S. KIMMEL

Classical Models of Computation

- Circuit
- Reversible
- Randomized

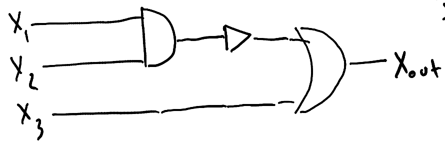
→ Quantum Computation

Circuit - way of describing functions on bits

bit: $x \in \{0,1\}$ time →

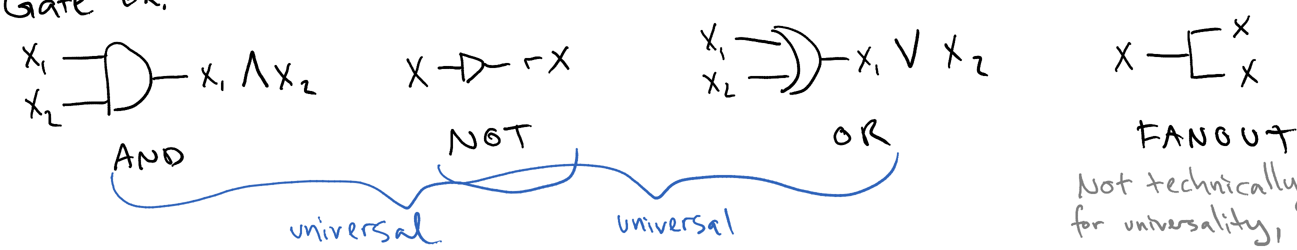
Set of all 3-bit strings

e.g. $\{0,1\}^2 = \{00, 01, 10, 11\}$ (x_1 x_2)



implements the function $\approx f: \{0,1\}^3 \to \{0,1\}$

Gate ex:



FANOUT: Not technically necessary for universality, but useful

Gate set is universal, can compute any $f: \{0,1\}^n \to \{0,1\}^m$

↑ Generalized Boolean function

Circuits provide way of determining how difficult it is to compute a function.

Measures of circuit complexity:

- Gate count = # gates used
- Depth = # time steps (depends on parallelization)
- Width/Size = # input wires

} want to minimize

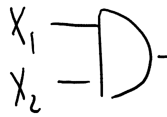
Q

Ex: • find circuit that implements function $f: \{0,1\}^4 \to \{0,1\}$; $f(x_1 x_2) = \begin{cases} 1 & x_1 = x_2 = x_3 = x_4 = 0 \\ 0 & \text{otherwise} \end{cases}$

• what is gate count, depth, width?

S.KIMMEL

Reversible

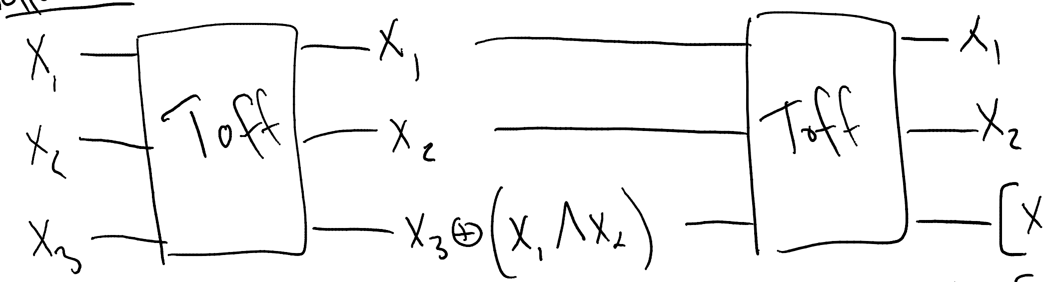


$y = 0 \implies x_1 x_2 = \{00, 01, 10\}$ NOT REVERSIBLE

(only 1 output bit for 2 input bits — information must have been destroyed)

Toffoli Gate

$\oplus$ = addition mod 2

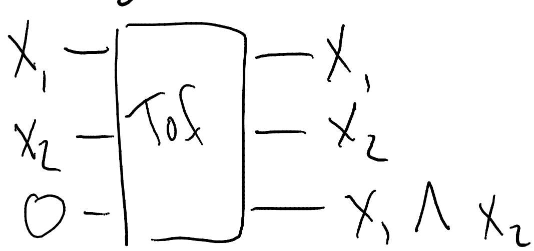


$= X_3 \oplus [(X_1 \wedge X_2) \oplus (X_1 \wedge X_2)]$

$= X_3 \oplus 0 = X_3$

Claim: Toffoli Gate is universal

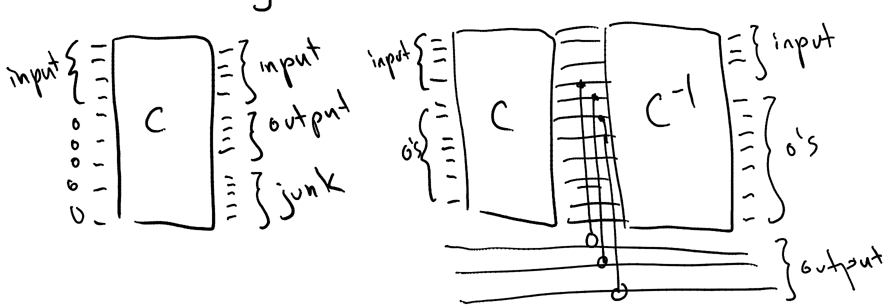
ex: AND using Toffoli



*cost: extra input bit

A.L. Use Toffoli to create NOT, FANOUT, OR

With reversible gates, can do all computations reversibly: (at cost of extra input bits)



If can efficiently compute $x \to f(x)$

Can efficiently & reversibly compute $(x, y, 0) \to (x, y \oplus f(x), 0)$

Q ops that preserve quantumness are reversible!

S. KIMMEL

Randomized Computation

Deterministic: $x_1 = 0$ or $x_1 = 1$

Randomized: $p(0) = .75$, $p(1) = .25$ $\rightarrow \begin{pmatrix} .75 \\ .25 \end{pmatrix}$

Probability vectors:

1-bit: $\begin{pmatrix} p(0) \\ p(1) \end{pmatrix}$

2-bit: $\begin{pmatrix} p(00) \\ p(01) \\ p(10) \\ p(11) \end{pmatrix} = \begin{pmatrix} .1 \\ .7 \\ .2 \\ 0 \end{pmatrix}$

n-bit:
- in $\mathbb{R}^{2^n}$: $\begin{pmatrix} p(00\ldots0) \\ \vdots \\ p(11\ldots1) \end{pmatrix}$
- vector elements $p(s) \geq 0 \;\; \forall\, s \in \{0,1\}^n$
- $\sum_{s \in \{0,1\}^n} p(s) = 1$

Gate: $2^n \times 2^n$ left stochastic matrices: (cols sum to 1 & non-negative entries)

ex: $\begin{pmatrix} 1/2 & 0 \\ 1/2 & 1 \end{pmatrix} \begin{pmatrix} p(0) \\ p(1) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}p(0) \\ \frac{1}{2}p(0) + p(1) \end{pmatrix}$

ex: $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ Identity

ex: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ NOT

AL: Write the 2 bit gate that does the following:

If 1st bit is
- $0 \longrightarrow$ probability of those strings stays same
- $1 \longrightarrow$ probability of those strings are averaged

Q connection: vectors rep. Q states; matrices rep Q. ops; Q is probabilistic

Independent & Correlated Variables

Variables x, y indep if $p(x=x_i, y=y_j) = p(x=x_i) \cdot p(y=y_j) \;\; \forall\, x_i, y_j$ valu

Ind vars:

$$\begin{pmatrix} p_1(0) \\ p_1(1) \end{pmatrix} \otimes \begin{pmatrix} p_2(0) \\ p_2(1) \end{pmatrix} = \begin{pmatrix} p_1(0)p_2(0) \\ p_1(0)p_2(1) \\ p_1(1)p_2(0) \\ p_1(1)p_2(1) \end{pmatrix} = \begin{pmatrix} p(00) \\ p(01) \\ p(10) \\ p(11) \end{pmatrix}$$

Ind ops

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \otimes \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}$$

ex: $\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1/2 \\ 0 \\ 1/2 \end{pmatrix}$

CNOT

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1/2 \\ 0 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1/2 \\ 1/2 \\ 0 \end{pmatrix} \overset{?}{=} \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix}$$

No!

Q Connection: tensor product for ind. states & ops

S.KIMMEL

Other Models: cellular automata, Turing Machines, etc.

Does model matter?

☑ **Church-Turing Thesis**: Any function that can be computed by a physically reasonable model of computation can be computed by a Turing Machine.

def: A computation is **efficient** if number of operations is polynomially related to the number of input bits.

☒. **Strong Church-Turing Thesis**: $\boxed{\text{efficiently}}$

Quantum Computers likely violate!

Q's

Summarize

- with partner, discuss 3 types of computation important for Q.C.
- what does Q.C. have in common w/ these models?