Digital Systems I

Instructor: Mohammad Ghamari

Electrical and Computer Engineering Department

These slides are modified version of the slides by Nozar Tabrizi, Introduction to Digital Circuits

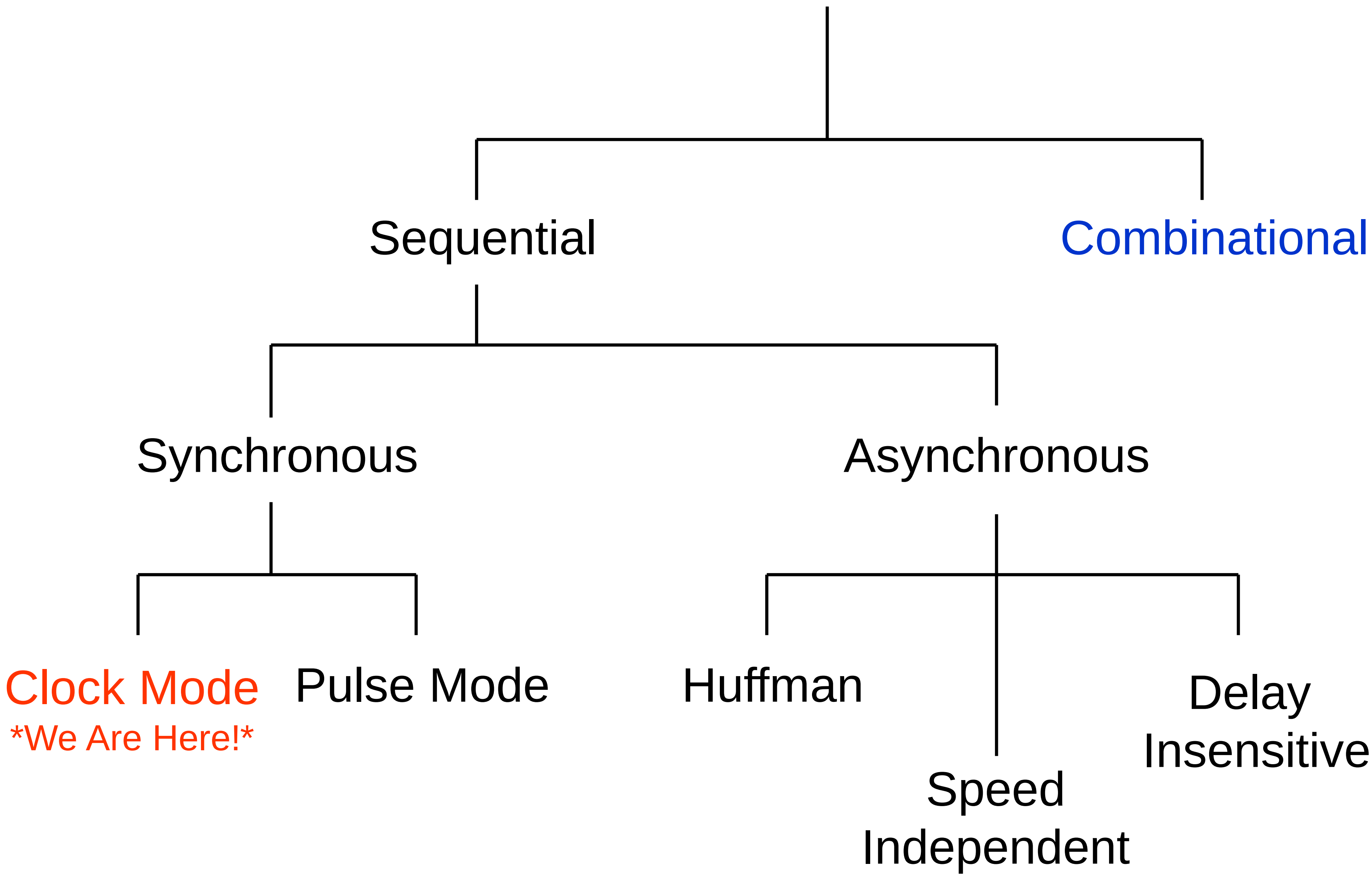
Chapter 8

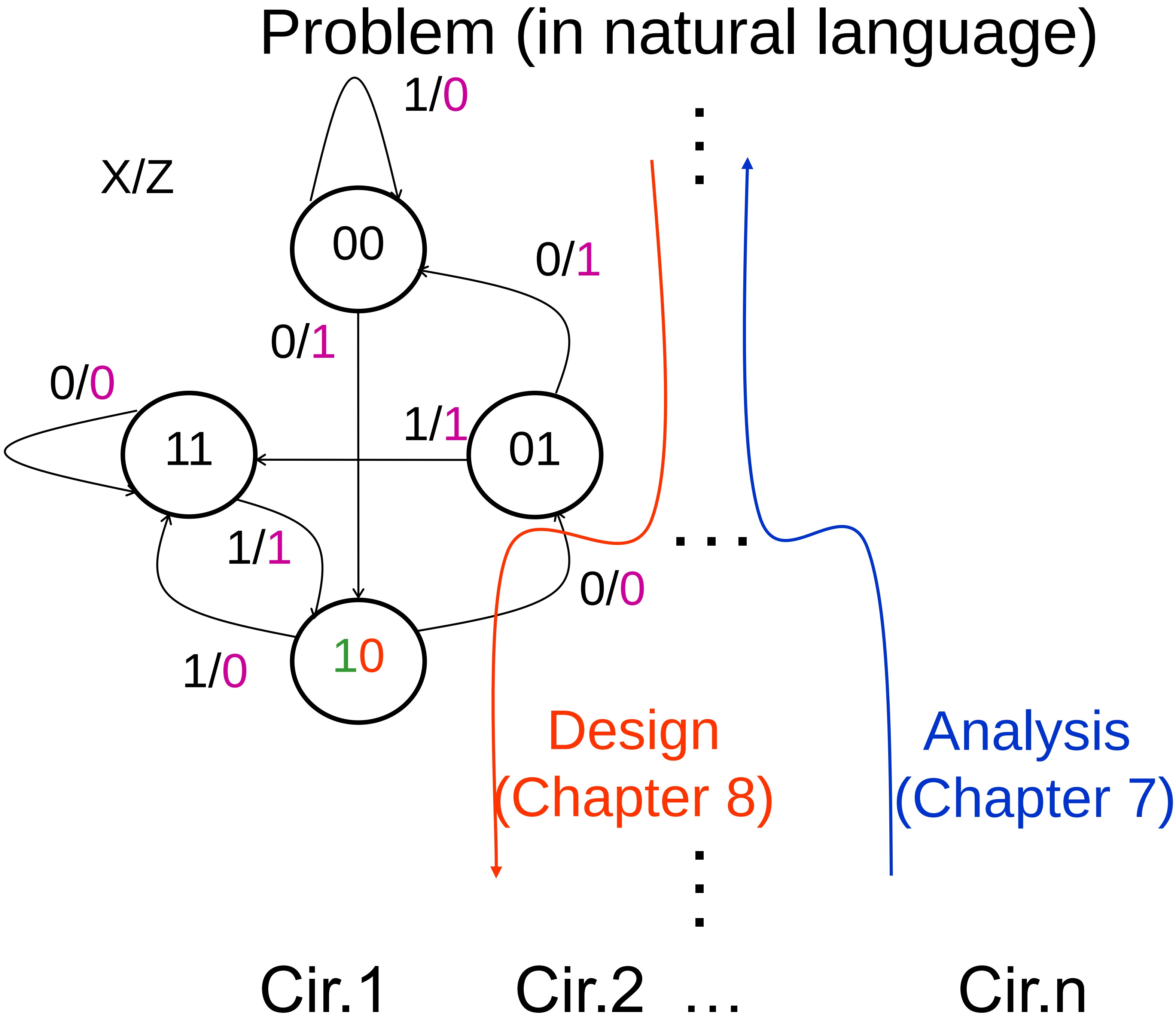
Design of Sequential Circuits

(Finite State Machines)

Digital Circuits

(some well-known types)





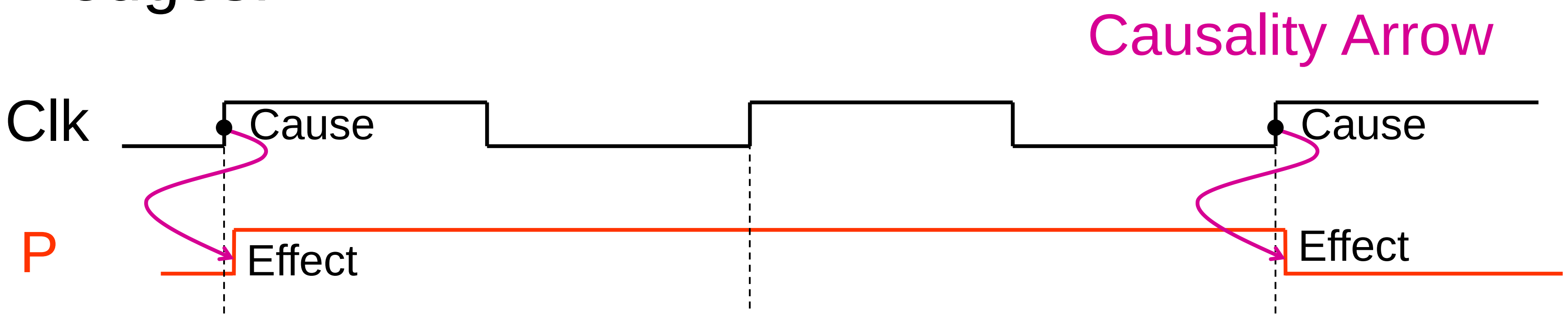
Problem (in natural language)
X/Z
1/0
00
0/1
0/1
0/0
11
1/1
01
1/1
0/0
10
1/0
Design
(Chapter 8)
Analysis
(Chapter 7)
Cir.1
Cir.2
…
Cir.n

Good News ☺

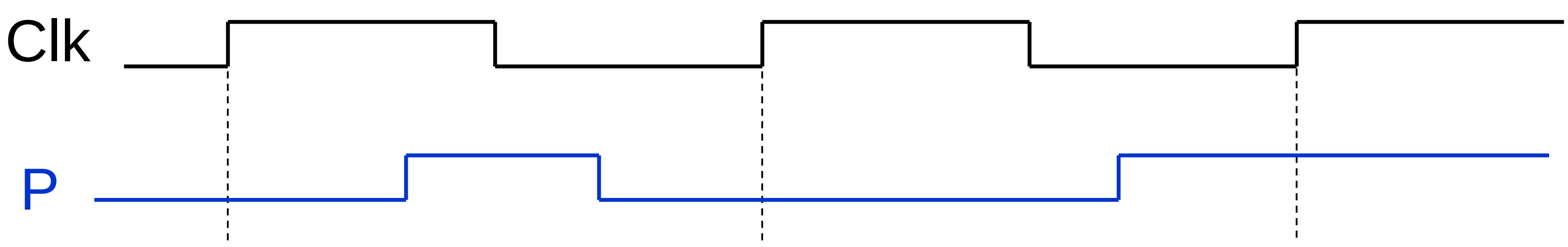
Almost the same steps that we took in the analysis of FSMs, but in reverse order

Synchronous and Asynchronous signals

Synchronous signal may only change with clock edges:

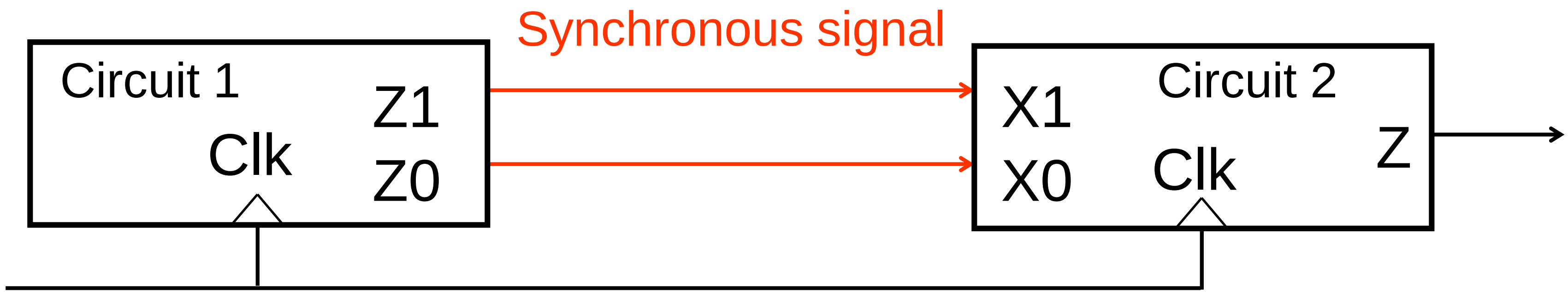


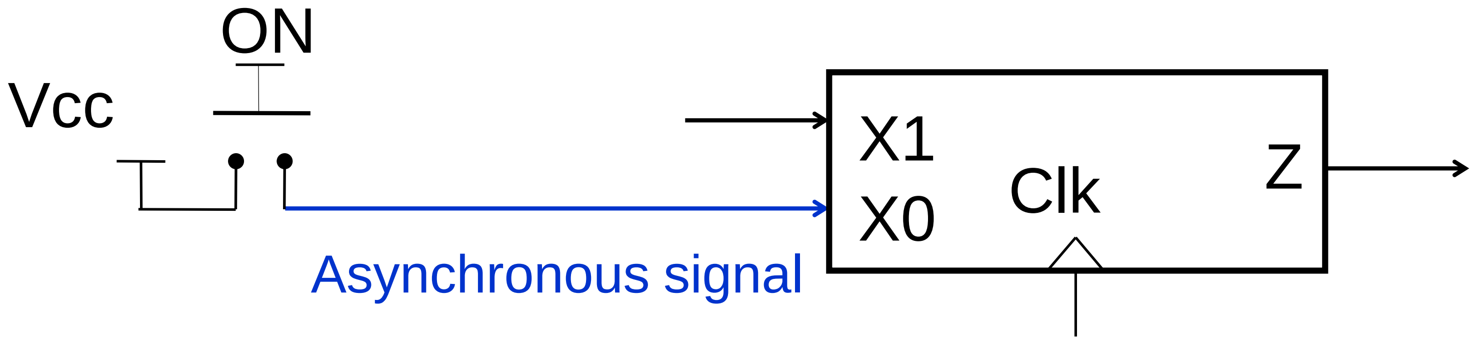
Asynchronous signal may change any time:



Synchronous and asynchronous signals

Synchronous signal is generated by a synchronous circuit with the same clock.





Design Steps

Step 1:

Translate natural language (word description) to state diagram

State machines Behavior

Can be observed in many devices in modern society **that perform a predetermined sequence of actions** depending on a **sequence of events with which they are presented**

- Vending Machines: which give products when the proper combination of coins is deposited

- Elevators: whose sequence of stops is determined by the floors requested by riders

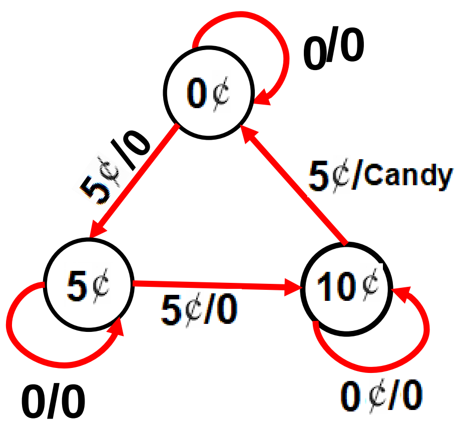
- Traffic Lights, which change sequence when cars are waiting

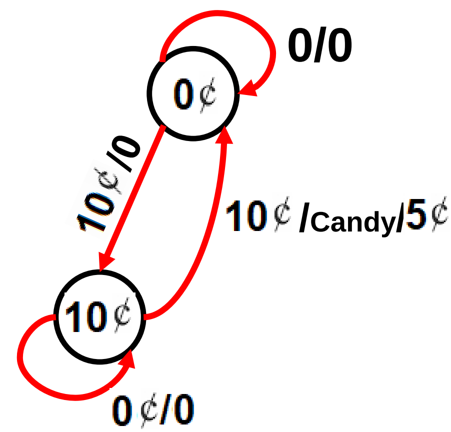
- Combination Locks, which require the input of a sequence of numbers in the proper order

Cost of a bar of candy = 15¢

Input N = $\begin{cases} 1 & 5¢ \\ 0 & 0¢ \end{cases}$

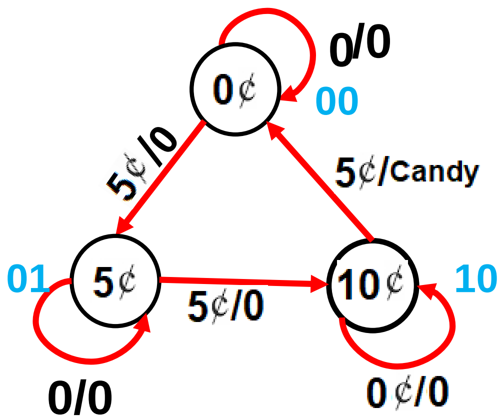
Input D = $\begin{cases} 1 & 10¢ \\ 0 & 0¢ \end{cases}$





Cost of a bar of candy = 15¢

Input N = $\begin{cases} 1 & 5¢ \\ 0 & 0¢ \end{cases}$



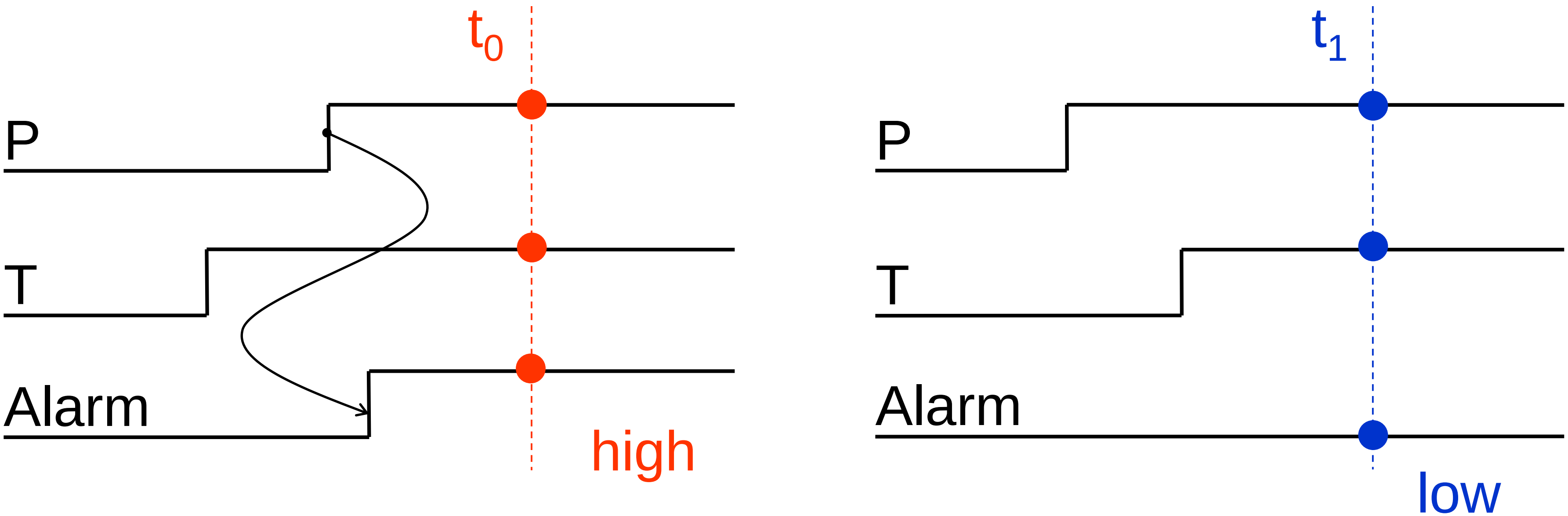
Current State		Input	Next State		Output
Q1	*Q0*	*Money In*	*Q'1*	*Q'0*	*Candy*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

Example 1. Develop a state diagram for the alarm control system of a chemical plant such that:

Starting with reset state ($P = 0$ & $T = 0$)

- If pressure goes up ($P = 1$) **after** temperature goes up ($T = 1$), alarm goes off while both P & T are high.
- Alarm turns off if P gets normal.
- Alarm turns on if P goes up again, while T is high.
- When T goes back to normal, above sequence repeats.

Timing Diagram

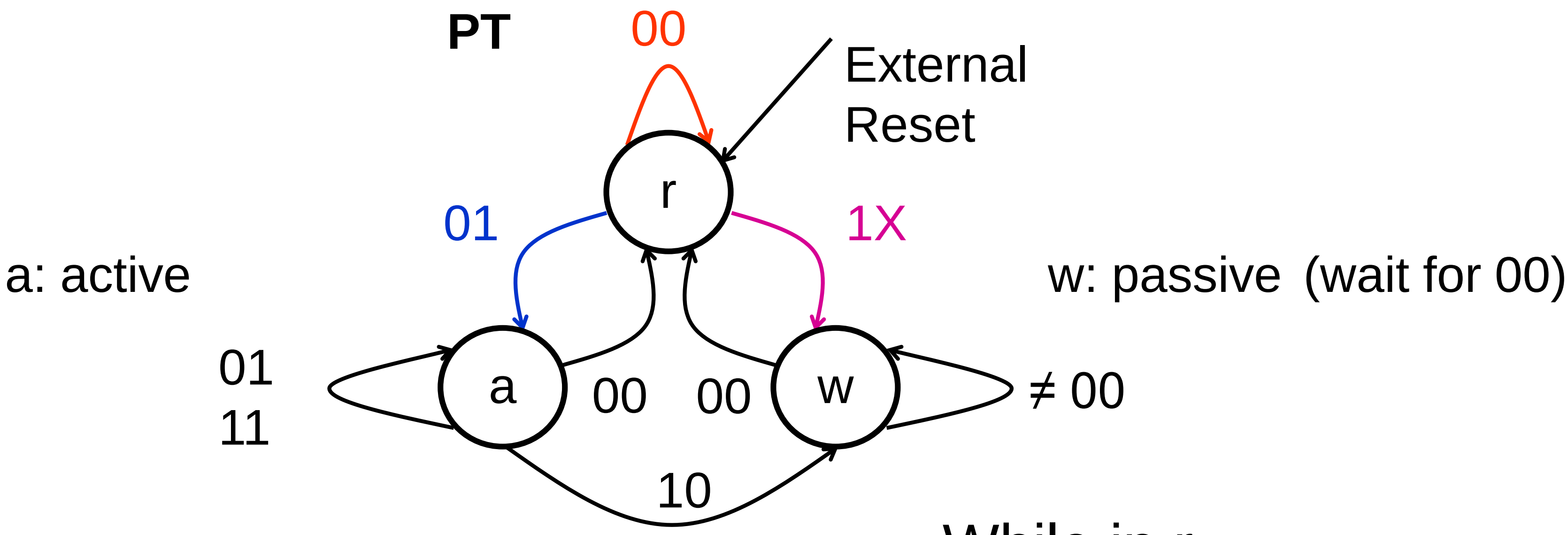


Same inputs (T=1, P=1) generate different outputs (alarm = 1 @ t0 but alarm = 0 @ t1

So we need a sequential circuit (FSM)

If pressure goes up (P = 1) **after** temperature goes up (T = 1), alarm goes off while both P & T are high.

State Diagram



While in r (as an example)

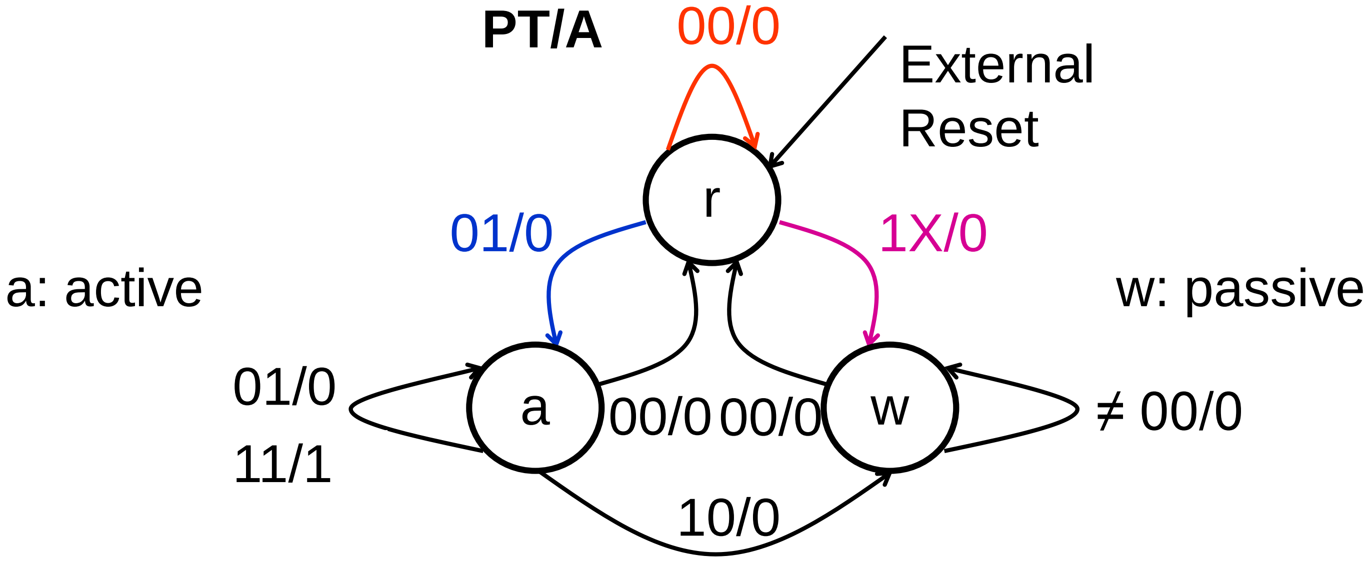
If PT = 00 then go to r

If PT = 10 then go to w

If PT = 11 then go to w

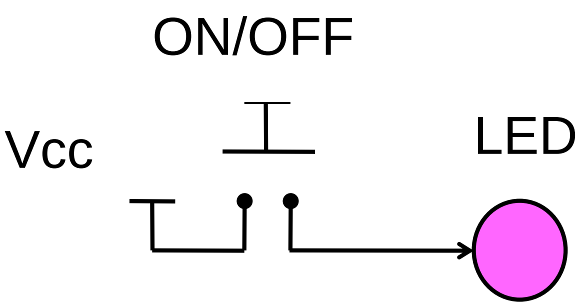
If PT = 01 then go to a

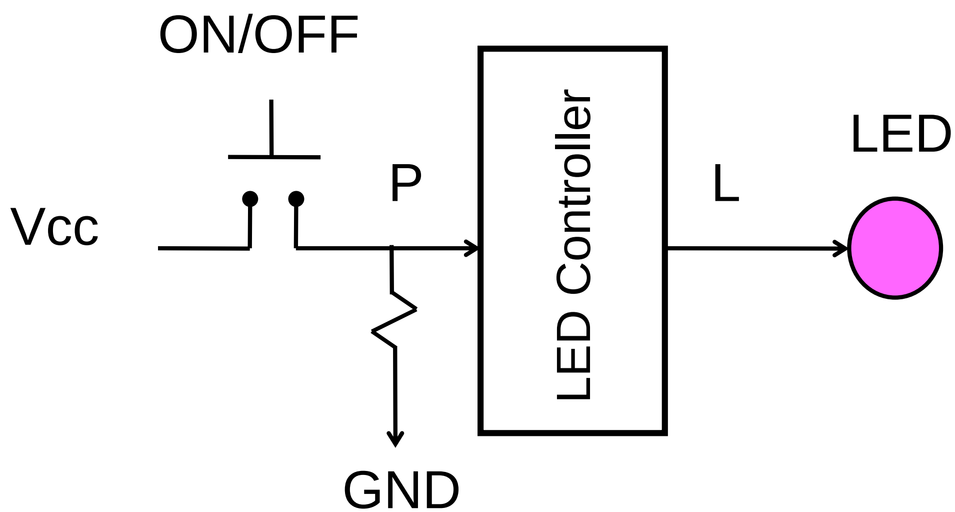
State Diagram: Output information added



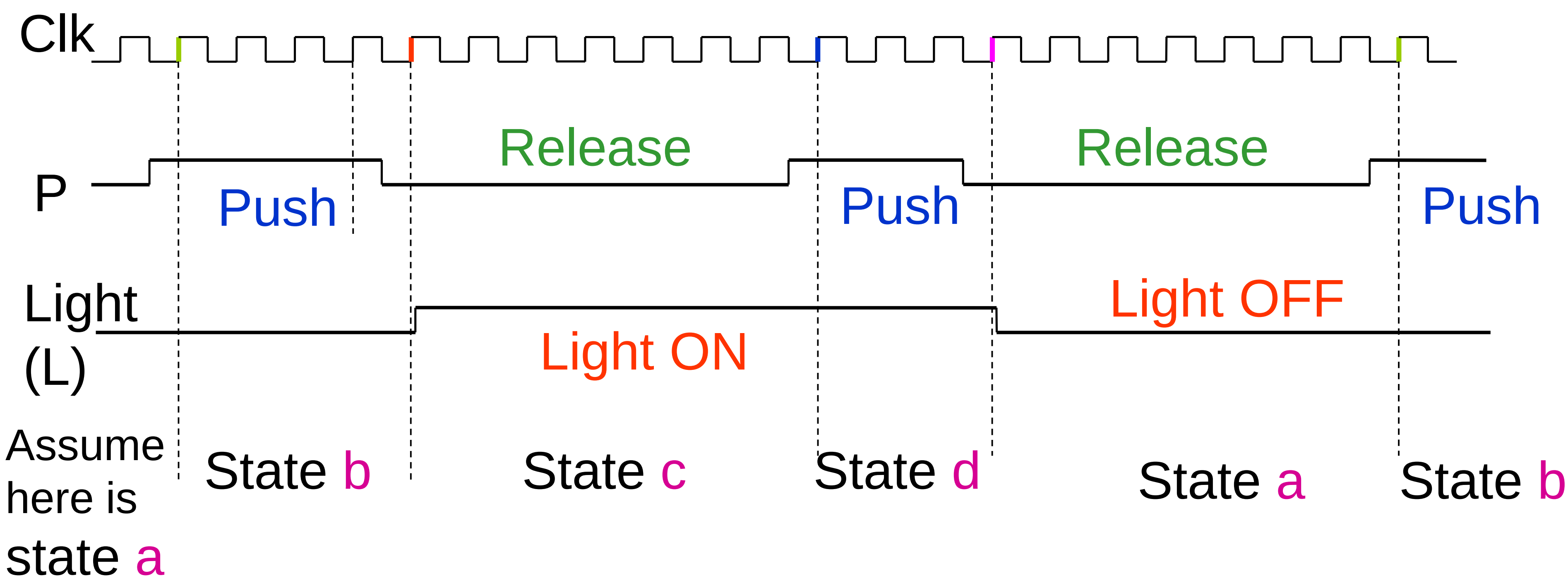
Example 7.

Add memory to a push-button; when button is pushed (P = 1) and then released (P = 0) an OFF light (L = 0) turns ON (L = 1) and remains ON until the button is pushed and released again.



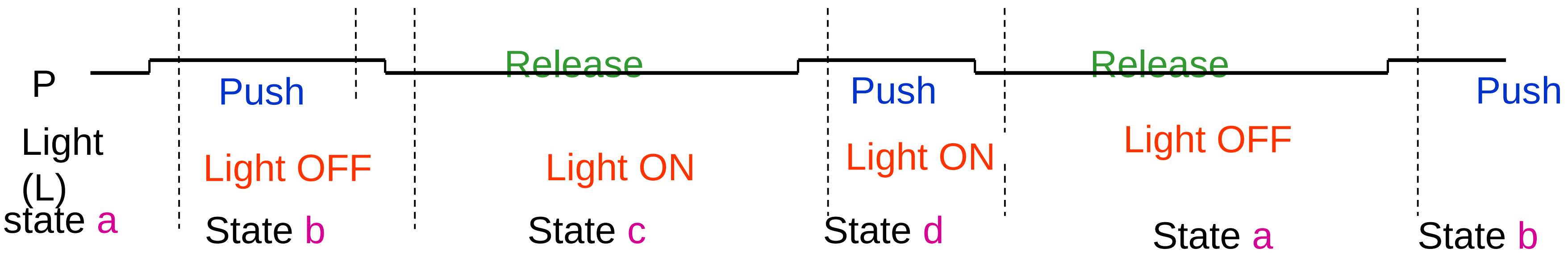


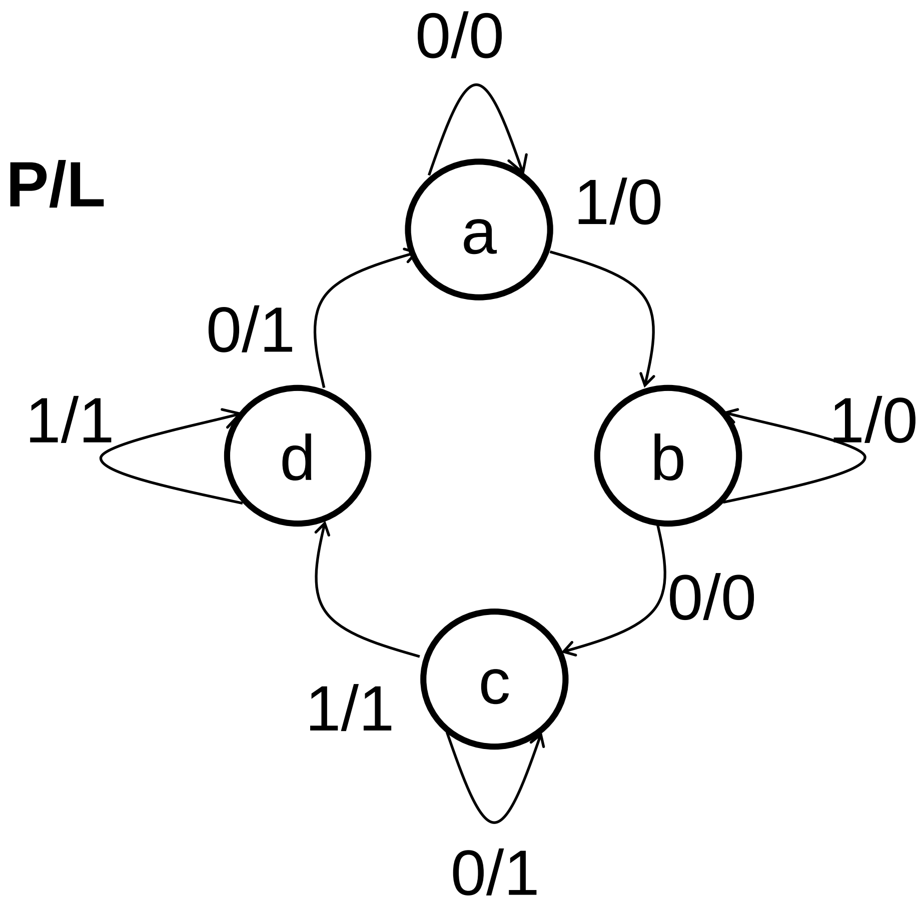
Timing Diagram



- With the first clock edge (after the button is pushed) the controller goes to state **b** (where the light is still off) and remains there until the button is released.
- The next clock edge takes controller to state **c**, where the light turns on.
- Again with the first clock edge (after the button is pushed) the controller goes to state **d** (where the light is still on) and remains there until the button is released.
- The next clock edge takes the controller to state **a**, where the light turns off again.

State Diagram





Finite State Machines (FSM)

Analysis:

Logic circuit

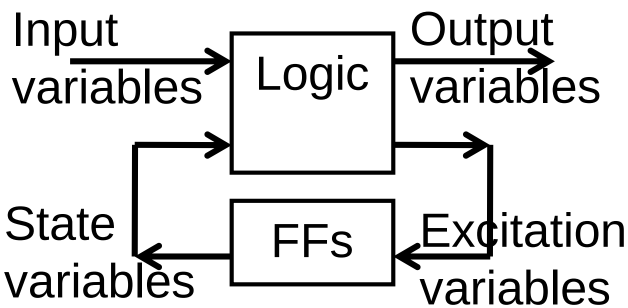
Excitation & output equations

Excitation & output maps

Partial transition tables

Transition table/graph

(State assignment, State graph)



Design:

Natural language

State graph/state table

State minimization

State assignment, transition table

Partial transition tables

Excitation and output maps & minimization

Excitation and output equations

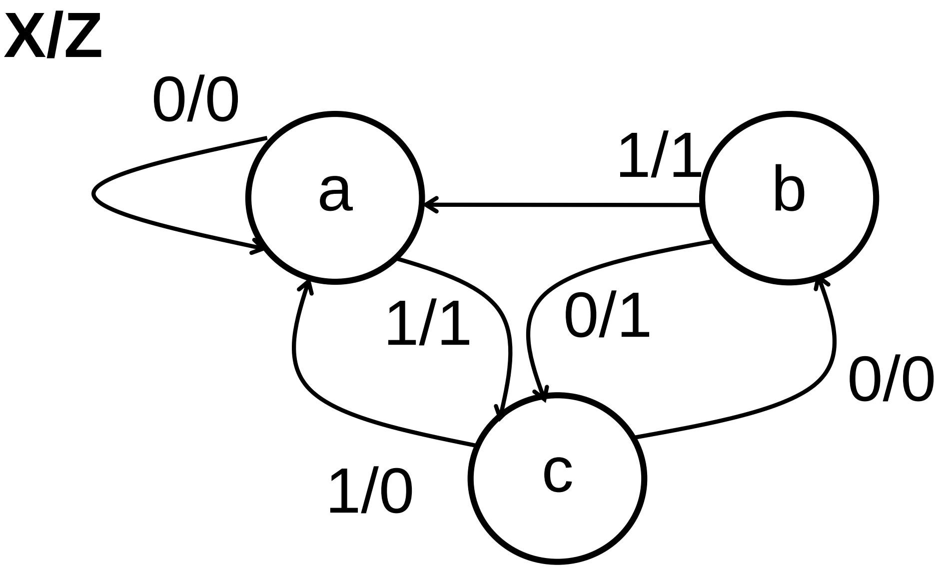
Logic circuit

Design of Sequential Circuits

Steps 2 - 7

Example. Use D-FFs

1) State diagram (graph)

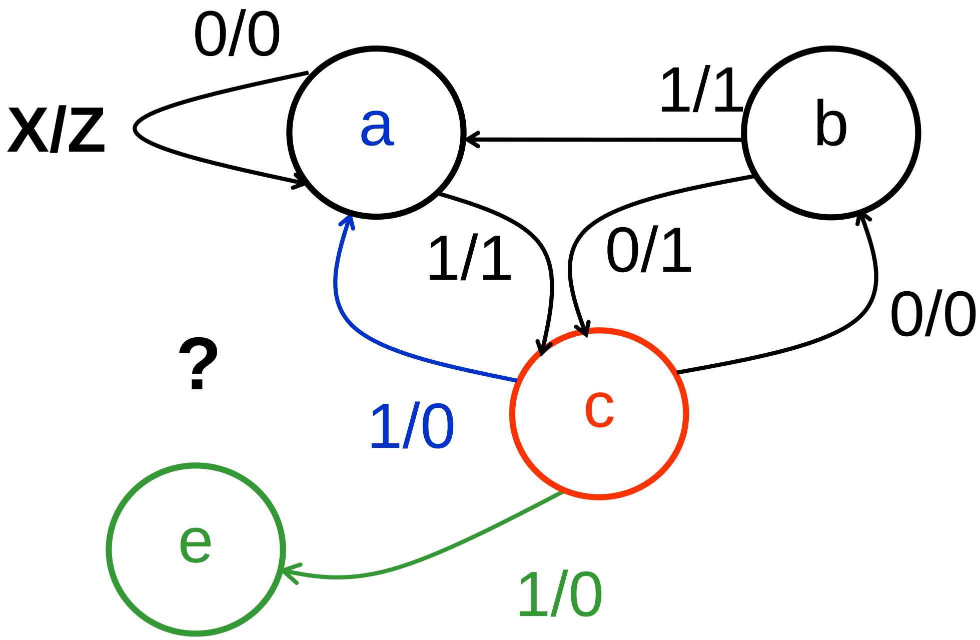


2) State table

Q \ X	0	1
a	a, 0	c, 1
b	c, 1	a, 1
c	b, 0	a, 0

Q^{n+1}, Z

3) State minimization



Which one?
a or **e**

**Redundant states will
be identified and removed in this state**

4) State assignment & transition table

Q1Q0	Q	X = 0	X = 1
00	a	a, 0	c, 1
01	b	c, 1	a, 1
11	c	b, 0	a, 0
10			

Q^{n+1}, Z

Q1Q0	X = 0	X = 1
00	00, 0	11, 1
01	11, 1	00, 1
11	01, 0	00, 0
10	xx, x	xx, x

$Q1^{n+1}$ $Q0^{n+1}$, Z

5) Partial transition tables

Q1Q0 \ X	0	1
00	00, 0	11, 1
01	11, 1	00, 1
11	01, 0	00, 0
10	xx, x	xx, x

$Q1^{n+1}\ Q0^{n+1}$, Z

Q1Q0 \ X	0	1
00	0	1
01	1	0
11	0	0
10	x	x

$Q1^{n+1}$

Q1Q0 \ X	0	1
00	0	1
01	1	0
11	1	0
10	x	x

$Q0^{n+1}$

6) Choose FFs

- D-type is the least expensive
- Different types may result in different amounts of hardware to realize combinational circuits
- Different types may be used in the same design

7) Excitation & output maps and minimizaion

$$Q^{n+1} = D$$

Excitation maps = partial transition tables

Q1Q0 \ X	0	1
00	0	1
01	1	0
11	0	0
10	x	x

D1

Q1Q0 \ X	0	1
00	0	1
01	1	0
11	1	0
10	x	x

D0

D1 = X.Q0’ + X’.Q1’.Q0

D0 = X.Q0’ + X’. Q0

7) Excitation & output maps and minimizaion

Q1Q0 \ X	0	1
00	0	1
01	1	1
11	0	0
10	X	X

Z

$$Z = Q1'.Q0 + X.Q1'$$

7) Excitation & output maps and minimizaion

$$Q^{n+1} = D$$

Excitation maps = partial transition tables

Q1Q0 \ X	0	1
00	0	1
01	1	0
11	0	0
10	X	X

D1

Q1Q0 \ X	0	1
00	0	1
01	1	0
11	1	0
10	X	X

D0

D1 = (X +Q0) . (X’ + Q0’) . Q1’

D0 =(X +Q0).(X’ + Q0’)

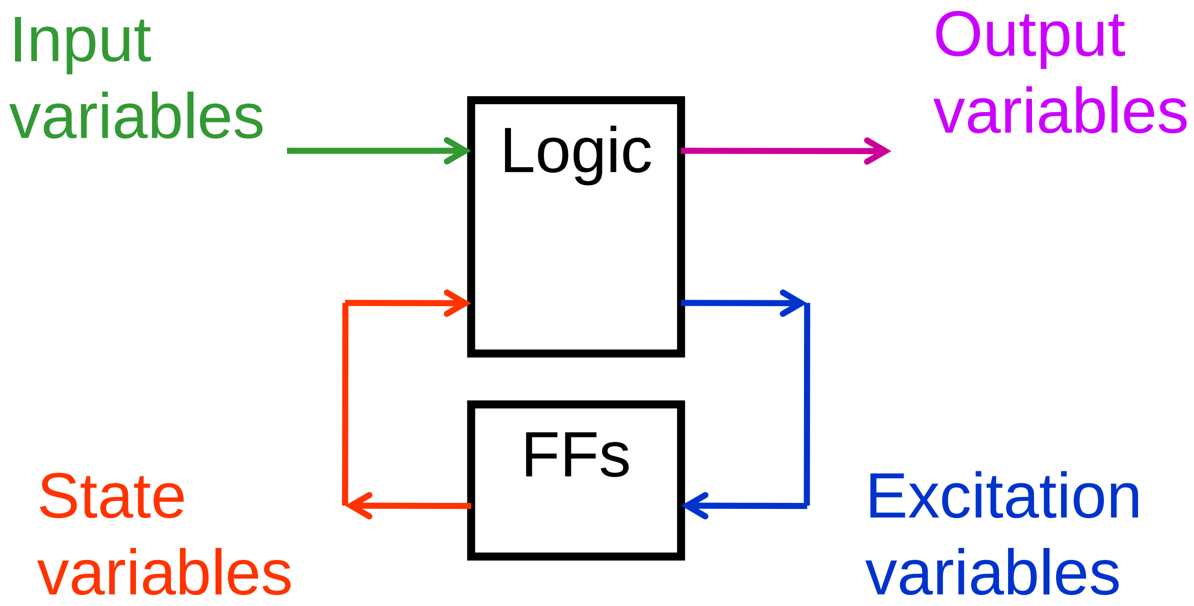
7) Excitation & output maps and minimizaion

Q1Q0 \ X	0	1
00	0	1
01	1	1
11	0	0
10	x	x

Z

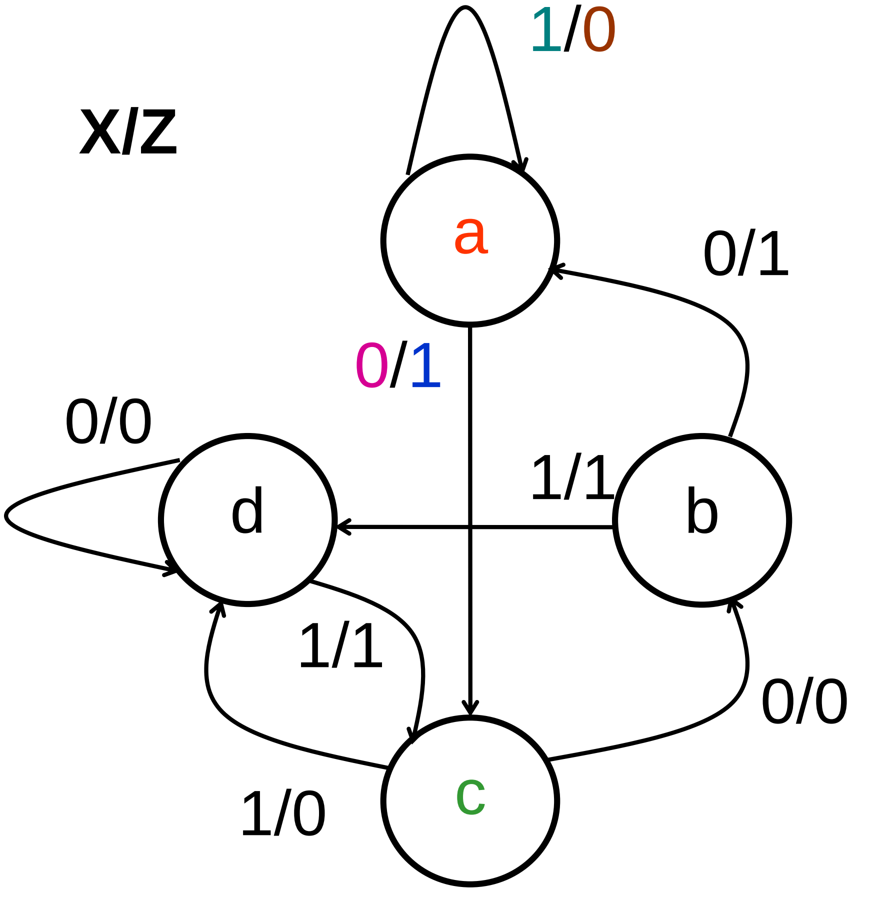
$$Z = Q1' \,.\, (X + Q0)$$

Finite-State Machines (Re-visited)



Example 9. Use D-FFs to realize the state diagram shown here.

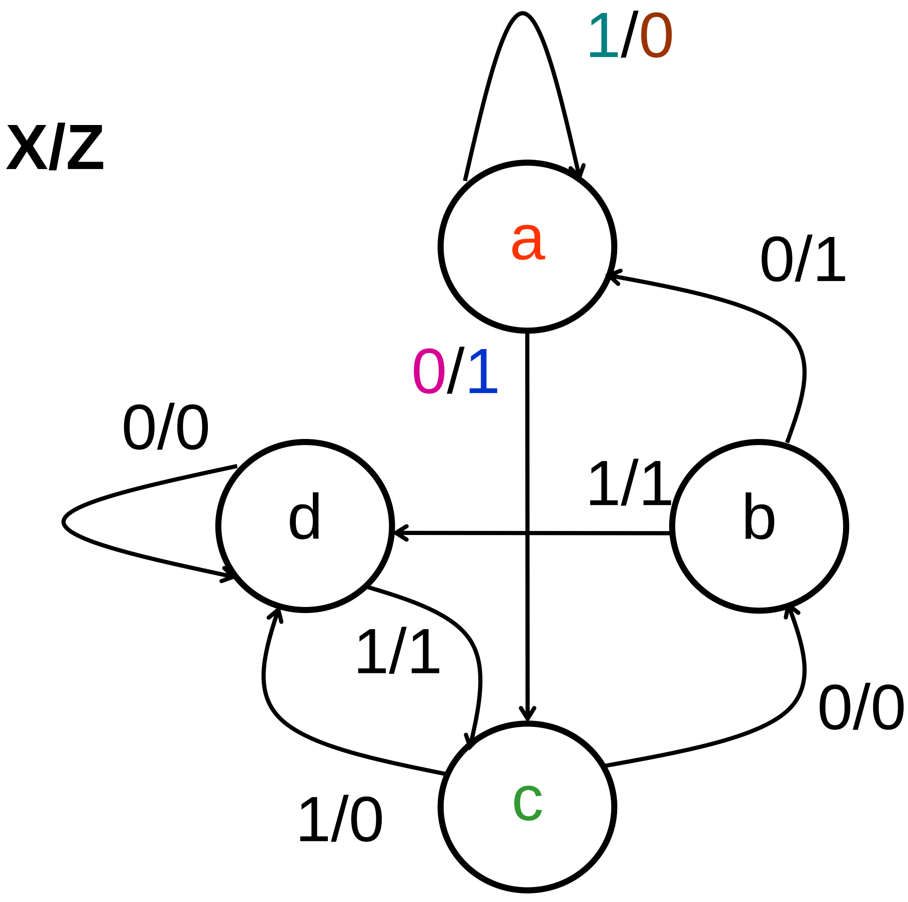
1) State diagram (graph)



Try to solve this

Example 9. Use D-FFs

1) State diagram (graph)



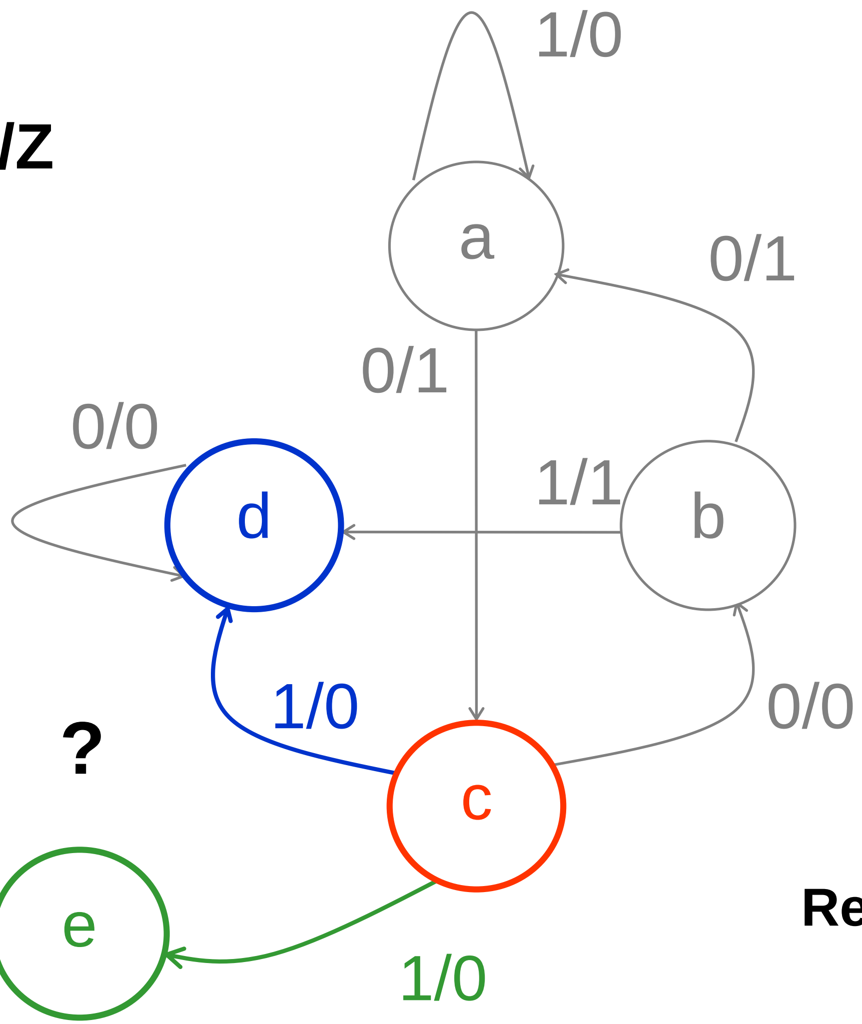
2) State table

Q \ X	0	1
a	c, 1	a, 0
b	a, 1	d, 1
d	d, 0	c, 1
c	b, 0	d, 0

Q^{n+1}, Z

3) State minimization

X/Z



Redundant states will be identified here

4) State assignment & transition table

Q1Q0	Q \ X	0	1
00	a	c, 1	a, 0
01	b	a, 1	d, 1
11	d	d, 0	c, 1
10	c	b, 0	d, 0

Q^{n+1}, Z

Q1Q0 \ X	0	1
00	10, 1	00, 0
01	00, 1	11, 1
11	11, 0	10, 1
10	01, 0	11, 0

$Q1^{n+1}\ Q0^{n+1}$, Z

5) Partial transition tables

Q1Q0 \ X	0	1
00	10, 1	00, 0
01	00, 1	11, 1
11	11, 0	10, 1
10	01, 0	11, 0

$Q1^{n+1}\ Q0^{n+1}$, Z

Q1Q0 \ X	0	1
00	1	0
01	0	1
11	1	1
10	0	1

$Q1^{n+1}$

Q1**Q0** \ X	0	1
0**0**	0	0
0**1**	0	1
1**1**	1	0
1**0**	1	1

$Q0^{n+1}$

6) Choose FFs

- D-type is the least expensive
- Different types may result in different amounts of hardware to realize combinational circuits
- Different types may be used in the same design

7) Excitation & output maps and minimizaion

$$Q^{n+1} = D$$

Excitation maps = partial transition tables

Q1Q0 \ X	0	1
00	1	0
01	0	1
11	1	1
10	0	1

D1

Q1Q0 \ X	0	1
00	0	0
01	0	1
11	1	0
10	1	1

D0

D1 = X.Q1 + Q1.Q0 + X.Q0 + X’.Q1’.Q0’

D0 = X’.Q1 + Q1.Q0’ + X.Q1’.Q0

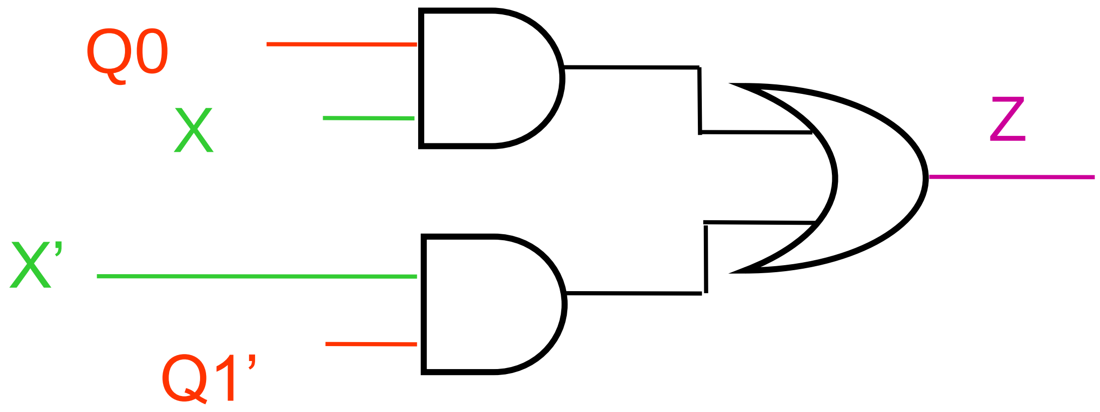
7) Excitation & output maps and minimizaion

Q1Q0 \ X	0	1
00	1	0
01	1	1
11	0	1
10	0	0

Z

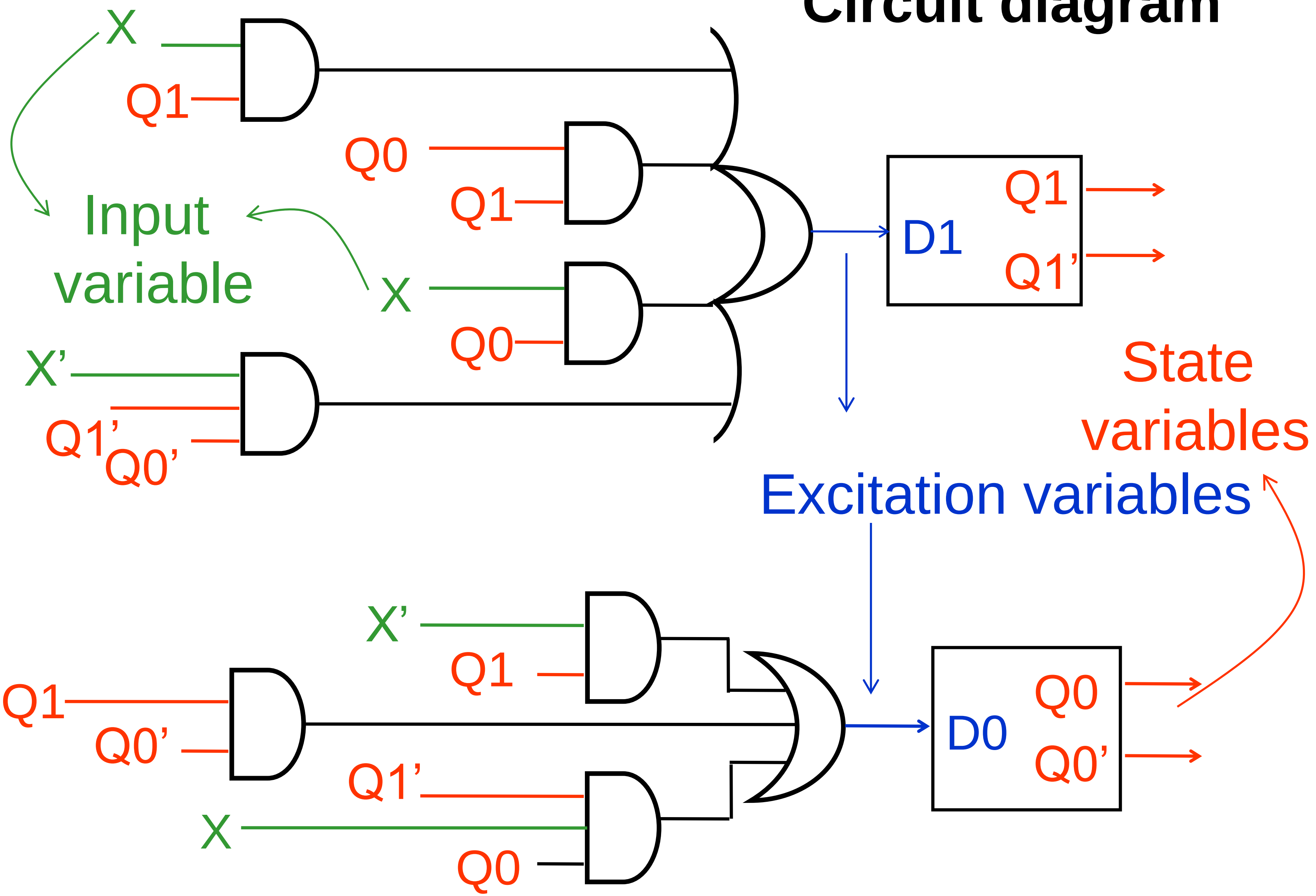
Z = X.Q0 + X'.Q1'

Circuit diagram



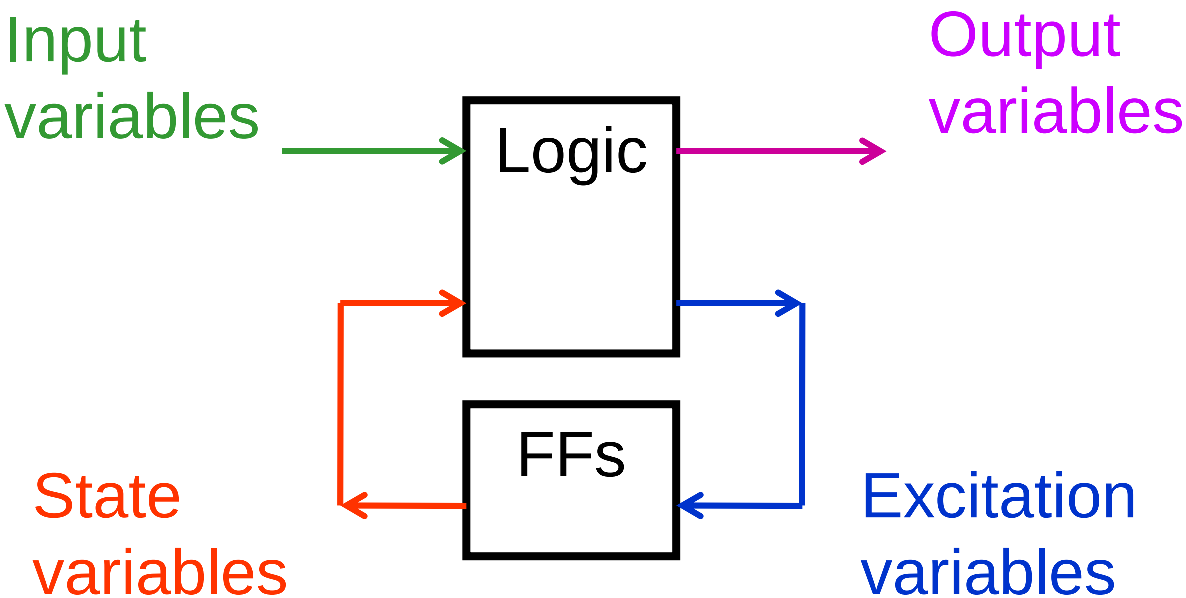
$$Z = X \cdot Q0 + X' \cdot Q1'$$

Circuit diagram



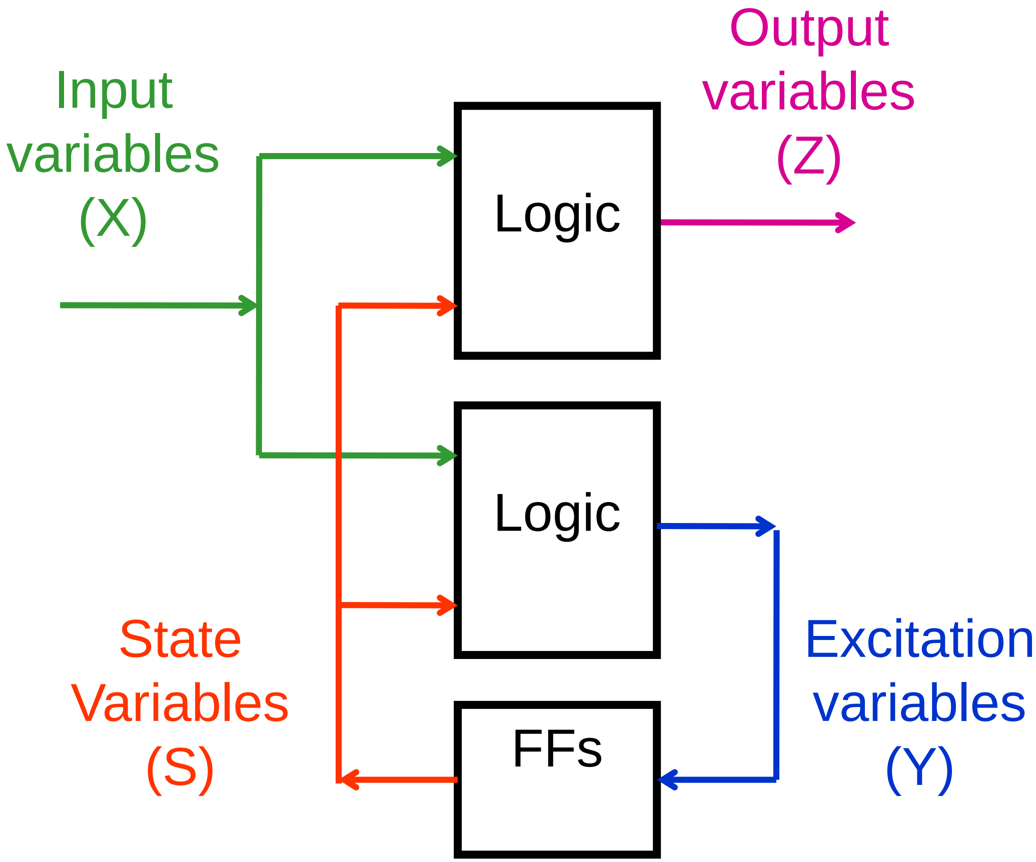
Design of synchronous circuits **using JK or T FFs** will not be covered, but you need to know what JK-FFs and T-FFs are and how they work.

Finite-State Machines (Re-visited)



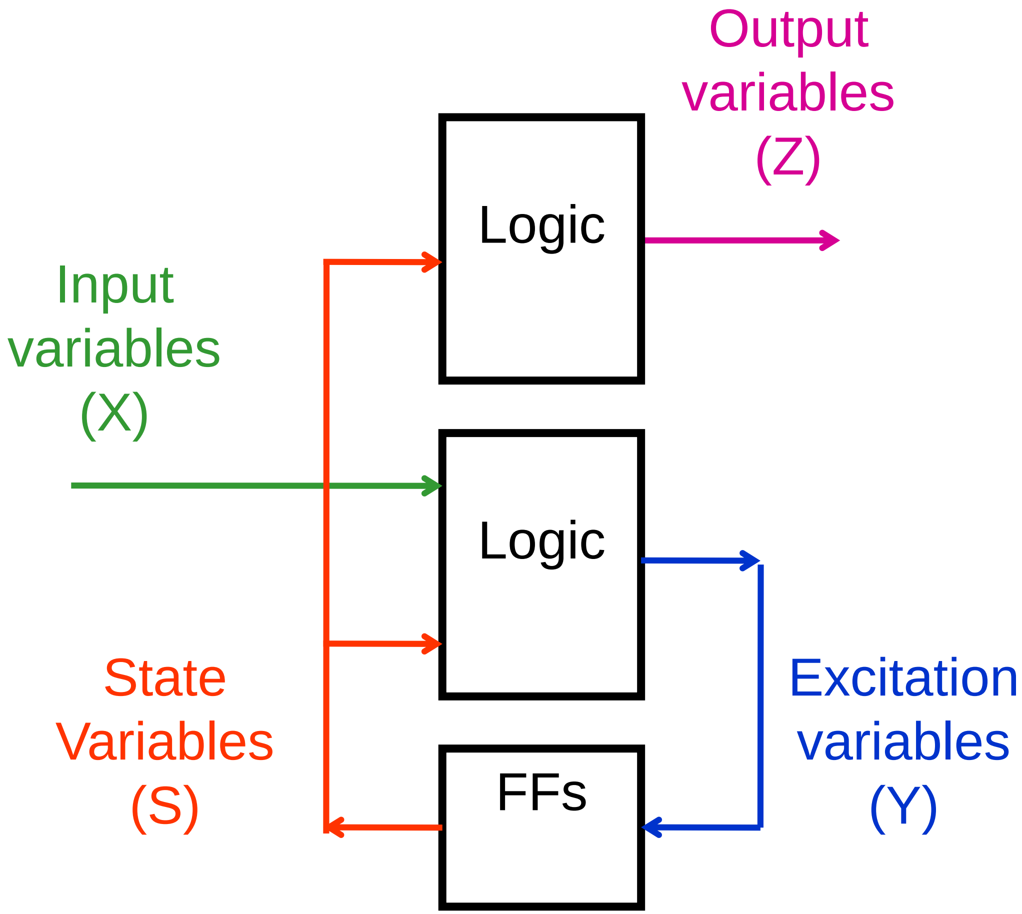
Mealy Machines

Output is a function of state variables and inputs variables

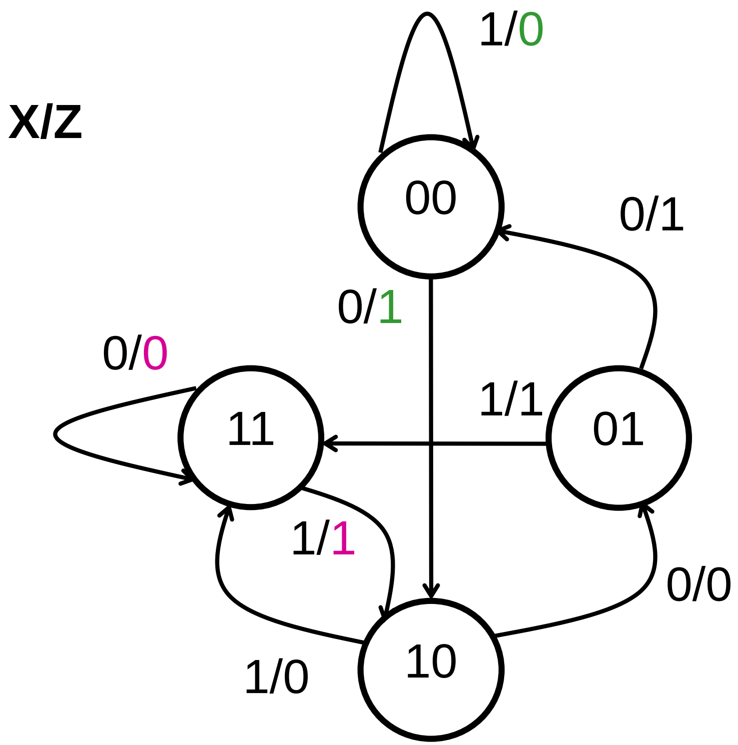


Moore Machines

Output is a function of state variables only.

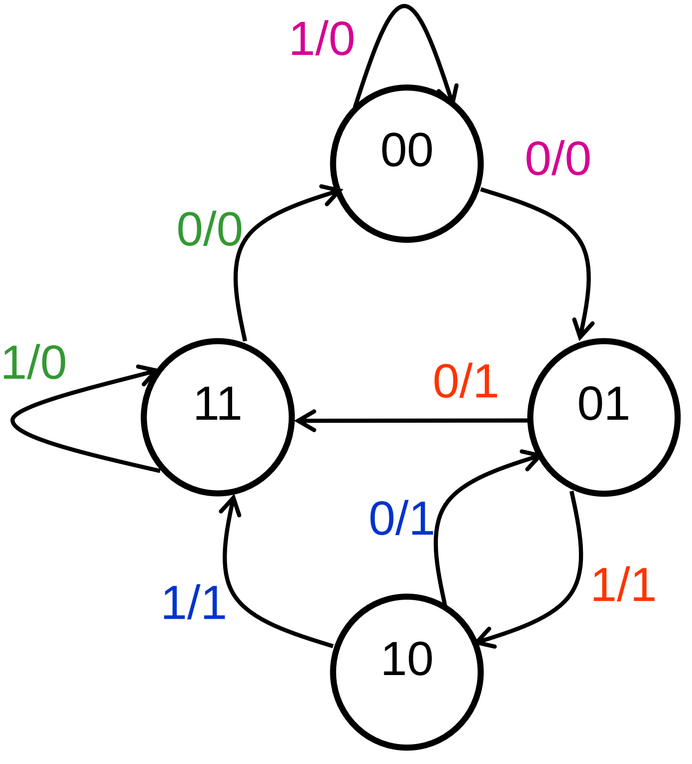


Mealy Machine (Example)

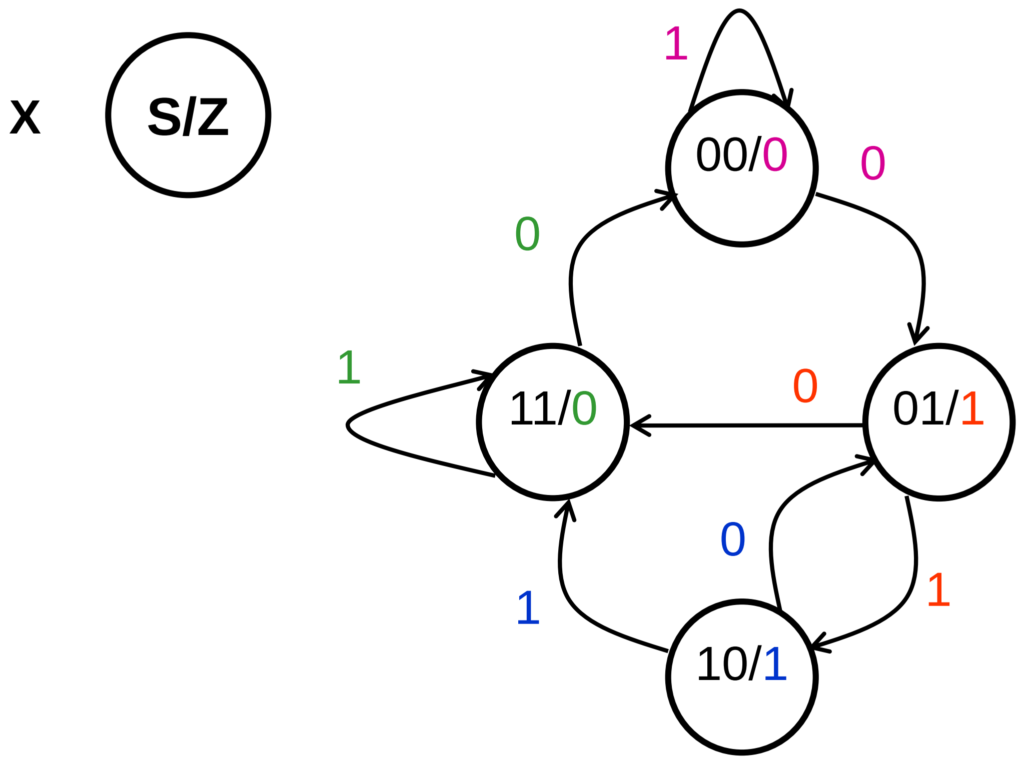


Moore Machine (Example)

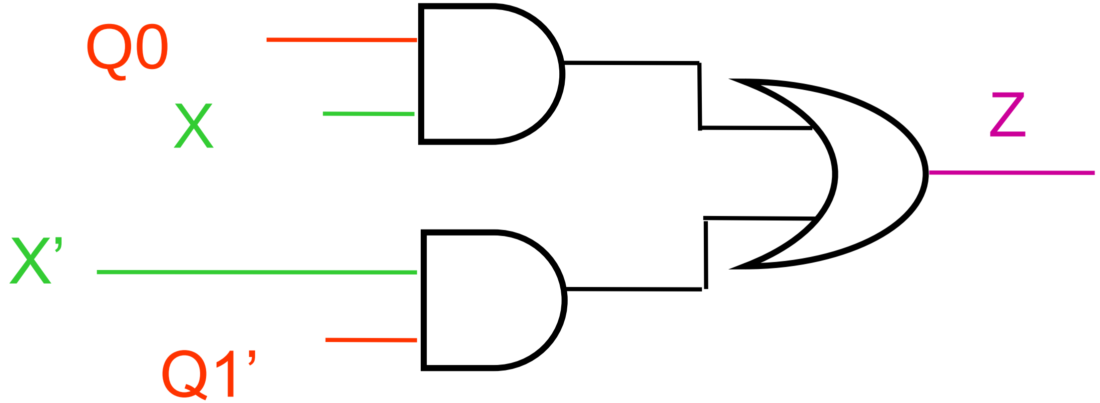
X/Z



Moore Machine (Example)

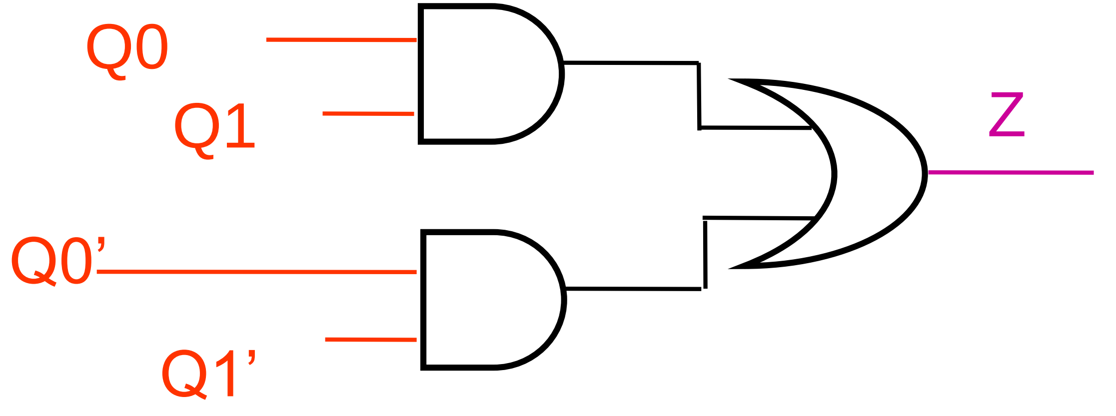


Mealy Machine (Example)



Z = X . Q0 + X' . Q1'

Moore Machine (Example)



$$Z = Q1 \,.\, Q0 + Q1' \,.\, Q0'$$

END