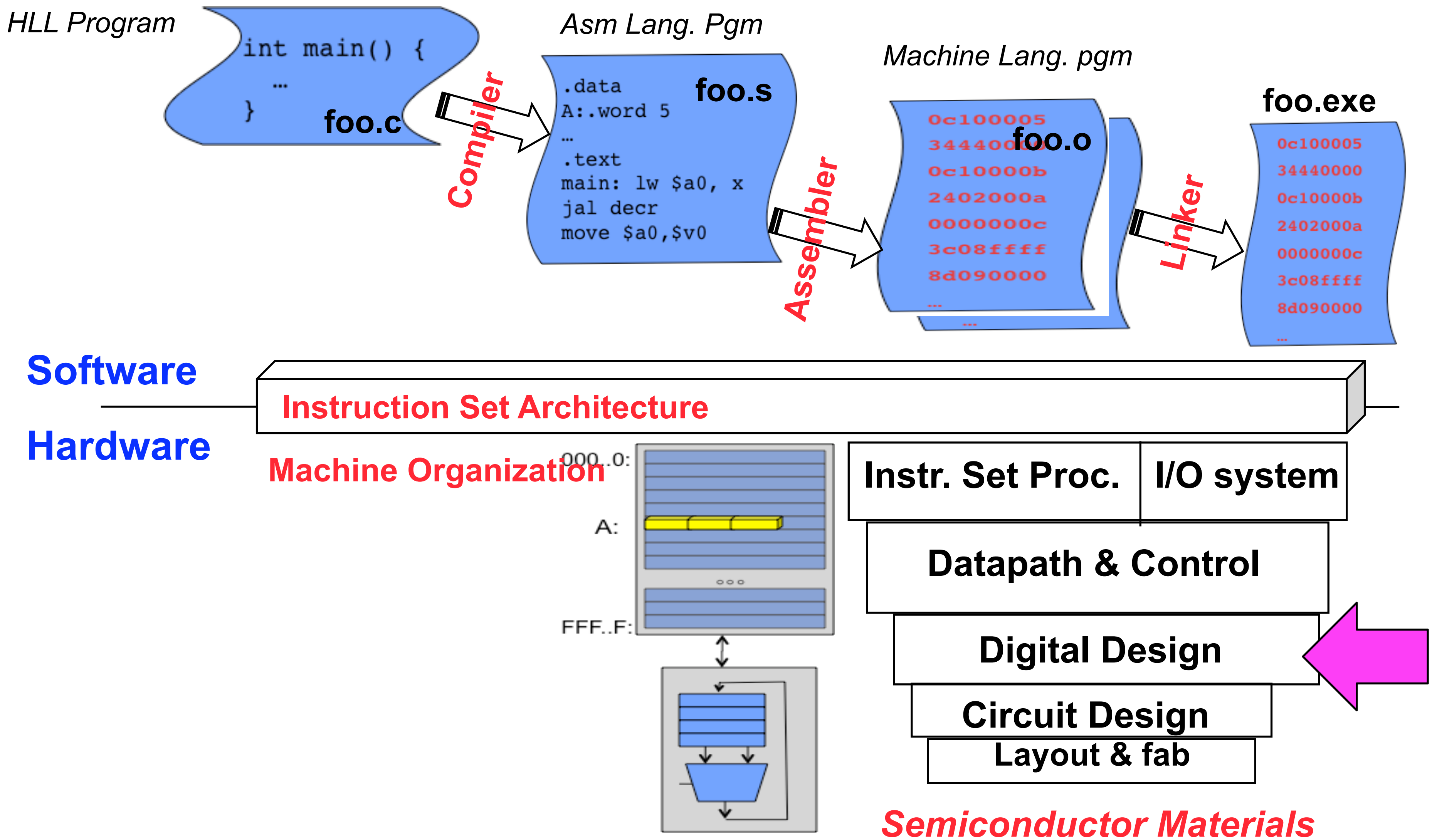
CS61CL Machine Structures

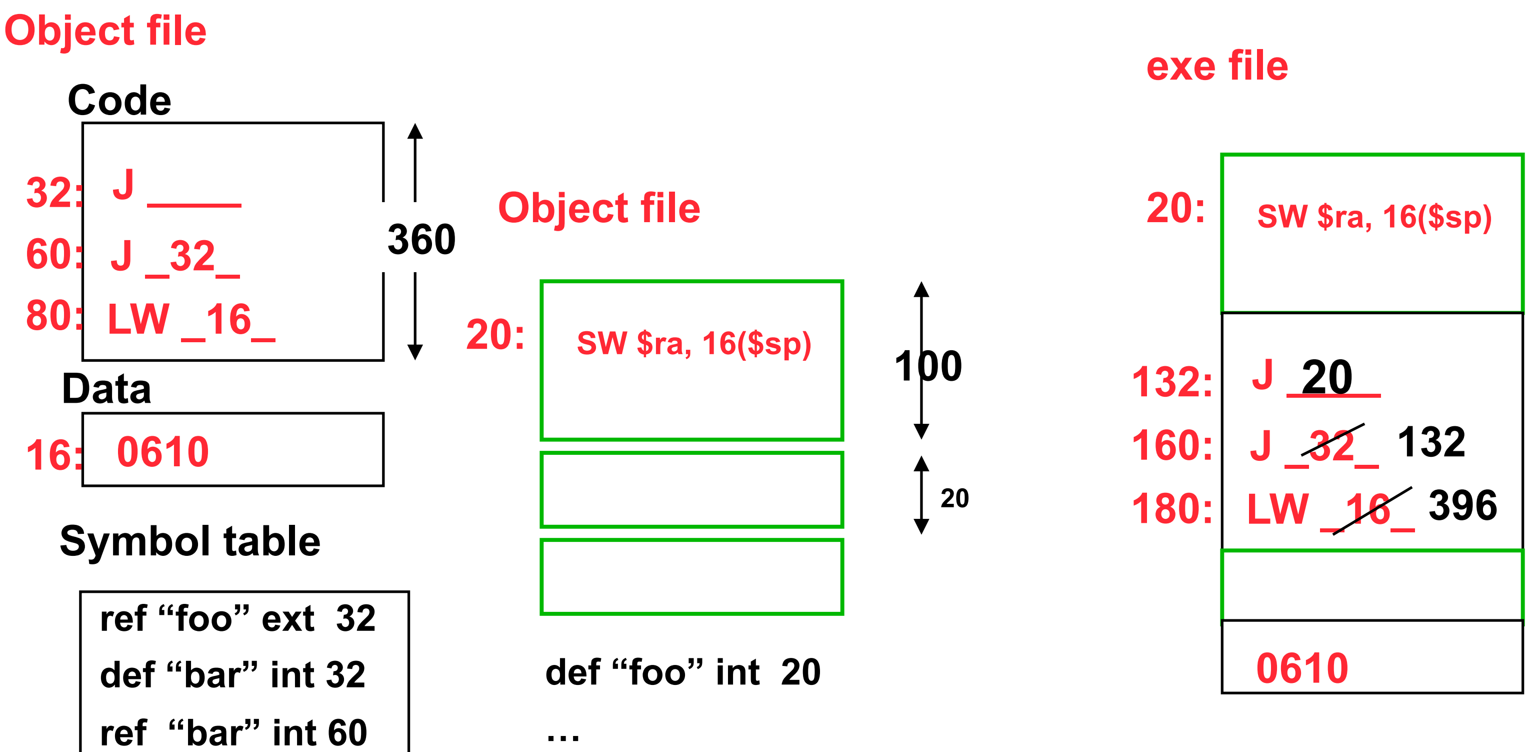
Lec 7 – Introduction to Digital Design

David Culler
Electrical Engineering and Computer Sciences
University of California, Berkeley

CS61CL Road Map



Linking



ref "foo" ext 32
def "bar" int 32
ref "bar" int 60
ref "xyz" int 80
ddef "xyz" int 16

- **Resolve names to addresses**
- **Relocate code and data blocks**
  - Adjust internally resolved addresses

Questions

Evolution of Instruction Sets

Single Accumulator (EDSAC 1950)

Accumulator + Index Registers

(Manchester Mark I, IBM 700 series 1953)

Separation of Programming Model
from Implementation

High-level Language Based (Stack)
(B5000 1963)

Concept of a Family
(IBM 360 1964)

General Purpose Register Machines

Complex Instruction Sets

(Vax, Intel 432 1977-80)

iX86?

Load/Store Architecture

(CDC 6600, Cray 1 1963-76)

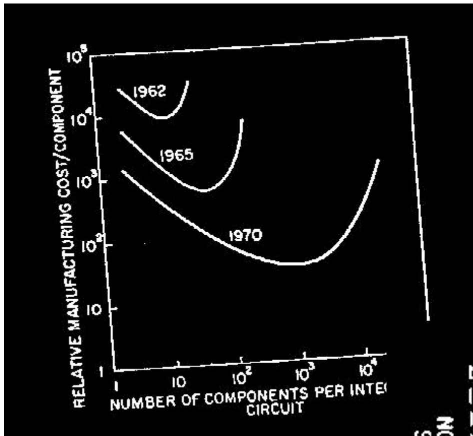
RISC

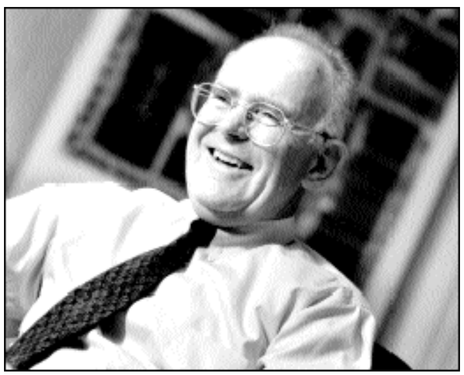
(MIPS,Sparc,HP-PA,IBM RS6000, 1987)

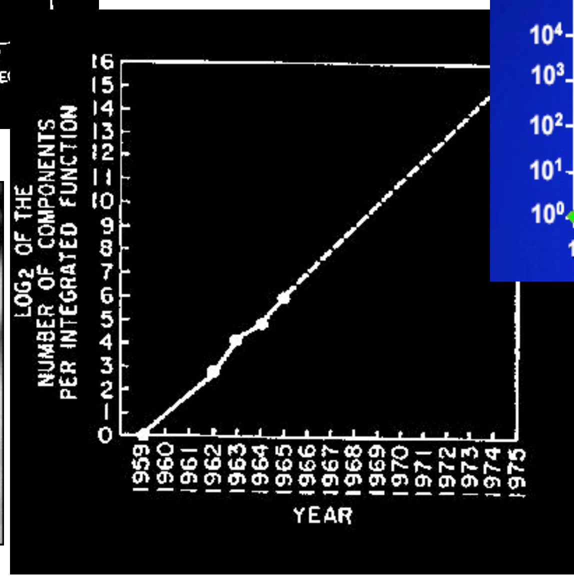
Dramatic Technology Advance

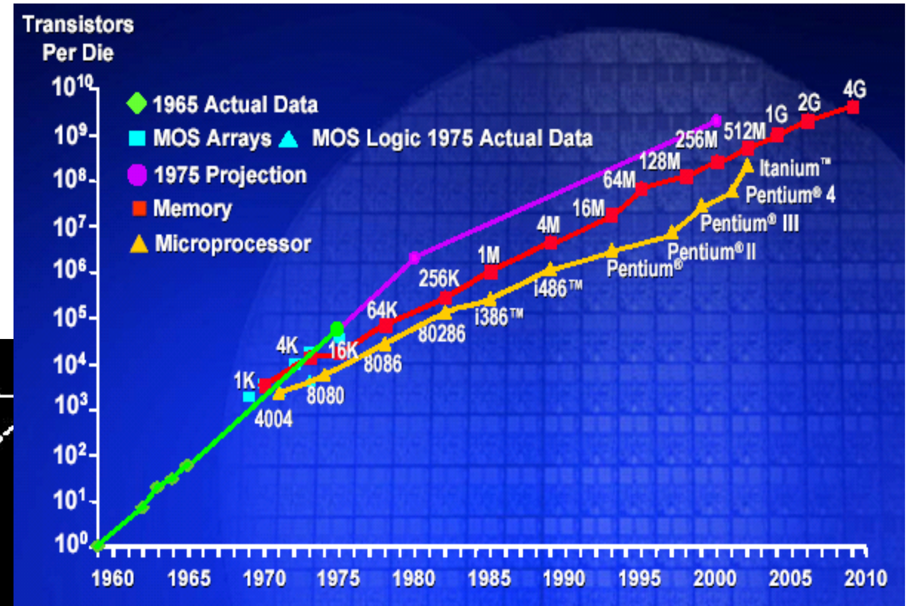
- **Prehistory: Generations**
  - 1st Tubes
  - 2nd Transistors
  - 3rd Integrated Circuits
  - 4th VLSI….
- **Discrete advances in each generation**
  - Faster, smaller, more reliable, easier to utilize
- **Modern computing: Moore's Law**
  - Continuous advance, fairly homogeneous technology

Moore's Law



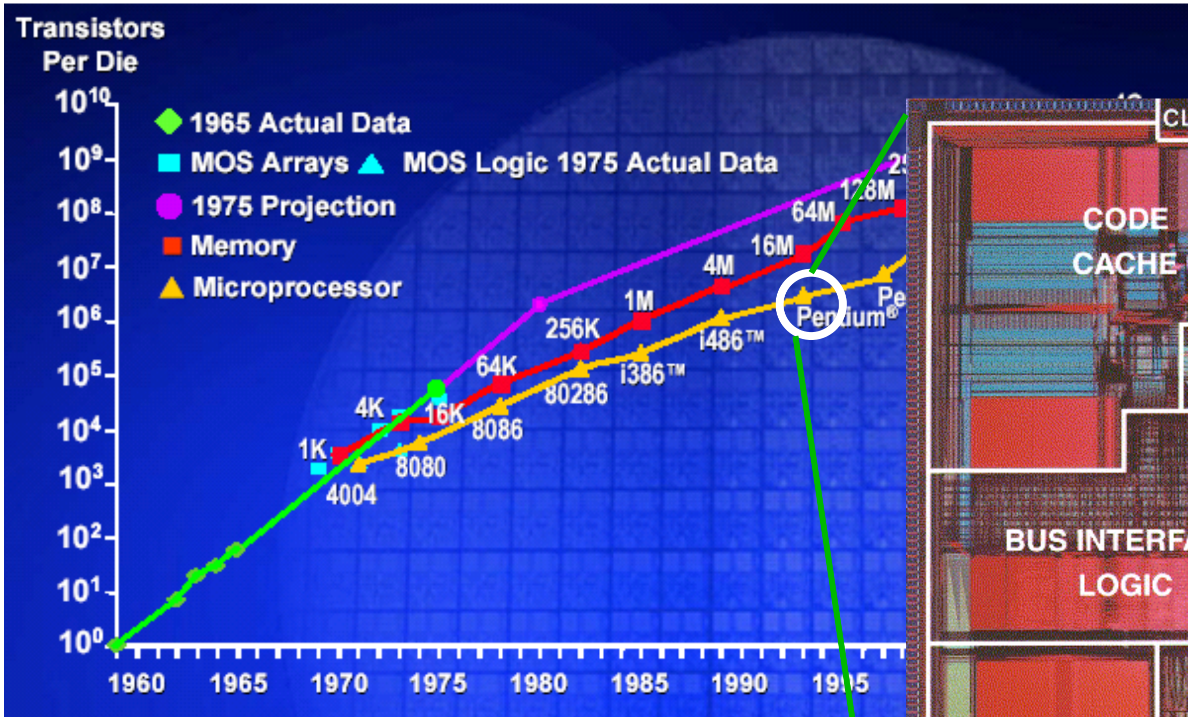


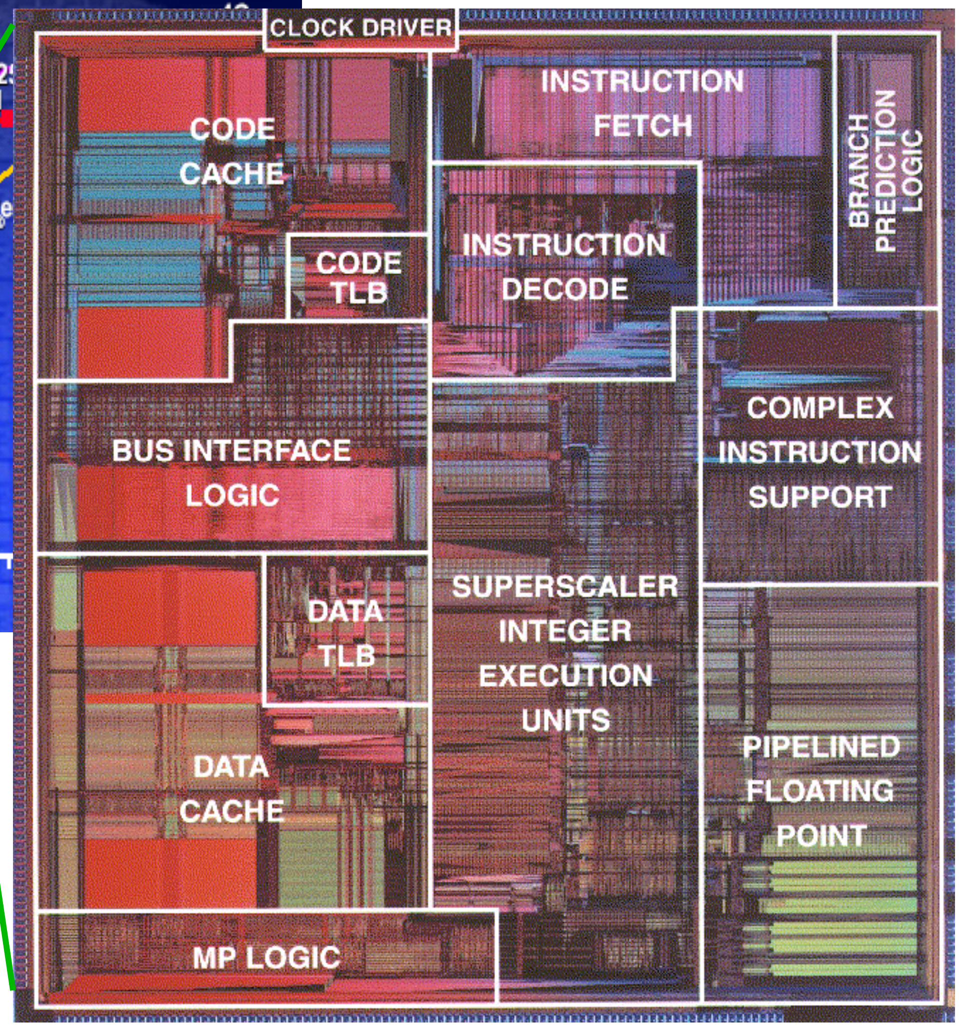




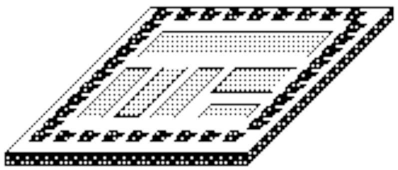
- "Cramming More Components onto Integrated Circuits"
  - Gordon Moore, Electronics, 1965
- # on transistors on cost-effective integrated circuit double every 18 months

Example: Intel Pentium



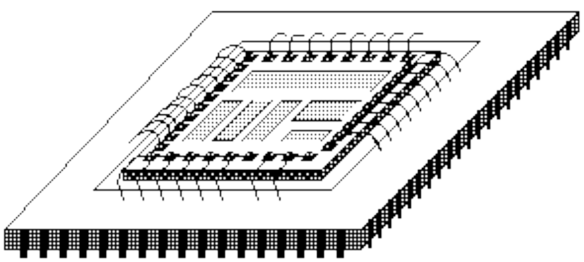


Integrated Circuits



- Primarily Crystalline Silicon
- 1mm - 25mm on a side
- 100 - 200M transistors
- (25 - 50M “logic gates")
- 3 - 10 conductive layers
- 2002 - feature size ~ 0.13um = 0.13×10^{-6} m
- “CMOS” most common - complementary metal oxide semiconductor

Chip in Package



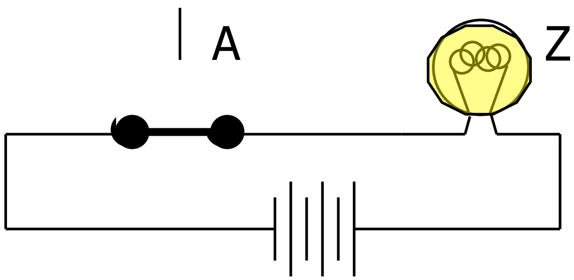
- Package provides:
  - spreading of chip-level signal paths to board-level
  - heat dissipation.
- Ceramic or plastic with gold wires.

Integrated Circuits

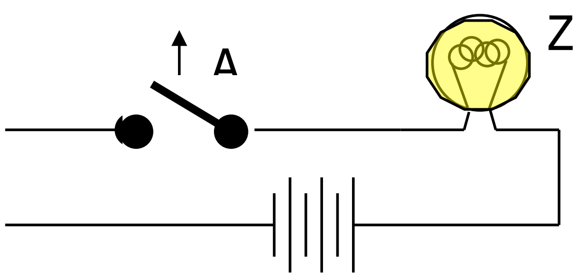
- **Uses for digital IC technology today:**
  - <u>standard microprocessors</u>
    - » used in desktop PCs, and embedded applications
    - » simple system design (mostly software development)
  - <u>memory chips (DRAM, SRAM)</u>
  - <u>application specific ICs (ASICs)</u>
    - » custom designed to match particular application
    - » can be optimized for low-power, low-cost, high-performance
    - » high-design cost / relatively low manufacturing cost
  - <u>field programmable logic devices (FPGAs, CPLDs)</u>
    - » customized to particular application after fabrication
    - » short time to market
    - » relatively high part cost
  - <u>standardized low-density components</u>
    - » still manufactured for compatibility with older system designs

Switches: the basic element

- **Implementing a simple circuit:**



close switch (if *A* is “1” or asserted)
and turn on light bulb (Z)



open switch (if *A* is “0” or unasserted)
and turn off light bulb (Z)

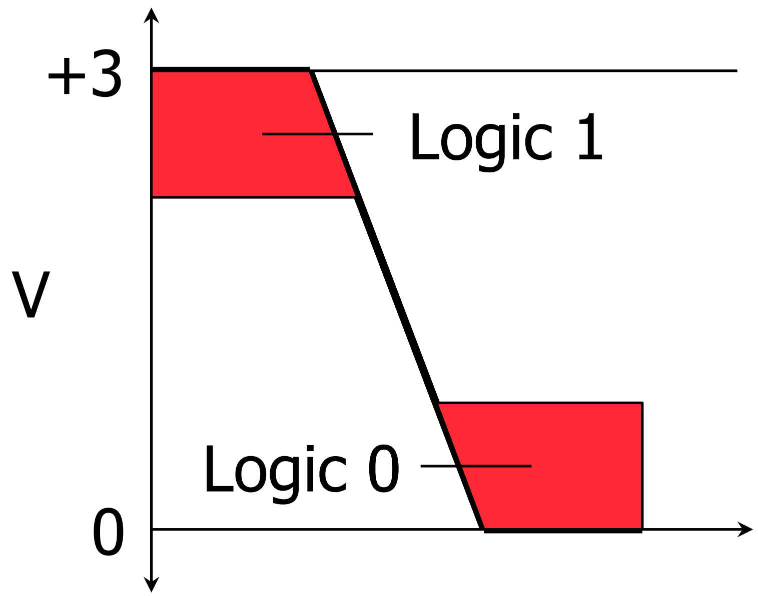
$$Z \equiv A$$

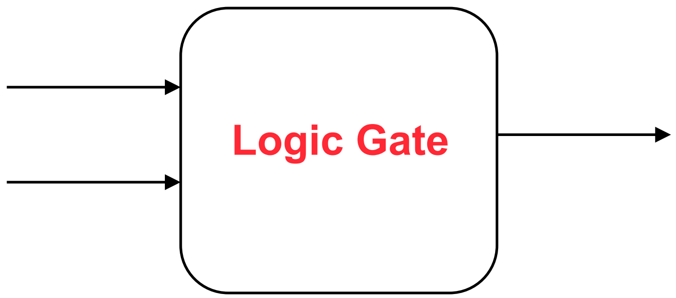
Physical world to Digital world

Technology	State "0"	State "1"
Relay logic	Circuit Open	Circuit Closed
CMOS logic	0.0-1.0 volts	2.0-3.0 volts
Transistor transistor logic (TTL)		0.0-0.8 volts 2.0-5.0 volts
Fiber Optics	Light off	Light on
Dynamic RAM	Discharged capacitor	Charged capacitor
Nonvolatile memory (erasable)	Trapped electrons	No trapped electrons
Programmable ROM	Fuse blown	Fuse intact
Bubble memory	No magnetic bubble	Bubble present
Magnetic disk	No flux reversal	Flux reversal
Compact disc	No pit	Pit

Sense the logical value, manipulate in a systematic fashion.

The Digital Abstraction



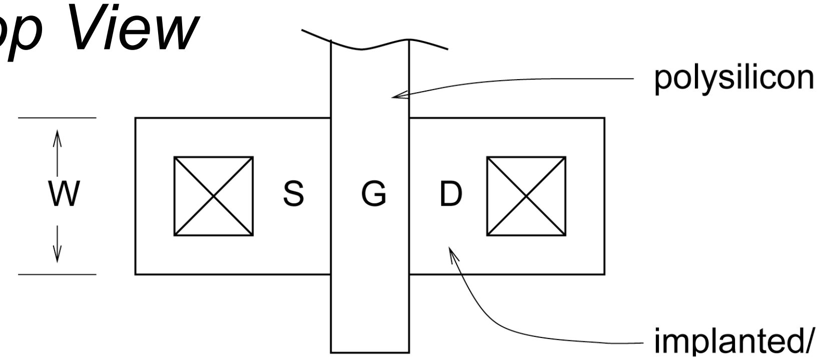


- **Logical 1 (true) : V > Vdd –V th**
- **Logical 0 (false) : V < Vth**
- **Logical Gates**
  - **behave like boolean operators on these voltage signals**
  - **Produce signals that can be treated as logical values**

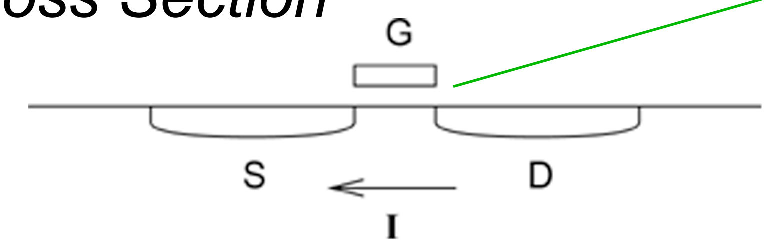
CMOS "Devices"

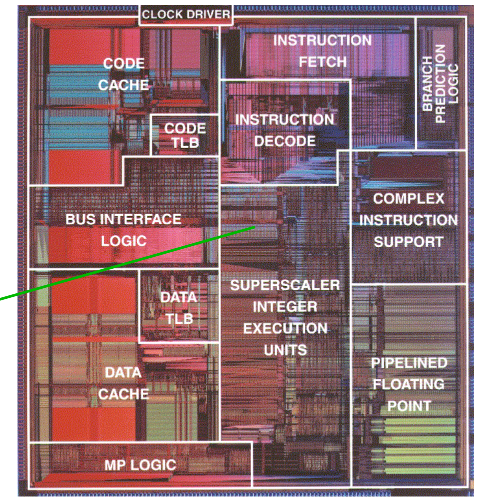
- **MOSFET (Metal Oxide Semiconductor Field Effect Transistor)**

Top View

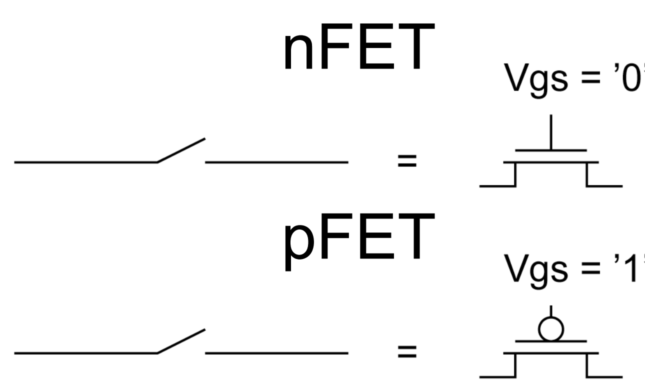


Cross Section



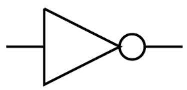


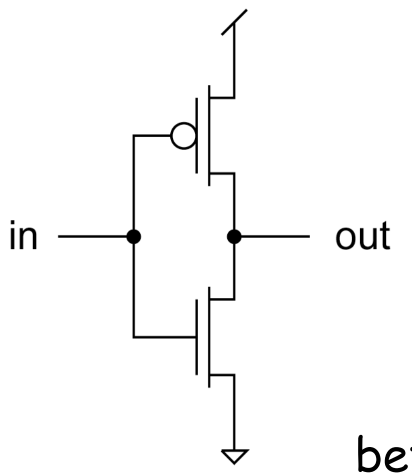
- **Essentially a voltage-controlled switch**
- **N: closed when gate is Hi**
- **P: closed when gate is Lo**



Transistor-level Logic Circuits (inv)

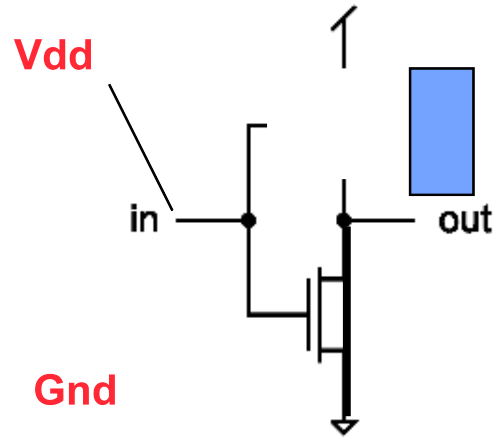
- **Inverter (NOT gate):**

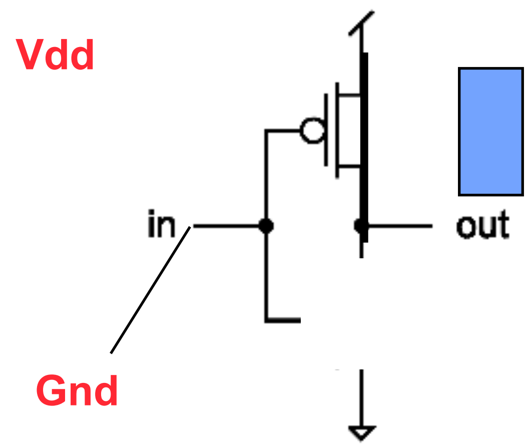




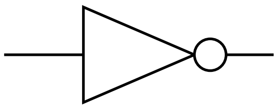
what is the relationship between in and out?

in	out
0 volts	3 volts
3 volts	0 volts



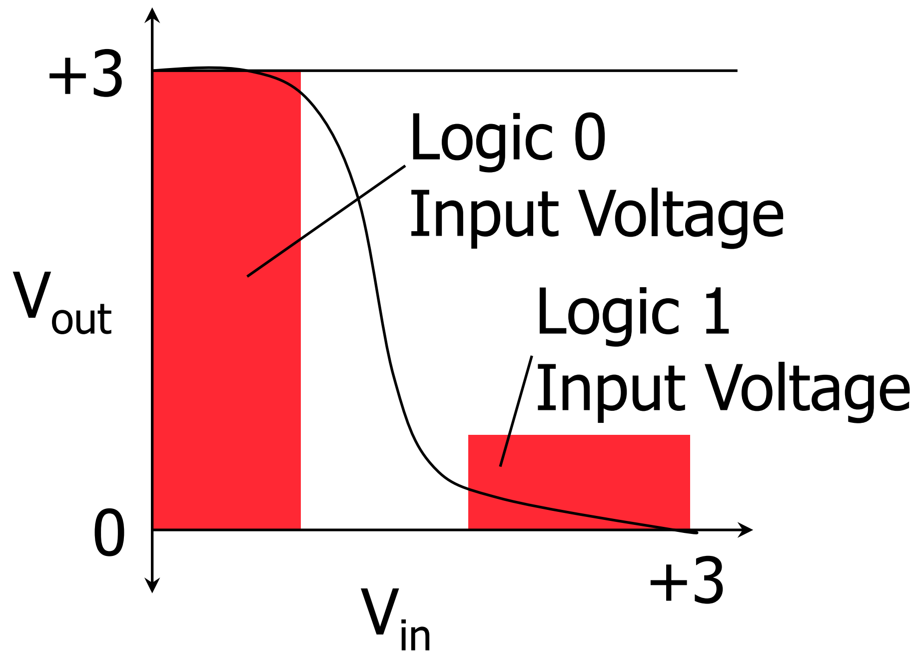


Example: NOT



not(out, in)

in	out
F	T
T	F



Big idea: Self-restoring logic

- **CMOS logic gates are *self-restoring***
  - **Even if the inputs are imperfect, switching time is fast and outputs go "rail to rail"**
  - **Doesn't matter how many you cascade**
    - » **Although propagation delay increases**
- **Limit fan-out to ensure sharp and complete transition**

Combinational Logic Symbols

- Common combinational logic systems have standard symbols called *logic gates*
  - Buffer, NOT

  - AND, NAND

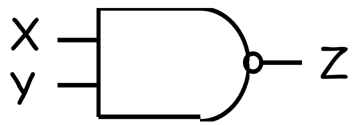
  - OR, NOR

A	B	A*B	A+B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Easy to implement with CMOS transistors (the switches we have available and use most)

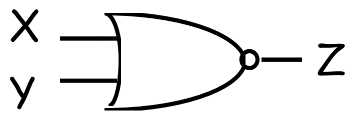
more Boolean Expressions to Logic Gates

- **NAND**



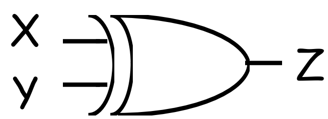
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

- **NOR**



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

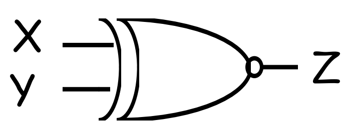
- **XOR**
  $X \oplus Y$



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

X xor Y = X Y' + X' Y
X or Y but not both
("inequality", "difference")

- **XNOR**
  X = Y

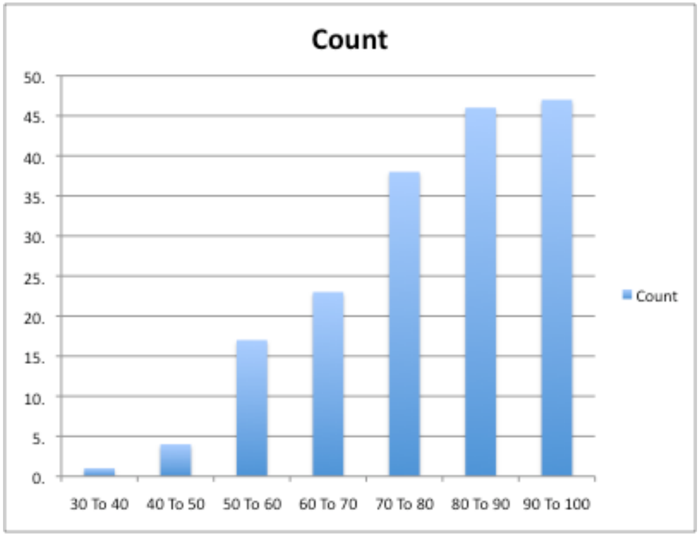


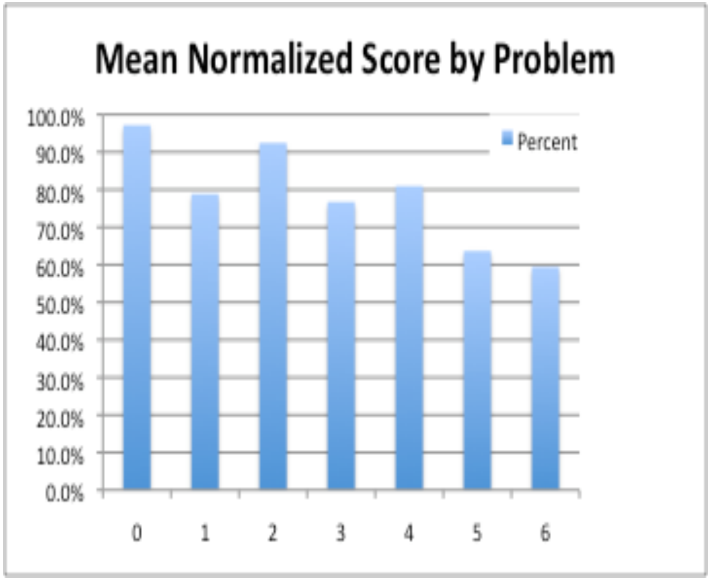
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

X xnor Y = X Y + X' Y'
X and Y are the same
("equality", "coincidence")

Administration

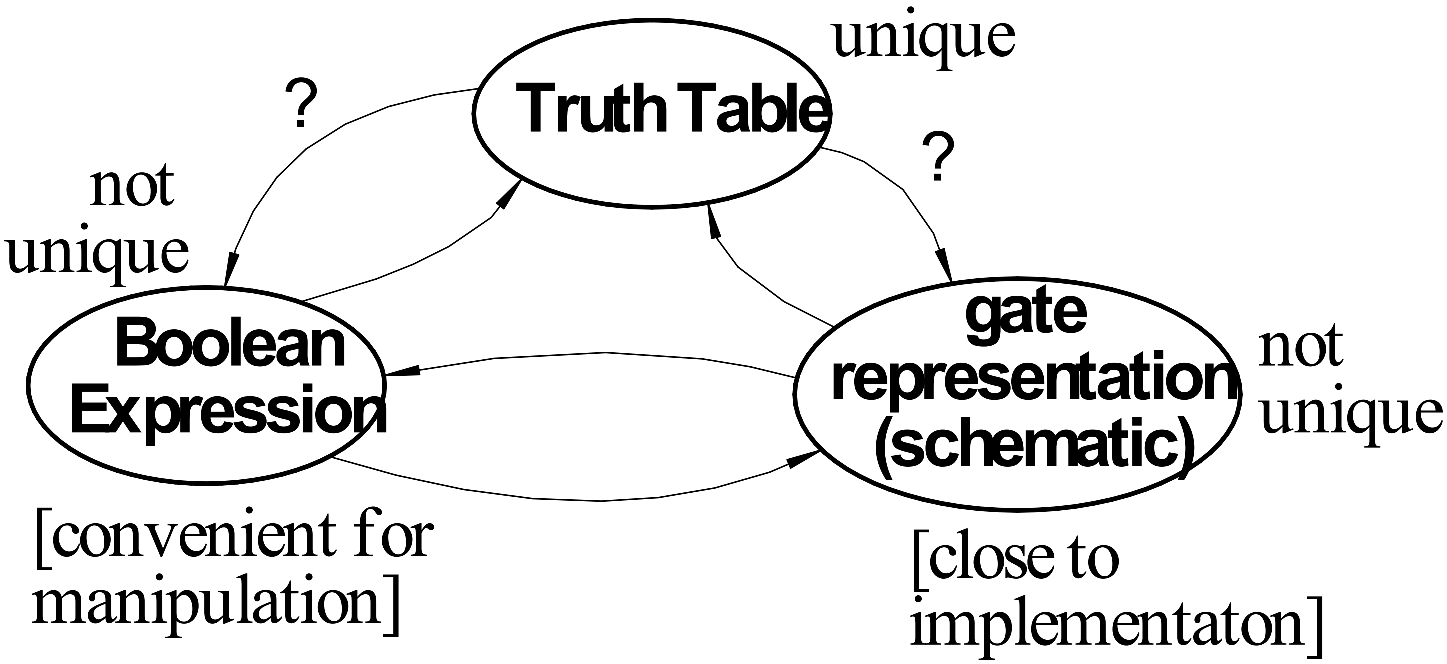
- Great job on Mid Term
  - Mean: 79%, Median: 82%, Min: 36, Max: 99 (3)
- Project 2 is due Monday 10/26
  - Work in pieces
    - » call snprintf / save / restore / rtn
    - » copy format to buffer respecting bufferSize
    - » dispatch to one format function
    - » add other format functions
- Homework 6 out tonight





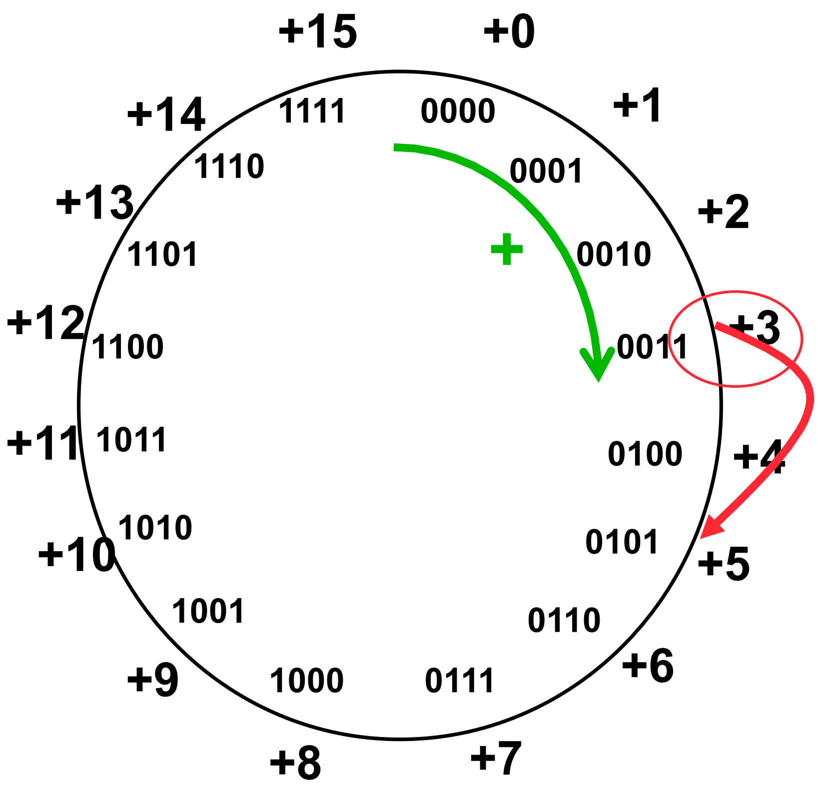
Relationship Among Representations

* Theorem: Any Boolean function that can be expressed as a truth table can be written as an expression in Boolean Algebra using AND, OR, NOT.



How do we convert from one to the other?

Recall: Addition



Example: 3 + 2 = 5

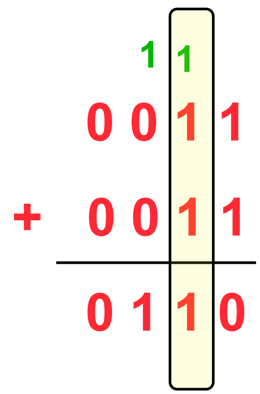
Unsigned binary addition

Is just addition, base 2

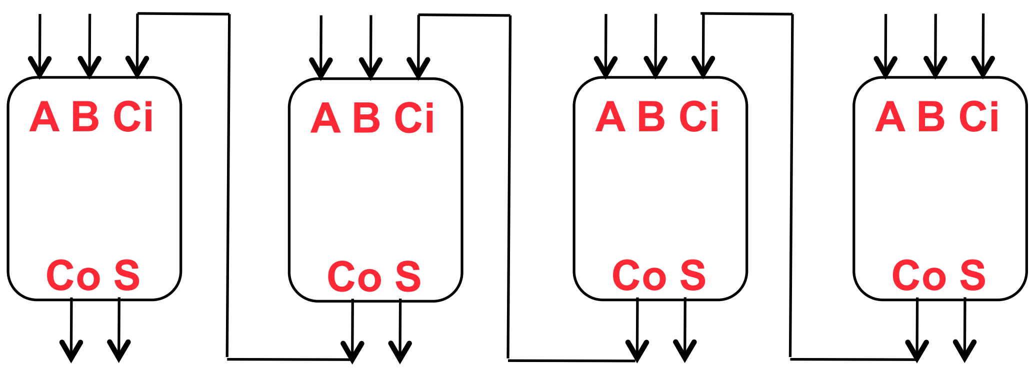
Add the bits in each position and carry

```
    1
  0 0 1 1
+ 0 0 1 0
---------
  0 1 0 1
```

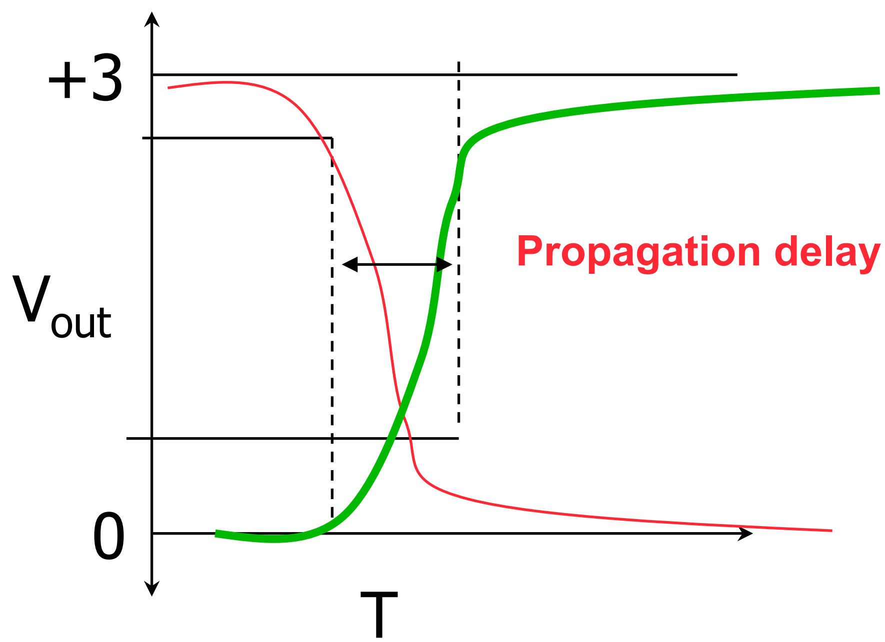
Design an Adder



Ci	A	B	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	1	1	1

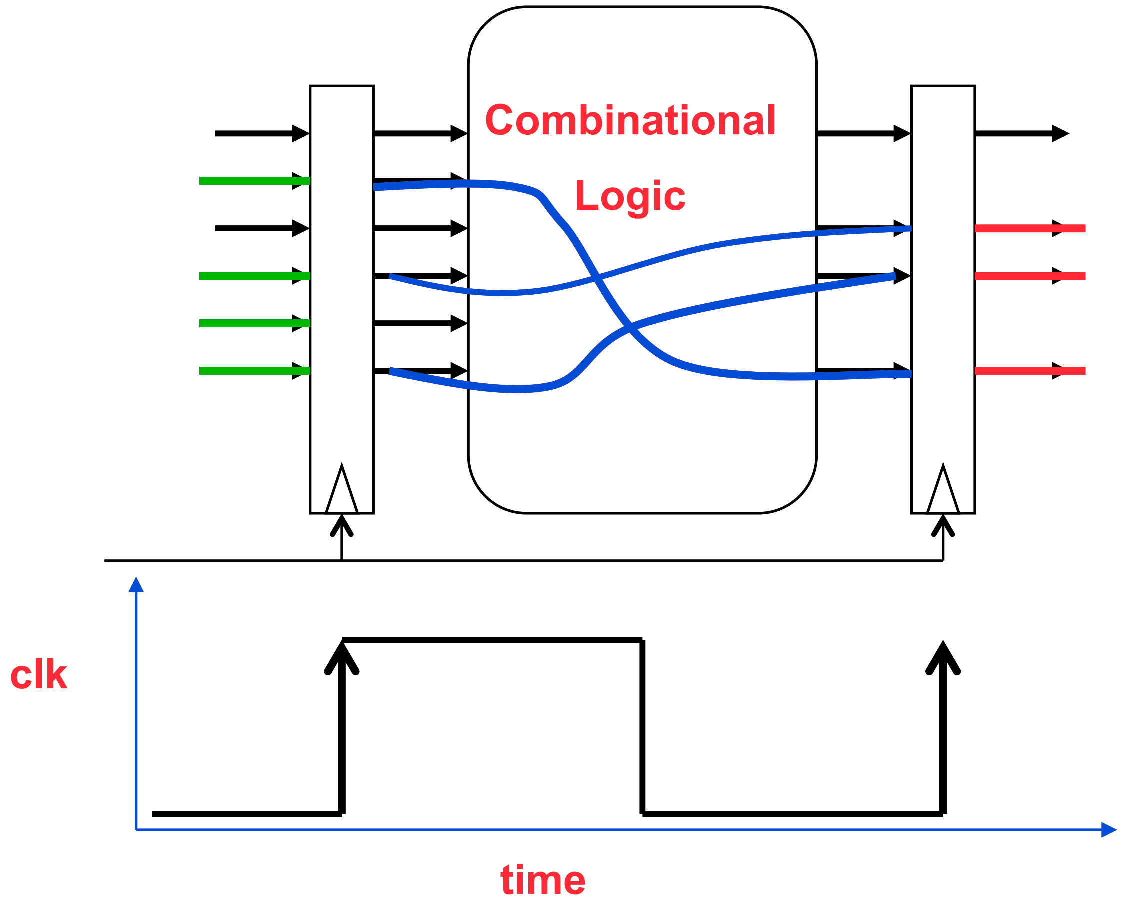


Element of Time

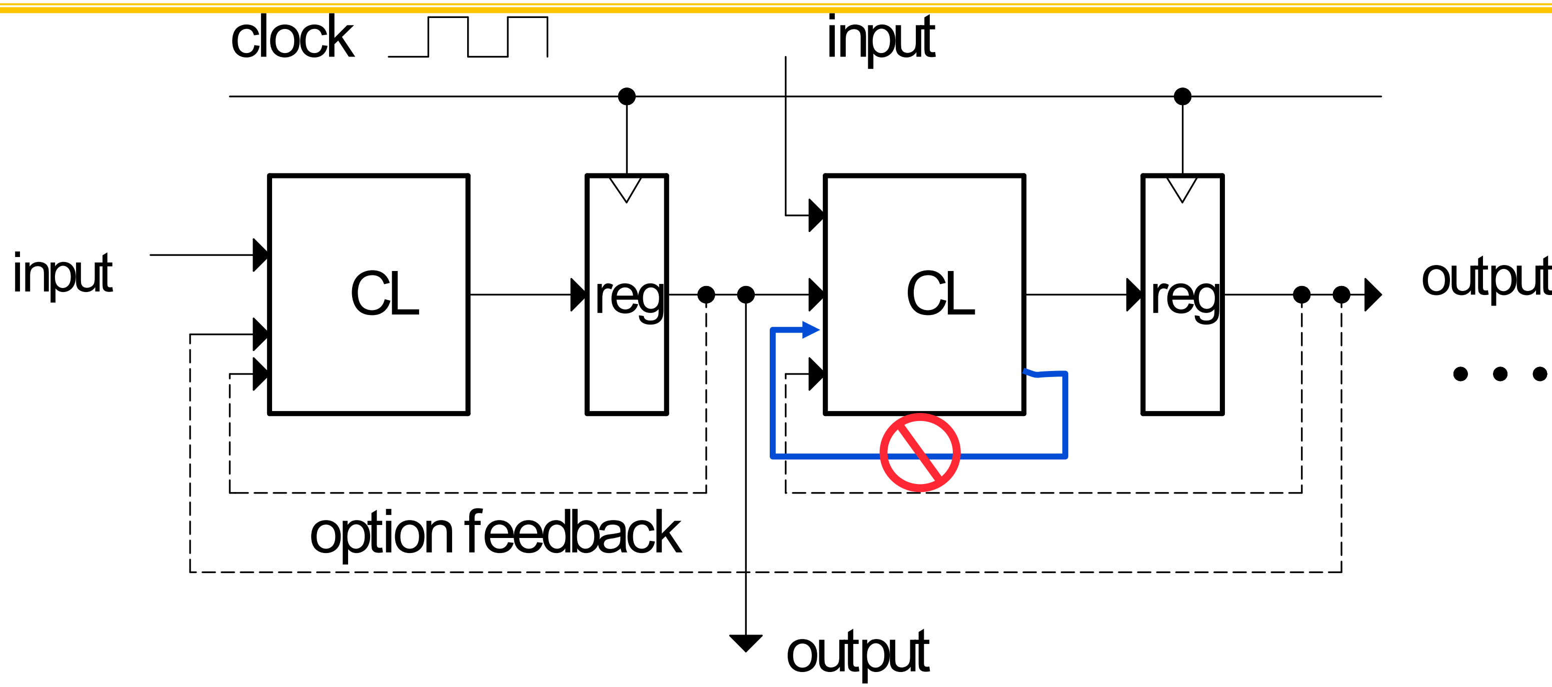


- **Logical change is not instantaneous**
- **Broader digital design methodology has to make it appears as such**
  - **Clocking, delay estimation, glitch avoidance**

What makes Digital Systems tick?



Synchronous Circuit Design

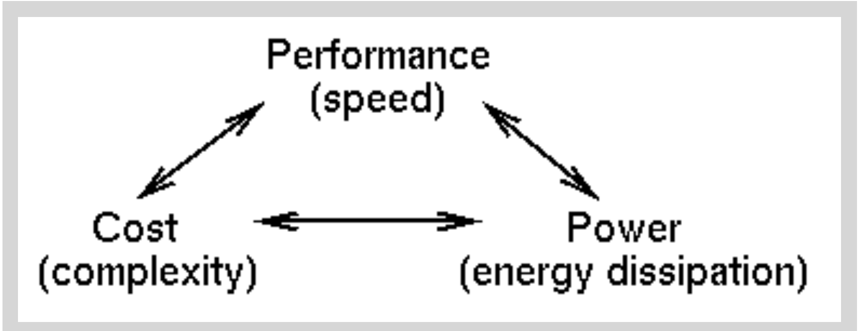


- **Combinational Logic Blocks (CL)**
  - **Acyclic**
  - **no internal state (no feedback)**
  - **output only a function of inputs**
- **Registers (reg)**
  - **collections of flip-flops**
- **clock**
  - **distributed to all flip-flops**
- **ALL CYCLES GO THROUGH A REG!**

Modern Hardware Design

- Extremely Software Intensive
  - Design tools (schematic capture, hardware description lang.)
  - Simulation tools
  - Optimization tools
  - Verification tools
  - Supply chain and project management
- Managing complexity of fundamental
  - Modularity
  - Methodology
  - Clarity
  - Technology independence
- Push the edge
  - Of the available tools
  - Of the technology

Basic Design Tradeoffs



- **You can usually improve on one at the expense of one or both of the others.**
- **These tradeoffs exist at every level in the system design - every sub-piece and component.**
- **Design Specification -**
  - **Functional Description.**
  - **Performance, cost, power constraints.**
- **As a designer you must make the tradeoffs necessary to achieve the function within the constraints.**