State Machines

(see section 7-14 in text for related material)

Exercise 1: Why do I need to know this?

for Lab 5 & Lab Exam #1

Exercise 2: In theory, what is a SM? What isn't a SM?

- anything with memory is a SM (in theory)
- if no memory (FF, registers) then it's not a SM.

Exercise 3: How large a SM is it practical to design? (how many states)

≃ 1 page
< 20 ? states.

} if larger, then decompose to smaller SMs.

Exercise 4: What are some things we could design as SMs?

- elevator
- traffic light
- alarms

Exercise 5: What are some things we would not design as a SM? Why?

counter – too many states

CPU – too complicated. (but parts of the CPU would be)

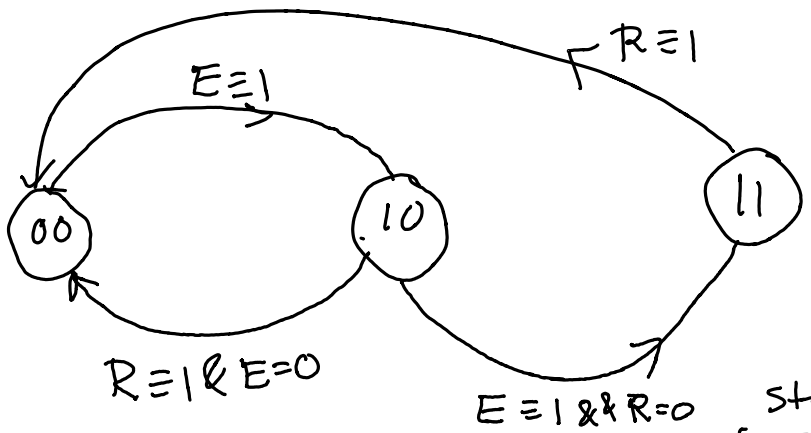
Exercise 6: What defines the state of a SM?

the contents of registers / FFs.

Exercise 7: How can we describe the operation of a SM?

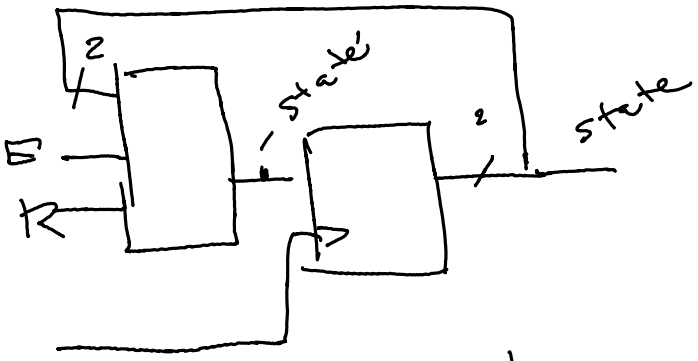
E = enable?
R = reset?

① as a state transition diagram



② as a table

state	input	state'
	E R	
0 0	1 X	1 0
1 0	0 1	0 0
	1 0	1 1
1 1	X 1	0 0



③ as a circuit

Exercise 8: Can a design have multiple state machines?

Yes, very common.

Exercise 9: How can one SM control another?

through their states or their state transitions (events).

Exercise 10: What are the steps in designing a SM?

Exercise 11: Show me some examples!

} see lecture 6.