Digial Design

Week 6: Finite State Machines

Fenerbahce University

Instructors

Faculty Member: Dr. Vecdi Emre Levent

Office: 311

Email : emre.levent@fbu.edu.tr

R.A. Ugur Ozbalkan

Office: 311

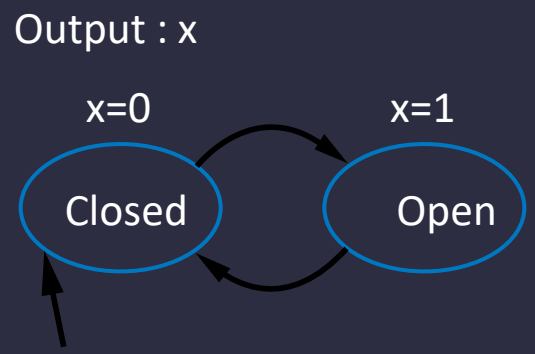
Email : ugur.ozbalkan@fbu.edu.tr

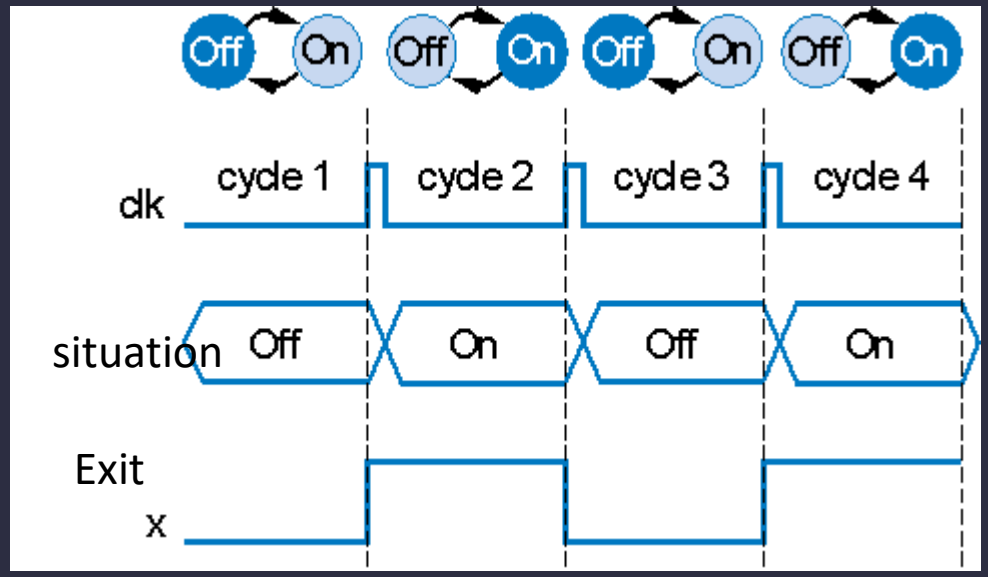
Course

- Finite State Machines

Determining the Behavior of the Sequential Circuit FSM

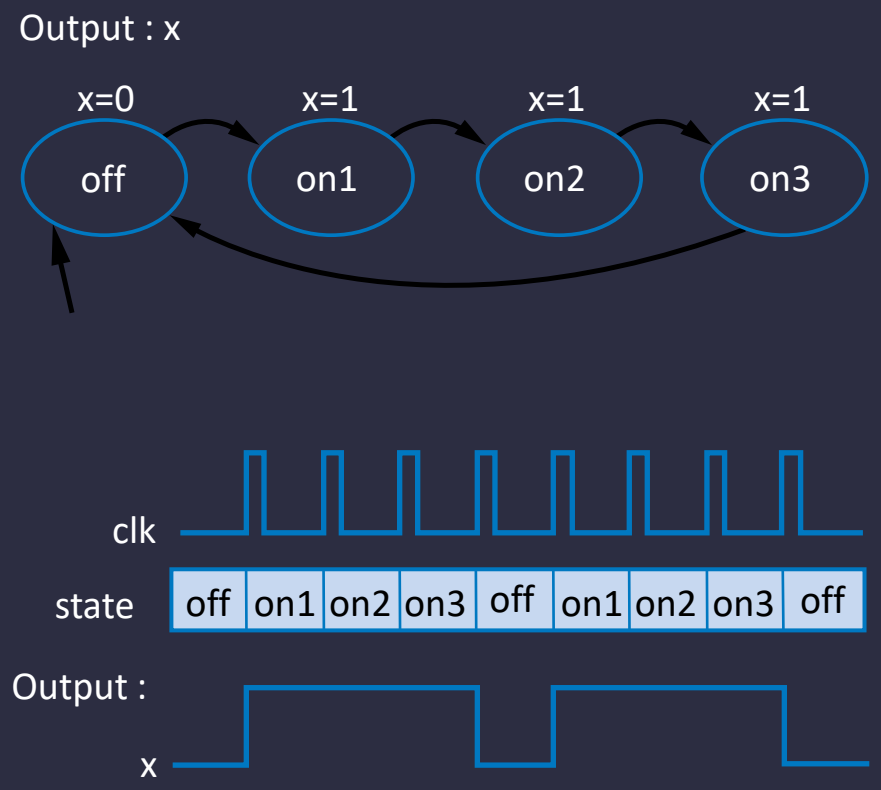
- Finite-State Machine (FSM)
  - It is a representation of the behavior of the sequential circuit according to the states.
  - Lists states and transitions between states
    - Example : Let there be an output signal named X and each clock change value in cycle
    - Two states : “Off” (x=0), and “On” (x=1)
    - There are transitions from the Off state to the On state and from the On state to the Off state.
    - The initial state is indicated by an arrow.





FSM Example : 0,1,1,1, repetition

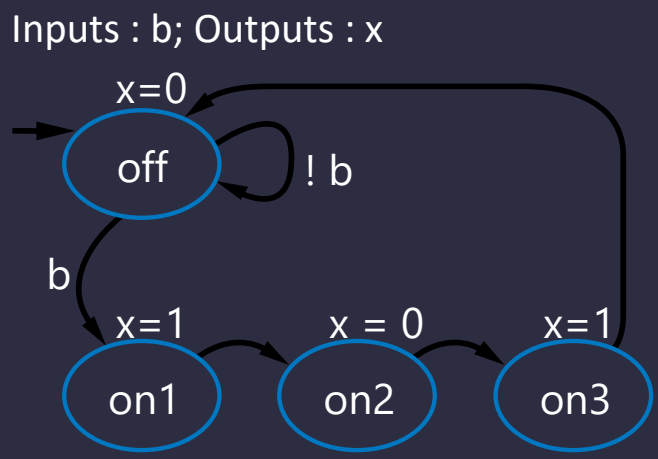
- In order 0, 1, 1, 1, 0, 1, 1, 1, ... A circuit that gives its outputs will be designed.
  - Each value is a clock It appears in the cycle .
- Can be designed with FSM
  - There are 4 states

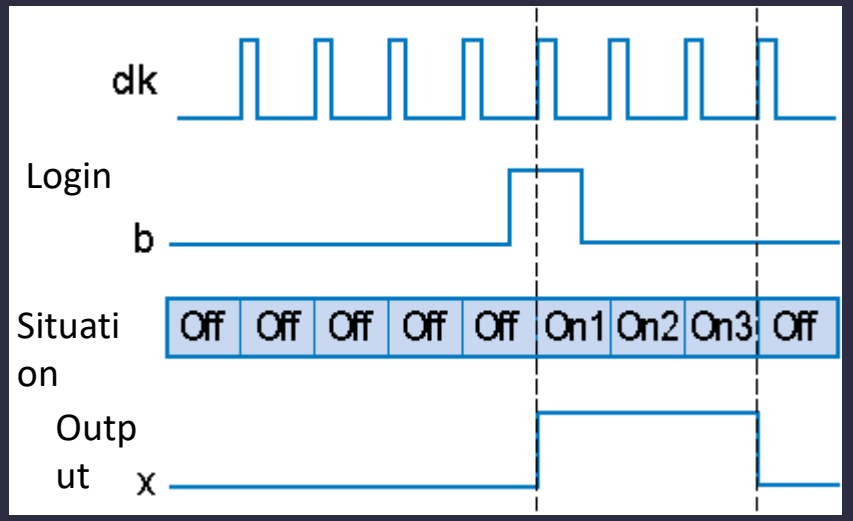


FSM Example

take a one-bit input named b,

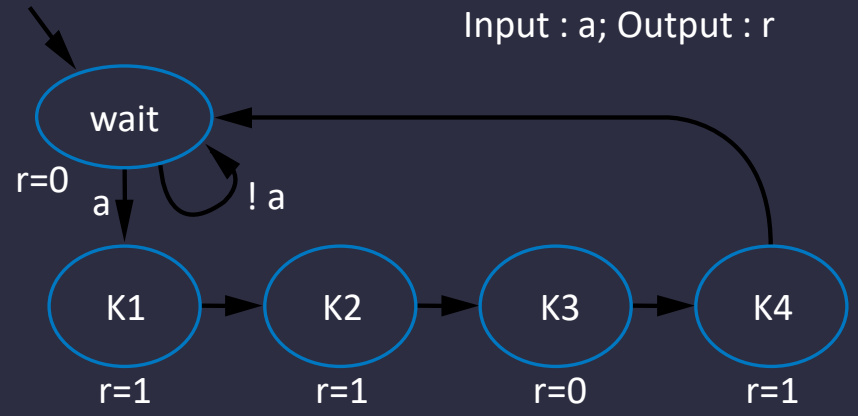
Output pattern 101 and waits for input b again.





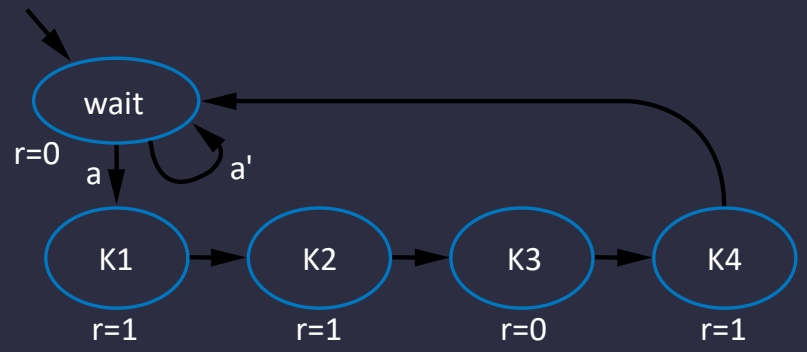
FSM Example: Car Key

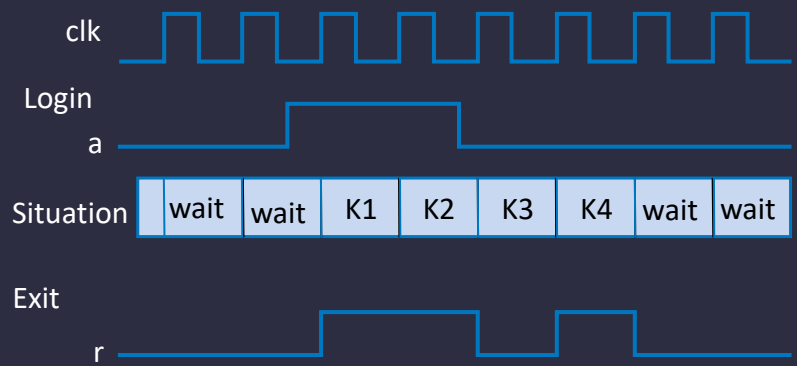
- Car keys contain a chip that holds the key's identity.
  - When the key is inserted in the car, it asks the key for identification.
  - Sends key credential, if not correct the tool won't start
- FSM
  - Wait for request (a=1)
  - Send ID (For example , 1101)

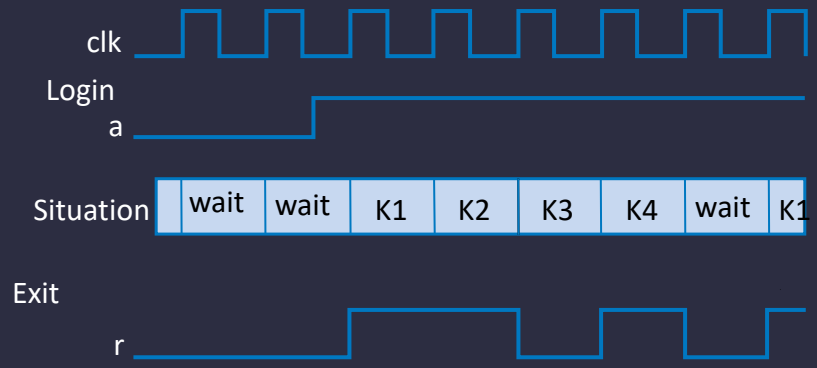


FSM Example: Car Key

- FSM Timings

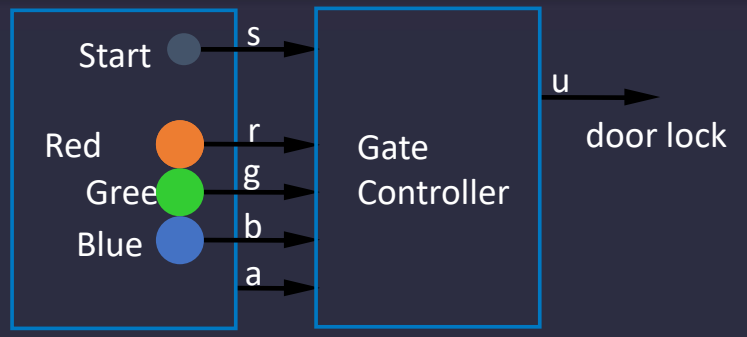


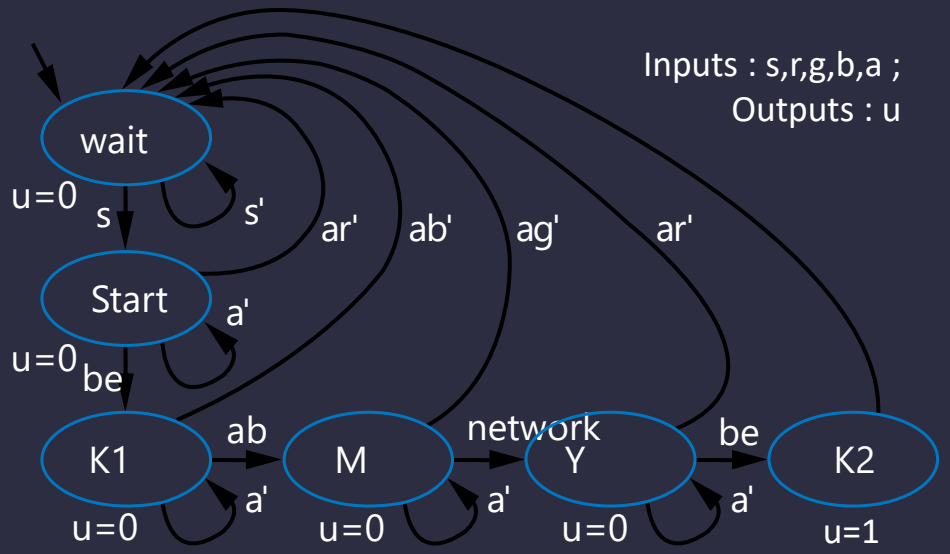




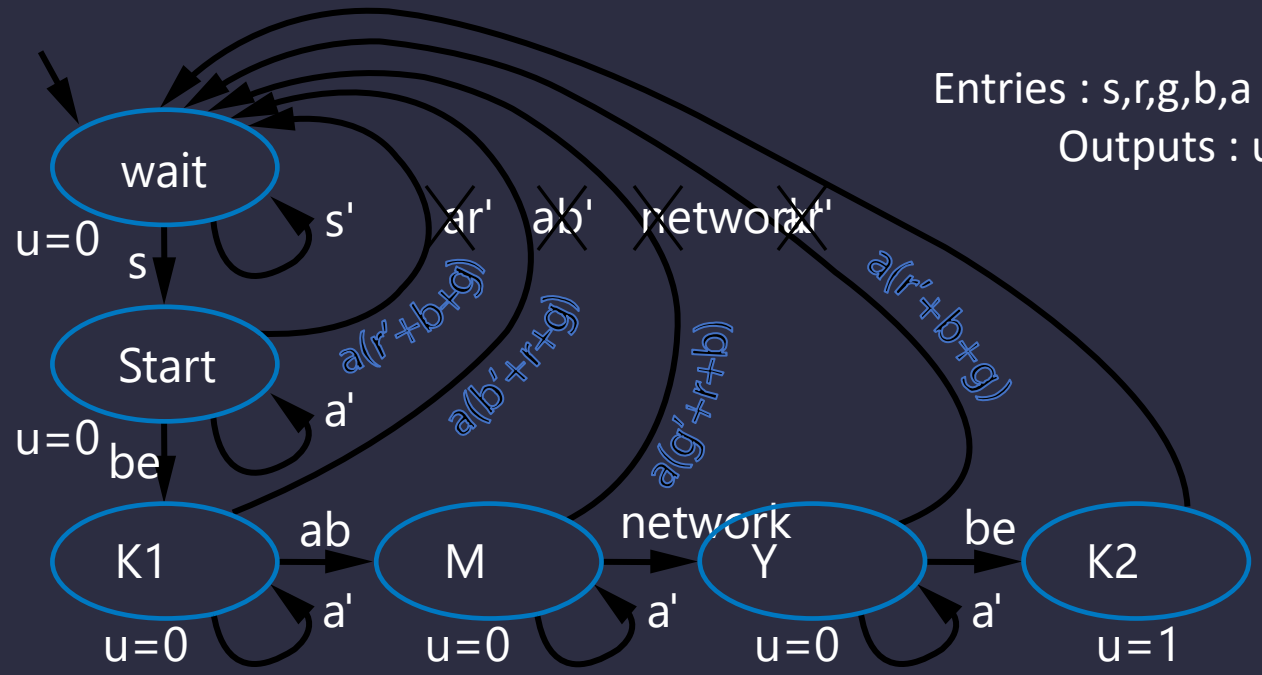
FSM Example: Code Detector

- A door is only opened when the following button sequence is pressed (u=1)
  - start , red , blue , green , red
- *s, r, g, b*
  - If input a, buttons of other color
- FSM
  - While in wait state, start when start
  - When start is pressed
    - If it's red, go to K1
    - Then, if it's blue, M
    - Then, if green, Y
    - Then, if it's red, K2
      - In this case, the signal to open the door is generated (u=1)
    - Any wrong input will revert to wait state





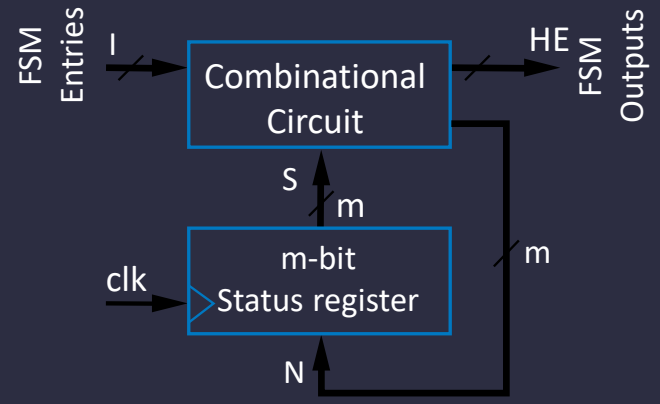
FSM Example: Code Detector, Improvements

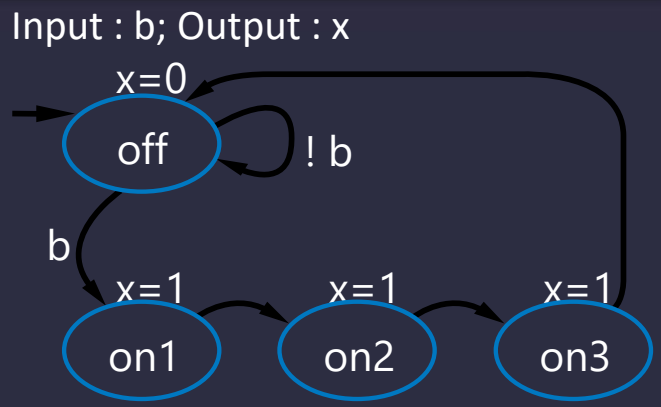


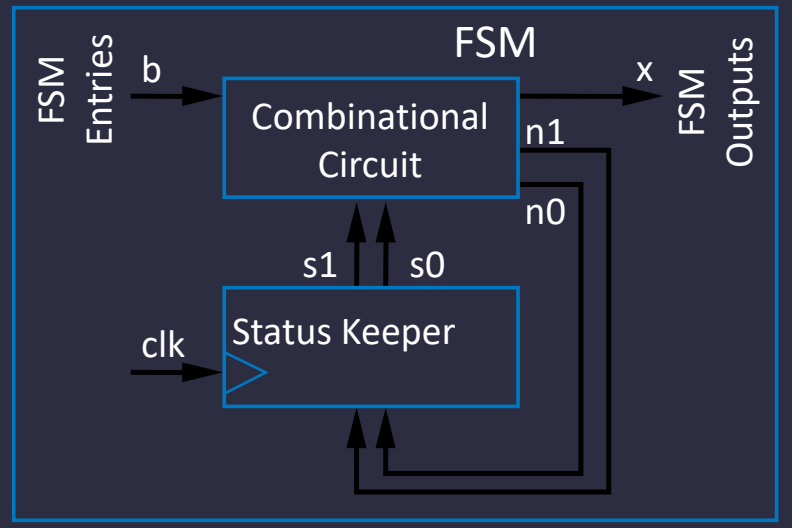
In the new state machine, pressing the wrong button returns to the "Wait" state.

FSM Architecture

- How is FSM implemented as a sequential circuit ?
  - Status register - to keep the current state
  - Combinational circuit – to calculate output and next state

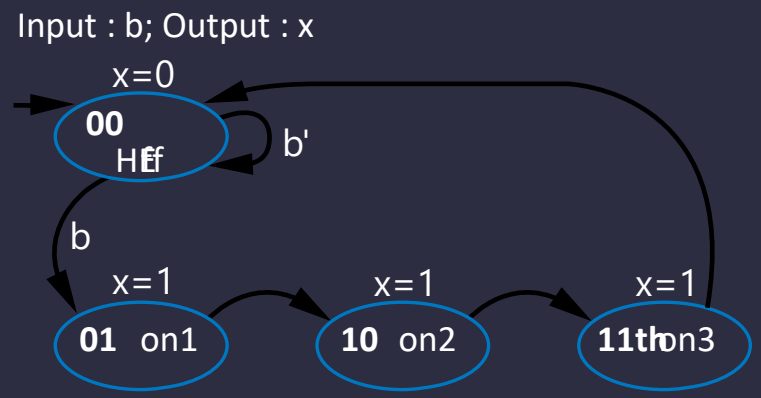


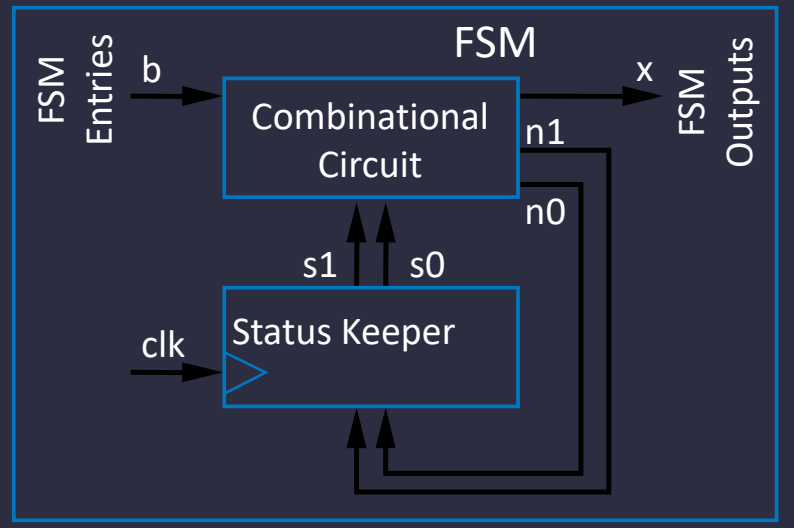




FSM Design Example

- Step 1: Analyze the requirements
- Step 2: Extract architecture
  - 2-bit state register (for 4 states)
  - Input b, output x
  - Next status signals n1, n0
- Step 3: Encode the states

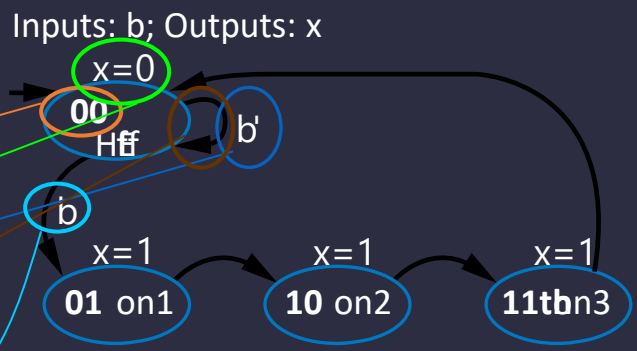


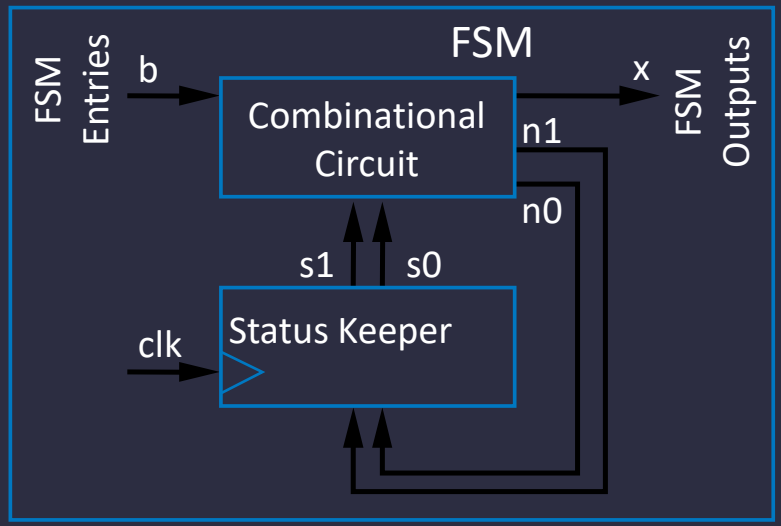


FSM Design Example

- Step 4: Create a Truth Table

	Entries			Outputs		
	s1	s0	b	x	n1	n0
Off	0	0	0	0	0	0
	0	0	1	0	0	1
On1	0	1	0	1	1	0
	0	1	1	1	1	0
On2	1	0	0	1	1	1
	1	0	1	1	1	1
On3	1	1	0	1	0	0
	1	1	1	1	0	0

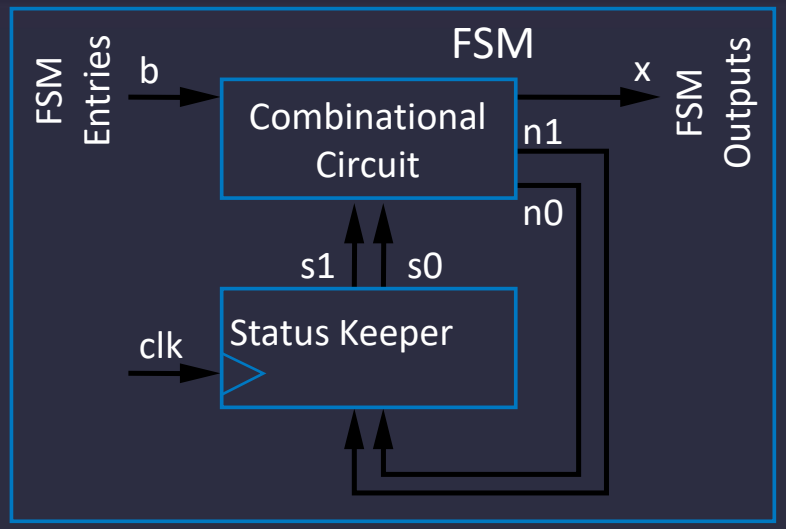




FSM Design Example

- Step 5: Combinational Circuit Implementation

	Entries			Outputs		
	s1	s0	b	x	n1	n0
Off	0	0	0	0	0	0
	0	0	1	0	0	1
On1	0	1	0	1	1	0
	0	1	1	1	1	0
On2	1	0	0	1	1	1
	1	0	1	1	1	1
On3	1	1	0	1	0	0
	1	1	1	1	0	0



x = s1 | s0

n1 = s1's0b' | s1's0b | s1s0'b' | s1s0'b

n1 = s1's0 | s1s0'

n0 = s1's0'b | s1s0'b' | s1s0'b

n0 = s1's0'b | s1s0'

FSM Design Example

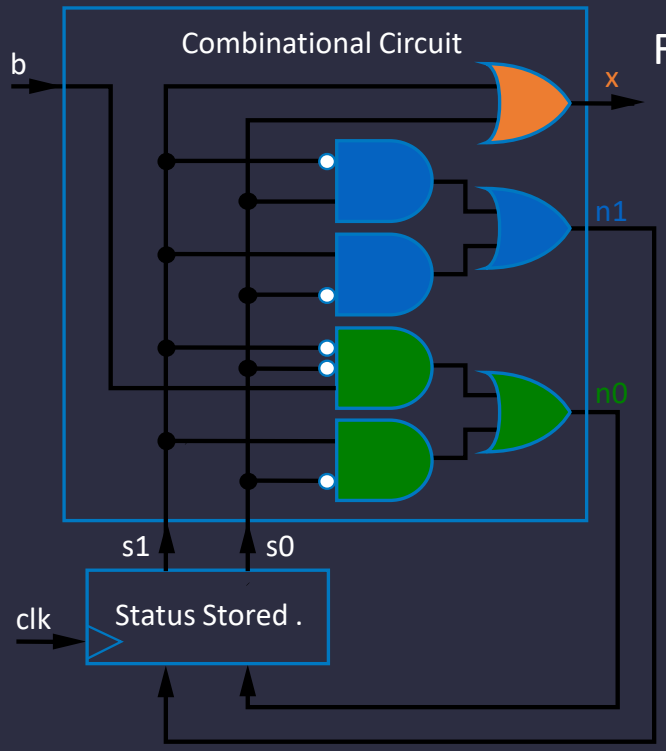
- Step 5: Build the combinational circuit

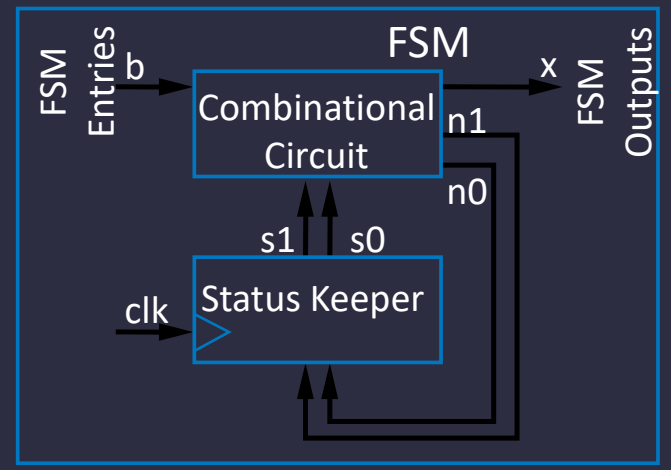
	Entries			Outputs		
	s1	s0	b	x	n1	n0
Off	0	0	0	0	0	0
	0	0	1	0	0	1
On1	0	1	0	1	1	0
	0	1	1	1	1	0
On2	1	0	0	1	1	1
	1	0	1	1	1	1
On3	1	1	0	1	0	0
	1	1	1	1	0	0

x = s1 | s0

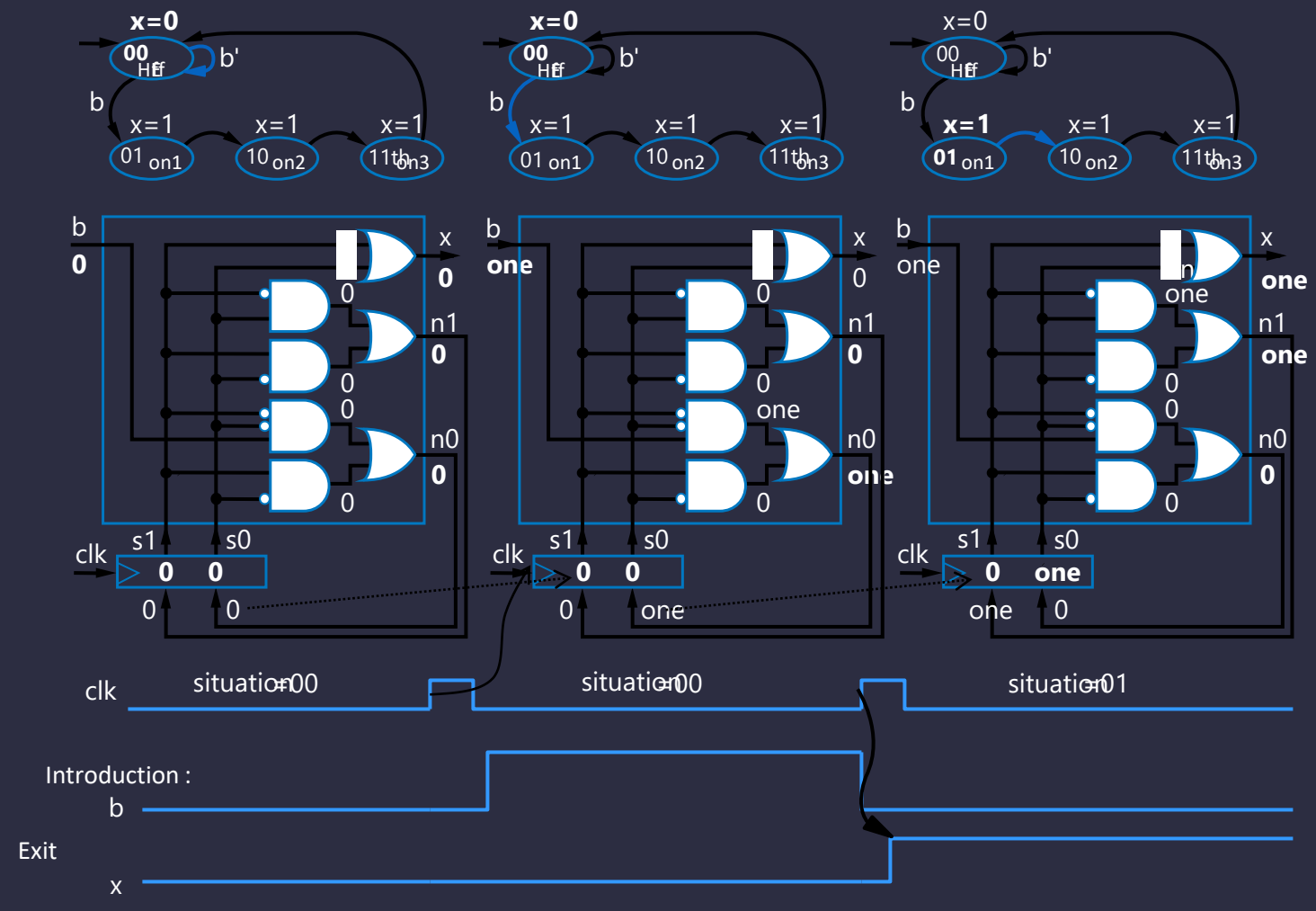
n1 = s1's0 | s1s0'

n0 = s1's0'b | s1s0'

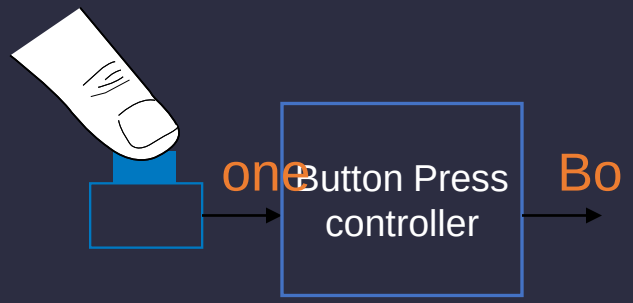


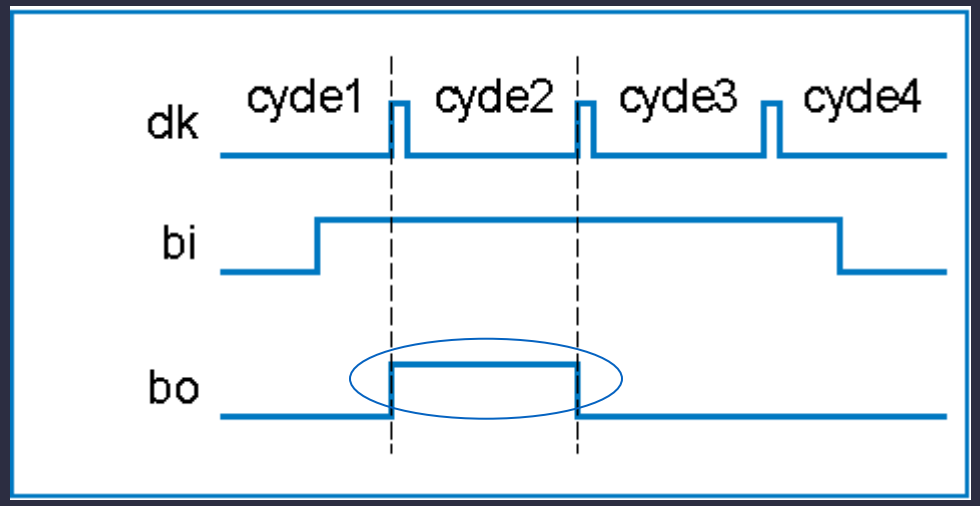


Internal Structure of



Button Filter Module Example

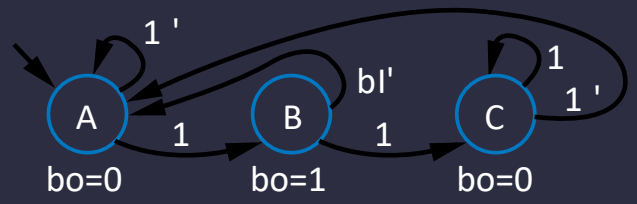




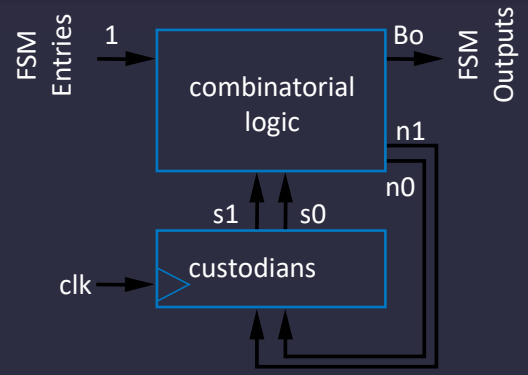
- cycle when the button is pressed A design that produces a pulse is desired.

Button Filter Module Example

FSM Inputs : bi; FSM outputs : bo

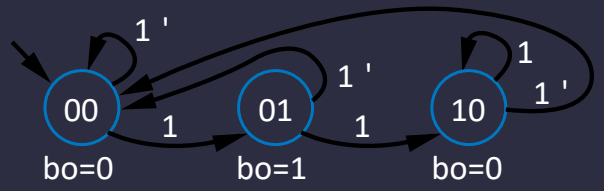


Step 1: FSM



Step 2: Build architecture

FSM input : bi; FSM output : bo

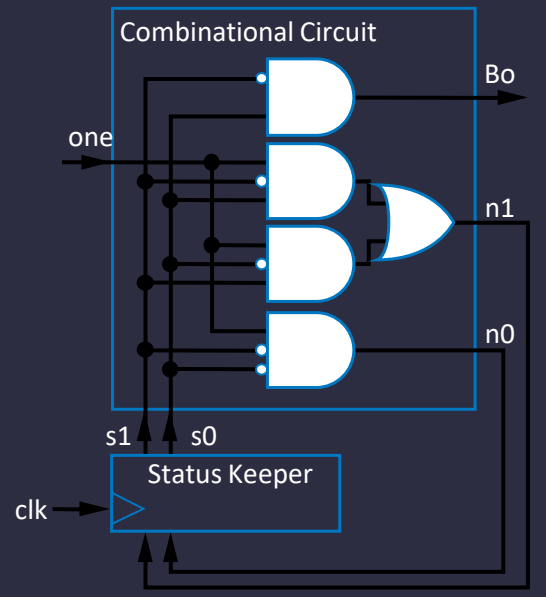


Step 3: State coding

Entries			Outputs		
s1	s0	bi	n1	n0	bo
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	0	0

Step 4: Status Chart

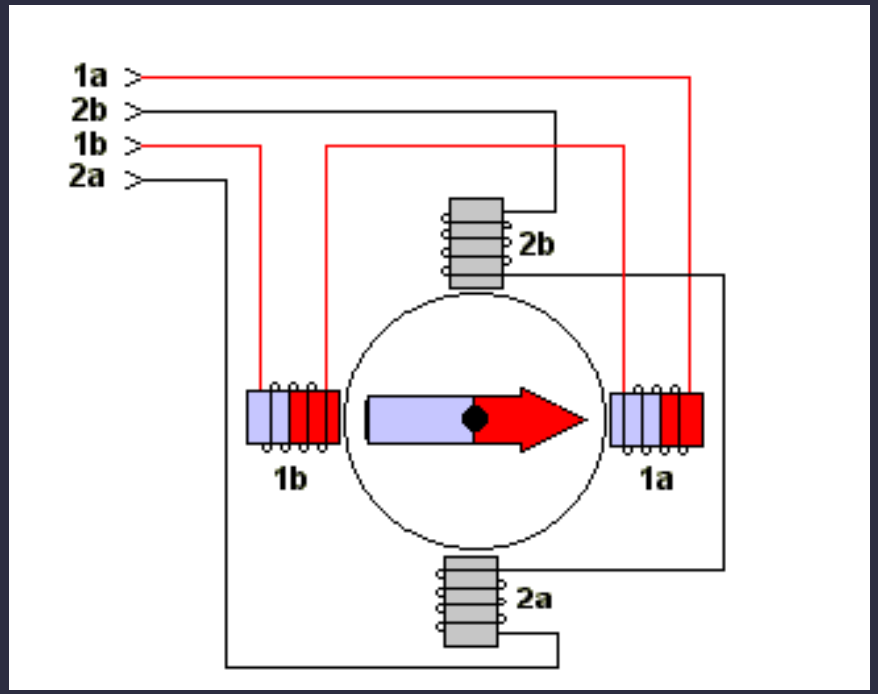
n1 = s1's0bi | s1s0bi
n0 = s1's0'bi
bo = s1's0bi' | s1's0bi = s1s0



Step 5: Build Circuit

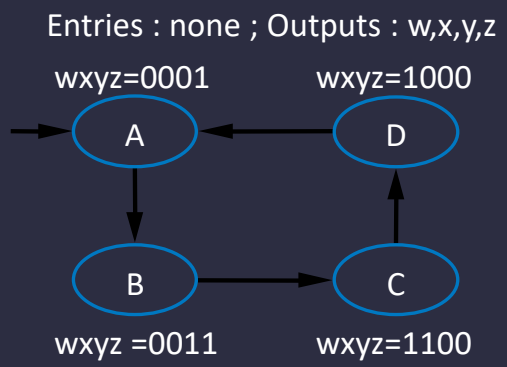
FSM Example

- Design an FSM that outputs 0001, 0011, 1100, 1000 sequentially.
  - For example, for stepper motor control

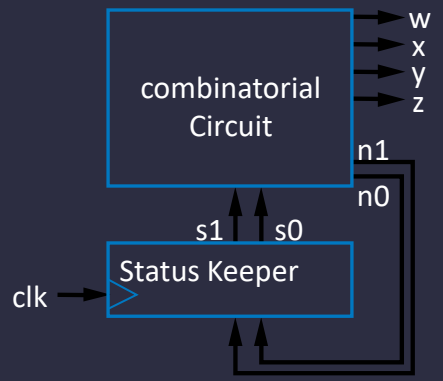


FSM Example

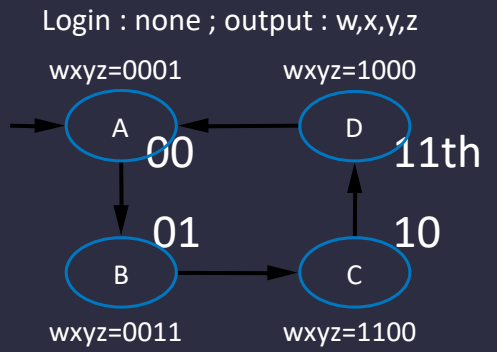
- Design an FSM that outputs 0001, 0011, 1100, 1000 sequentially.
  - For example, for stepper motor control



Step 1: Creating the FSM



Step 2: Creating Architecture

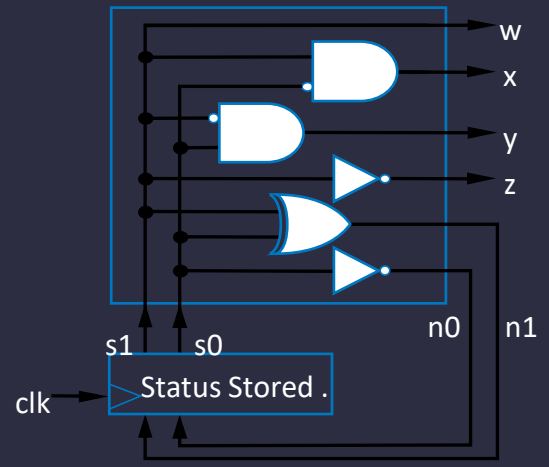


Step 3: State Coding

Entries		Outputs					
	s1 s0	w	x	y	z	n1	n0
A	0 0	0	0	0	1	0	1
B	0 1	0	0	1	1	1	0
C	1 0	1	1	0	0	1	1
D	1 1	1	0	0	0	0	0

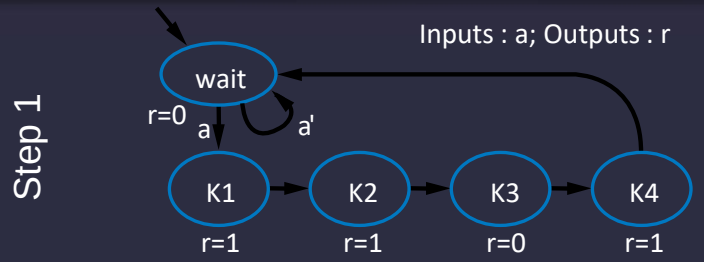
Step 4: Create a Truth Table

w = s1
x = s1s0'
y = s1's0
z = s1'
n1 = s1 x or s0
n0 = s0'

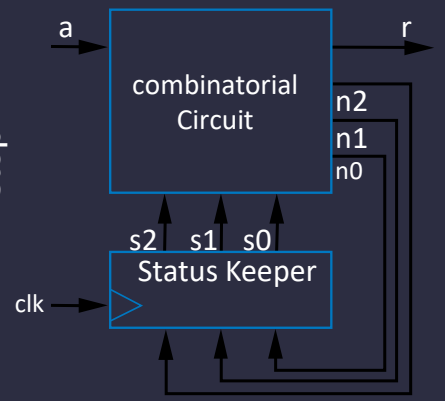


FSM Example, Car Key

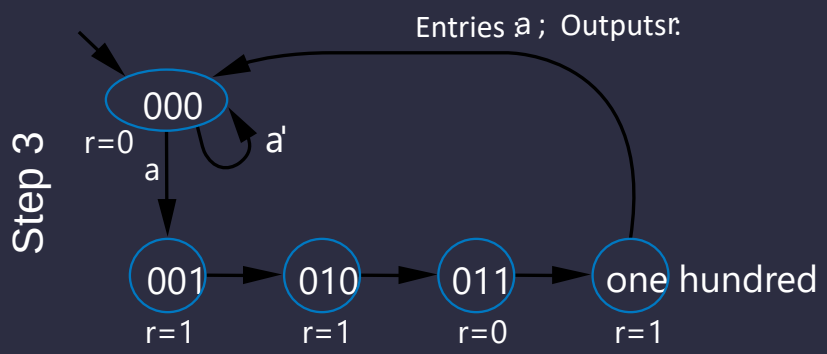
Step 1



step 2



Step 3

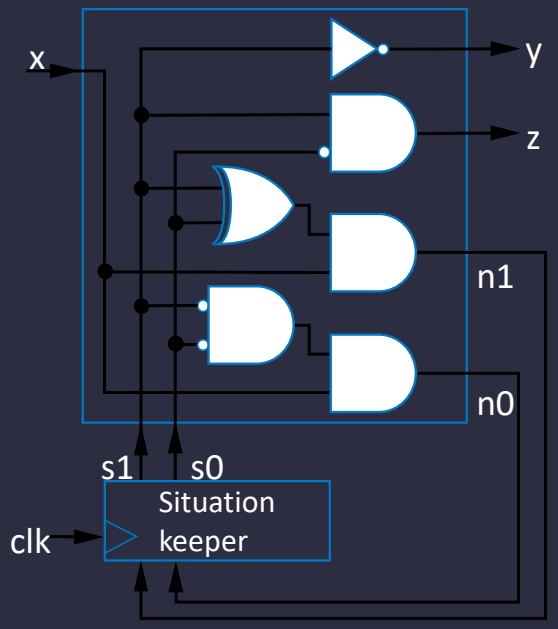


	Entries				Outputs			
	s2	s1	s0	a	r	n2	n1	n0
wait	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	1
K1	0	0	1	0	1	0	1	0
	0	0	1	1	1	0	1	0
K2	0	1	0	0	1	0	1	1
	0	1	0	1	1	0	1	1
K3	0	1	1	0	0	1	0	0
	0	1	1	1	0	1	0	0
K4	1	0	0	0	1	0	0	0
	1	0	0	1	1	0	0	0
-	1	0	1	0	0	0	0	0
	1	0	1	1	0	0	0	0
	1	1	0	0	0	0	0	0
	1	1	0	1	0	0	0	0
	1	1	1	0	0	0	0	0
	1	1	1	1	0	0	0	0

Step 4

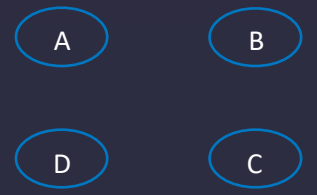
Reverting to

What does this circuit do?

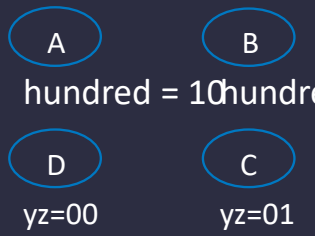


y=s1'
z = s1s0'
n1=(s1 x or s0)x
n0=(s1'*s0')x

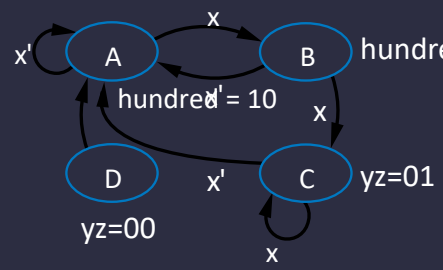
	Entries			Outputs			
	s1	s0	x	n1	n0	y	z
A	0	0	0	0	0	1	0
	0	0	1	0	1	1	0
B	0	1	0	0	0	1	0
	0	1	1	1	0	1	0
C	1	0	0	0	0	0	1
	1	0	1	1	0	0	1
D	1	1	0	0	0	0	0
	1	1	1	0	0	0	0



situations



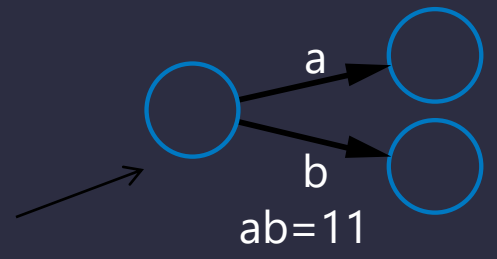
States and exits



States, exits and transitions

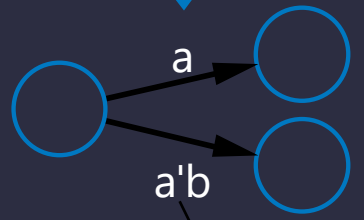
FSM Design Errors

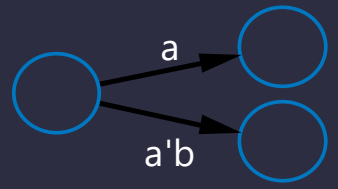
- *In conditions in transitions between states, only one transition condition must be true at a time*



ab=11

What will be the next situation?





ab=00?

What will be the next situatio