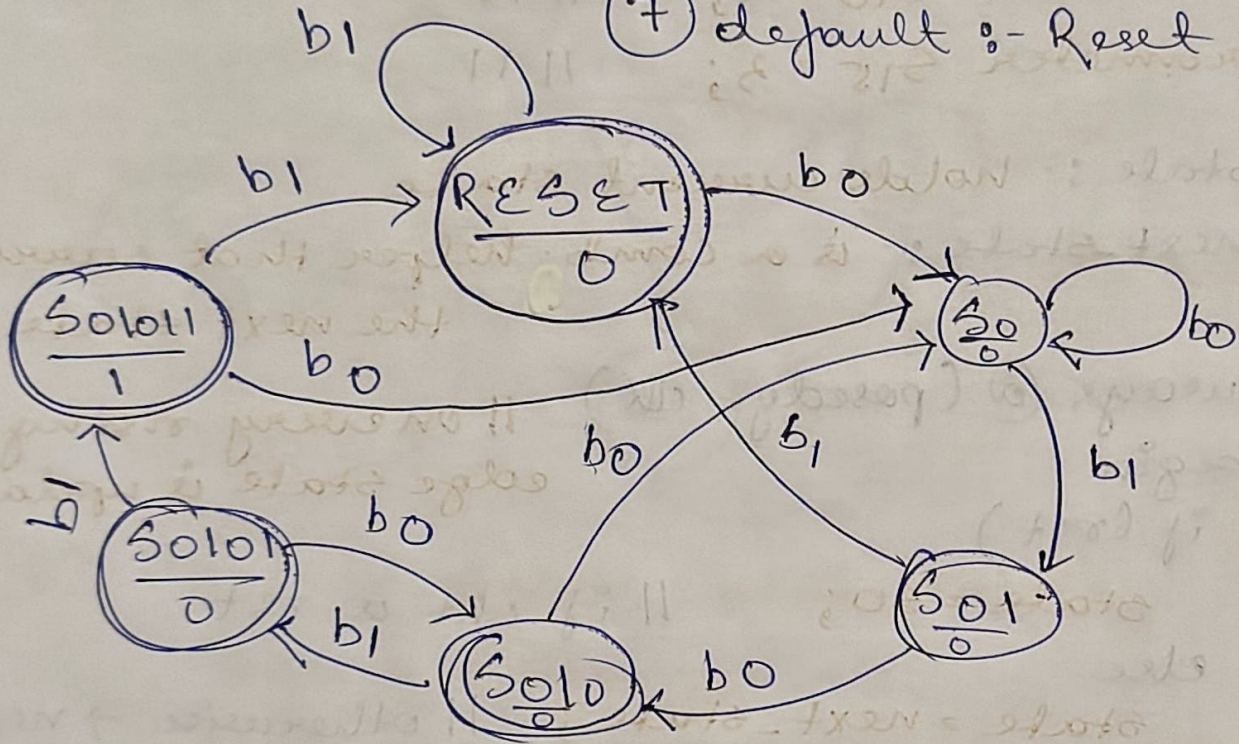


# Digital Lock :- (01011 $\rightarrow$ sequence detector)

- let's say 01011 is the combination to unlock a digital lock.
- using finite state machine

State Assign :- (1) Reset (00) (2) S0 (001)  
b<sub>0</sub>/b<sub>1</sub>  $\rightarrow$  user button (3) S01 (010) (4) S010 (011)  
(5) S0101 (100) (6) S01011 (101)  
(7) default :- Reset



- i/p's and signals

(1) clk

(2) rst - synch / Asynch ctrl

(3) b<sub>0</sub>, b<sub>1</sub> - 2 mutually exclusive buttons that encode incoming bit  
b<sub>0</sub> = logical 0, b<sub>1</sub> = logical 1

(4) unlock - 1 when sequence 01011, otherwise 0

- Next state logic :- on each clk the FSM

samples  $b_0/b_1$ .

⇒ Allows overlapping patterns to be recognised

- At power-up or after rst, the machine is in S-RESET and unlock = 0

- Verilog code

```
module Lock (input clk, rst, b0, b1, output unlock);
```

```
parameter S-RESET = 0;
```

```
parameter S-0 = 1;
```

```
parameter S-01 = 2;
```

```
" S-010 = 3;
```

```
" S-0101 = 4;
```

```
" S-01011 = 5;
```

```
reg [2:0] state, next-state;
```

⇒ State transitions (using Ternary operator)

always @(\*) begin

if (rst)

next-state = S-RESET

else

case (state)

S-RESET :

next-state =  $b_0 ? S-0 : b_1 ? S-RESET : state$ ;

S-0 :

next-state =  $b_0 ? S-0 : b_1 ? S-01 : state$ ;

\* Remember ( $next-state = b_0 ? S-0$ ) ⇒ if  $b_0 ⇒ S-0$

( $b_0 ? S-0 : b_1 ? S-RESET$ ) ⇒ if  $b_0 ⇒ S-0$

( $S-RESET : state$ ) ⇒ stay in state else  $b_1 ⇒ S-RESET$

⇒ always @ (posedge clk)

state <= next\_state;

assign unlock = state == 5'b01011 ? 1:0

{ // (if (state == 5'b01011))

\*

unlock = 1

else

unlock = 0

}

endmodule

## • Testbench

module digital\_lock\_tb;

reg clk, b0, b1, rst;

wire unlock;

integer i

reg [4:0] data

always #10 clk = ~clk

lock uut (clk, rst, b0, b1, unlock);

initial begin

clk = 0; rst = 1; // initialize

#50 reset = 0; // clk is on

data = 5'b01011; i = 0;

#130 \$finish

end

always @ (posedge clk) begin

b0 = data[5-i] ? 0:1;

b1 = data[5-i] ? 1:0;

end

endmodule

\* index i ↑ with each clk

if selected data = 0, ⇒ b0 = 1, b1 = 0

if selected data = 1, ⇒ b1 = 0, b0 = 0