UC Berkeley
Teaching Professor
Dan Garcia

CS61C

Great Ideas in Computer Architecture

(a.k.a. Machine Structures)

UC Berkeley
Professor
Bora Nikolić

Introduction to Synchronous Digital Systems (SDS): Switches, Transistors, Signals & Waveforms

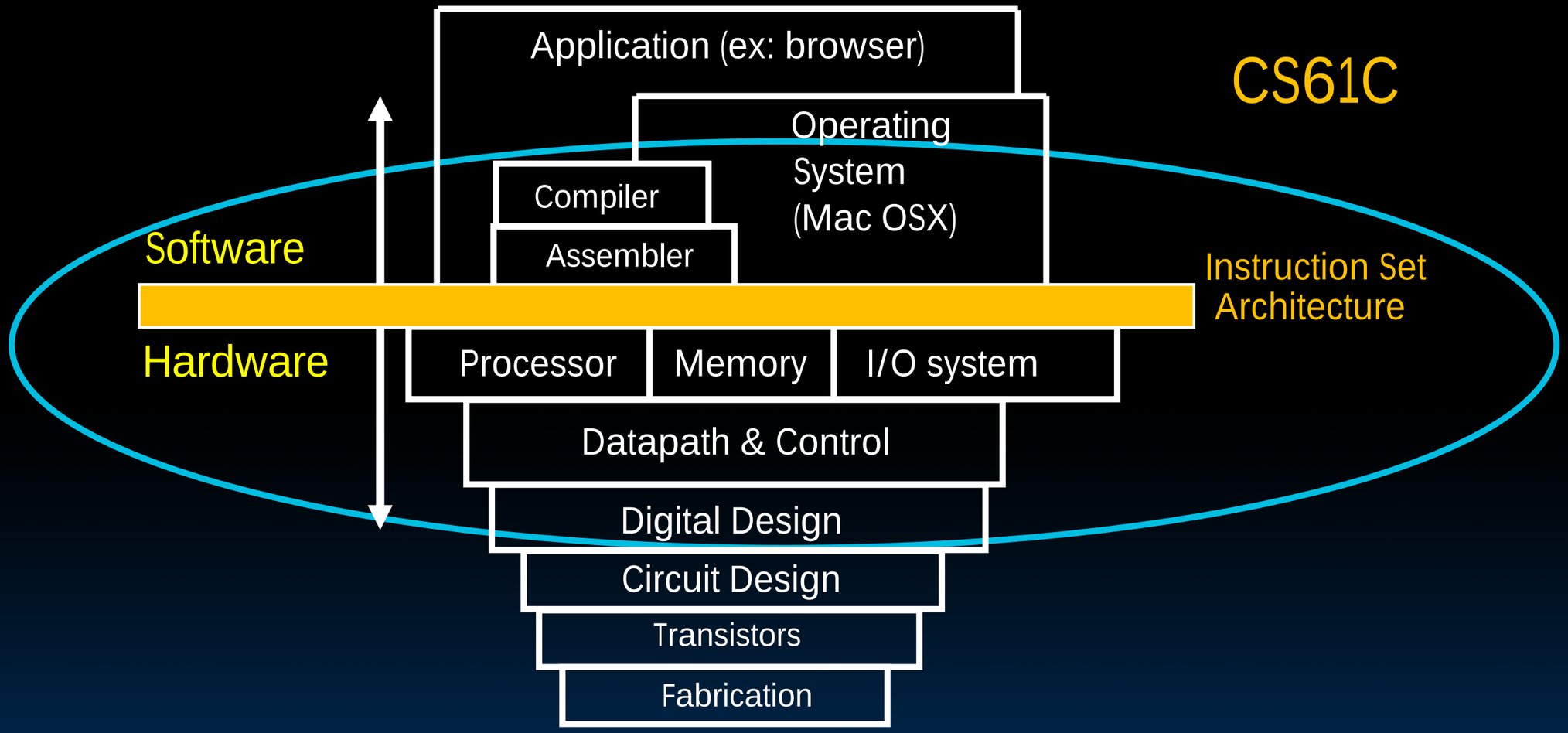
Berkeley
UNIVERSITY OF CALIFORNIA

cs61c.org

Garcia, Nikolić

Switches

Machine Structures



New-School Machine Structures

Software

Parallel Requests
Assigned to computer
e.g., Search “Cats”

Parallel Threads
Assigned to core e.g., Lookup, Ads

Parallel Instructions
>1 instruction @ one time
e.g., 5 pipelined instructions

Parallel Data
>1 data item @ one time
e.g., Add of 4 pairs of words

Hardware descriptions
All gates work in parallel at same time

Harness Parallelism & Achieve High Performance

Hardware

Warehouse Scale Computer

Smart Phone

Great Idea #1: Abstraction (Levels of Representation/Interpretation)

High Level Language Program (e.g., C)

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

Compiler

Assembly Language Program (e.g., RISC-V)

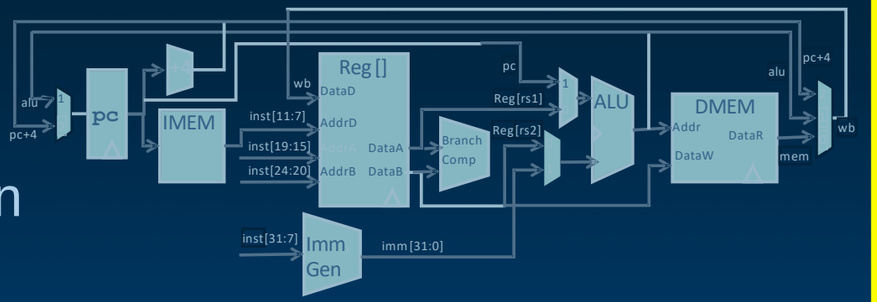
```
lw    x3, 0(x10)
lw    x4, 4(x10)
sw    x4, 0(x10)
sw    x3, 4(x10)
```

Assembler

Machine Language Program (RISC-V)

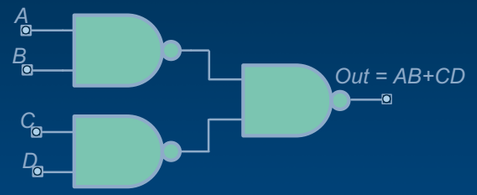
```
1000 1101 1110 0010 0000 0000 0000 0000
1000 1110 0001 0000 0000 0000 0000 0100
1010 1110 0001 0010 0000 0000 0000 0000
1010 1101 1110 0010 0000 0000 0000 0100
```

Hardware Architecture Description (e.g., block diagrams)



Architecture Implementation

Logic Circuit Description (Circuit Schematic Diagrams)



Synchronous Digital Systems

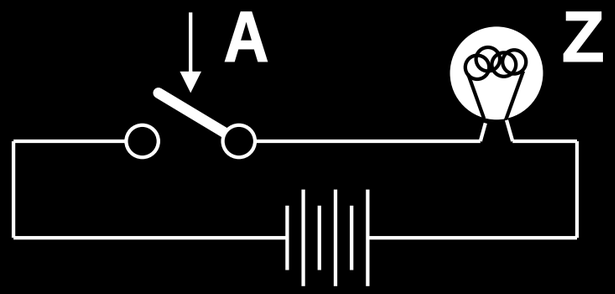
- **Hardware of a processor, e.g., RISC-V, is a Synchronous Digital System**
- **Synchronous:**
  - All operations coordinated by a central clock
  - "Heartbeat" of the system!
- **Digital:**
  - All values represented by discrete values
  - Electrical signals are treated as 1s and 0s; grouped together to form words

Logic Design

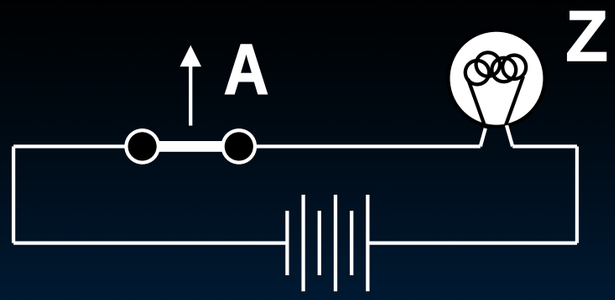
- **Next several weeks: we'll study how a modern processor is built; starting with basic elements as building blocks**
- **Why study hardware design?**
  - Understand capabilities and limitations of hw in general and processors in particular
  - What processors can do fast and what they can't do fast (avoid slow things if you want your code to run fast!)
  - Background for more in depth HW courses (150, 152)
  - There is just so much you can do with standard processors: you may need to design own custom hw for extra performance

Switches: Basic Element of Physical Circuit

- **Implementing a simple circuit**
  - Close switch when A is 1, open when A is 0



Close switch (if A is "1" or asserted) and turn on light bulb (Z)

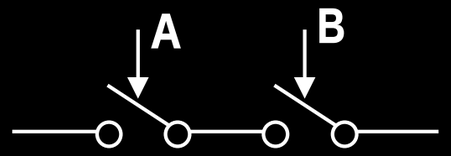


Open switch (if A is "0" or unasserted) and turn off light bulb (Z)

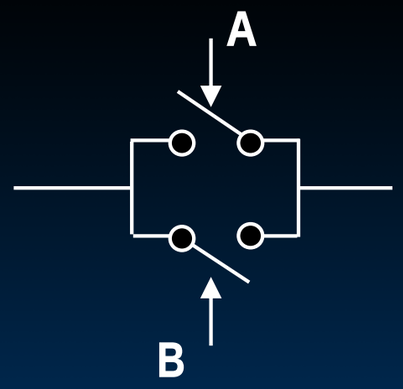
$$Z \equiv A$$

Switches (continued)

- Compose switches into more complex ones (Boolean functions):

AND 

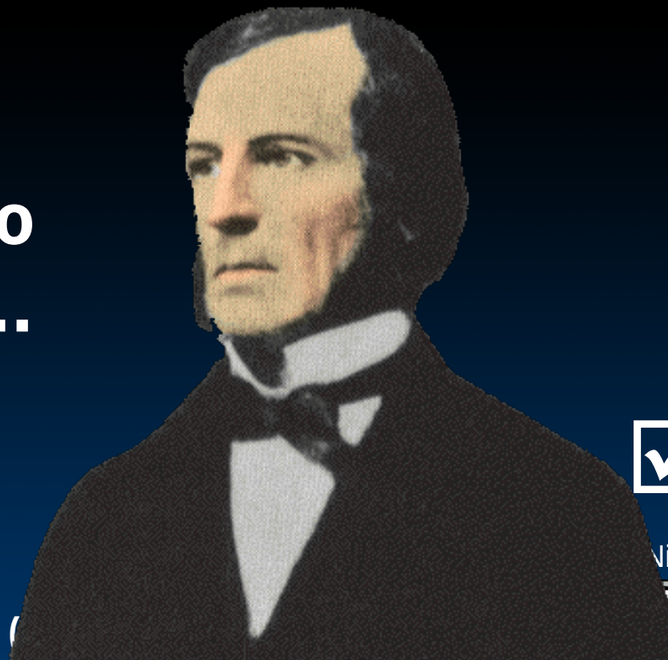
Z ≡ A and B

OR 

Z ≡ A or B

Historical Note

- Early computer designers built ad hoc circuits from switches
- Began to notice common patterns in their work: ANDs, ORs, …
- Master's thesis (by Claude Shannon) made link between transistors and 19th Century Mathematician George Boole
  - Called it "Boolean" in his honor
- Could apply math to give theory to hardware design, minimization, …

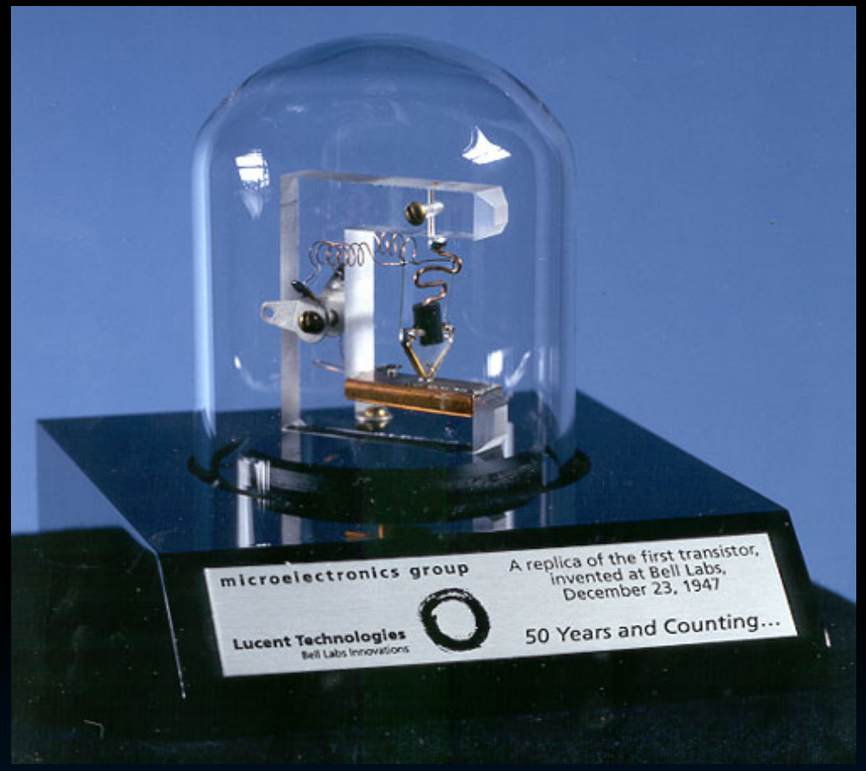


Transistors

The Transistor ("born" 1947-12-23)

en.wikipedia.org/wiki/History_of_the_transistor
www.pbs.org/transistor
youtu.be/-td7YT-Pums
youtu.be/OwS9aTE2Go4

- **Semiconductor device to amplify or switch signals**
  - Key component in ALL modern electronics
- **Who?**
  - John Bardeen, William Shockley, Walter Brattain
- **Before that?**
  - Vacuum Tubes
- **After that?**
  - Integrated circuit, microprocessor



"The Transistor was probably THE most important invention of the 20th Century"
- Ira Flatow, Transistorized! (PBS Special)

Transistor Networks

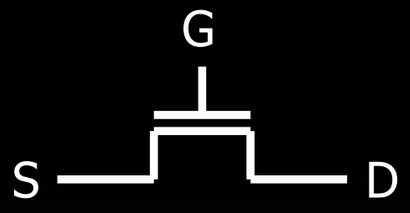
- **Modern digital systems designed in CMOS**
  - MOS: Metal-Oxide on Semiconductor
  - C for complementary: normally-open and normally-closed switches

- **MOS transistors act as voltage-controlled switches**

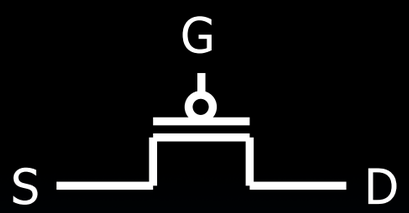
MOS Transistors

- **Three terminals: Drain, Gate, Source**
  - Switch action: Dan Garcia Says
    if voltage on gate terminal is (some amount) higher/lower than source terminal then conducting path established between drain and source terminals

To remember:
n ("normal")
p (has a circle, like the top part of P itself)



n-channel
open when voltage at G is low
closes when:
voltage(G) > voltage (S) + ε



p-channel
closed when voltage at G is low
opens when:
voltage(G) > voltage (S) + ε

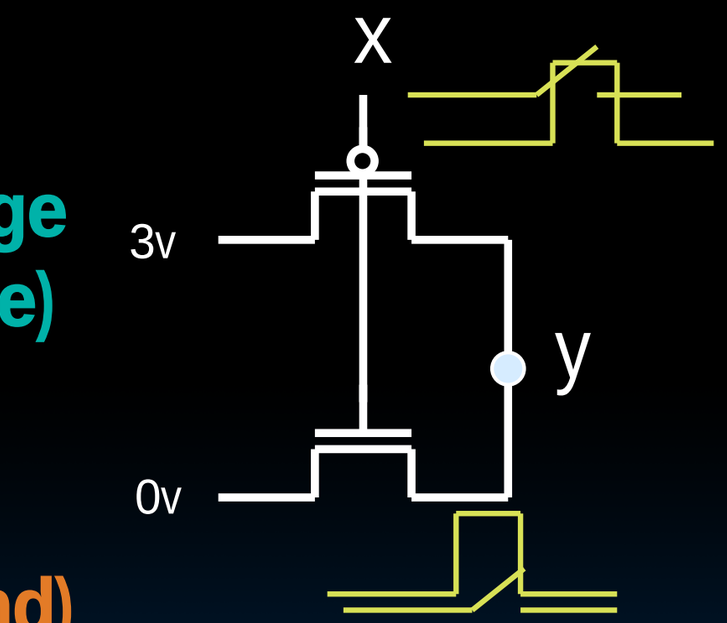
G LOW

G HIGH

MOS Networks

“1” (voltage source)

“0” (ground)

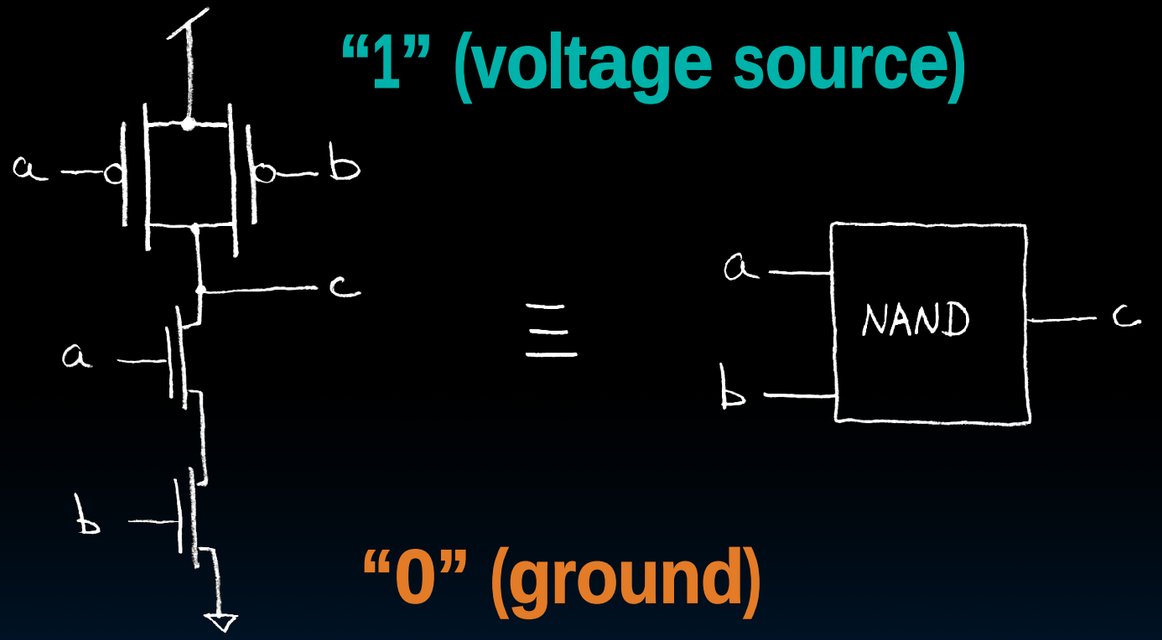


what is the relationship between x and y ?

x	y
0 volts	3 volts
3 volts	0 volts

Transistor Circuit Rep. vs. Block diagram

- Chips are composed of nothing but transistors and wires.
- Small groups of transistors form useful building blocks.

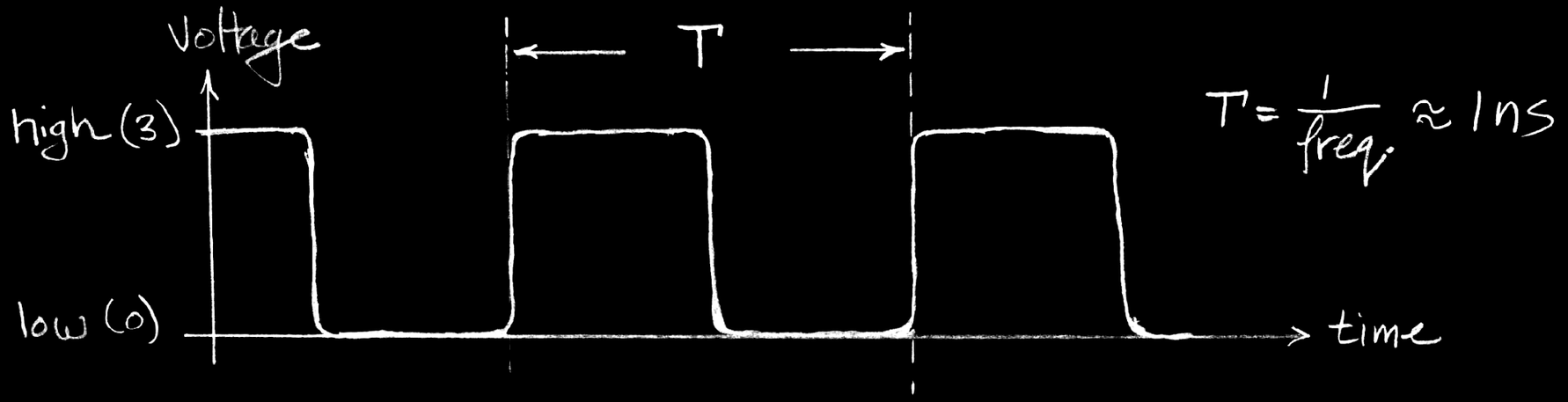


a	b	c
0	0	1
0	1	1
1	0	1
1	1	0

- Block are organized in a hierarchy to build higher-level blocks: ex: adders.
- You can build AND, OR, NOT out of NAND!

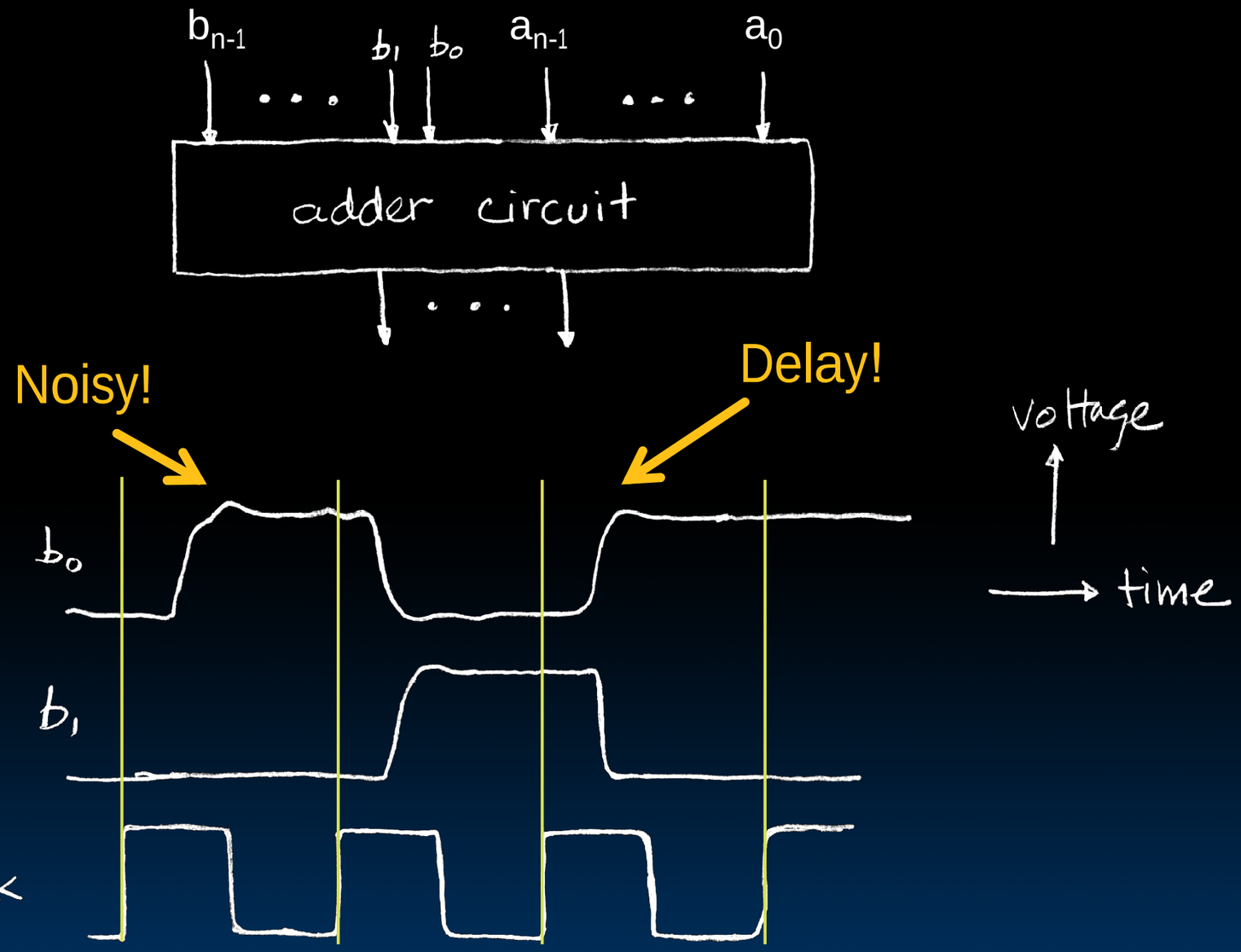
Signals and Waveforms

Signals and Waveforms: Clocks

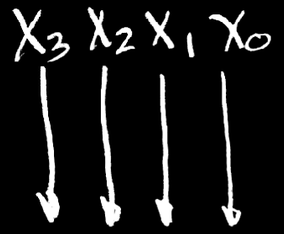


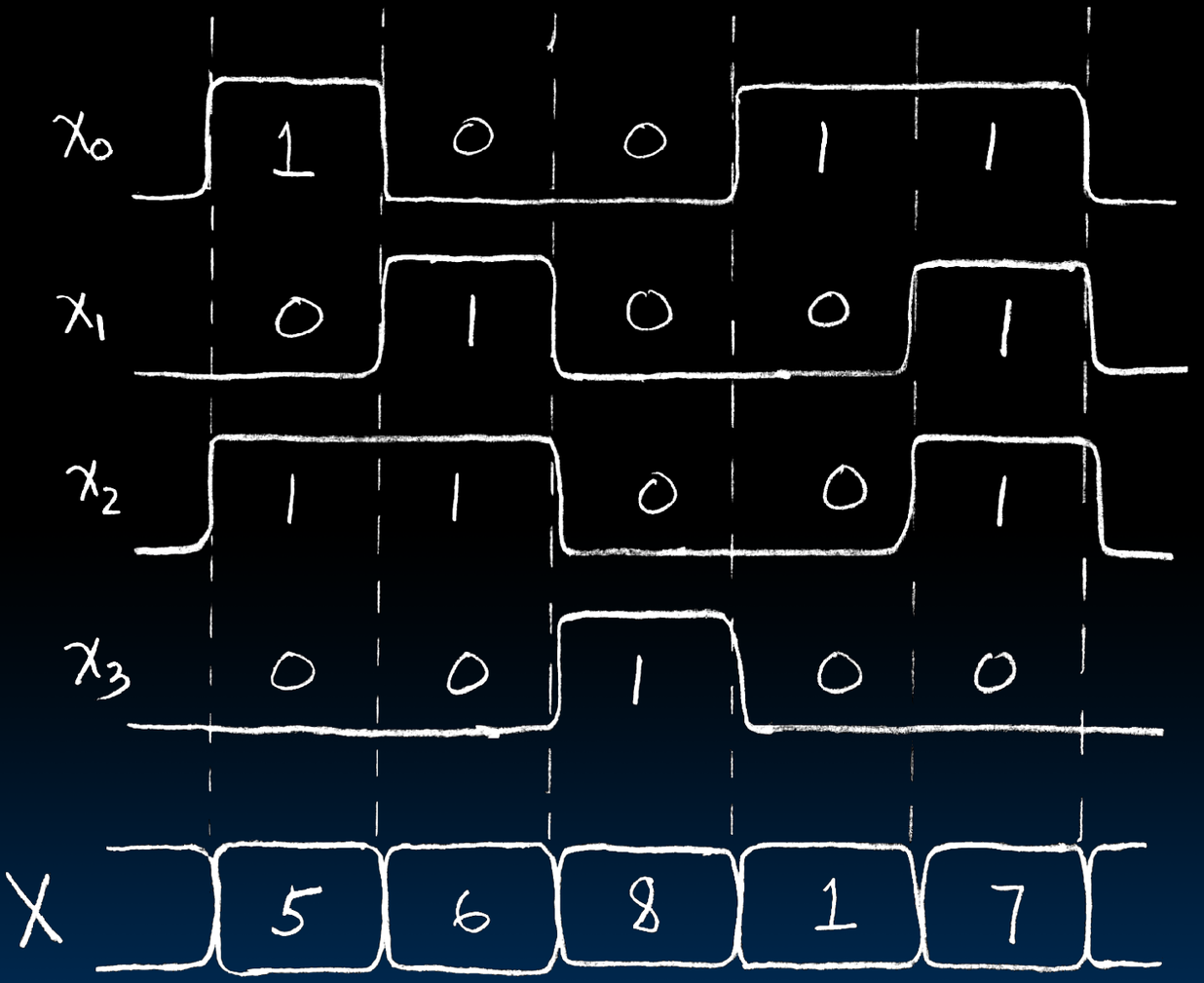
- **Signals**
  - When digital is only treated as 1 or 0
  - Is transmitted over wires continuously
  - Transmission is effectively instant
  - Implies that a wire contains 1 value at a time

Signals and Waveforms

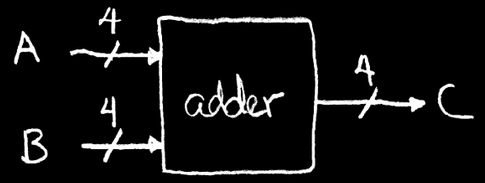


Signals and Waveforms: Grouping



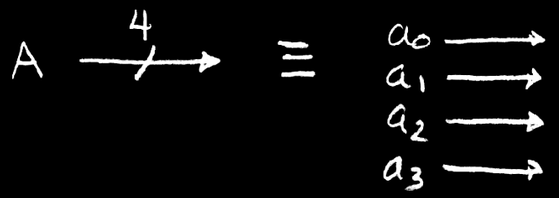


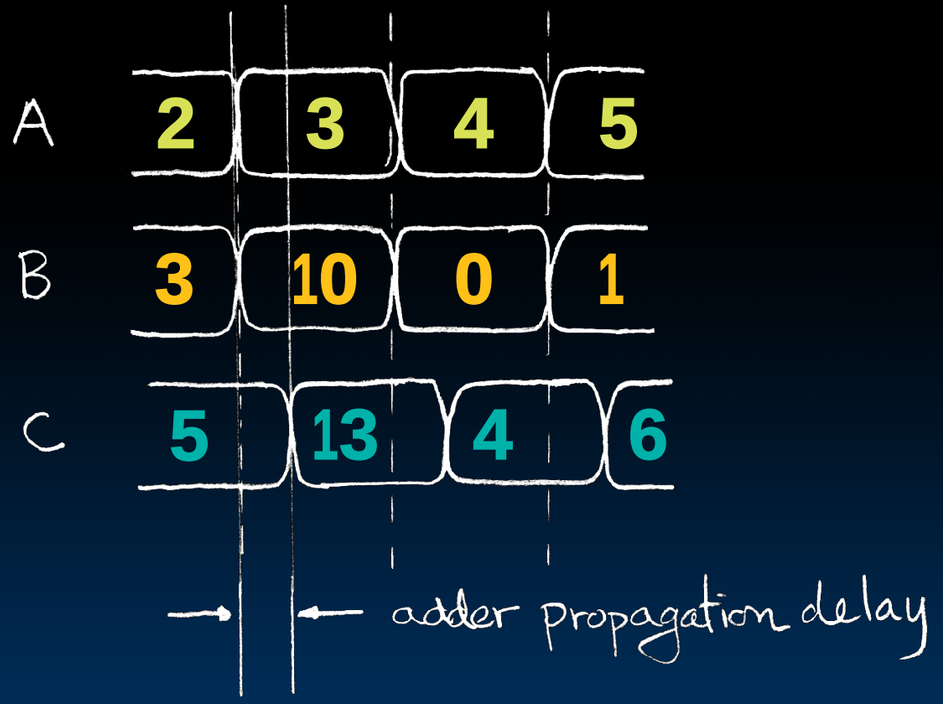
Signals and Waveforms: Circuit Delay



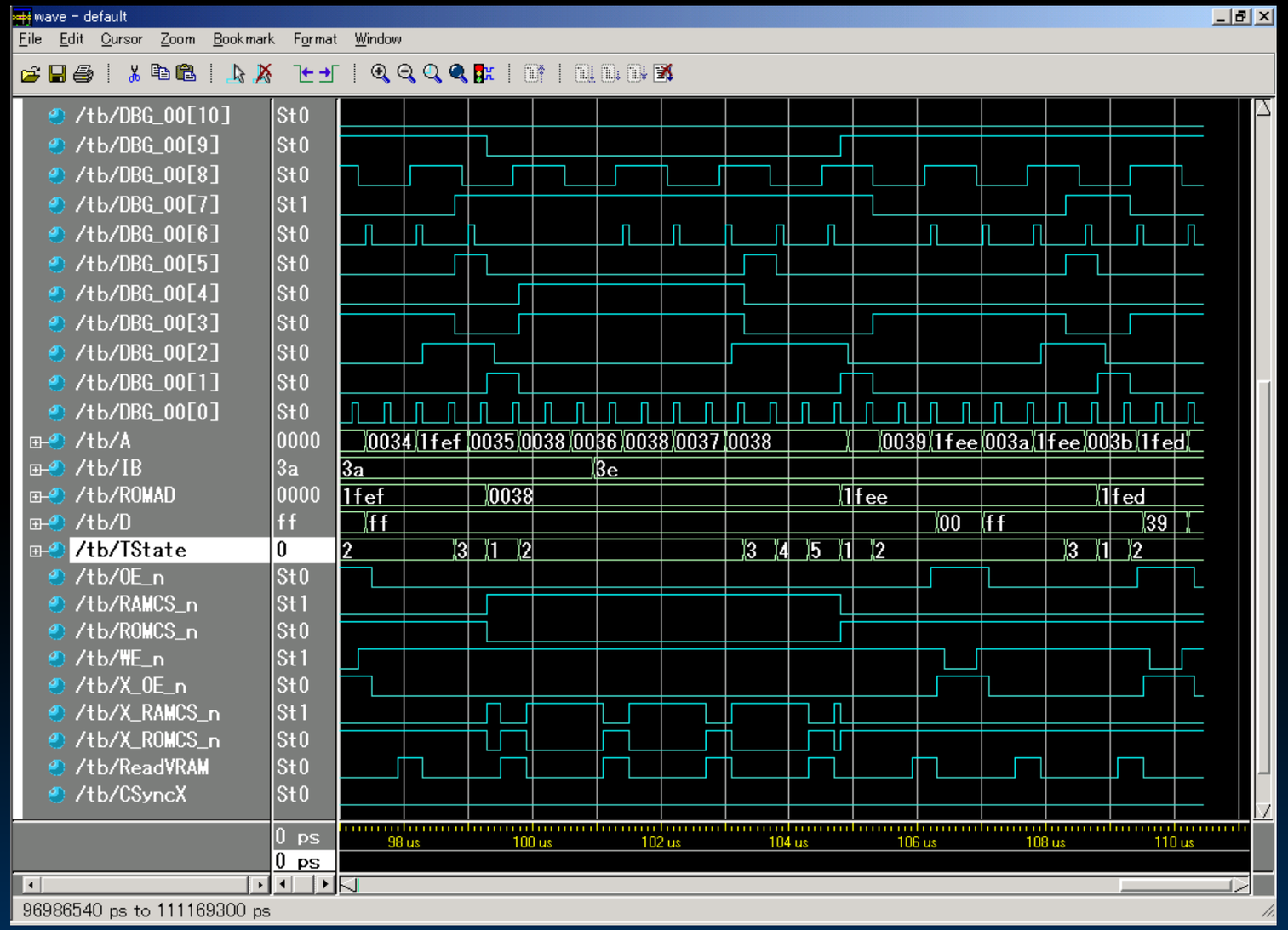
$A = [a_3, a_2, a_1, a_0]$

$B = [b_3, b_2, b_1, b_0]$





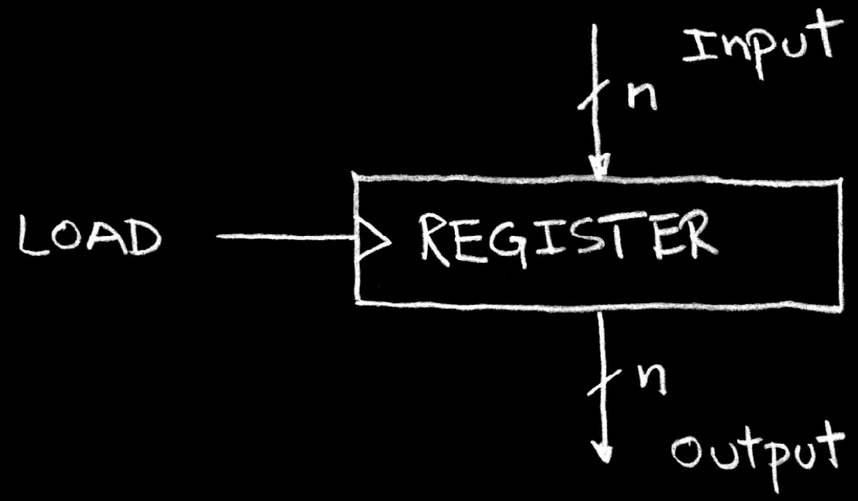
Sample Debugging Waveform



Type of Circuits

- **Synchronous Digital Systems are made up of two basic types of circuits:**
- **Combinational Logic (CL) circuits**
  - Our previous adder circuit is an example.
  - Output is a function of the inputs only.
  - Similar to a pure function in mathematics, y = f(x). (No way to store information from one invocation to the next. No side effects)
- **State Elements**
  - circuits that store information.

Circuits with STATE (e.g., register)



And in conclusion…

- Clocks control pulse of our circuits
- Voltages are analog, quantized to 0/1
- Circuit delays are fact of life
- Two types of circuits:
  - Stateless Combinational Logic (&,|,~)
  - State circuits (e.g., registers)

UC Berkeley
Teaching Professor
Dan Garcia

CS61C

Great Ideas in Computer Architecture

(a.k.a. Machine Structures)

UC Berkeley
Professor
Bora Nikolić

Introduction to Synchronous Digital Systems (SDS) State

Berkeley
UNIVERSITY OF CALIFORNIA

cs61c.org

Garcia, Nikolić

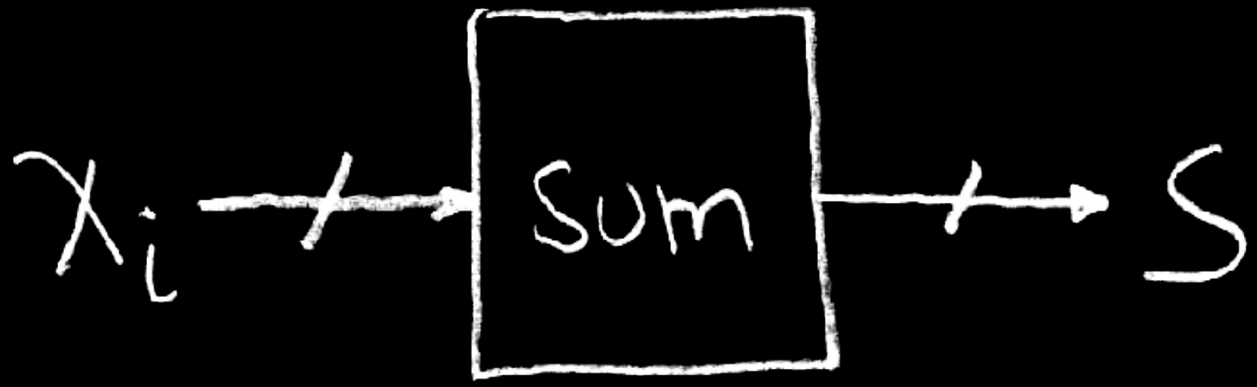
Accumulator

Uses for State Elements

- **As a place to store values for some indeterminate amount of time:**
  - Register files (like x0-x31 on the RISC-V)
  - Memory (caches, and main memory)
- **Help control the flow of information between combinational logic blocks.**
  - State elements are used to hold up the movement of information at the inputs to combinational logic blocks and allow for orderly passage.

Accumulator Example

- **Why do we need to control the flow of information?**

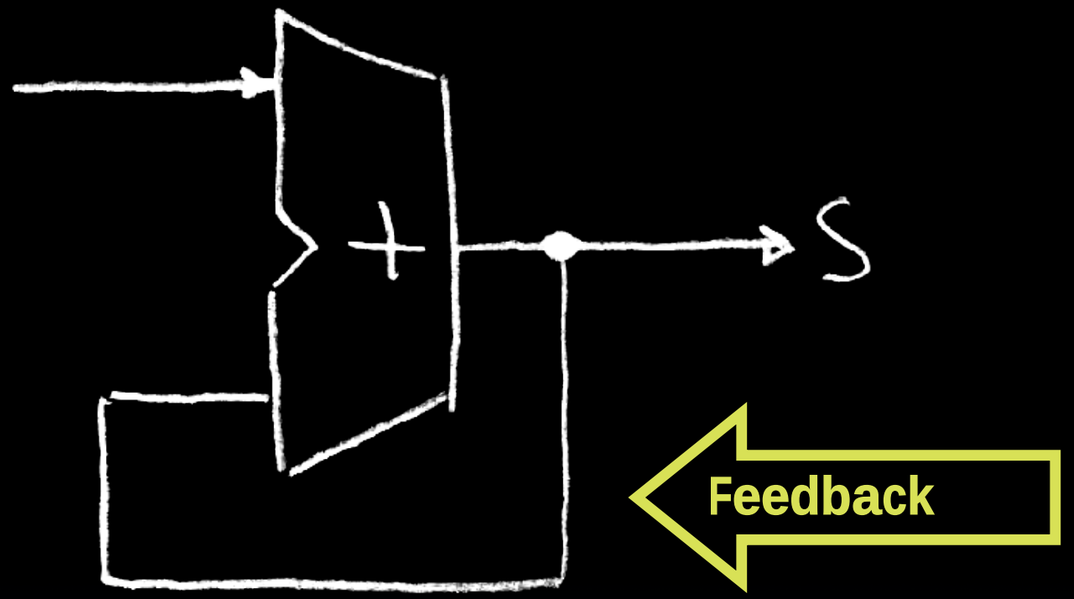


- **Want:**

```
S=0;
for (i=0;i<n;i++)
   S = S + Xi
```

- **Assume**
  - Each **X** value is applied in succession, one per cycle.
  - After **n** cycles the sum is present on **S**.

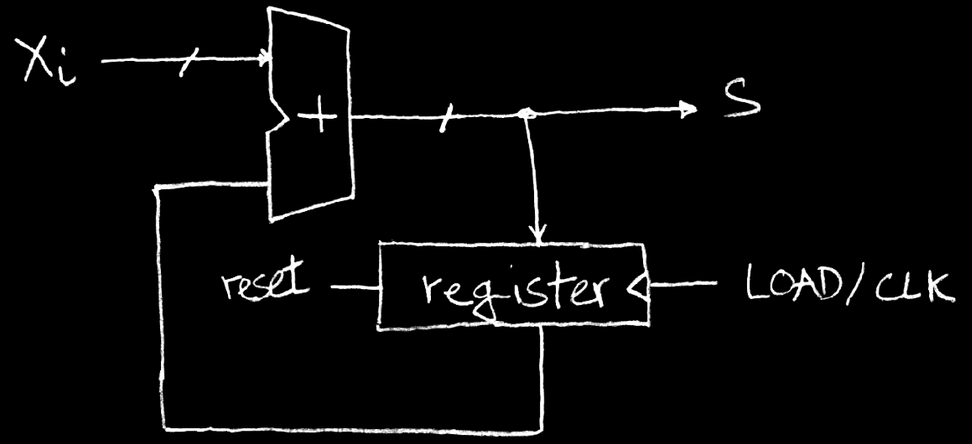
First try…Does this work?



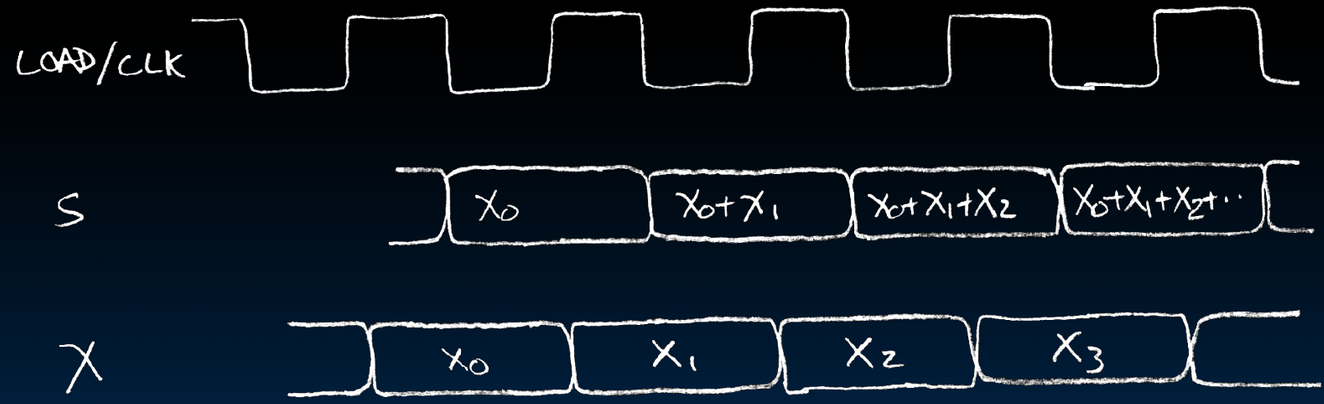
- **Nope!**
  - Reason #1… What is there to control the next iteration of the '`for`' loop?
  - Reason #2… How do we say: '`S=0`'?

Second try…How about this?

- Register is used to hold up the transfer of data to adder.

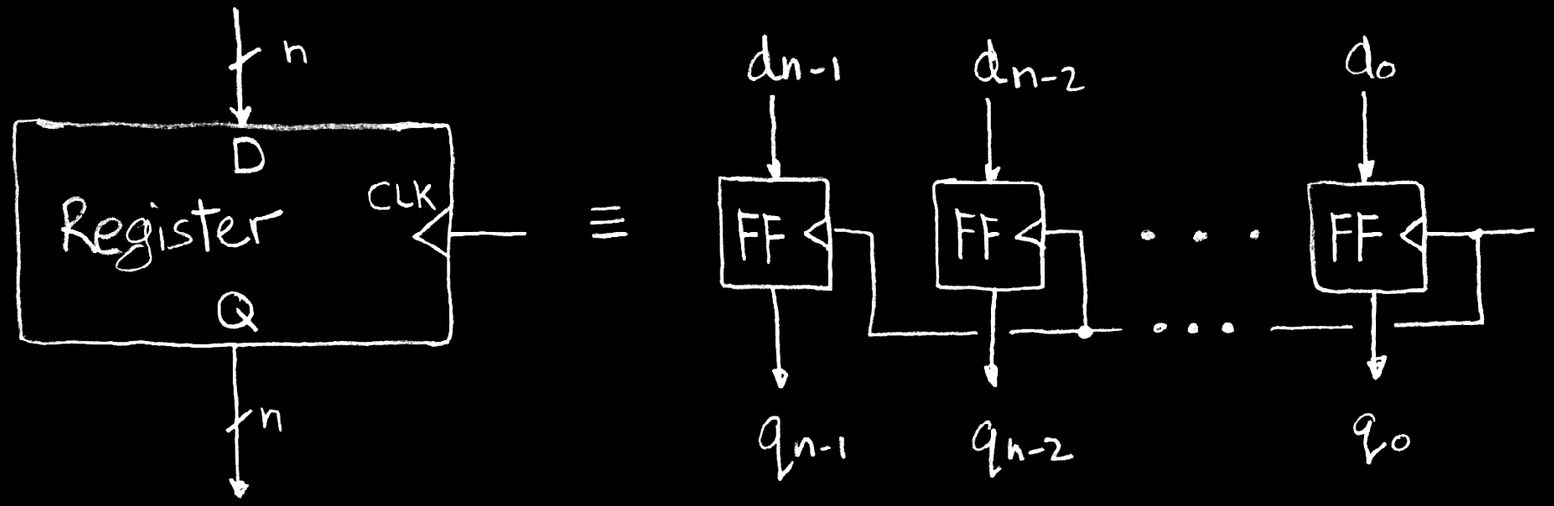


Rough timing…



Register Details
Flip-flops

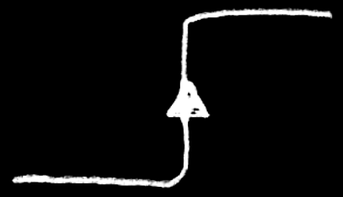
Register Details…What's inside?



- n instances of a "Flip-Flop"
- Flip-flop name because the output flips and flops between and 0,1
- D is "data", Q is "output"
- Also called "D-type Flip-Flop"
  - There used to be other types of flip-flops

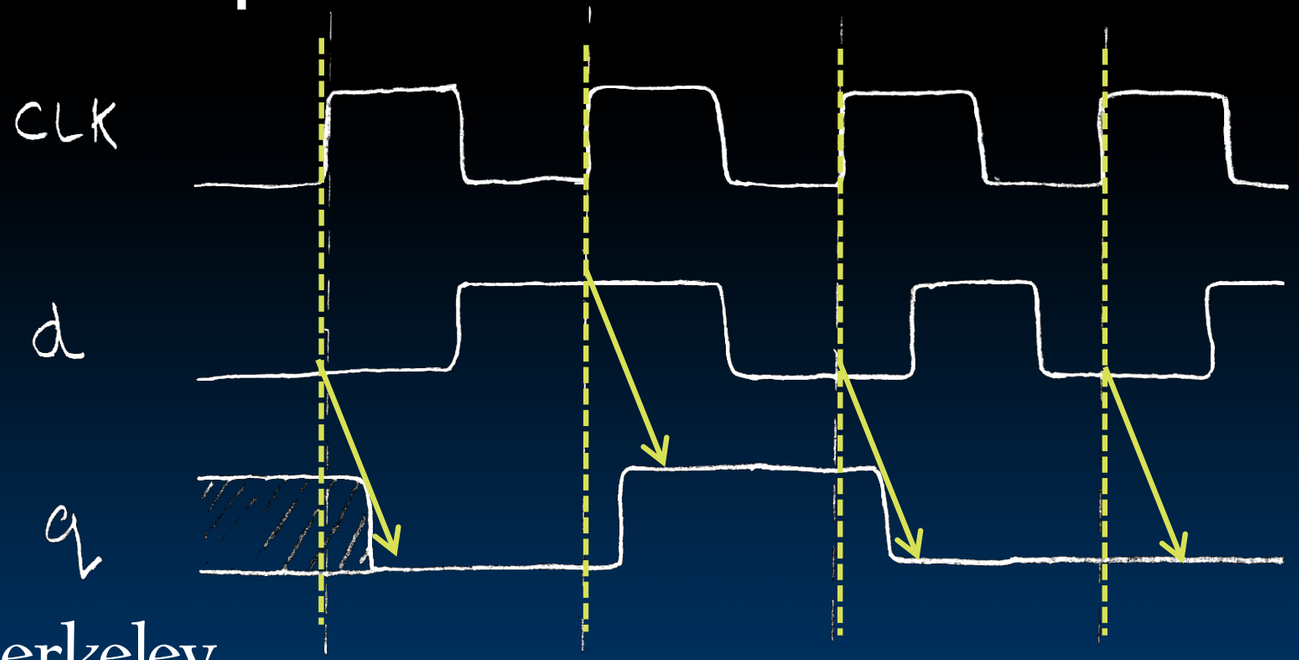
What's the timing of a Flip-flop? (1/2)

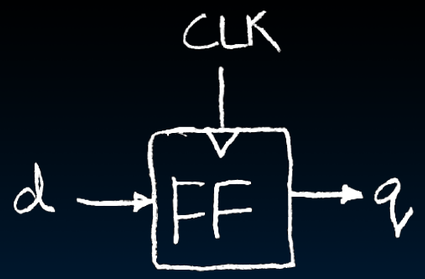
- **Edge-triggered d-type flip-flop**
  - This one is "rising edge-triggered"
  - Also called "positive edge"



There also exist "falling edge" FFs

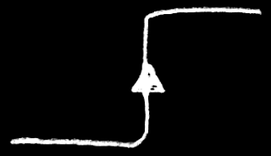
- **"On the rising edge of the clock, the input d is sampled and transferred to the output. At all other times, the input d is ignored."**
- **Example waveforms:**



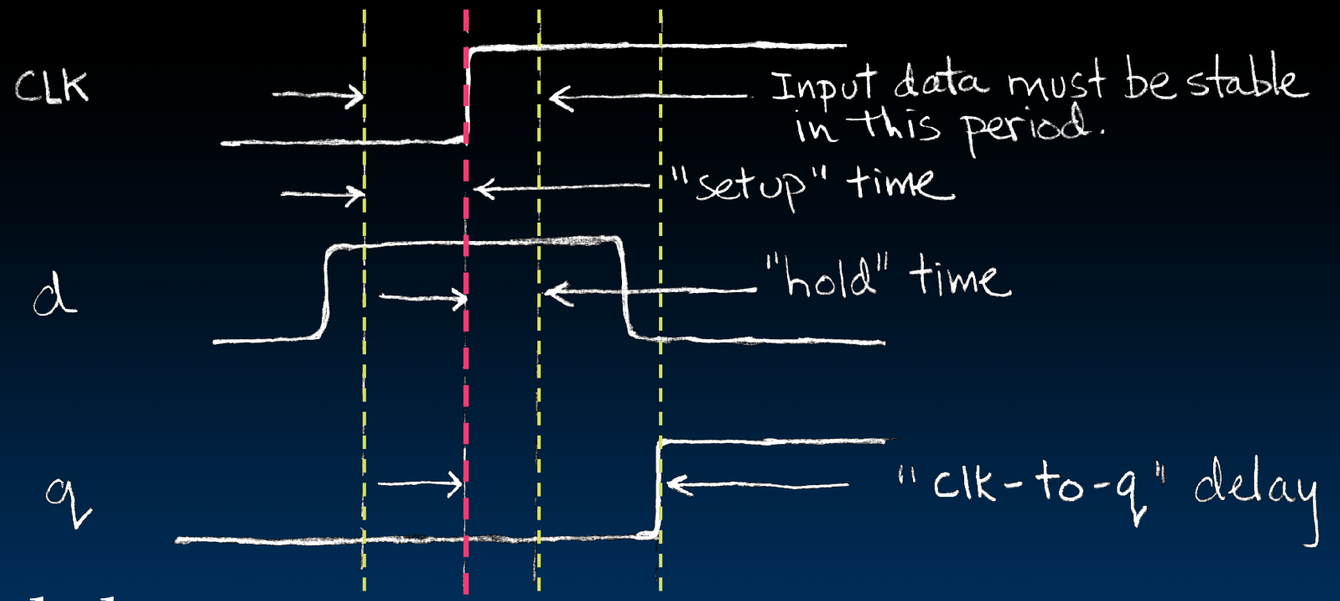


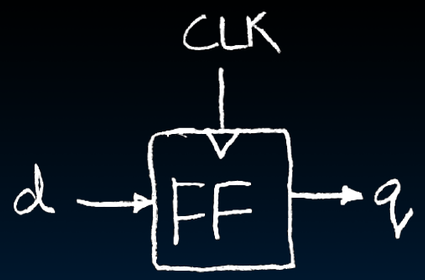
What's the timing of a Flip-flop? (2/2)

- **Edge-triggered d-type flip-flop**
  - This one is "rising edge-triggered"



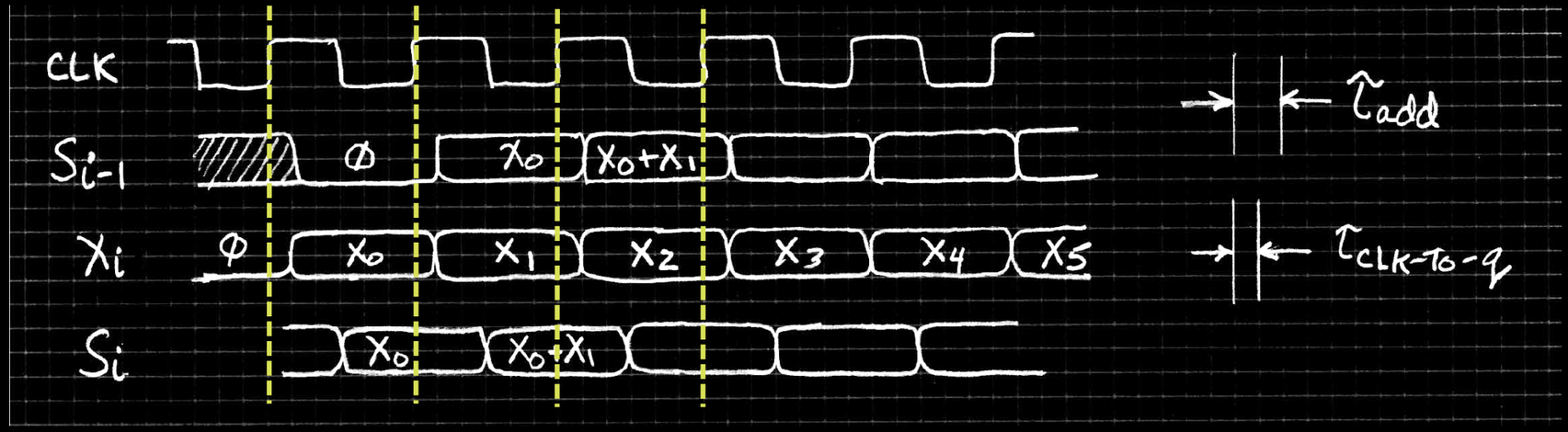
- **"On the rising edge of the clock, the input d is sampled and transferred to the output. At all other times, the input d is ignored."**
- **Example waveforms (more detail):**

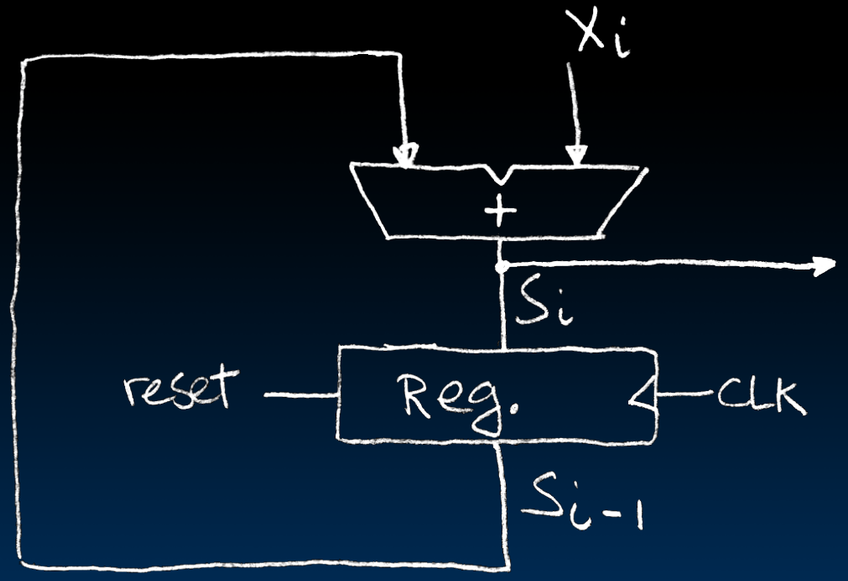




Accumulator Revisited

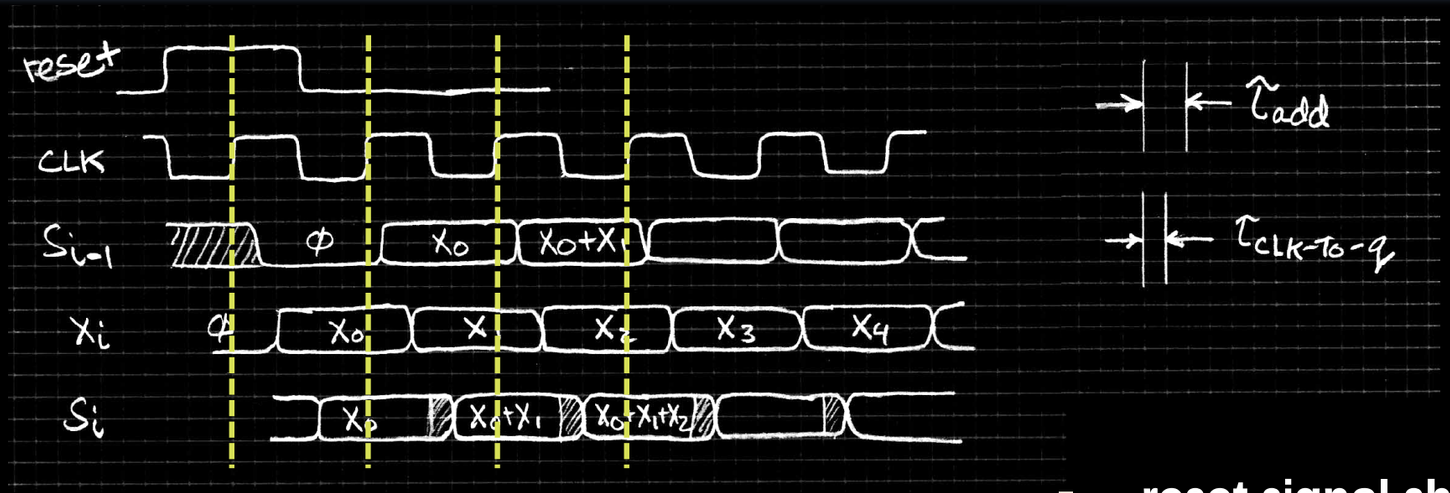
Accumulator Revisited (proper timing 1/2)

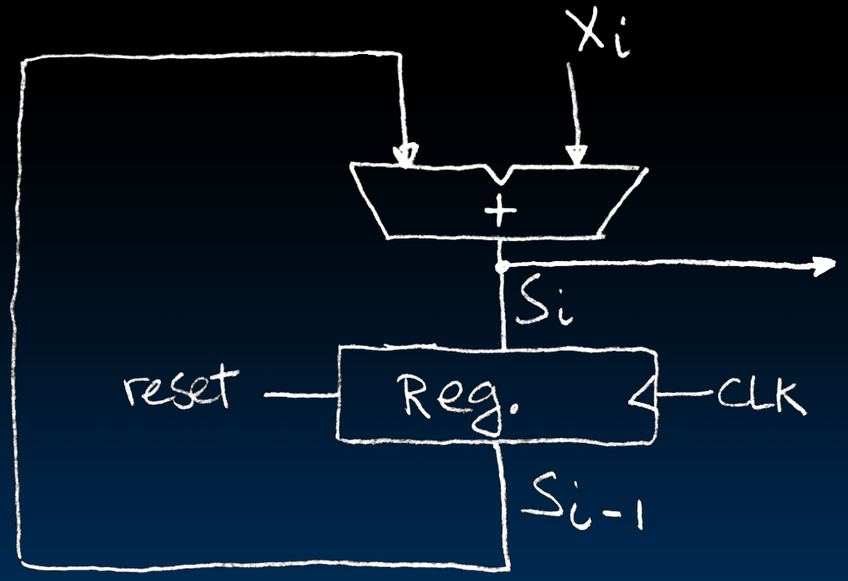




- Reset input to register is used to force it to all zeros (takes priority over D input).
- S_{i-1} holds the result of the i^{th}-1 iteration.
- Analyze circuit timing starting at the output of the register.

Accumulator Revisited (proper timing 2/2)



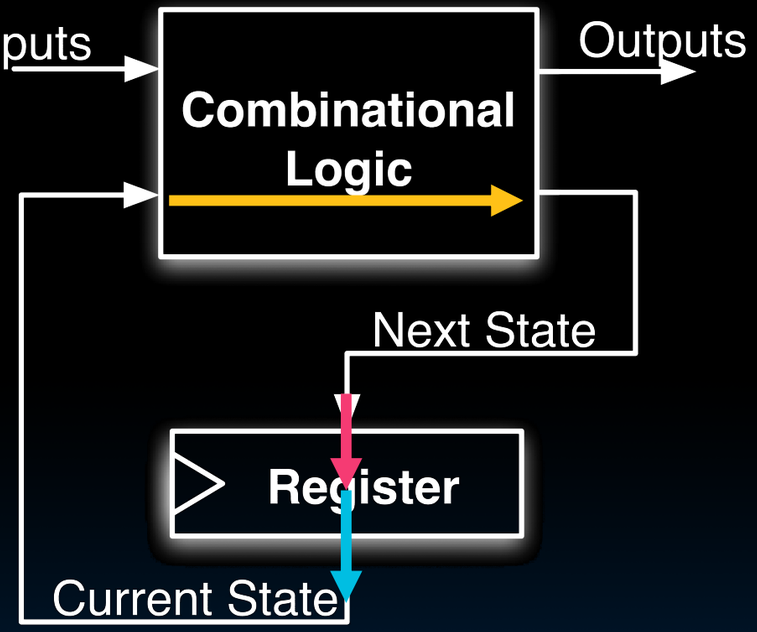


- reset signal shown.
- Also, in practice X might not arrive to the adder at the same time as S_{i-1}
- S_i temporarily is wrong, but register always captures correct value.
- In good circuits, instability never happens around rising edge of clk.

Pipelining for Performance

Maximum Clock Frequency

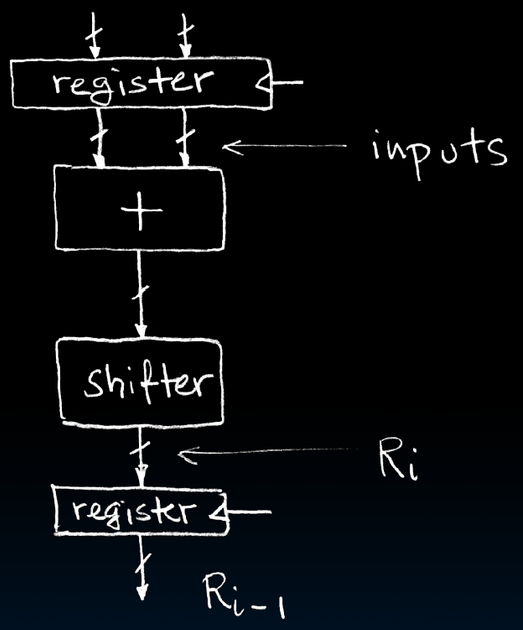
- What is the maximum clock frequency of this circuit? (Hint: Frequency = 1 / Period)



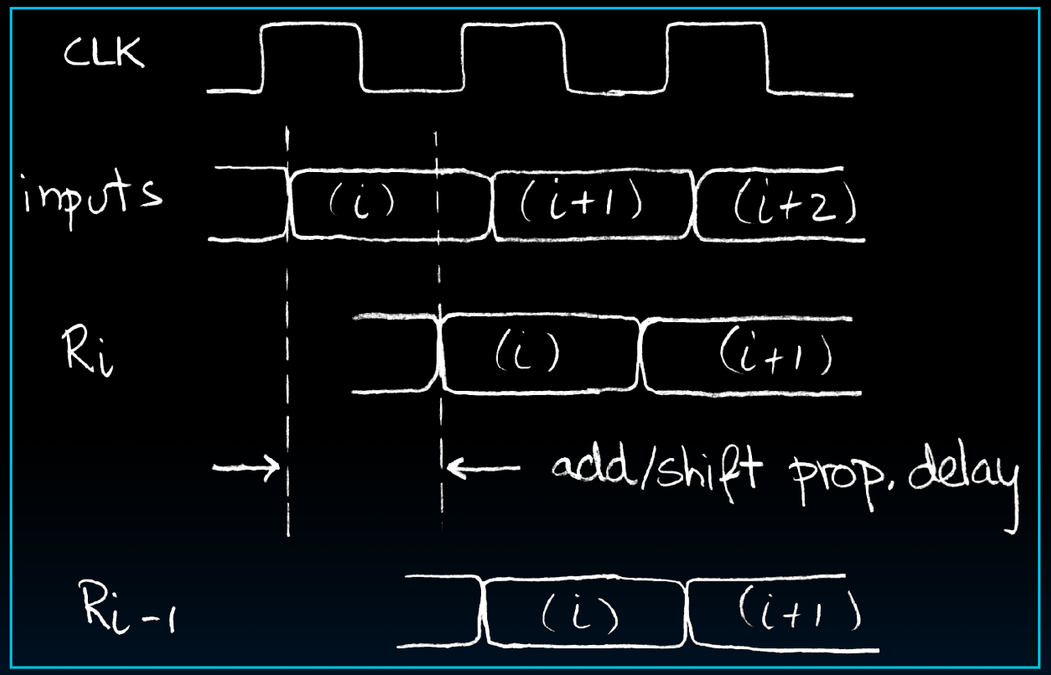
Max Delay = CLK-to-Q Delay + CL Delay + Setup Time

Pipelining to improve performance (1/2)

- **Extra Registers are often added to help speed up the clock rate.**



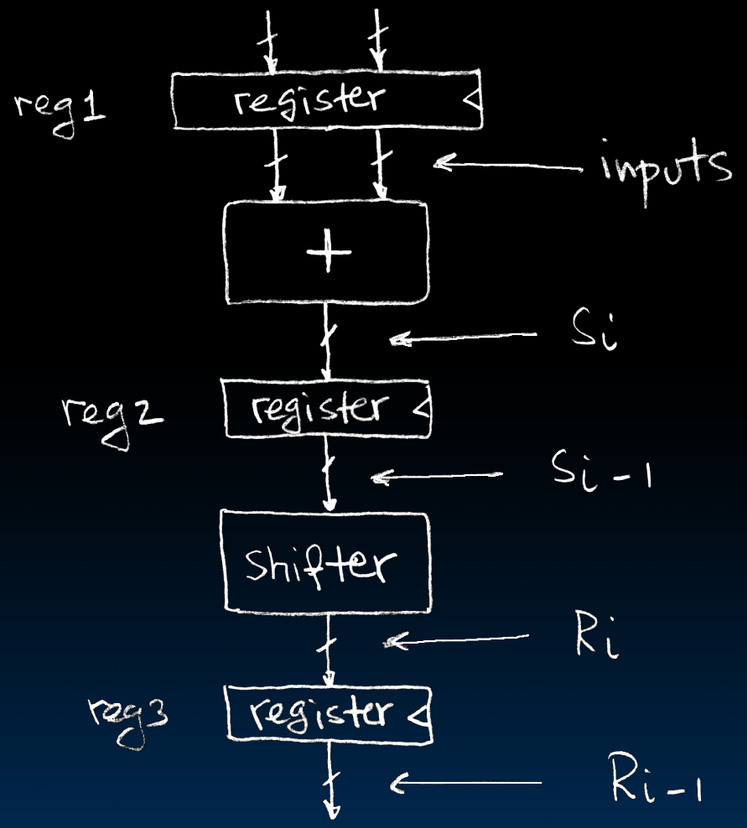
Timing…



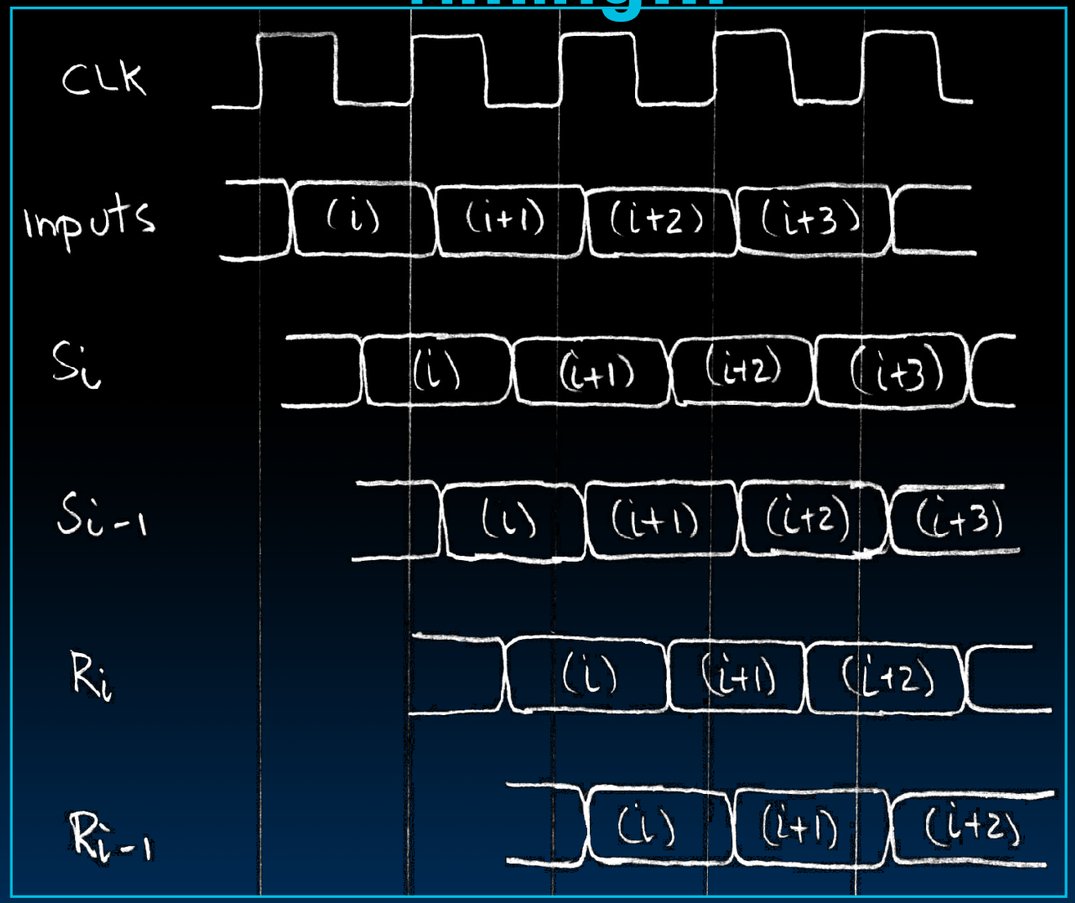
- **Note: Delay of 1 clock cycle from input to output.**
- **Clock period limited by propagation delay of adder/shifter.**

Pipelining to improve performance (2/2)

- Insertion of register allows higher clock frequency.
- More outputs per second.



Timing…



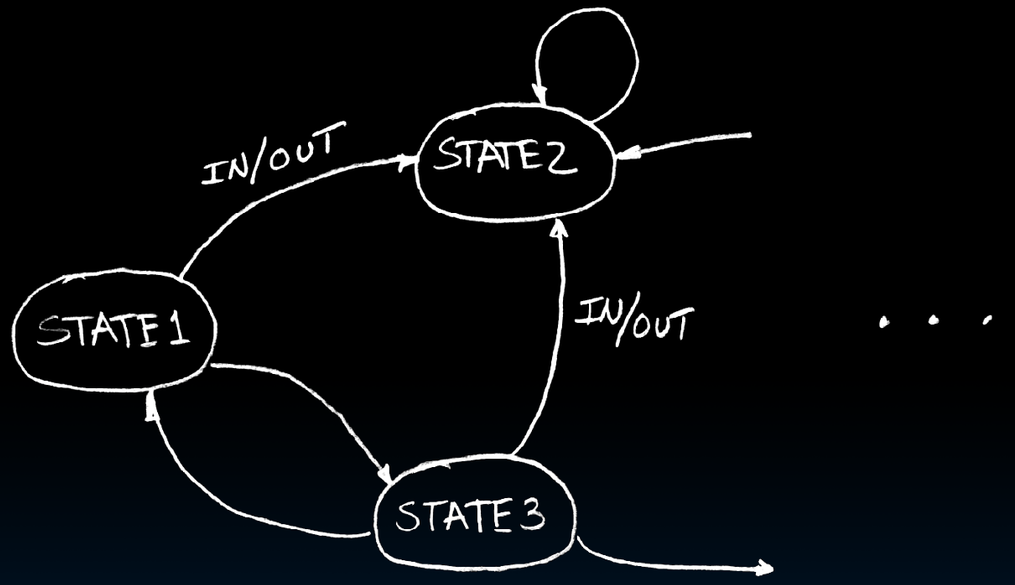
Recap of Timing Terms

- **Clock (CLK)** - steady square wave that synchronizes system
- **Setup Time** - when the input must be stable before the rising edge of the CLK
- **Hold Time** - when the input must be stable after the rising edge of the CLK
- **"CLK-to-Q" Delay** - how long it takes the output to change, measured from the rising edge of the CLK
- **Flip-flop** - one bit of state that samples every rising edge of the CLK (positive edge-triggered)
- **Register** - several bits of state that samples on rising edge of CLK or on LOAD (positive edge-triggered)

Finite State Machines

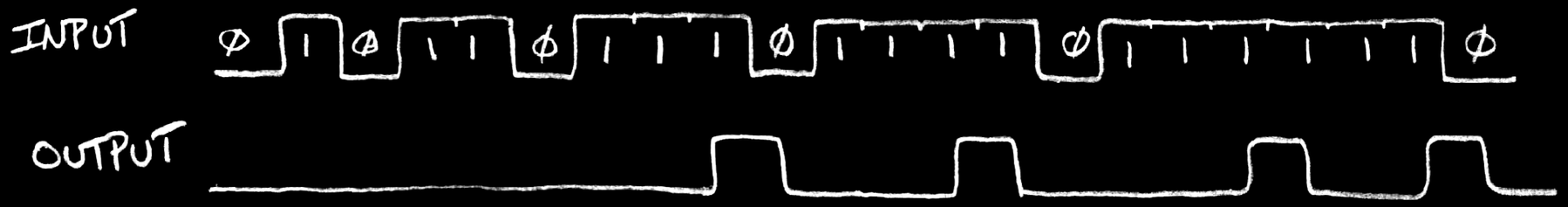
Finite State Machines (FSM) Introduction

- You have seen FSMs in other classes
  - Same basic idea
- The function can be represented with a "state transition diagram"
- With combinational logic and registers, any FSM can be implemented in hardware.

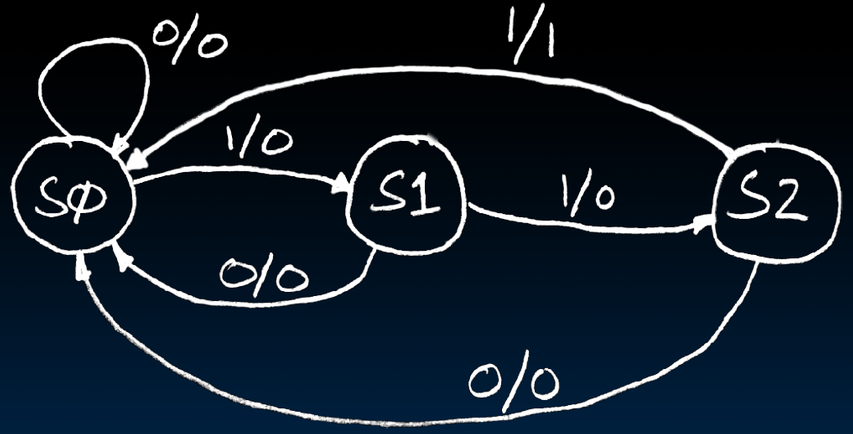


Finite State Machine Example: 3 ones…

- **FSM to detect the occurrence of 3 consecutive 1's in the input.**

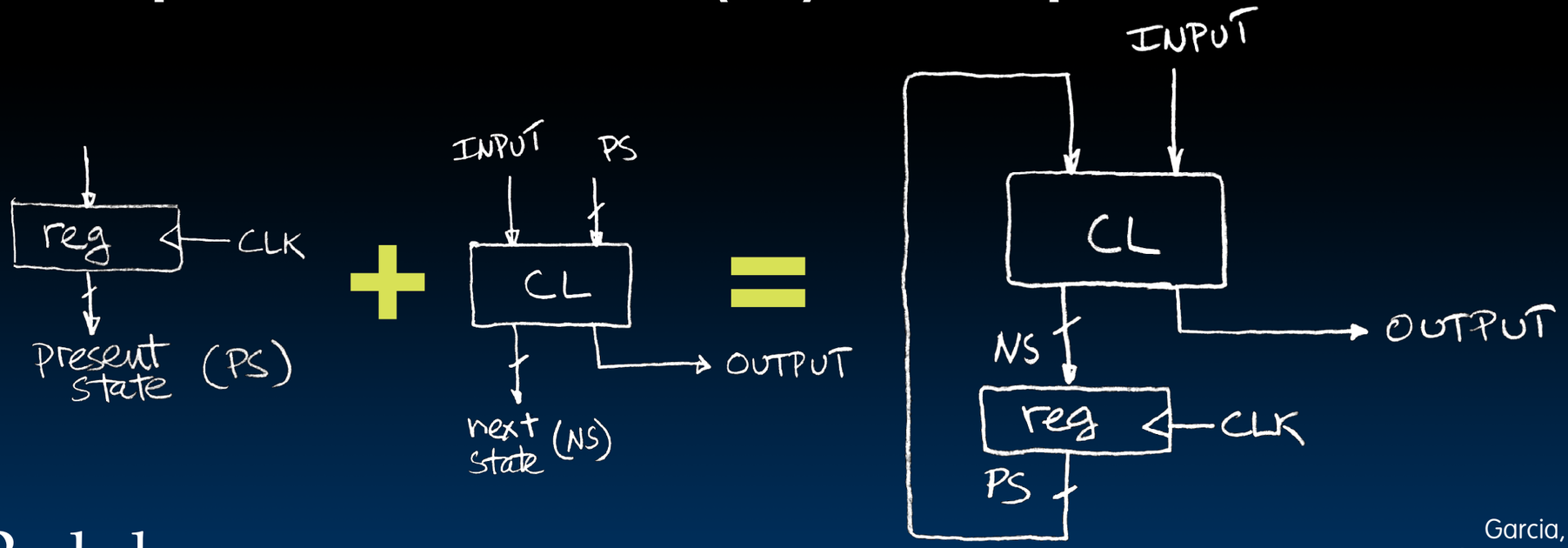


- **Draw the FSM…**
  - Assume state transitions are controlled by the clock: on each clock cycle the machine checks the inputs and moves to a new state and produces a new output…



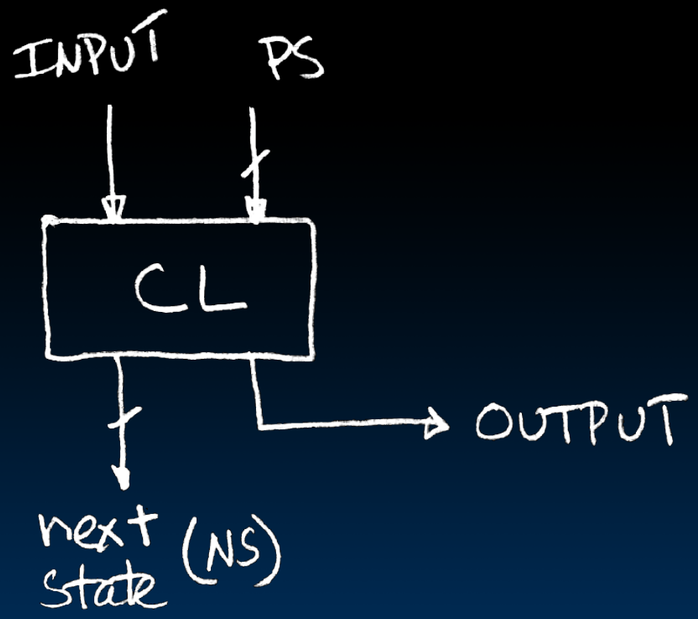
Hardware Implementation of FSM

- **… Therefore a register is needed to hold the representation of which state the machine is in.**
  - Use a unique bit pattern for each state.
- **Combinational logic circuit is used to implement a function mapping the input and present state (PS) input to the next state (NS) and output.**



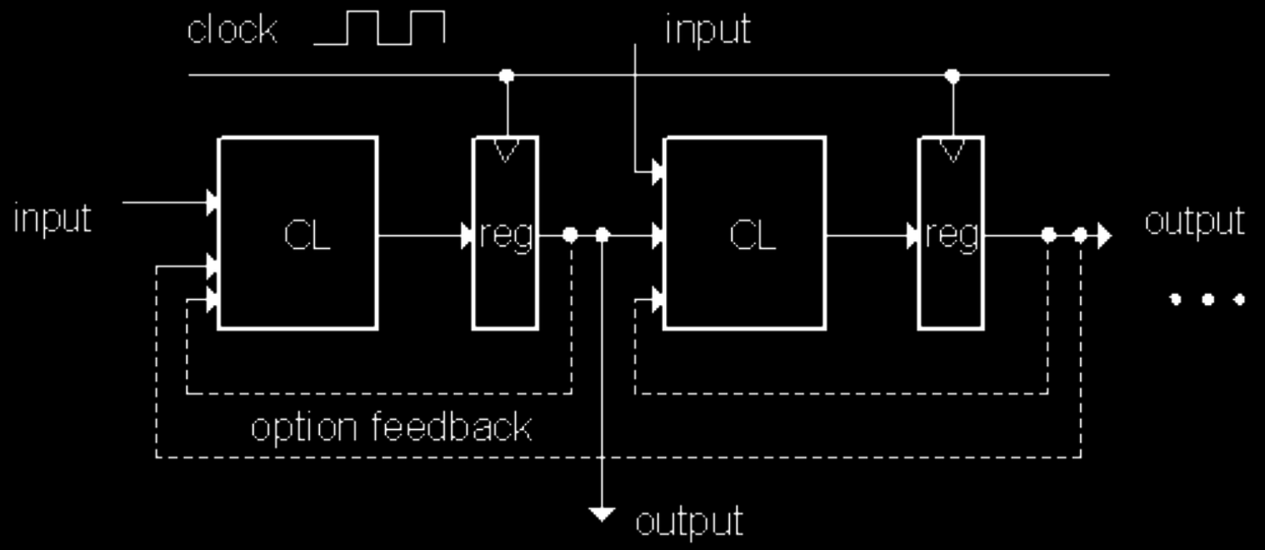
Hardware for FSM: Combinational Logic

- **Next lecture we will discuss the detailed implementation, but for now can look at its functional specification, truth table form.**



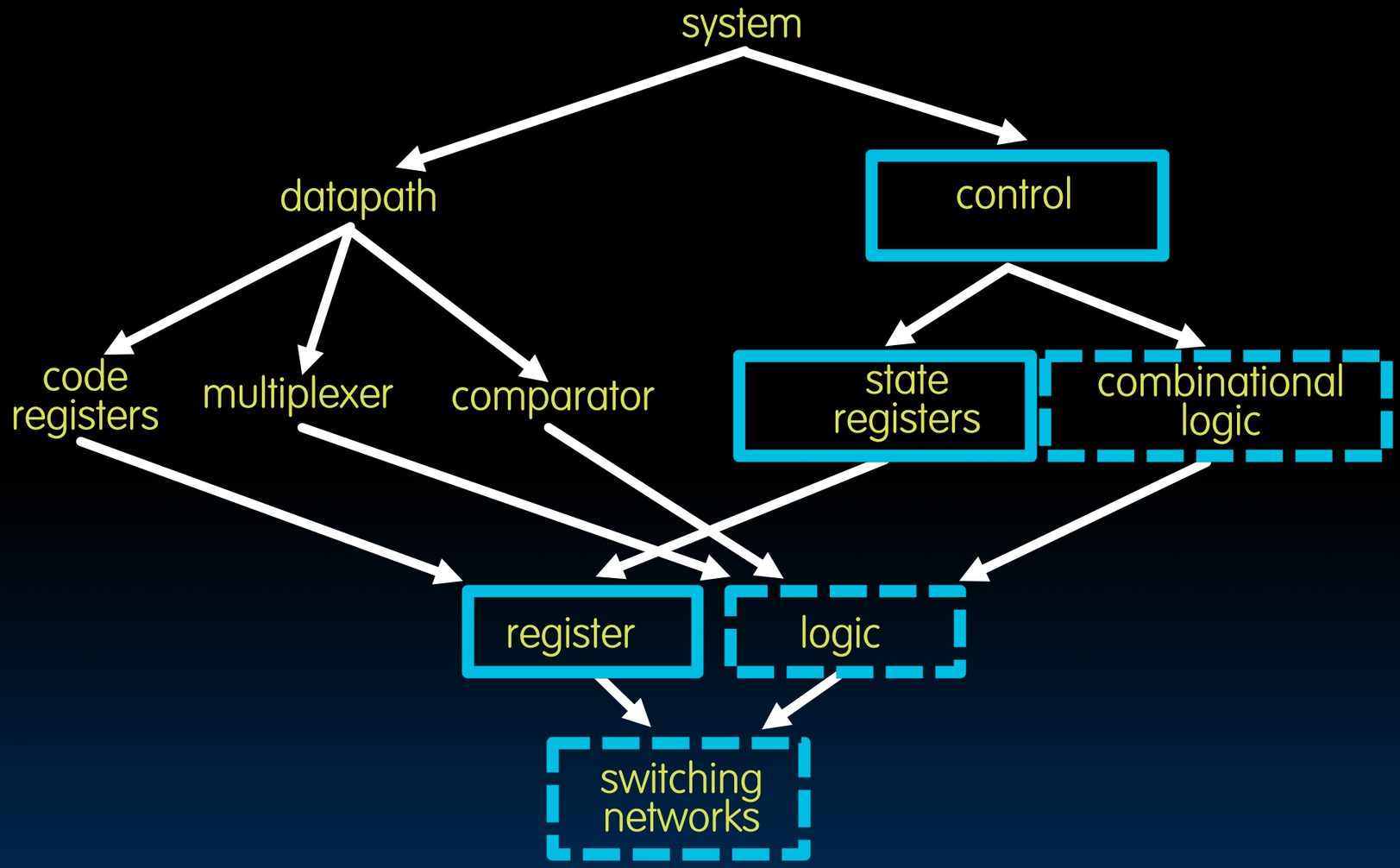
PS	Input	NS	Output
00	0	00	0
00	1	01	0
01	0	00	0
01	1	10	0
10	0	00	0
10	1	00	1

General Model for Synchronous Systems



- Collection of CL blocks separated by registers.
- Registers may be back-to-back and CL blocks may be back-to-back.
- Feedback is optional.
- Clock signal(s) connects only to clock input of registers.

Design Hierarchy



"And In conclusion…"

- **State elements are used to:**
  - Build memories
  - Control the flow of information between other state elements and combinational logic
- **D-flip-flops used to build registers**
- **Clocks tell us when D-flip-flops change**
  - setup and hold times are important
- **We pipeline long-delay CL for faster clock**
- **Finite state machines extremely useful**
  - You'll see them again 151A, 152, 164, 172, …