What do quantum computers do?

Daniel J. Bernstein
University of Illinois at Chicago

"Quantum algorithm" means an algorithm that a quantum computer can run.

i.e. a sequence of instructions, where each instruction is in a quantum computer's supported instruction set.

How do we know which instructions a quantum computer will support?

Quantum computer type 1 (QC1): contains many "qubits"; can efficiently perform "NOT gate", "Hadamard gate", "controlled NOT gate", "T gate".

What do quantum computers do?

Daniel J. Bernstein
University of Illinois at Chicago

"Quantum algorithm" means an algorithm that a quantum computer can run.

i.e. a sequence of instructions, where each instruction is in a quantum computer's supported instruction set.

How do we know which instructions a quantum computer will support?

Quantum computer type 1 (QC1): contains many "qubits"; can efficiently perform "NOT gate", "Hadamard gate", "controlled NOT gate", "T gate".

Making these instructions work is the main goal of quantum-computer engineering.

What do quantum computers do?

Daniel J. Bernstein
University of Illinois at Chicago

"Quantum algorithm"
means an algorithm that
a quantum computer can run.

i.e. a sequence of instructions,
where each instruction is
in a quantum computer's
supported instruction set.

How do we know which instructions a quantum computer will support?

Quantum computer type 1 (QC1):
contains many "qubits";
can efficiently perform
"NOT gate", "Hadamard gate",
"controlled NOT gate", "T gate".

Making these instructions work is the main goal of quantum-computer engineering.

Combine these instructions
to compute "Toffoli gate";
... "Simon's algorithm";
... "Shor's algorithm"; etc.

What do quantum computers do?

Daniel J. Bernstein
University of Illinois at Chicago

"Quantum algorithm" means an algorithm that a quantum computer can run.

i.e. a sequence of instructions, where each instruction is in a quantum computer's supported instruction set.

How do we know which instructions a quantum computer will support?

Quantum computer type 1 (QC1): contains many "qubits"; can efficiently perform "NOT gate", "Hadamard gate", "controlled NOT gate", "T gate".

Making these instructions work is the main goal of quantum-computer engineering.

Combine these instructions to compute "Toffoli gate"; . . . "Simon's algorithm"; . . . "Shor's algorithm"; etc.

General belief: Traditional CPU isn't QC1; e.g. can't factor quickly.

o quantum computers do?

. Bernstein
ty of Illinois at Chicago

um algorithm"
n algorithm that
um computer can run.

quence of instructions,
ach instruction is
ntum computer's
ed instruction set.

we know which
ions a quantum
er will support?

Quantum computer type 1 (QC1):
contains many "qubits";
can efficiently perform
"NOT gate", "Hadamard gate",
"controlled NOT gate", "*T* gate".

Making these instructions work is the main goal of quantum-computer engineering.

Combine these instructions
to compute "Toffoli gate";
. . . "Simon's algorithm";
. . . "Shor's algorithm"; etc.

General belief: Traditional CPU
isn't QC1; e.g. can't factor quickly.

Quantur
stores a
efficientl
laws of c
with as r

This is t
quantum
by 1982
physics v

computers do?

n

is at Chicago

---

hm"

m that

ter can run.

instructions,

ction is

puter's

ion set.

which

antum

pport?

Quantum computer type 1 (QC1):
contains many "qubits";
can efficiently perform
"NOT gate", "Hadamard gate",
"controlled NOT gate", "*T* gate".

Making these instructions work is the main goal of quantum-computer engineering.

Combine these instructions
to compute "Toffoli gate";
. . . "Simon's algorithm";
. . . "Shor's algorithm"; etc.

General belief: Traditional CPU isn't QC1; e.g. can't factor quickly.

Quantum compute

stores a simulated

efficiently simulate

laws of quantum p

with as much accu

This is the origina

quantum compute

by 1982 Feynman

physics with comp

rs do?

ago

n.

ns,

Quantum computer type 1 (QC1): contains many "qubits"; can efficiently perform "NOT gate", "Hadamard gate", "controlled NOT gate", "*T* gate".

Making these instructions work is the main goal of quantum-computer engineering.

Combine these instructions to compute "Toffoli gate"; . . . "Simon's algorithm"; . . . "Shor's algorithm"; etc.

General belief: Traditional CPU isn't QC1; e.g. can't factor quickly.

Quantum computer type 2 (

stores a simulated universe; efficiently simulates the laws of quantum physics with as much accuracy as de

This is the original concept quantum computers introdu by 1982 Feynman "Simulati physics with computers".

Quantum computer type 1 (QC1): contains many "qubits"; can efficiently perform "NOT gate", "Hadamard gate", "controlled NOT gate", "*T* gate".

Making these instructions work is the main goal of quantum-computer engineering.

Combine these instructions to compute "Toffoli gate"; ... "Simon's algorithm"; ... "Shor's algorithm"; etc.

General belief: Traditional CPU isn't QC1; e.g. can't factor quickly.

Quantum computer type 2 (QC2): stores a simulated universe; efficiently simulates the laws of quantum physics with as much accuracy as desired.

This is the original concept of quantum computers introduced by 1982 Feynman "Simulating physics with computers".

Quantum computer type 1 (QC1):
contains many "qubits";
can efficiently perform
"NOT gate", "Hadamard gate",
"controlled NOT gate", "*T* gate".

Making these instructions work is the main goal of quantum-computer engineering.

Combine these instructions
to compute "Toffoli gate";
. . . "Simon's algorithm";
. . . "Shor's algorithm"; etc.

General belief: Traditional CPU
isn't QC1; e.g. can't factor quickly.

Quantum computer type 2 (QC2):
stores a simulated universe;
efficiently simulates the
laws of quantum physics
with as much accuracy as desired.

This is the original concept of quantum computers introduced by 1982 Feynman "Simulating physics with computers".

General belief: any QC1 is a QC2.
Partial proof: see, e.g.,
2011 Jordan–Lee–Preskill
"Quantum algorithms for
quantum field theories".

m computer type 1 (QC1):
many "qubits";
iently perform
ate", "Hadamard gate",
led NOT gate", "T gate".

these instructions work
ain goal of quantum-
er engineering.

these instructions
ute "Toffoli gate";
non's algorithm";
r's algorithm"; etc.

belief: Traditional CPU
1; e.g. can't factor quickly.

Quantum computer type 2 (QC2):
stores a simulated universe;
efficiently simulates the
laws of quantum physics
with as much accuracy as desired.

This is the original concept of
quantum computers introduced
by 1982 Feynman "Simulating
physics with computers".

General belief: any QC1 is a QC2.
Partial proof: see, e.g.,
2011 Jordan–Lee–Preskill
"Quantum algorithms for
quantum field theories".

Quantur
efficientl
that any
compute

er type 1 (QC1):
ubits";
form
damard gate",
gate", "T gate".

structions work
of quantum-
ering.

tructions
oli gate";
rithm";
thm"; etc.

aditional CPU
n't factor quickly.

Quantum computer type 2 (QC2): stores a simulated universe; efficiently simulates the laws of quantum physics with as much accuracy as desired.

This is the original concept of quantum computers introduced by 1982 Feynman "Simulating physics with computers".

General belief: any QC1 is a QC2. Partial proof: see, e.g., 2011 Jordan–Lee–Preskill "Quantum algorithms for quantum field theories".

Quantum compute
efficiently compute
that any possible p
computer can com

(QC1):

ate”,
gate”.

work
um-

CPU
quickly.

Quantum computer type 2 (QC2):
stores a simulated universe;
efficiently simulates the
laws of quantum physics
with as much accuracy as desired.

This is the original concept of quantum computers introduced by 1982 Feynman “Simulating physics with computers”.

General belief: any QC1 is a QC2.
Partial proof: see, e.g.,
2011 Jordan–Lee–Preskill
“Quantum algorithms for
quantum field theories”.

Quantum computer type 3 (
efficiently computes anythin
that any possible physical
computer can compute effici

Quantum computer type 2 (QC2): stores a simulated universe; efficiently simulates the laws of quantum physics with as much accuracy as desired.

This is the original concept of quantum computers introduced by 1982 Feynman "Simulating physics with computers".

General belief: any QC1 is a QC2. Partial proof: see, e.g., 2011 Jordan–Lee–Preskill "Quantum algorithms for quantum field theories".

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

Quantum computer type 2 (QC2): stores a simulated universe; efficiently simulates the laws of quantum physics with as much accuracy as desired.

This is the original concept of quantum computers introduced by 1982 Feynman "Simulating physics with computers".

General belief: any QC1 is a QC2. Partial proof: see, e.g., 2011 Jordan–Lee–Preskill "Quantum algorithms for quantum field theories".

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3. Argument for belief: any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

Quantum computer type 2 (QC2): stores a simulated universe; efficiently simulates the laws of quantum physics with as much accuracy as desired.

This is the original concept of quantum computers introduced by 1982 Feynman "Simulating physics with computers".

General belief: any QC1 is a QC2. Partial proof: see, e.g., 2011 Jordan–Lee–Preskill "Quantum algorithms for quantum field theories".

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3. Argument for belief: any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

General belief: any QC3 is a QC1. Argument for belief: look, we're building a QC1.

m computer type 2 (QC2):
simulated universe;
y simulates the
quantum physics
much accuracy as desired.

he original concept of
n computers introduced
Feynman "Simulating
with computers".

belief: any QC1 is a QC2.
proof: see, e.g.,
rdan–Lee–Preskill
um algorithms for
n field theories".

Quantum computer type 3 (QC3):
efficiently computes anything
that any possible physical
computer can compute efficiently.

General belief: any QC2 is a QC3.
Argument for belief:
any physical computer must
follow the laws of quantum
physics, so a QC2 can efficiently
simulate any physical computer.

General belief: any QC3 is a QC1.
Argument for belief:
look, we're building a QC1.

A note c

Apparen
Current
from D-
can be r
simulate

er type 2 (QC2):
universe;
es the
physics
uracy as desired.

l concept of
rs introduced
"Simulating
uters".

y QC1 is a QC2.
e.g.,
Preskill
hms for
ories".

Quantum computer type 3 (QC3):
efficiently computes anything
that any possible physical
computer can compute efficiently.

General belief: any QC2 is a QC3.
Argument for belief:
any physical computer must
follow the laws of quantum
physics, so a QC2 can efficiently
simulate any physical computer.

General belief: any QC3 is a QC1.
Argument for belief:
look, we're building a QC1.

A note on D-Wave

Apparent scientific
Current "quantum
from D-Wave are
can be more cost-
simulated by tradi

(QC2):

esired.

of

ced

ng

QC2.

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3.
Argument for belief:
any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

General belief: any QC3 is a QC1.
Argument for belief:
look, we're building a QC1.

A note on D-Wave

Apparent scientific consensu
Current "quantum compute
from D-Wave are useless—
can be more cost-effectively
simulated by traditional CP

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3.
Argument for belief:
any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

General belief: any QC3 is a QC1.
Argument for belief:
look, we're building a QC1.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3. Argument for belief: any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

General belief: any QC3 is a QC1. Argument for belief: look, we're building a QC1.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless—can be more cost-effectively simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3. Argument for belief: any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

General belief: any QC3 is a QC1. Argument for belief: look, we're building a QC1.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;
- selling some machines;

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3. Argument for belief: any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

General belief: any QC3 is a QC1. Argument for belief: look, we're building a QC1.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3. Argument for belief: any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

General belief: any QC3 is a QC1. Argument for belief: look, we're building a QC1.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Quantum computer type 3 (QC3): efficiently computes anything that any possible physical computer can compute efficiently.

General belief: any QC2 is a QC3. Argument for belief: any physical computer must follow the laws of quantum physics, so a QC2 can efficiently simulate any physical computer.

General belief: any QC3 is a QC1. Argument for belief: look, we're building a QC1.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is
- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

m computer type 3 (QC3):
y computes anything
possible physical
er can compute efficiently.

belief: any QC2 is a QC3.
nt for belief:
sical computer must
ne laws of quantum
so a QC2 can efficiently
any physical computer.

belief: any QC3 is a QC1.
nt for belief:
're building a QC1.

A note on D-Wave

Apparent scientific consensus:
Current "quantum computers"
from D-Wave are useless—
can be more cost-effectively
simulated by traditional CPUs.

But D-Wave is
- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

The stat

Data ("s
a list of
e.g.: (0,

er type 3 (QC3):
es anything
physical
npute efficiently.

y QC2 is a QC3.
ef:
puter must
quantum
can efficiently
ical computer.

y QC3 is a QC1.
ef:
g a QC1.

A note on D-Wave

Apparent scientific consensus:
Current "quantum computers"
from D-Wave are useless—
can be more cost-effectively
simulated by traditional CPUs.

But D-Wave is
- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

The state of a con

Data ("state") sto
a list of 3 element
e.g.: $(0, 0, 0)$.

(QC3):
g
iently.
n QC3.
ently
uter.
n QC1.

A note on D-Wave

Apparent scientific consensus:
Current "quantum computers"
from D-Wave are useless—
can be more cost-effectively
simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

The state of a computer

Data ("state") stored in 3 b
a list of 3 elements of $\{0, 1\}$
e.g.: $(0, 0, 0)$.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

The state of a computer

Data ("state") stored in 3 bits: a list of 3 elements of $\{0, 1\}$. e.g.: $(0, 0, 0)$.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is
- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

The state of a computer

Data ("state") stored in 3 bits: a list of 3 elements of $\{0, 1\}$.

e.g.: $(0, 0, 0)$.

e.g.: $(1, 1, 1)$.

A note on D-Wave

Apparent scientific consensus:
Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

The state of a computer

Data ("state") stored in 3 bits: a list of 3 elements of $\{0, 1\}$.

e.g.: $(0, 0, 0)$.

e.g.: $(1, 1, 1)$.

e.g.: $(0, 1, 1)$.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

The state of a computer

Data ("state") stored in 3 bits: a list of 3 elements of $\{0, 1\}$.

e.g.: $(0, 0, 0)$.

e.g.: $(1, 1, 1)$.

e.g.: $(0, 1, 1)$.

Data stored in 64 bits: a list of 64 elements of $\{0, 1\}$.

A note on D-Wave

Apparent scientific consensus: Current "quantum computers" from D-Wave are useless— can be more cost-effectively simulated by traditional CPUs.

But D-Wave is

- collecting venture capital;
- selling some machines;
- collecting possibly useful engineering expertise;
- not being punished for deceiving people.

Is D-Wave a bad investment?

The state of a computer

Data ("state") stored in 3 bits: a list of 3 elements of $\{0, 1\}$.

e.g.: $(0, 0, 0)$.

e.g.: $(1, 1, 1)$.

e.g.: $(0, 1, 1)$.

Data stored in 64 bits: a list of 64 elements of $\{0, 1\}$.

e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

on D-Wave

t scientific consensus:
"quantum computers"
Wave are useless—
more cost-effectively
d by traditional CPUs.

Vave is
ting venture capital;
some machines;
ting possibly useful
eering expertise;
eing punished
ceiving people.

ve a bad investment?

The state of a computer

Data ("state") stored in 3 bits:
a list of 3 elements of $\{0, 1\}$.
e.g.: $(0, 0, 0)$.
e.g.: $(1, 1, 1)$.
e.g.: $(0, 1, 1)$.

Data stored in 64 bits:
a list of 64 elements of $\{0, 1\}$.
e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The stat

Data sto
a list of
e.g.: (3,

e

c consensus:
n computers”
useless—
effectively
tional CPUs.

re capital;
chines;
ly useful
ertise;
ned
ople.

nvestment?

The state of a computer

Data (“state”) stored in 3 bits:
a list of 3 elements of $\{0, 1\}$.
e.g.: $(0, 0, 0)$.
e.g.: $(1, 1, 1)$.
e.g.: $(0, 1, 1)$.

Data stored in 64 bits:
a list of 64 elements of $\{0, 1\}$.
e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The state of a qua

Data stored in 3 q
a list of 8 number
e.g.: $(3, 1, 4, 1, 5,$

s:

rs"

Us.

t?

The state of a computer

Data ("state") stored in 3 bits:

a list of 3 elements of $\{0, 1\}$.

e.g.: $(0, 0, 0)$.

e.g.: $(1, 1, 1)$.

e.g.: $(0, 1, 1)$.

Data stored in 64 bits:

a list of 64 elements of $\{0, 1\}$.

e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The state of a quantum com

Data stored in 3 qubits:

a list of 8 numbers, not all z

e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.

The state of a computer

Data ("state") stored in 3 bits:
a list of 3 elements of $\{0, 1\}$.
e.g.: $(0, 0, 0)$.
e.g.: $(1, 1, 1)$.
e.g.: $(0, 1, 1)$.

Data stored in 64 bits:
a list of 64 elements of $\{0, 1\}$.
e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.

The state of a computer

Data ("state") stored in 3 bits:

a list of 3 elements of $\{0, 1\}$.

e.g.: $(0, 0, 0)$.

e.g.: $(1, 1, 1)$.

e.g.: $(0, 1, 1)$.

Data stored in 64 bits:

a list of 64 elements of $\{0, 1\}$.

e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The state of a quantum computer

Data stored in 3 qubits:

a list of 8 numbers, not all zero.

e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.

e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.

The state of a computer

Data ("state") stored in 3 bits:
a list of 3 elements of $\{0, 1\}$.
e.g.: $(0, 0, 0)$.
e.g.: $(1, 1, 1)$.
e.g.: $(0, 1, 1)$.

Data stored in 64 bits:
a list of 64 elements of $\{0, 1\}$.
e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

The state of a computer

Data ("state") stored in 3 bits:
a list of 3 elements of $\{0, 1\}$.
e.g.: $(0, 0, 0)$.
e.g.: $(1, 1, 1)$.
e.g.: $(0, 1, 1)$.

Data stored in 64 bits:
a list of 64 elements of $\{0, 1\}$.
e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

The state of a computer

Data ("state") stored in 3 bits:
a list of 3 elements of $\{0, 1\}$.
e.g.: $(0, 0, 0)$.
e.g.: $(1, 1, 1)$.
e.g.: $(0, 1, 1)$.

Data stored in 64 bits:
a list of 64 elements of $\{0, 1\}$.
e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

The state of a computer

Data ("state") stored in 3 bits:
a list of 3 elements of $\{0, 1\}$.
e.g.: $(0, 0, 0)$.
e.g.: $(1, 1, 1)$.
e.g.: $(0, 1, 1)$.

Data stored in 64 bits:
a list of 64 elements of $\{0, 1\}$.
e.g.: $(1, 1, 1, 1, 1, 0, 0, 0, 1,$
$0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,$
$0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,$
$0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,$
$1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1)$.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

Data stored in 1000 qubits: a list of 2^{1000} numbers, not all zero.

te of a computer

state”) stored in 3 bits:

3 elements of $\{0, 1\}$.

0, 0).

1, 1).

1, 1).

ored in 64 bits:

64 elements of $\{0, 1\}$.

1, 1, 1, 1, 0, 0, 0, 1,

, 0, 0, 1, 1, 0, 0, 0,

, 1, 0, 0, 0, 0, 0, 1,

, 0, 0, 1, 0, 0, 0, 1,

, 1, 0, 0, 1, 0, 0, 0,

, 1, 0, 0, 1, 0, 0, 1).

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.

e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.

e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.

e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

Data stored in 1000 qubits: a list of 2^{1000} numbers, not all zero.

Measuri

Can sim

Cannot

of numb

mputer

ored in 3 bits:

s of $\{0, 1\}$.

bits:

nts of $\{0, 1\}$.

$0, 0, 0, 1,$

$0, 0, 0,$

$0, 0, 1,$

$0, 0, 1,$

$0, 0, 0,$

$0, 0, 1)$.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

Data stored in 1000 qubits: a list of 2^{1000} numbers, not all zero.

Measuring a quant

Can simply look a

Cannot simply loo

of numbers stored

its:

·.

}.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.

e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.

e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.

e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.: $(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

Data stored in 1000 qubits: a list of 2^{1000} numbers, not all zero.

Measuring a quantum comp

Can simply look at a bit.

Cannot simply look at the li

of numbers stored in n qubi

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

Data stored in 1000 qubits: a list of 2^{1000} numbers, not all zero.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

Data stored in 1000 qubits: a list of 2^{1000} numbers, not all zero.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits

- produces n bits and
- destroys the state.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

Data stored in 1000 qubits: a list of 2^{1000} numbers, not all zero.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits

- produces n bits and
- destroys the state.

If n qubits have state $(a_0, a_1, \ldots, a_{2^n-1})$ then measurement produces q with probability $|a_q|^2 / \sum_r |a_r|^2$.

The state of a quantum computer

Data stored in 3 qubits:
a list of 8 numbers, not all zero.
e.g.: $(3, 1, 4, 1, 5, 9, 2, 6)$.
e.g.: $(-2, 7, -1, 8, 1, -8, -2, 8)$.
e.g.: $(0, 0, 0, 0, 0, 1, 0, 0)$.

Data stored in 4 qubits: a list of 16 numbers, not all zero. e.g.:
$(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

Data stored in 64 qubits:
a list of 2^{64} numbers, not all zero.

Data stored in 1000 qubits: a list of 2^{1000} numbers, not all zero.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits

- produces n bits and
- destroys the state.

If n qubits have state $(a_0, a_1, \ldots, a_{2^n-1})$ then measurement produces q with probability $|a_q|^2 / \sum_r |a_r|^2$.

State is then all zeros except 1 at position q.

te of a quantum computer

ored in 3 qubits:

8 numbers, not all zero.

$1, 4, 1, 5, 9, 2, 6)$.

$2, 7, -1, 8, 1, -8, -2, 8)$.

$0, 0, 0, 0, 1, 0, 0)$.

ored in 4 qubits: a list of

bers, not all zero. e.g.:

$1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3)$.

ored in 64 qubits:

2^{64} numbers, not all zero.

ored in 1000 qubits: a list

numbers, not all zero.

Measuring a quantum computer

Can simply look at a bit.

Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits

- produces n bits and
- destroys the state.

If n qubits have state $(a_0, a_1, \ldots, a_{2^n-1})$ then measurement produces q with probability $|a_q|^2 / \sum_r |a_r|^2$.

State is then all zeros except 1 at position q.

e.g.: Say

$(1, 1, 1,$

antum computer

ubits:

s, not all zero.

$9, 2, 6)$.

$, 1, -8, -2, 8)$.

$1, 0, 0)$.

ubits: a list of

ll zero. e.g.:

$, 5, 3, 5, 8, 9, 7, 9, 3)$.

qubits:

ers, not all zero.

00 qubits: a list

not all zero.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list
of numbers stored in n qubits.

Measuring n qubits

- produces n bits and
- destroys the state.

If n qubits have state
$(a_0, a_1, \ldots, a_{2^n-1})$ then
measurement produces q
with probability $|a_q|^2 / \sum_r |a_r|^2$.

State is then all zeros
except 1 at position q.

e.g.: Say 3 qubits

$(1, 1, 1, 1, 1, 1, 1, 1$

nputer

zero.

2, 8).

st of
g.:
9, 7, 9, 3).

l zero.

a list
ro.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits
- produces n bits and
- destroys the state.

If n qubits have state $(a_0, a_1, \ldots, a_{2^n-1})$ then measurement produces q with probability $|a_q|^2 / \sum_r |a_r|^2$.

State is then all zeros except 1 at position q.

e.g.: Say 3 qubits have stat
$(1, 1, 1, 1, 1, 1, 1, 1)$.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits

- produces n bits and
- destroys the state.

If n qubits have state $(a_0, a_1, \ldots, a_{2^n-1})$ then measurement produces q with probability $|a_q|^2 / \sum_r |a_r|^2$.

State is then all zeros except 1 at position q.

e.g.: Say 3 qubits have state $(1, 1, 1, 1, 1, 1, 1, 1)$.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits

- produces n bits and
- destroys the state.

If n qubits have state $(a_0, a_1, \ldots, a_{2^n-1})$ then measurement produces q with probability $|a_q|^2 / \sum_r |a_r|^2$.

State is then all zeros except 1 at position q.

e.g.: Say 3 qubits have state $(1, 1, 1, 1, 1, 1, 1, 1)$.

Measurement produces
$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits
- produces n bits and
- destroys the state.

If n qubits have state $(a_0, a_1, \ldots, a_{2^n-1})$ then measurement produces q with probability $|a_q|^2 / \sum_r |a_r|^2$.

State is then all zeros except 1 at position q.

e.g.: Say 3 qubits have state $(1, 1, 1, 1, 1, 1, 1, 1)$.

Measurement produces
$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

"Quantum RNG."

Measuring a quantum computer

Can simply look at a bit.
Cannot simply look at the list of numbers stored in n qubits.

Measuring n qubits
- produces n bits and
- destroys the state.

If n qubits have state $(a_0, a_1, \ldots, a_{2^n-1})$ then measurement produces q with probability $|a_q|^2 / \sum_r |a_r|^2$.

State is then all zeros except 1 at position q.

e.g.: Say 3 qubits have state $(1, 1, 1, 1, 1, 1, 1, 1)$.

Measurement produces
$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

"Quantum RNG."

Warning: Quantum RNGs sold today are measurably biased.

ng a quantum computer

ply look at a bit.
simply look at the list
pers stored in n qubits.

ing n qubits
ces n bits and
ys the state.

its have state
$\ldots, a_{2^n-1})$ then
ment produces q
bability $|a_q|^2 / \sum_r |a_r|^2$.

then all zeros
at position q.

e.g.: Say 3 qubits have state
$(1, 1, 1, 1, 1, 1, 1, 1)$.

Measurement produces
$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

"Quantum RNG."

Warning: Quantum RNGs sold
today are measurably biased.

e.g.: Say
$(3, 1, 4,$

tum computer

t a bit.

k at the list

in n qubits.

its

and

te.

ate

then

luces q

$_q|^2/\sum_r |a_r|^2$.

eros

on q.

e.g.: Say 3 qubits have state $(1,1,1,1,1,1,1,1)$.

Measurement produces

$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

"Quantum RNG."

Warning: Quantum RNGs sold today are measurably biased.

e.g.: Say 3 qubits

$(3,1,4,1,5,9,2,6$

uter

st

ts.

$a_r|^2$.

e.g.: Say 3 qubits have state $(1, 1, 1, 1, 1, 1, 1, 1)$.

Measurement produces
$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

"Quantum RNG."

Warning: Quantum RNGs sold today are measurably biased.

e.g.: Say 3 qubits have state $(3, 1, 4, 1, 5, 9, 2, 6)$.

e.g.: Say 3 qubits have state $(1,1,1,1,1,1,1,1)$.

Measurement produces
$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

"Quantum RNG."

Warning: Quantum RNGs sold today are measurably biased.

e.g.: Say 3 qubits have state $(3,1,4,1,5,9,2,6)$.

e.g.: Say 3 qubits have state $(1, 1, 1, 1, 1, 1, 1, 1)$.

Measurement produces
$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

"Quantum RNG."

Warning: Quantum RNGs sold today are measurably biased.

e.g.: Say 3 qubits have state $(3, 1, 4, 1, 5, 9, 2, 6)$.

Measurement produces
$000 = 0$ with probability $9/173$;
$001 = 1$ with probability $1/173$;
$010 = 2$ with probability $16/173$;
$011 = 3$ with probability $1/173$;
$100 = 4$ with probability $25/173$;
$101 = 5$ with probability $81/173$;
$110 = 6$ with probability $4/173$;
$111 = 7$ with probability $36/173$.

e.g.: Say 3 qubits have state $(1, 1, 1, 1, 1, 1, 1, 1)$.

Measurement produces
$000 = 0$ with probability $1/8$;
$001 = 1$ with probability $1/8$;
$010 = 2$ with probability $1/8$;
$011 = 3$ with probability $1/8$;
$100 = 4$ with probability $1/8$;
$101 = 5$ with probability $1/8$;
$110 = 6$ with probability $1/8$;
$111 = 7$ with probability $1/8$.

"Quantum RNG."

Warning: Quantum RNGs sold today are measurably biased.

e.g.: Say 3 qubits have state $(3, 1, 4, 1, 5, 9, 2, 6)$.

Measurement produces
$000 = 0$ with probability $9/173$;
$001 = 1$ with probability $1/173$;
$010 = 2$ with probability $16/173$;
$011 = 3$ with probability $1/173$;
$100 = 4$ with probability $25/173$;
$101 = 5$ with probability $81/173$;
$110 = 6$ with probability $4/173$;
$111 = 7$ with probability $36/173$.

5 is most likely outcome.

y 3 qubits have state

$1, 1, 1, 1, 1)$.

ement produces

with probability $1/8$;

with probability $1/8$;

with probability $1/8$;

with probability $1/8$;

with probability $1/8$;

with probability $1/8$;

with probability $1/8$;

with probability $1/8$.

um RNG."

g: Quantum RNGs sold

e measurably biased.

e.g.: Say 3 qubits have state

$(3, 1, 4, 1, 5, 9, 2, 6)$.

Measurement produces

$000 = 0$ with probability $9/173$;

$001 = 1$ with probability $1/173$;

$010 = 2$ with probability $16/173$;

$011 = 3$ with probability $1/173$;

$100 = 4$ with probability $25/173$;

$101 = 5$ with probability $81/173$;

$110 = 6$ with probability $4/173$;

$111 = 7$ with probability $36/173$.

5 is most likely outcome.

e.g.: Say

$(0, 0, 0, 0$

have state

).

duces

ability 1/8;

ability 1/8;

ability 1/8;

ability 1/8;

ability 1/8;

ability 1/8;

ability 1/8;

ability 1/8.

m RNGs sold

bly biased.

e.g.: Say 3 qubits have state

$(3, 1, 4, 1, 5, 9, 2, 6)$.

Measurement produces

$000 = 0$ with probability $9/173$;

$001 = 1$ with probability $1/173$;

$010 = 2$ with probability $16/173$;

$011 = 3$ with probability $1/173$;

$100 = 4$ with probability $25/173$;

$101 = 5$ with probability $81/173$;

$110 = 6$ with probability $4/173$;

$111 = 7$ with probability $36/173$.

5 is most likely outcome.

e.g.: Say 3 qubits

$(0, 0, 0, 0, 0, 1, 0, 0$

e

3;

3;

3;

3;

3;

3;

3;

3.

old

.

e.g.: Say 3 qubits have state $(3, 1, 4, 1, 5, 9, 2, 6)$.

Measurement produces

$000 = 0$ with probability $9/173$;

$001 = 1$ with probability $1/173$;

$010 = 2$ with probability $16/173$;

$011 = 3$ with probability $1/173$;

$100 = 4$ with probability $25/173$;

$101 = 5$ with probability $81/173$;

$110 = 6$ with probability $4/173$;

$111 = 7$ with probability $36/173$.

5 is most likely outcome.

e.g.: Say 3 qubits have state $(0, 0, 0, 0, 0, 1, 0, 0)$.

e.g.: Say 3 qubits have state $(3, 1, 4, 1, 5, 9, 2, 6)$.

Measurement produces
$000 = 0$ with probability $9/173$;
$001 = 1$ with probability $1/173$;
$010 = 2$ with probability $16/173$;
$011 = 3$ with probability $1/173$;
$100 = 4$ with probability $25/173$;
$101 = 5$ with probability $81/173$;
$110 = 6$ with probability $4/173$;
$111 = 7$ with probability $36/173$.

5 is most likely outcome.

e.g.: Say 3 qubits have state $(0, 0, 0, 0, 0, 1, 0, 0)$.

e.g.: Say 3 qubits have state $(3, 1, 4, 1, 5, 9, 2, 6)$.

Measurement produces
$000 = 0$ with probability $9/173$;
$001 = 1$ with probability $1/173$;
$010 = 2$ with probability $16/173$;
$011 = 3$ with probability $1/173$;
$100 = 4$ with probability $25/173$;
$101 = 5$ with probability $81/173$;
$110 = 6$ with probability $4/173$;
$111 = 7$ with probability $36/173$.

5 is most likely outcome.

e.g.: Say 3 qubits have state $(0, 0, 0, 0, 0, 1, 0, 0)$.

Measurement produces
$000 = 0$ with probability 0;
$001 = 1$ with probability 0;
$010 = 2$ with probability 0;
$011 = 3$ with probability 0;
$100 = 4$ with probability 0;
$101 = 5$ with probability 1;
$110 = 6$ with probability 0;
$111 = 7$ with probability 0.

e.g.: Say 3 qubits have state
$(3, 1, 4, 1, 5, 9, 2, 6)$.

Measurement produces
$000 = 0$ with probability $9/173$;
$001 = 1$ with probability $1/173$;
$010 = 2$ with probability $16/173$;
$011 = 3$ with probability $1/173$;
$100 = 4$ with probability $25/173$;
$101 = 5$ with probability $81/173$;
$110 = 6$ with probability $4/173$;
$111 = 7$ with probability $36/173$.

5 is most likely outcome.

e.g.: Say 3 qubits have state
$(0, 0, 0, 0, 0, 1, 0, 0)$.

Measurement produces
$000 = 0$ with probability 0;
$001 = 1$ with probability 0;
$010 = 2$ with probability 0;
$011 = 3$ with probability 0;
$100 = 4$ with probability 0;
$101 = 5$ with probability 1;
$110 = 6$ with probability 0;
$111 = 7$ with probability 0.

5 is guaranteed outcome.

y 3 qubits have state
1, 5, 9, 2, 6).

ement produces
with probability 9/173;
with probability 1/173;
with probability 16/173;
with probability 1/173;
with probability 25/173;
with probability 81/173;
with probability 4/173;
with probability 36/173.

t likely outcome.

e.g.: Say 3 qubits have state
$(0, 0, 0, 0, 0, 1, 0, 0)$.

Measurement produces
$000 = 0$ with probability 0;
$001 = 1$ with probability 0;
$010 = 2$ with probability 0;
$011 = 3$ with probability 0;
$100 = 4$ with probability 0;
$101 = 5$ with probability 1;
$110 = 6$ with probability 0;
$111 = 7$ with probability 0.

5 is guaranteed outcome.

NOT ga

NOT_0 g
(3, 1, 4, 1
(1, 3, 1, 4

have state

).

duces

ability $9/173$;

ability $1/173$;

ability $16/173$;

ability $1/173$;

ability $25/173$;

ability $81/173$;

ability $4/173$;

ability $36/173$.

tcome.

e.g.: Say 3 qubits have state
$(0, 0, 0, 0, 0, 1, 0, 0)$.

Measurement produces

$000 = 0$ with probability 0;

$001 = 1$ with probability 0;

$010 = 2$ with probability 0;

$011 = 3$ with probability 0;

$100 = 4$ with probability 0;

$101 = 5$ with probability 1;

$110 = 6$ with probability 0;

$111 = 7$ with probability 0.

5 is guaranteed outcome.

NOT gates

NOT_0 gate on 3 q

$(3, 1, 4, 1, 5, 9, 2, 6$

$(1, 3, 1, 4, 9, 5, 6, 2$

e

173;

173;

/173;

173;

/173;

/173;

173;

/173.

e.g.: Say 3 qubits have state $(0, 0, 0, 0, 0, 1, 0, 0)$.

Measurement produces

$000 = 0$ with probability 0;

$001 = 1$ with probability 0;

$010 = 2$ with probability 0;

$011 = 3$ with probability 0;

$100 = 4$ with probability 0;

$101 = 5$ with probability 1;

$110 = 6$ with probability 0;

$111 = 7$ with probability 0.

5 is guaranteed outcome.

NOT gates

NOT_0 gate on 3 qubits: $(3, 1, 4, 1, 5, 9, 2, 6) \mapsto (1, 3, 1, 4, 9, 5, 6, 2)$.

e.g.: Say 3 qubits have state $(0, 0, 0, 0, 0, 1, 0, 0)$.

Measurement produces
$000 = 0$ with probability 0;
$001 = 1$ with probability 0;
$010 = 2$ with probability 0;
$011 = 3$ with probability 0;
$100 = 4$ with probability 0;
$101 = 5$ with probability 1;
$110 = 6$ with probability 0;
$111 = 7$ with probability 0.

5 is guaranteed outcome.

NOT gates

NOT_0 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto (1, 3, 1, 4, 9, 5, 6, 2)$.

e.g.: Say 3 qubits have state $(0, 0, 0, 0, 0, 1, 0, 0)$.

Measurement produces
$000 = 0$ with probability 0;
$001 = 1$ with probability 0;
$010 = 2$ with probability 0;
$011 = 3$ with probability 0;
$100 = 4$ with probability 0;
$101 = 5$ with probability 1;
$110 = 6$ with probability 0;
$111 = 7$ with probability 0.

5 is guaranteed outcome.

NOT gates

NOT_0 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto (1, 3, 1, 4, 9, 5, 6, 2)$.

NOT_0 gate on 4 qubits:
$(3,1,4,1,5,9,2,6,5,3,5,8,9,7,9,3) \mapsto (1,3,1,4,9,5,6,2,3,5,8,5,7,9,3,9)$.

e.g.: Say 3 qubits have state
$(0, 0, 0, 0, 0, 1, 0, 0)$.

Measurement produces
$000 = 0$ with probability 0;
$001 = 1$ with probability 0;
$010 = 2$ with probability 0;
$011 = 3$ with probability 0;
$100 = 4$ with probability 0;
$101 = 5$ with probability 1;
$110 = 6$ with probability 0;
$111 = 7$ with probability 0.

5 is guaranteed outcome.

NOT gates

NOT_0 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(1, 3, 1, 4, 9, 5, 6, 2)$.

NOT_0 gate on 4 qubits:
$(3,1,4,1,5,9,2,6,5,3,5,8,9,7,9,3) \mapsto$
$(1,3,1,4,9,5,6,2,3,5,8,5,7,9,3,9)$.

NOT_1 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(4, 1, 3, 1, 2, 6, 5, 9)$.

e.g.: Say 3 qubits have state $(0,0,0,0,0,1,0,0)$.

Measurement produces
$000 = 0$ with probability 0;
$001 = 1$ with probability 0;
$010 = 2$ with probability 0;
$011 = 3$ with probability 0;
$100 = 4$ with probability 0;
$101 = 5$ with probability 1;
$110 = 6$ with probability 0;
$111 = 7$ with probability 0.

5 is guaranteed outcome.

NOT gates

NOT_0 gate on 3 qubits:
$(3,1,4,1,5,9,2,6) \mapsto (1,3,1,4,9,5,6,2)$.

NOT_0 gate on 4 qubits:
$(3,1,4,1,5,9,2,6,5,3,5,8,9,7,9,3) \mapsto (1,3,1,4,9,5,6,2,3,5,8,5,7,9,3,9)$.

NOT_1 gate on 3 qubits:
$(3,1,4,1,5,9,2,6) \mapsto (4,1,3,1,2,6,5,9)$.

NOT_2 gate on 3 qubits:
$(3,1,4,1,5,9,2,6) \mapsto (5,9,2,6,3,1,4,1)$.

y 3 qubits have state

$0, 0, 1, 0, 0)$.

ement produces

with probability 0;

with probability 0;

with probability 0;

with probability 0;

with probability 0;

with probability 1;

with probability 0;

with probability 0.

ranteed outcome.

NOT gates

NOT_0 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(1, 3, 1, 4, 9, 5, 6, 2)$.

NOT_0 gate on 4 qubits:
$(3,1,4,1,5,9,2,6,5,3,5,8,9,7,9,3) \mapsto$
$(1,3,1,4,9,5,6,2,3,5,8,5,7,9,3,9)$.

NOT_1 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(4, 1, 3, 1, 2, 6, 5, 9)$.

NOT_2 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(5, 9, 2, 6, 3, 1, 4, 1)$.

$(1, 0, 0,$

$(0, 1, 0,$

$(0, 0, 1,$

$(0, 0, 0,$

$(0, 0, 0,$

$(0, 0, 0,$

$(0, 0, 0,$

$(0, 0, 0,$

Operatio

NOT_0, s

Operatio

flipping

Flip: ou

have state
).

duces
ability 0;
ability 0;
ability 0;
ability 0;
ability 0;
ability 1;
ability 0;
ability 0.

tcome.

NOT gates

NOT_0 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(1, 3, 1, 4, 9, 5, 6, 2)$.

NOT_0 gate on 4 qubits:
$(3,1,4,1,5,9,2,6,5,3,5,8,9,7,9,3) \mapsto$
$(1,3,1,4,9,5,6,2,3,5,8,5,7,9,3,9)$.

NOT_1 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(4, 1, 3, 1, 2, 6, 5, 9)$.

NOT_2 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(5, 9, 2, 6, 3, 1, 4, 1)$.

state
(1, 0, 0, 0, 0, 0, 0, 0
(0, 1, 0, 0, 0, 0, 0, 0
(0, 0, 1, 0, 0, 0, 0, 0
(0, 0, 0, 1, 0, 0, 0, 0
(0, 0, 0, 0, 1, 0, 0, 0
(0, 0, 0, 0, 0, 1, 0, 0
(0, 0, 0, 0, 0, 0, 1, 0
(0, 0, 0, 0, 0, 0, 0, 1

Operation on quar
NOT_0, swapping p
Operation after m
flipping bit 0 of re
Flip: output is not

e

NOT gates

NOT_0 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(1, 3, 1, 4, 9, 5, 6, 2)$.

NOT_0 gate on 4 qubits:
$(3,1,4,1,5,9,2,6,5,3,5,8,9,7,9,3) \mapsto$
$(1,3,1,4,9,5,6,2,3,5,8,5,7,9,3,9)$.

NOT_1 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(4, 1, 3, 1, 2, 6, 5, 9)$.

NOT_2 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(5, 9, 2, 6, 3, 1, 4, 1)$.

state	measure
$(1, 0, 0, 0, 0, 0, 0, 0)$	000
$(0, 1, 0, 0, 0, 0, 0, 0)$	001
$(0, 0, 1, 0, 0, 0, 0, 0)$	010
$(0, 0, 0, 1, 0, 0, 0, 0)$	011
$(0, 0, 0, 0, 1, 0, 0, 0)$	100
$(0, 0, 0, 0, 0, 1, 0, 0)$	101
$(0, 0, 0, 0, 0, 0, 1, 0)$	110
$(0, 0, 0, 0, 0, 0, 0, 1)$	111

Operation on quantum state
NOT_0, swapping pairs.
Operation after measuremen
flipping bit 0 of result.
Flip: output is not input.

NOT gates

NOT_0 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(1, 3, 1, 4, 9, 5, 6, 2)$.

NOT_0 gate on 4 qubits:
$(3,1,4,1,5,9,2,6,5,3,5,8,9,7,9,3) \mapsto$
$(1,3,1,4,9,5,6,2,3,5,8,5,7,9,3,9)$.

NOT_1 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(4, 1, 3, 1, 2, 6, 5, 9)$.

NOT_2 gate on 3 qubits:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(5, 9, 2, 6, 3, 1, 4, 1)$.

state	measurement
$(1, 0, 0, 0, 0, 0, 0, 0)$	000
$(0, 1, 0, 0, 0, 0, 0, 0)$	001
$(0, 0, 1, 0, 0, 0, 0, 0)$	010
$(0, 0, 0, 1, 0, 0, 0, 0)$	011
$(0, 0, 0, 0, 1, 0, 0, 0)$	100
$(0, 0, 0, 0, 0, 1, 0, 0)$	101
$(0, 0, 0, 0, 0, 0, 1, 0)$	110
$(0, 0, 0, 0, 0, 0, 0, 1)$	111

Operation on quantum state:
NOT_0, swapping pairs.
Operation after measurement:
flipping bit 0 of result.
Flip: output is not input.

tes

ate on 3 qubits:

1, 5, 9, 2, 6) ↦

4, 9, 5, 6, 2).

ate on 4 qubits:

5,9,2,6,5,3,5,8,9,7,9,3) ↦

9,5,6,2,3,5,8,5,7,9,3,9).

ate on 3 qubits:

1, 5, 9, 2, 6) ↦

1, 2, 6, 5, 9).

ate on 3 qubits:

1, 5, 9, 2, 6) ↦

6, 3, 1, 4, 1).

state	measurement
$(1,0,0,0,0,0,0,0)$	000
$(0,1,0,0,0,0,0,0)$	001
$(0,0,1,0,0,0,0,0)$	010
$(0,0,0,1,0,0,0,0)$	011
$(0,0,0,0,1,0,0,0)$	100
$(0,0,0,0,0,1,0,0)$	101
$(0,0,0,0,0,0,1,0)$	110
$(0,0,0,0,0,0,0,1)$	111

Operation on quantum state:

NOT_0, swapping pairs.

Operation after measurement:

flipping bit 0 of result.

Flip: output is not input.

Controll

e.g. CN

(3, 1, 4, 1

(3, 1, 1, 4

ubits:

) $\mapsto$

).

ubits:

3,5,8,9,7,9,3) $\mapsto$

5,8,5,7,9,3,9).

ubits:

) $\mapsto$

).

ubits:

) $\mapsto$

).

state	measurement
$(1,0,0,0,0,0,0,0)$	000
$(0,1,0,0,0,0,0,0)$	001
$(0,0,1,0,0,0,0,0)$	010
$(0,0,0,1,0,0,0,0)$	011
$(0,0,0,0,1,0,0,0)$	100
$(0,0,0,0,0,1,0,0)$	101
$(0,0,0,0,0,0,1,0)$	110
$(0,0,0,0,0,0,0,1)$	111

Operation on quantum state:

NOT_0, swapping pairs.

Operation after measurement:

flipping bit 0 of result.

Flip: output is not input.

Controlled-NOT g

e.g. $\text{CNOT}_{1,0}$:

$(3, 1, 4, 1, 5, 9, 2, 6$

$(3, 1, 1, 4, 5, 9, 6, 2$

$,3) \mapsto$

$,9)$.

state	measurement
$(1,0,0,0,0,0,0,0)$	000
$(0,1,0,0,0,0,0,0)$	001
$(0,0,1,0,0,0,0,0)$	010
$(0,0,0,1,0,0,0,0)$	011
$(0,0,0,0,1,0,0,0)$	100
$(0,0,0,0,0,1,0,0)$	101
$(0,0,0,0,0,0,1,0)$	110
$(0,0,0,0,0,0,0,1)$	111

Operation on quantum state:
NOT_0, swapping pairs.
Operation after measurement:
flipping bit 0 of result.
Flip: output is not input.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:
$(3,1,4,1,5,9,2,6) \mapsto$
$(3,1,1,4,5,9,6,2)$.

state	measurement
$(1,0,0,0,0,0,0,0)$	000
$(0,1,0,0,0,0,0,0)$	001
$(0,0,1,0,0,0,0,0)$	010
$(0,0,0,1,0,0,0,0)$	011
$(0,0,0,0,1,0,0,0)$	100
$(0,0,0,0,0,1,0,0)$	101
$(0,0,0,0,0,0,1,0)$	110
$(0,0,0,0,0,0,0,1)$	111

Operation on quantum state:
NOT_0, swapping pairs.
Operation after measurement:
flipping bit 0 of result.
Flip: output is not input.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:
$(3,1,4,1,5,9,2,6) \mapsto$
$(3,1,1,4,5,9,6,2)$.

state	measurement
$(1,0,0,0,0,0,0,0)$	000
$(0,1,0,0,0,0,0,0)$	001
$(0,0,1,0,0,0,0,0)$	010
$(0,0,0,1,0,0,0,0)$	011
$(0,0,0,0,1,0,0,0)$	100
$(0,0,0,0,0,1,0,0)$	101
$(0,0,0,0,0,0,1,0)$	110
$(0,0,0,0,0,0,0,1)$	111

Operation on quantum state: NOT_0, swapping pairs.

Operation after measurement: flipping bit 0 of result.

Flip: output is not input.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$: $(3,1,4,1,5,9,2,6) \mapsto (3,1,1,4,5,9,6,2)$.

Operation after measurement: flipping bit 0 *if* bit 1 is set; i.e., $(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

state	measurement
$(1,0,0,0,0,0,0,0)$	000
$(0,1,0,0,0,0,0,0)$	001
$(0,0,1,0,0,0,0,0)$	010
$(0,0,0,1,0,0,0,0)$	011
$(0,0,0,0,1,0,0,0)$	100
$(0,0,0,0,0,1,0,0)$	101
$(0,0,0,0,0,0,1,0)$	110
$(0,0,0,0,0,0,0,1)$	111

Operation on quantum state: NOT_0, swapping pairs.

Operation after measurement: flipping bit 0 of result.

Flip: output is not input.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$: $(3,1,4,1,5,9,2,6) \mapsto (3,1,1,4,5,9,6,2)$.

Operation after measurement: flipping bit 0 *if* bit 1 is set; i.e., $(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

e.g. $\text{CNOT}_{2,0}$: $(3,1,4,1,5,9,2,6) \mapsto (3,1,4,1,9,5,6,2)$.

state	measurement
$(1,0,0,0,0,0,0,0)$	000
$(0,1,0,0,0,0,0,0)$	001
$(0,0,1,0,0,0,0,0)$	010
$(0,0,0,1,0,0,0,0)$	011
$(0,0,0,0,1,0,0,0)$	100
$(0,0,0,0,0,1,0,0)$	101
$(0,0,0,0,0,0,1,0)$	110
$(0,0,0,0,0,0,0,1)$	111

Operation on quantum state: NOT_0, swapping pairs.

Operation after measurement: flipping bit 0 of result.

Flip: output is not input.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:
$(3,1,4,1,5,9,2,6) \mapsto$
$(3,1,1,4,5,9,6,2)$.

Operation after measurement: flipping bit 0 *if* bit 1 is set; i.e., $(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

e.g. $\text{CNOT}_{2,0}$:
$(3,1,4,1,5,9,2,6) \mapsto$
$(3,1,4,1,9,5,6,2)$.

e.g. $\text{CNOT}_{0,2}$:
$(3,1,4,1,5,9,2,6) \mapsto$
$(3,9,4,6,5,1,2,1)$.

state	measurement
0, 0, 0, 0, 0)	000
0, 0, 0, 0, 0)	001
0, 0, 0, 0, 0)	010
1, 0, 0, 0, 0)	011
0, 1, 0, 0, 0)	100
0, 0, 1, 0, 0)	101
0, 0, 0, 1, 0)	110
0, 0, 0, 0, 1)	111

on on quantum state:
swapping pairs.
on after measurement:
bit 0 of result.
tput is not input.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 1, 4, 5, 9, 6, 2)$.

Operation after measurement:
flipping bit 0 *if* bit 1 is set; i.e.,
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

e.g. $\text{CNOT}_{2,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 9, 5, 6, 2)$.

e.g. $\text{CNOT}_{0,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 9, 4, 6, 5, 1, 2, 1)$.

Toffoli g

Also kno
controlle

e.g. CCN
(3, 1, 4, 1
(3, 1, 4, 1

measurement

0)	000
0)	001
0)	010
0)	011
0)	100
0)	101
0)	110
1)	111

ntum state:

pairs.

easurement:

sult.

t input.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:

$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$

$(3, 1, 1, 4, 5, 9, 6, 2)$.

Operation after measurement:

flipping bit 0 *if* bit 1 is set; i.e.,

$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

e.g. $\text{CNOT}_{2,0}$:

$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$

$(3, 1, 4, 1, 9, 5, 6, 2)$.

e.g. $\text{CNOT}_{0,2}$:

$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$

$(3, 9, 4, 6, 5, 1, 2, 1)$.

Toffoli gates

Also known as

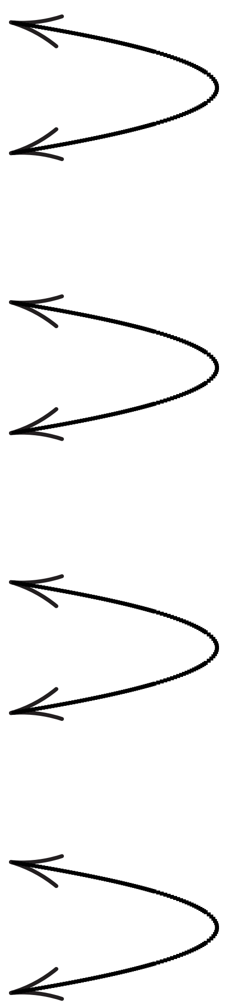
controlled-controll

e.g. $\text{CCNOT}_{2,1,0}$:

(3, 1, 4, 1, 5, 9, 2, 6

(3, 1, 4, 1, 5, 9, 6, 2

ement



e:

nt:

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 1, 4, 5, 9, 6, 2)$.

Operation after measurement:
flipping bit 0 *if* bit 1 is set; i.e.,
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

e.g. $\text{CNOT}_{2,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 9, 5, 6, 2)$.

e.g. $\text{CNOT}_{0,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 9, 4, 6, 5, 1, 2, 1)$.

Toffoli gates

Also known as
controlled-controlled-NOT g

e.g. $\text{CCNOT}_{2,1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 5, 9, 6, 2)$.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 1, 4, 5, 9, 6, 2)$.

Operation after measurement:
flipping bit 0 *if* bit 1 is set; i.e.,
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

e.g. $\text{CNOT}_{2,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 9, 5, 6, 2)$.

e.g. $\text{CNOT}_{0,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 9, 4, 6, 5, 1, 2, 1)$.

Toffoli gates

Also known as
controlled-controlled-NOT gates.

e.g. $\text{CCNOT}_{2,1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 5, 9, 6, 2)$.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 1, 4, 5, 9, 6, 2)$.

Operation after measurement:
flipping bit 0 *if* bit 1 is set; i.e.,
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

e.g. $\text{CNOT}_{2,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 9, 5, 6, 2)$.

e.g. $\text{CNOT}_{0,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 9, 4, 6, 5, 1, 2, 1)$.

Toffoli gates

Also known as
controlled-controlled-NOT gates.

e.g. $\text{CCNOT}_{2,1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 5, 9, 6, 2)$.

Operation after measurement:
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1 q_2)$.

Controlled-NOT gates

e.g. $\text{CNOT}_{1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 1, 4, 5, 9, 6, 2)$.

Operation after measurement:
flipping bit 0 *if* bit 1 is set; i.e.,
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

e.g. $\text{CNOT}_{2,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 9, 5, 6, 2)$.

e.g. $\text{CNOT}_{0,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 9, 4, 6, 5, 1, 2, 1)$.

Toffoli gates

Also known as
controlled-controlled-NOT gates.

e.g. $\text{CCNOT}_{2,1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 5, 9, 6, 2)$.

Operation after measurement:
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1 q_2)$.

e.g. $\text{CCNOT}_{0,1,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 6, 5, 9, 2, 1)$.

ed-NOT gates

OT$_{1,0}$:

1, 5, 9, 2, 6) $\mapsto$

4, 5, 9, 6, 2).

on after measurement:

bit 0 *if* bit 1 is set; i.e.,

$q_0) \mapsto (q_2, q_1, q_0 \oplus q_1)$.

OT$_{2,0}$:

1, 5, 9, 2, 6) $\mapsto$

1, 9, 5, 6, 2).

OT$_{0,2}$:

1, 5, 9, 2, 6) $\mapsto$

6, 5, 1, 2, 1).

Toffoli gates

Also known as controlled-controlled-NOT gates.

e.g. CCNOT$_{2,1,0}$:

$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$

$(3, 1, 4, 1, 5, 9, 6, 2)$.

Operation after measurement:

$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1 q_2)$.

e.g. CCNOT$_{0,1,2}$:

$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$

$(3, 1, 4, 6, 5, 9, 2, 1)$.

More sh

Combine

to build

ates

) $\mapsto$

).

easurement:

t 1 is set; i.e.,

, $q_1, q_0 \oplus q_1$).

) $\mapsto$

).

) $\mapsto$

).

Toffoli gates

Also known as controlled-controlled-NOT gates.

e.g. $\text{CCNOT}_{2,1,0}$:

$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$

$(3, 1, 4, 1, 5, 9, 6, 2)$.

Operation after measurement:

$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1 q_2)$.

e.g. $\text{CCNOT}_{0,1,2}$:

$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$

$(3, 1, 4, 6, 5, 9, 2, 1)$.

More shuffling

Combine NOT, CN

to build other perr

nt:
i.e.,
q_1).

Toffoli gates

Also known as controlled-controlled-NOT gates.

e.g. $\text{CCNOT}_{2,1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto (3, 1, 4, 1, 5, 9, 6, 2)$.

Operation after measurement:
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1 q_2)$.

e.g. $\text{CCNOT}_{0,1,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto (3, 1, 4, 6, 5, 9, 2, 1)$.

More shuffling

Combine NOT, CNOT, Toff
to build other permutations.

Toffoli gates

Also known as controlled-controlled-NOT gates.

e.g. $\text{CCNOT}_{2,1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto (3, 1, 4, 1, 5, 9, 6, 2)$.

Operation after measurement:
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1 q_2)$.

e.g. $\text{CCNOT}_{0,1,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto (3, 1, 4, 6, 5, 9, 2, 1)$.

More shuffling

Combine NOT, CNOT, Toffoli to build other permutations.

Toffoli gates

Also known as
controlled-controlled-NOT gates.

e.g. $\text{CCNOT}_{2,1,0}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 1, 5, 9, 6, 2)$.

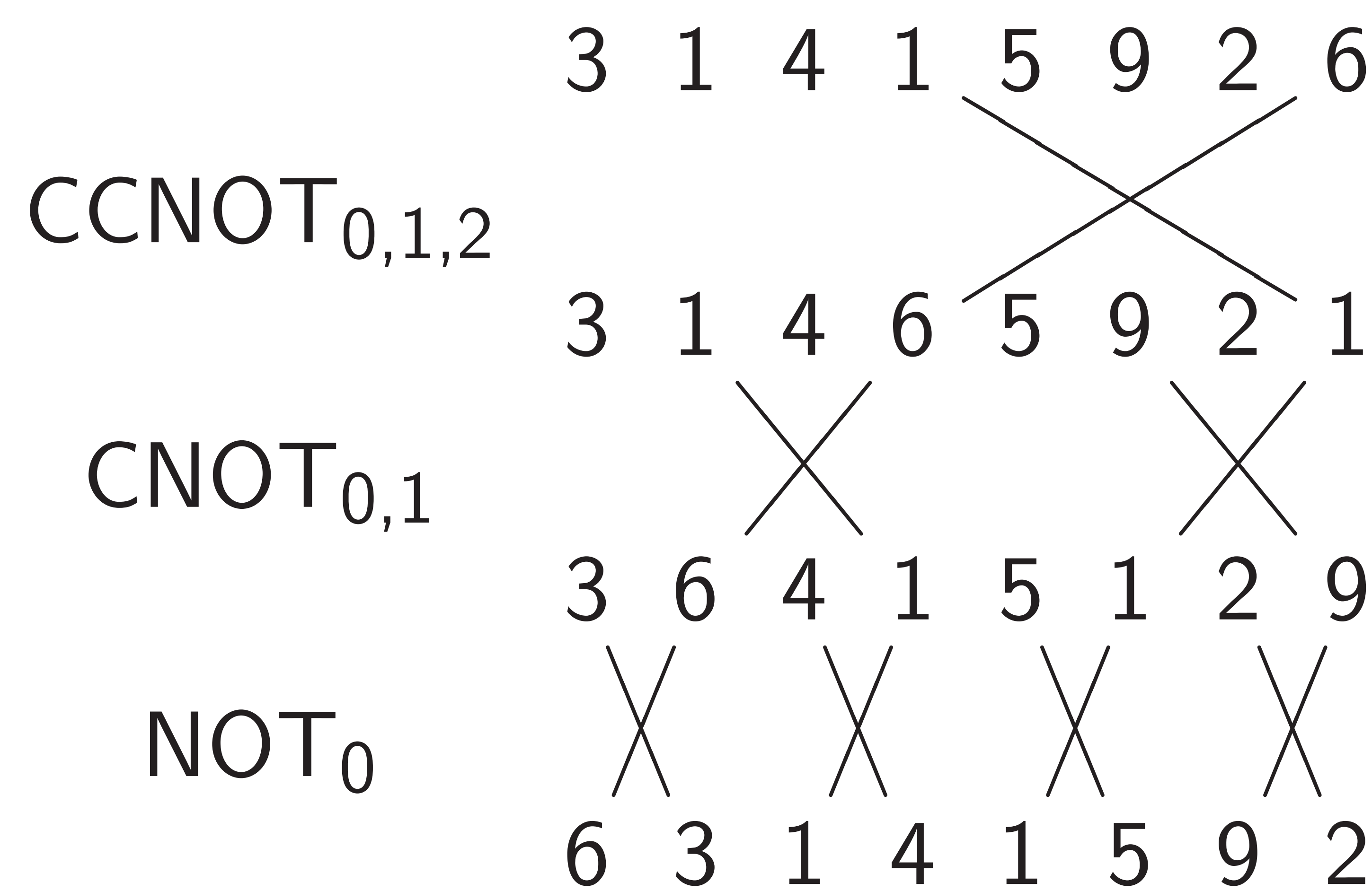
Operation after measurement:
$(q_2, q_1, q_0) \mapsto (q_2, q_1, q_0 \oplus q_1 q_2)$.

e.g. $\text{CCNOT}_{0,1,2}$:
$(3, 1, 4, 1, 5, 9, 2, 6) \mapsto$
$(3, 1, 4, 6, 5, 9, 2, 1)$.

More shuffling

Combine NOT, CNOT, Toffoli
to build other permutations.

e.g. series of gates to
rotate 8 positions by distance 1:



ates

own as

ed-controlled-NOT gates.

$\text{NOT}_{2,1,0}$:

$1, 5, 9, 2, 6) \mapsto$

$1, 5, 9, 6, 2)$.

on after measurement:

$q_0) \mapsto (q_2, q_1, q_0 \oplus q_1 q_2)$.

$\text{NOT}_{0,1,2}$:

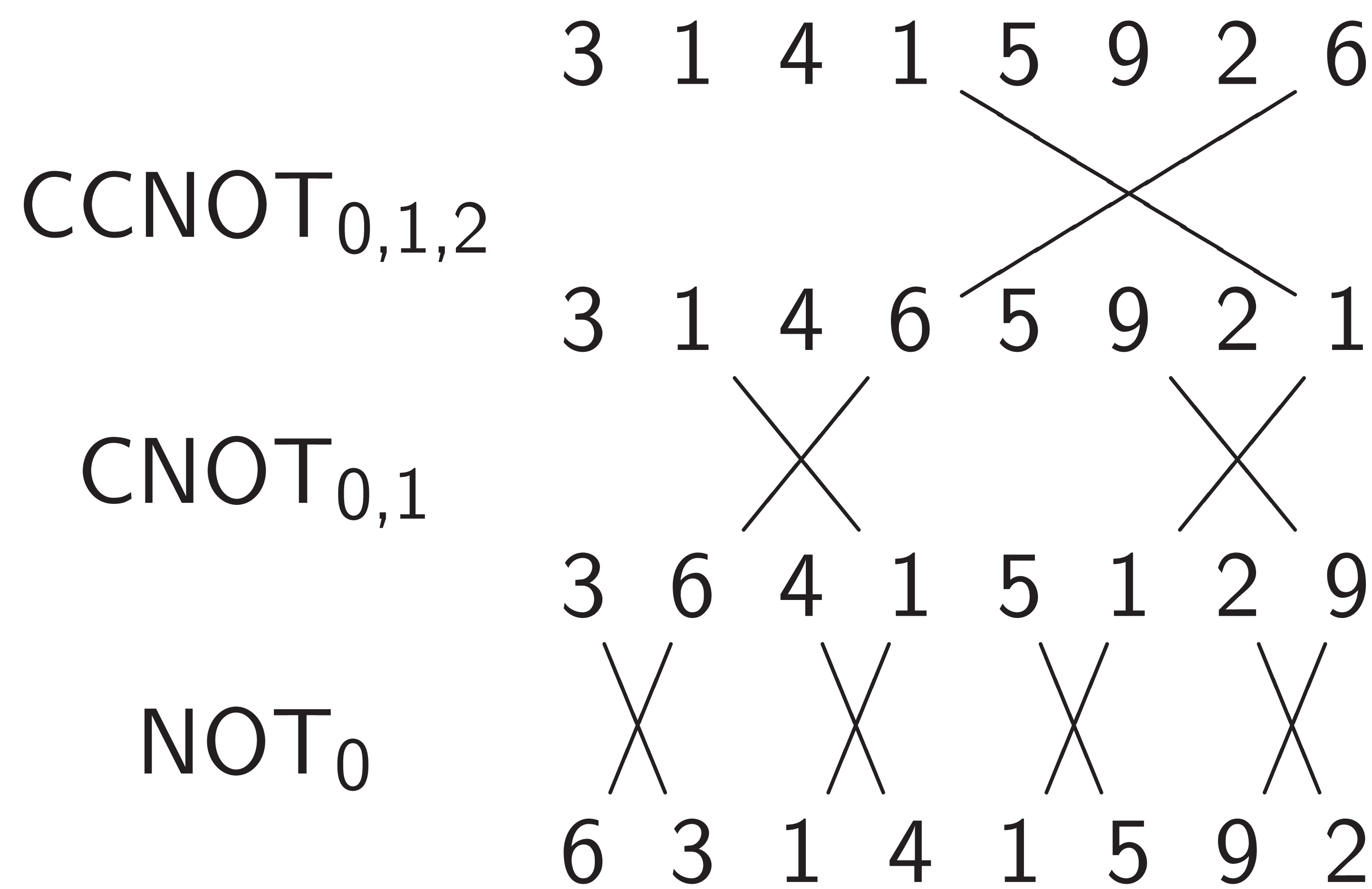
$1, 5, 9, 2, 6) \mapsto$

$6, 5, 9, 2, 1)$.

More shuffling

Combine NOT, CNOT, Toffoli to build other permutations.

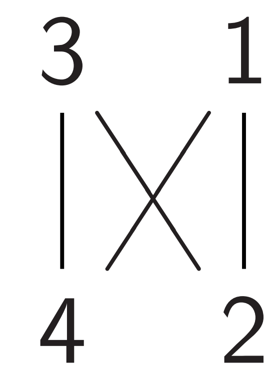
e.g. series of gates to rotate 8 positions by distance 1:



Hadama

Hadama

$(a, b) \mapsto$



ed-NOT gates.

$) \mapsto$

$).$

easurement:

$, q_1, q_0 \oplus q_1 q_2).$

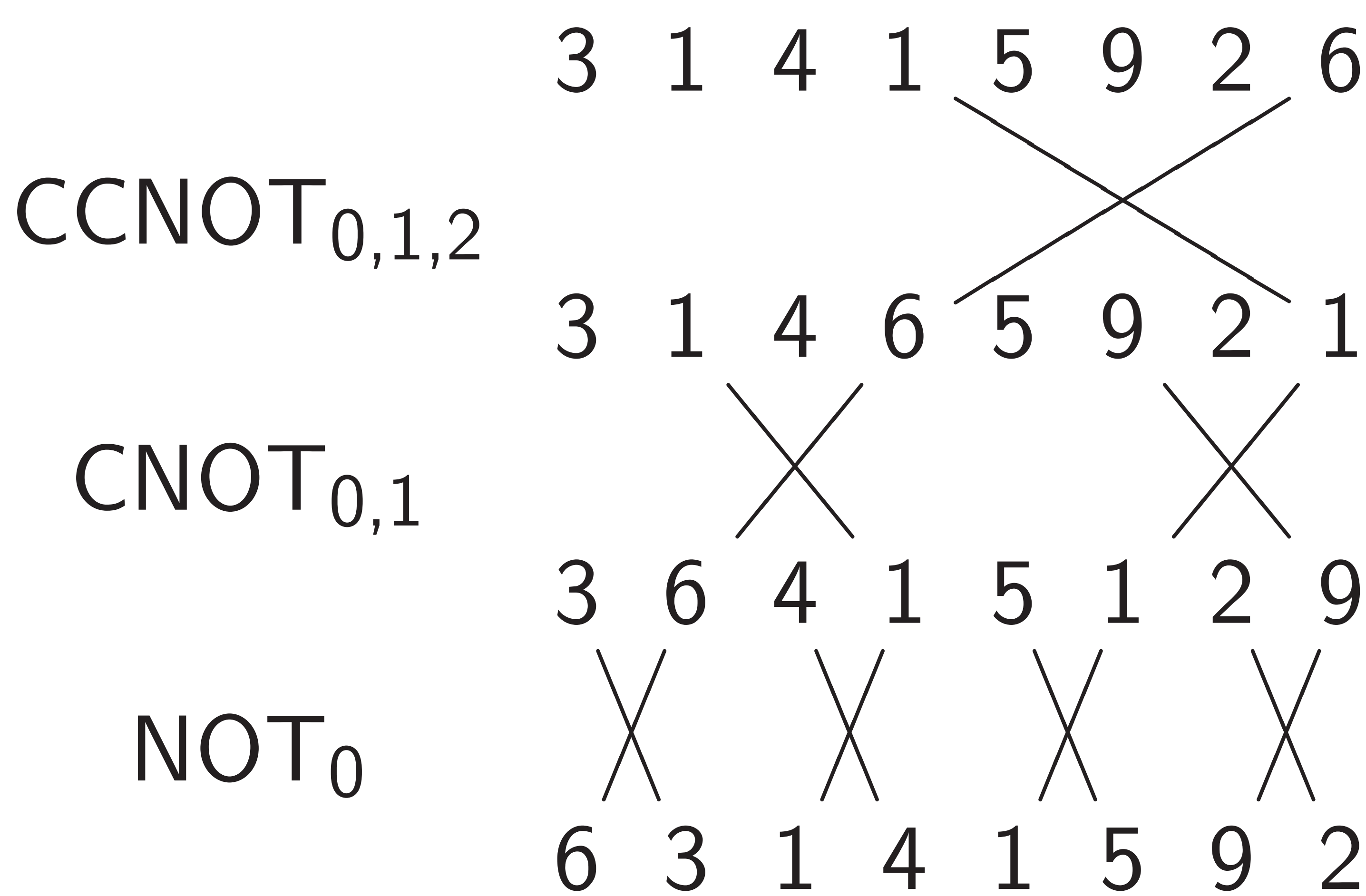
$) \mapsto$

$).$

More shuffling

Combine NOT, CNOT, Toffoli
to build other permutations.

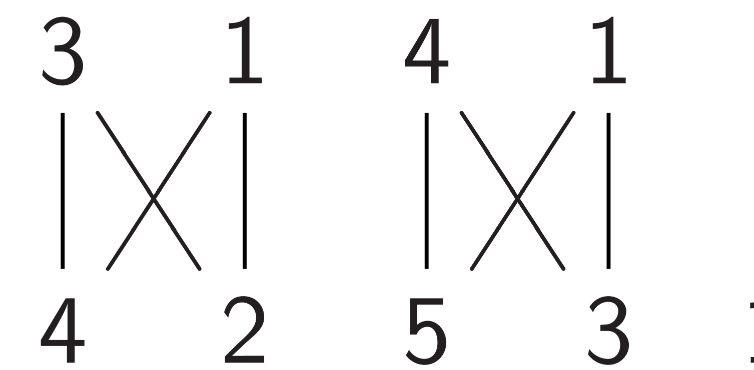
e.g. series of gates to
rotate 8 positions by distance 1:



Hadamard gates

$Hadamard_0$:

$(a, b) \mapsto (a + b, a$



ates.

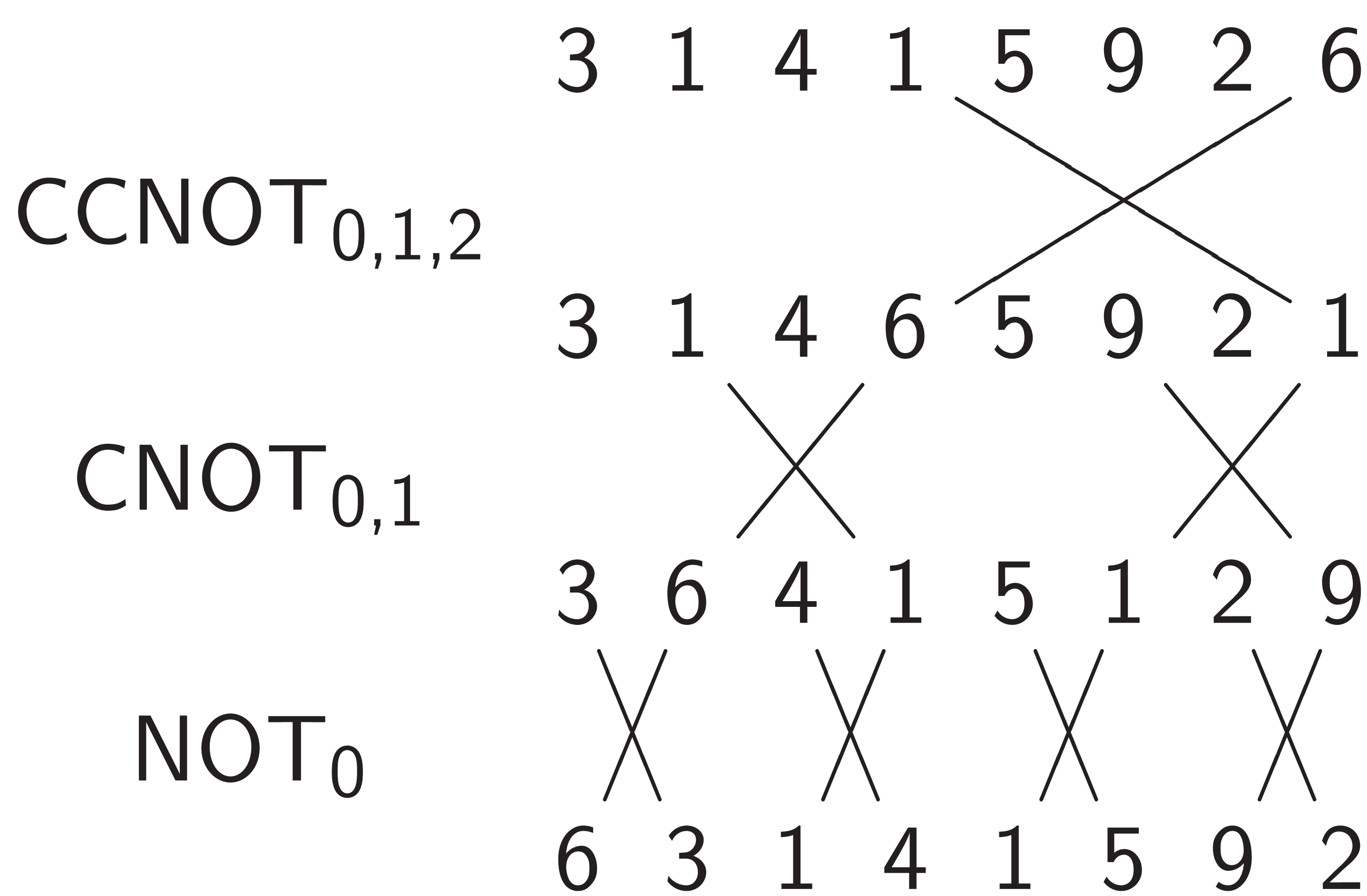
nt:

$q_1 q_2$).

More shuffling

Combine NOT, CNOT, Toffoli to build other permutations.

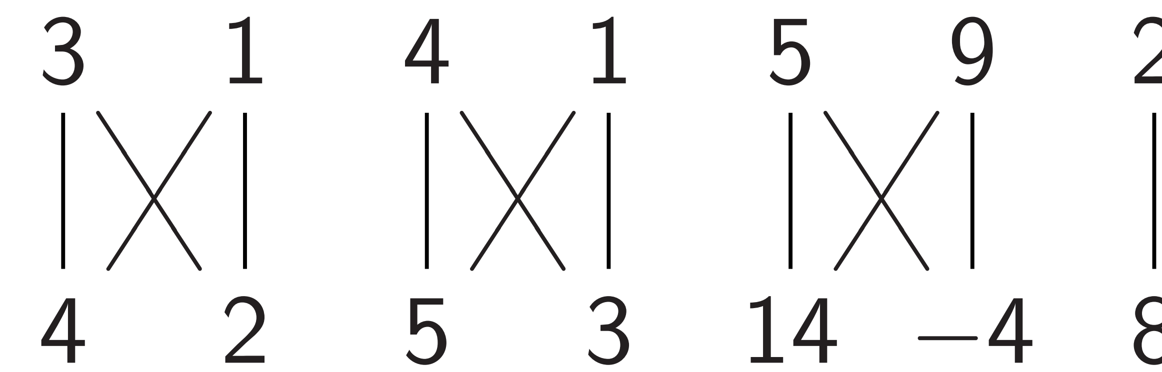
e.g. series of gates to rotate 8 positions by distance 1:



Hadamard gates

Hadamard$_0$:

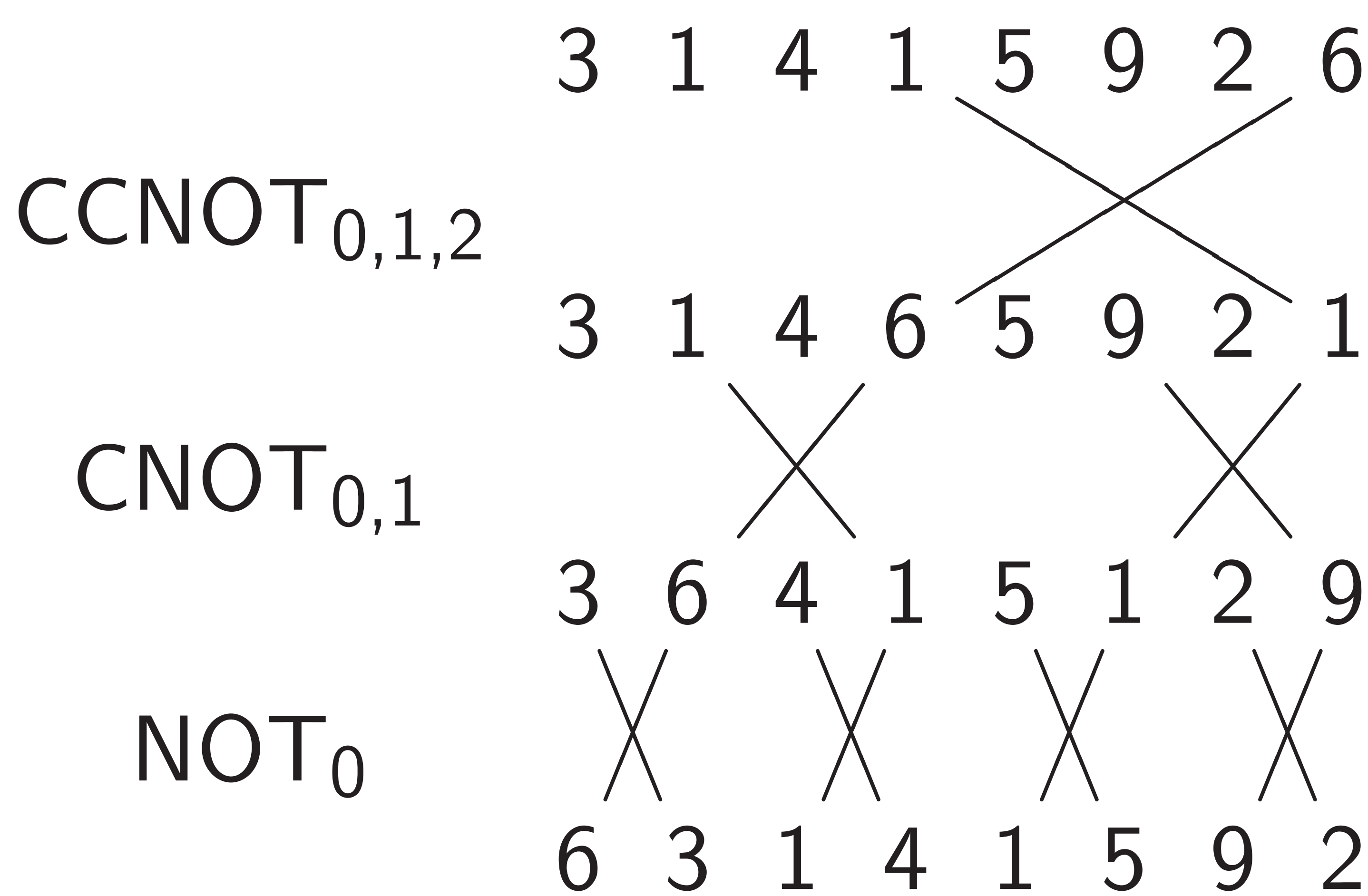
$(a, b) \mapsto (a + b, a - b)$.



More shuffling

Combine NOT, CNOT, Toffoli to build other permutations.

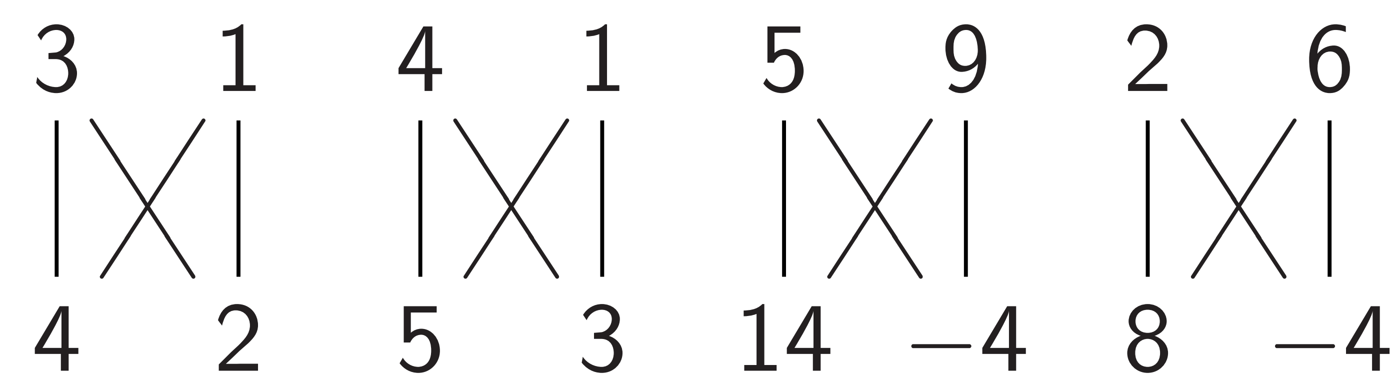
e.g. series of gates to rotate 8 positions by distance 1:



Hadamard gates

Hadamard_0:

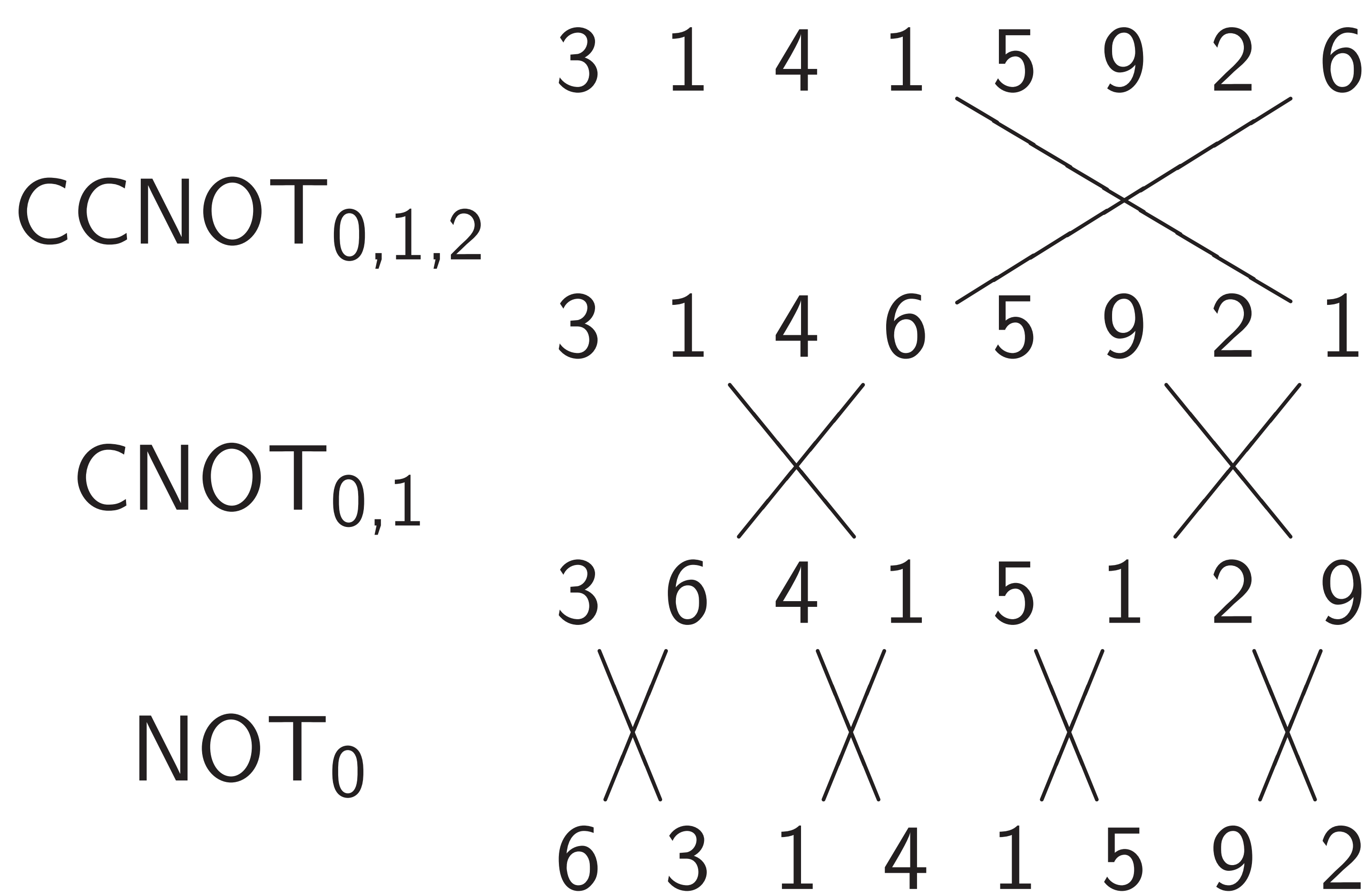
$(a, b) \mapsto (a + b, a - b)$.



More shuffling

Combine NOT, CNOT, Toffoli to build other permutations.

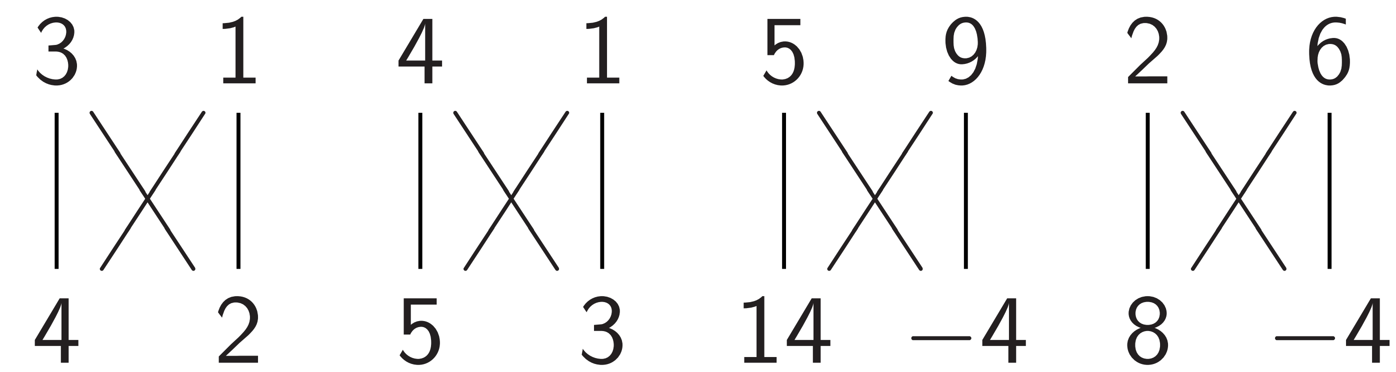
e.g. series of gates to rotate 8 positions by distance 1:



Hadamard gates

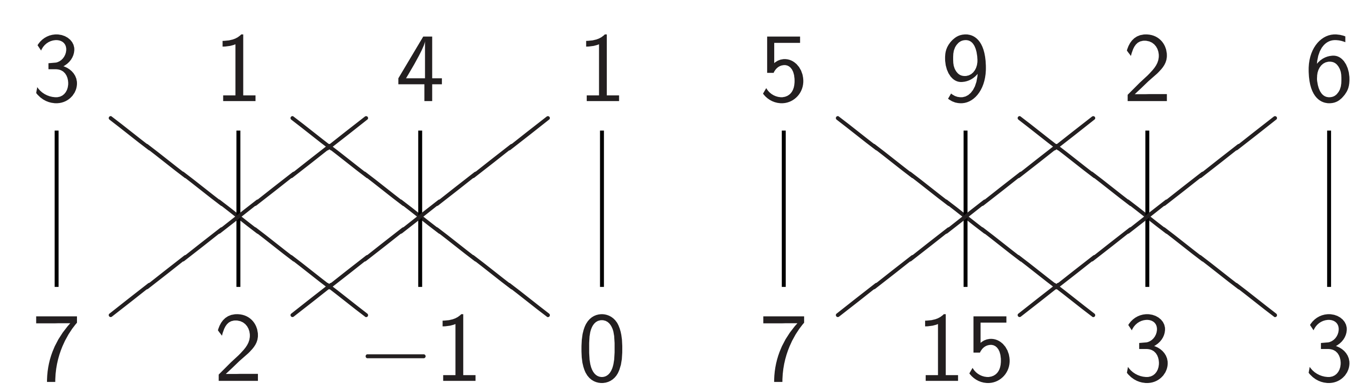
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

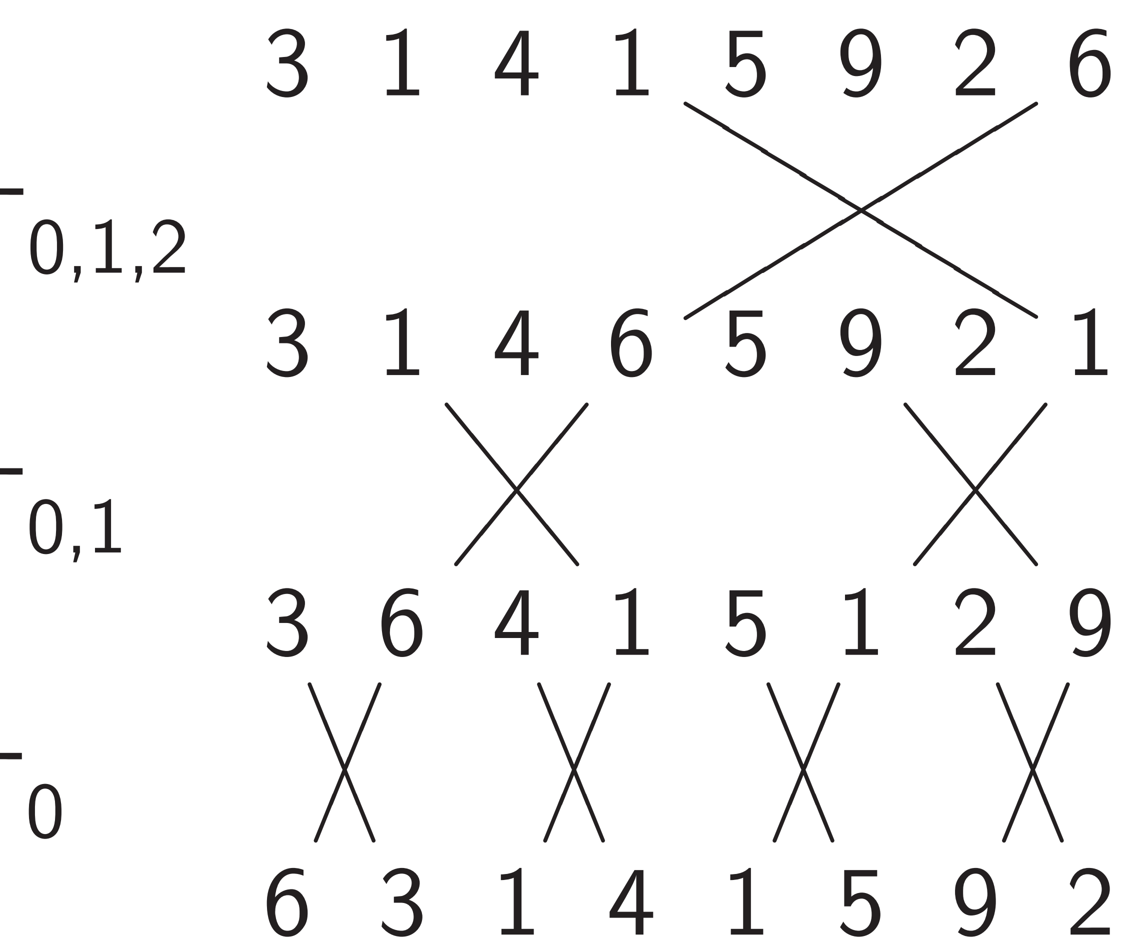
$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



uffling

e NOT, CNOT, Toffoli
other permutations.

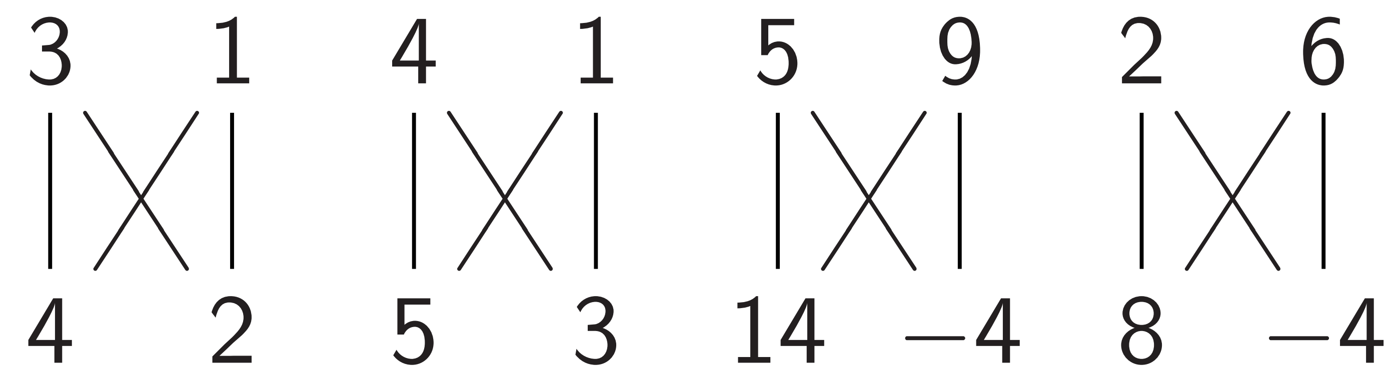
es of gates to
positions by distance 1:



Hadamard gates

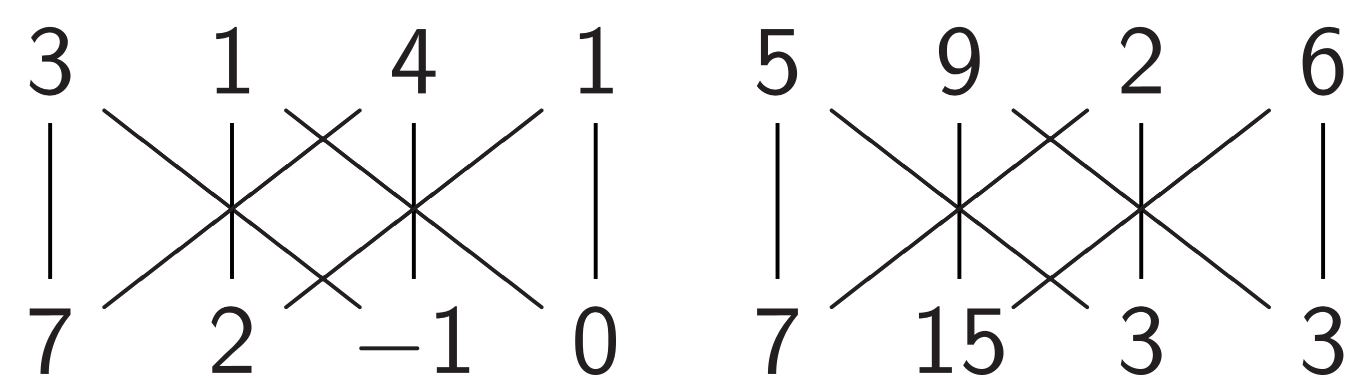
Hadamard_0:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard_1:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's

Step 1.

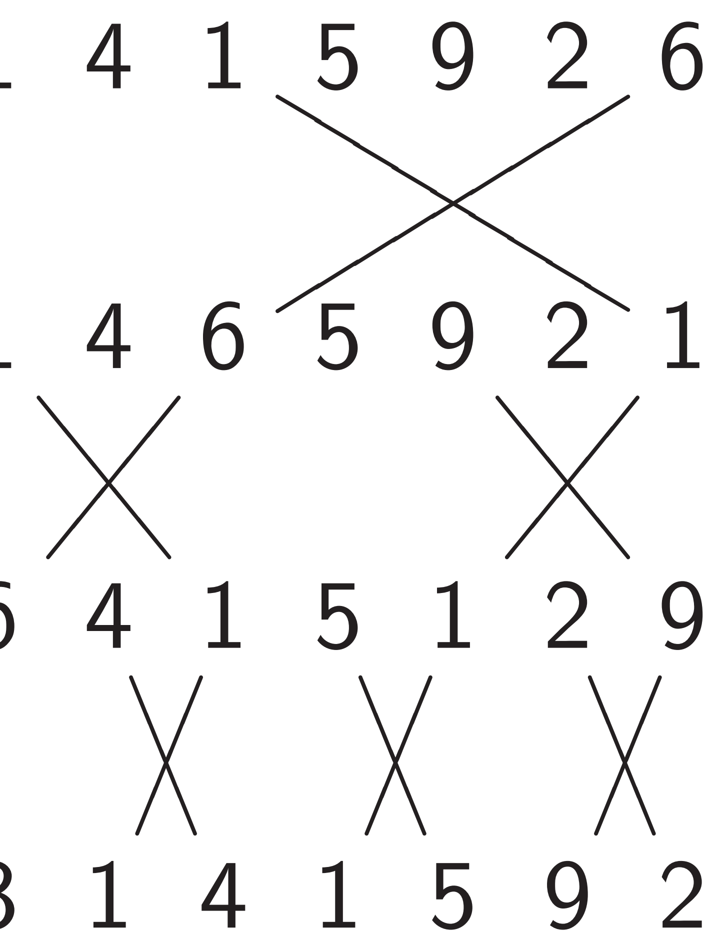
1, 0, 0, 0
0, 0, 0, 0
0, 0, 0, 0
0, 0, 0, 0
0, 0, 0, 0
0, 0, 0, 0
0, 0, 0, 0
0, 0, 0, 0

NOT, Toffoli

mutations.

s to

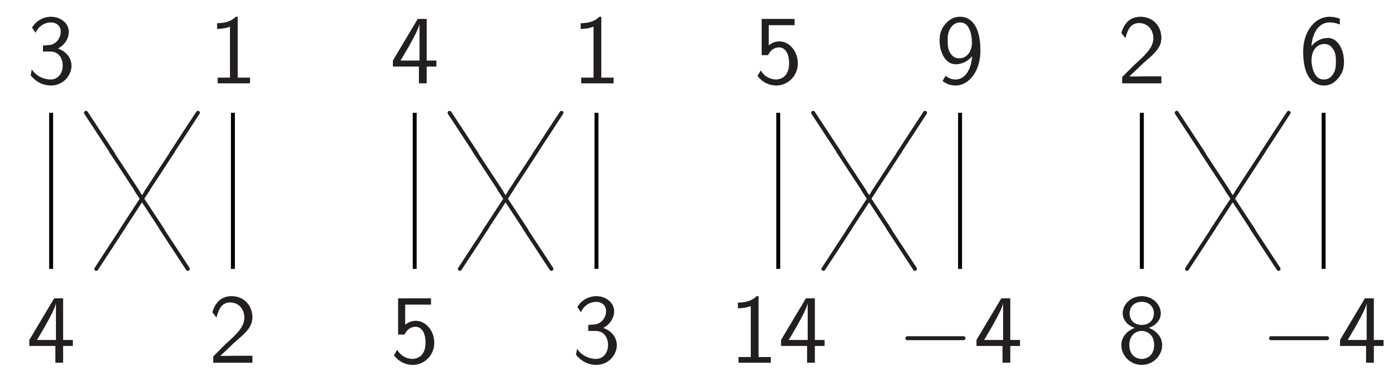
by distance 1:



Hadamard gates

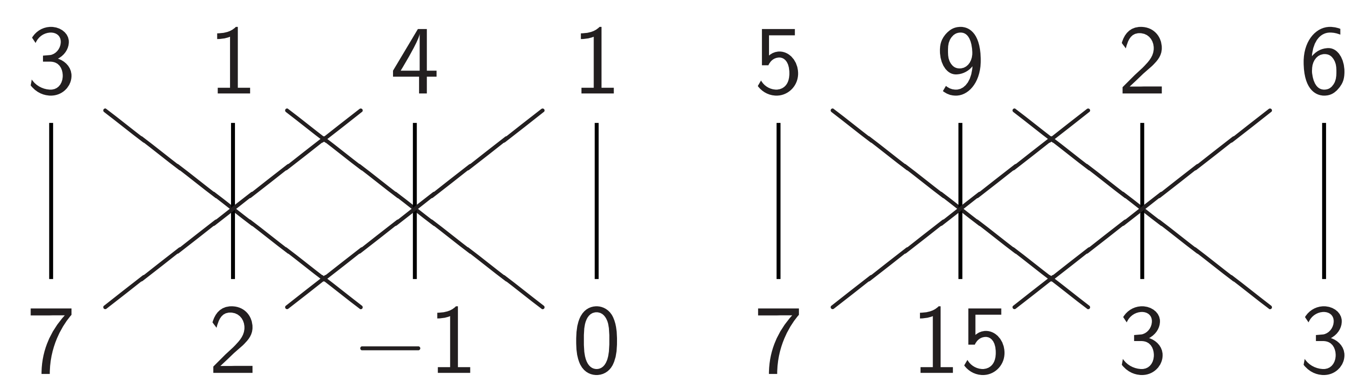
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



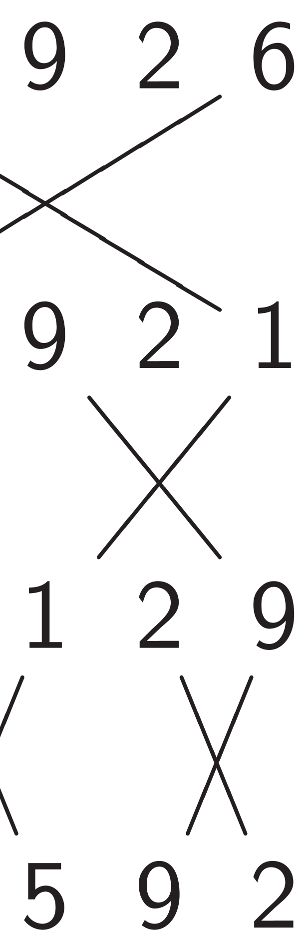
Simon's algorithm

Step 1. Set up pu

1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.

foli

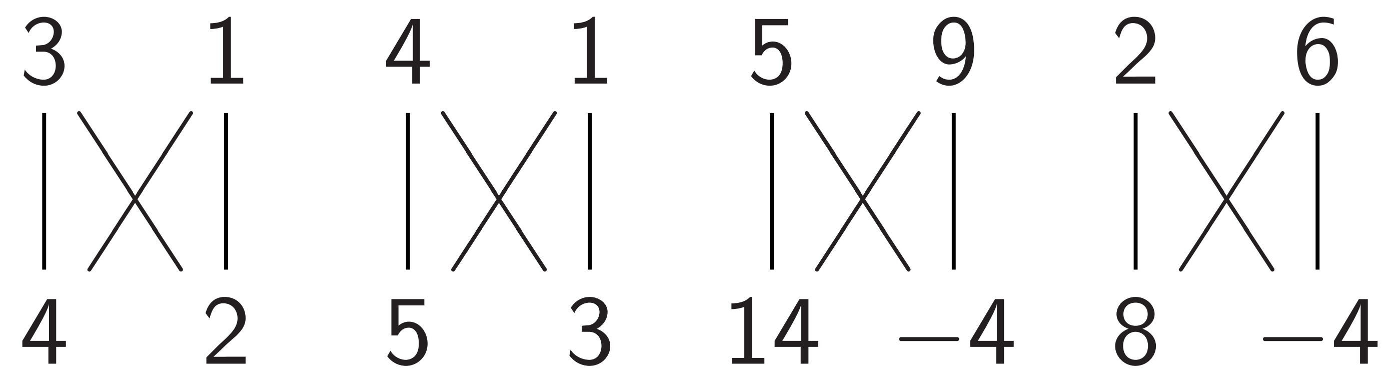
ce 1:



Hadamard gates

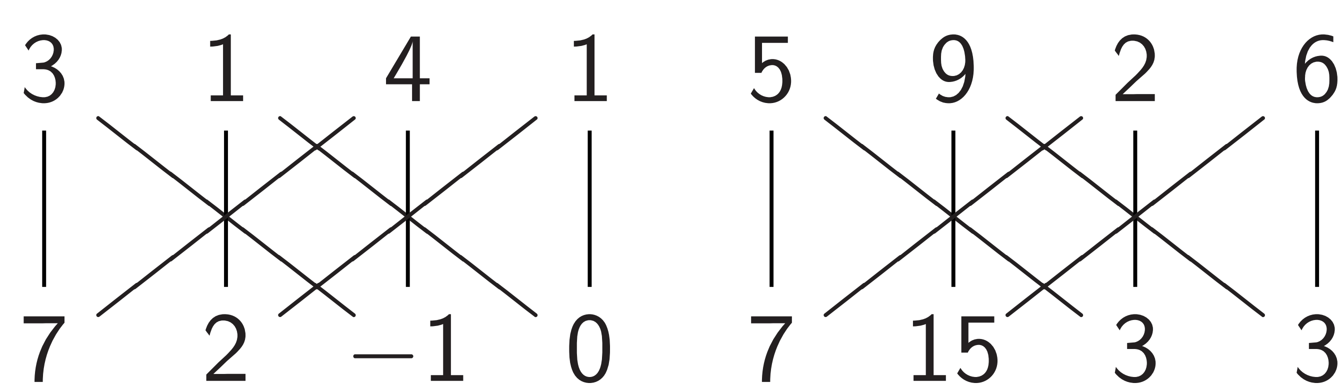
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

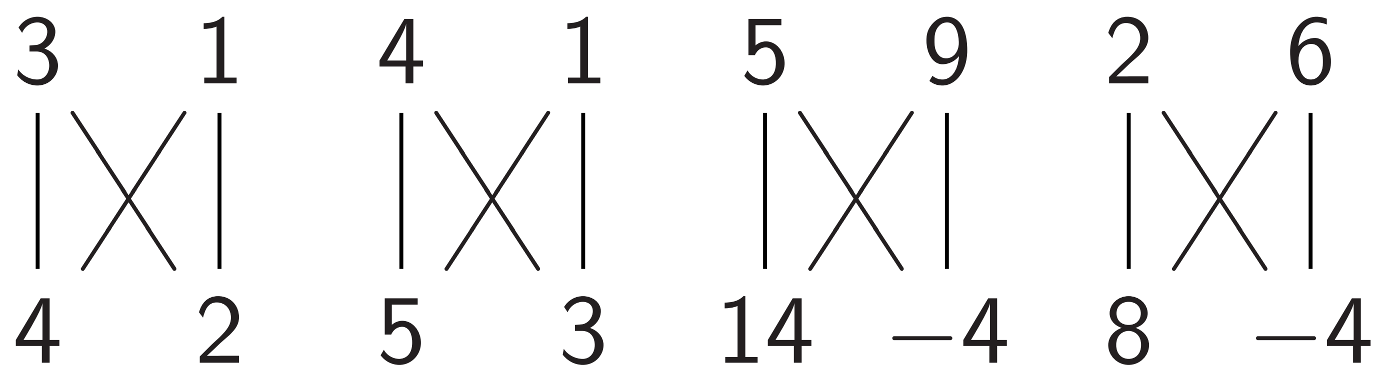
Step 1. Set up pure zero sta

1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.

Hadamard gates

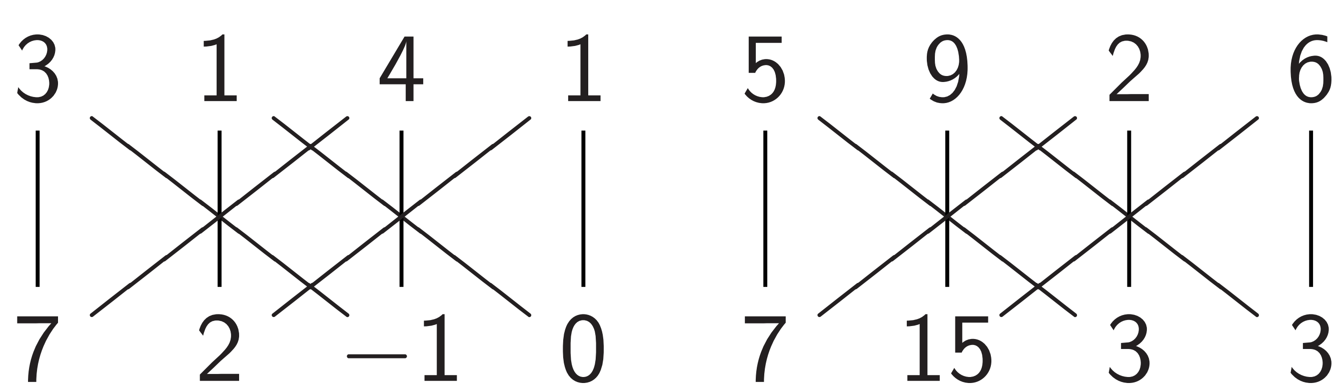
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

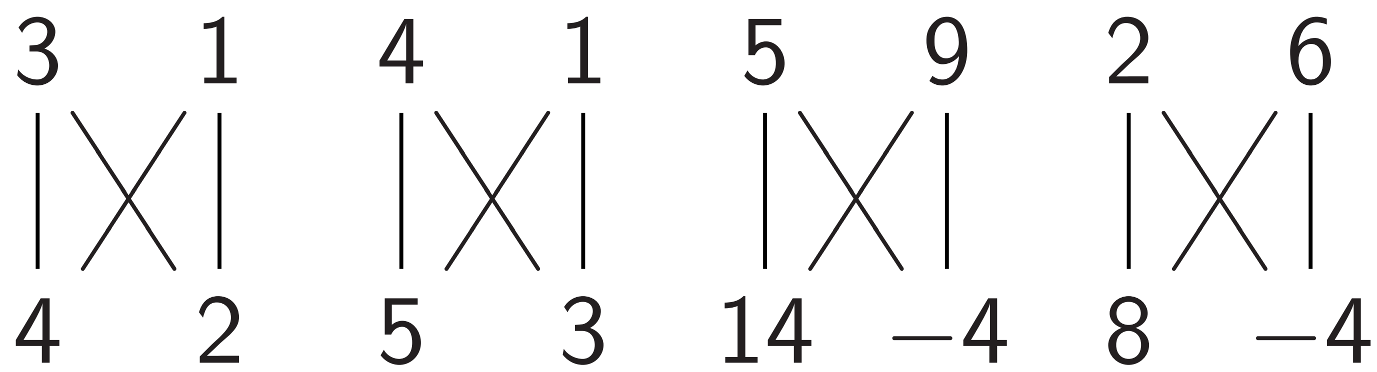
Step 1. Set up pure zero state:

1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.

Hadamard gates

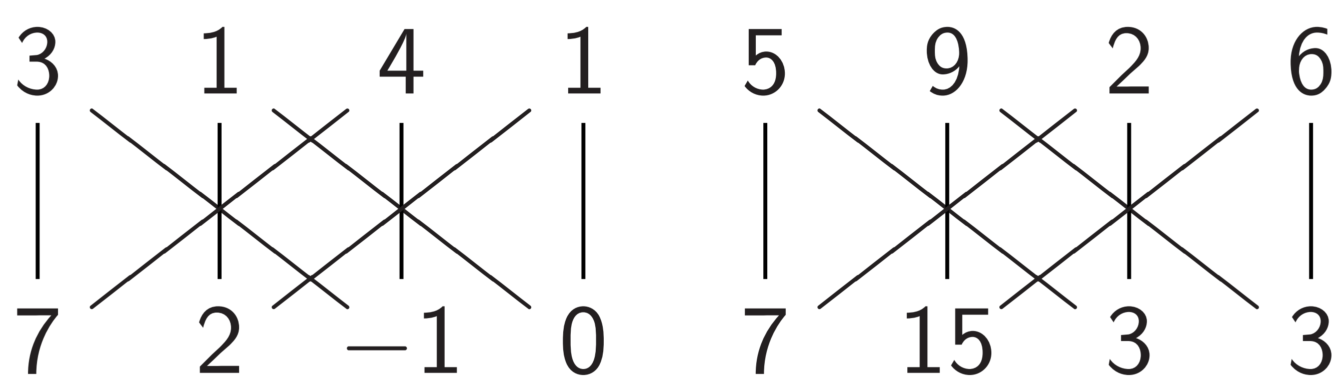
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

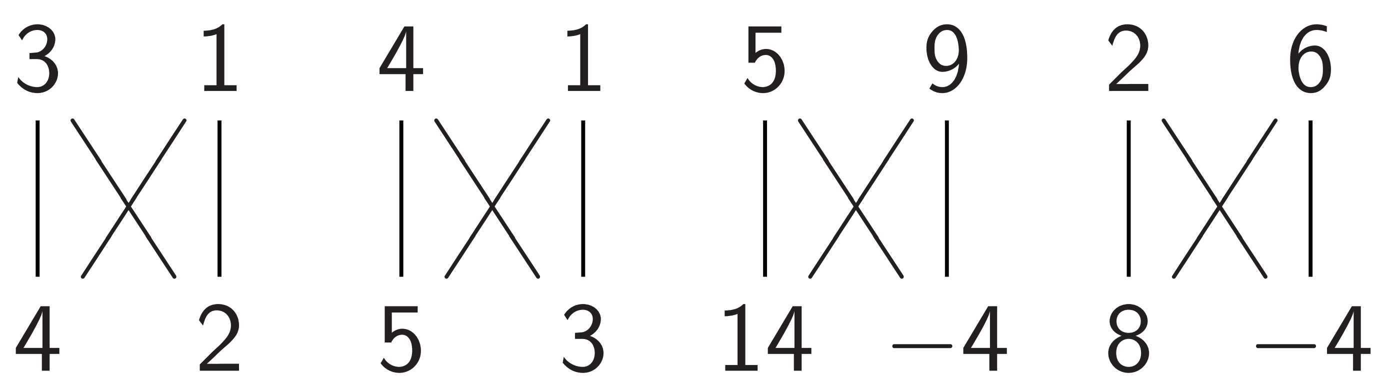
Step 2. Hadamard$_0$:

1, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.

Hadamard gates

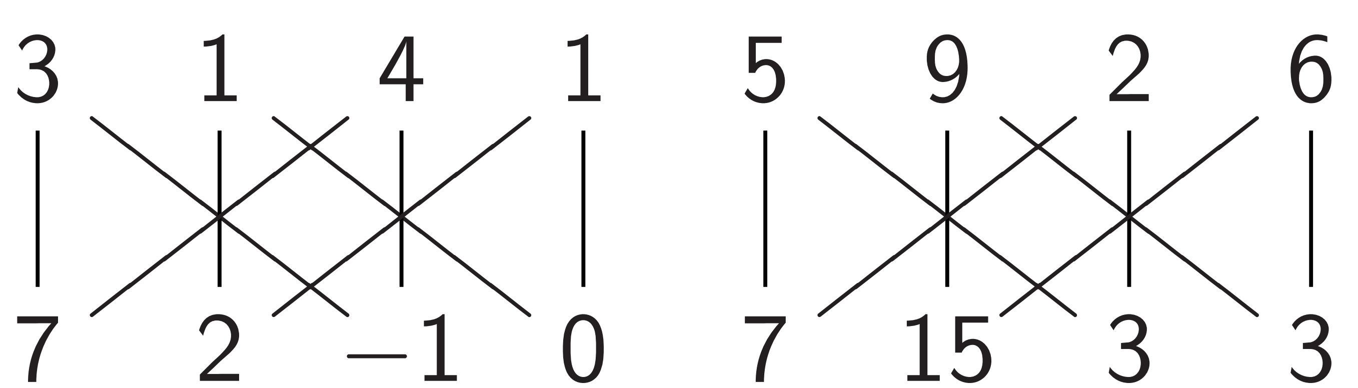
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

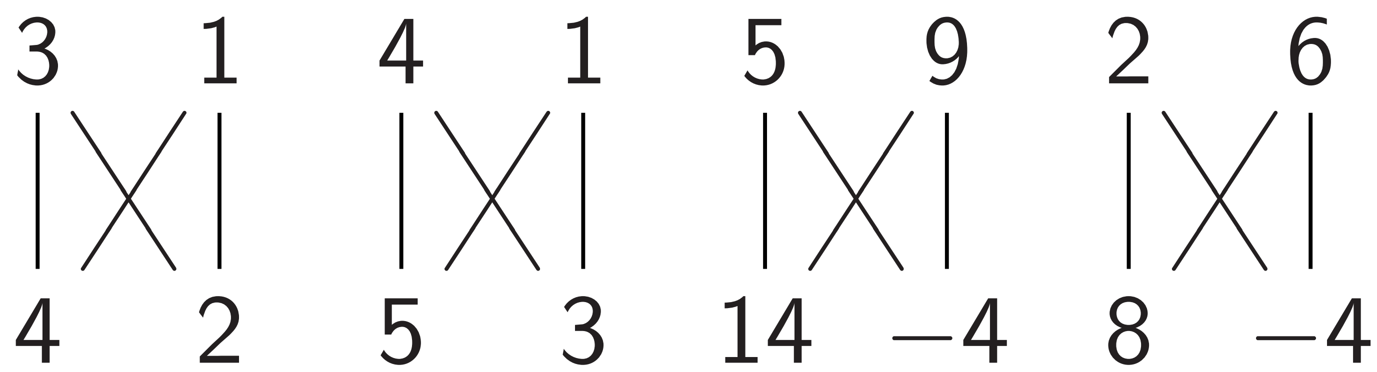
Step 3. Hadamard$_1$:

1, 1, 1, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.

Hadamard gates

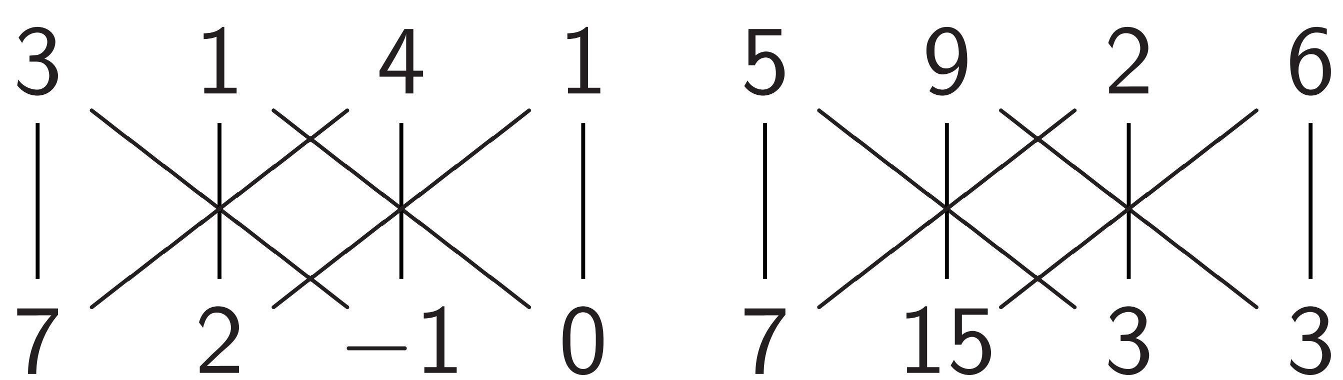
Hadamard_0:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard_1:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 4. Hadamard_2:

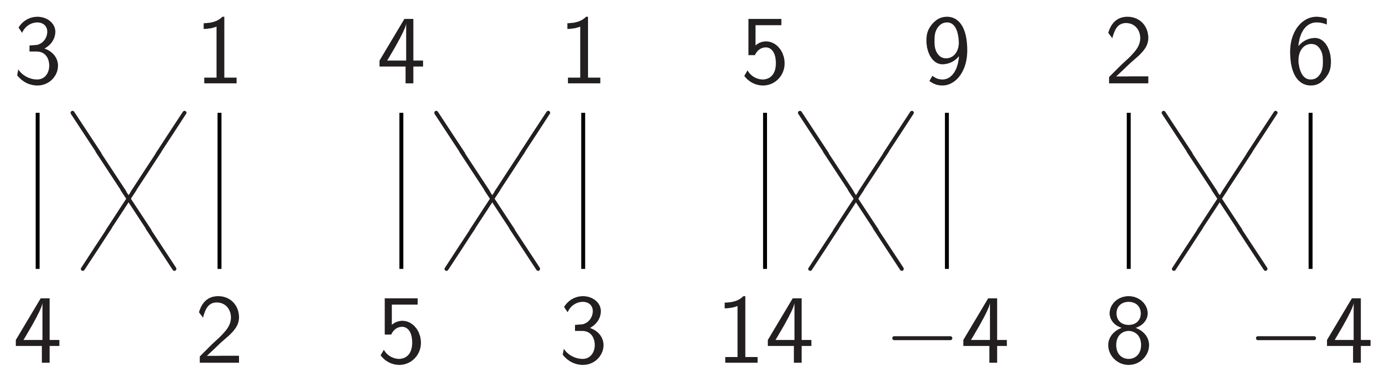
1, 1, 1, 1, 1, 1, 1, 1,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.

Each column is a parallel universe.

Hadamard gates

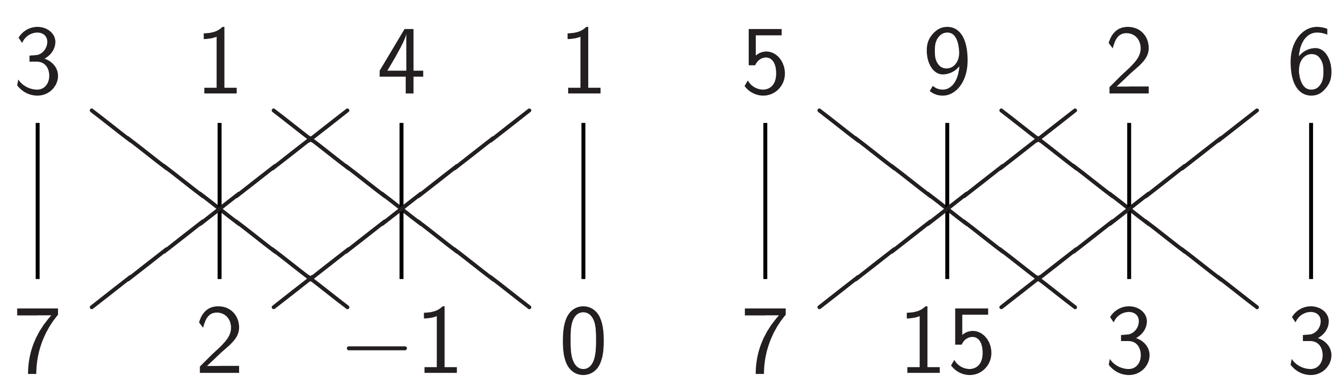
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5. CNOT$_{0,3}$:

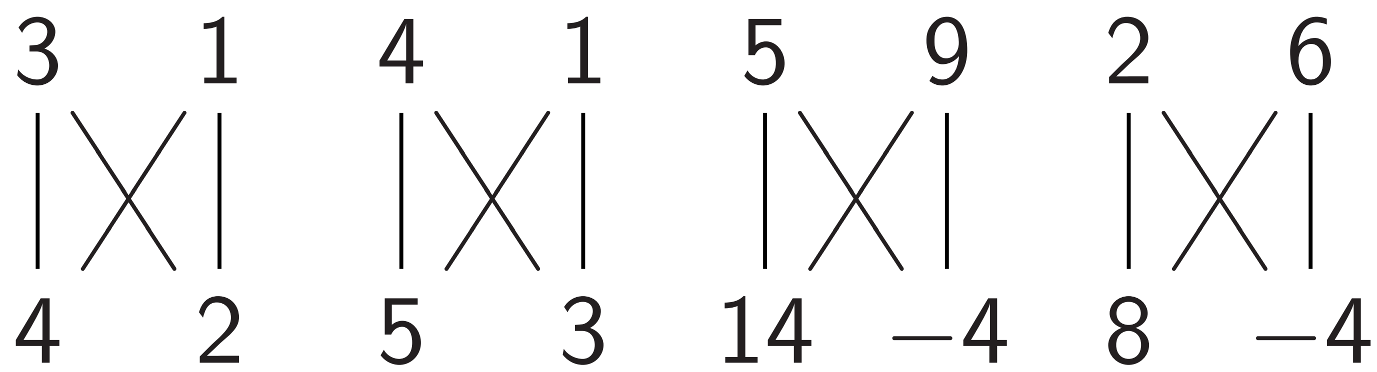
1, 0, 1, 0, 1, 0, 1, 0,
0, 1, 0, 1, 0, 1, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.

Each column is a parallel universe performing its own computations.

Hadamard gates

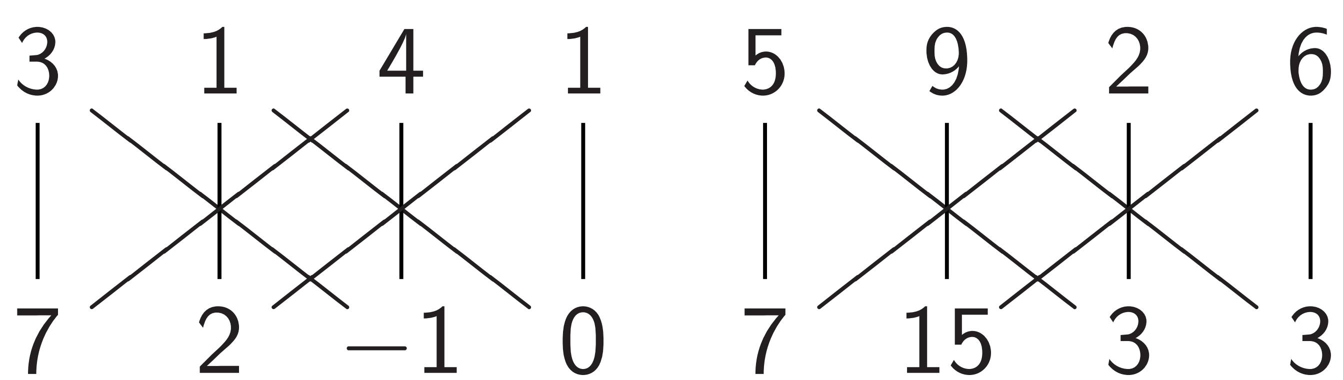
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5b. More shuffling:

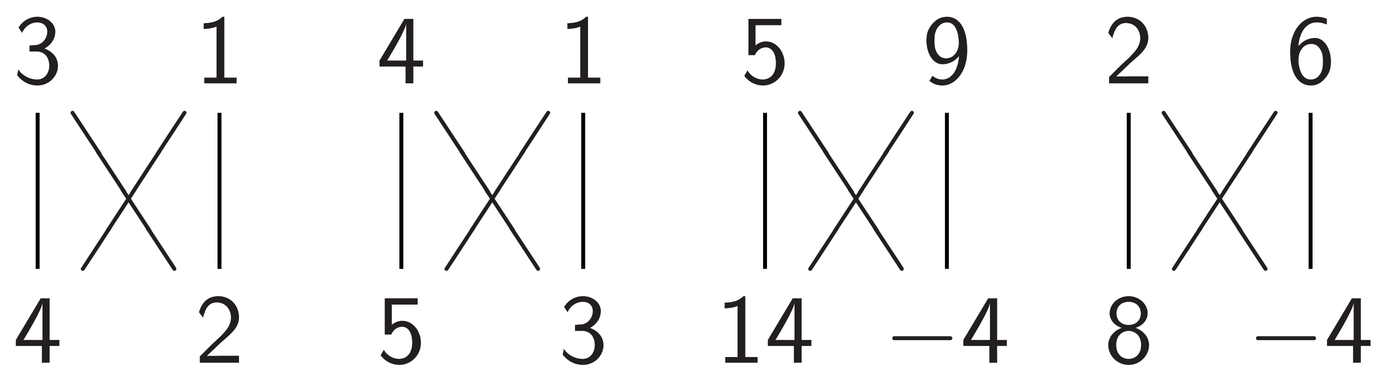
1, 0, 0, 0, 1, 0, 0, 0,
0, 1, 0, 0, 0, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.

Each column is a parallel universe performing its own computations.

Hadamard gates

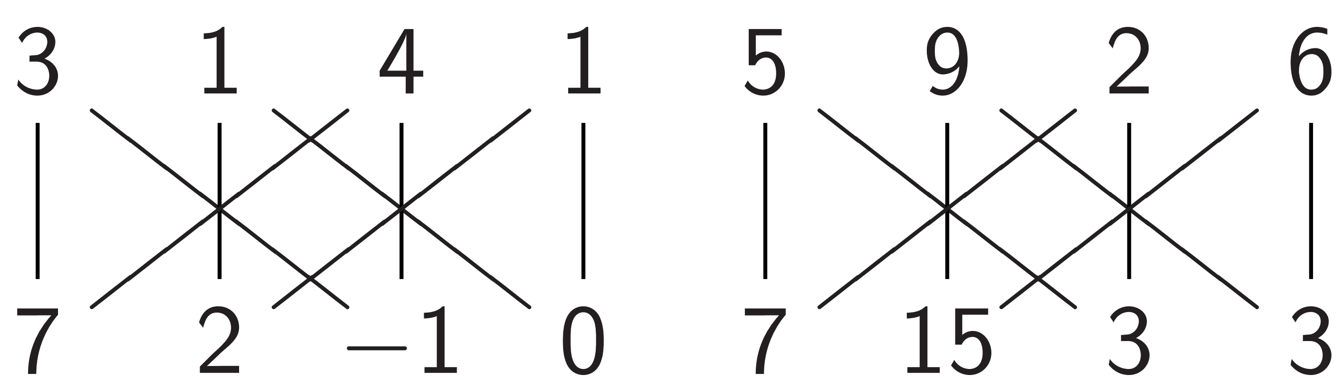
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5c. More shuffling:

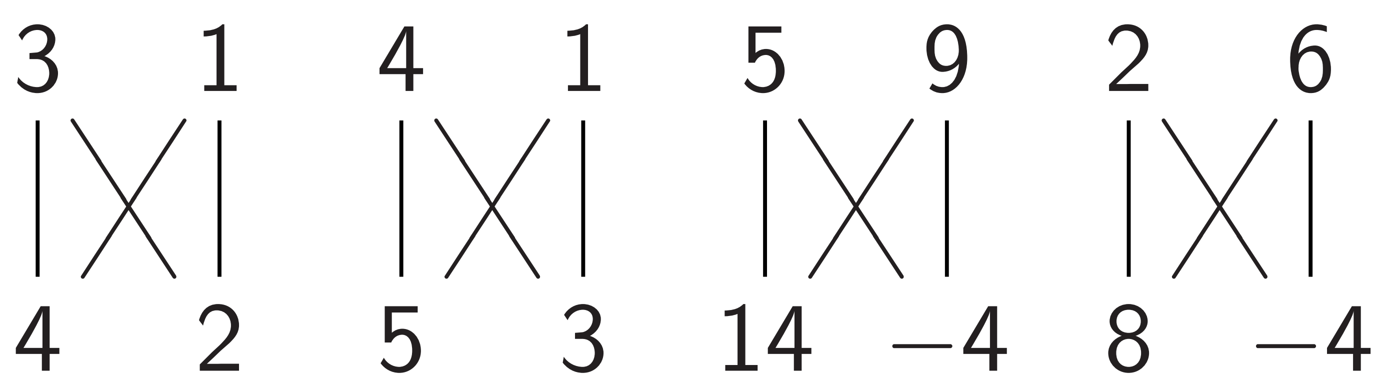
1, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 1.

Each column is a parallel universe performing its own computations.

Hadamard gates

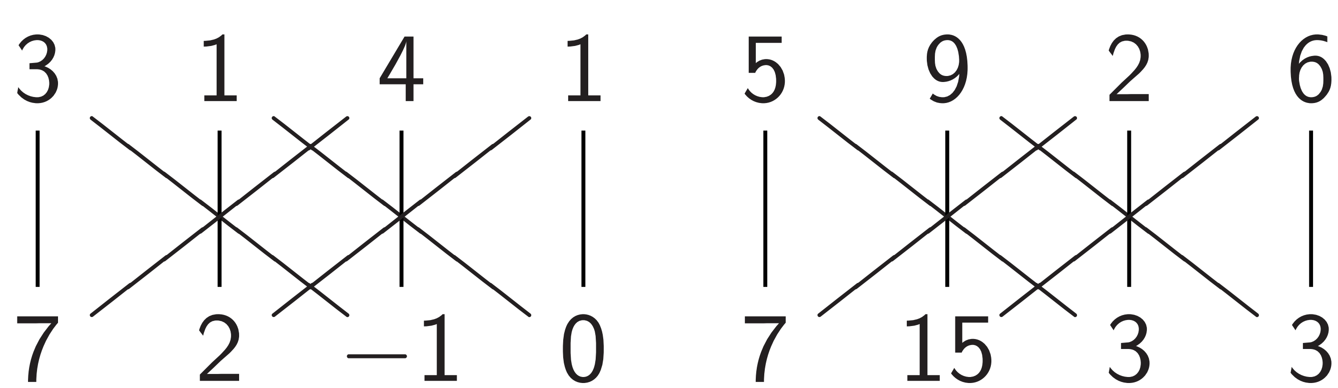
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5d. More shuffling:

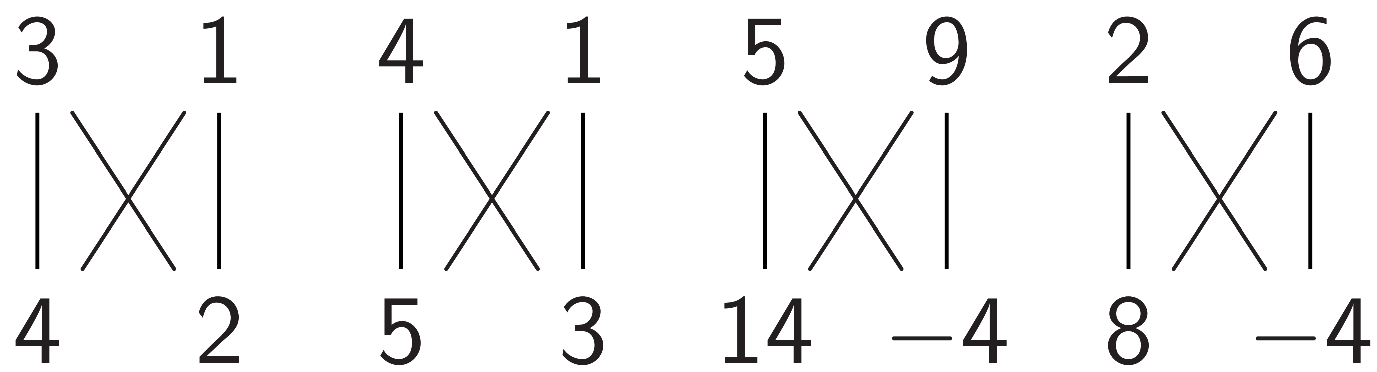
1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 0.

Each column is a parallel universe performing its own computations.

Hadamard gates

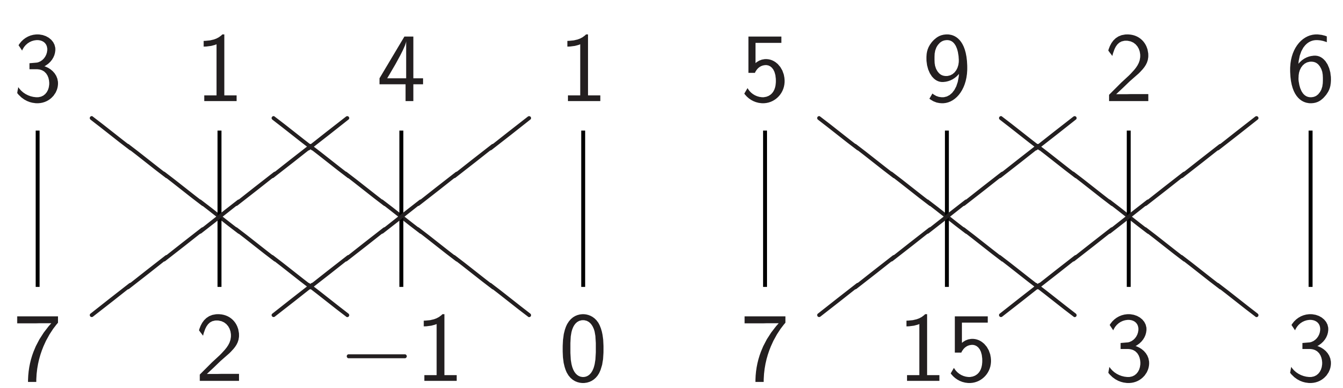
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5e. More shuffling:

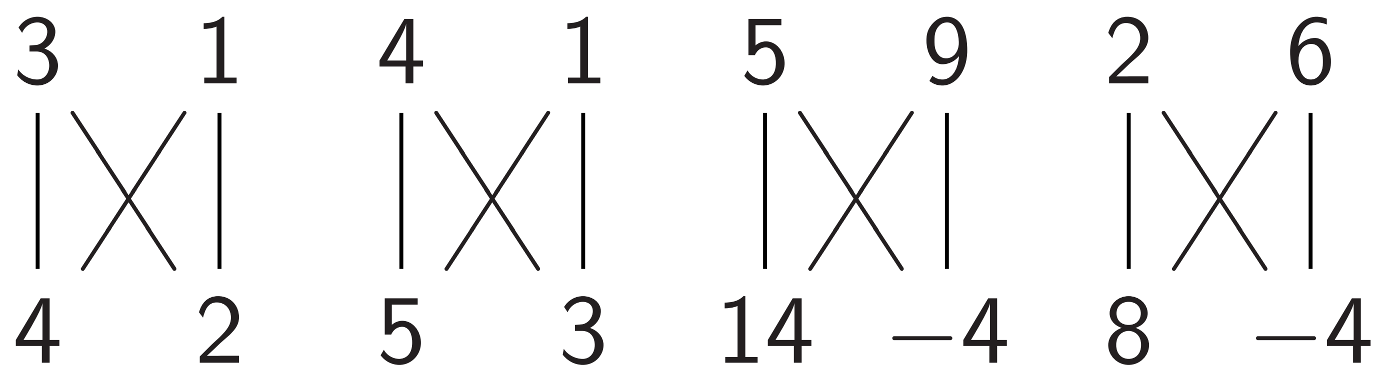
1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0.

Each column is a parallel universe performing its own computations.

Hadamard gates

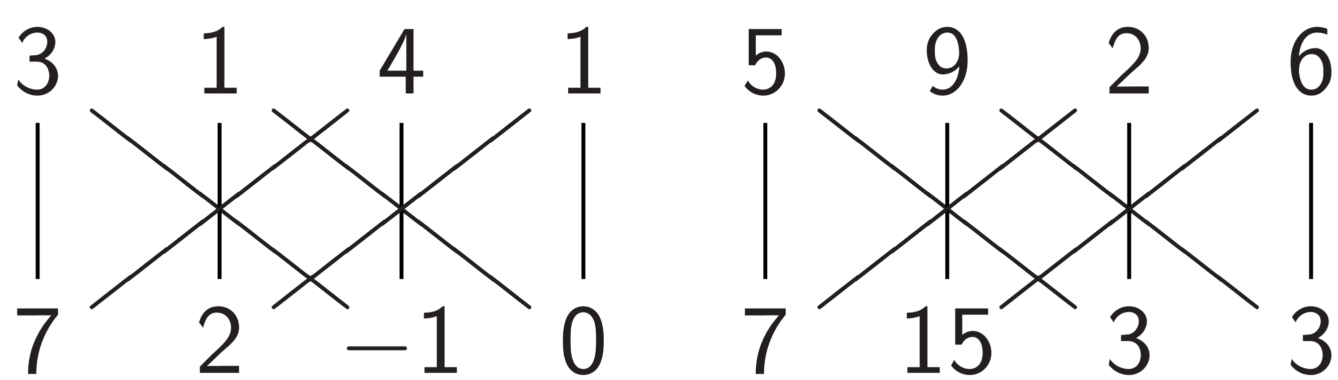
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5f. More shuffling:

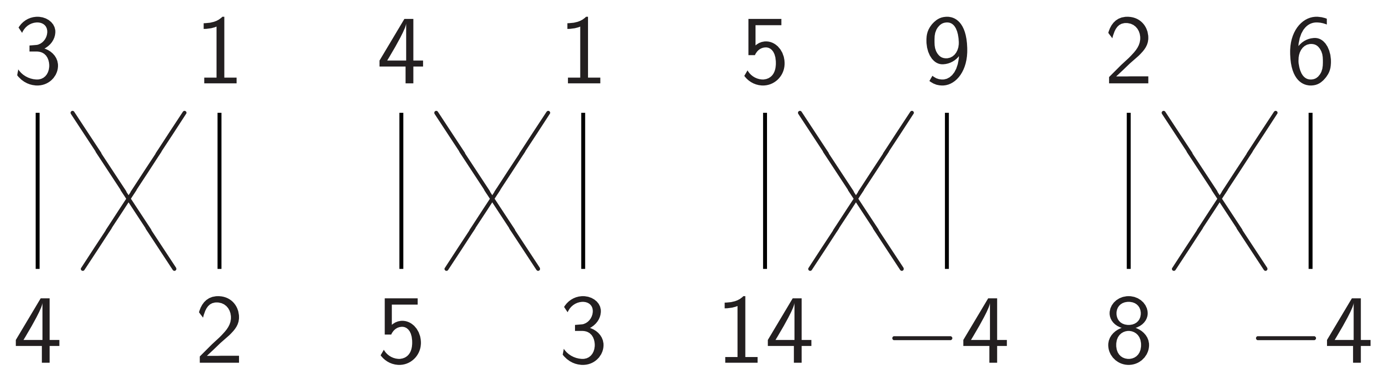
0, 0, 0, 0, 0, 1, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 1, 0.

Each column is a parallel universe performing its own computations.

Hadamard gates

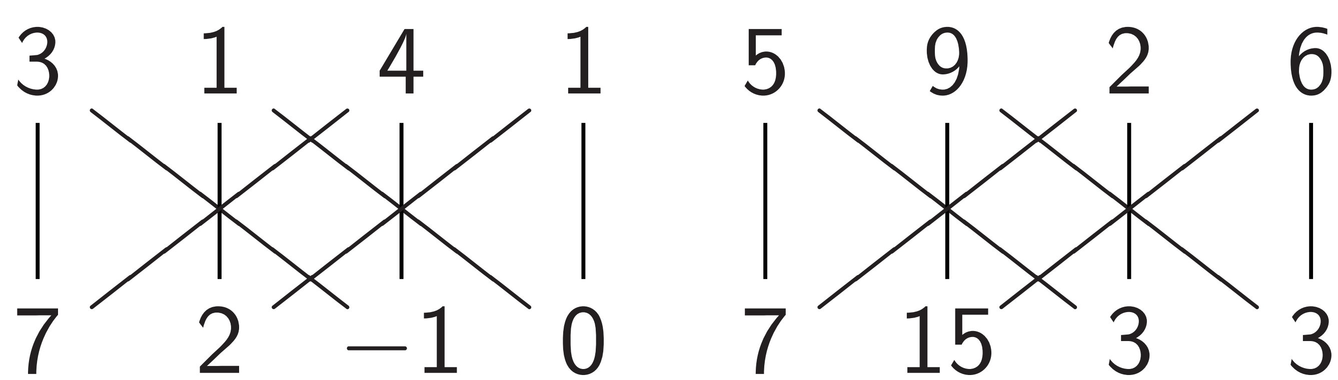
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5g. More shuffling:

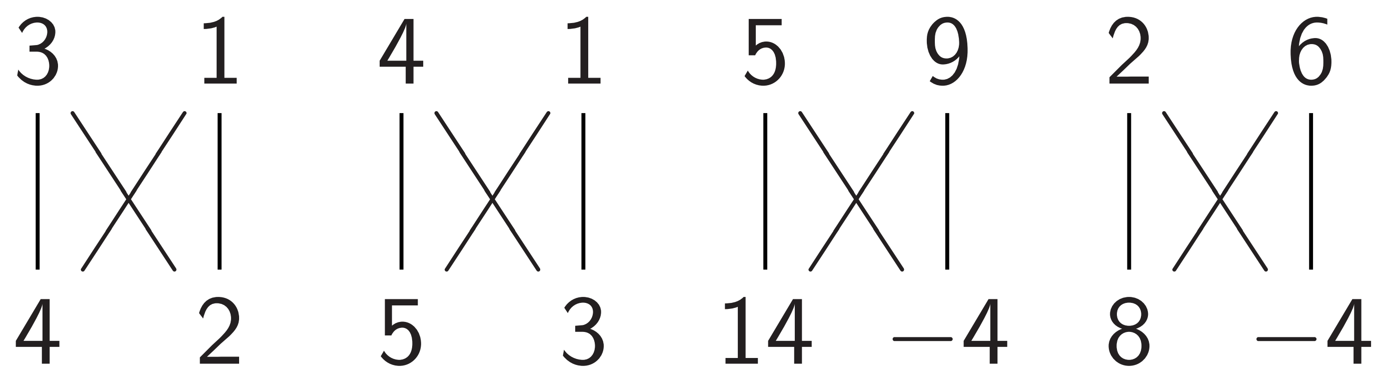
0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 1.

Each column is a parallel universe performing its own computations.

Hadamard gates

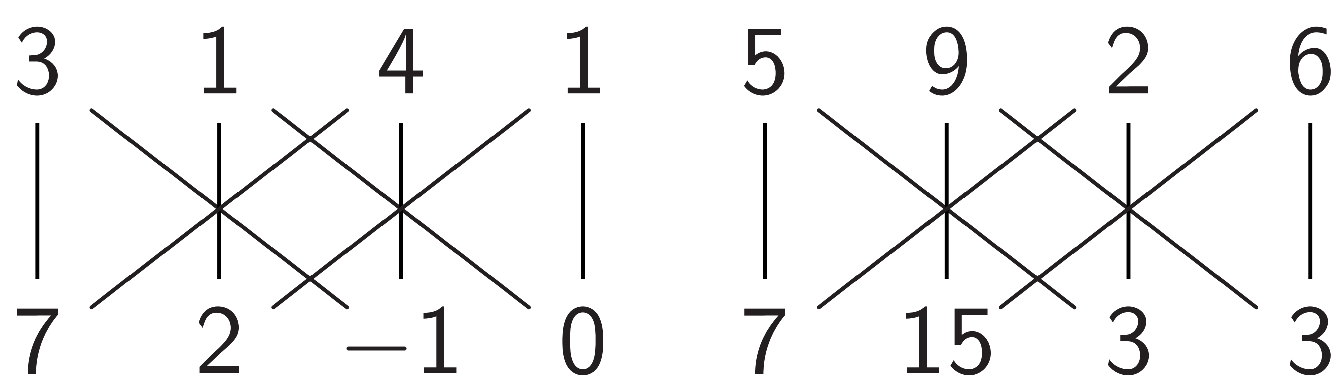
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5h. More shuffling:

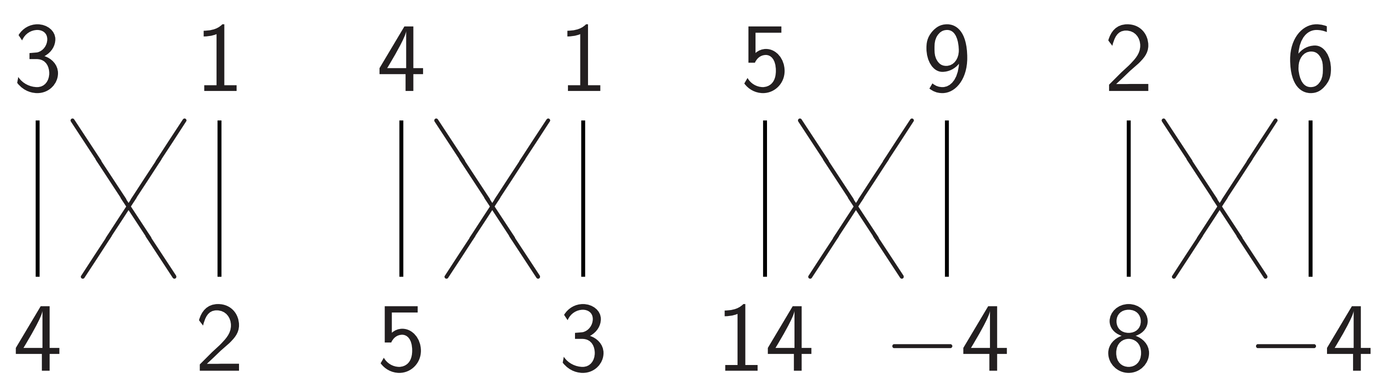
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0.

Each column is a parallel universe performing its own computations.

Hadamard gates

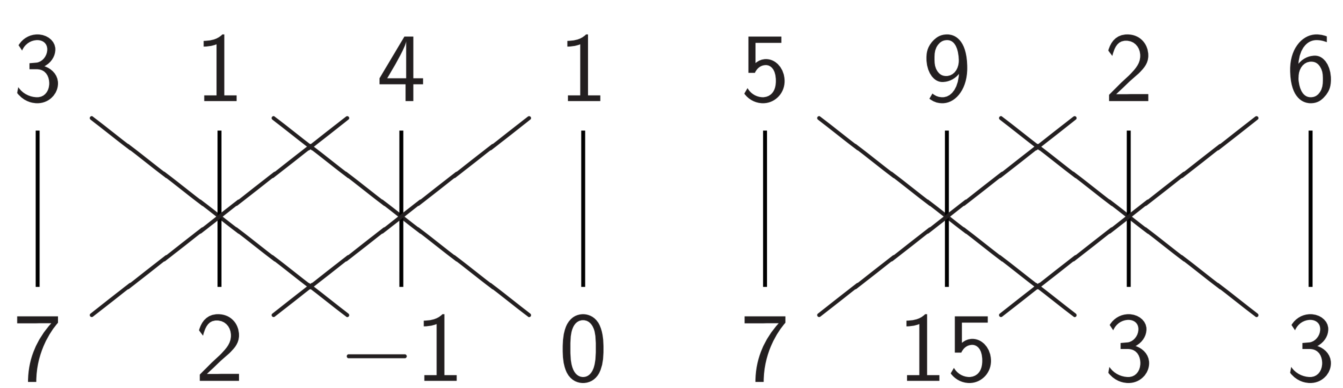
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5i. More shuffling:

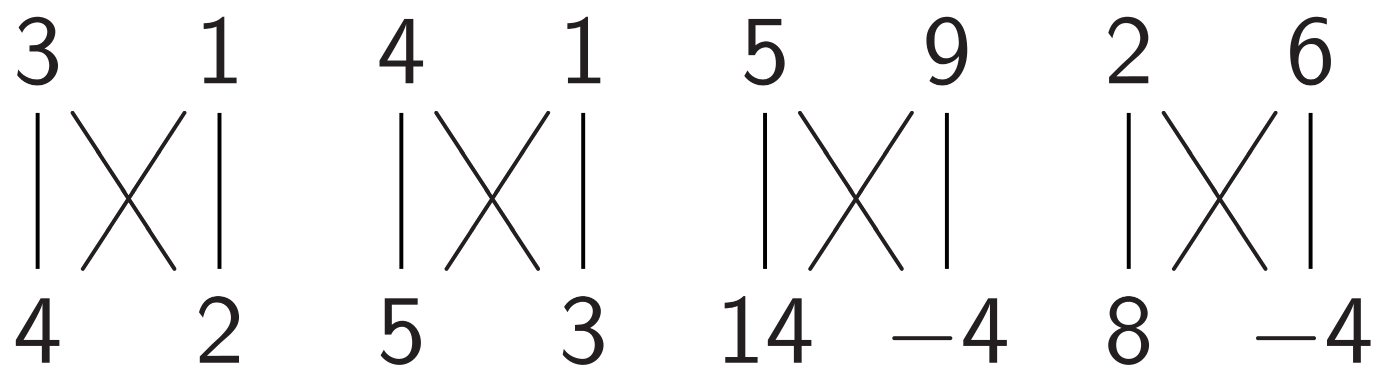
0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 1, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0.

Each column is a parallel universe performing its own computations.

Hadamard gates

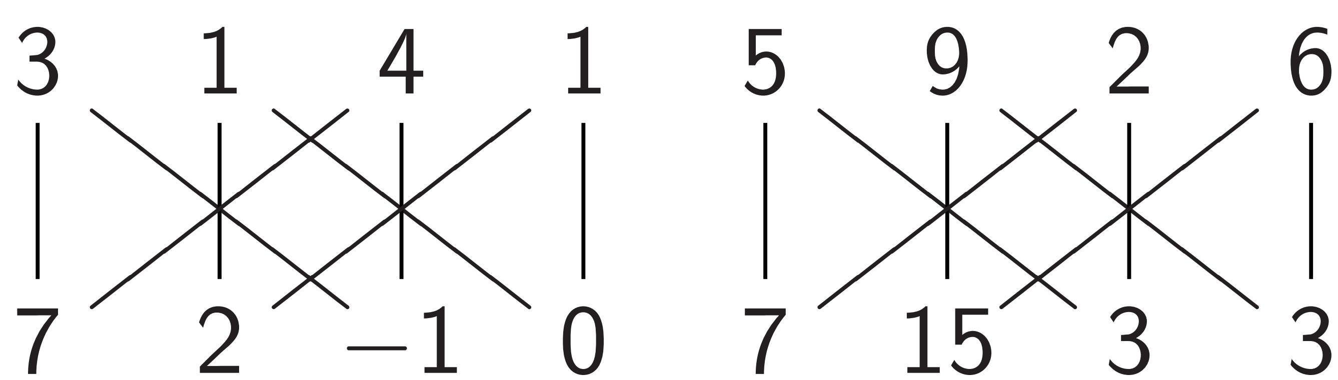
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5j. Final shuffling:

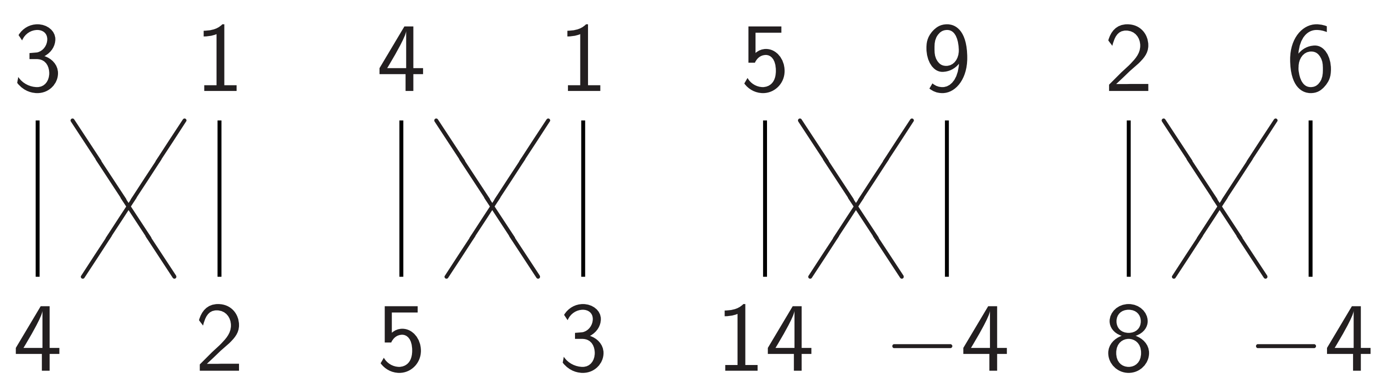
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 1,
0, 1, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 1, 0, 0.

Each column is a parallel universe performing its own computations.

Hadamard gates

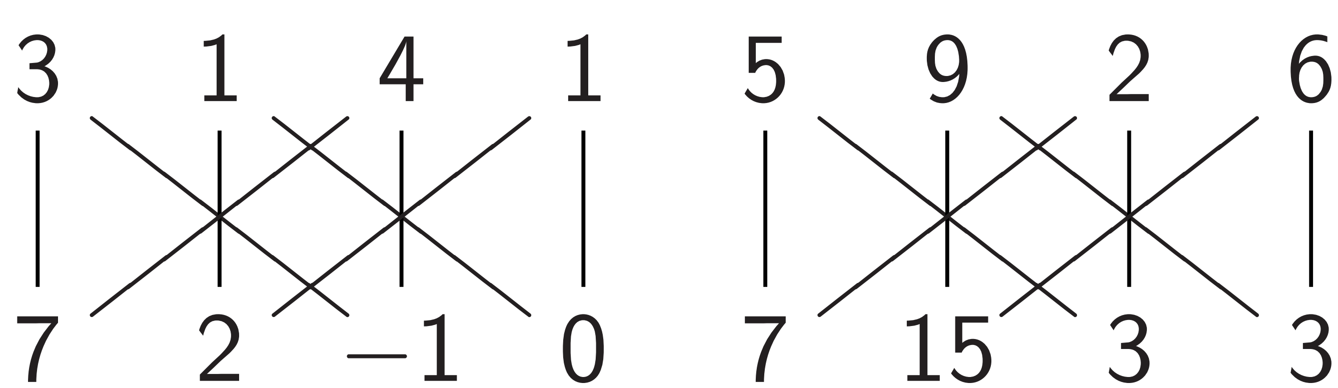
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 5j. Final shuffling:

0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 1,
0, 1, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 1, 0, 0.

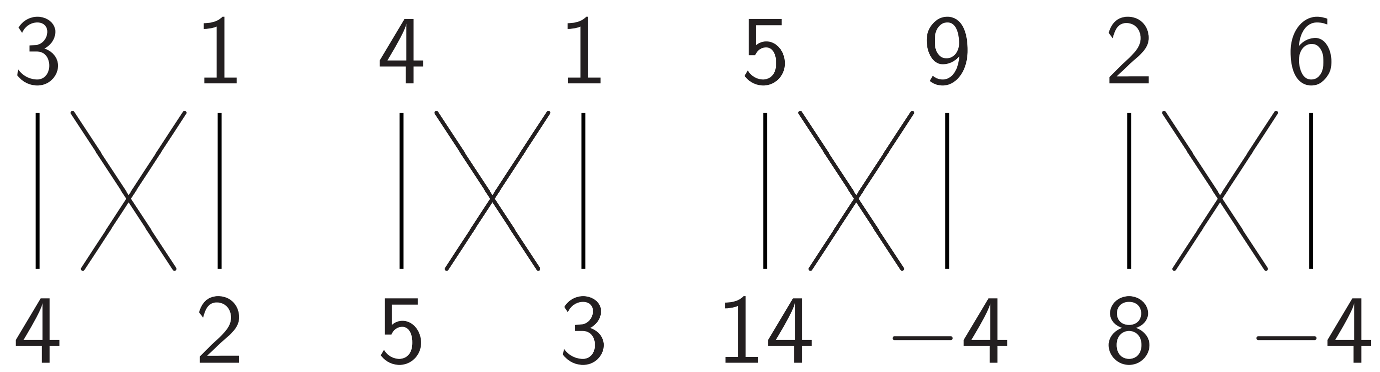
Each column is a parallel universe performing its own computations.

Surprise: u and $u \oplus 101$ match.

Hadamard gates

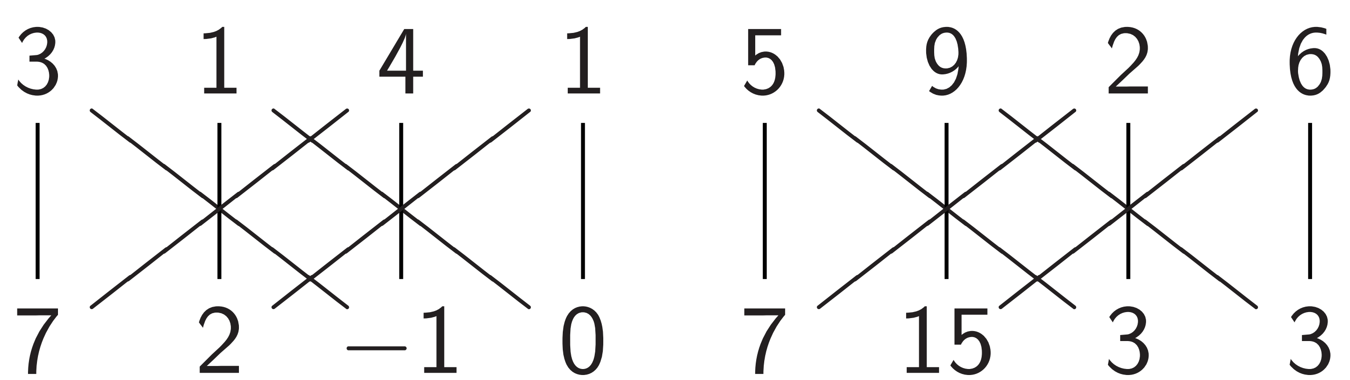
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 6. Hadamard$_0$:

$0, 0, 0, 0, 0, 0, 0, 0,$

$0, 0, 1, \overline{1}, 0, 0, 1, 1,$

$0, 0, 0, 0, 0, 0, 0, 0,$

$0, 0, 1, 1, 0, 0, 1, \overline{1},$

$1, \overline{1}, 0, 0, 1, 1, 0, 0,$

$0, 0, 0, 0, 0, 0, 0, 0,$

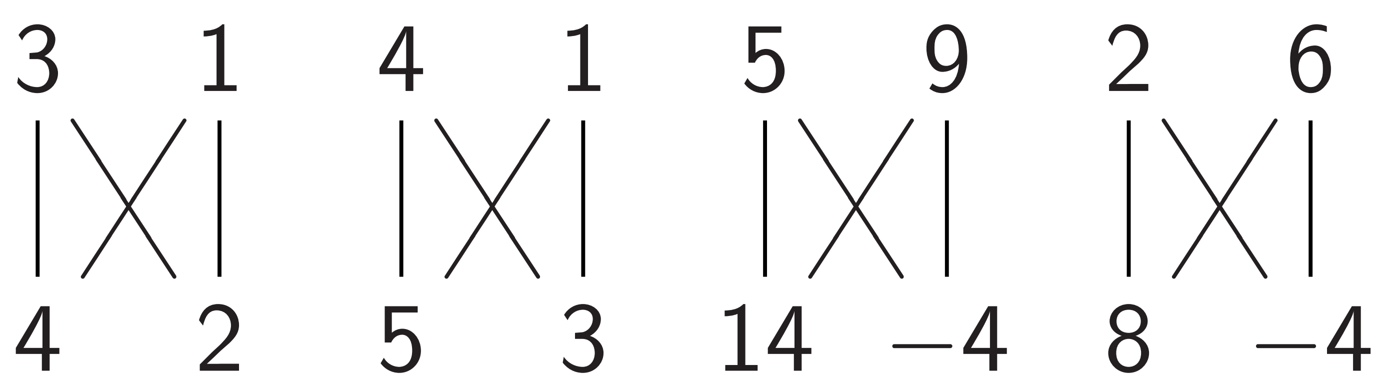
$0, 0, 0, 0, 0, 0, 0, 0,$

$1, 1, 0, 0, 1, \overline{1}, 0, 0.$

Hadamard gates

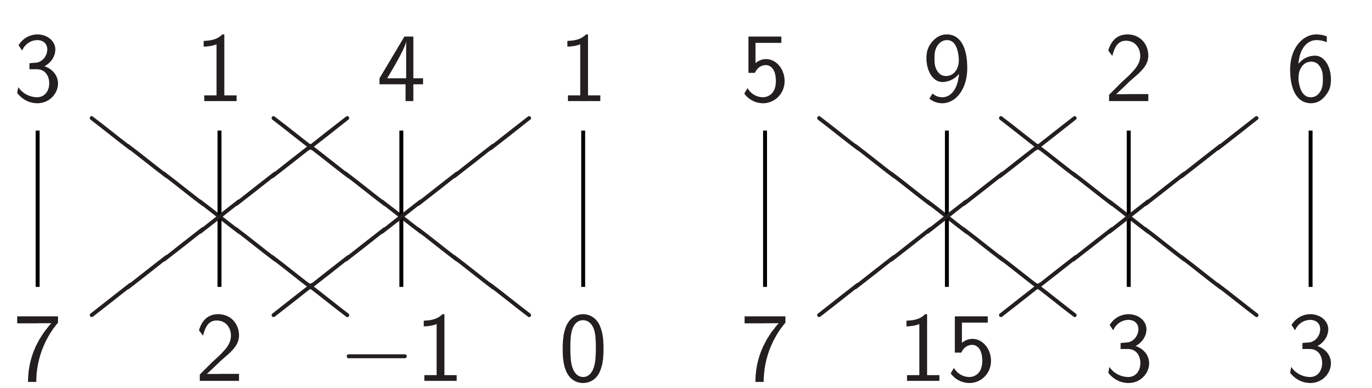
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

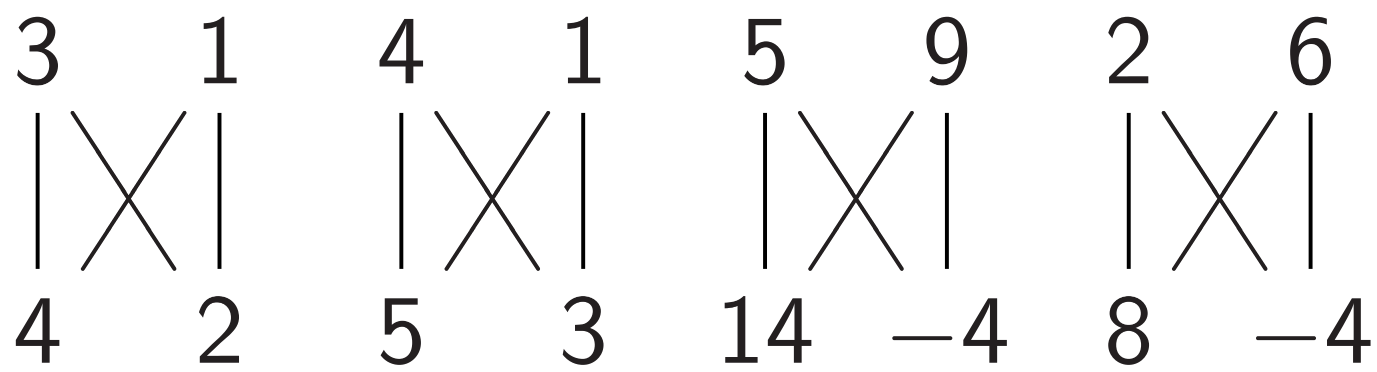
Step 7. Hadamard$_1$:

$0, 0, 0, 0, 0, 0, 0, 0,$
$1, \overline{1}, \overline{1}, 1, 1, 1, \overline{1}, \overline{1},$
$0, 0, 0, 0, 0, 0, 0, 0,$
$1, 1, \overline{1}, \overline{1}, 1, \overline{1}, \overline{1}, 1,$
$1, \overline{1}, 1, \overline{1}, 1, 1, 1, 1,$
$0, 0, 0, 0, 0, 0, 0, 0,$
$0, 0, 0, 0, 0, 0, 0, 0,$
$1, 1, 1, 1, 1, \overline{1}, 1, \overline{1}.$

Hadamard gates

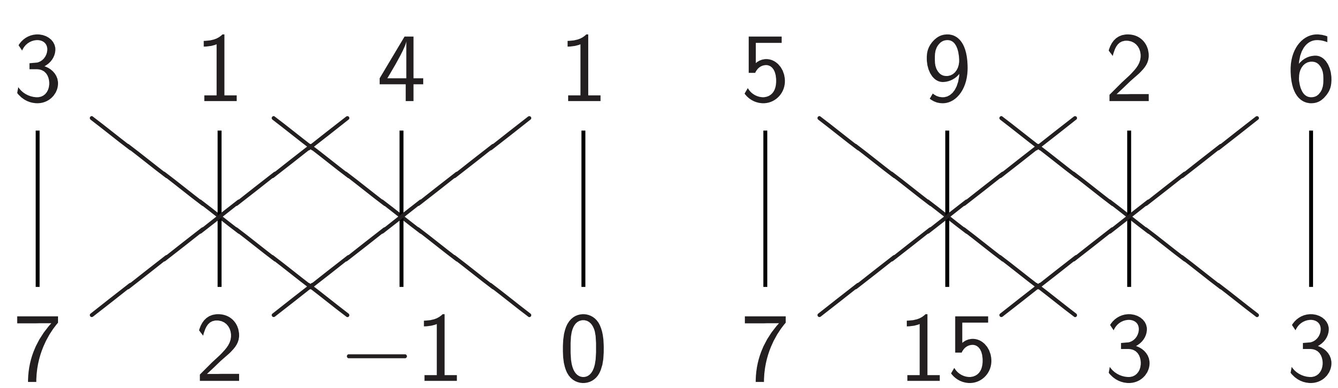
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

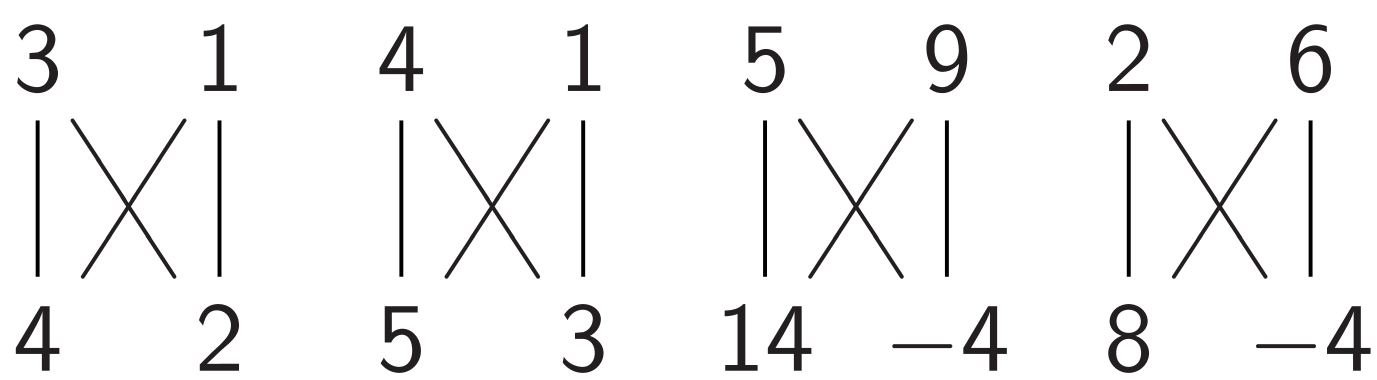
Step 8. Hadamard$_2$:

0, 0, 0, 0, 0, 0, 0, 0,
2, 0, $\overline{2}$, 0, 0, $\overline{2}$, 0, 2,
0, 0, 0, 0, 0, 0, 0, 0,
2, 0, $\overline{2}$, 0, 0, 2, 0, $\overline{2}$,
2, 0, 2, 0, 0, $\overline{2}$, 0, $\overline{2}$,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
2, 0, 2, 0, 0, 2, 0, 2.

Hadamard gates

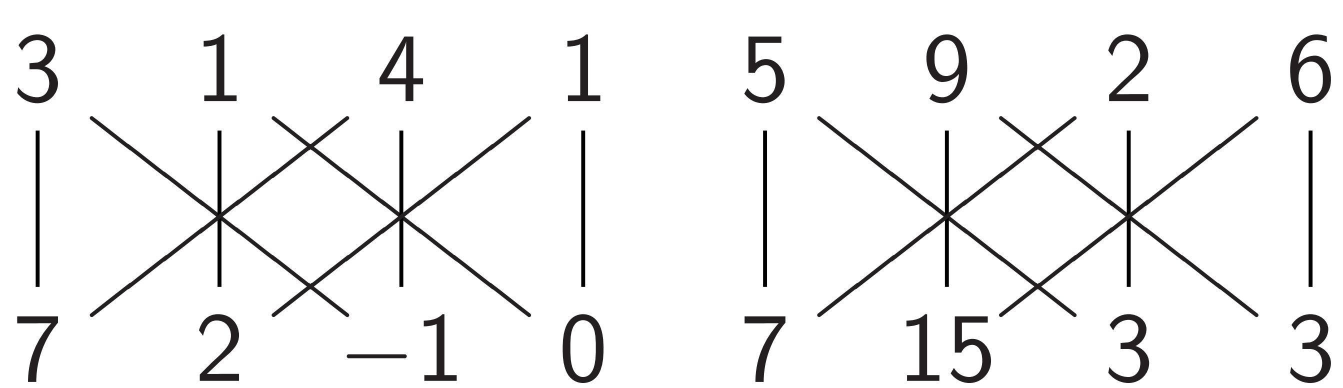
Hadamard$_0$:

$(a, b) \mapsto (a + b, a - b)$.



Hadamard$_1$:

$(a, b, c, d) \mapsto$
$(a + c, b + d, a - c, b - d)$.



Simon's algorithm

Step 8. Hadamard$_2$:

$0, 0, 0, 0, 0, 0, 0, 0,$

$2, 0, \overline{2}, 0, 0, \overline{2}, 0, 2,$

$0, 0, 0, 0, 0, 0, 0, 0,$

$2, 0, \overline{2}, 0, 0, 2, 0, \overline{2},$

$2, 0, 2, 0, 0, \overline{2}, 0, \overline{2},$

$0, 0, 0, 0, 0, 0, 0, 0,$

$0, 0, 0, 0, 0, 0, 0, 0,$

$2, 0, 2, 0, 0, 2, 0, 2.$

Step 9: Measure. Obtain some information about the surprise: a random vector orthogonal to 101.