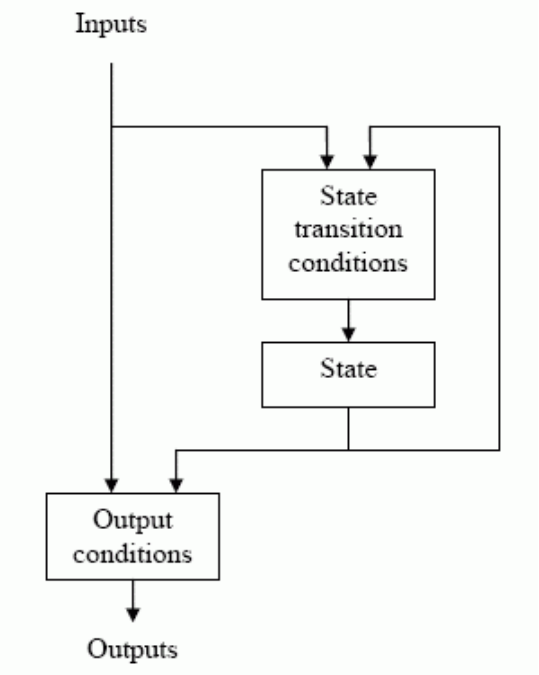
inst.eecs.berkeley.edu/~cs61c/su05

CS61C : Machine Structures

Lecture #14: State and FSMs



2005-07-13

Andy Carle

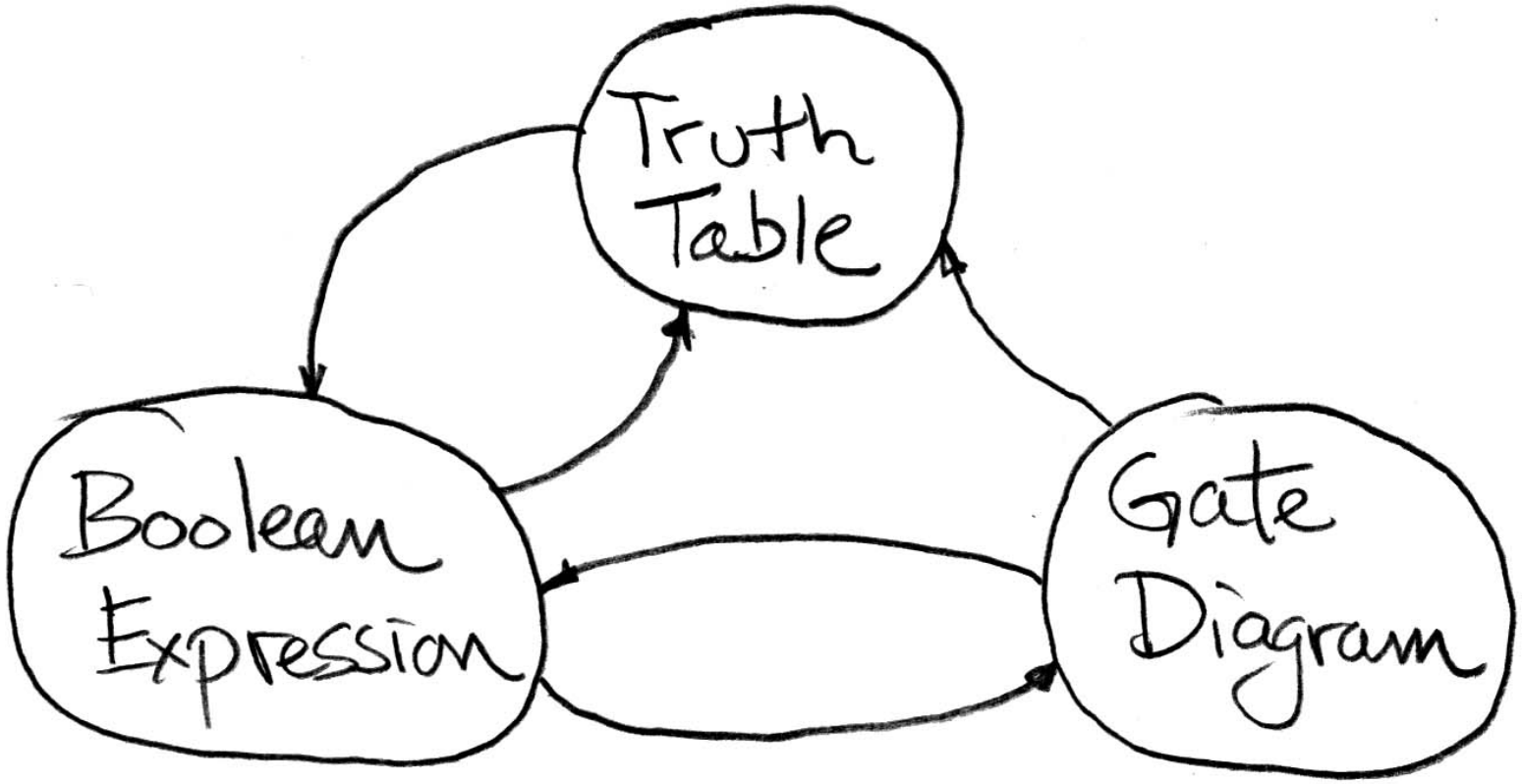
Outline

- Waveforms
- State
- Clocks

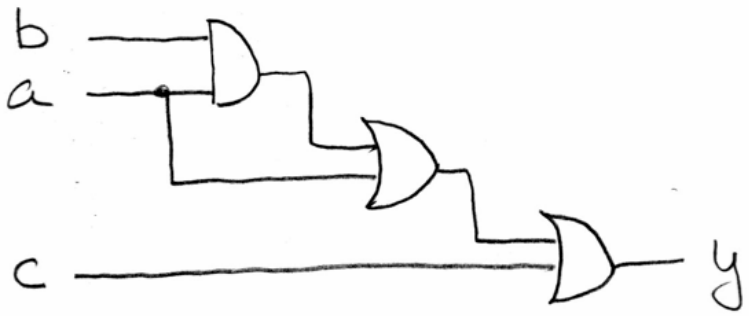
- FSMs

Review (1/3)

- Use this table and techniques we learned to transform from 1 to another



(2/3): Circuit & Algebraic Simplification



original circuit

$$\downarrow$$

$$y = ((ab) + a) + c$$

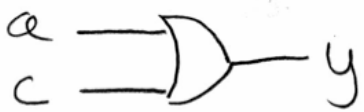
equation derived from original circuit

$$\downarrow$$

$$\begin{aligned} &= ab + a + c \\ &= a(b + 1) + c \\ &= a(1) + c \\ &= a + c \end{aligned}$$

algebraic simplification

$$\downarrow$$



simplified circuit

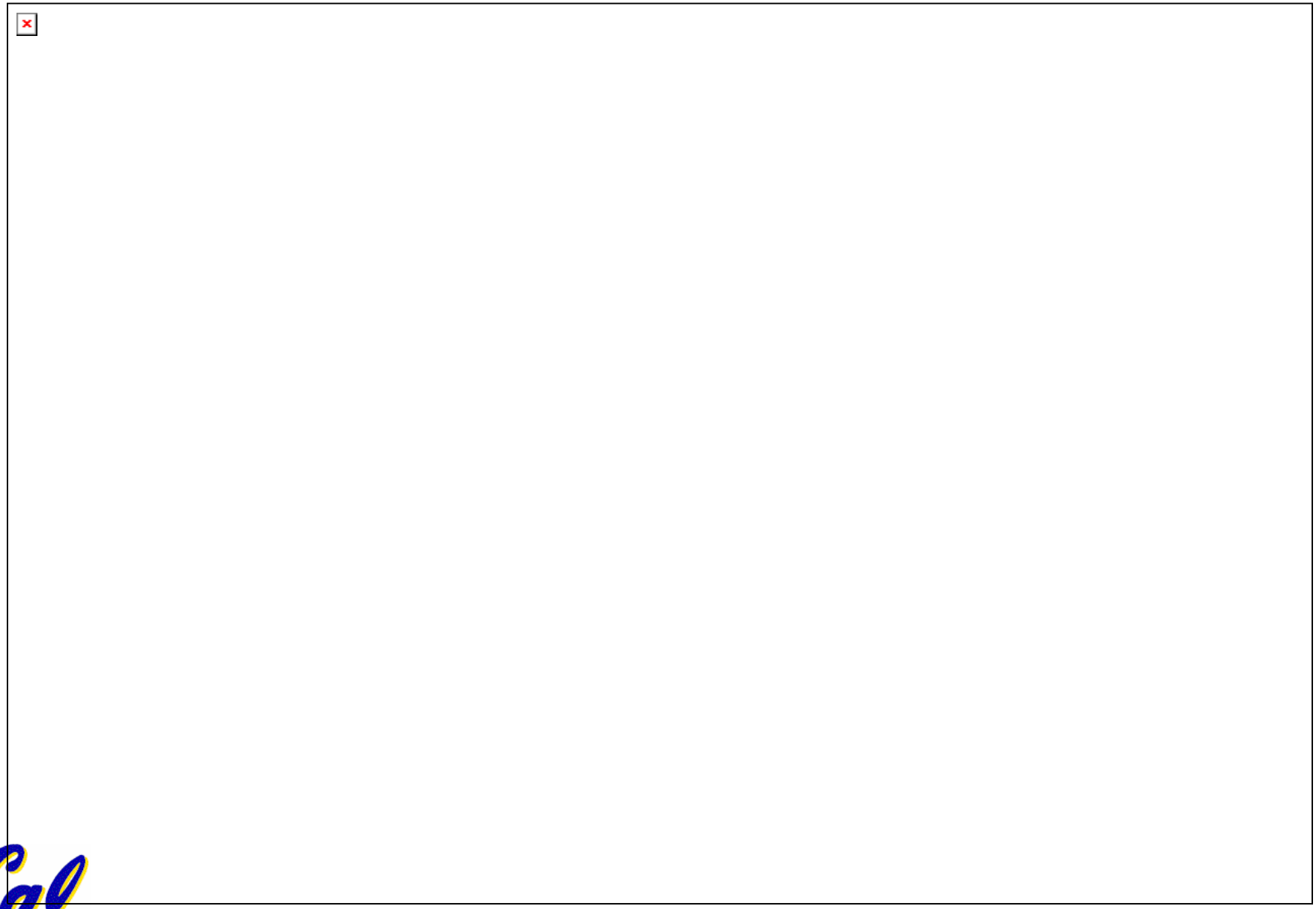
(3/3):Laws of Boolean Algebra

$x \cdot \overline{x} = 0$	$x + \overline{x} = 1$	complementarity
$x \cdot 0 = 0$	$x + 1 = 1$	laws of 0's and 1's
$x \cdot 1 = x$	$x + 0 = x$	identities
$x \cdot x = x$	$x + x = x$	idempotent law
$x \cdot y = y \cdot x$	$x + y = y + x$	commutativity
$(xy)z = x(yz)$	$(x + y) + z = x + (y + z)$	associativity
$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$	distribution
$xy + x = x$	$(x + y)x = x$	uniting theorem
$\overline{x \cdot y} = \overline{x} + \overline{y}$	$\overline{(x + y)} = \overline{x} \cdot \overline{y}$	DeMorgan's Law

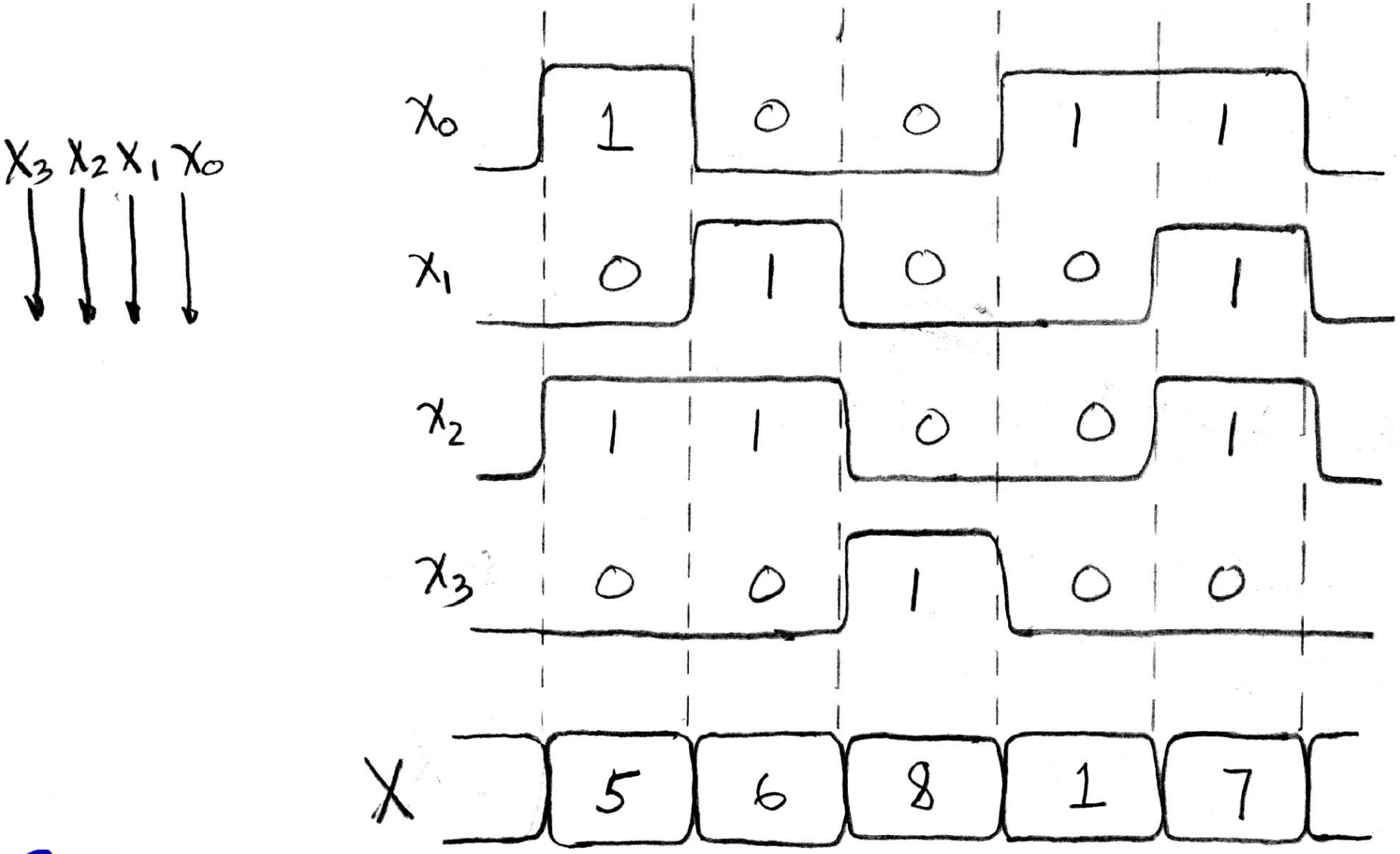
Signals and Waveforms

- Outputs of CL change over time
  - With what? → Change in inputs
  - Can graph changes with waveforms …

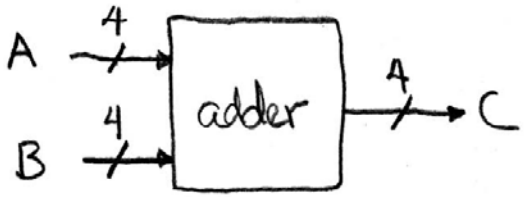
Signals and Waveforms: Adders



Signals and Waveforms: Grouping

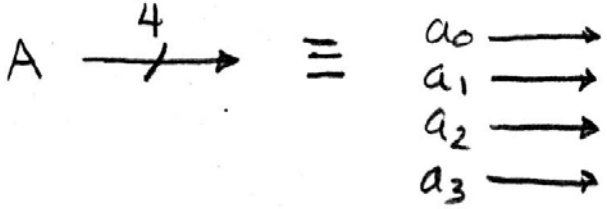


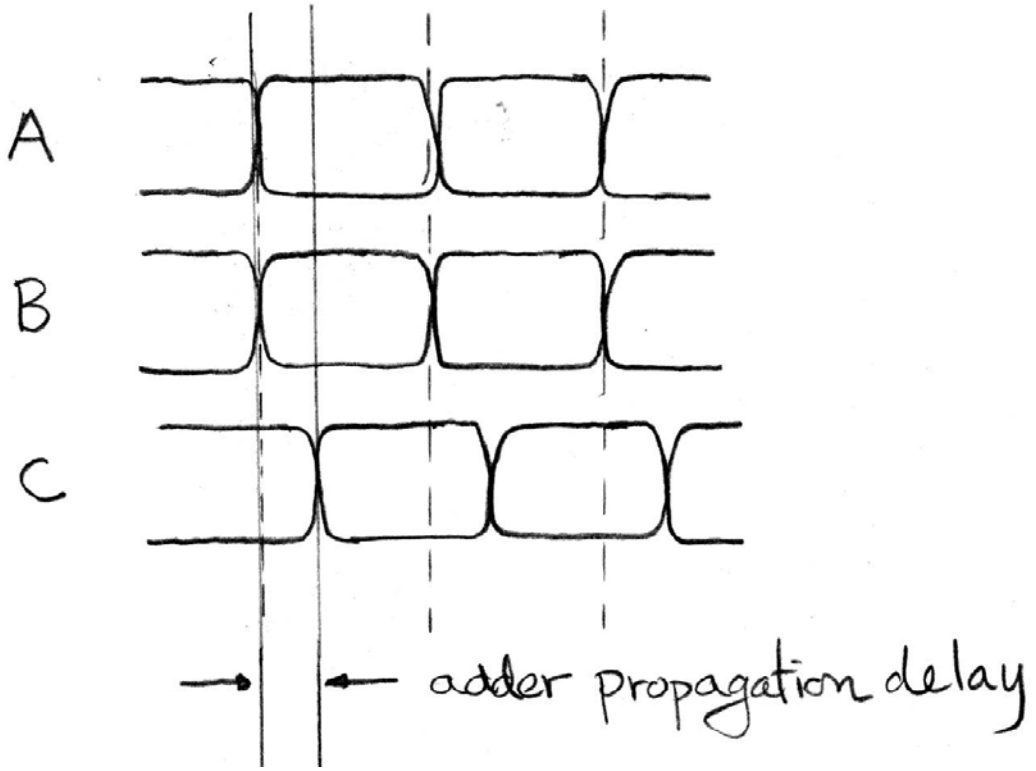
Signals and Waveforms: Circuit Delay



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

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

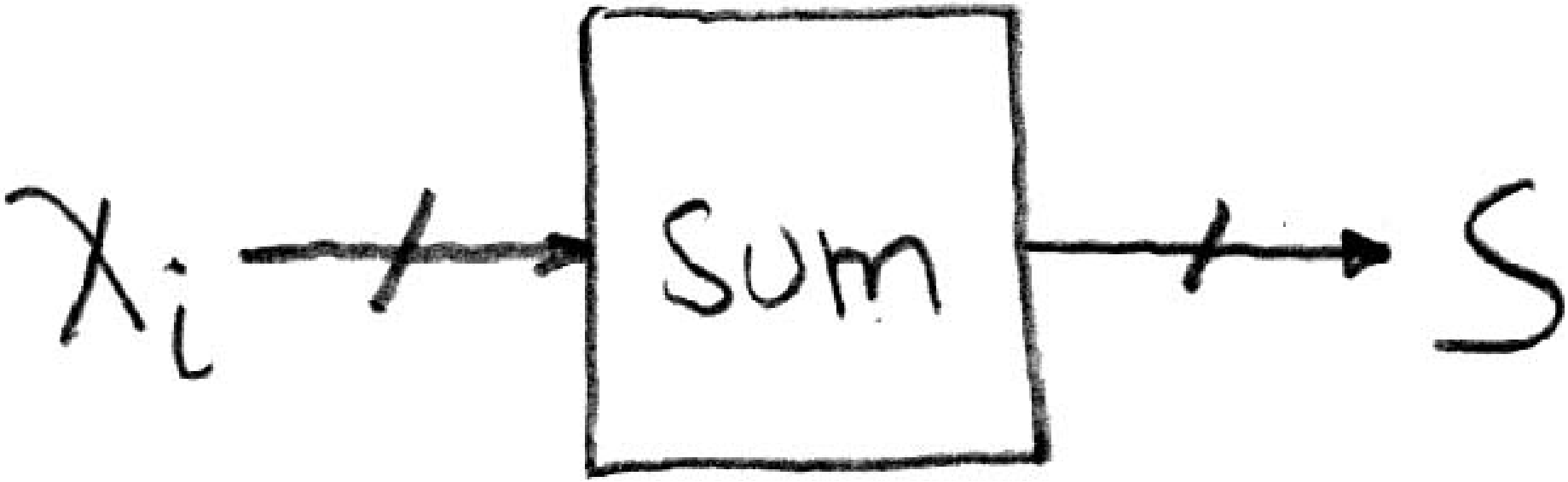




State

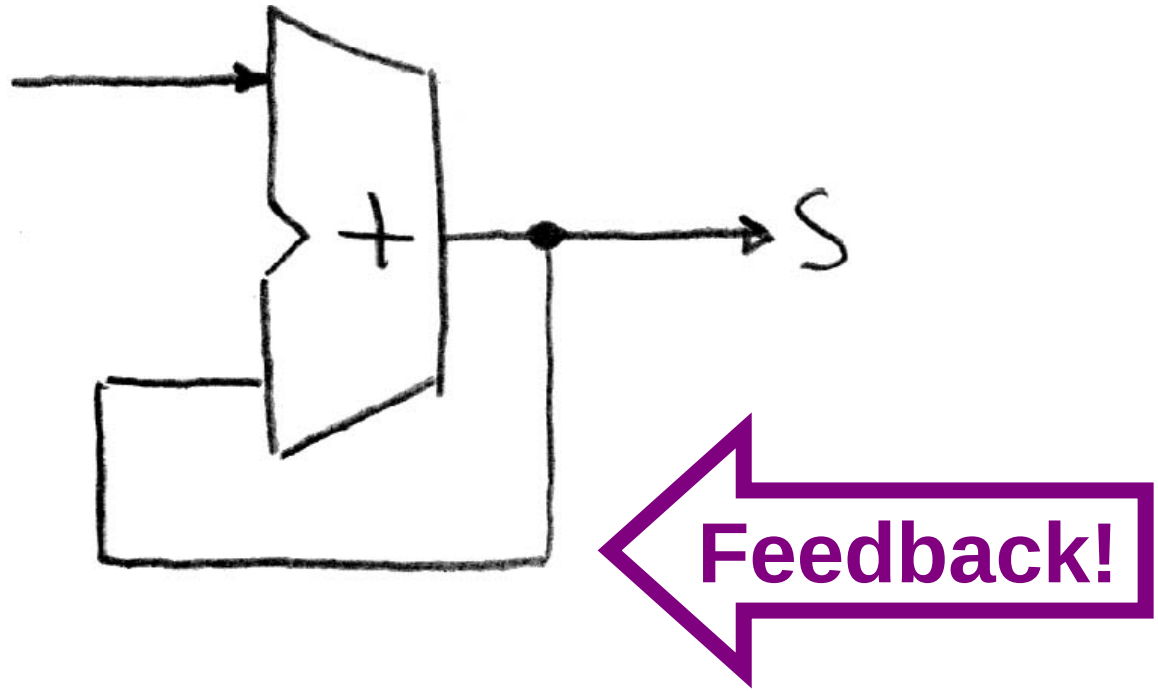
- With CL, output is always a function of CURRENT input
  - With some (variable) propagation delay
- Clearly, we need a way to introduce state into computation

Accumulator Example



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

First try…Does this work?



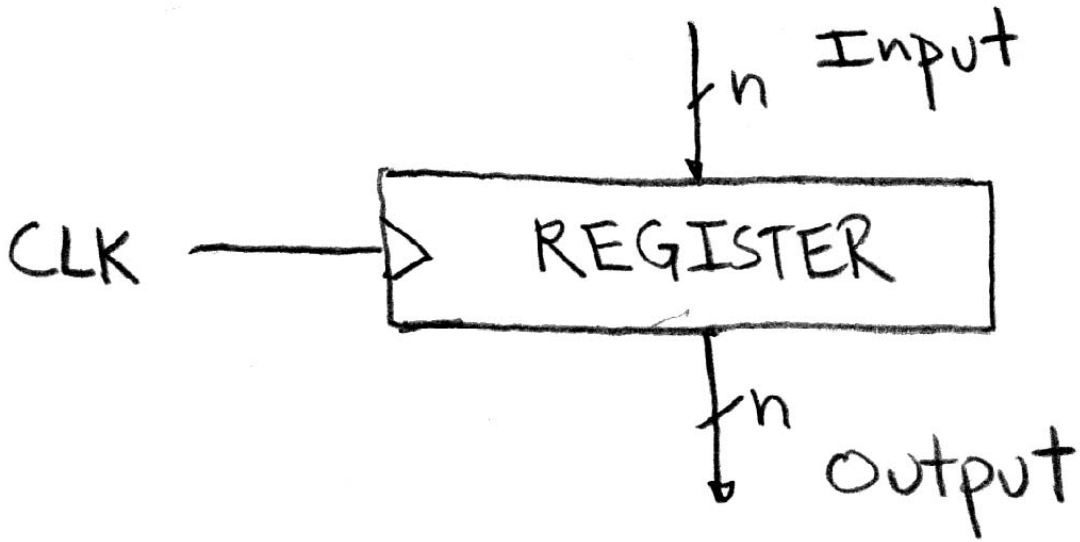
Nope!

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

Reason #2… How do we say: ‘`S=0`’?

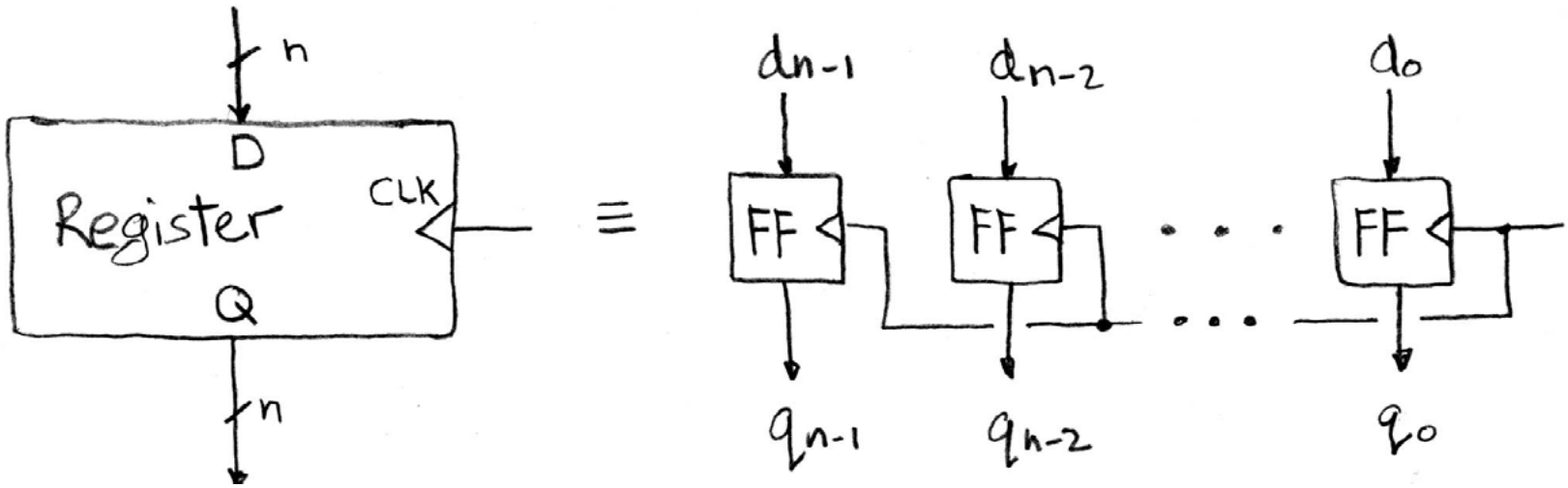
Need a way to store partial sums! …

Circuits with STATE (e.g., register)



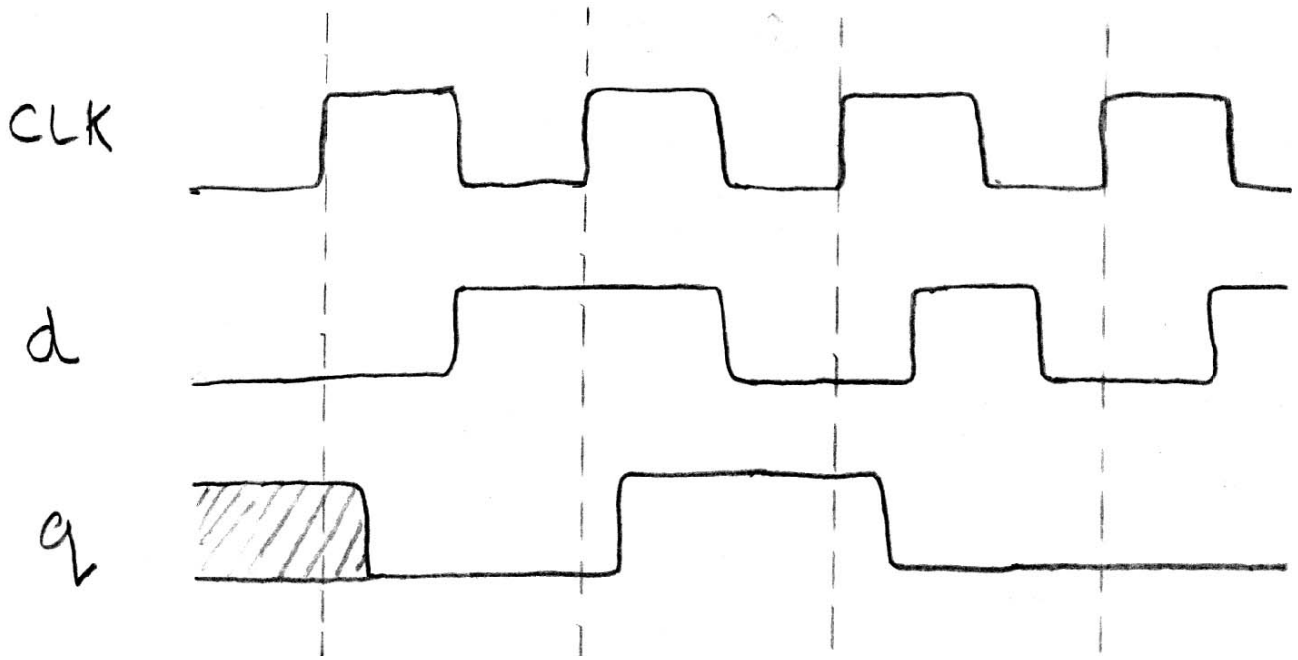
Need a Logic Block that will:
1. store output (partial sum) for a while,
2. until we tell it to update with a new value.

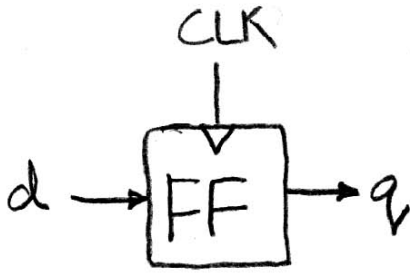
Register Details…What's in it anyway?



- 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)

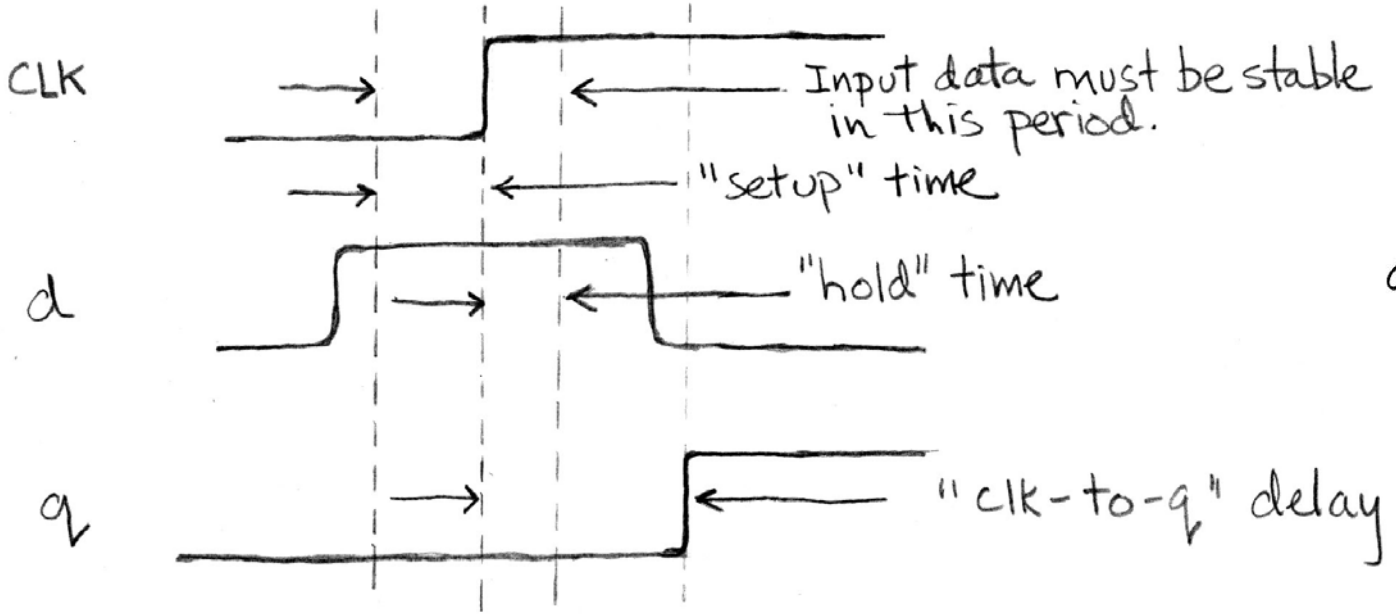


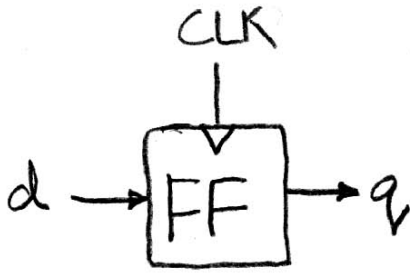


- **Edge-triggered D-type flip-flop**
  - **This one is "positive edge-triggered"**

- **"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)

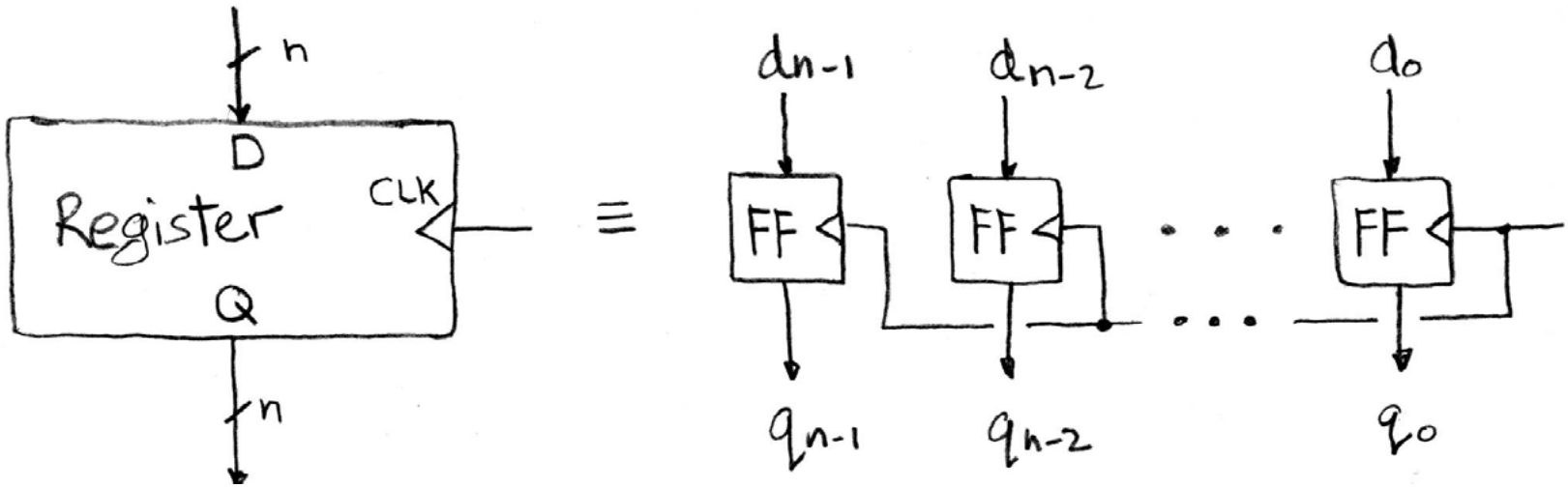




- **Edge-triggered D-type flip-flop**
  - **This one is "positive edge-triggered"**

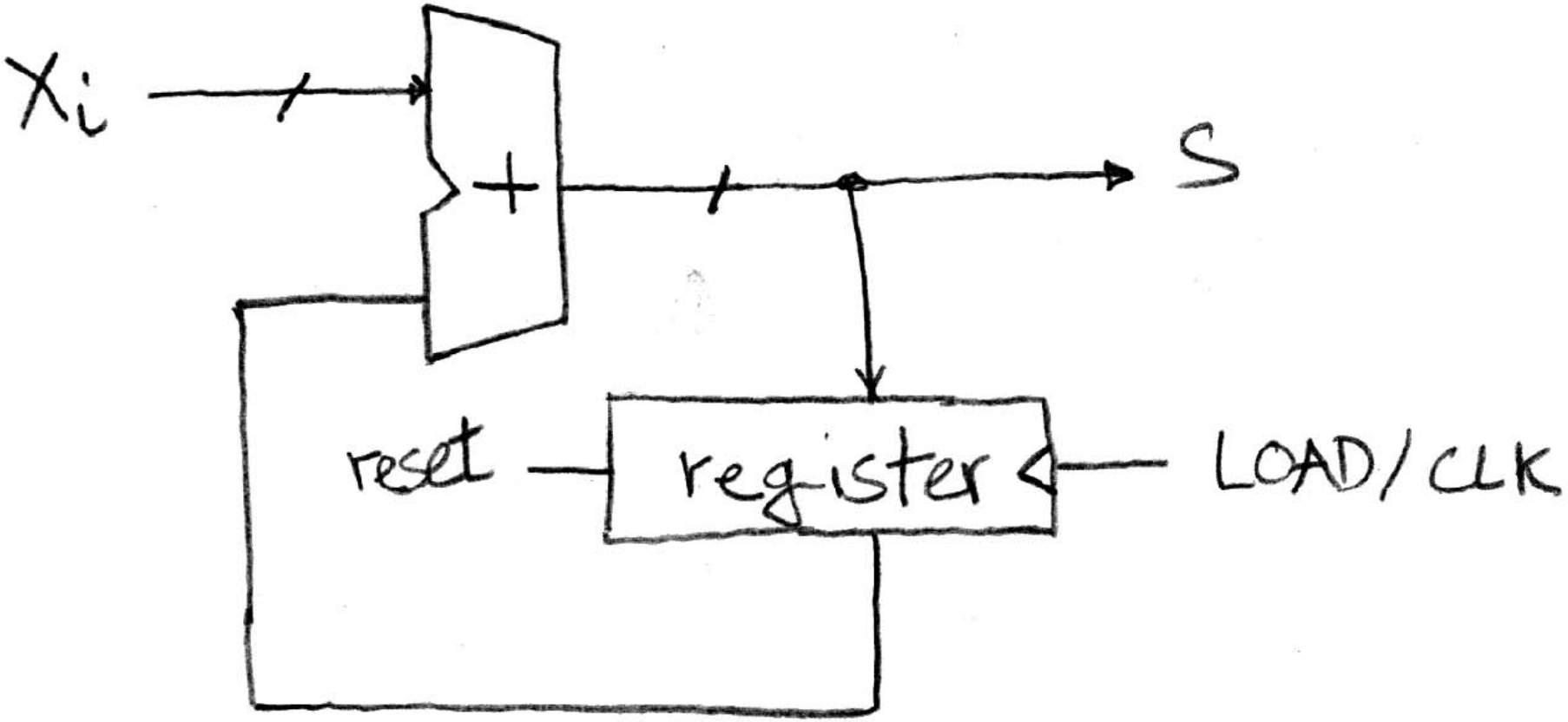
- **"On the rising edge of the clock, the input d is sampled and transferred to the output. At all other times, the input d is ignored."**

Bus a bunch of D FFs together …

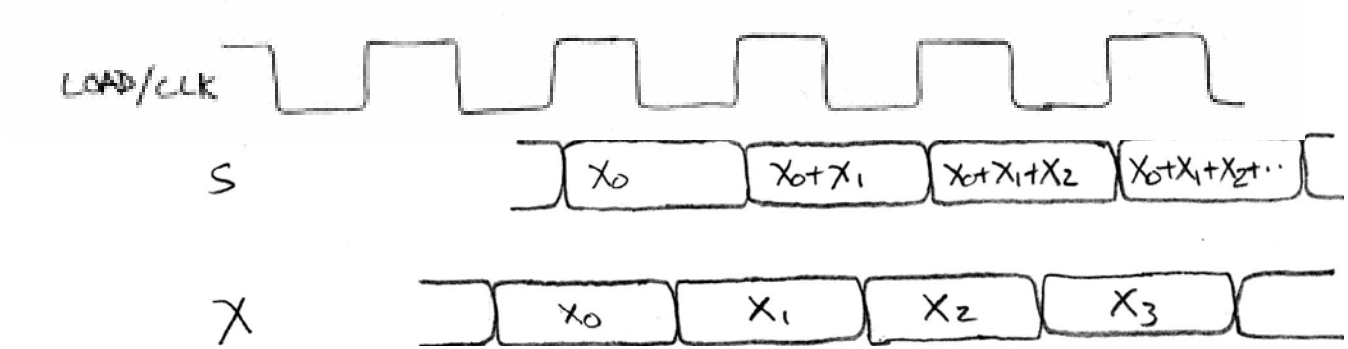


- **Register of size N:**
  - **n instances of D Flip-Flop**

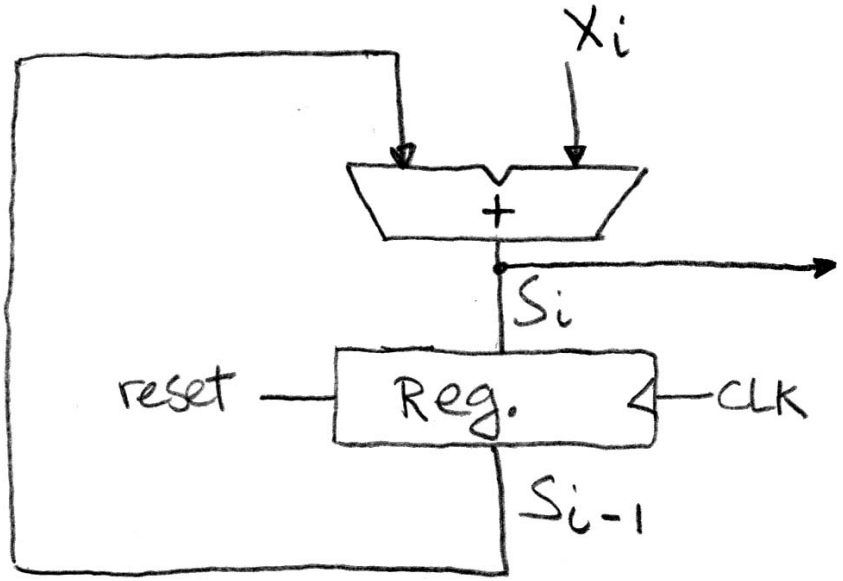
Second try…How about this? Yep!

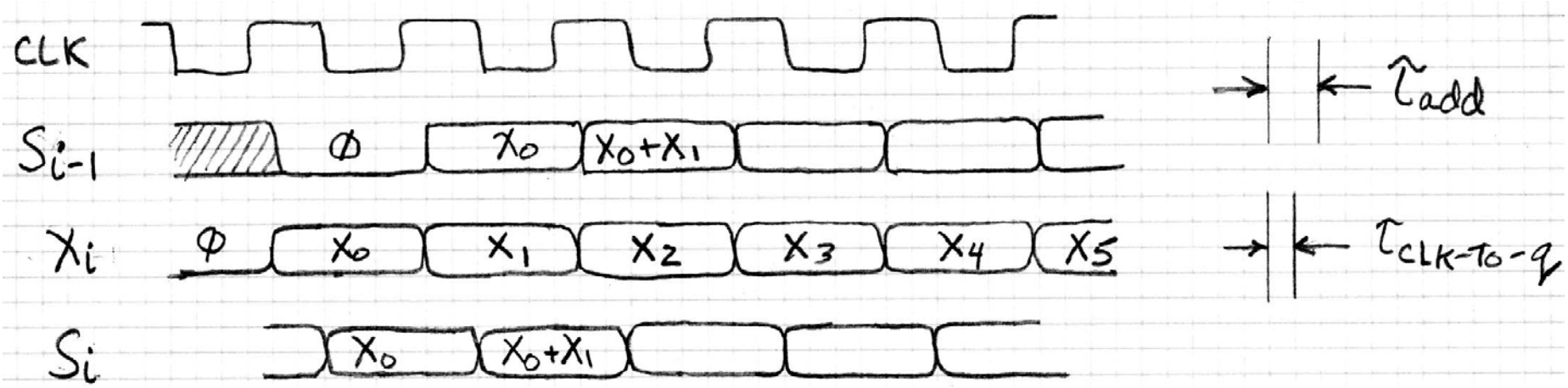


Rough timing…

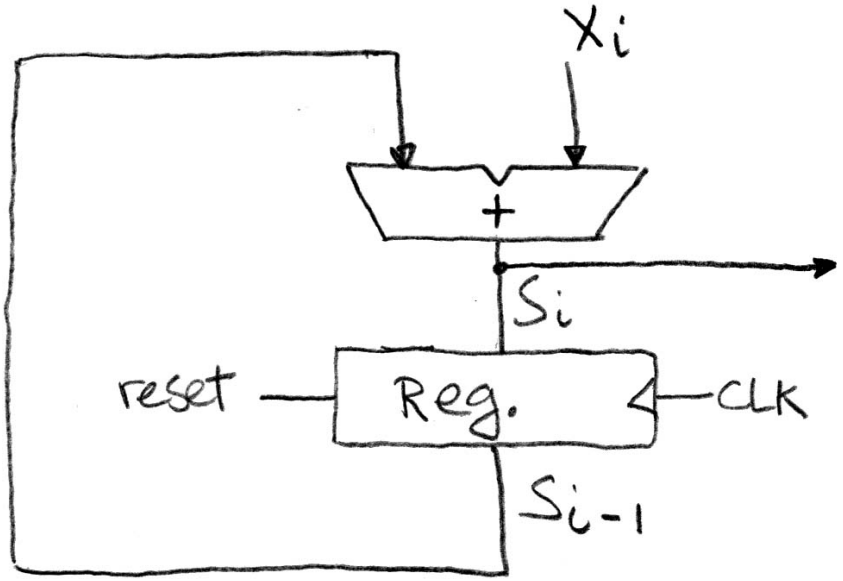


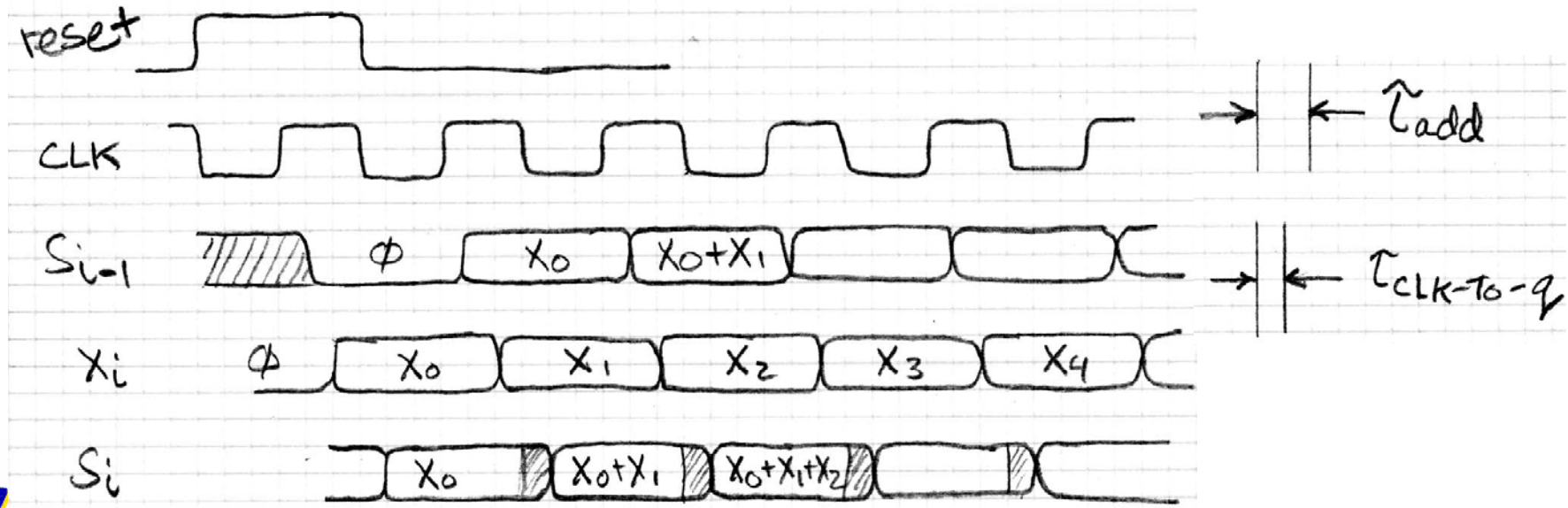
Accumulator Revisited (proper timing 1/2)



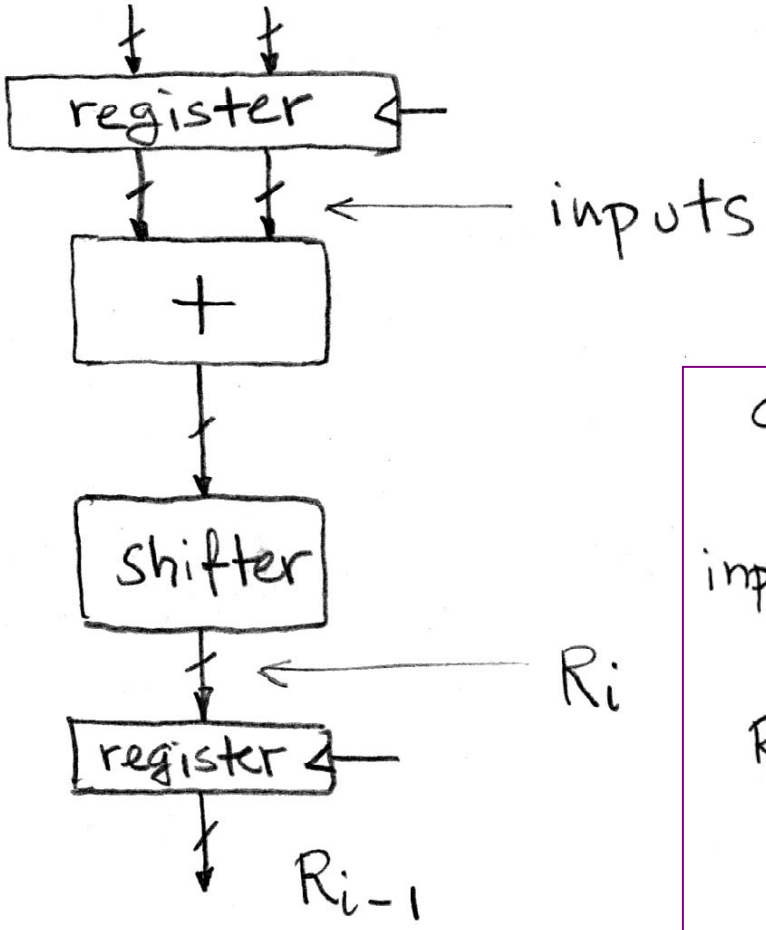


Accumulator Revisited (proper timing 2/2)

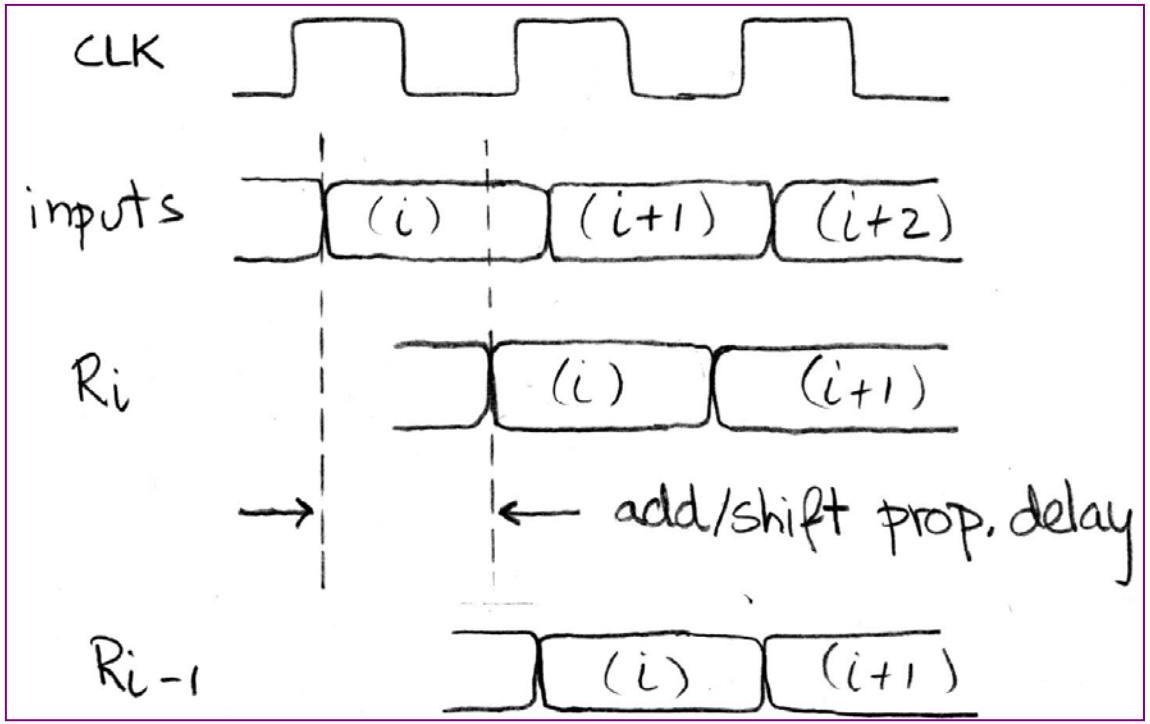




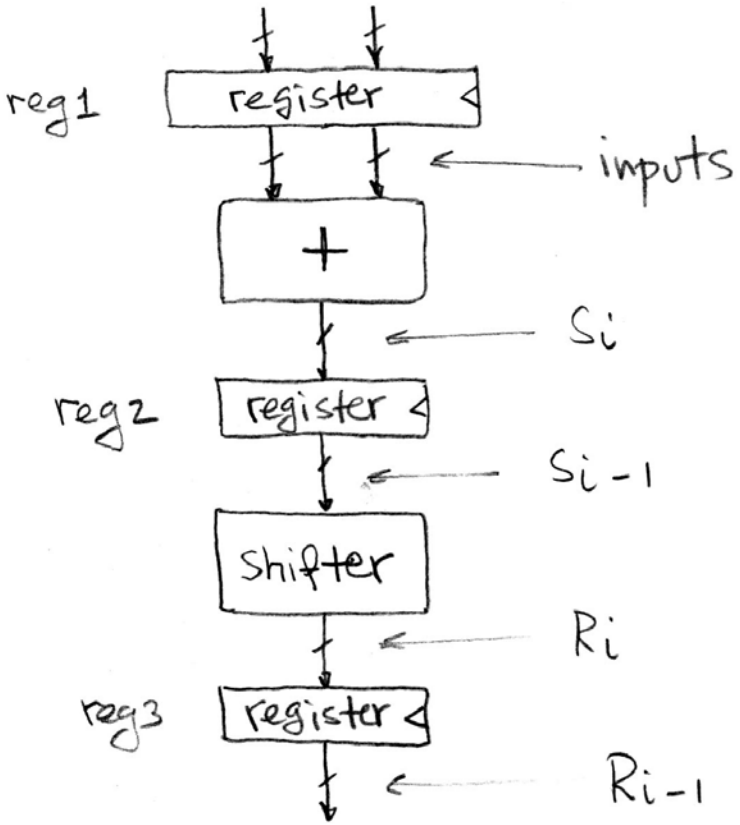
Pipelining to improve performance (1/2)



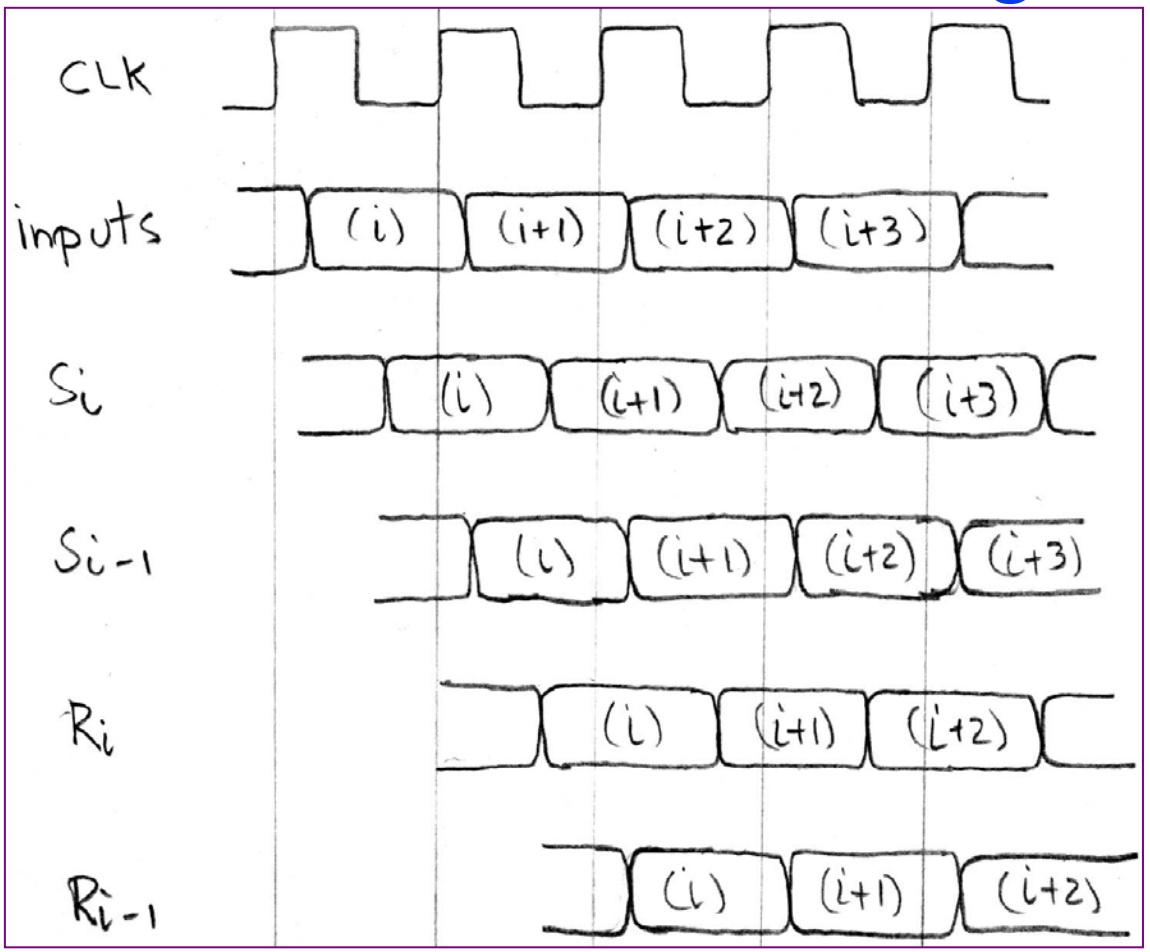
Timing…



Pipelining to improve performance (2/2)



Timing…



Peer Instruction 1

- Simplify the following Boolean algebra equation:

- Q = !(A*B) + !(!A * C)
- Use algebra, individual steps, etc.
  - Don't just look at it and figure it out, or I'll have to start using harder examples. ☺

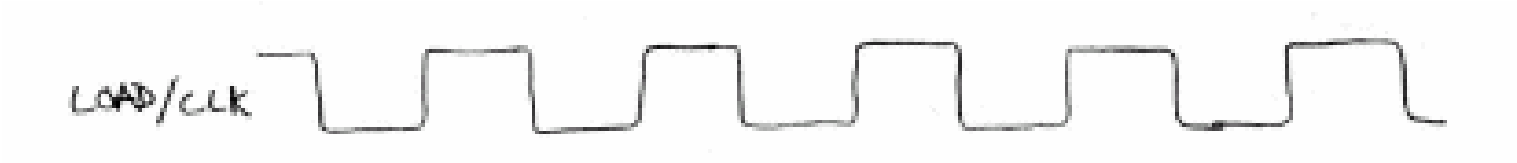
Administrivia

- HW 45 due Monday
- Project 2 will be released soon

- If you want to get a little bit ahead (in a moderately fun sort of way), start playing with Logisim:
  - http://ozark.hendrix.edu/~burch/logisim/

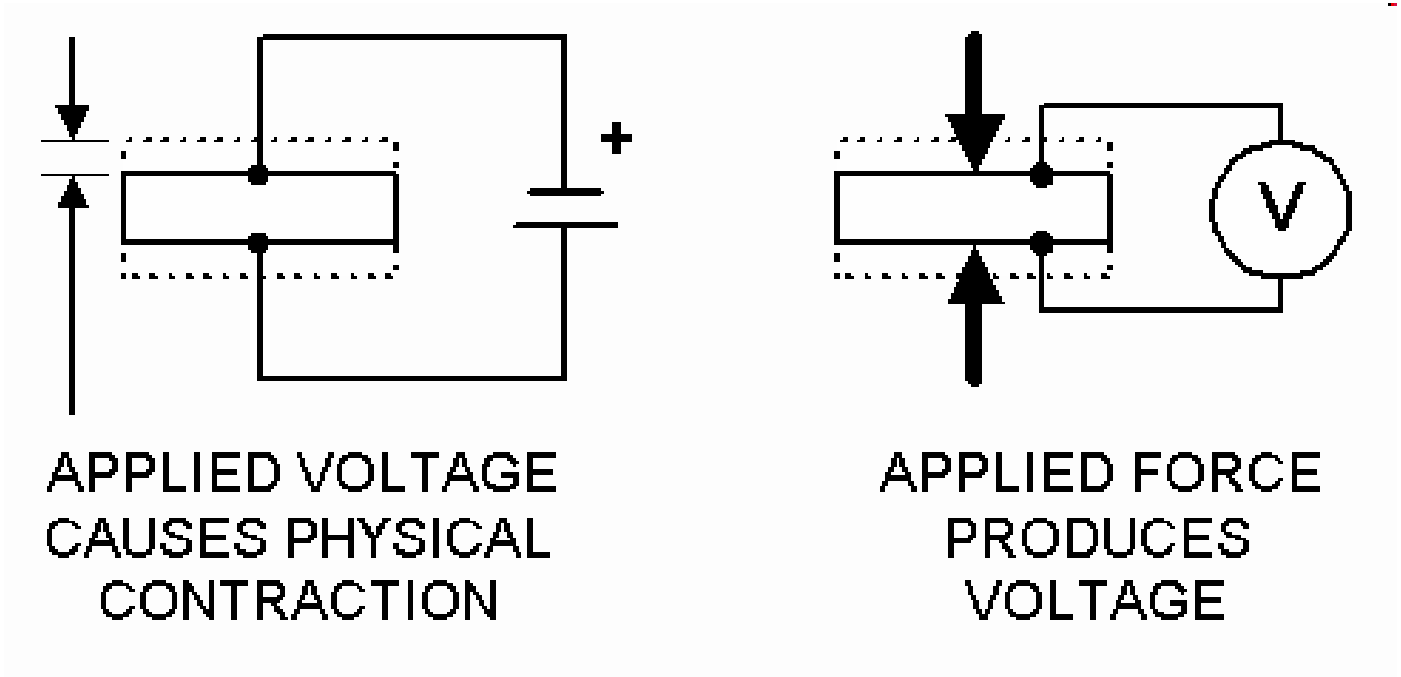
Clocks

- Need a regular oscillator:



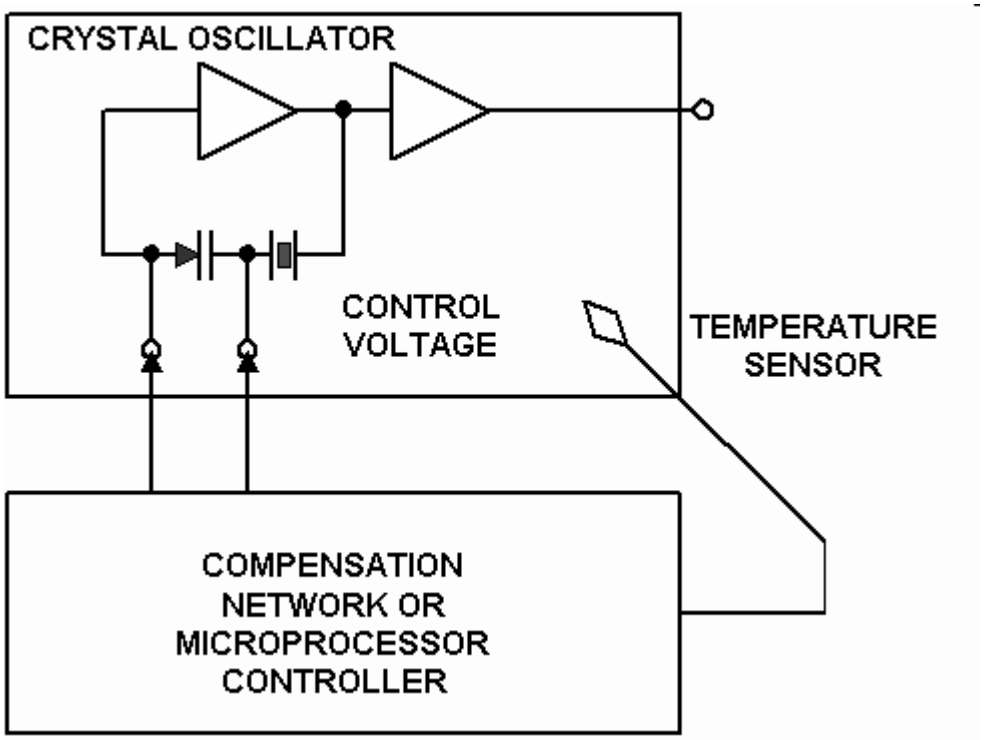
- Wire up three inverters in feedback?…
  - Not stable enough
  - 1->0 and 0->1 transitions not symmetric.
- Solution: Base oscillation on a natural resonance. But of what?

Clocks



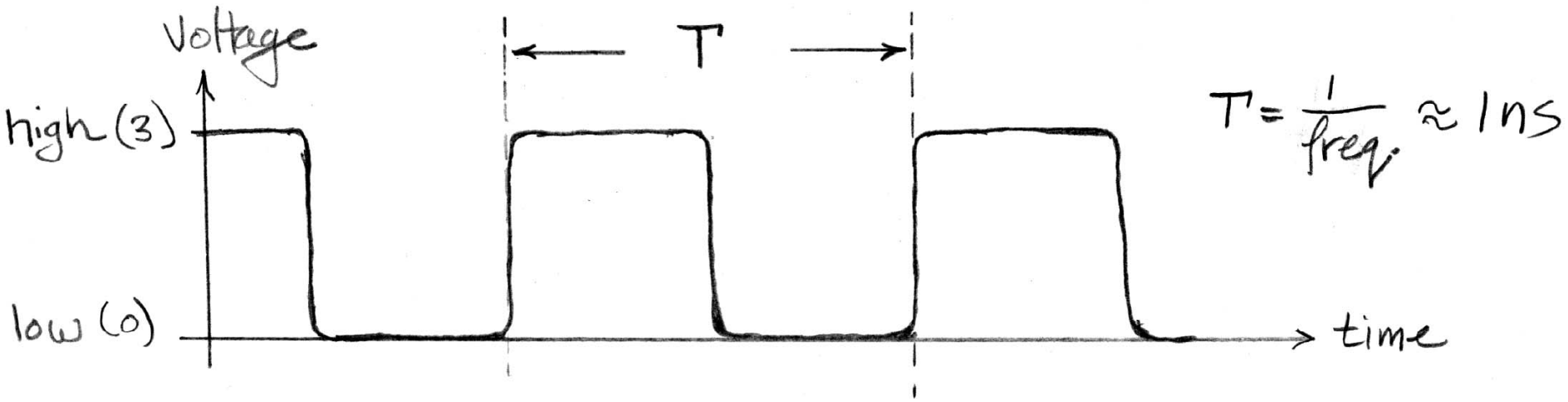
- **Crystals and the Piezoelectric effect:**
  - **Voltage → deformation → voltage → …**
  - **Deformations have a resonant freq.**
    - **Function of crystal cut**

Clocks



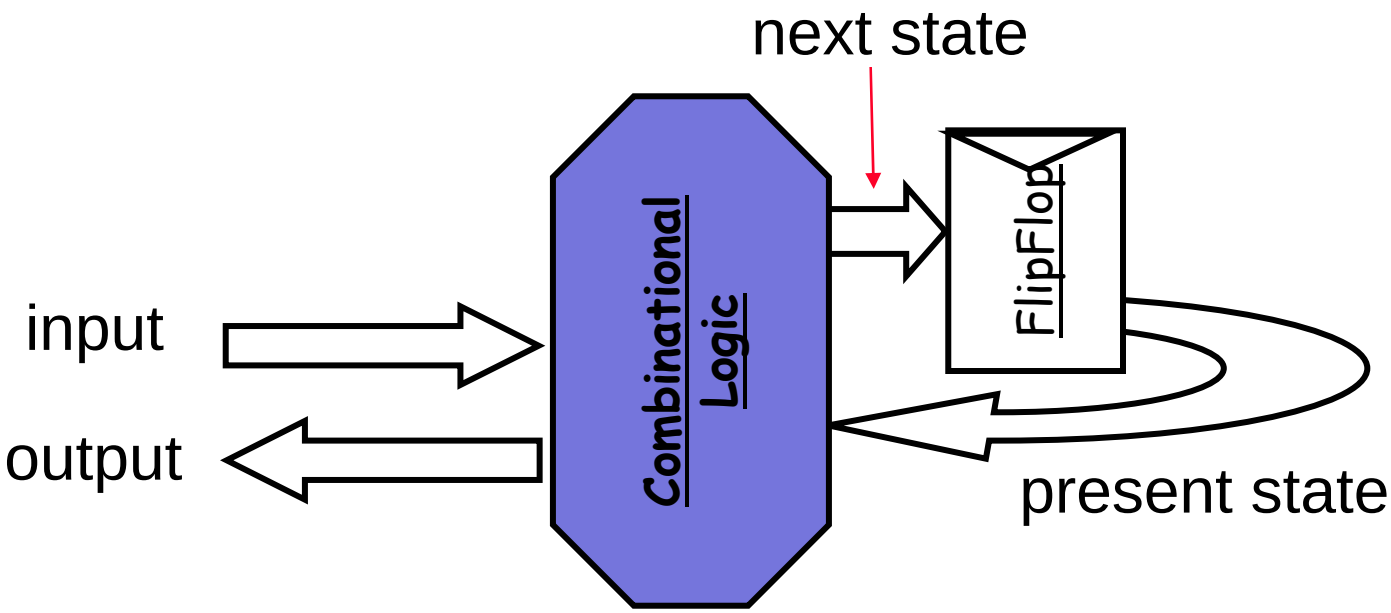
- **Controller puts AC across crystal:**
  - **At anything but resonant freqs → destructive interference**
  - **Resonant freq → CONSTRUCTIVE!**

Signals and Waveforms: Clocks



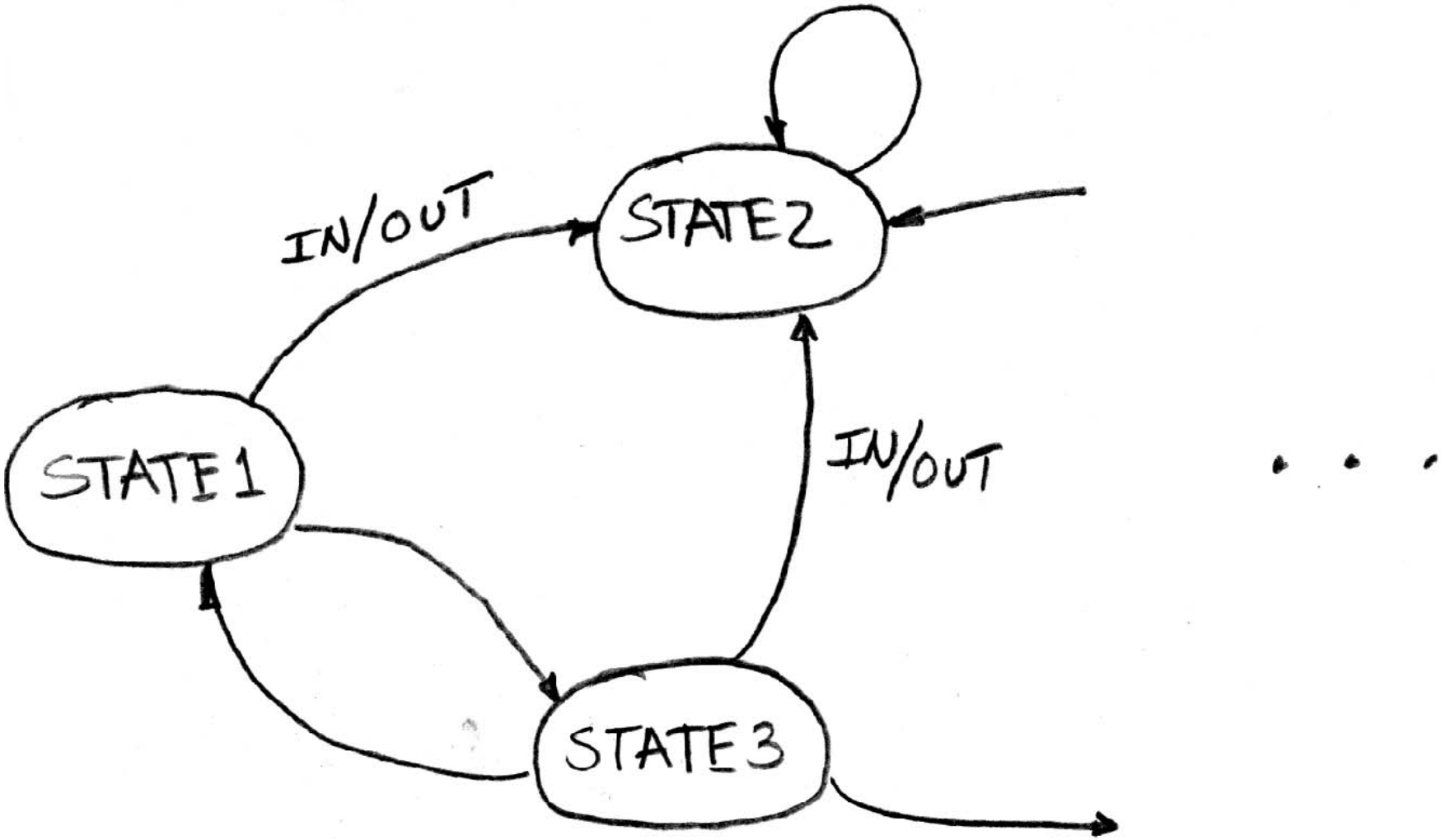
FSMs

- **With state elements, we can build circuits whose output is a function of inputs and current state.**

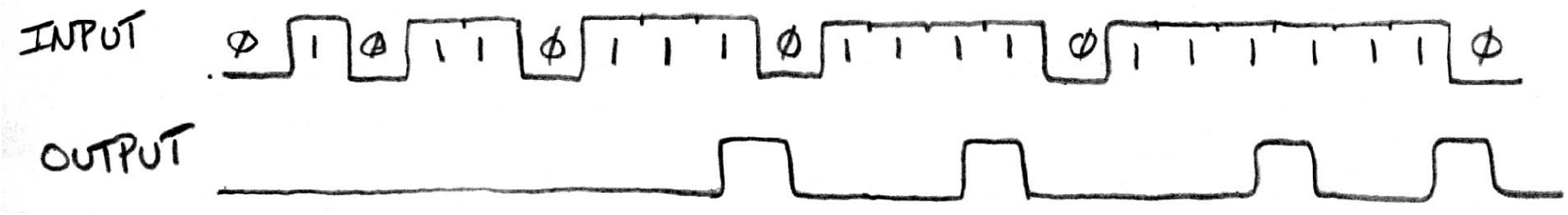


- **State transitions will occur on clock edges.**

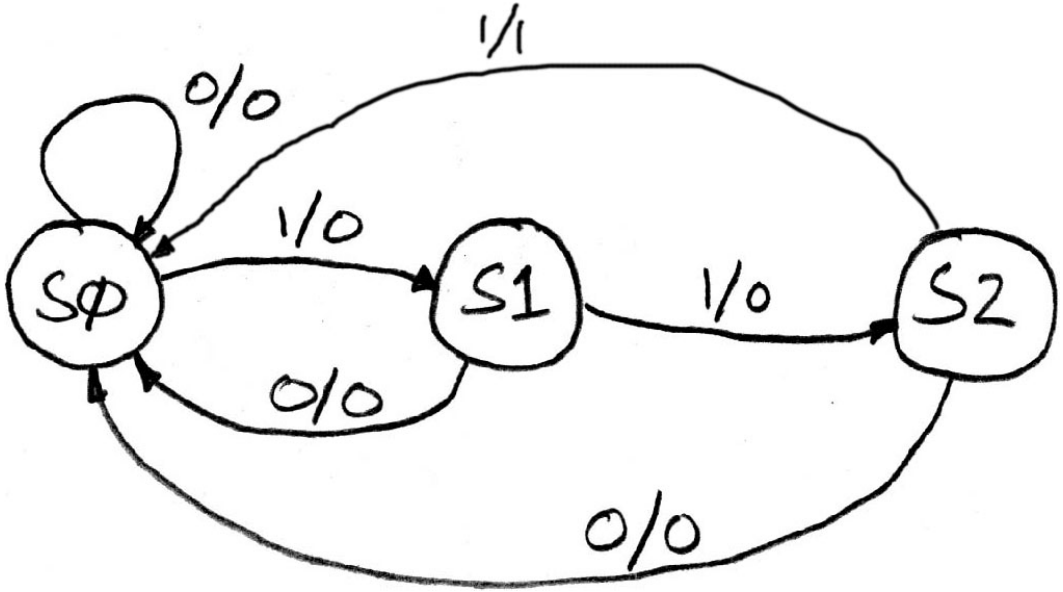
Finite State Machines Introduction



Finite State Machine Example: 3 ones…

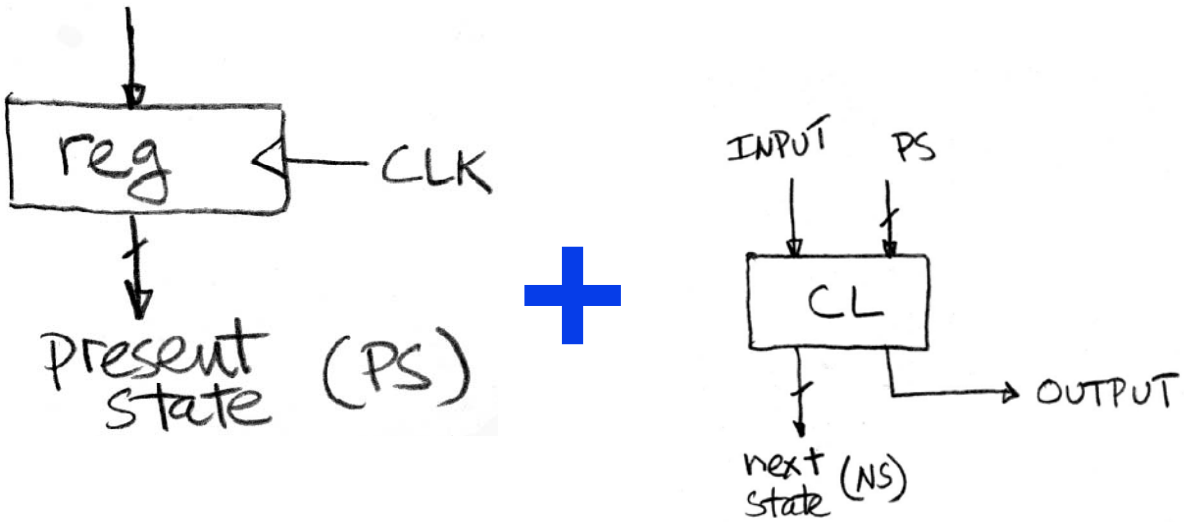


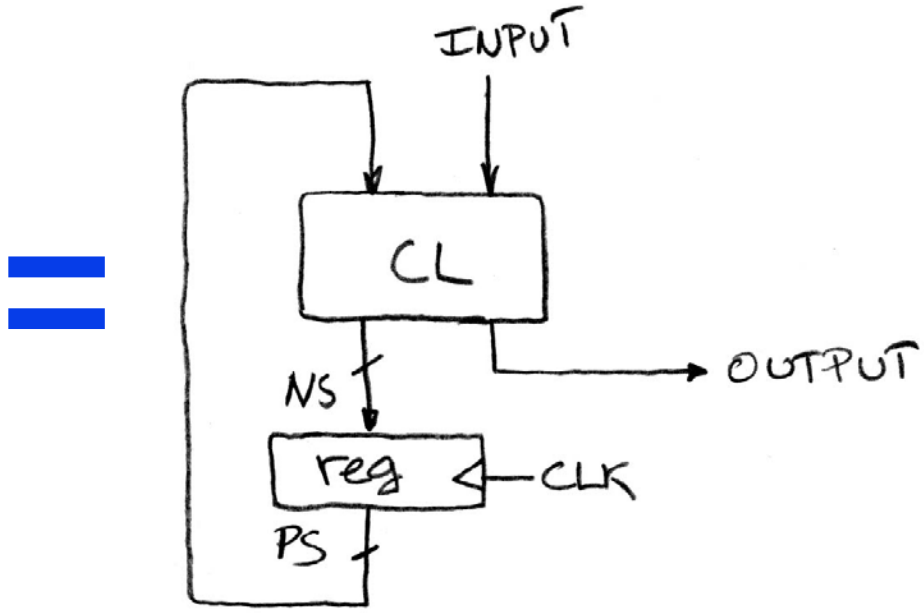
Draw the FSM…



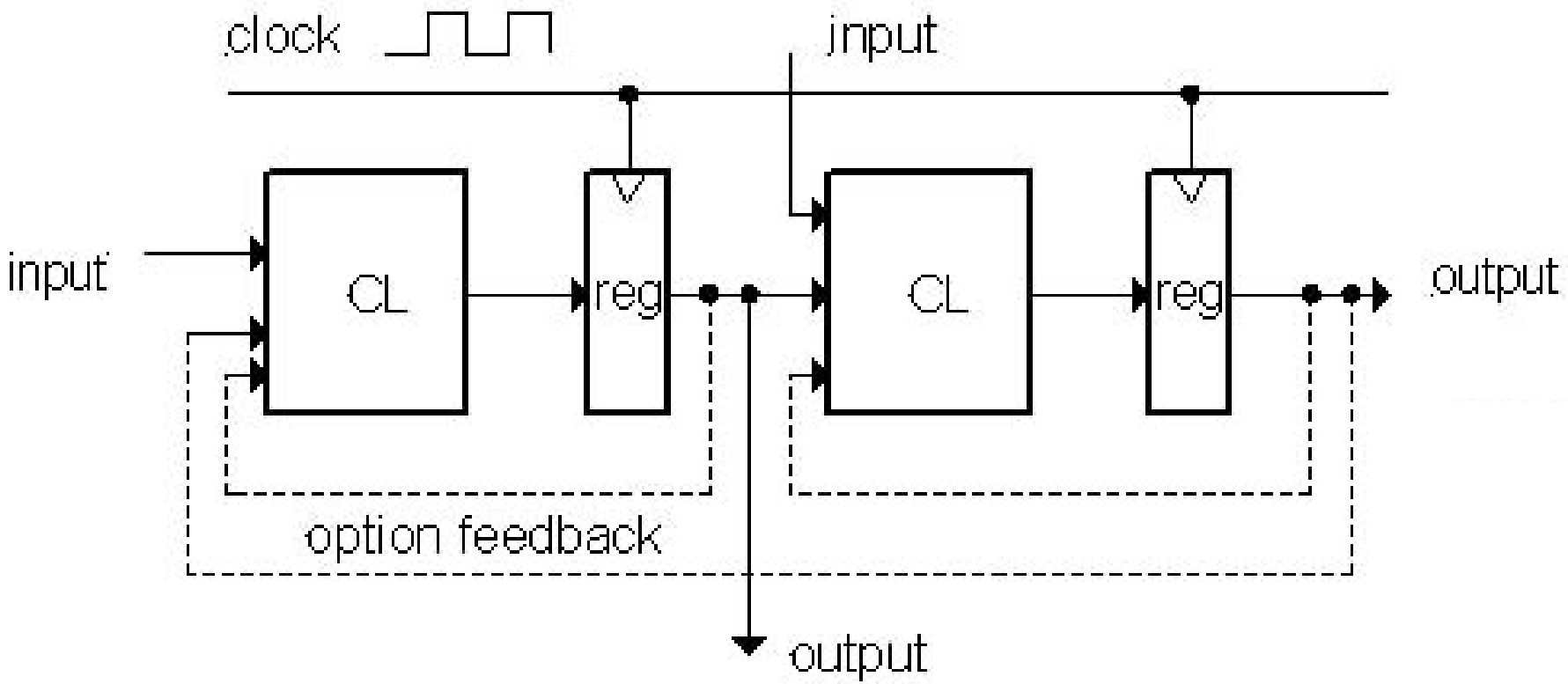
PS	Input	NS	Output
00	0	00	0
00	1	01	0
01	0	00	0
01	1	10	0
10	0	00	0
10	1	00	1

Hardware Implementation of FSM





General Model for Synchronous Systems



Peer Instruction 2

- Two bit counter:
  - 4 States: 0, 1, 2, 3
  - When input c is high, go to next state
    - (3->0)
  - When input is low, don't change state
  - On the transition from state 3 to state 0, output a 1. At all other times, output 0.