CS221: Digital Design

Finite State Machine (Examples)

A. Sahu

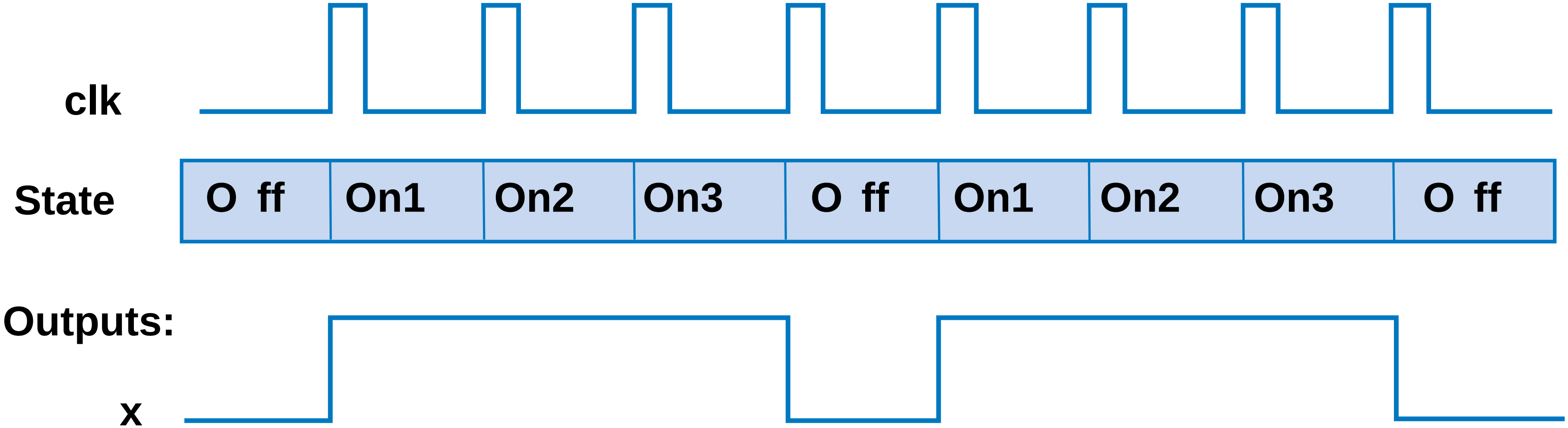
Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

FSM Example 6:
Counter that repeat 0,1,1,1,

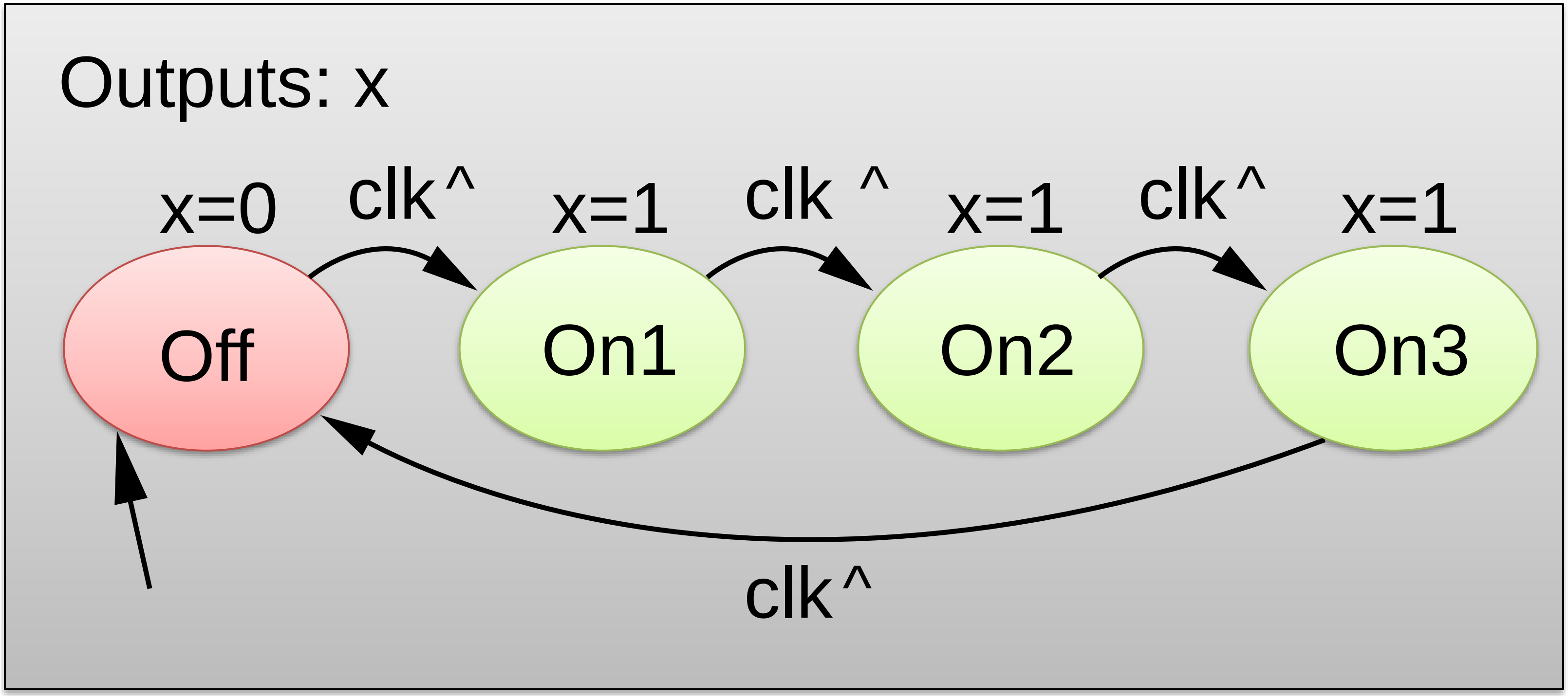
FSM Example: 0,1,1,1,repeat

- Want 0, 1, 1, 1, 0, 1, 1, 1, ...
  - Each value for one clock cycle
- Can describe as FSM: Four states, Transition on rising clock edge to next state



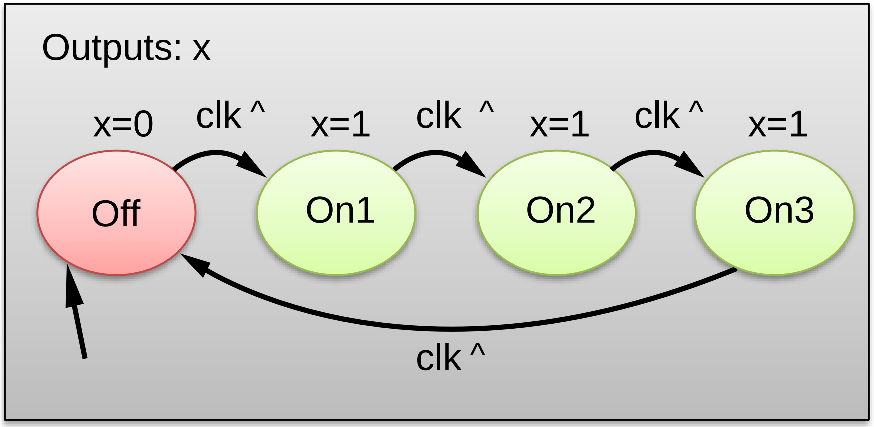
FSM Example: 0,1,1,1,repeat

- Want 0, 1, 1, 1, 0, 1, 1, 1, ...
  - Each value for one clock cycle
- Can describe as FSM: Four states, Transition on rising clock edge to next state



FSM Example: 0,1,1,1,repeat

- Can describe as FSM: Four states, Transition on rising clock edge to next state



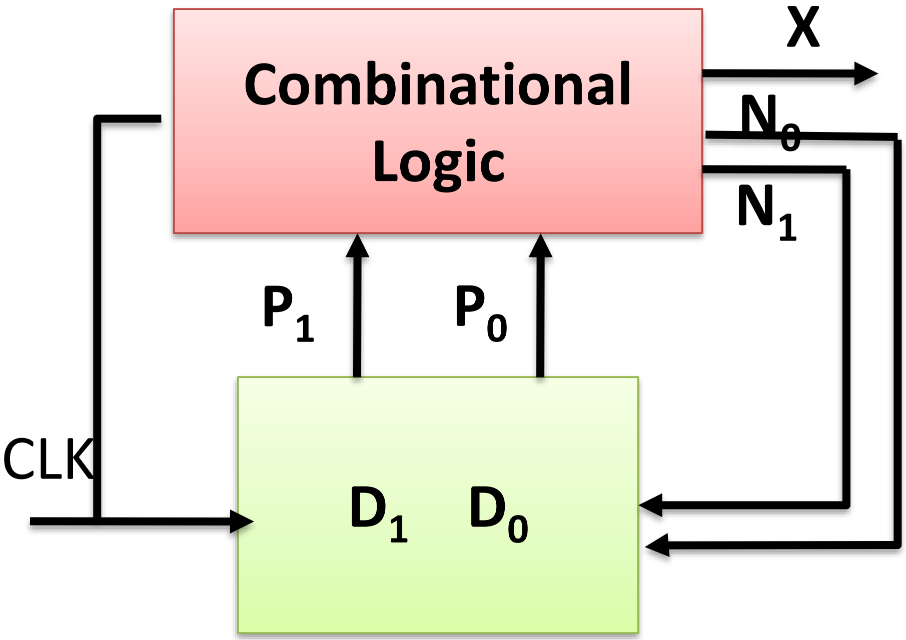
PS	NS	X
00	01	0
01	10	1
10	11	1
11	00	1

Require two FF to store states

Controller of FSM Example: 0,1,1,1,repeat

Input			Output		
CLK	PS		NS		X
	P_1	P_0	N_1	N_0	
RE 1	0	0	0	1	0
RE 1	0	1	1	0	1
RE 1	1	0	1	1	1
RE 1	1	1	0	0	1

D-FF used to store the Present state



$X=P_1+P_0$

$N_1=P_0 \text{ xor } P_1$

$N_0=P_0'$

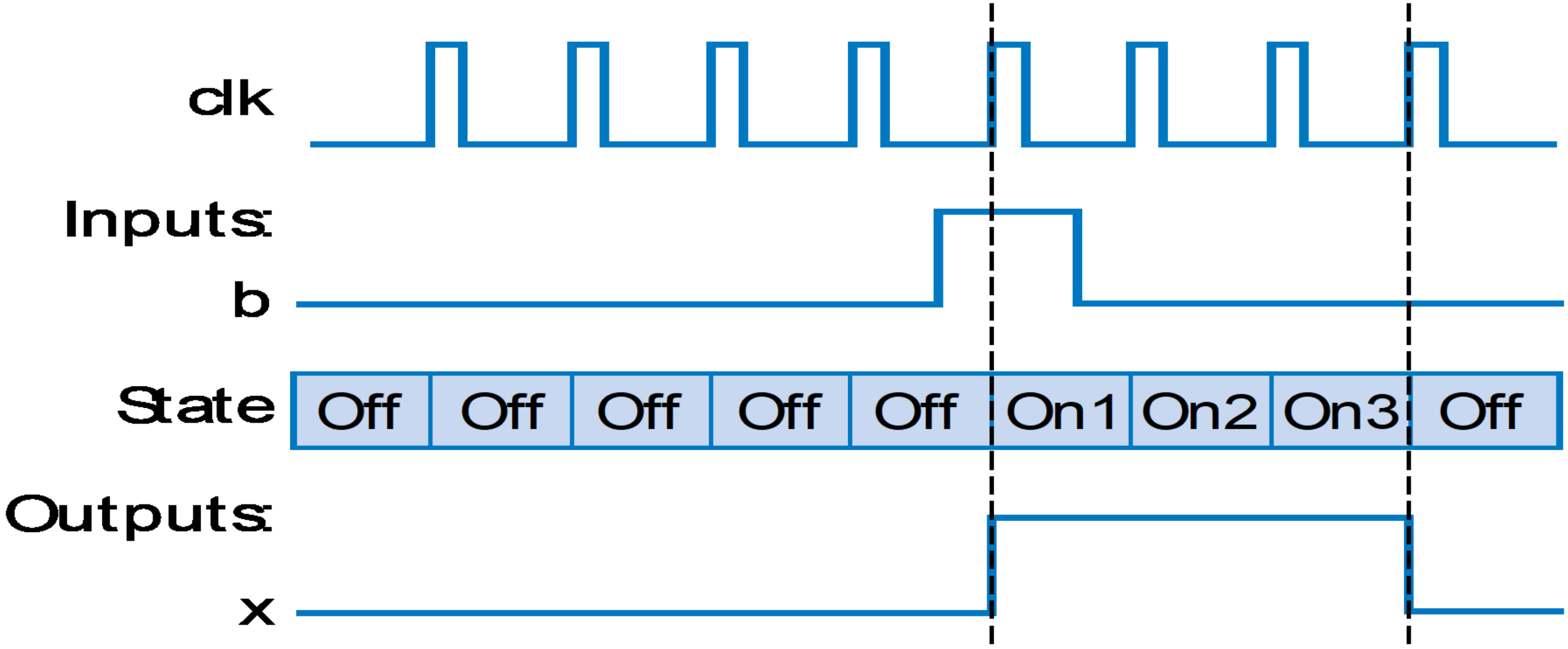
Rising Edge: Clock implicit

FSM Example 7:
Three-Cycles High Laser Timer

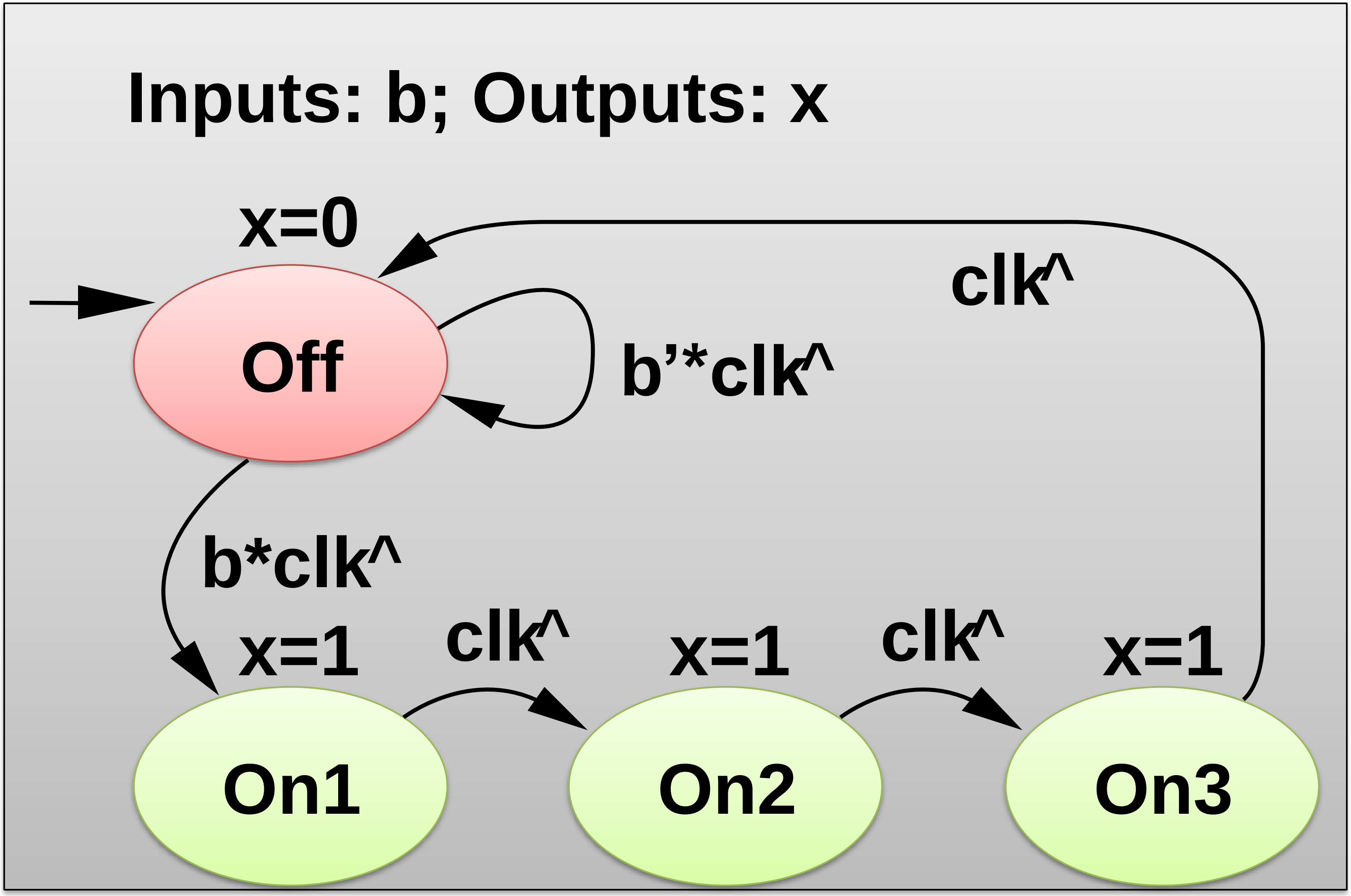
Extend FSM to Three-Cycles High Laser Timer

- Four states: Wait in “Off” state while b is 0 (b’)
- When b=1 (& rising clock edge), transition to On1
  - Sets X=1
  - On next two clock edges, transition to On2, then On3, which also set x=1
- So x=1 for three cycles after button pressed

Extend FSM to Three-Cycles High Laser Timer



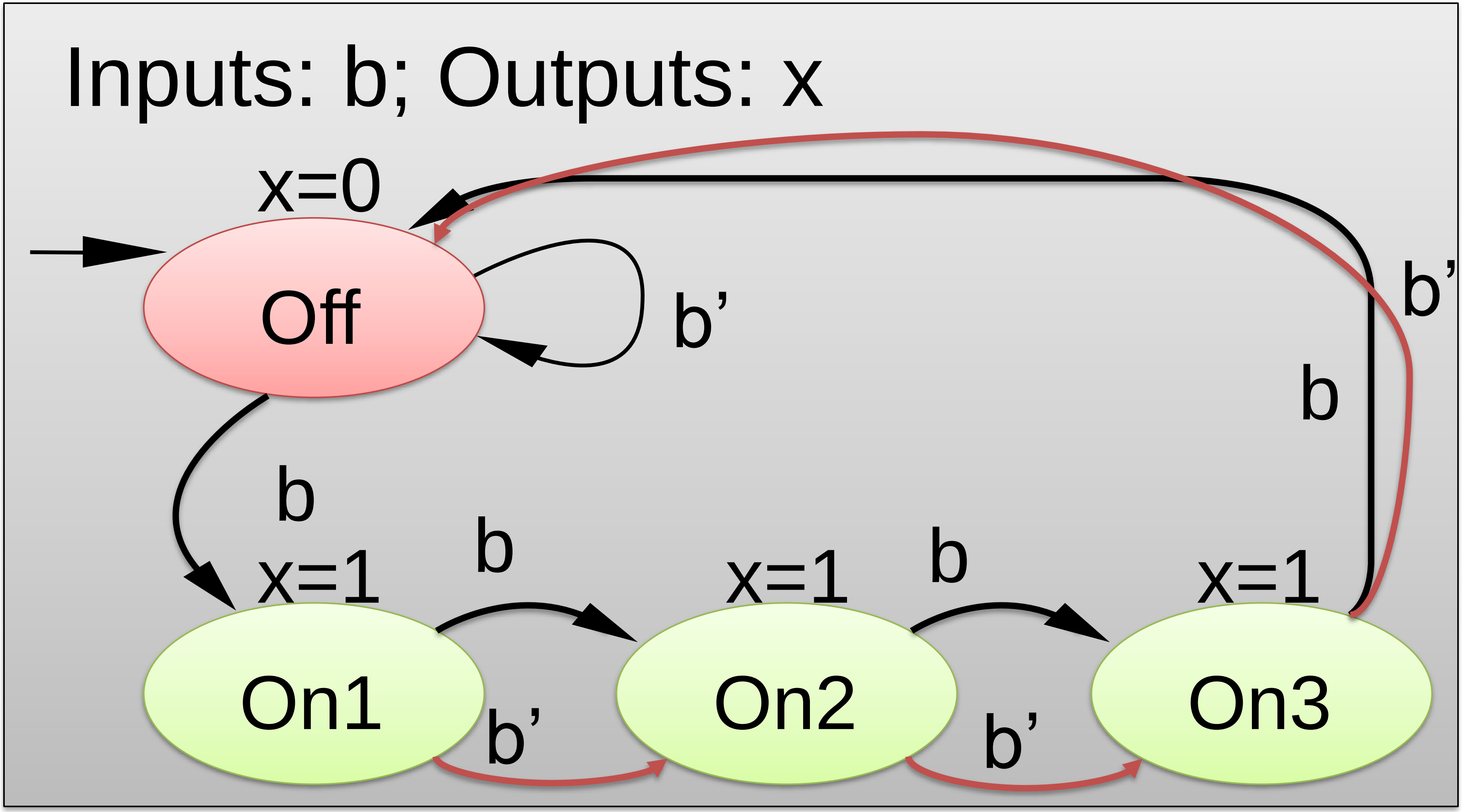
Extend FSM to Three-Cycles High Laser Timer



FSM Simplification: Rising Clock Edges Implicit

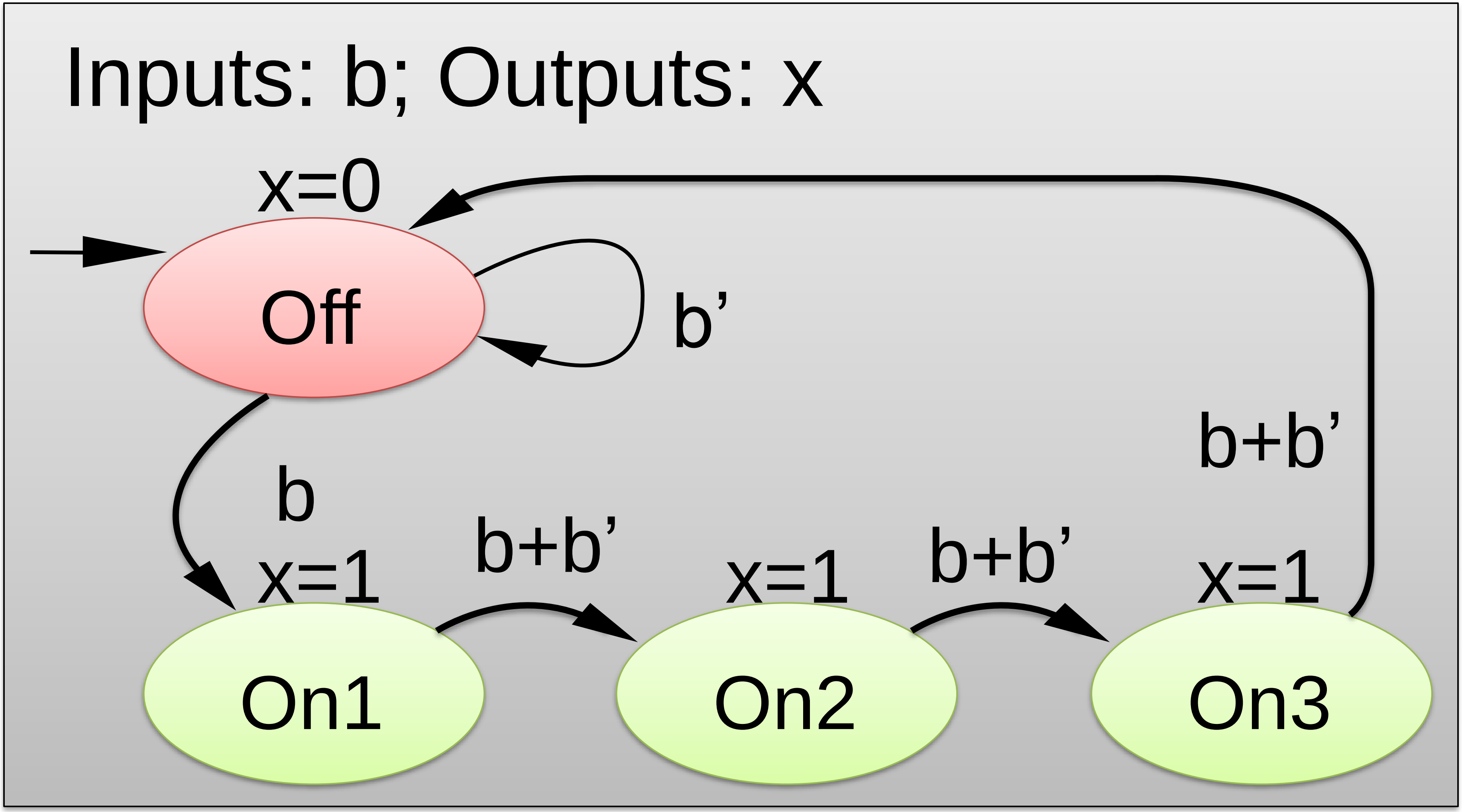
- Showing rising clock on every transition: cluttered
- Make implicit -- assume every edge has rising clock
- What if we wanted a transition *without* a rising edge
  - **Asynchronous FSMs -- less common, and advanced topic**
  - We consider ***synchronous*** FSMs
  - ***All*** **transition on rising edge**

FSM Simplification: Rising Clock Edges Implicit



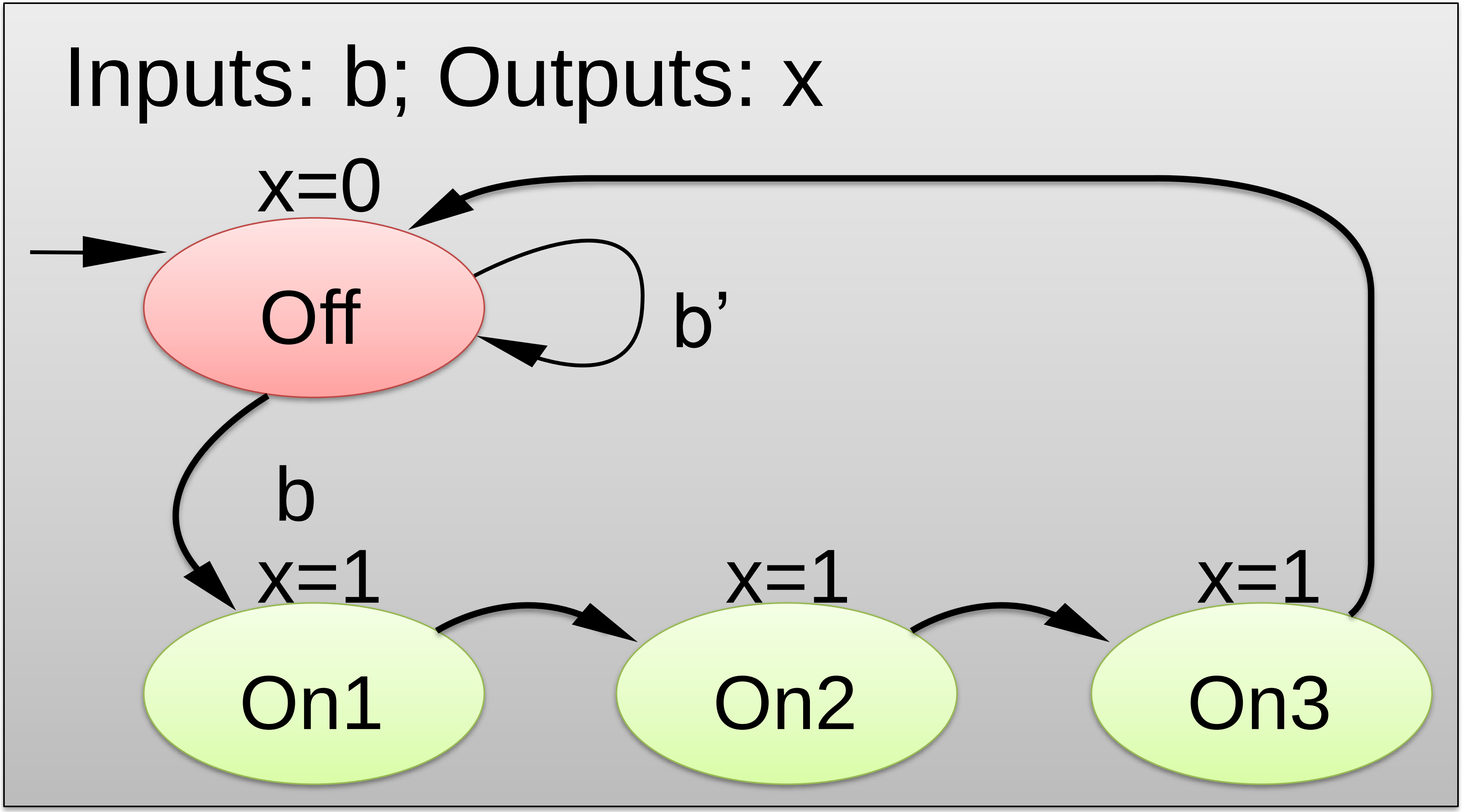
Value of b=1: 0111..repeat, ***Is this FSM is complete?***

FSM Simplification: Rising Clock Edges Implicit

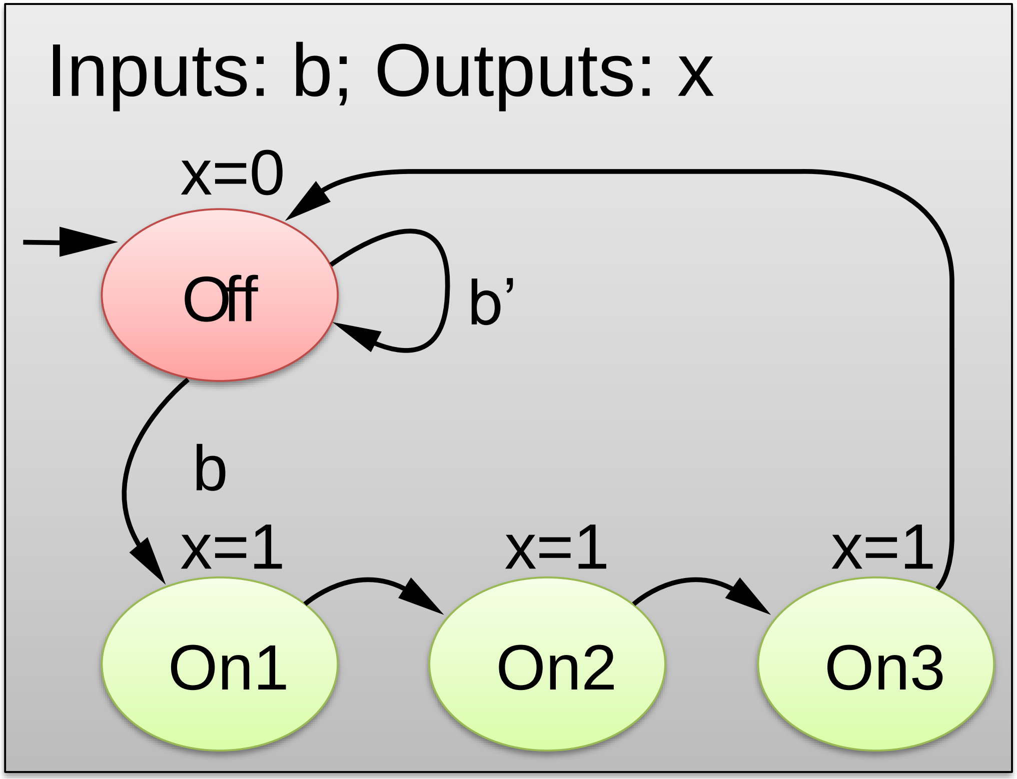


Value of b=1:   0111..repeat,   ***Is this FSM is complete?***

FSM Simplification: Rising Clock Edges Implicit



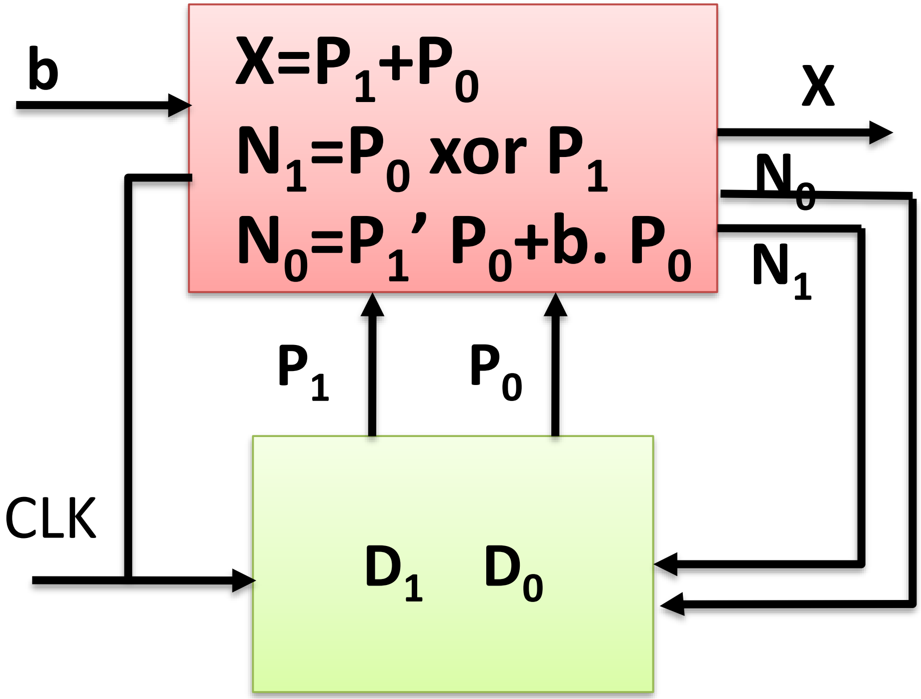
FSM Implementation : Three-Cycles High Laser Timer



PS	b	NS	X
00	0	00	0
00	1	01	0
01	x	10	1
10	x	11	1
11	x	00	1

FSM Implementation : Three-Cycles High Laser Timer

PS	b	NS	X
00	0	00	0
00	1	01	0
01	x	10	1
10	x	11	1
11	x	00	1



$X=P_1+P_0$

$N_1=P_0 \text{ xor } P_1$

$N_0=P_1' P_0+b. P_0$

Once we specify FSM for a problem/system

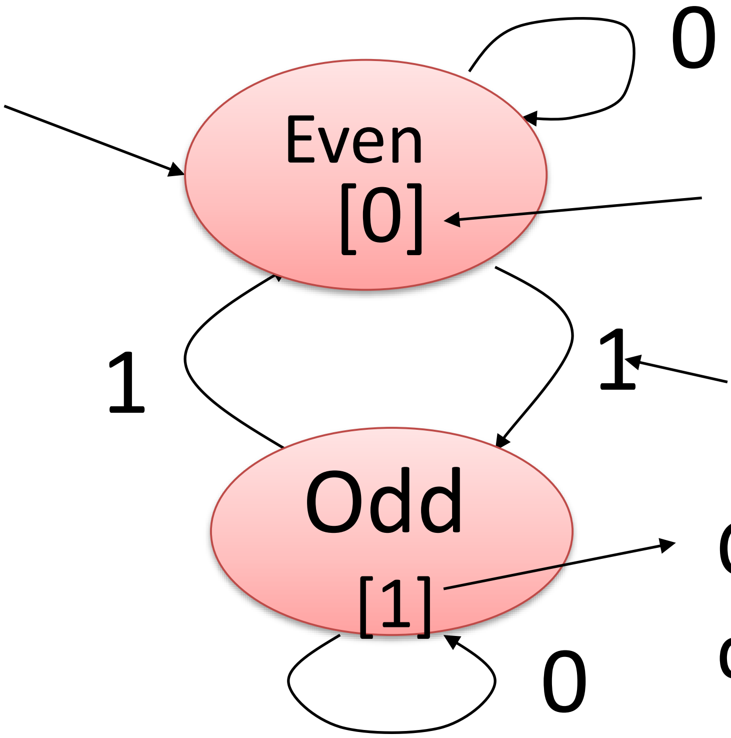
=== >

Implementation is not difficult

FSM Example 8:
Parity Encoder

FSM Example 8: Parity Encoder

- Input: 1 or 0 // entering as stream
- Out put: output a 1 when total number of 1 is even



Reset

Output=0’=1

Input

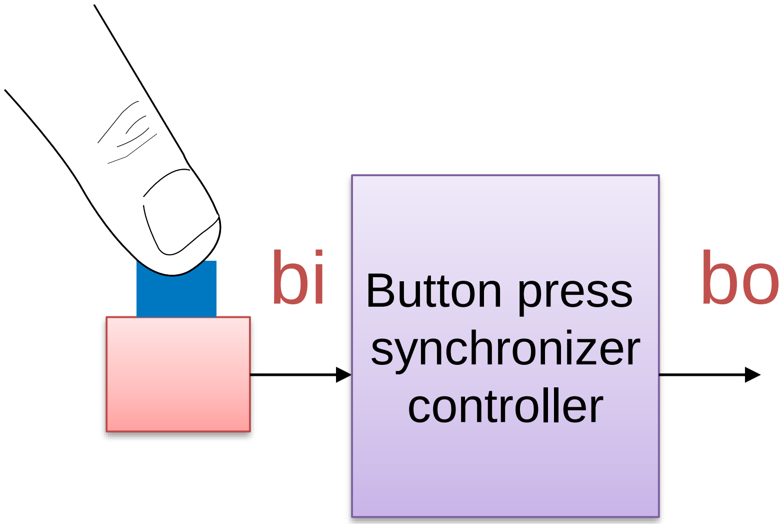
Output is dependent only on current state

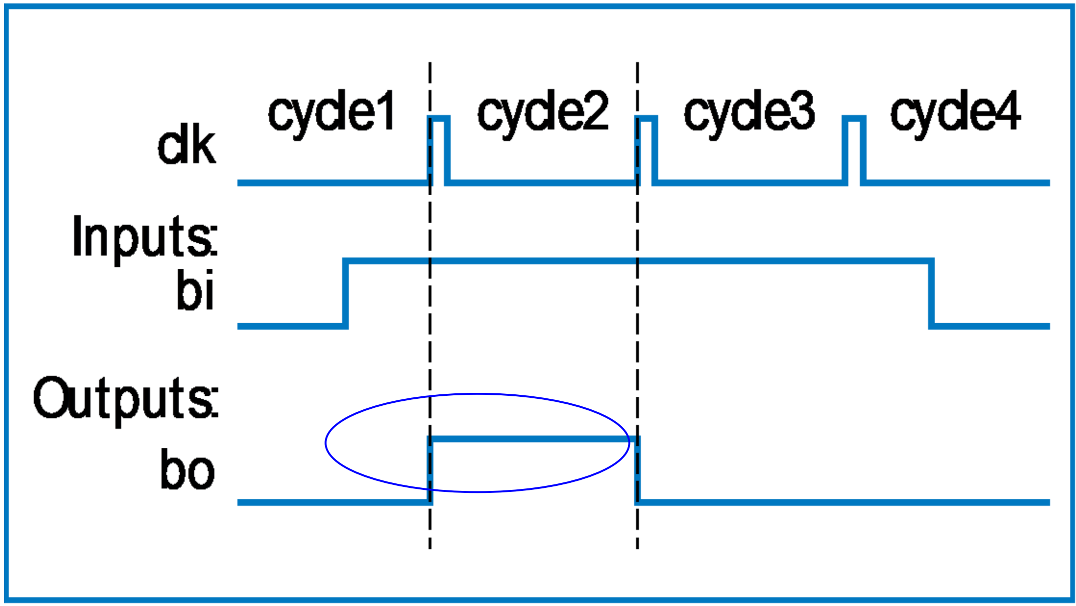
T- FF : designed using D-FF

FSM Example 9:
Button Press Synchronizer

Example 9 : Button Press Synchronizer

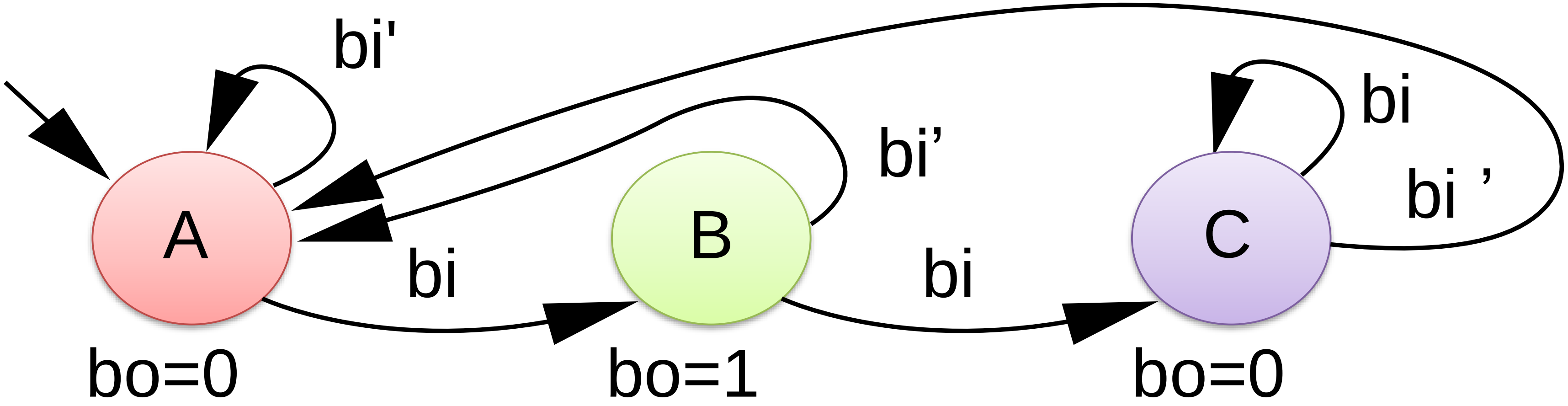
- **English Language Specification**
- **All most all the keyboards use this method**
- We want simple sequential circuit
  - Converts button press to single cycle duration
  - Regardless of length of time that button actually pressed





FSM Example 9 : Button Press Synchronizer

FSM inputs: bi; FSM outputs: bo



I am Off
When B=0

I am On in First
CLK and
When B=1

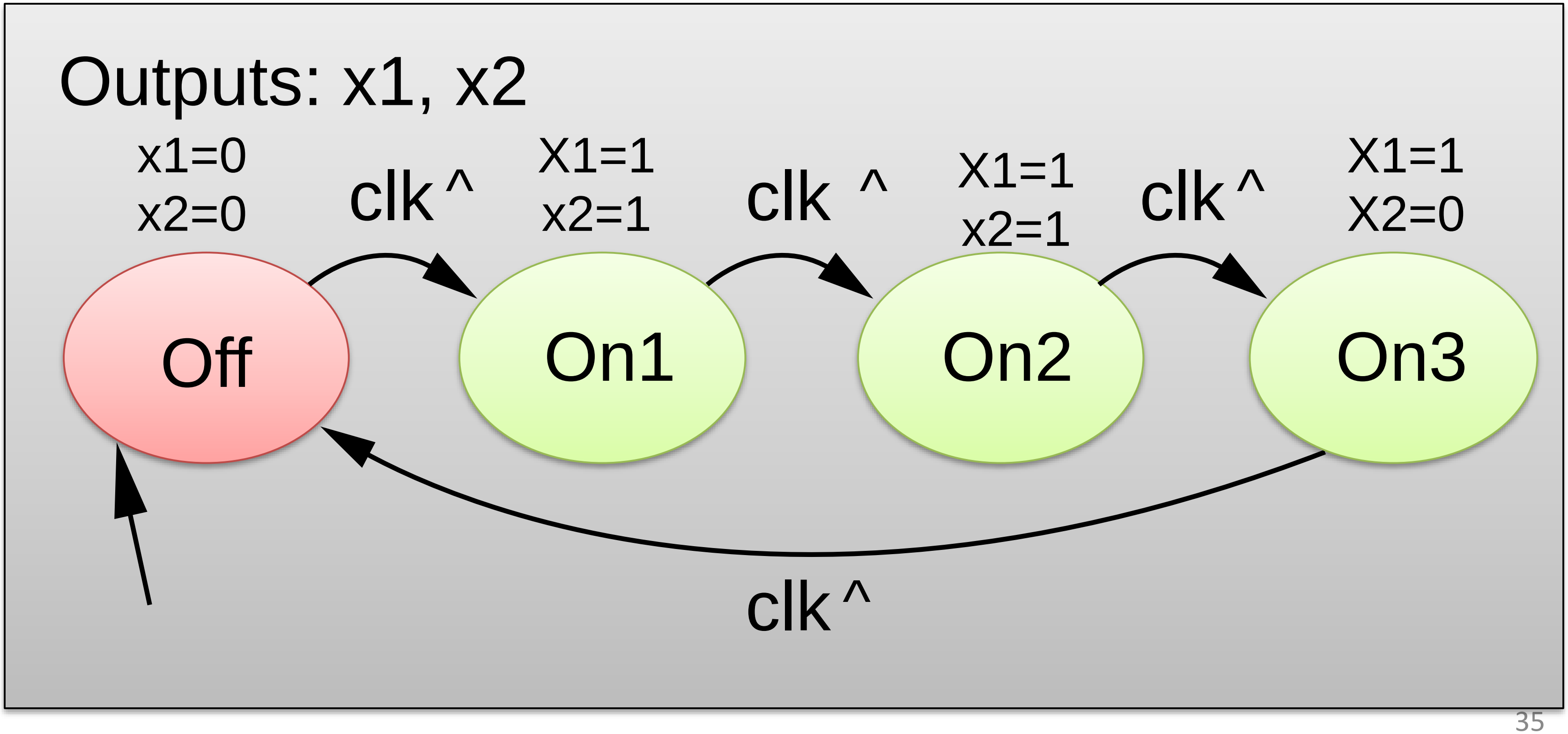
I am Off
Even if B=1

Step 1: Design FSM

FSM Example 10: Sequence generator

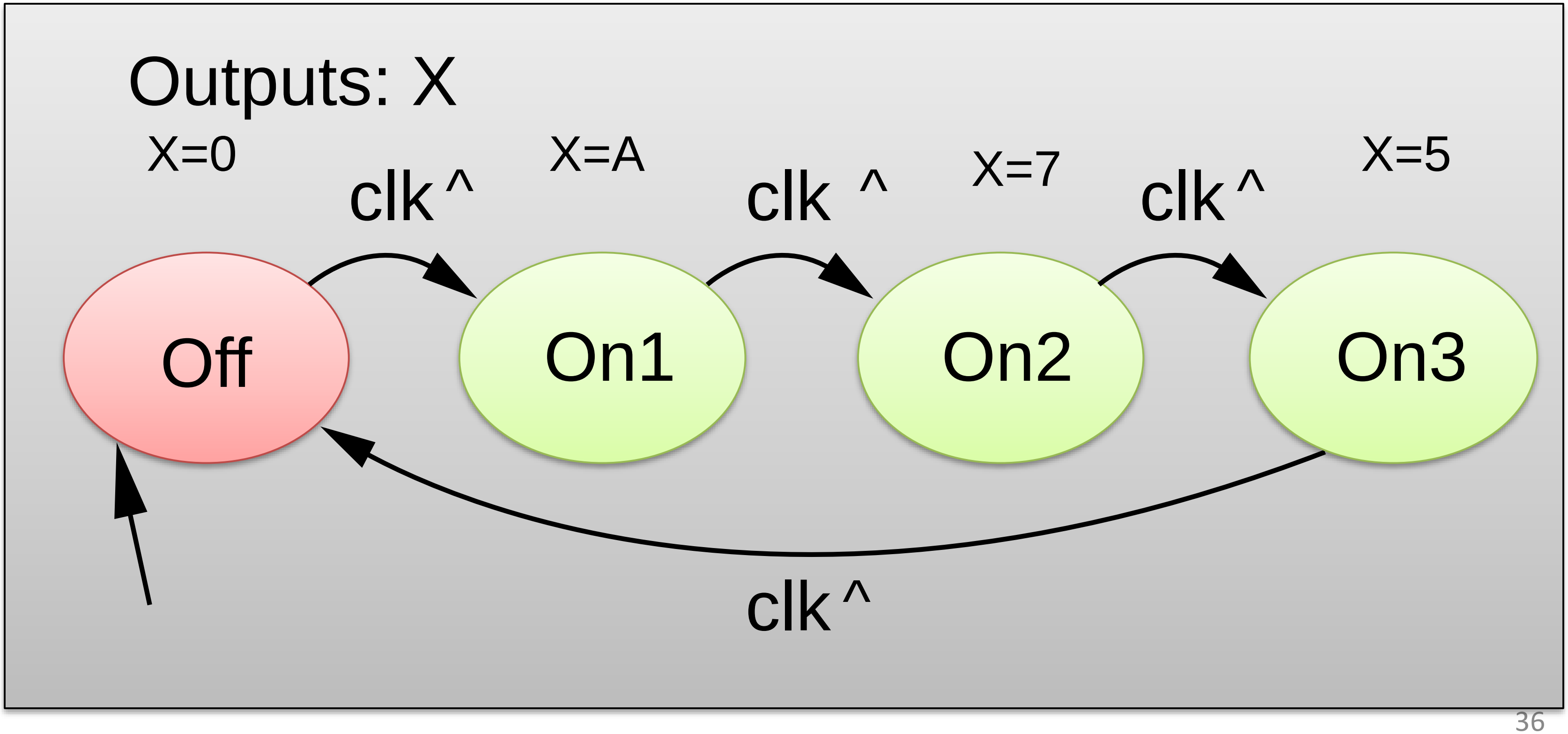
FSM Example 10: Sequence generator

- Generate two output sequence
  - X1= 0111….repeat
  - X2= 0110…repeat



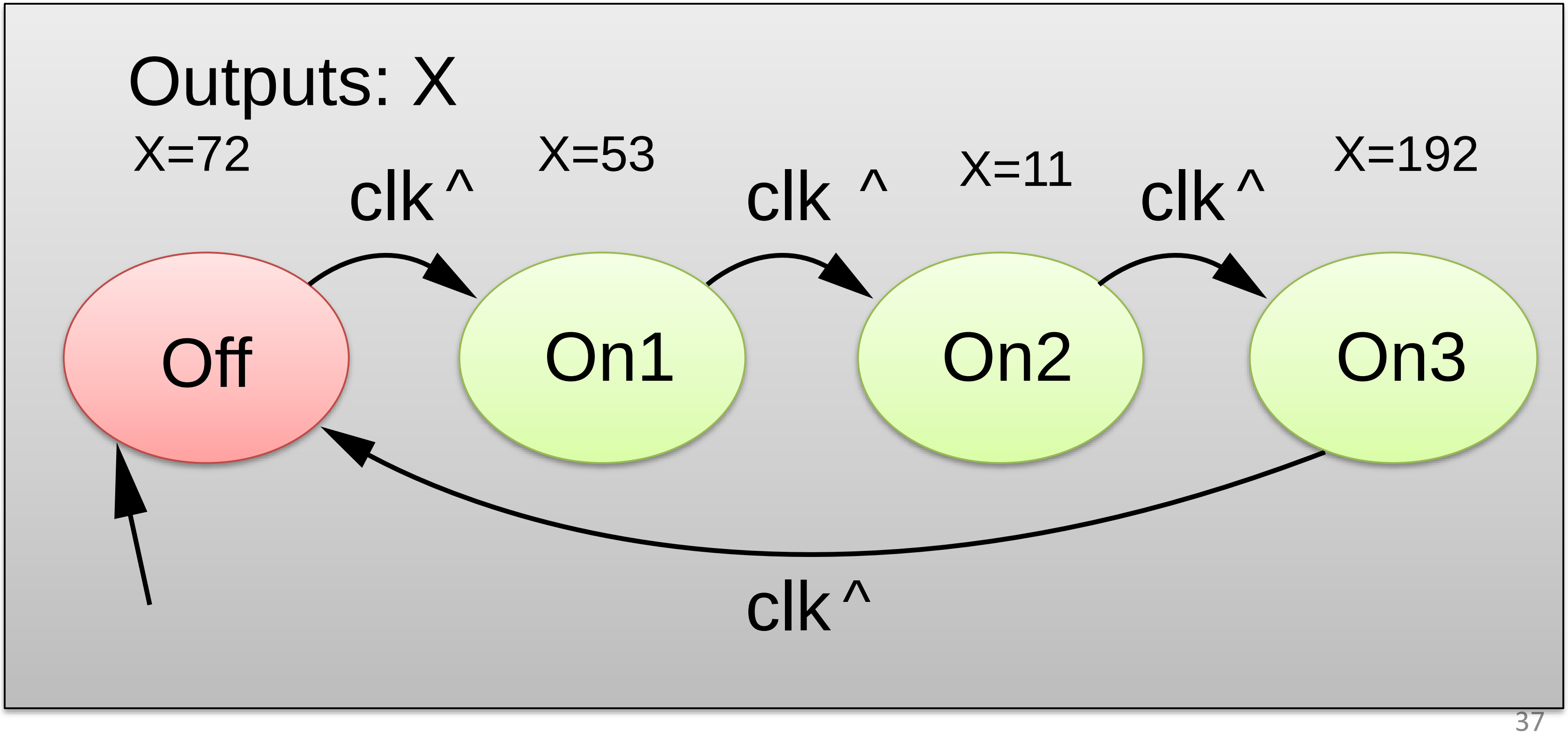
FSM Example 10-E1: Sequence generator

- Generate 4 bit integer output sequence
  - X= 0, A, 7, 5….repeat
  -



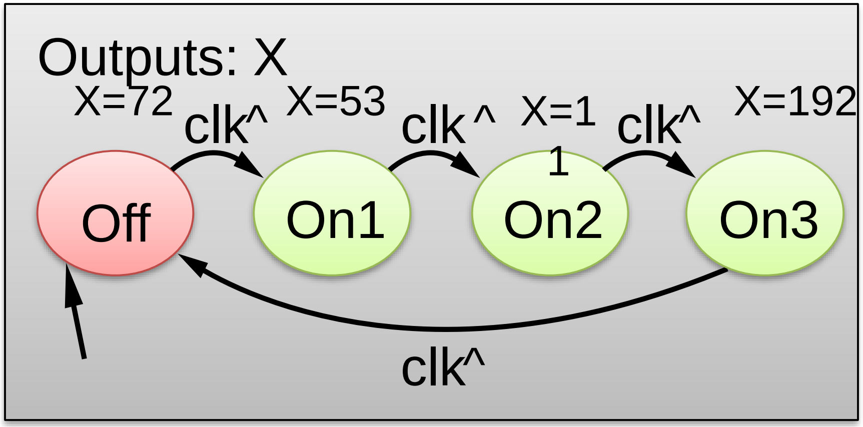
FSM Example 10-E2: Sequence generator

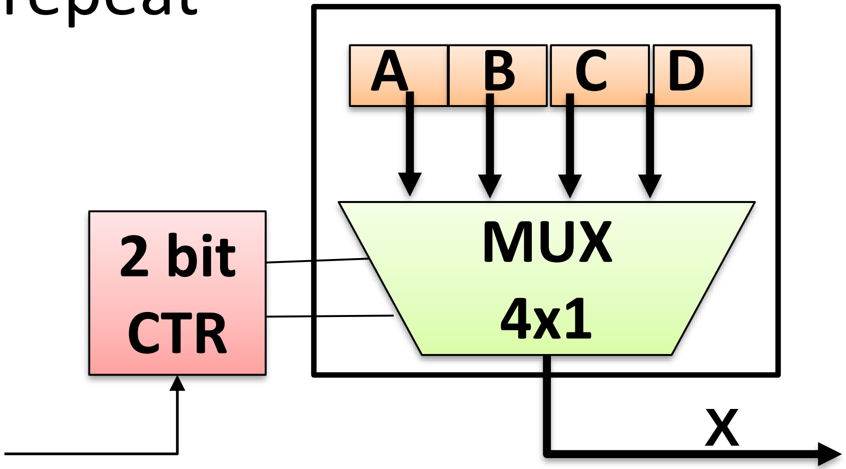
- Generate 8 bit integer output sequence
  - X= 72, 53, 11, 192,….repeat
  -



FSM Example 10-E2: Sequence generator

- Generate 8 bit integer output sequence
  - X= 72, 53, 11, 192,….repeat





A,B,C,D values can stored in 4 register (With Mux it act as memory)

FSM Output logic may be Simpler than Register and Mux

Simpler to Implemen Outputlogic: Eight 2 input binary function