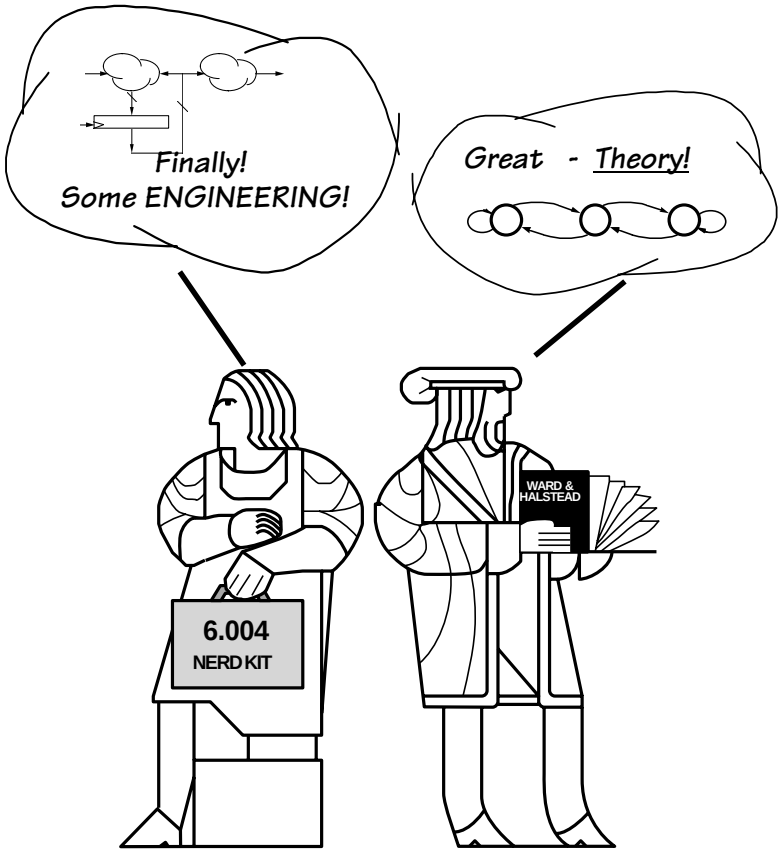
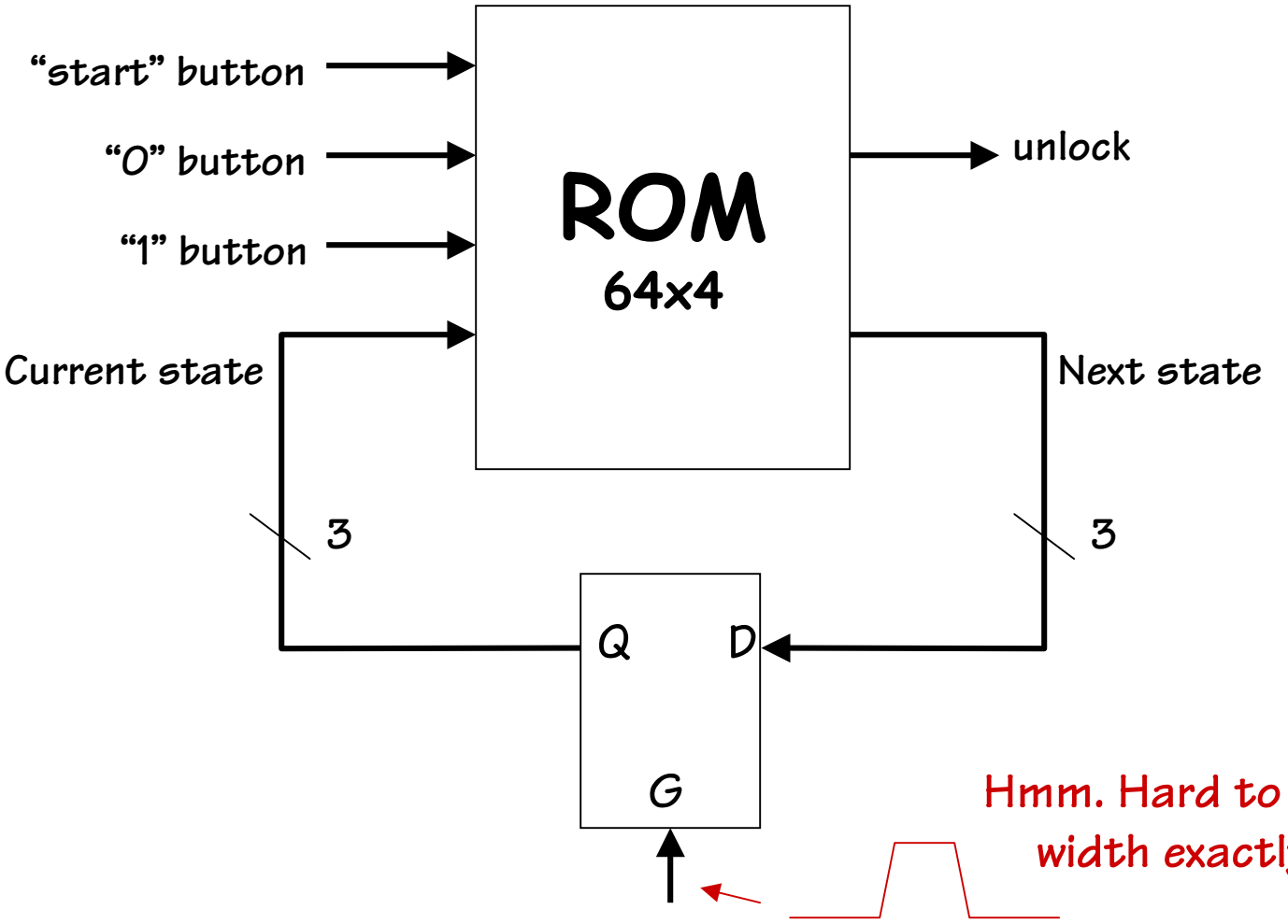
Finite State Machines



Handouts: Lecture Slides, PS4, Lab3

From Last Time...

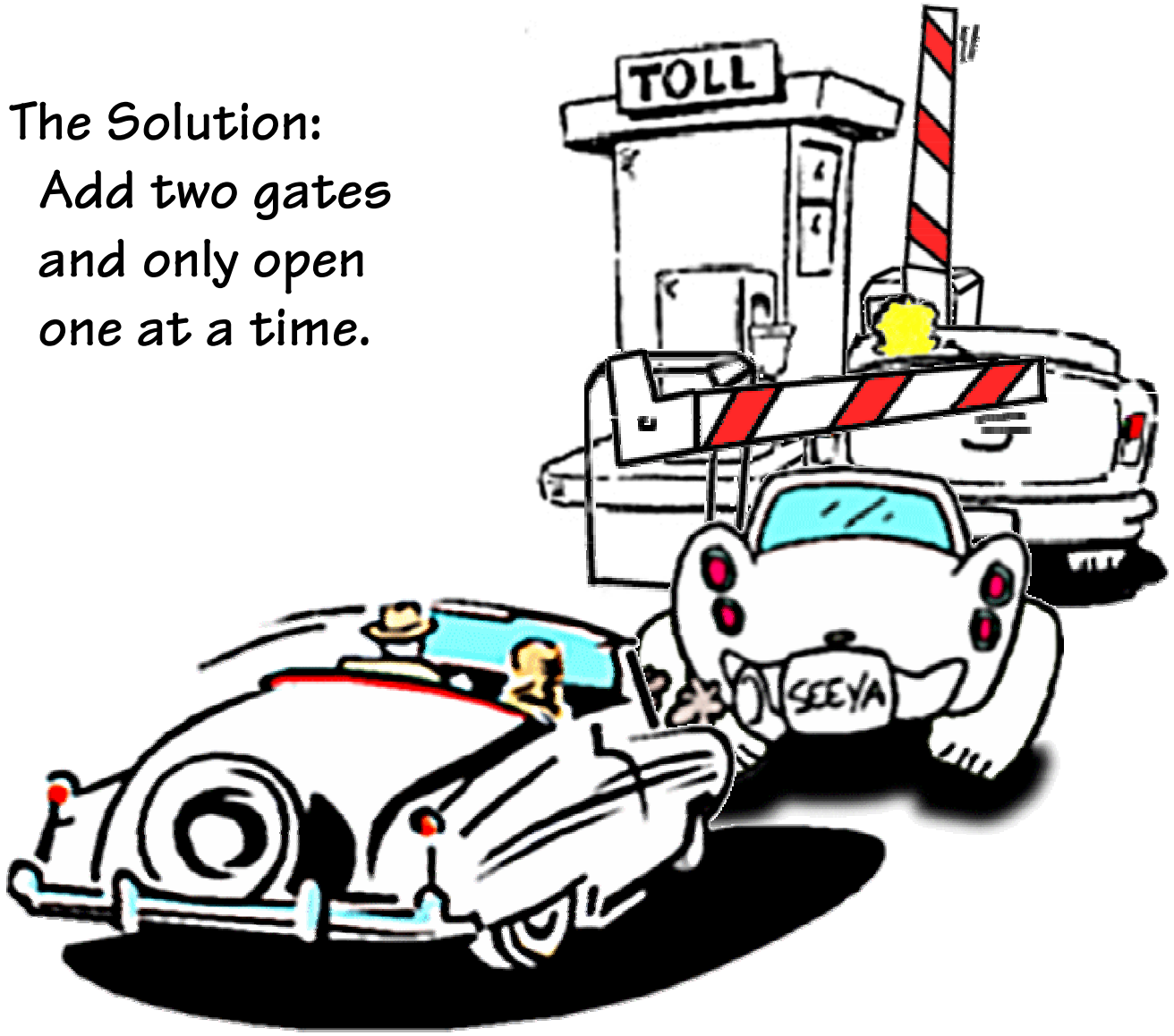
we still haven't gotten a working implementation of an FSM



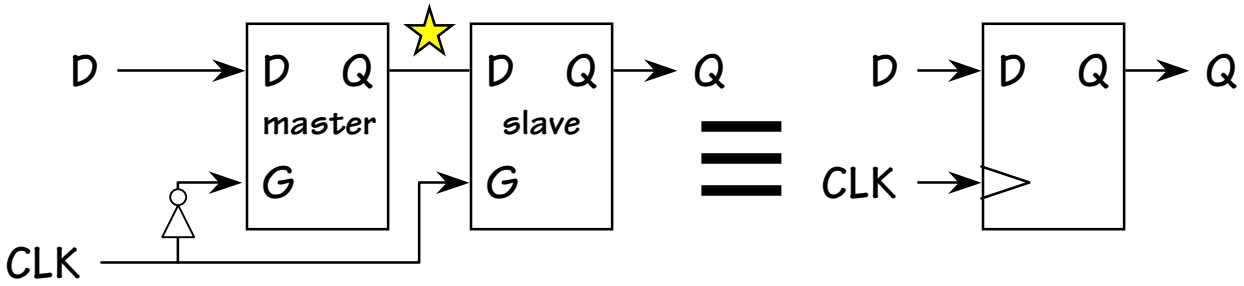
Two Gates are Better than One

The Solution:

Add two gates and only open one at a time.



Edge-triggered Flip Flop



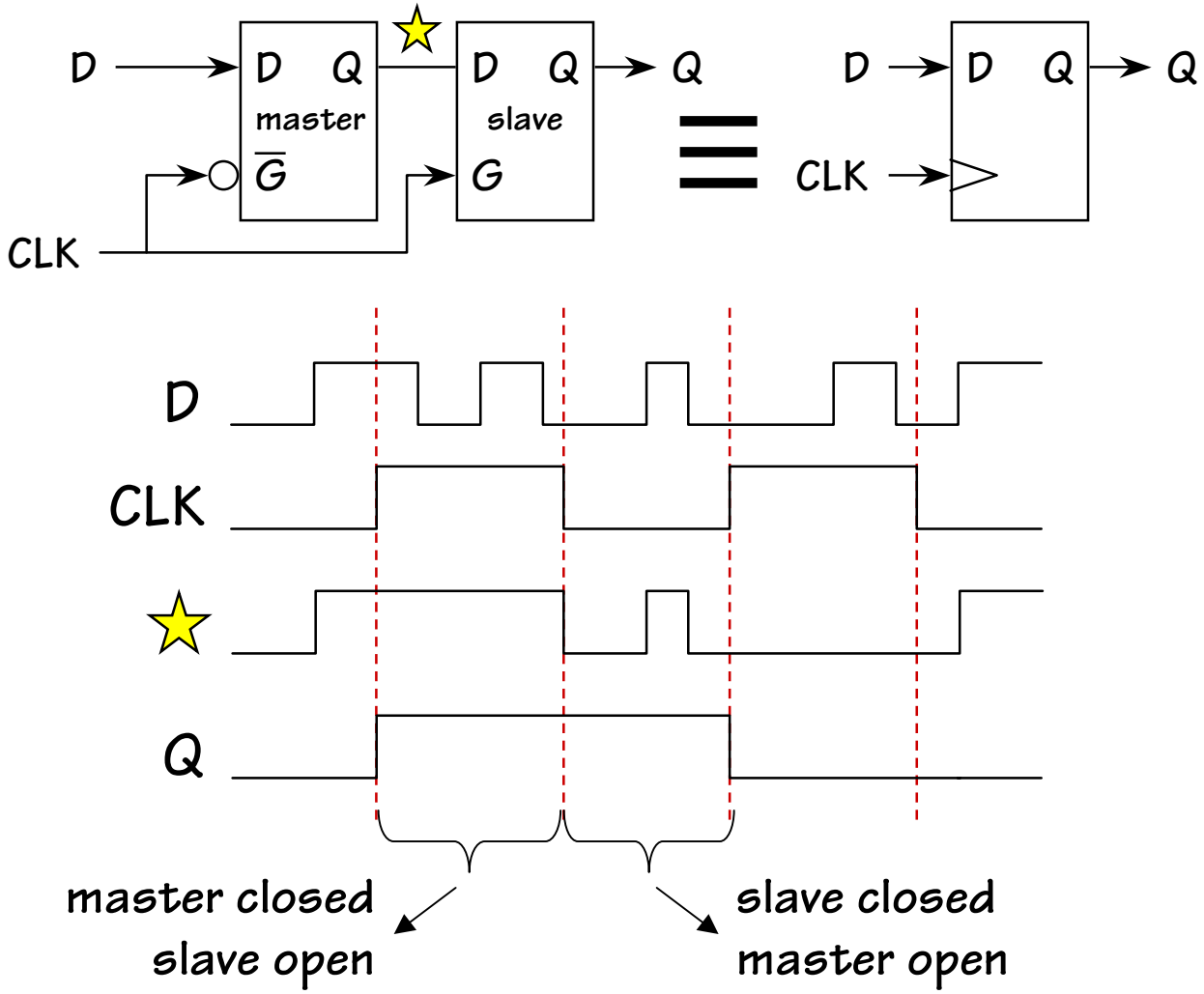
Observations:

- only one latch "transparent" at any time:
  - master closed when slave is open
  - slave closed when master is open

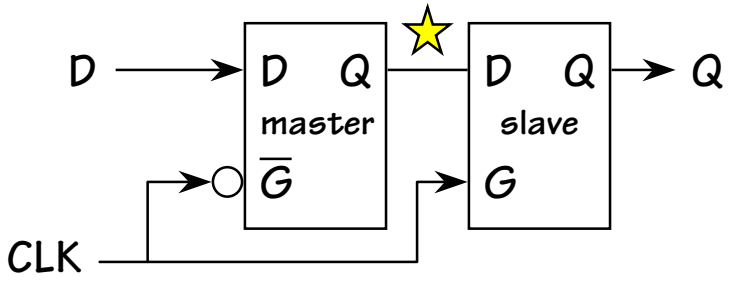
  → no combinational path through flip flop

- Q only changes shortly after 0 →1 transition of CLK, so flip flop appears to be "triggered" by rising edge of CLK

Flip Flop Waveforms



Um, about that hold time...

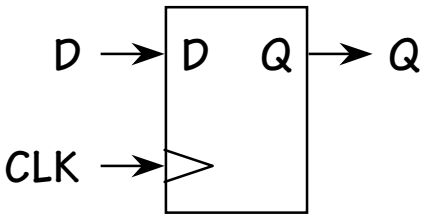


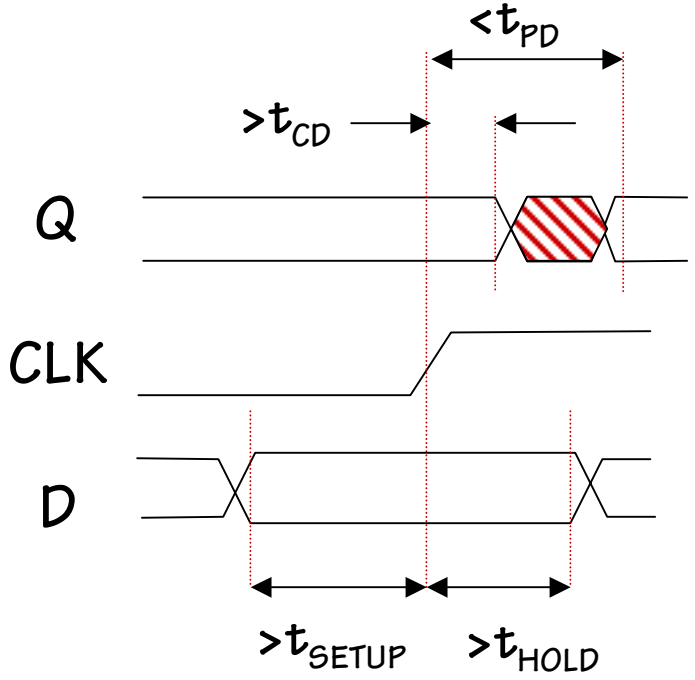
Consider HOLD TIME requirement for slave:

- Negative (1 →0) clock transition → slave freezes data:
  - SHOULD be no output glitch, since master held constant data; BUT
  - master output contaminated by change in G input!
- HOLD TIME of slave not met, UNLESS we assume sufficient contamination delay in the path to its D input!

Accumulated t_{CD} thru inverter, G → Q path of master must cover slave t_{HOLD} for this design to work!

Flip Flop Timing - I





t_{PD}: maximum propagation delay, CLK →Q

t_{CD}: minimum contamination delay, CLK →Q

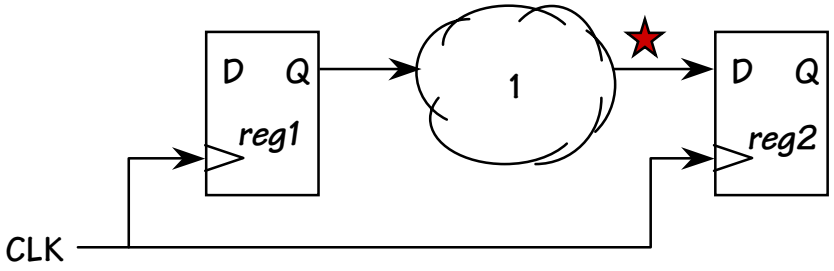
t_{SETUP}: setup time

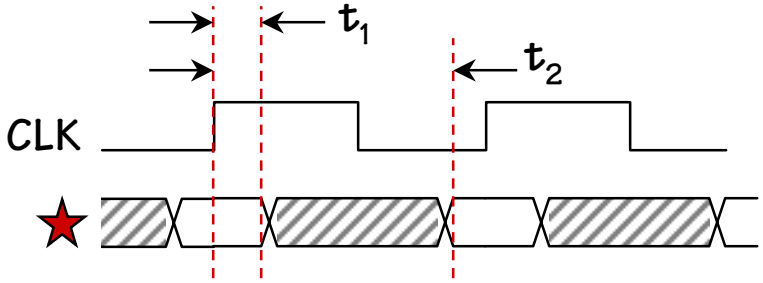
guarantee that D has propagated through feedback path before master closes

t_{HOLD}: hold time

guarantee master is closed and data is stable before allowing D to change

Flip Flop Timing - II





$$t_1 = t_{CD,reg1} + t_{CD,1} > t_{HOLD,reg2}$$

$$t_2 = t_{PD,reg1} + t_{PD,1} < t_{CLK} - t_{SETUP,reg2}$$

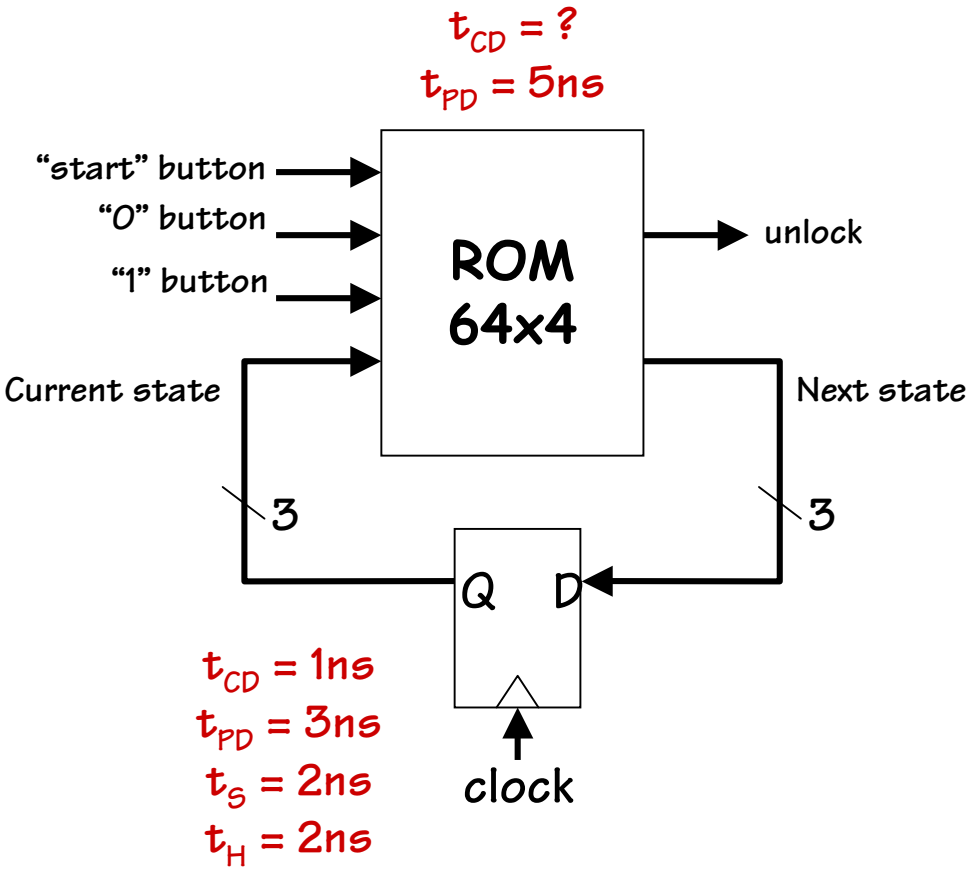
Questions for register-based designs:

- how much time for useful work (i.e. for combinational logic delay)?
- does it help to guarantee a minimum t_{CD}? How 'bout designing registers so that

  $$t_{CD,reg} > t_{HOLD} \quad ?$$

- what happens if CLK signal doesn't arrive at the two registers at exactly the same time (a phenomenon known as "clock skew")?

Example: Flip Flop Timing



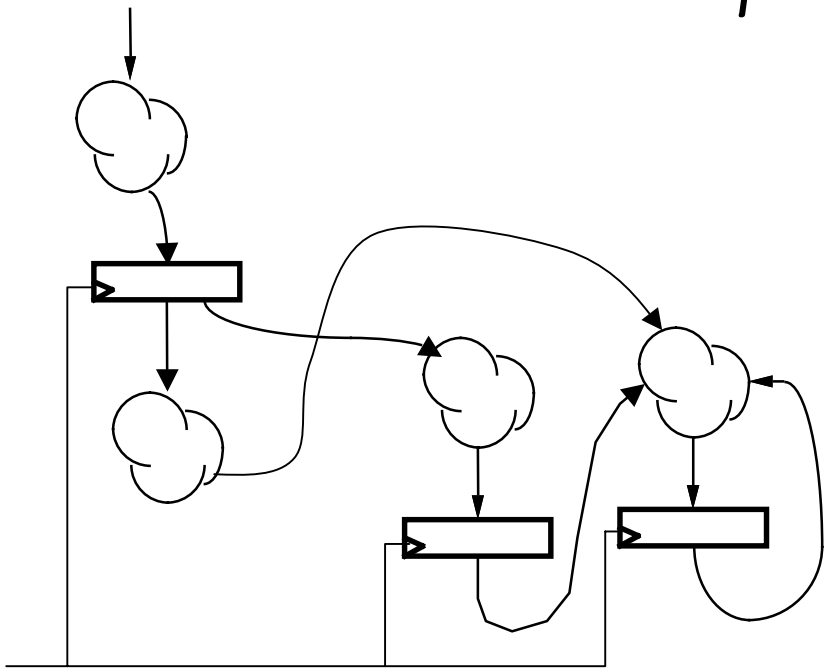
Questions:

1. t_{CD} for the ROM?

2. Min. clock period?

3. Constraints on inputs?

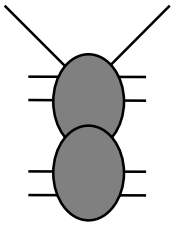
Single Synchronous Clock



A recipe for robust sequential circuits:

- No combinational cycles
- Only care about value of combinational circuits just before rising edge of clock
- Period greater than every combinational delay
- Change saved state after noise-inducing logic transitions have stopped!

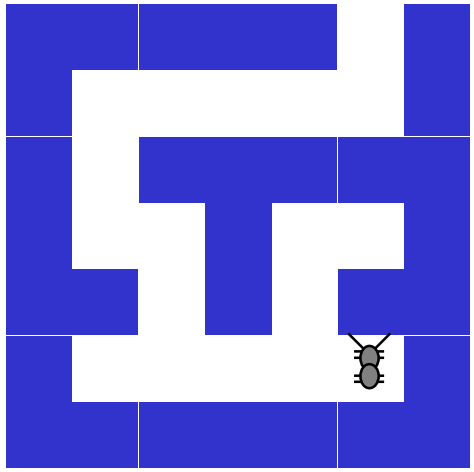
Example FSM: Roboant®



SENSORS: antennae L and R, each 1 if in contact with something.

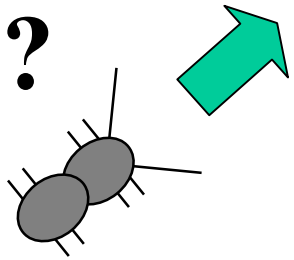
ACTUATORS: Forward Step F, ten-degree turns TL and TR (left, right).

GOAL: Make our ant smart enough to get out of a maze like:

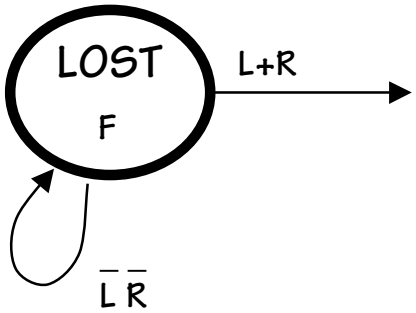


STRATEGY: "Right antenna to the wall"

Lost in space

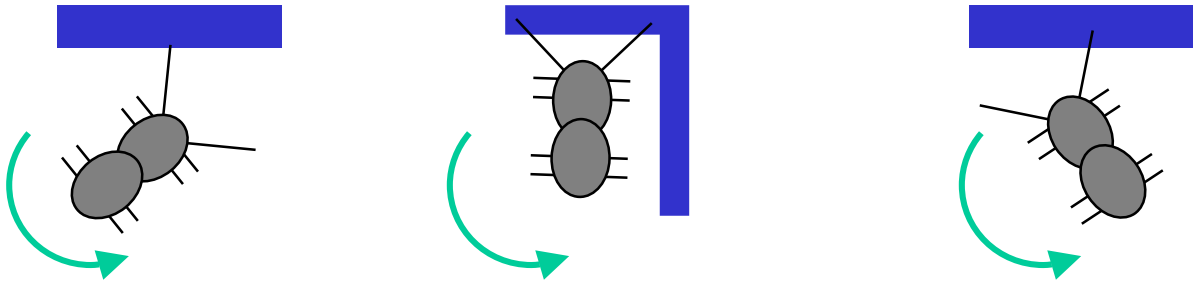


Action: *Go* forward until we hit something.

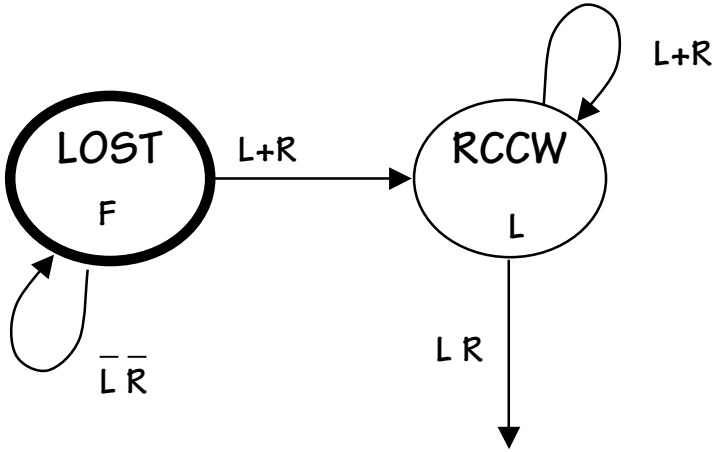


"lost" is the initial state

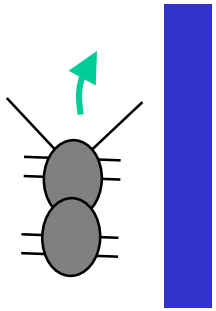
Bonk!



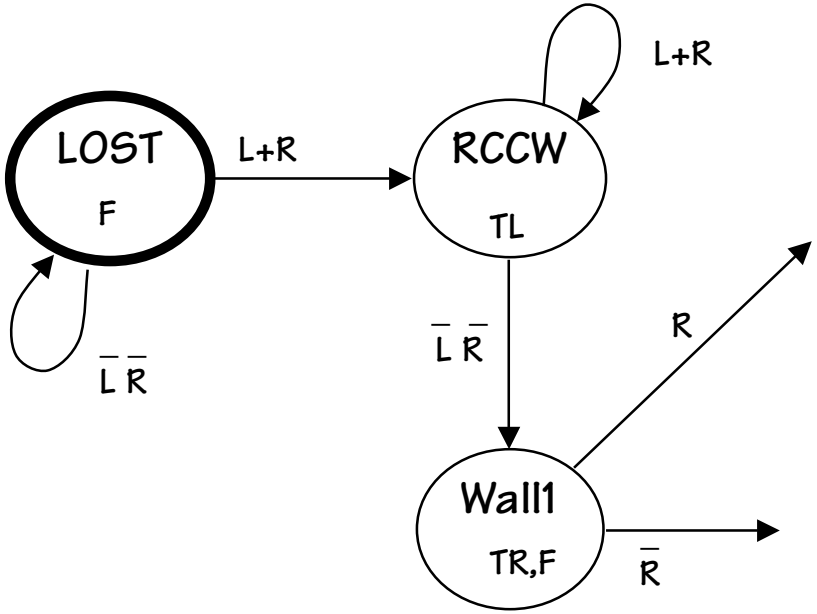
Action: Turn left (CCW) until we don't touch anymore



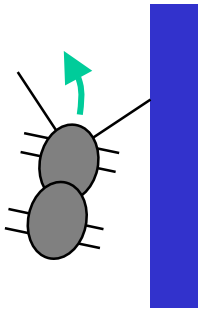
A little to the right…



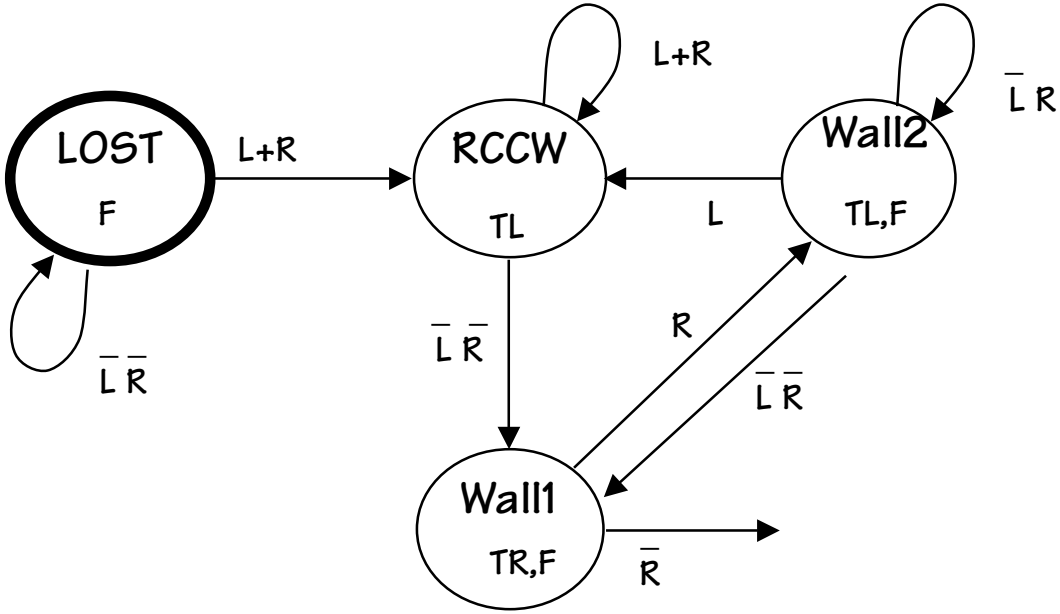
Action: Step and turn right a little, look for wall



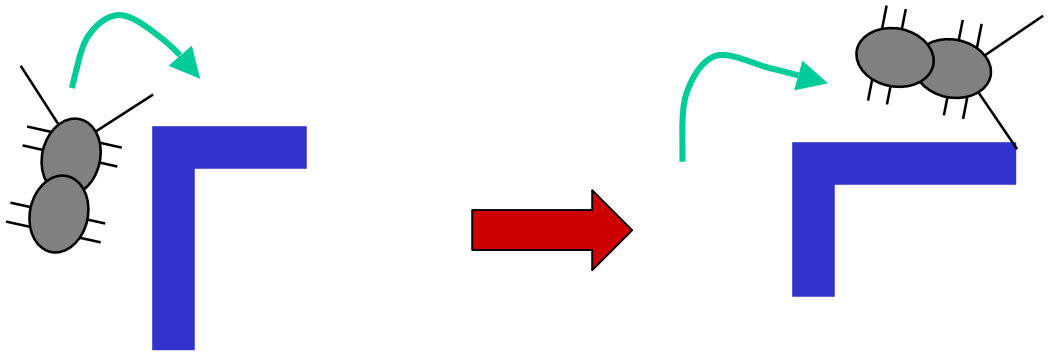
Then a little to the left



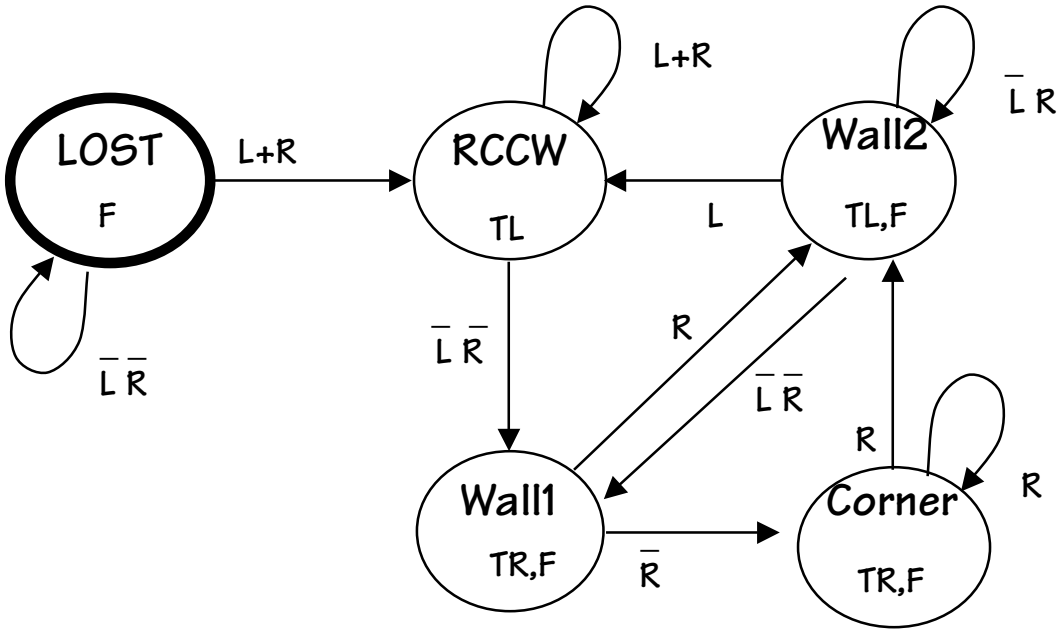
Action: Step and turn left a little, till not touching (again)



Dealing with corners



Action: Step and turn right until we hit perpendicular wall



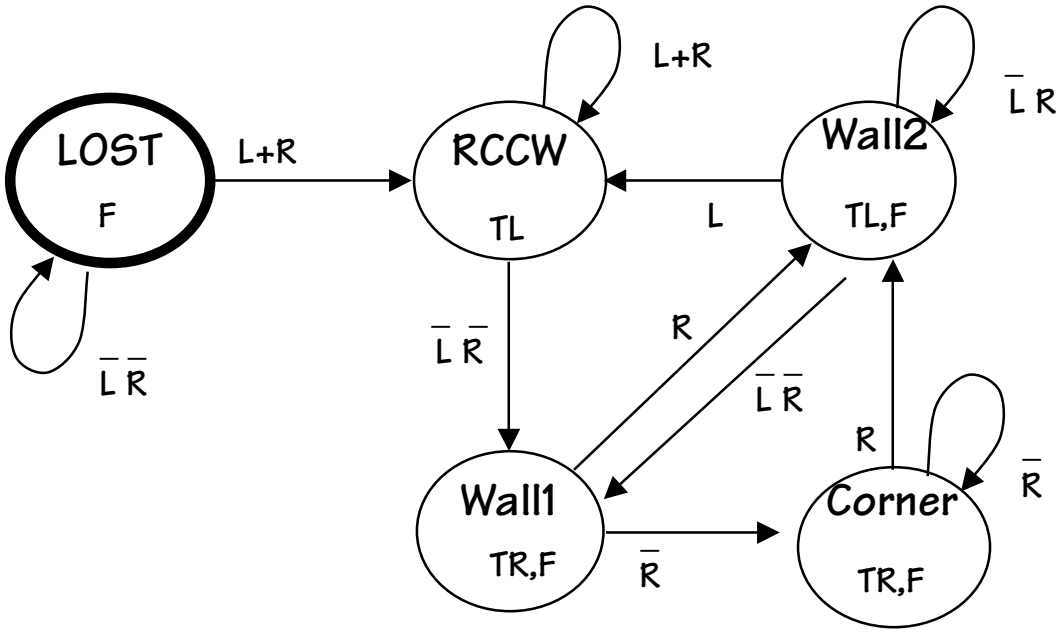
Equivalent State Reduction

Observation: $S_i \equiv S_j$ if

1. States have identical outputs; AND
2. Every input $\rightarrow$ equivalent states.

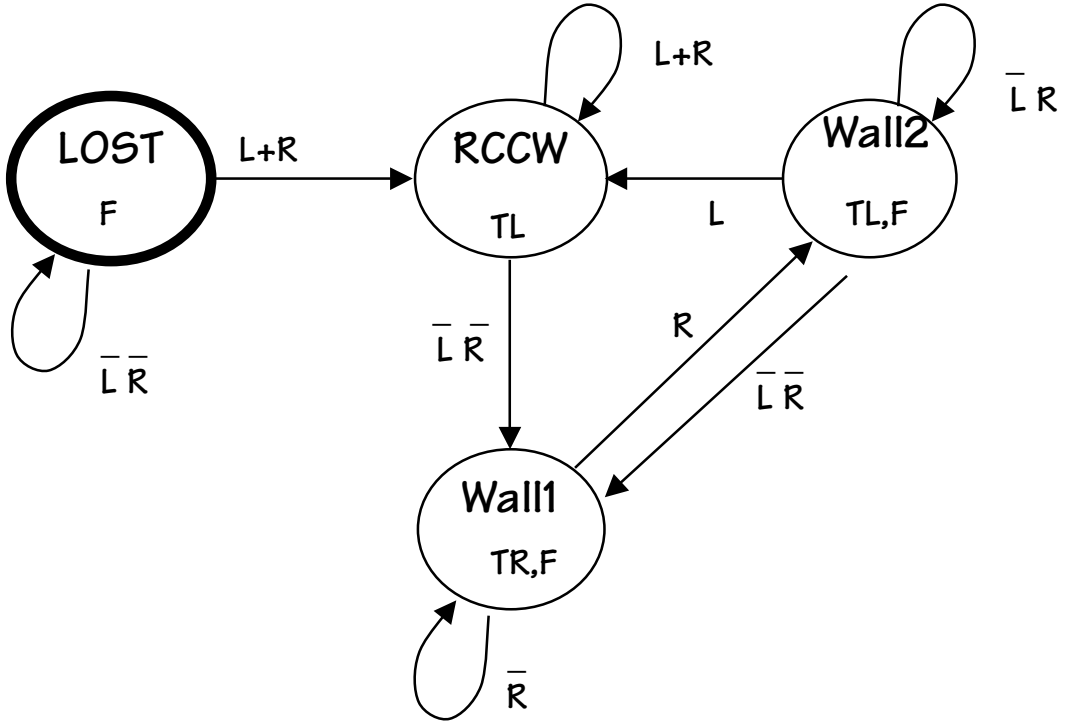
Reduction Strategy:

Find pairs of equivalent states, MERGE them.



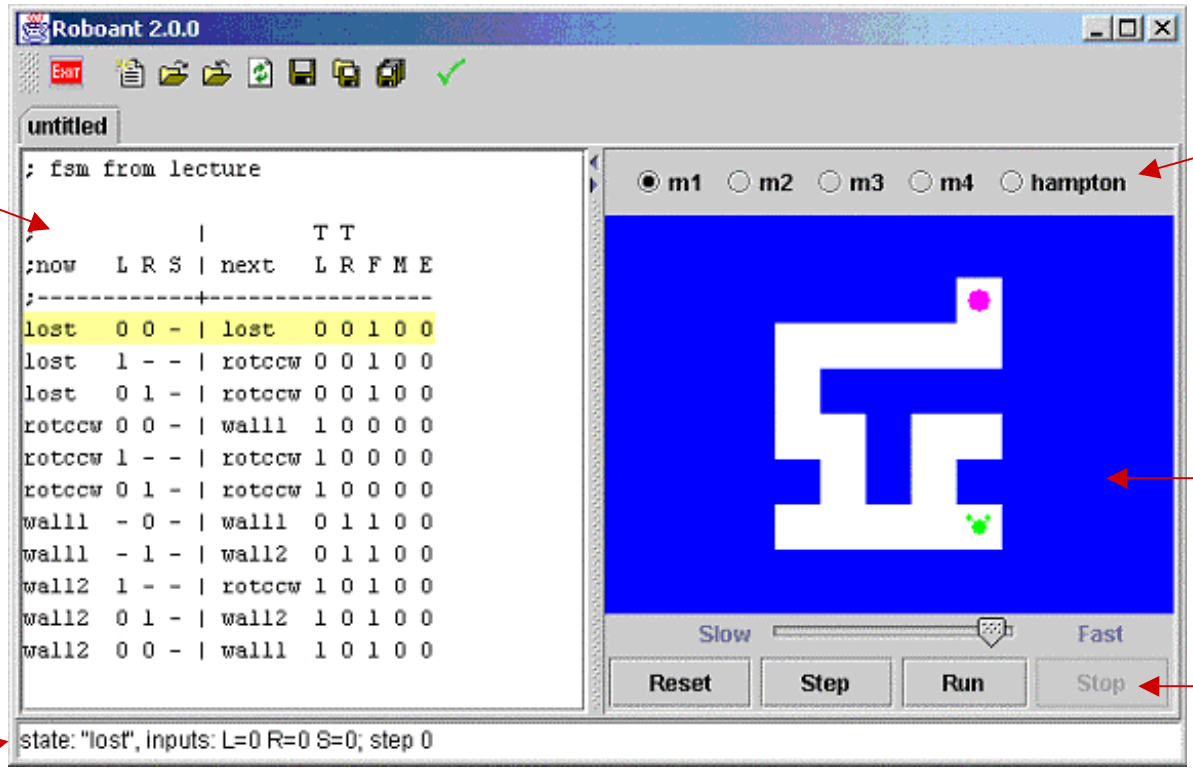
An Evolutionary Step

Merge equivalent states Wall1 and Corner into a single new, combined state.



Behaves exactly as previous (5-state) FSM, but requires <u>half</u> the ROM in its implementation!

Roboant®



FSM state table

Maze selection

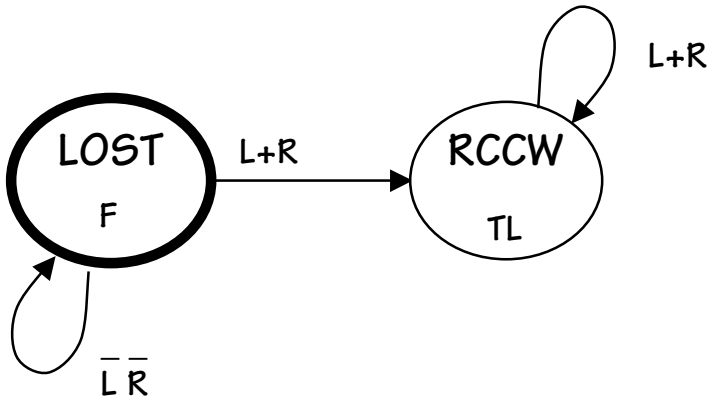
Plan view of maze

Simulation controls

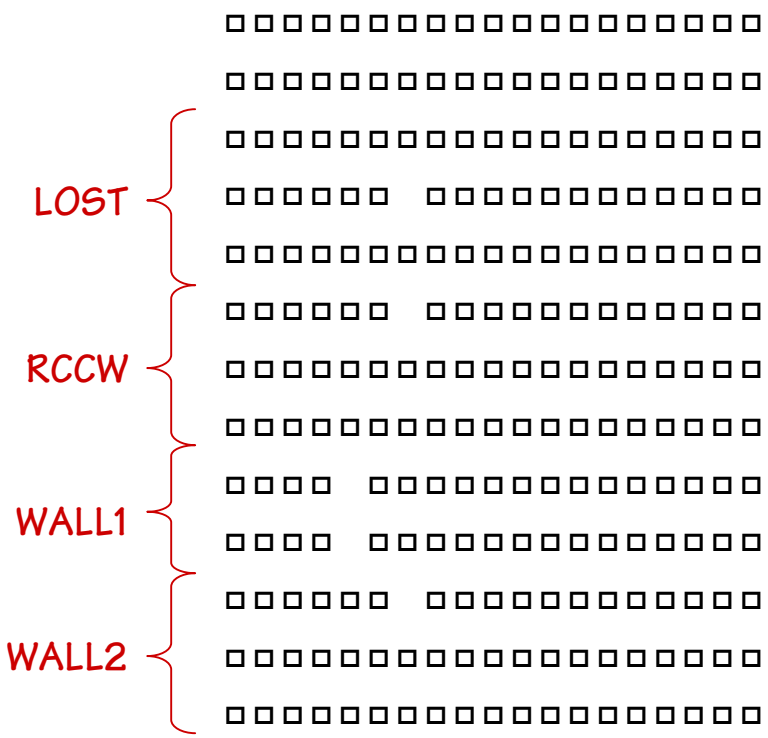
Status display

Featuring the new Mark-II ant: can add (M), erase (E), and sense (S) marks along its path.

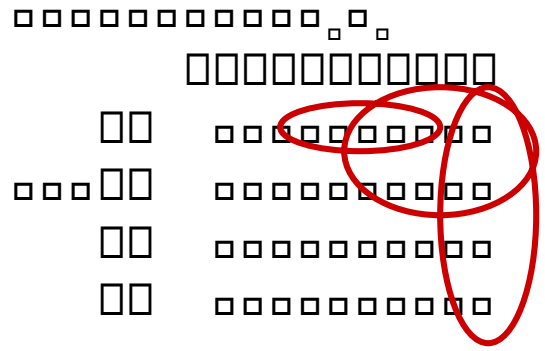
Building the Transition Table



Implementation Details



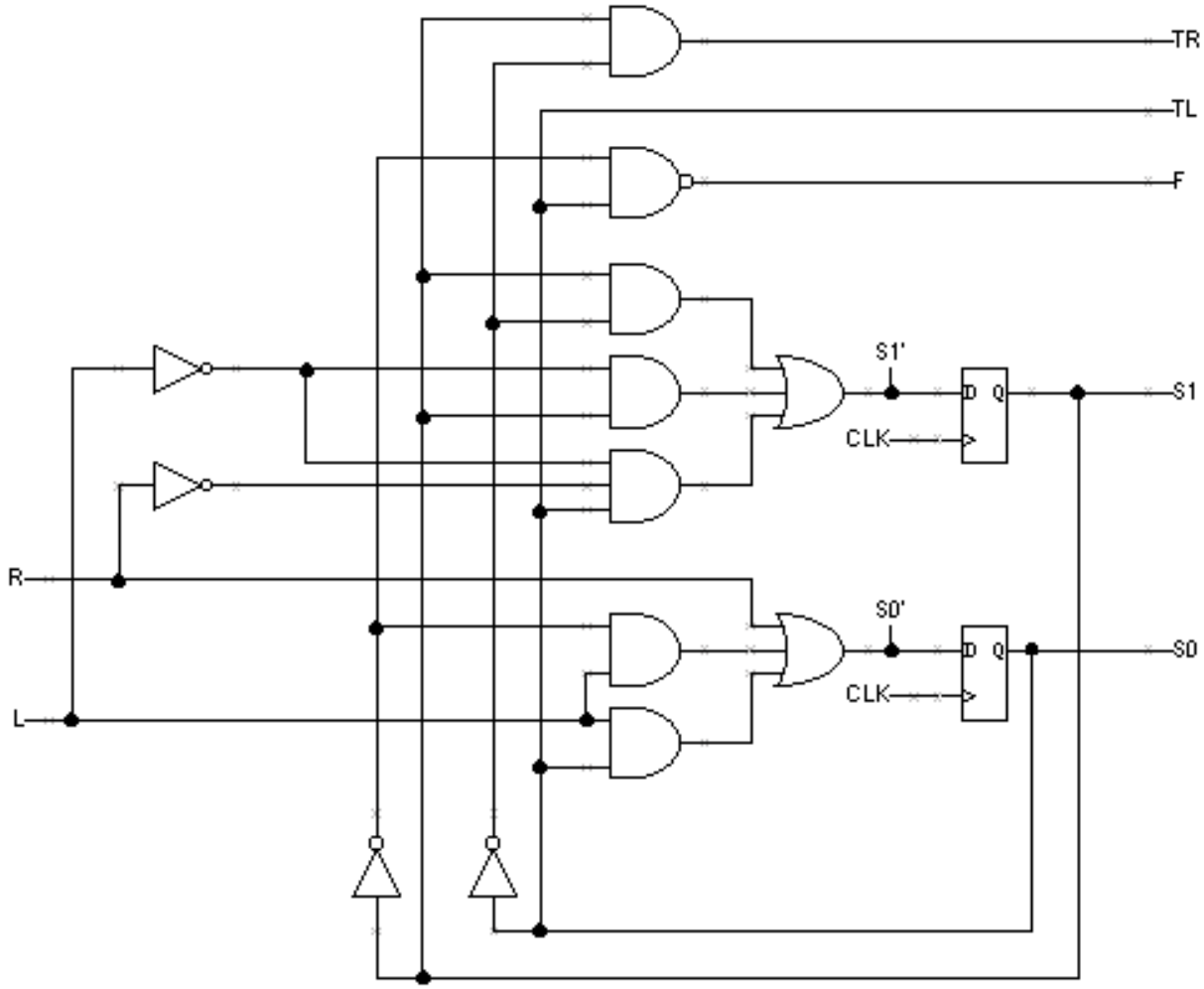
Complete Transition table



$$S_1' = S_1\overline{S_0} + \overline{L}S_1 + \overline{L}\,\overline{R}S_0$$

$$S_0' = R + L\overline{S_1} + LS_0$$

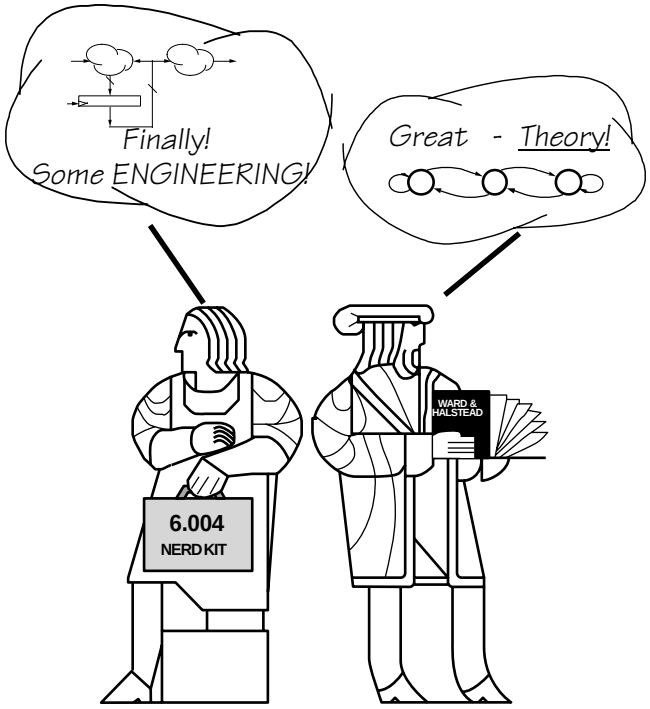
Ant Schematic



Finite State Machines: An important abstraction

- FORMAL MODEL:
  bstantial literat rebt eoreti al res lts
- PRACTICAL IMPACT:
  Major engineering tool…
- METAPHYSICAL IMPACT:
  Are we all just complicated FSMs?
  Perhaps with imperfect state registers and "fuzzy" logic…

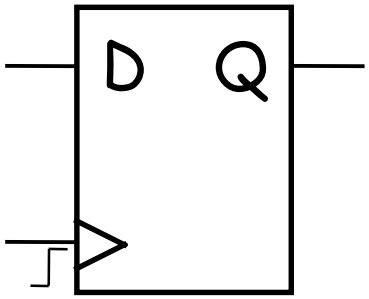
… You'll see FSMs for the rest of your life!



Summary

Synchronous 1-clock logic:

In order to build RELIABLE circuits we need only follow these simple rules.



An essential element of the DYNAMIC DISCIPLINE is the specification of setup and hold times relative to a global clock edge.

1) Cloc period greater than: $FF(t_{pd})+Logic(t_{pd})+FF(t_{su})$
2) Satisfy hold time: $FF(t_{cd})+Logic(t_{cd})>FF(t_h)$
3) o combinational cycles