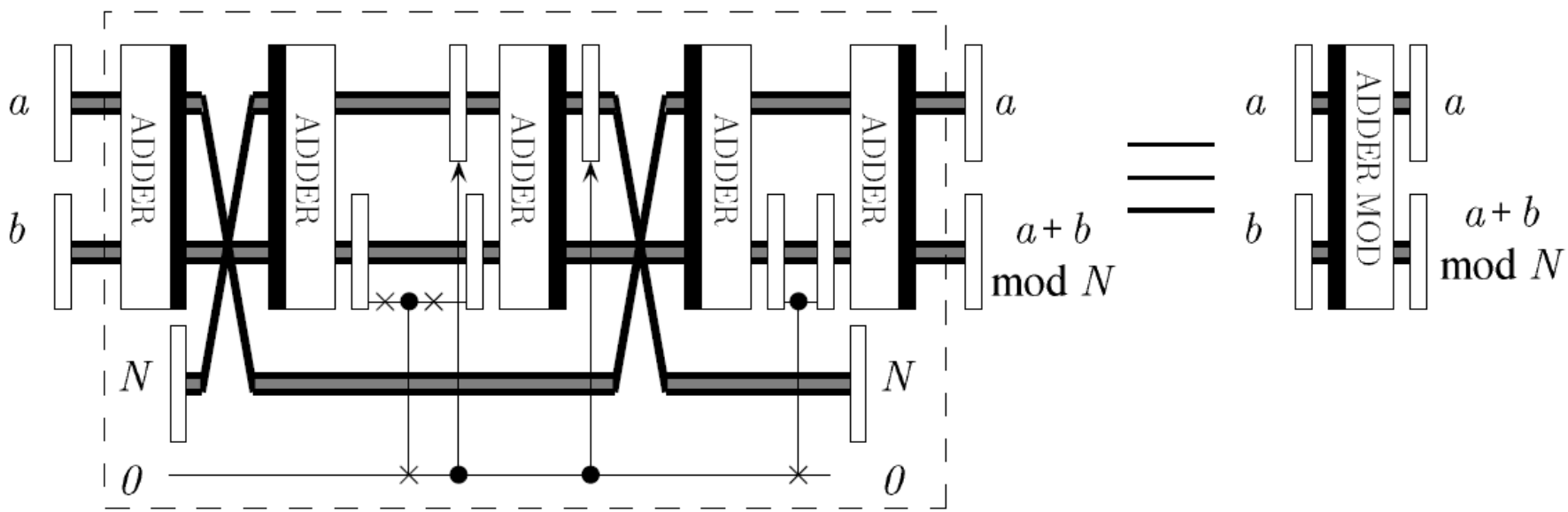
Addition


a_0
a_1
a_n
b_0
b_1
b_n
$c_0=0$
a_0
b_0
$c_1=0$
a_1
b_1
$c_2=0$
a_2
b_2
$c_{n-1}=0$
a_{n-1}
b_{n-1}
$c_n=0$
a_n
b_n
$b_{n+1}=0$
CARRY
SUM
$c_0=0$
$c_1=0$
$c_2=0$
$c_{n-1}=0$
$c_n=0$
a
$a+b$
ADDER

Modular addition

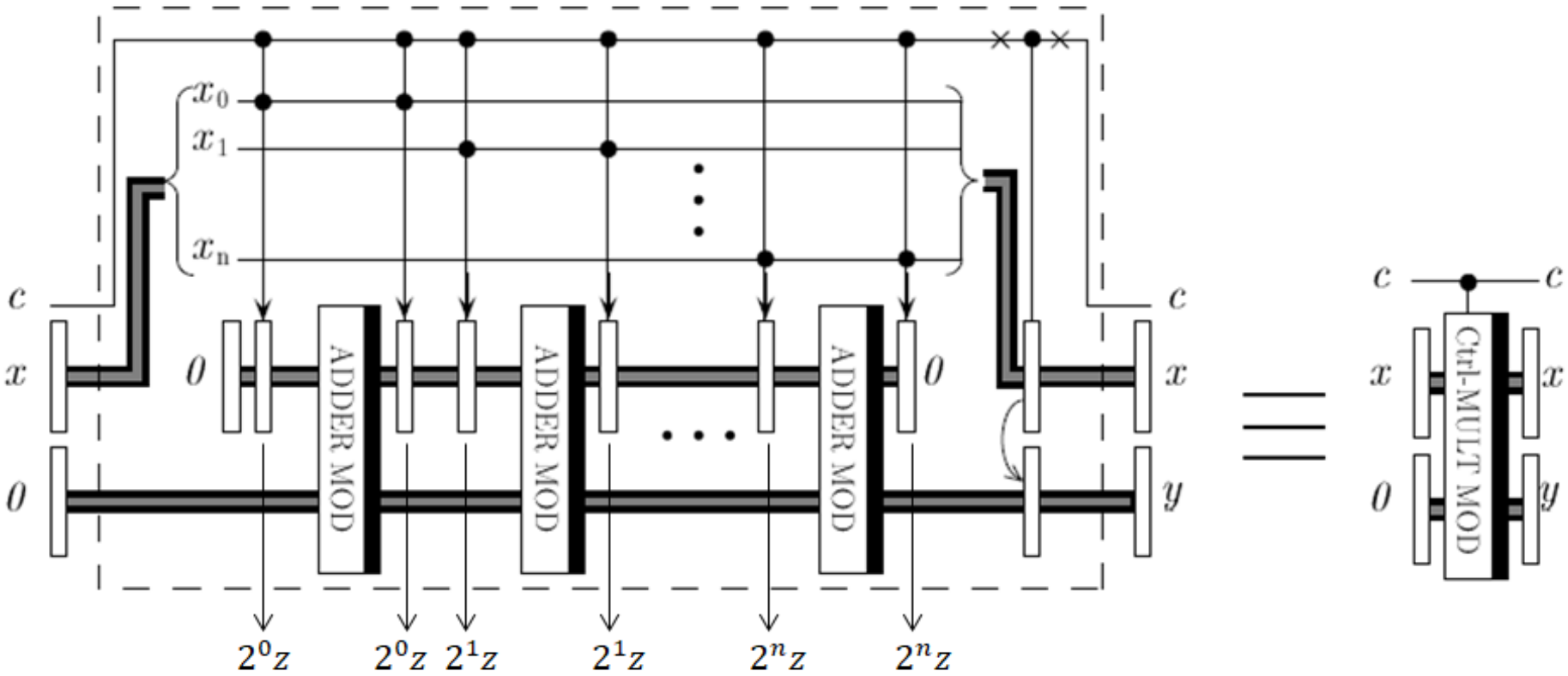
Modular arithmetics

- $ab \bmod N = (a \bmod N \quad b \bmod N) \bmod N$
- $a + b \bmod N = (a \bmod N + b \bmod N) \bmod N$



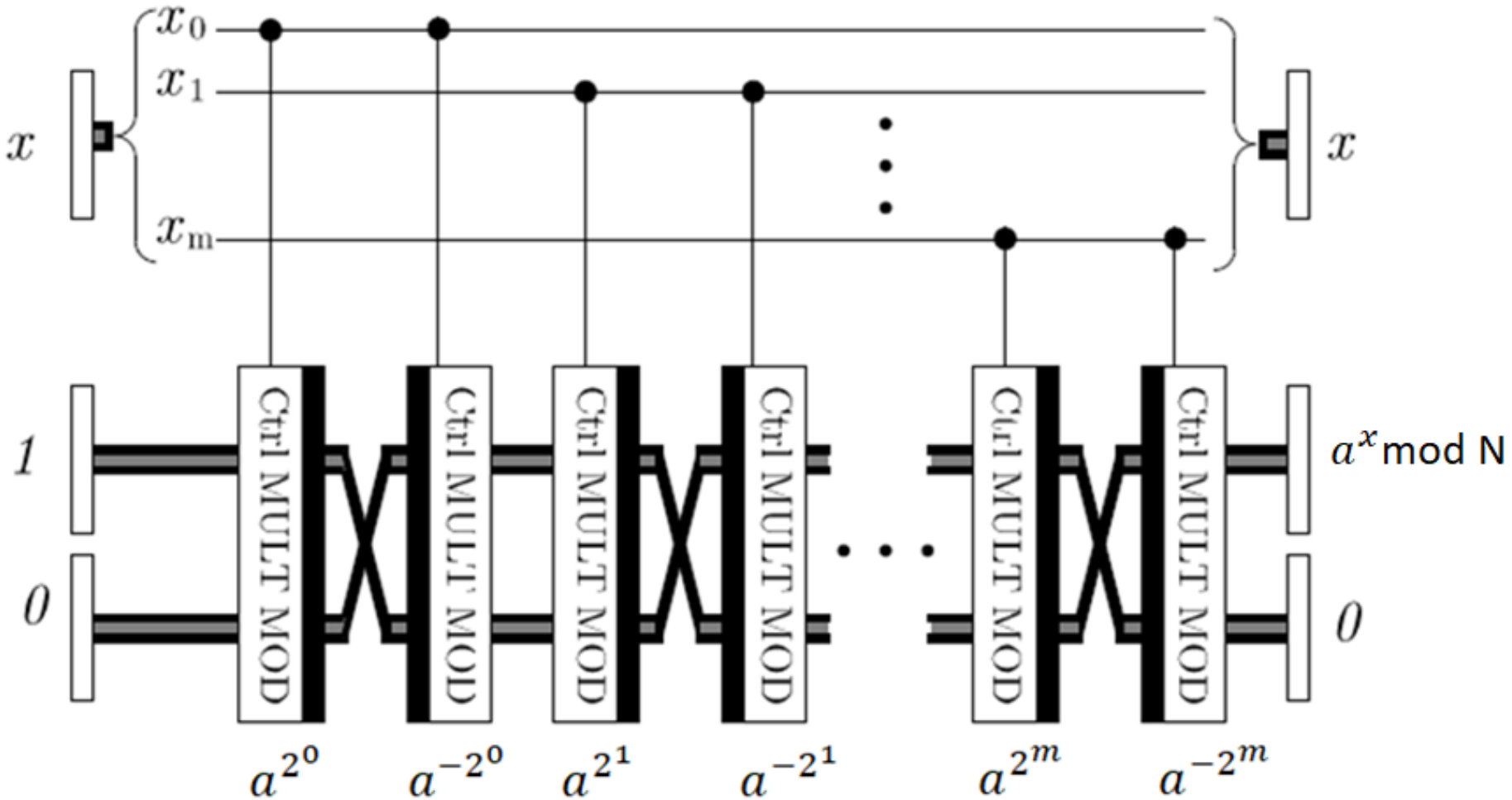
Controlled multiplication

$$y = zx = z2^n x_n + \cdots + z2^1 x_1 + z2^0 x_0$$



Modular exponentiation

$$a^x = a^{2^n x_n} \cdots a^{2^1 x_1} a^{2^0 x_0}$$

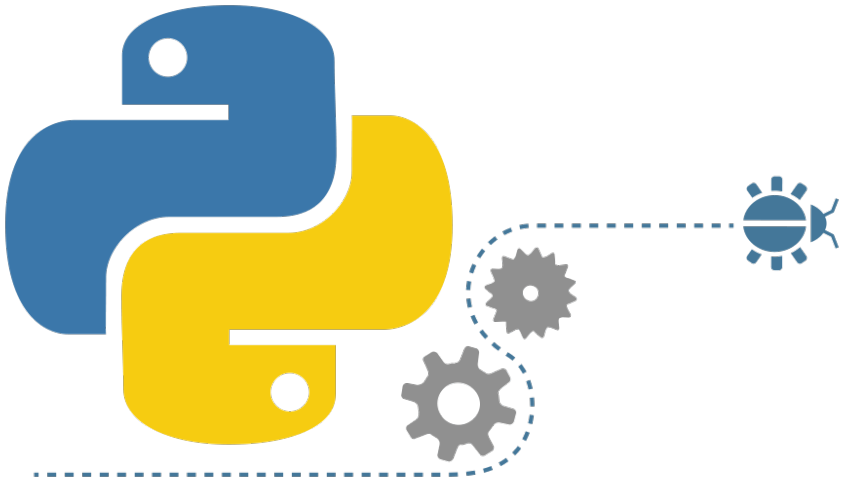


Software

QX

- Easy and intuitive to create a simple script
- No built in recursive operations to create a longer/more complex script

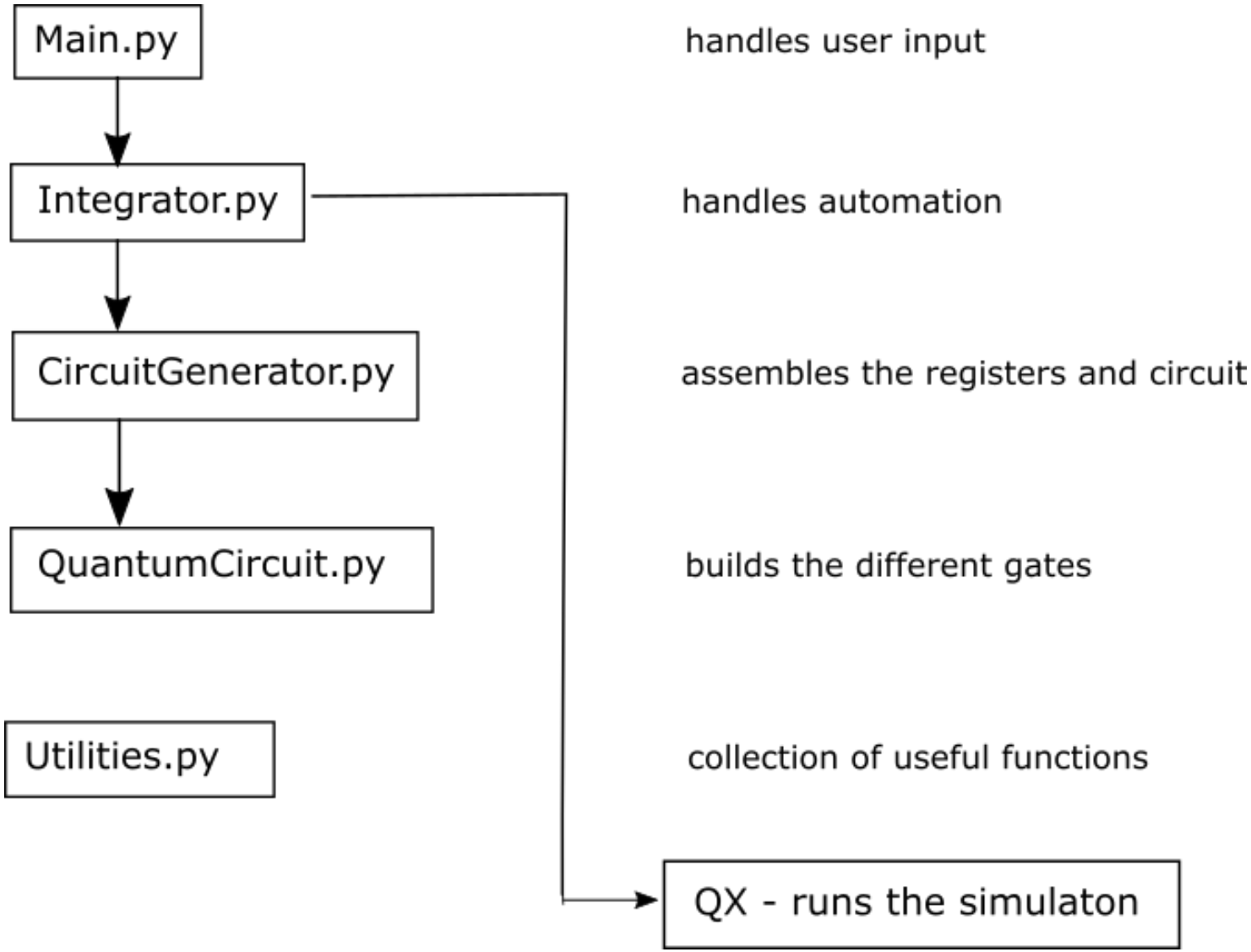
Software



Python

Python wrapper implementing the classical part and the automation of the qc file generation

Python and QX



Results

What can we factorize?

- Factorizing 21 $\rightarrow$ 37 qubits needed - Impossible even in our quantum dreams
- Factorizing 15 $\rightarrow$ 30 qubits needed - Impossible on student budget
- Factorizing 15 (cheating) $\rightarrow$ 26 qubits needed - Possible, 15 is indeed $3 * 5$!

The end

Questions and Answers