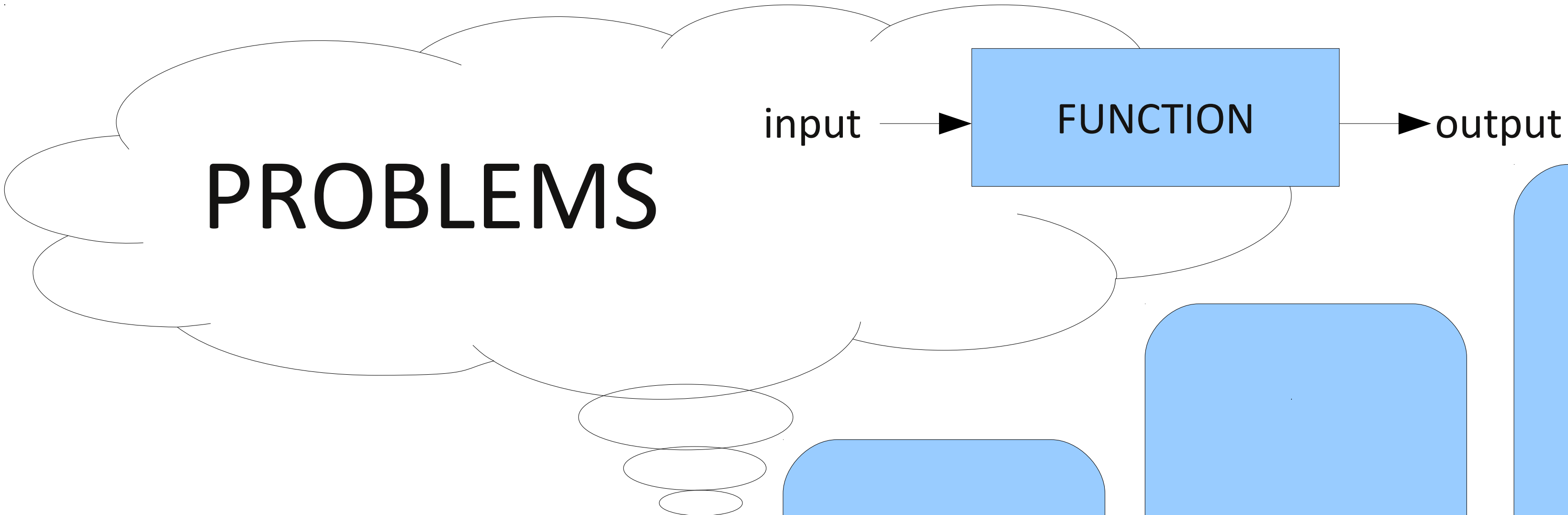
QANSAS 2010

COMPUTATIONAL COMPLEXITY OF THE QUANTUM CIRCUIT MODEL

Debajyoti Bera
IIIT-Delhi

Computational Complexity Theory



- Compute XOR of n binary inputs
- Multiply two n-bit integers
- Sorting n integers between 1 and 100 in incr. order
- Does an directed graph on n nodes contain a cycle?
- Solve a linear system of n equations with integer values

Complexity Theory

Sorting n integers between 1 and 100 in incr. order

Sorting Procedure

Type of Machine

Complexity Theory

Sorting of n integers between 1 and 100

```
quicksort(int a[], int l, int r)
  {
    int v, i, j, t;
    if (r > l)
      {
        v = a[r]; i = l-1; j = r;
        for (;;)
          {
            while (a[++i] < v) ;
            while (a[--j] > v) ;
            if (i >= j) break;
            t = a[i]; a[i] = a[j]; a[j] = t;
          }
        t = a[i]; a[i] = a[r]; a[r] = t;
        quicksort(a, l, i-1);
        quicksort(a, i+1, r);
      }
  }
```

Turing Machine

Random Access Machine

Boolean Circuit

Communication Network

Complexity Theory

Sorting of n integers between 1 and 100

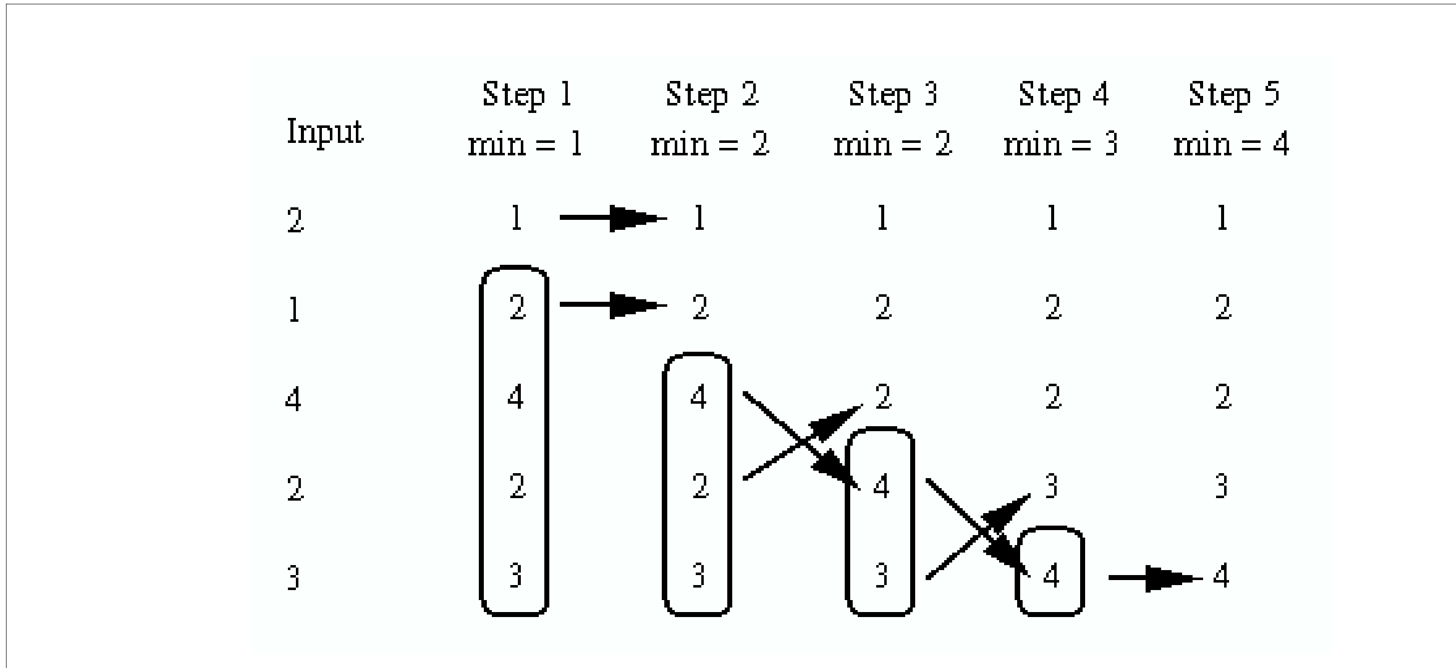
Integer sorting in Word RAM model: $O\left(n\sqrt{\log\frac{w}{\log n}}\right)$

- Y. Han, M. Thorup:
Integer Sorting in O(n √(log log n)) Expected Time and Linear Space, FOCS 2002
- D.G. Kirkpatrick, S. Reisch:
Upper Bounds for Sorting Integers on Random Access Machines, Theoretical Computer Science 28, 1984

Turing Machine | Random Access Machine | Boolean Circuit | Commun-ication Network

Complexity Theory

Sorting of n integers between 1 and 100



Turing Machine

Random Access Machine

Boolean Circuit

Communication Network

Complexity Theory

Sorting of n integers between 1 and 100

Parallel Computing 16 (1990) 183–190 183
North-Holland

Distributed selectsort sorting algorithms on broadcast communication networks

Jau-Hsiung HUANG * and Leonard KLEINROCK **

* *Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.*
** *Computer Science Department, University of California, Los Angeles, California, USA*

Received 14 May 1990
Revised 16 July 1990

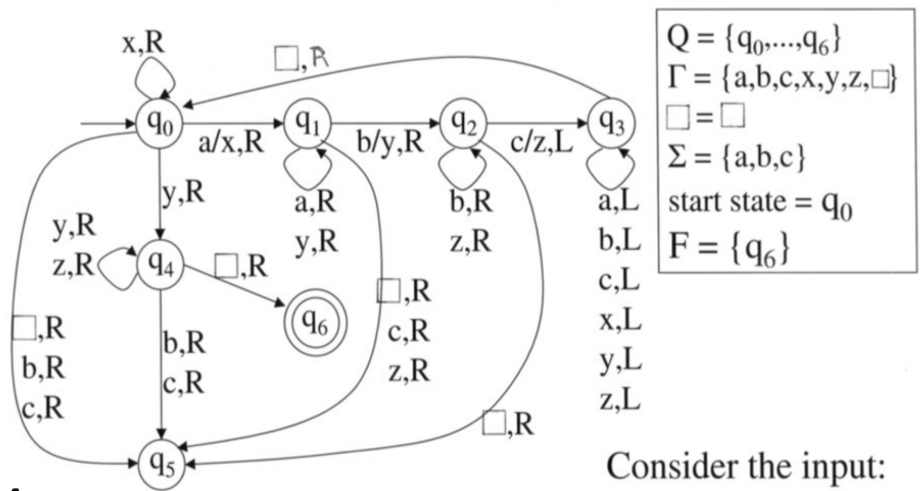
- Turing Machine
- Random Access Machine
- Boolean Circuit
- Commun-ication Network

Complexity Theory

COMPUTATION

ALGORITHM

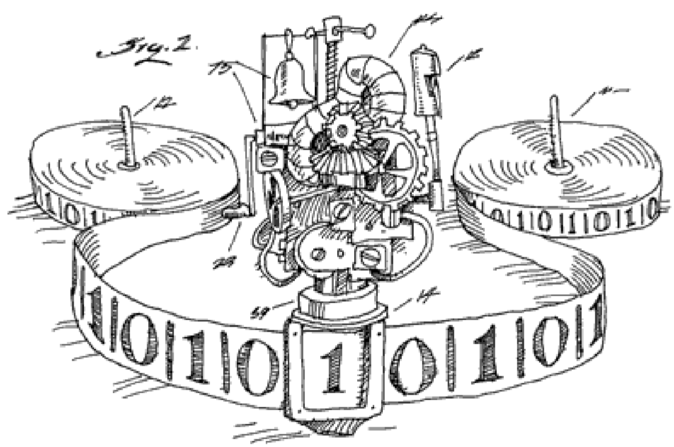
TM for $\{a^k b^k c^k \mid k \geq 1\}$



Consider the input:

…□aabbcc□□…..

MODEL



Complexity Theory

Aim:

Comparison of computations

Analysis of *properties* of computations

Measure:

✓ Define "hardness"/complexity

✓ Use complexity metrics, e.g.,

- running time
- size of local variables
- amount of randomness
- # communication bits

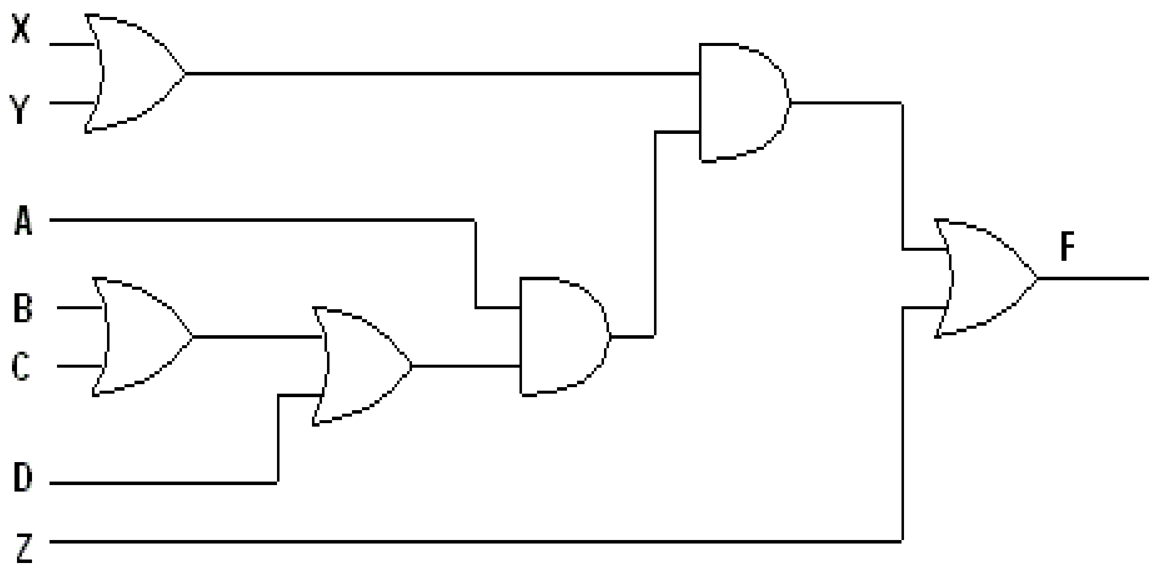
✓ "Equivalence" of models

Outline

- (Classical) Boolean Circuit model
- Quantum Circuit model
- Interesting upper bounds
  - What all are possible ?
- Currently known lower bounds
  - What are not possible ?
- Challenges for tomorrow

Boolean Circuit Model

- (Acyclic) network of Boolean gates
- Gates connected by wires
- “Equivalent” to Turing Machine
- Computes a Boolean function of its inputs



Boolean Circuit Model

Complexity of circuit computation : parameters ?

- circuit family $\{C_n\}$ to compute some function
  - one circuit for each input length
  - circuit C_n computes function on n inputs
- parameter as a function of n (no. of inputs)
  - Multiplication of two n-bit integers – O(n^2) gates

Boolean Circuit Model

- Parameters to measure complexity
  - Types of gates
  - Number of gate input wires
  - Number of gates in circuit
  - “Depth” of circuit

Universal gates
- AND, NOT
- OR, NOT
- NAND

Monotone circuits
- AND, OR

Boolean Circuit Model

- Parameters to measure complexity
  - Types of gates
  - Number of gate input wires
  - Number of gates in circuit
  - “Depth” of circuit

Fanin
(max no. of input)
- Constant (2)
- Unbounded

Boolean Circuit Model

- Parameters to measure complexity
  - Types of gates
  - Number of gate input wires
  - Number of gates in circuit
  - "Depth" of circuit

Size of circuit (number of gates)
- linear in n
- polynomial in n

Boolean Circuit Model

- Parameters to measure complexity
- Types of gates
- Number of gate input wires
- Number of gates in circuit
- "Depth" of circuit

Depth
(maximum number of gates from input to output)
- Constant
- $log_2 n$
- powers of $log_2 n$

Boolean Circuit Model

? $NP = P$

- Polynomial time algorithm for Integer Programming, Satisfiability, TSP

? $NP \subseteq P/poly$

- Polynomial size circuits for NP problems

? $NEXP \subseteq ACC0$ (disproved 1 month ago)

- Poly size circuits with MOD gates for N-EXP problems

We believe the answer is NO!

Mainly used for proving *lower bounds*

Boolean Circuit Model

Interesting results (lower bounds)!

- ✓ Computing CLIQUE requires super-polynomial size circuits using unbounded (fanin) AND, OR, NOT gates

- ✓ Computing PARITY requires exponential size circuits of constant-depth and using unbounded (fanin) AND, OR, NOT gates

- ✓ Computing Mod-3 requires exponential size circuits of constant-depth and using unbounded (fanin) AND, XOR (Mod-2), NOT gates

Outline

- (Classical) Boolean Circuit model
- Quantum Circuit model
- Interesting upper bounds
  - W
- Cur
  - W
- Cha

Motivation:
(1) Lower bounds for computational problems
(2) comparison with classical circuits

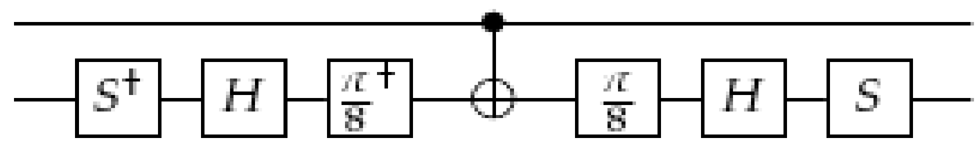
Quantum Circuit Model

Notations

- H – 2-dimensional Hilbert space
  - 2 computational basis states $\{|0\rangle, |1\rangle\}$
- B^n – 2^n-dimensional Hilbert space
  - 2^n computational basis states $\{|0\rangle, \cdots, |2^n - 1\rangle\}$
  - State over n qubits – state in B^n
- quantum gate G – unitary operator acting on states in B^n
- Circuits compute "classical" functions : $(x_i \in \{0,1\})$
  - Input $x_1 \cdots x_n$ – initial state $|x_1\rangle \cdots |x_n\rangle$
  - Output $f(x_1 \cdots x_n)$ – state of output qubit after measurement

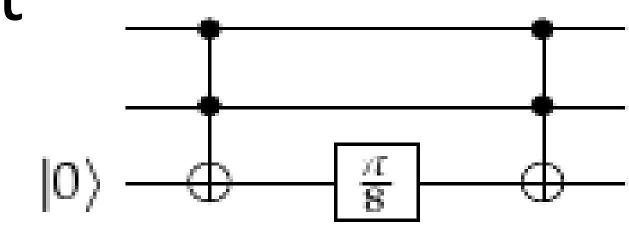
Skipping general introduction to Quantum Computing

Quantum Circuit



Parameters for complexity

- Types of gates
- Number of gate input wires
- Number of gates in circuit
- "Depth" of circuit
- Number of ancilla
  - Extra workspace qubits, initialised to $|0\rangle$
  - Unlike classical circuits, cannot reuse/overwrite
  - Clean circuits: ancilla returned to initial state
  - Robust circuits: accepts ancilla in any initial state



Quantum Gates

- Fixed family
- Single qubit gates – any reasonable set
  - Hadamard (H), Phase, $\pi/8$, Z gate etc.
  - Provide "quantum" behaviour
- Multi-qubit (classical) gates – unbounded fanin
  - Generalized Toffoli (T)
  - Generalized Z
  - Parity gate
  - Threshold gate
  - "Fanout" gate!
- $T, H, \pi/8$ – "universal" family

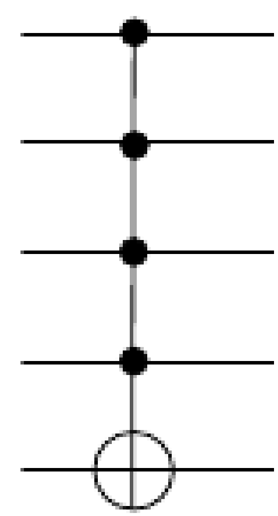
Multi Qubit Gates

$$f : \{0,1\}^n \rightarrow \{0,1\}$$

$$|x_1, \cdots, x_n, b\rangle \xrightarrow{G} |x_1, \cdots, x_n, b \oplus f(x_1, \cdots, x_n)\rangle$$

- (Generalized) Toffoli – AND
- Parity – MOD2
  - MODq
- Threshold
- Generalized Z
- "Fanout"

$$f = \bigwedge_{i=1}^{n} x_i$$



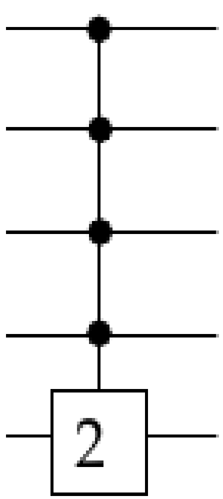
Multi Qubit Gates

$$f : \{0,1\}^n \rightarrow \{0,1\}$$

$$|x_1, \cdots, x_n, b\rangle \xrightarrow{G} |x_1, \cdots, x_n, b \oplus f(x_1, \cdots, x_n)\rangle$$

- (Generalized) Toffoli – AND
- Parity – MOD2
  - MODq
- Threshold
- Generalized Z
- "Fanout"

$$f = \bigoplus_{i=1}^{n} x_i$$



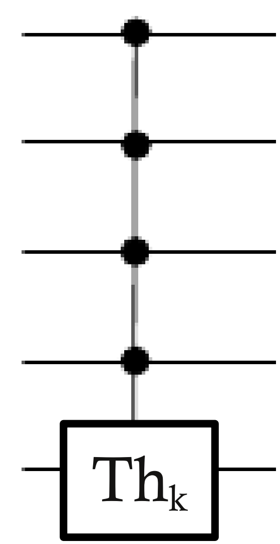
Multi Qubit Gates

$$f : \{0,1\}^n \rightarrow \{0,1\}$$

$$|x_1, \cdots, x_n, b\rangle \xrightarrow{G} |x_1, \cdots, x_n, b \oplus f(x_1, \cdots, x_n)\rangle$$

- (Generalized) Toffoli – AND
- Parity – MOD2
  - MODq
- Threshold$_k$
- Generalized Z
- “Fanout”

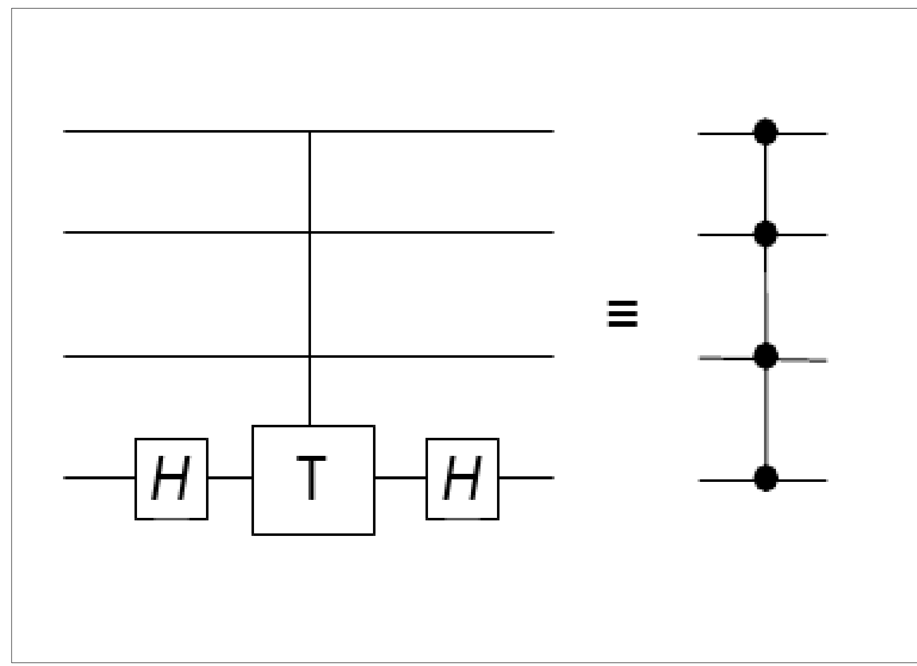
$$f(x_1, \cdots, x_n) = \begin{cases} 1 & \sum_i x_i \geq k \\ 0 & \sum_i x_i < k \end{cases}$$



Multi Qubit Gates

$$|x_1, \cdots, x_n, b\rangle \xrightarrow{G} (-1)^{x_1 \cdots x_n} |x_1, \cdots, x_n, b\rangle$$

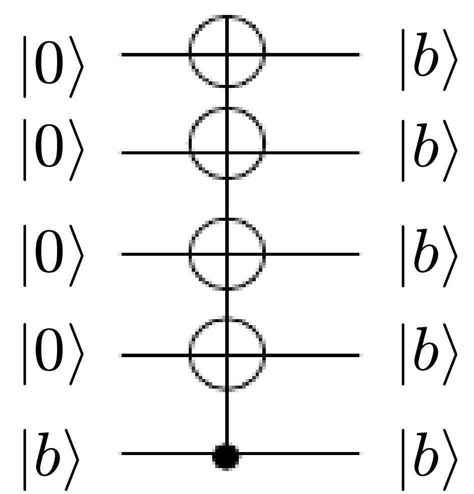
- (Generalized) Toffoli – AND
- Parity – MOD2
  - MODq
- Threshold
- Generalized Z
- "Fanout"



Multi Qubit Gates

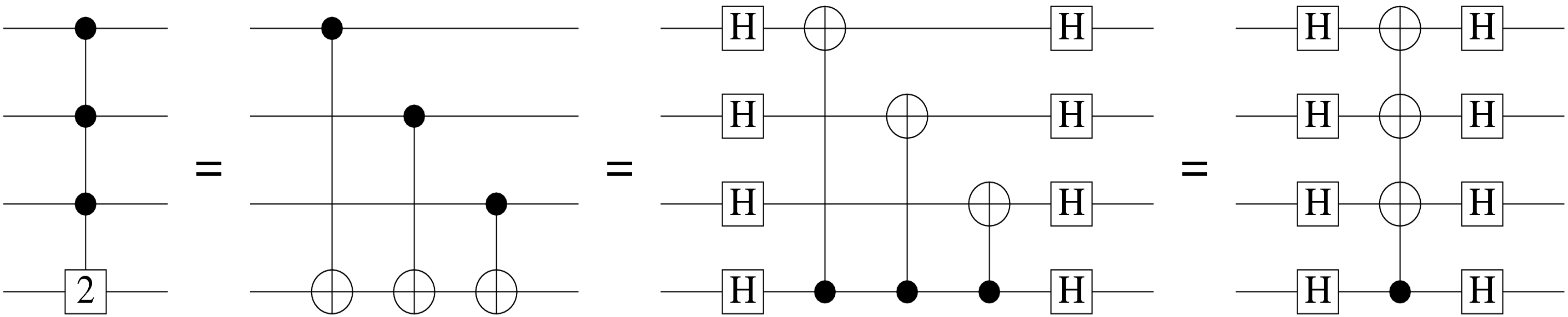
$$|x_1, \cdots, x_n, b\rangle \xrightarrow{G} |b \oplus x_1, \cdots, b \oplus x_n, b\rangle$$

- (Generalized) Toffoli – AND
- Parity – MOD2
  - MODq
- Threshold
- Generalized Z
- Fanout gate!
  - Only copies basis states



MOD2 function

Power of fanout gate



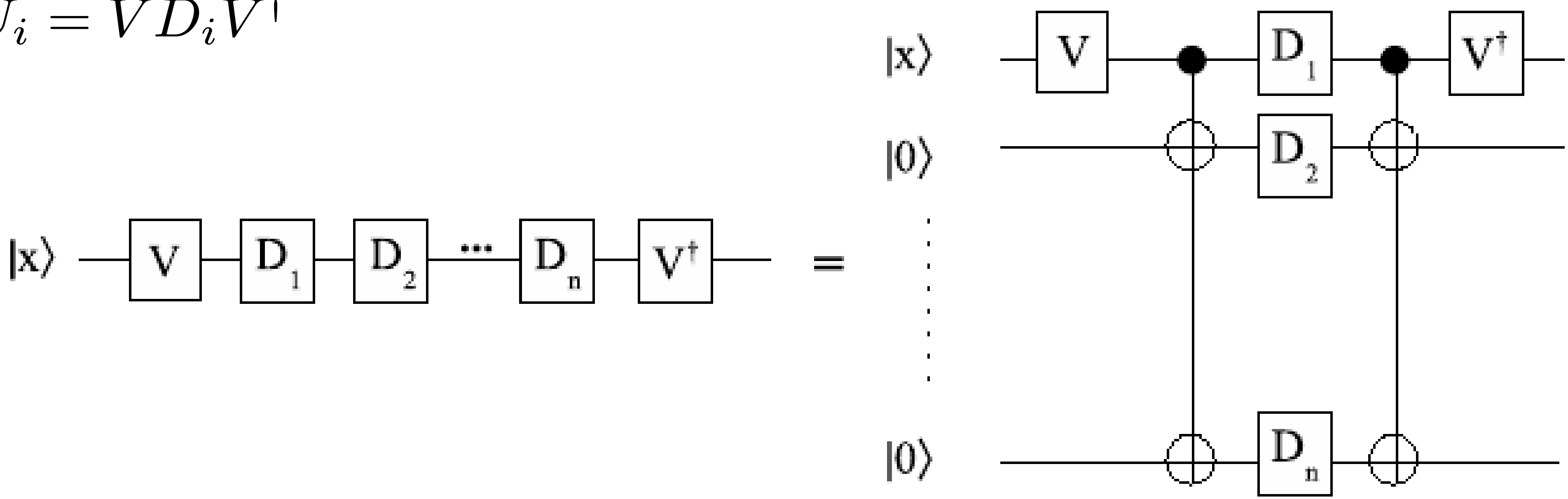
QUANTUM
single qubit gate + Toffoli gate + fanout gate
in
constant depth, linear size

CLASSICAL
Not gate + unbounded AND gate
requires at least
exponential gates and constant depth

MOD*q* function

$U_1, \ldots, U_n$ are simultaneously diagonalizable:

$U_i = V D_i V^\dagger$



QUANTUM

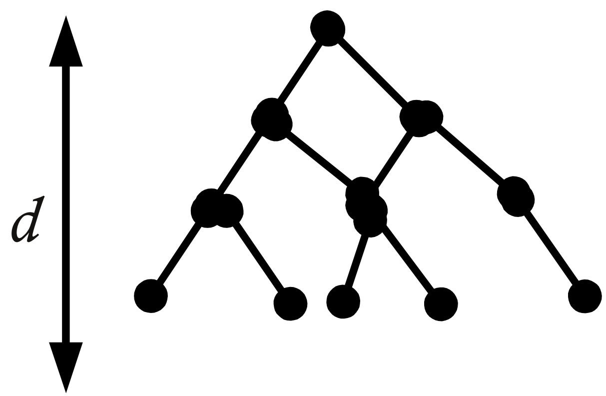
MOD*q* function = constant depth using polynomial number of single qubit, T and MOD*p* gates

CLASSICAL

MOD*q* funtion requires exponentially many MOD*p* gates, for constant-depth circuits, for primes *p*, *q*

Arithmetic functions

Constant-depth poly size circuit with bounded fanin gates

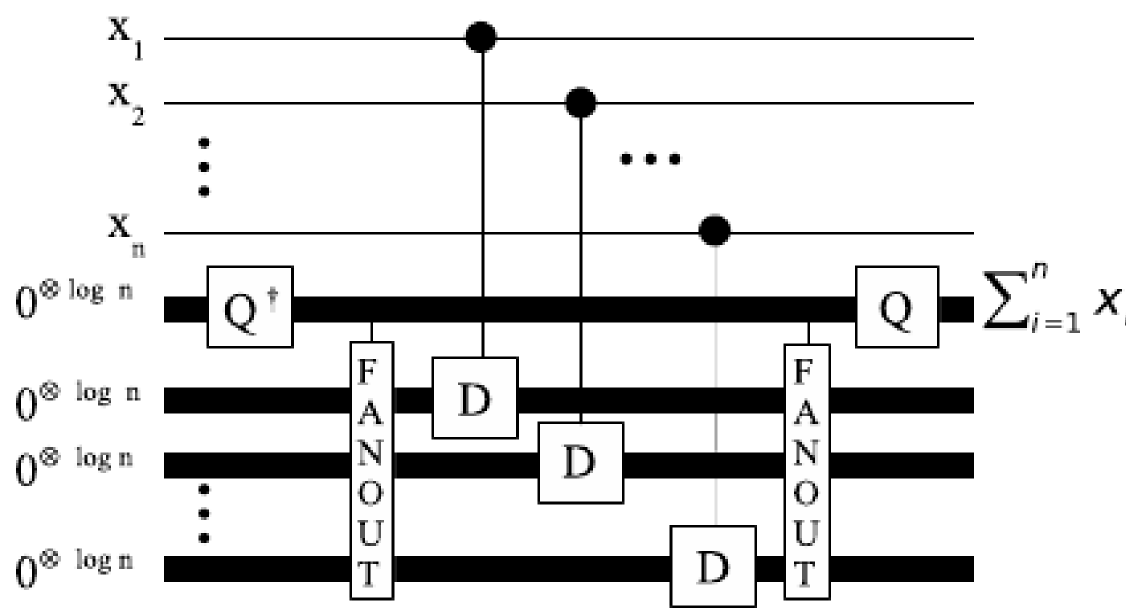


CLASSICAL

Even with unbounded fanout, the output bit can depend on only a constant number of inputs

QUANTUM

Threshold, parity, QFT, addition, multiplication, division, sorting can be *approximated*



Limitations : Parity / MOD2

MOD2 cannot be computed in constant depth

QUANTUM

MOD2 requires more than constant-depth using single qubit and Toffoli gates and linearly many ancilla

CLASSICAL

MOD2 requires exponentially many AND, OR, NOT gates for constant-depth circuits

Work to do...

- Fanout gates seems too powerful. Yet necessary to adhere to our understanding.
  - Better replacement ?
- Is fanout really that powerful ?
  - Compute fanout using unbounded Toffoli and unlimited ancilla ?
  - Can one compute Threshold using fanout and vice versa?
- Approximate Threshold/MOD in constant depth with bounded fanin gates with *exponentially* small error ?

Work to do...

- Fanout is equivalent to constructing a cat state in constant depth: $\frac{1}{\sqrt{2}}\left(|00\cdots 0\rangle + |11\cdots 1\rangle\right)$
  - Useful measure of entanglement to prove circuit lower bounds ?

- Ancilla – important resource
  - Constant / Linear / Polynomial ancilla ?

- Circuit computing probabilistic functions ?

Thank you. Questions?