CS221: Digital Design

http://jatinga.iitg.ernet.in/~asahu/cs221

Finite State Machine

A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

- Discussion about Quiz
  - Q1: JK FF, Q2: value just before clock edge,
  - Q3: FSM implementation using D-FF of 3 bit Grey Ctr
  - Q4: FSM of 3 bit Ctr, $Z_1Z_2Z_3$ are output based on state
  - Q5: FSM of 3 bit Gray Ctr, $Z_1Z_2Z_3$ are output based on state
- FSM Examples
- FSM of Register and Counter
- Design of Counters using FSM and other FFs

Outline

- Q3, Q4 and Q5
- Z1: 00101000 00101000 00101000….
- Z2: 11011011 11011011 11011011….
- Z3: 00110111 00110111 00110111….

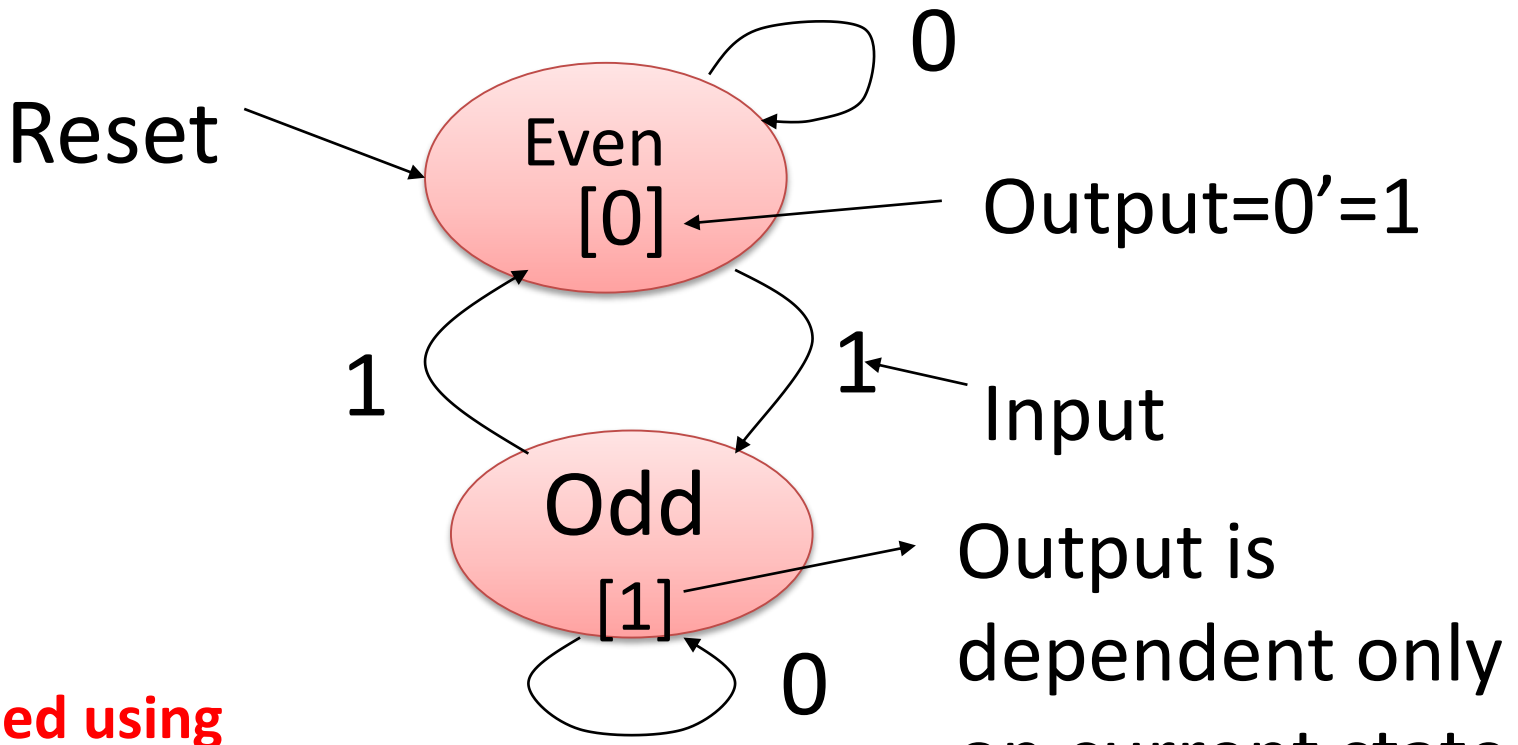
Grey Code:

CBA	CnBnAn	Z1Z2Z3
000	001	010
001	010	010
010	011	101
011	100	011
100	101	110
101	110	001
110	111	011
111	000	011

CBA	CnBnAn	Z1Z2Z3
000	001	010
001	011	010
010	110	101
011	010	011
100	000	110
101	100	001
110	111	011
111	101	011

FSM Example 1: Parity Encoder

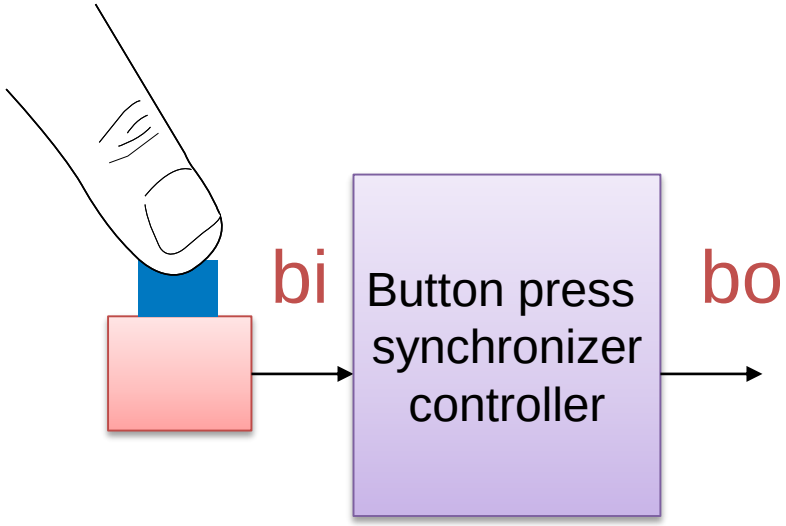
- Input: 1 or 0 // entering as stream
- Out put: output a 1 when total number of 1 is even

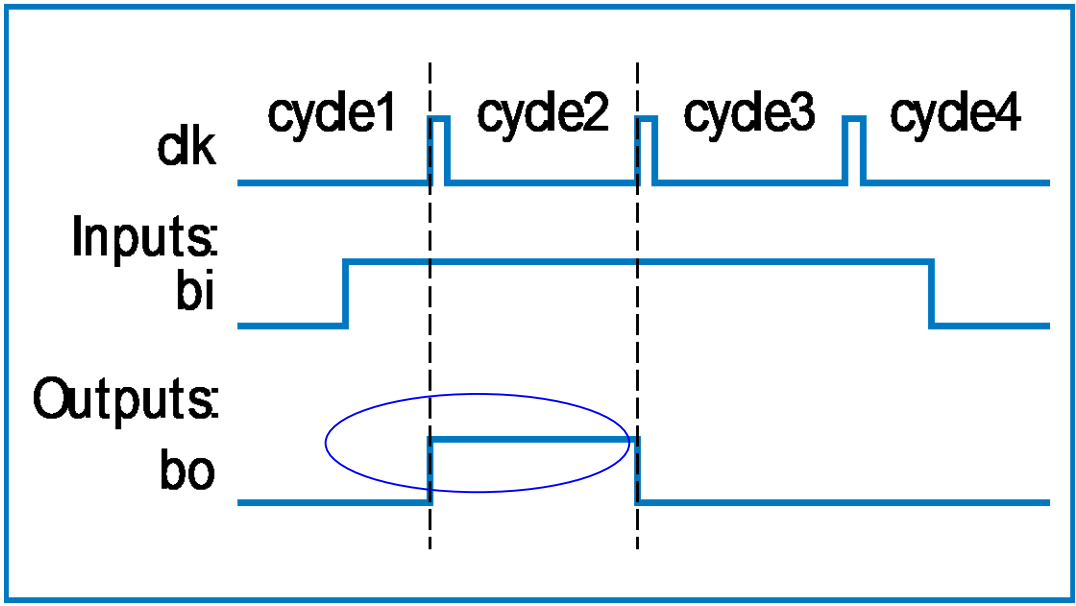


T- FF : designed using D-FF

FSM Example 2 : Button Press Synchronizer

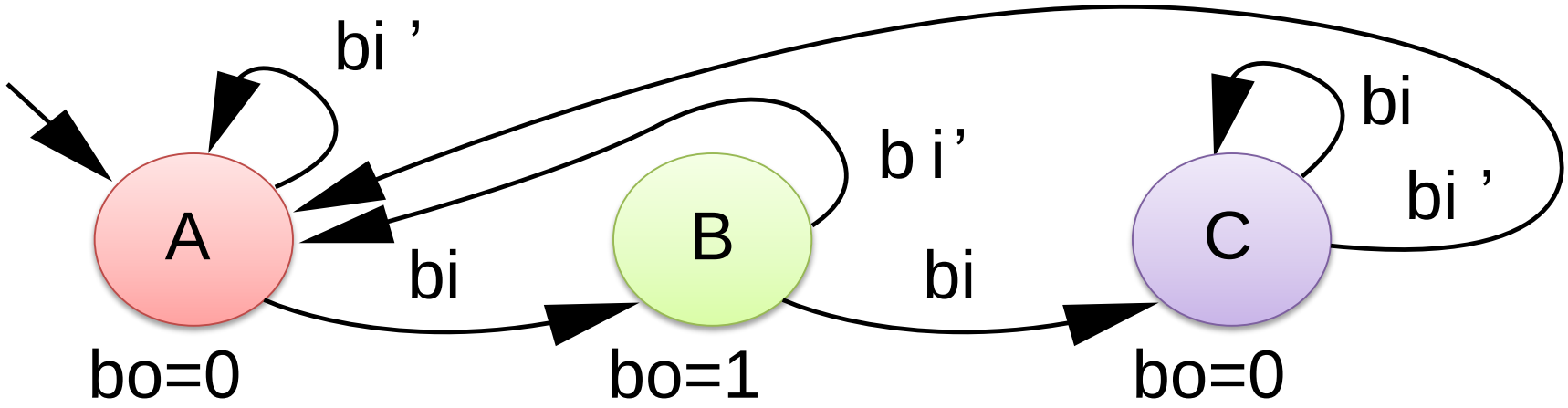
- **English Language Specification**
- We want simple sequential circuit
  - Converts button press to single cycle duration
  - Regardless of length of time that button actually pressed





FSM Example 2 : Button Press Synchronizer

FSM inputs: bi; FSM outputs: bo



I am Off When B=0

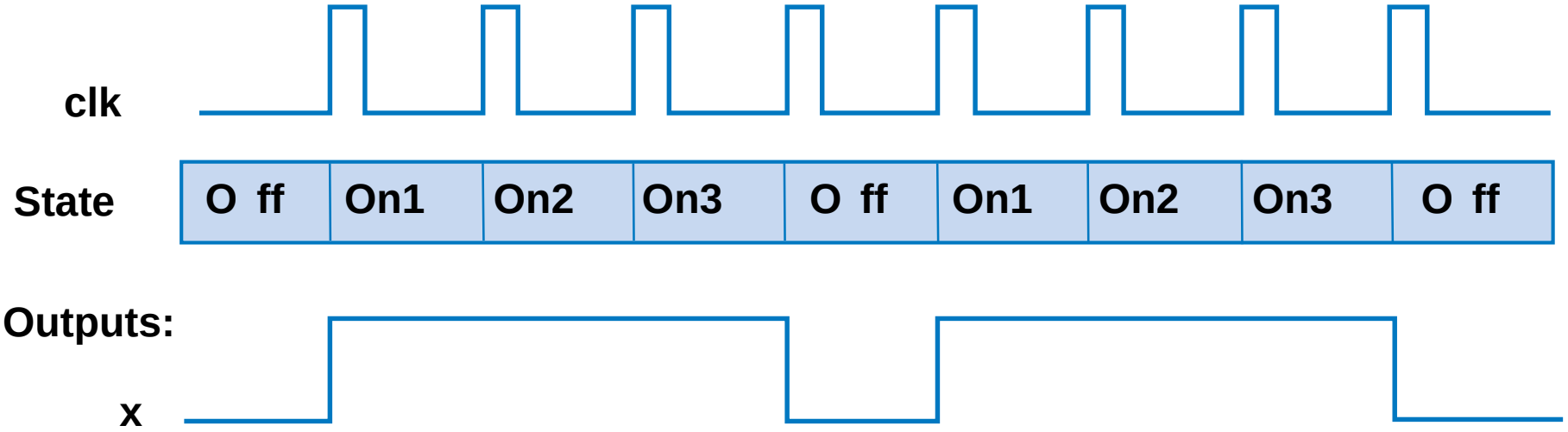
I am On in First CLK and When B=1

I am Off Even if B=1

Step 1: Design FSM

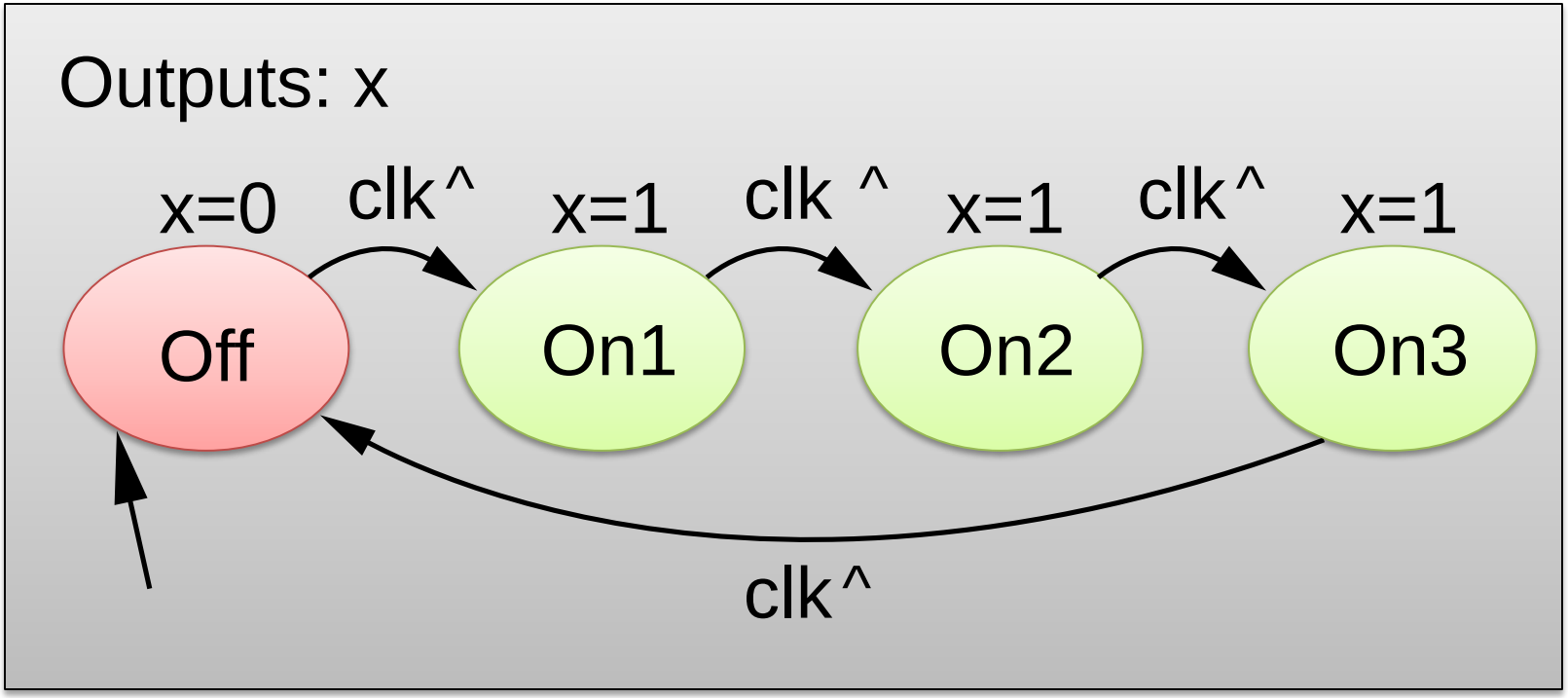
FSM Example 3: 1: 0,1,1,1,repeat

- 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 3: : 0,1,1,1,repeat

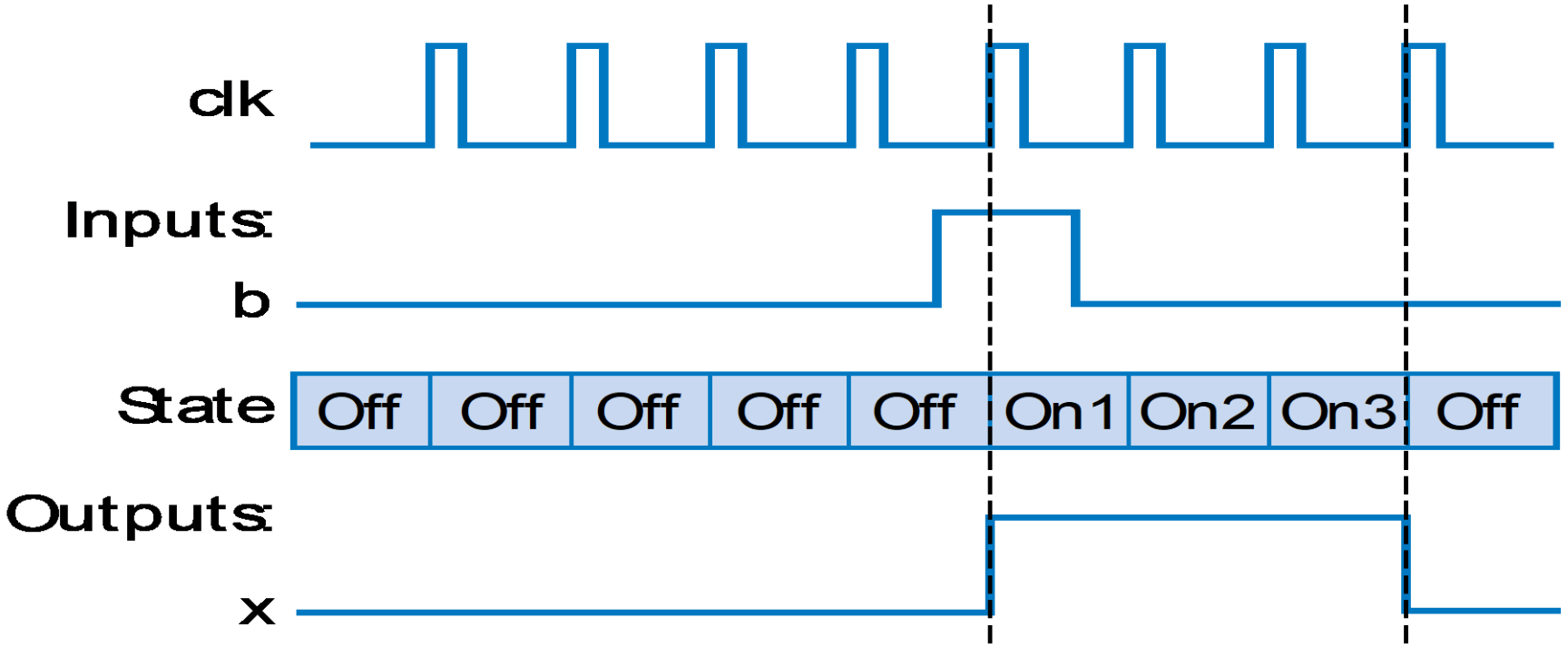
- Want 0, 1, 1, 1, 0, 1, 1, 1, ...
  - Each value for one clock cycle
- Can describe as FSM: Four states, Transition on rising clock edge to next state



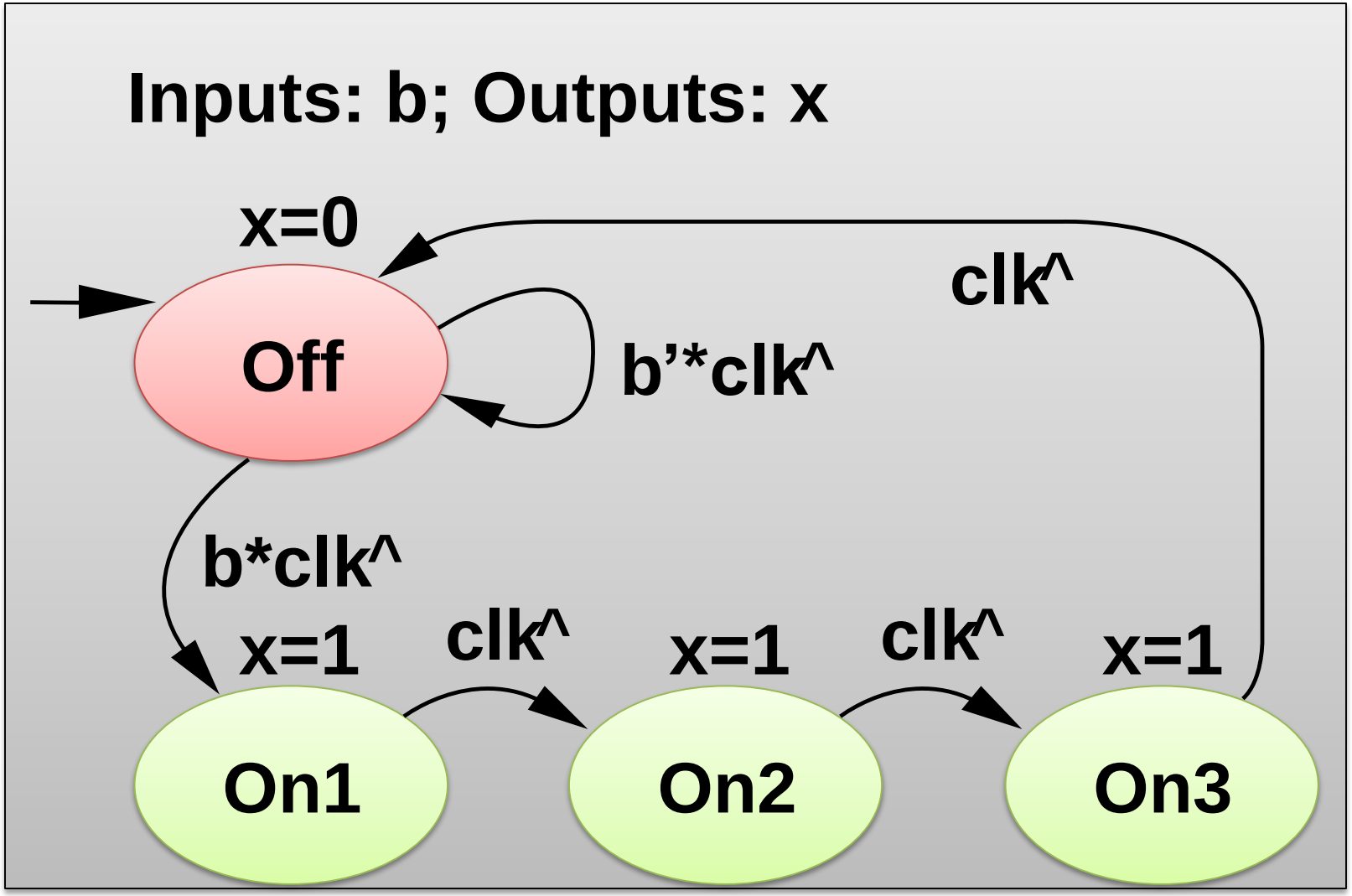
FSM Example 4: Three-Cycles High Laser Timer

- 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

FSM Example 4 : Three-Cycles High Laser Timer



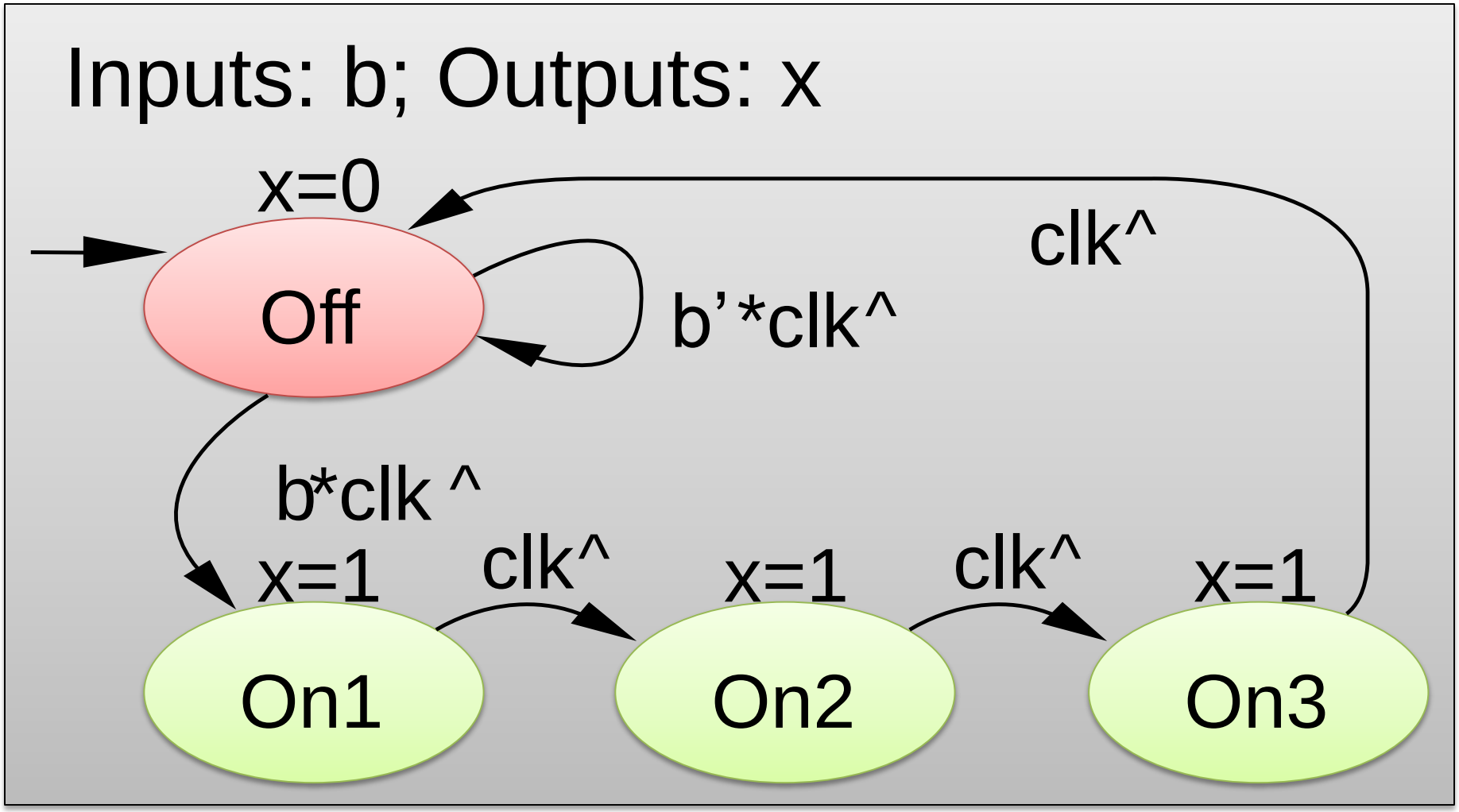
FSM Example 4 : Three-Cycles High Laser Timer



FSM Example 4 : Three-Cycles High Laser Timer

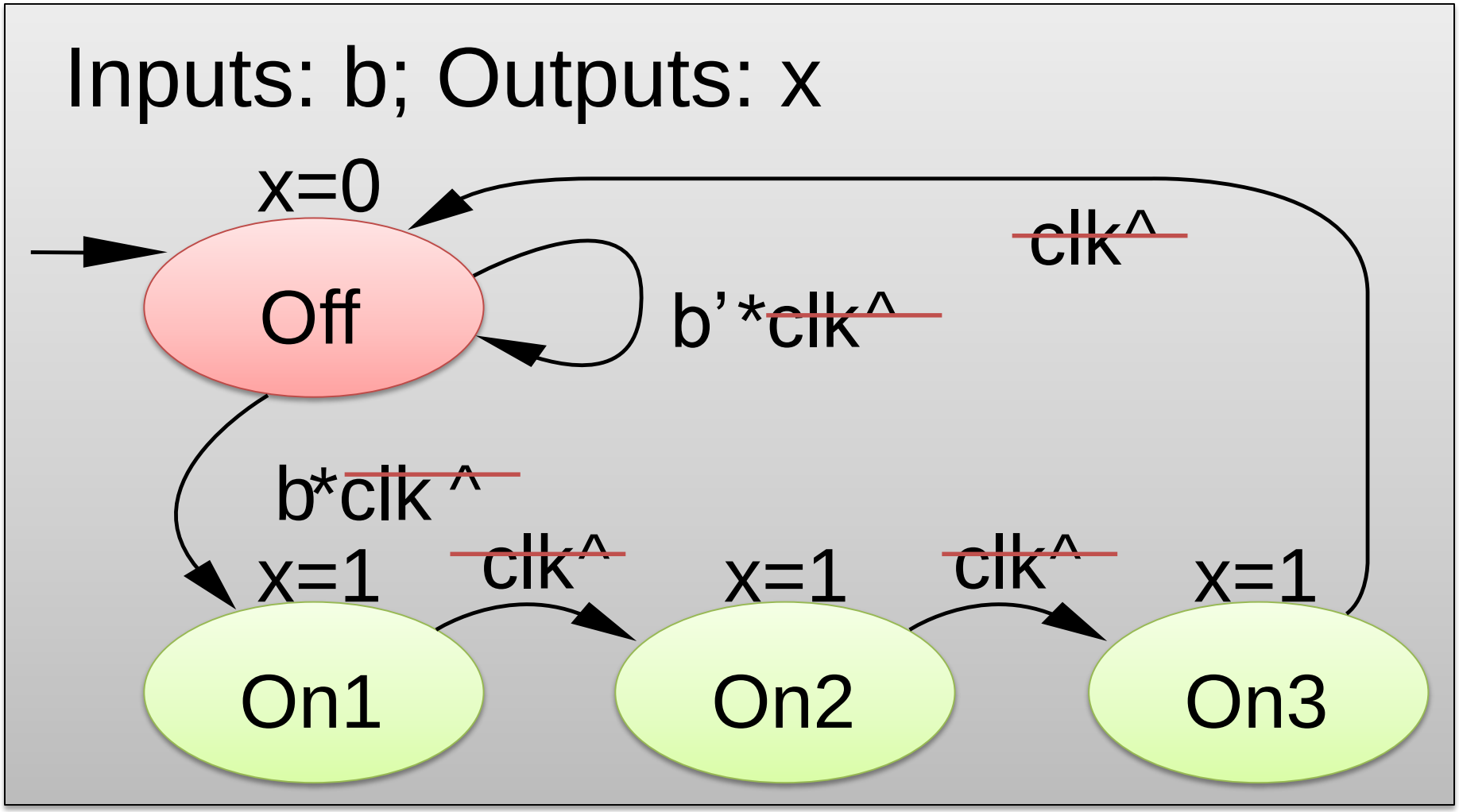
- Showing rising clock on every transition: cluttered
- Make implicit -- assume every edge has rising clock
- What if we wanted a transition *without* a rising edge
  - Asynchronous FSMs -- less common, and advanced topic
  - We consider ***synchronous*** FSMs
  - ***All*** **transition on rising edge**

FSM Simplification: Rising Clock Edges Implicit



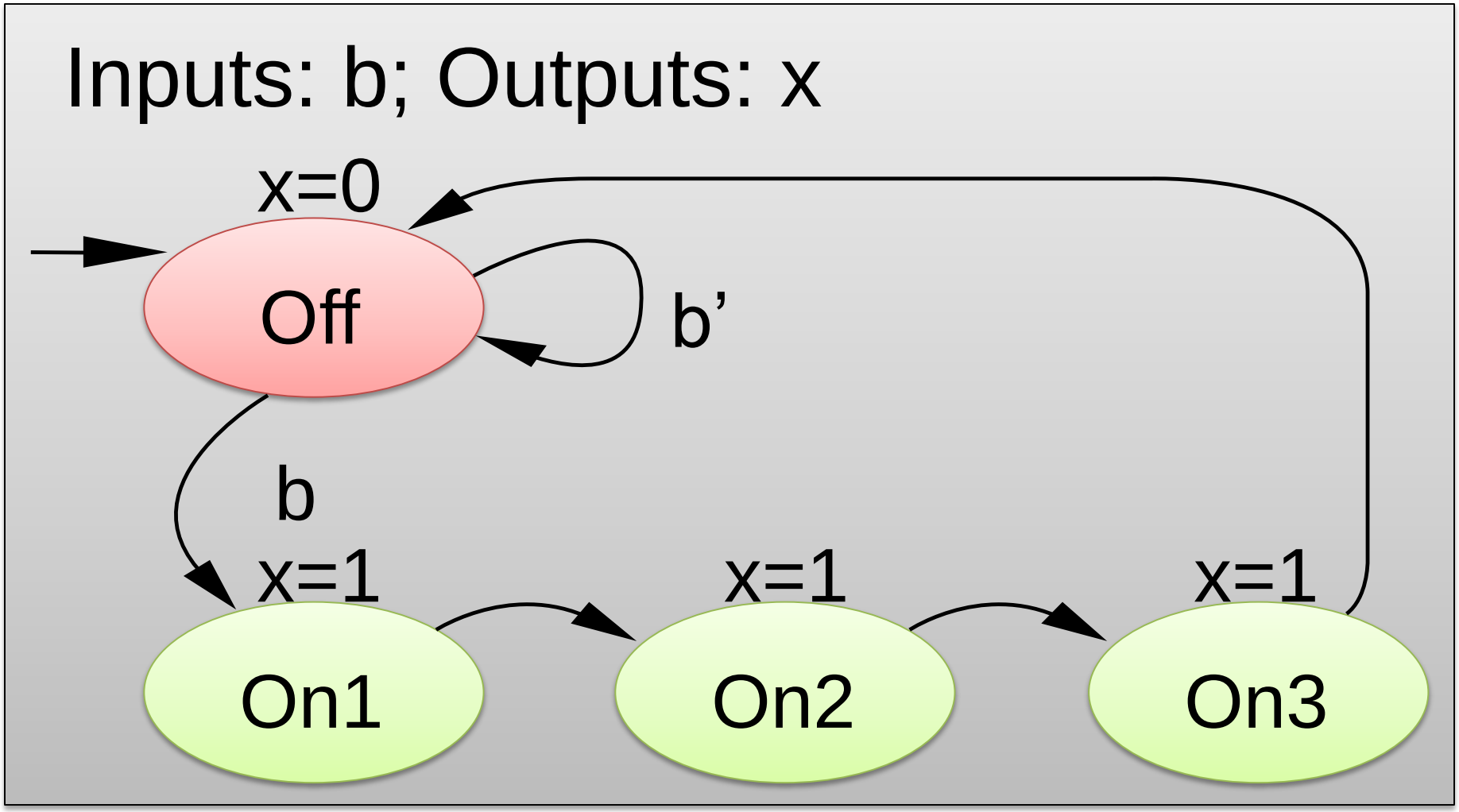
Note: Transition with no associated condition thus transistions to next state on next clock cycle

FSM Simplification: Rising Clock Edges Implicit



Note: Transition with no associated condition thus transistions to next state on next clock cycle

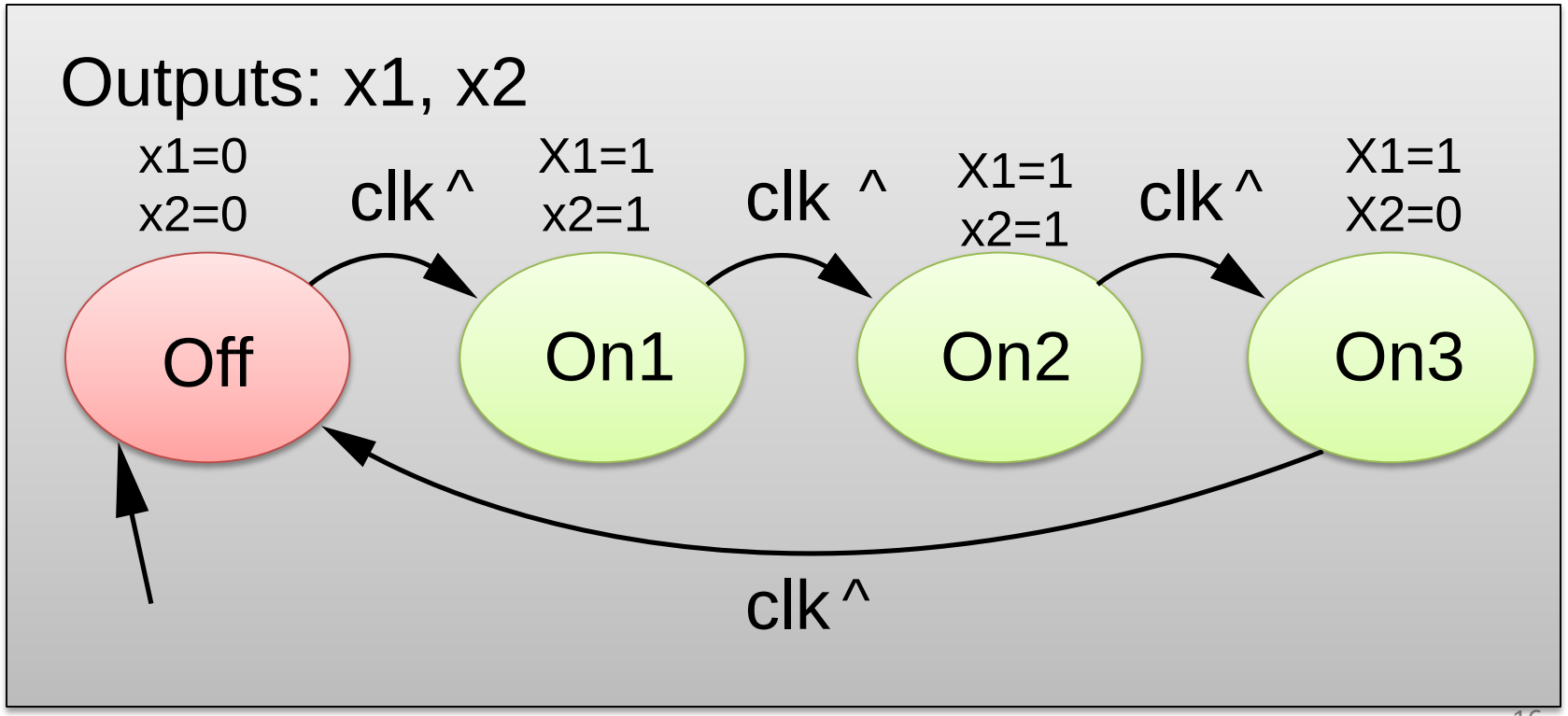
FSM Simplification: Rising Clock Edges Implicit



Note: Transition with no associated condition thus transistions to next state on next clock cycle

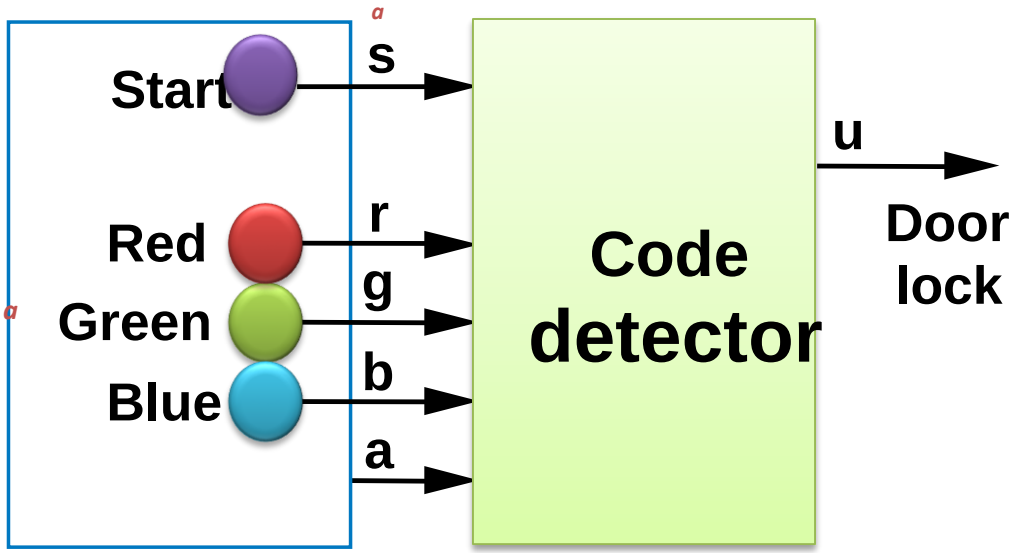
FSM Example 4A: Sequence generator

- Generate two output sequence
  - X1= 01110111….repeat
  - X2=01100110…repeat



FSM Example 5 : Code Detector

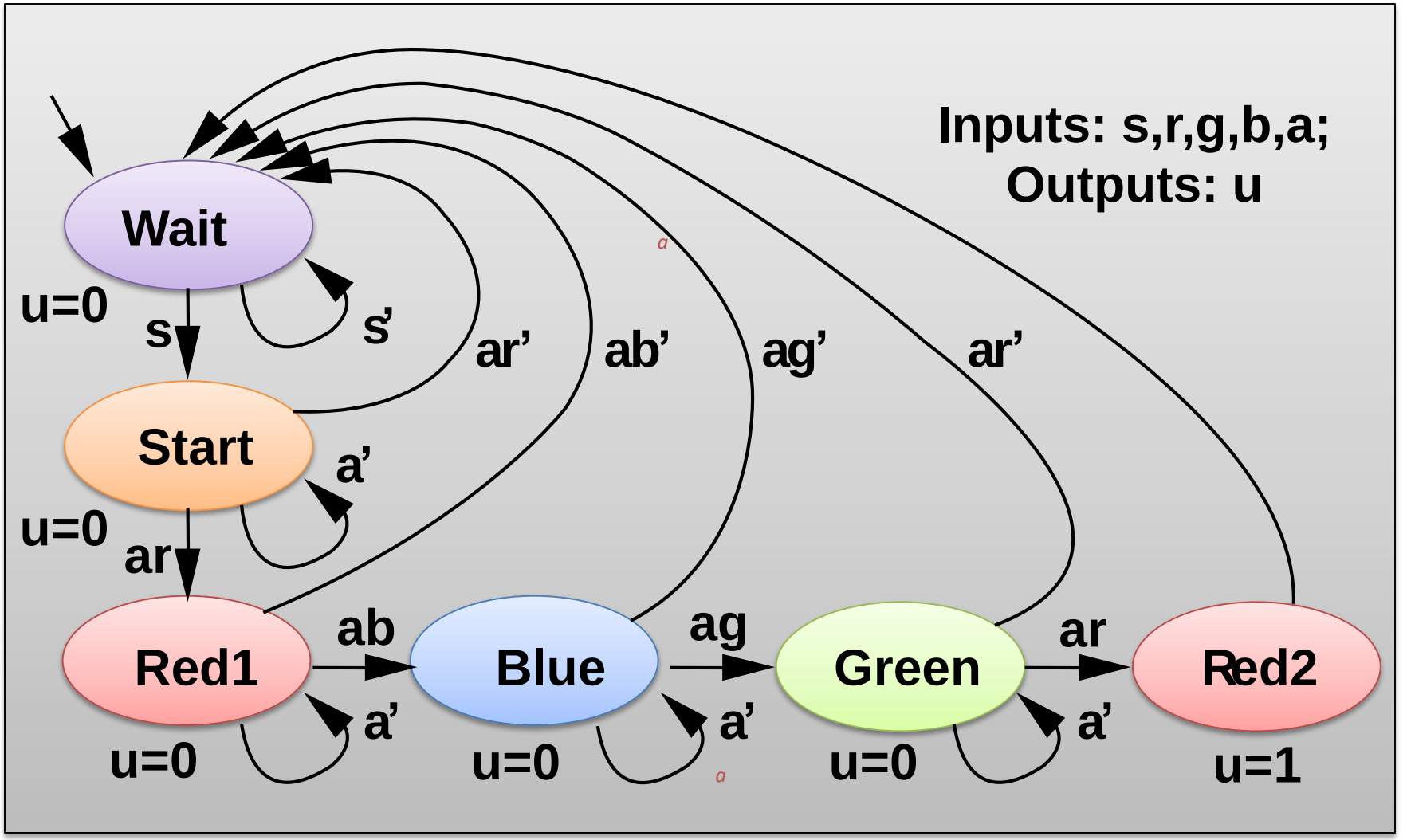
- Unlock door (u=1) only when buttons pressed in sequence:
  - **start, then red, blue, green, red**
- Input from each button: *s, r, g, b*
  - Also, output *a* indicates that some colored button pressed



FSM Example 5: Code Detector

- Wait for start (s=1) in “Wait”,
- **Once started (“Start”)**
  - If see red, go to “Red1”
  - Then, if see blue, go to “Blue”, Then, if see green, go to “Green”, Then, if see red, go to “Red2”
  - In that state, open the door (u=1)
  - Wrong button at any step, return to “Wait”

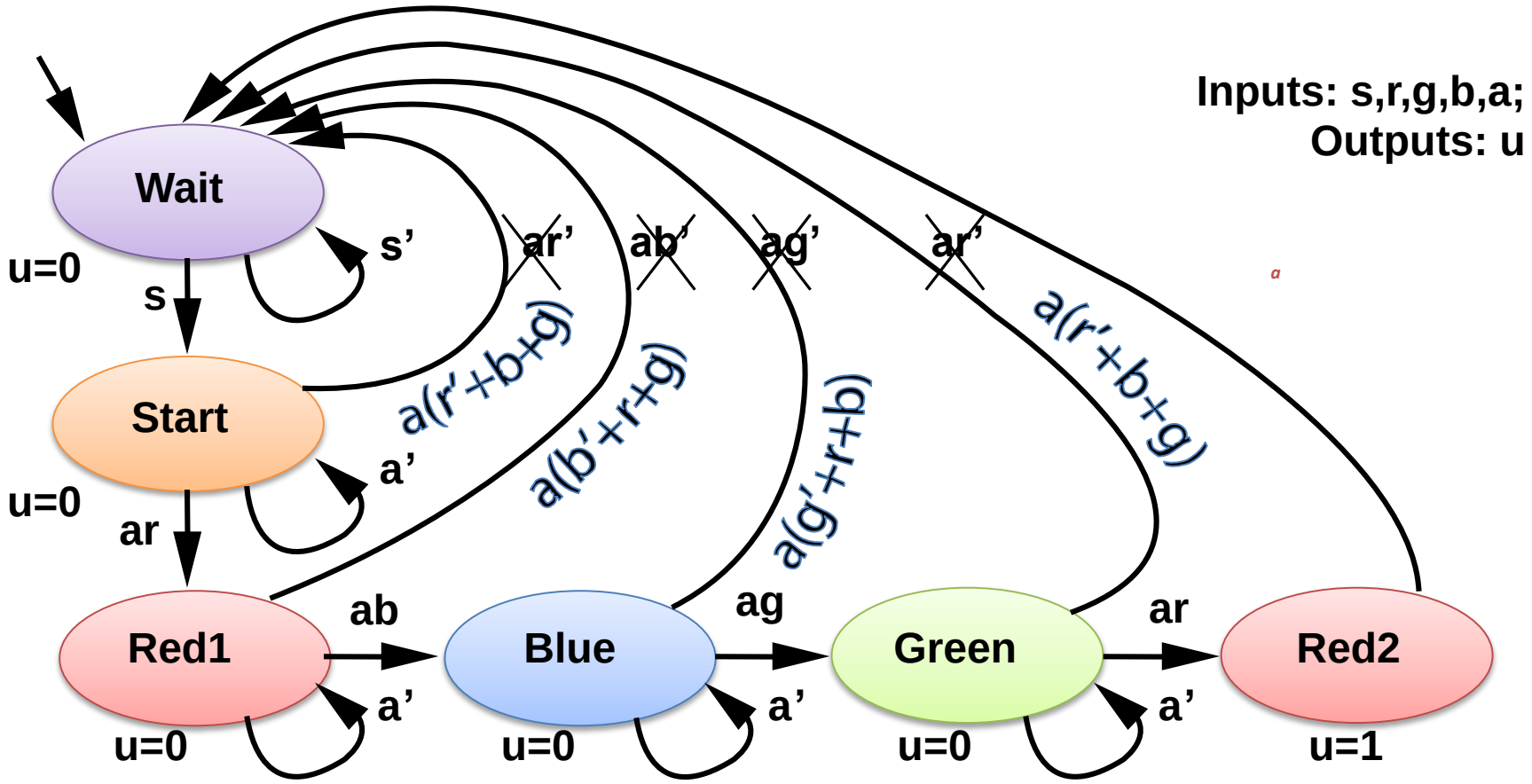
FSM Example 5 : Code Detector



Q: Can you trick this FSM to open the door, without knowing the code?

A: Yes, hold all buttons simultaneously

Improve FSM for Code Detector



- New transition conditions detect if wrong button pressed, returns to "Wait"
- FSM provides formal, concrete means to accurately define desired behavior