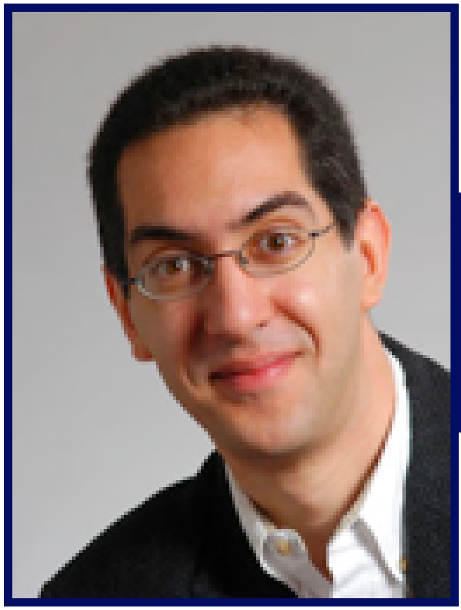
inst.eecs.berkeley.edu/~cs61c

CS61C : Machine Structures

Lecture 21 – State Elements: Circuits That Remember



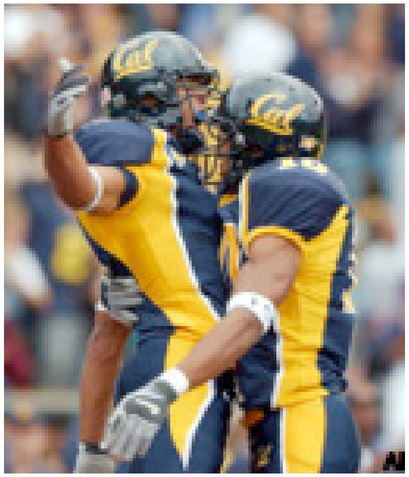
2004-10-18

Lecturer PSOE Dan Garcia

www.cs.berkeley.edu/~ddgarcia

We whomp UCLA! ⇒

After outplaying the #1-ranked team, we come back at our homecoming game and beat UCLA (ex-powerhouse) 45-28. Arrington rushes for 205 yards. 550 yards total! At AZ next week. We're ranked 7th!

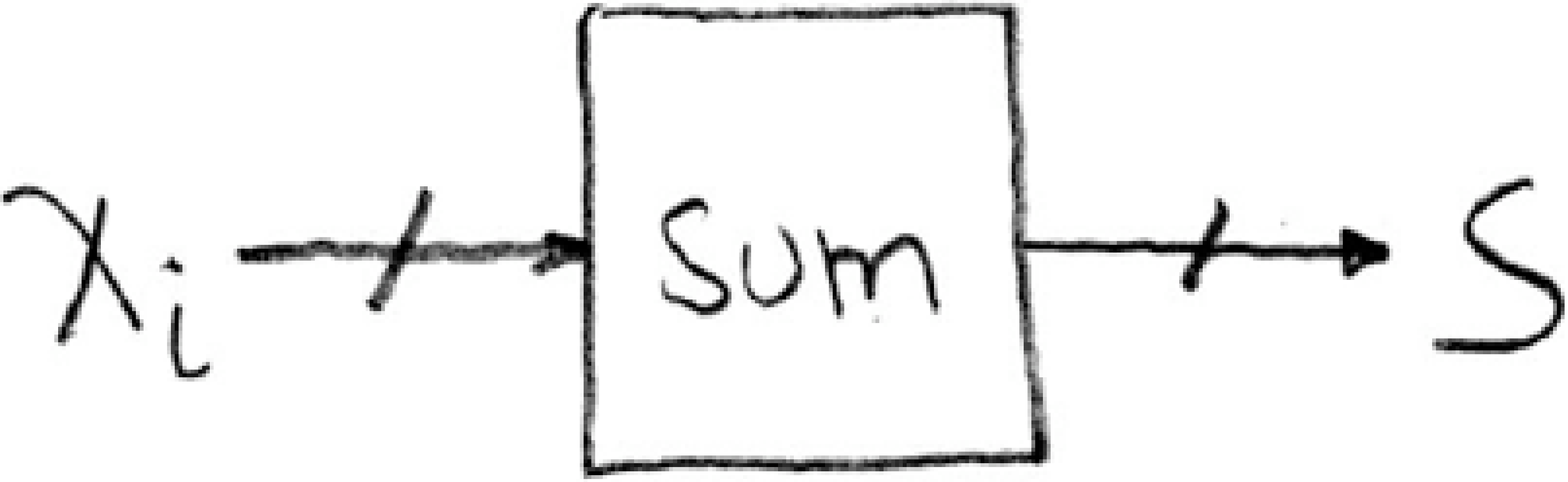


calbears.com

Review…

- ISA is very important abstraction layer
  - Contract between HW and SW
- Basic building blocks are logic ***gates***
- Clocks control pulse of our circuits
- Voltages are analog, quantized to 0/1
- Circuit delays are fact of life
- Two types
  - Stateless Combinational Logic (&,l,~), in which output is function of input only
  - State circuits (e.g., registers)

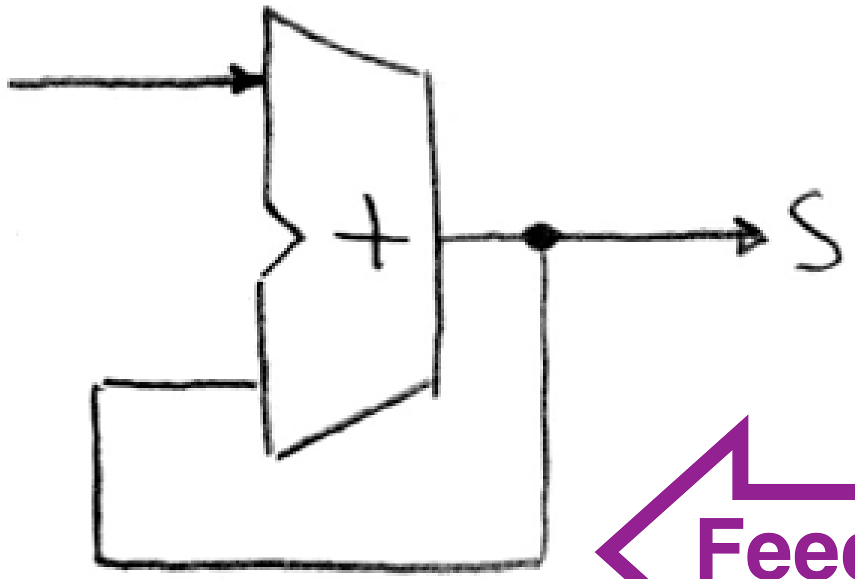
Accumulator Example



Want: `S=0; for i from 0 to n-1`

`S = S + X`$_i$

First try…Does this work?



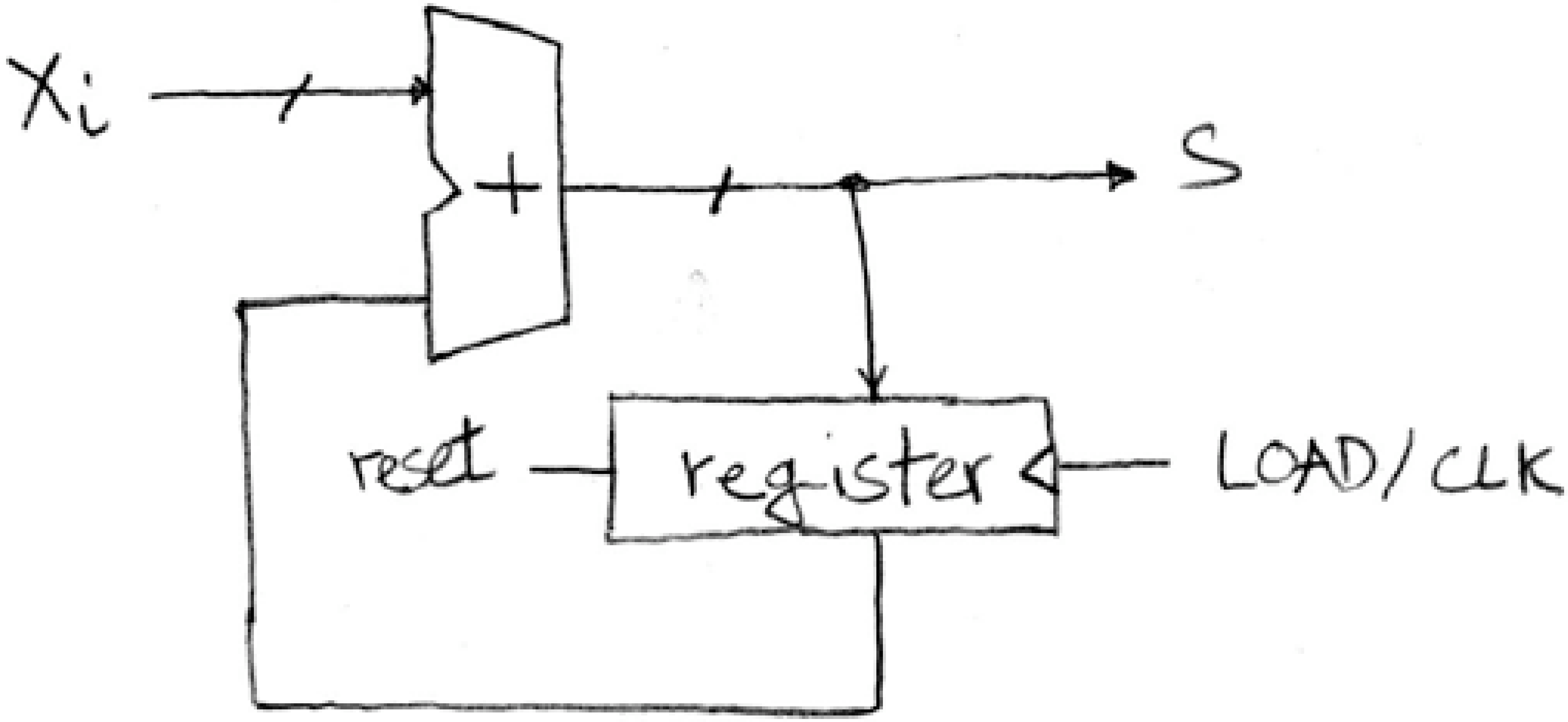
Feedback!

Nope!

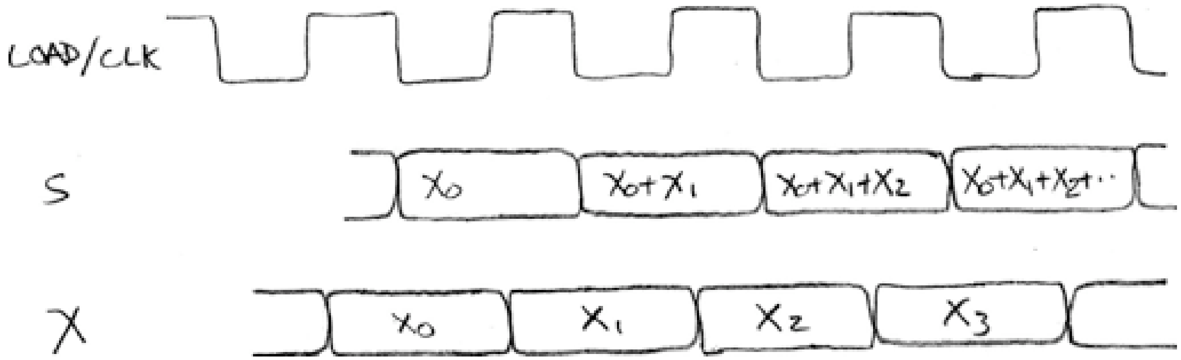
Reason #1… What is there to control the next iteration of the ‘`for`’ loop?

Reason #2… How do we say: ‘`S=0`’?

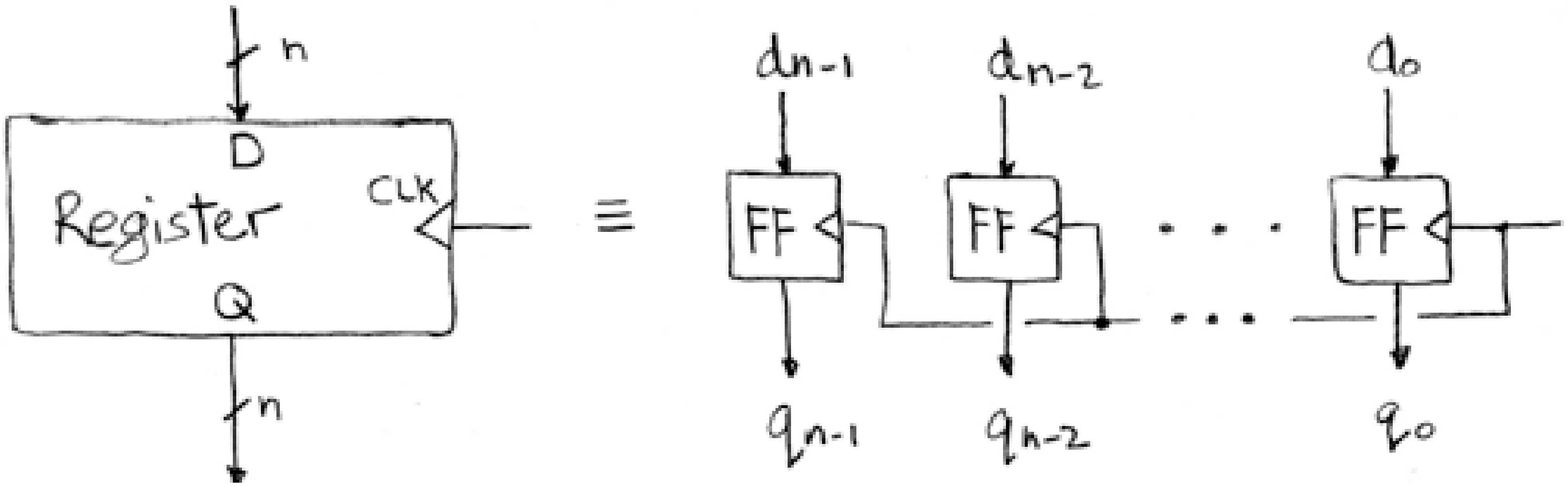
Second try…How about this? Yep!



Rough timing…



Register Details…What's in it anyway?



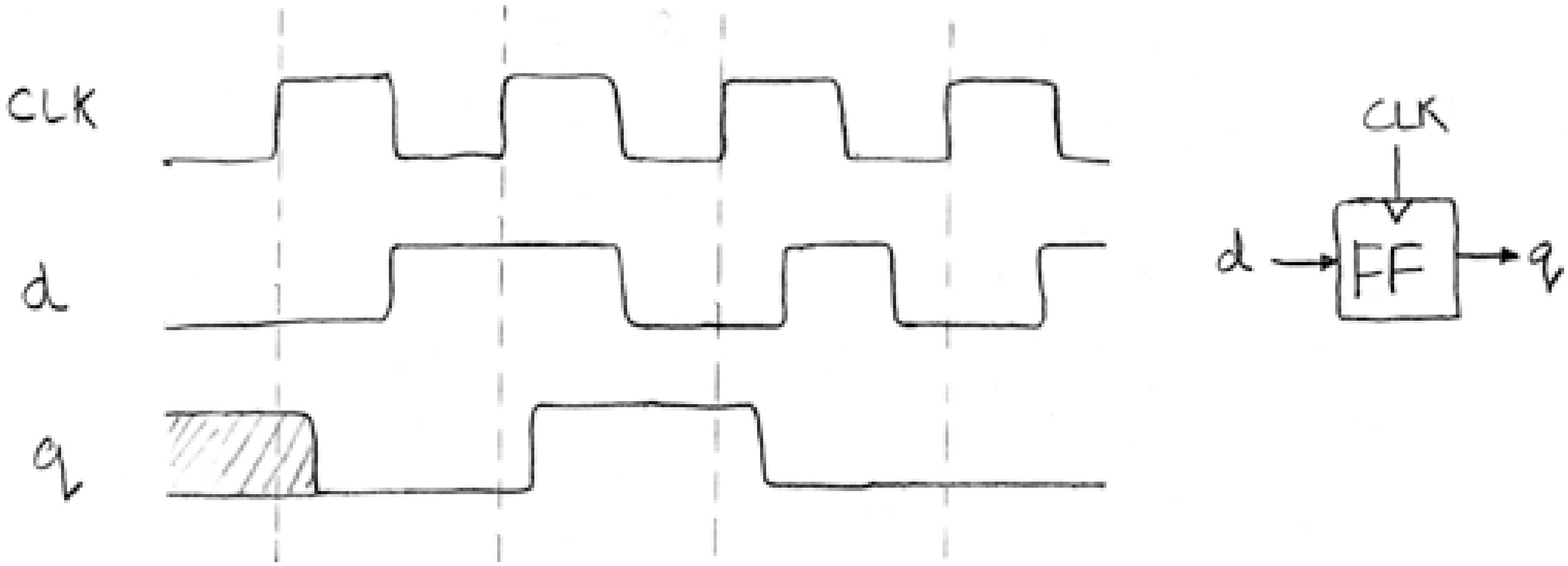
- n instances of a "Flip-Flop", called that because the output flips and flops betw. 0,1

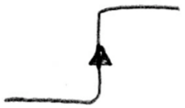
- D is "data"

- Q is "output"

- Also called "d-q Flip-Flop","d-type Flip-Flop"

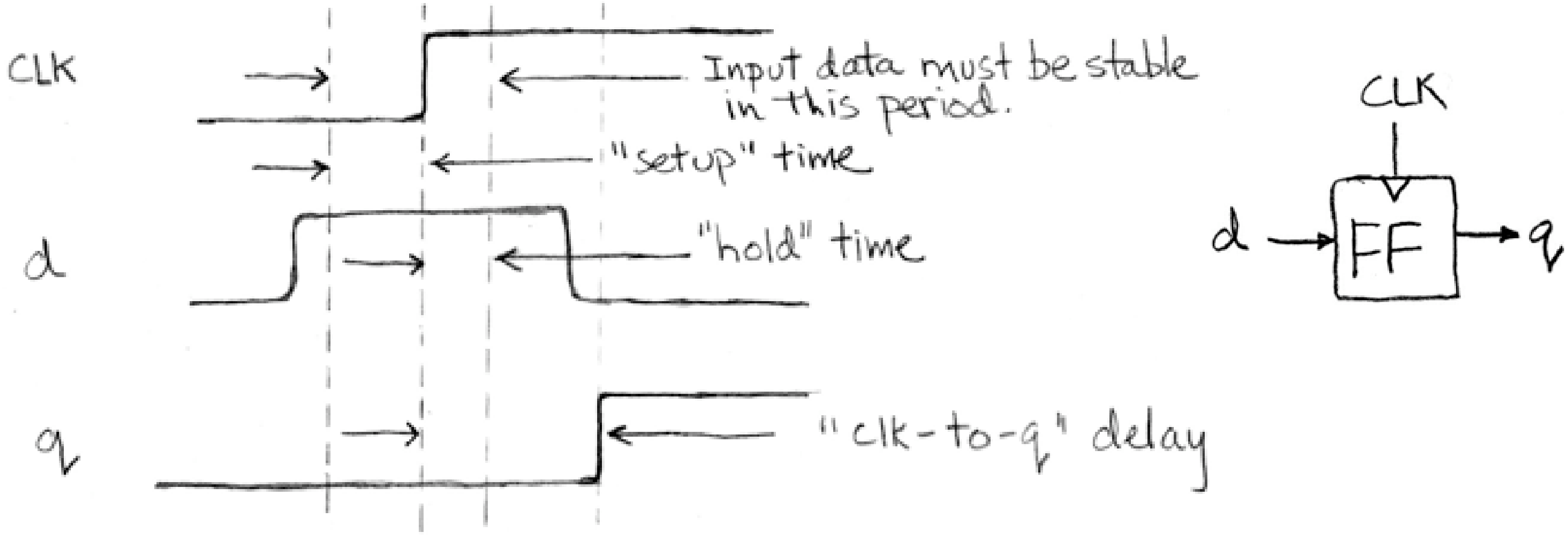
What's the timing of a Flip-flop? (1/2)



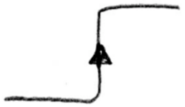
- **Edge-triggered d-type flip-flop**
  - **This one is "positive edge-triggered"** 

- **"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."**

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



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

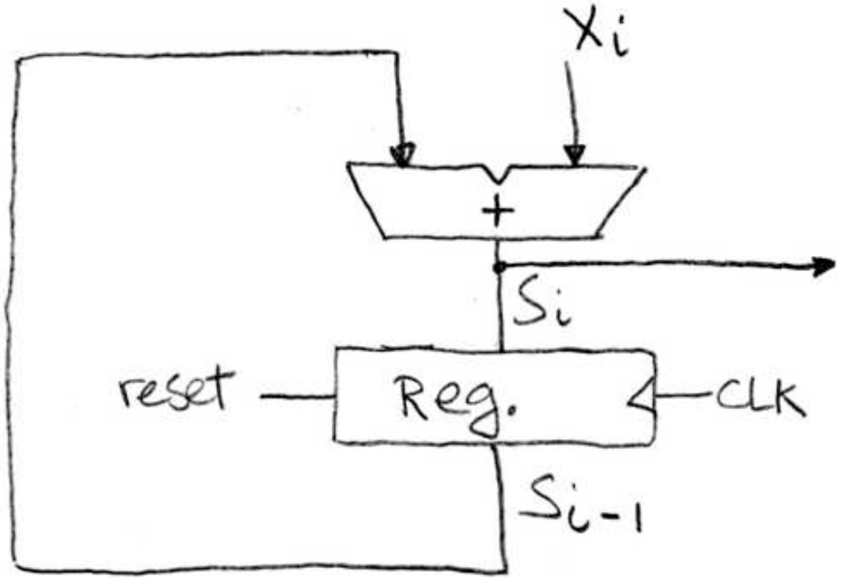


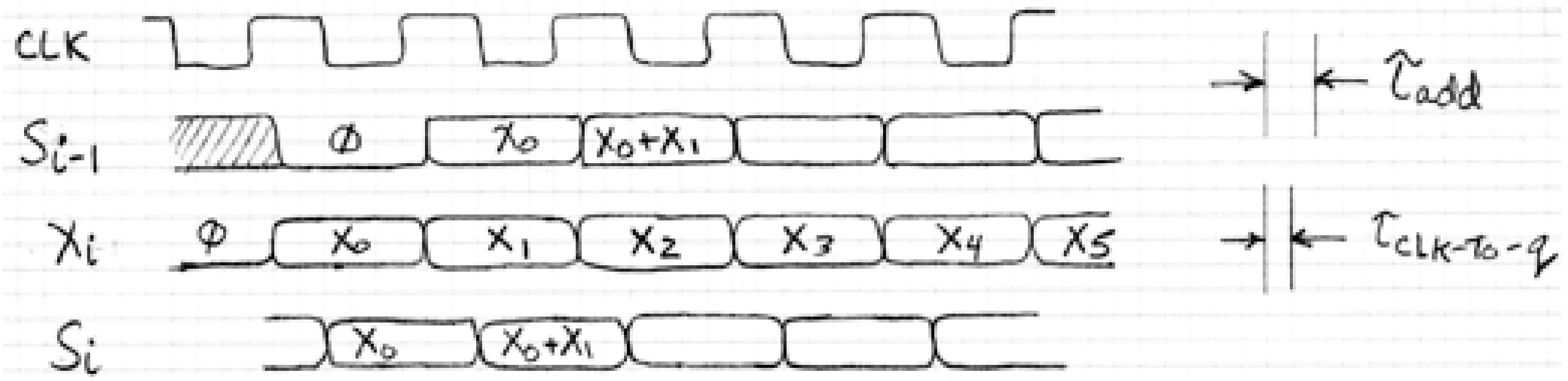
- "On the rising edge of the clock, the input d is sampled and transferred to the output. At all other times, the input d is ignored."

Administrivia

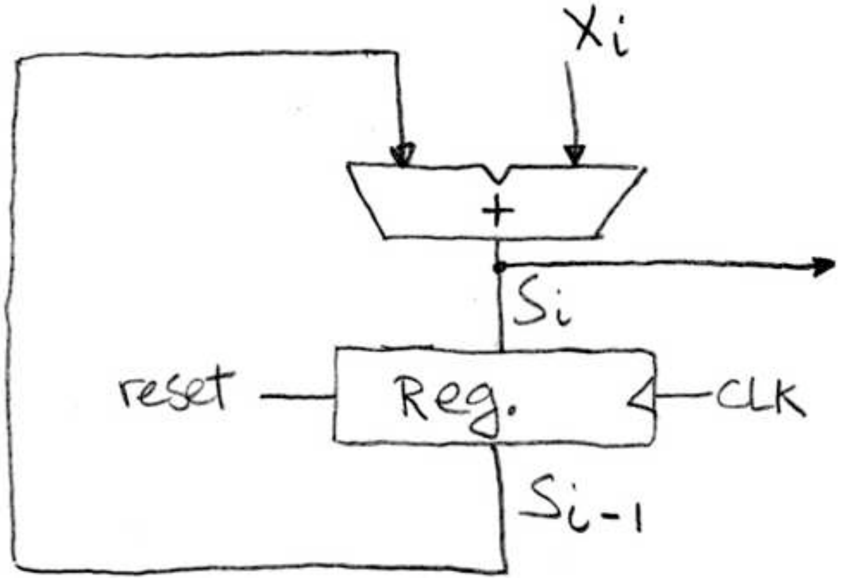
- Midterm tonight!
    - 1 Pimintel 7pm-10pm

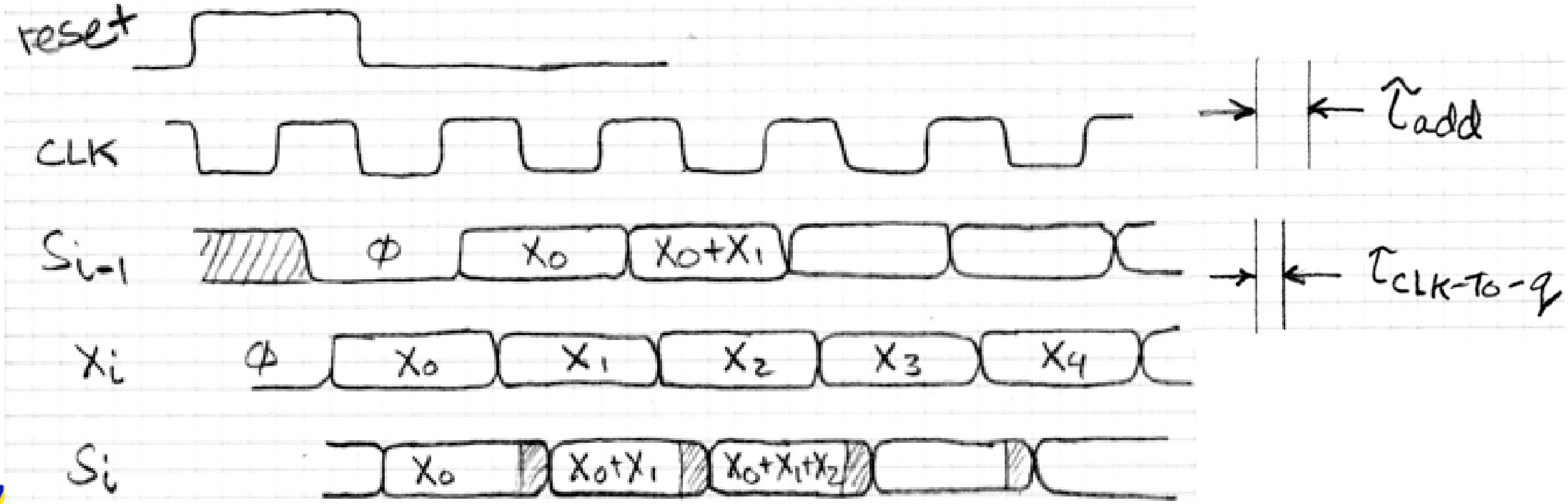
Accumulator Revisited (proper timing 1/2)



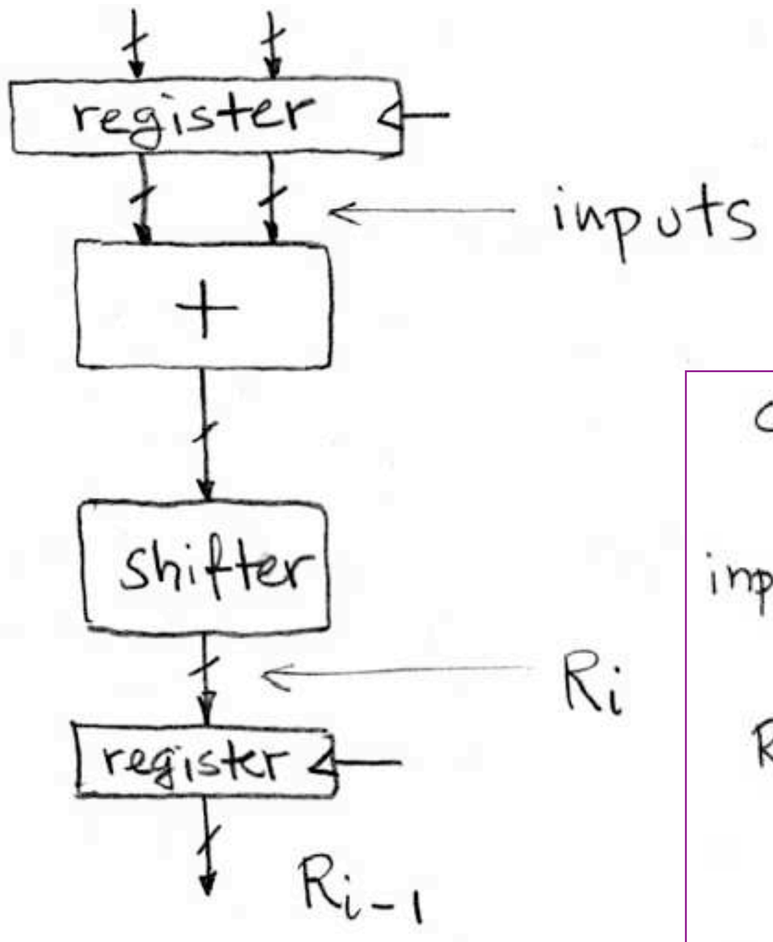


Accumulator Revisited (proper timing 2/2)

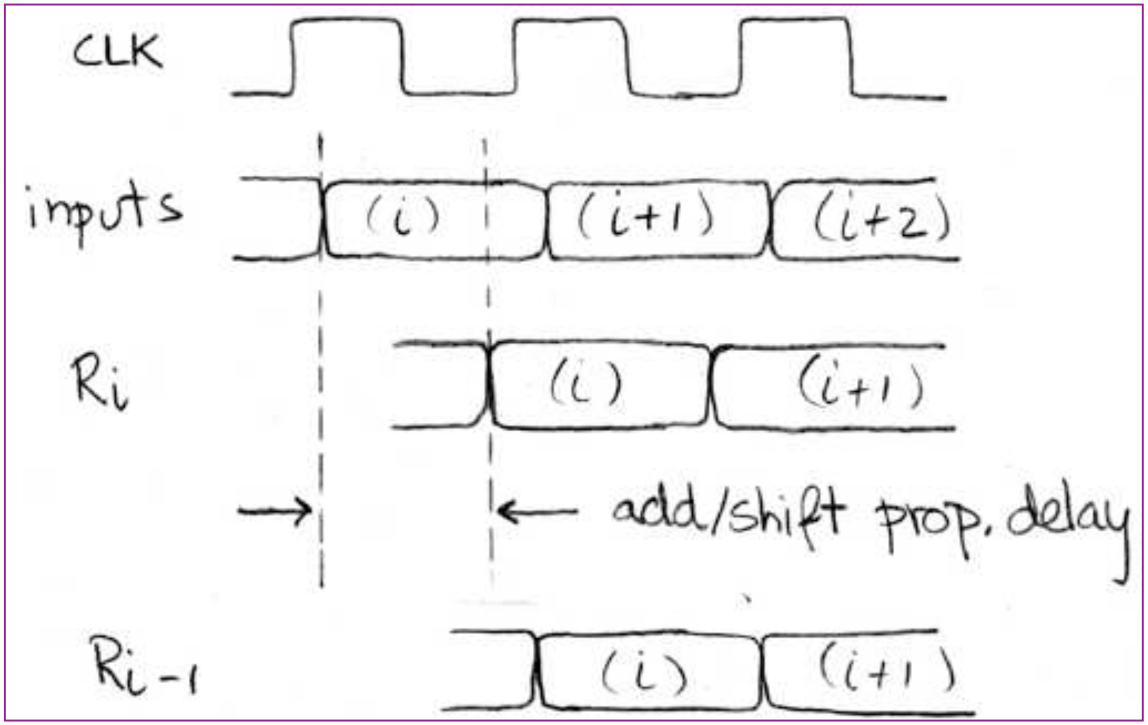




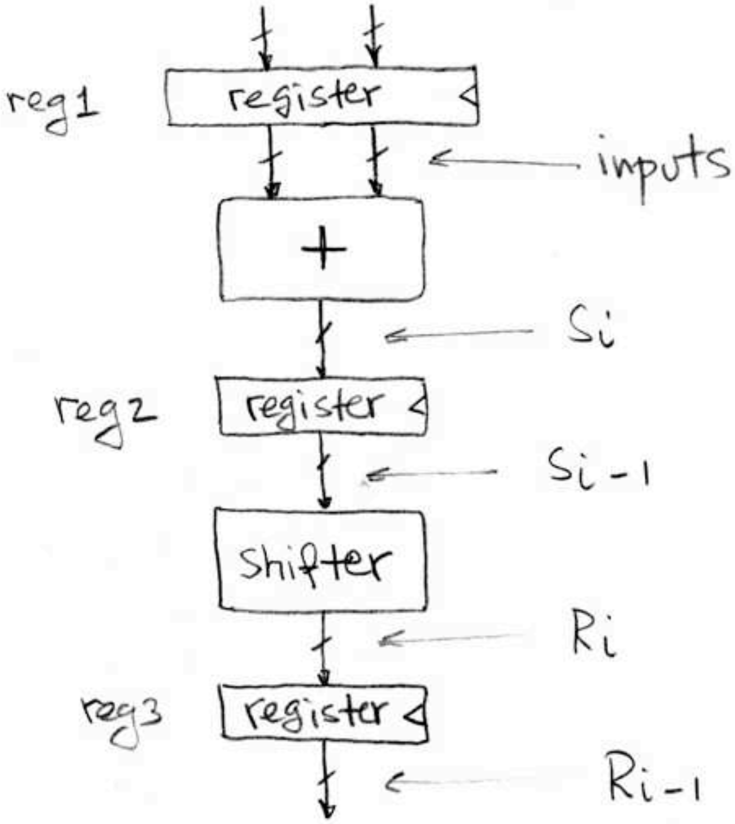
Pipelining to improve performance (1/2)



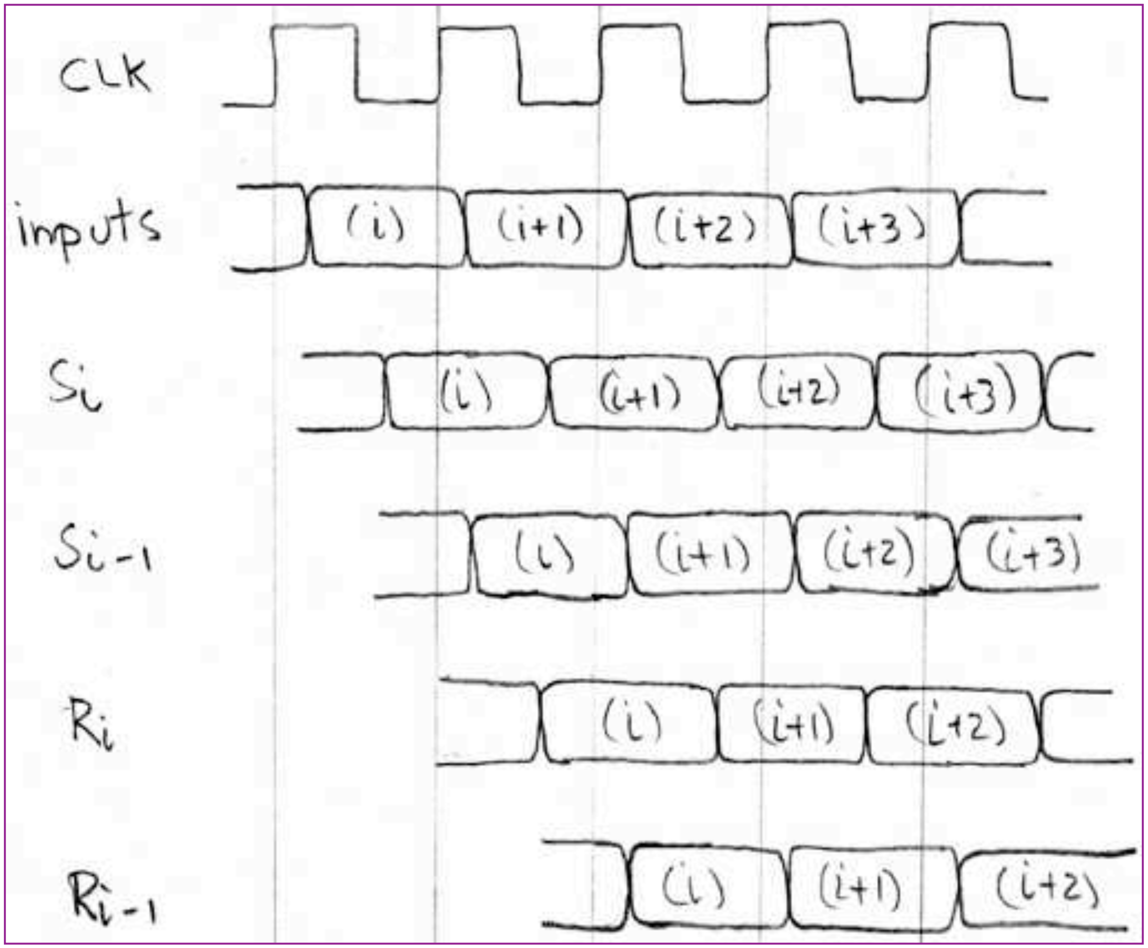
Timing…



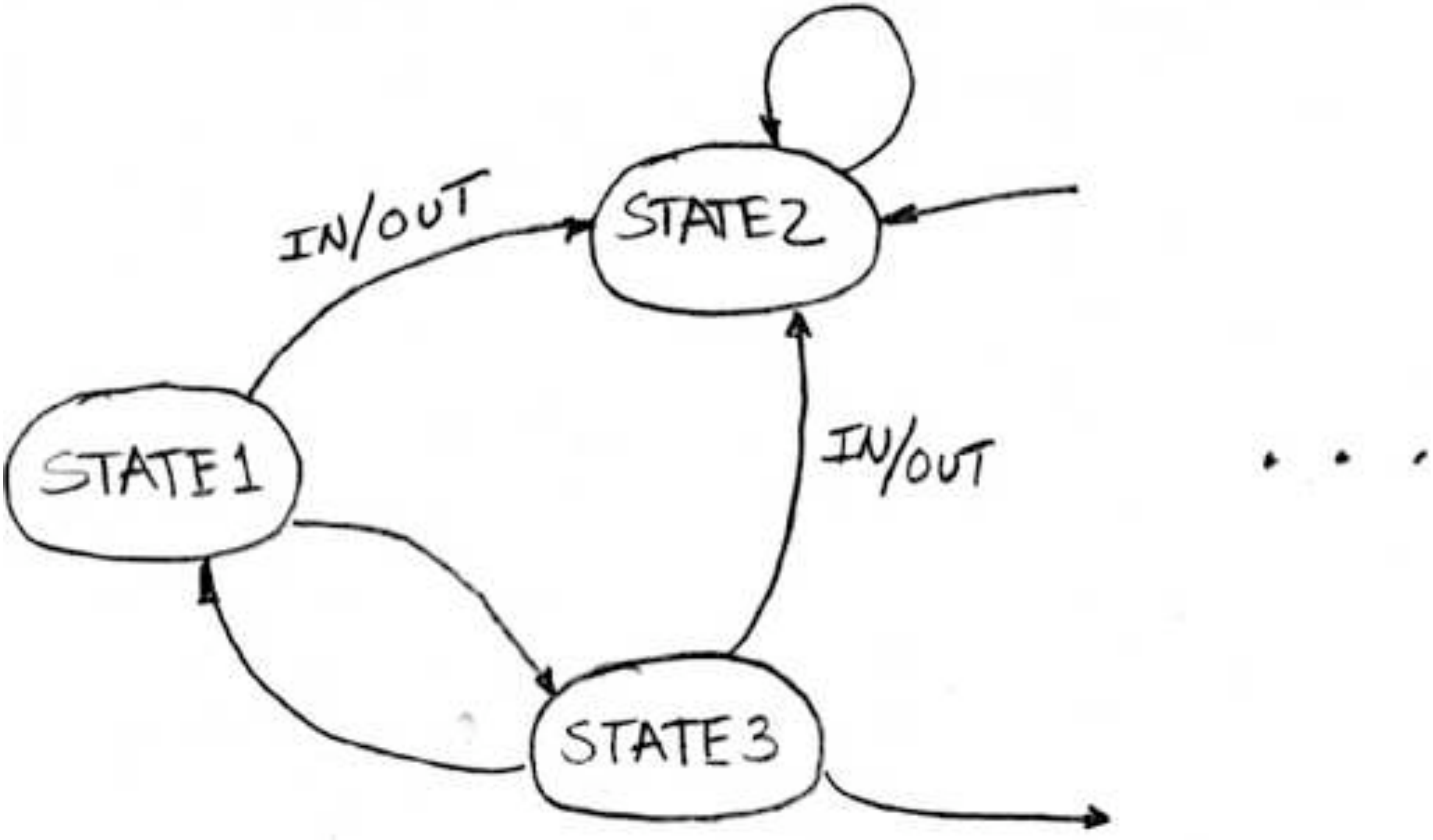
Pipelining to improve performance (2/2)



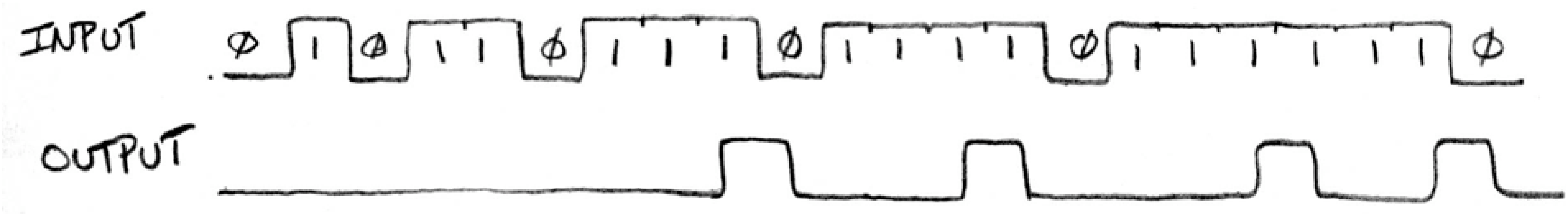
Timing…



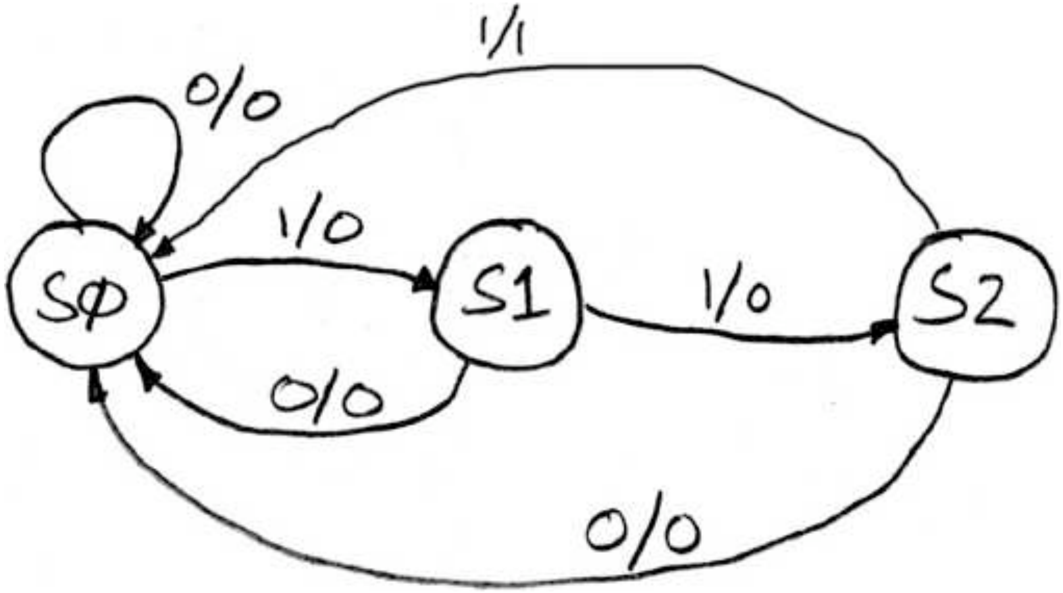
Finite State Machines Introduction



Finite State Machine Example: 3 ones…



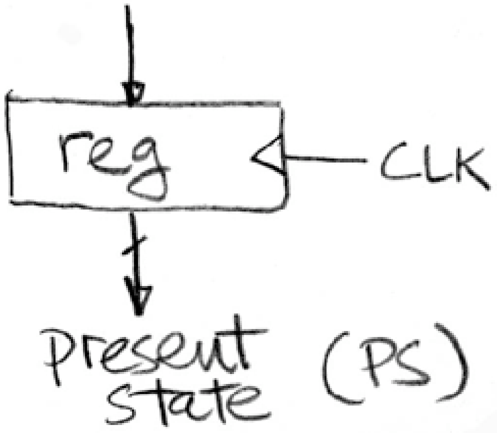
Draw the FSM…



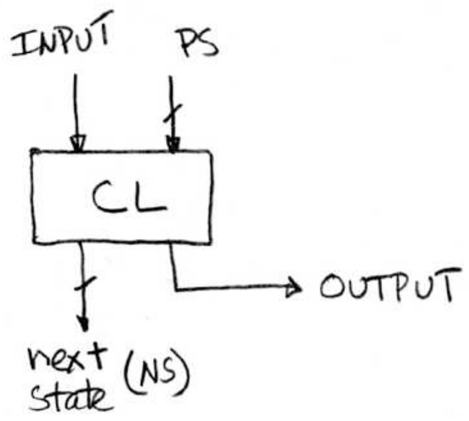
Truth table…

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

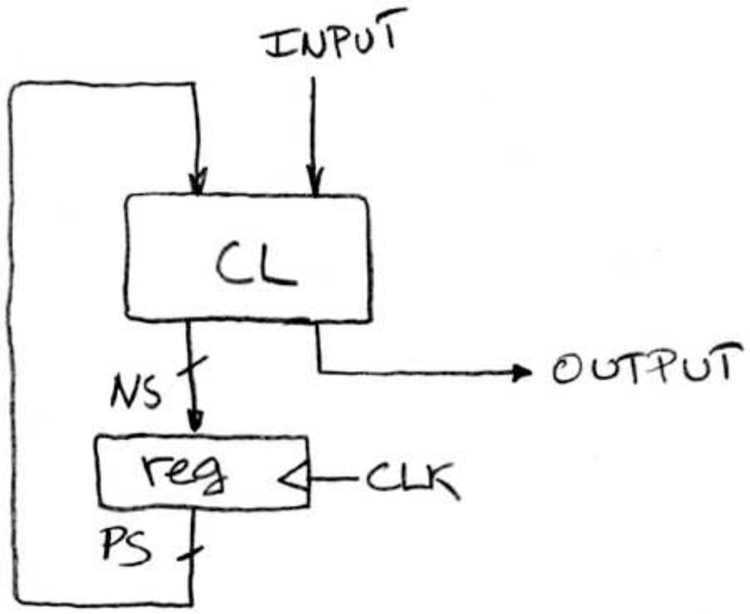
Hardware Implementation of FSM



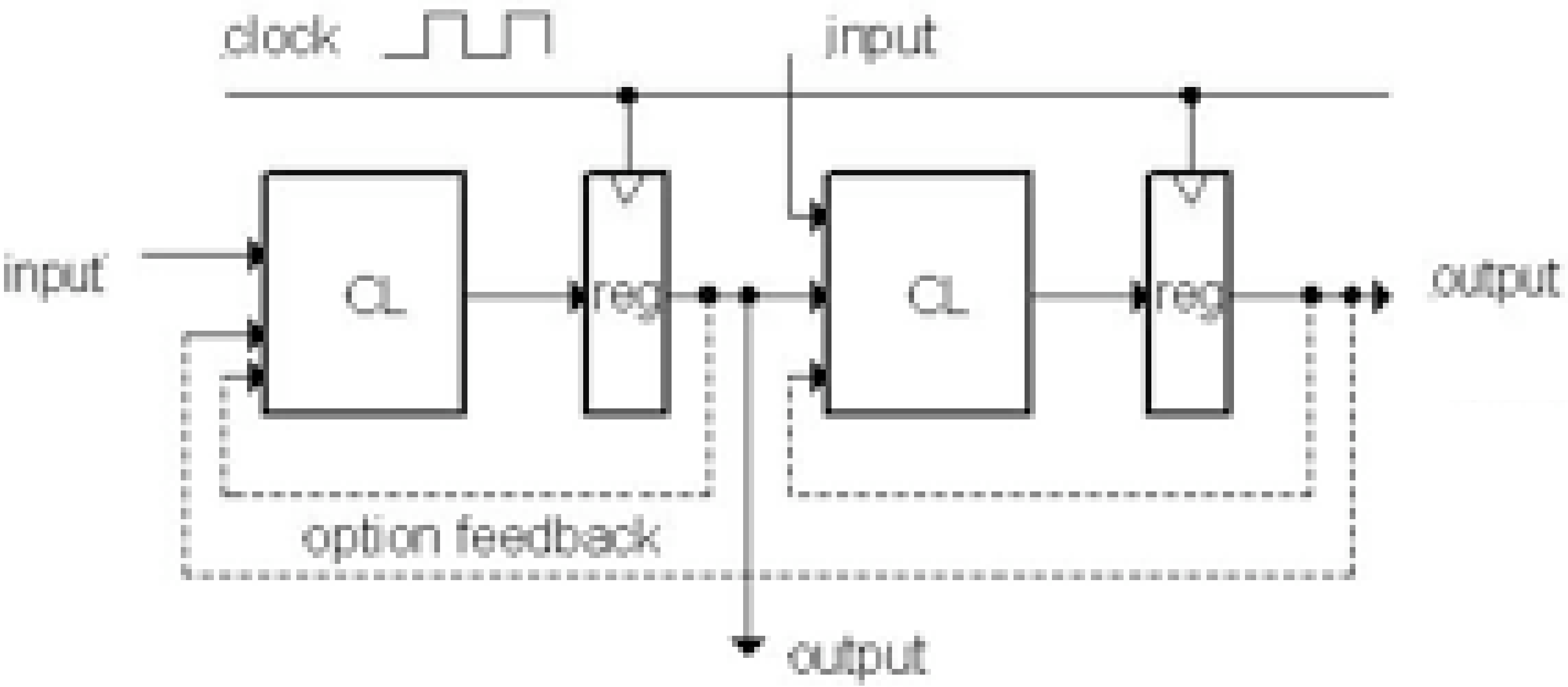
+



=



General Model for Synchronous Systems



Peer Instruction

A. HW feedback akin to SW recursion

B. We can implement a D-Q flipflop as simple CL (And, Or, Not gates)

C. You can build a FSM to signal when an equal number of 0s and 1s has appeared in the input.

	ABC
1:	FFF
2:	FFT
3:	FTF
4:	FTT
5:	TFF
6:	TFT
7:	TTF
8:	TTT

Peer Instruction Answer

"And In conclusion…"

- We use feedback to maintain state
- Register files used to build memories
- D-FlipFlops used to build Register files
- Clocks tell us when D-FlipFlops change
  - Setup and Hold times important
- We pipeline big-delay CL for faster clock
- Finite State Machines extremely useful
  - You'll see them in HW classes (150,152) & 164