Reading 14

15.4 FSMs to circuits (design)

- An FSM converts to a sequential circuit known as a controller, consisting of a register and combinational logic.

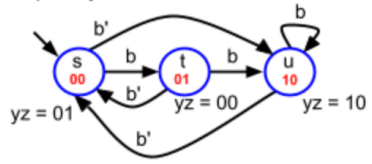
- A state register is the register in a controller, holding an FSMs present state.
  - ↳ Each state requires a unique bit encoding.

- Converting behavior (like an FSM) to a circuit is called design.
  - ↳ In contrast, converting a circuit to behavior is called analysis.

- Controllers inputs are typically the normal inputs AND the present state bits.
- Controllers outputs are typically the normal outputs AND the next state bits.

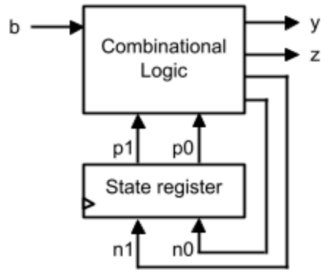
- Designers convert FSM to sequential circuit by first doing FSM → truth table, then truth table → equations, then equations → combinational circuit

Inputs: b
Outputs: y, z



*State encodings are in red

	p1	p0	b	n1	n0	y	z
s	0	0	0	1	0	0	1
	0	0	1	0	(F)	0	1
t	0	1	0	0	0	(C)	(D)
	0	1	1	1	0	0	(E)
u	1	0	0	(G)	0	1	0
	1	0	1	1	0	1	0
unused	1	1	0	0	0	0	0
	1	1	1	0	0	0	0



c = 0

d = 0

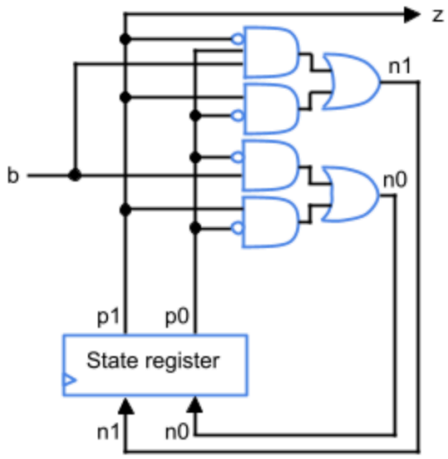
e = 0

f = 1

g = 0

15.6 Circuits to FSMs (analysis)

- Converting from circuit to behavior (like an FSM) is called analysis.



p1	p0	b	n1	n0	z
0	0	0	0	0	(Q)
0	0	1	0	1	0
0	1	0	(R)	(S)	0
0	1	1	(T)	0	0
1	0	0	1	1	(U)
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

q = 0

r = 0

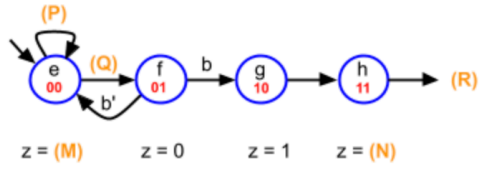
s = 0

t = 1

u = 1

	p1	p0	b	n1	n0	z
e	0	0	0	0	0	0
	0	0	1	0	1	0
f	0	1	0	0	0	0
	0	1	1	1	0	0
g	1	0	0	1	1	1
	1	0	1	1	1	1
h	1	1	0	0	0	1
	1	1	1	0	0	1

Inputs: b
Outputs: z



*State encodings are in red

m = 0

n = 1

p = b'

Q = b

R = e

15.8 Clock Frequency

- A sequential system's clock frequency f_c cannot be arbitrarily fast.
  - ↳ registers could capture incorrect values.

- The critical path delay t_c for a sequential circuit is the longest propogation time through flip-flop outputs, through logic, then back to flip-flop data inputs.
  - ↳ delay interval starts at the clocks active edge and ends at the flip-flop data input.
  - ↳ the interval includes the flip-flop delay t_{ff} and combinational gate propogation delays t_p.

$$t_c = t_{ff} + \max\left(\sum t_p\right)$$

- For proper operation, the clock period T must exceed the critical path delay: $T > t_c$.

- The clock frequency $f_c = 1/T$, in Hz, is the inverse of clock period in seconds.

- In practice, designers typically increase the clock period by a "safety" margin of 10% to 30%. (Ex: $T = 1.3\, t_c$)