

# Realization of Sequential Circuits

## Design using FSM(Finite State Machine):

### Design Specification

A simple digital circuit is to be designed for the coin acceptor of an electronic newspaper vending machine.

- Assume that the newspaper cost 15 cents.
- The coin acceptor takes only nickels and dimes (1nickel = 5 cent and 1dime = 10 cent).
- Exact change must be provided. The acceptor does not return extra money.
- Valid combinations including order of coins are
  - One nickel and one dime
  - Three nickels
  - One dime and one nickel
  - Two dimes are valid, but the acceptor does not return money

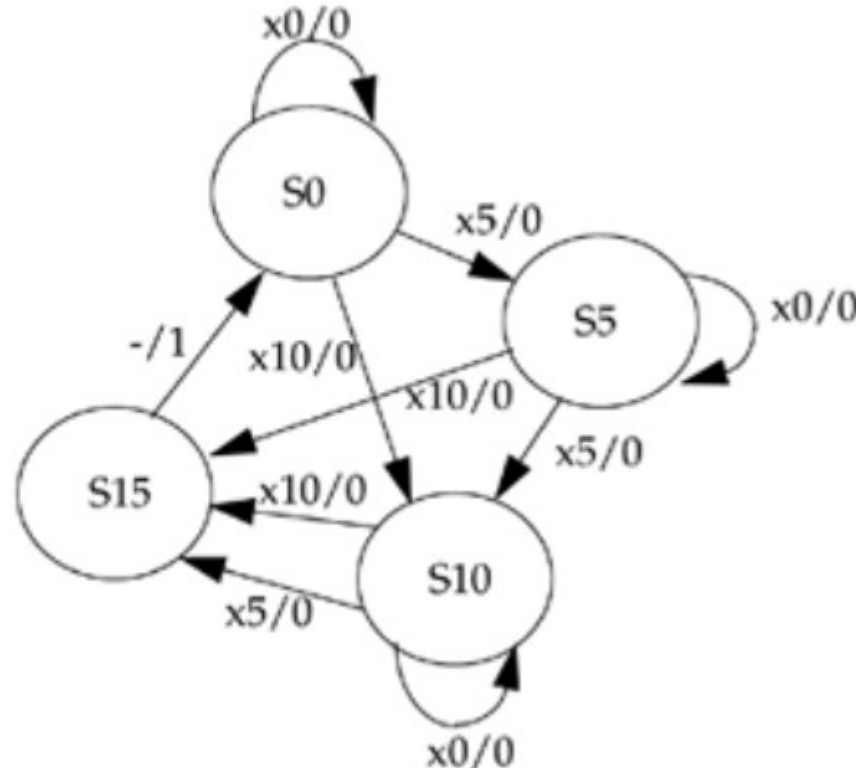
## Example FSM newspaper vending machine

- input: 2-bit, coin[1:0]?no coin x0= 2'b00, nickel x5 = 2'b01, dime x10 = 2'b10.
- output: 1-bit, newspaper?release door when newspaper = 1'b1
- states: 4 states?s0 = 0 cents; s5 = 5 cents; s10 = 10 cents; s15 = 15 cents

State	Money
S0	0 cents
S5	5 cents
S10	10 cents
S15	15 cents

Input	coin[1:0]
x0	2'b00
x5	2'b01
x10	2'b10
-	don't care



# Example FSM newspaper vending machine – Verilog code

//Design the newspaper vending machine coin acceptor  
//using a FSM approach

```
module vend( coin, clock, reset, newspaper);
```

//Input output port declarations

```
input [1:0] coin;
```

```
input clock;
```

```
input reset;
```

```
output newspaper;
```

```
wire newspaper;
```

//internal FSM state declarations

```
wire [1:0] NEXT_STATE;
```

```
reg [1:0] PRES_STATE;
```

//state encodings

```
parameter s0 = 2'b00;
```

```
parameter s5 = 2'b01;
```

```
parameter s10 = 2'b10;
```

```
parameter s15 = 2'b11;
```

//Combinational logic

```
function [2:0] fsm;
```

```
input [1:0] fsm_coin;
```

```
input [1:0] fsm_PRES_STATE;
```

```
reg fsm_newspaper;
```

```
reg [1:0] fsm_NEXT_STATE;
```

```
begin
```

```
case (fsm_PRES_STATE)
```

```
s0: //state = s0
```

```
begin
```

```
if (fsm_coin == 2'b10)
```

```
begin
```

```
fsm_newspaper = 1'b0;
```

```
fsm_NEXT_STATE = s10;
```

```
end
```

```
else if (fsm_coin == 2'b01)
```

```
begin
```

```
fsm_newspaper = 1'b0;
```

```
fsm_NEXT_STATE = s5;
```

```
end
```

```
else
```

```
begin
```

```
fsm_newspaper = 1'b0;
```

```
fsm_NEXT_STATE = s0;
```

```
end
```

```
end
```

```
s5: //state = s5
```

```
begin
```

```
if (fsm_coin == 2'b10)
```

```
begin
```

```
fsm_newspaper = 1'b0;
```

```
fsm_NEXT_STATE = s15;
```

```
end
```

```
else if (fsm_coin == 2'b01)
```

```
begin
```

```
fsm_newspaper = 1'b0;
```

```
fsm_NEXT_STATE = s10;
```

```
end
```

## Example FSM newspaper vending machine – Verilog code

```
else
begin
fsm_newspaper = 1'b0;
fsm_NEXT_STATE = s5;
end
s10: //state = s10
begin
if (fsm_coin == 2'b10)
begin
fsm_newspaper = 1'b0;
fsm_NEXT_STATE = s15;
end
else if (fsm_coin == 2'b01)
begin
fsm_newspaper = 1'b0;
fsm_NEXT_STATE = s15;
end
else
```

```
begin
fsm_newspaper = 1'b0;
fsm_NEXT_STATE = s10;
end
end
s15: //state = s15
begin
fsm_newspaper = 1'b1;
fsm_NEXT_STATE = s0;
end
endcase
fsm = {fsm_newspaper, fsm_NEXT_STATE};
end
endfunction
//Reevaluate combinational logic each time a coin
//is put or the present state changes
assign {newspaper, NEXT_STATE} = fsm(coin,
PRES_STATE);
```

```
//clock the state flip-flops.
//use synchronous reset
always @(posedge clock)
begin
if (reset == 1'b1)
PRES_STATE <= s0;
else
PRES_STATE <= NEXT_STATE;
end
endmodule
```

# Stimulus for Newspaper Vending Machine FSM

```

module stimulus;
reg clock;
reg [1:0] coin;
reg reset;
wire newspaper;
//instantiate the vending state machine
vend vendY (coin, clock, reset, newspaper);
//Display the output
initial
begin
$display("\t\tTime Reset Newspaper\n");
$monitor("%d %d %d", $time, reset, newspaper);
end
//Apply stimulus to the vending machine
initial
begin
clock = 0;
coin = 0;
reset = 1;
#50 reset = 0;
@(negedge clock); //wait until negative edge of clock

```

```

//Put 3 nickels to get newspaper
#80 coin = 1; #40 coin = 0;
#80 coin = 1; #40 coin = 0;
#80 coin = 1; #40 coin = 0;
//Put one nickel and then one dime to get newspaper
#180 coin = 1; #40 coin = 0;
#80 coin = 2; #40 coin = 0;
//Put two dimes;
#180 coin = 2; #40 coin = 0;
#80 coin = 2; #40 coin = 0;
//Put one dime and then one nickel to get newspaper
#180 coin = 2; #40 coin = 0;
#80 coin = 1; #40 coin = 0;
#80 $finish;
end
//setup clock; cycle time = 40 units
always
begin
#20 clock = ~clock;
end
endmodule

```

# Output of Newspaper Vending Machine FSM

Time	Reset	Newspaper
0	1	x
20	1	0
50	0	0
420	0	1
460	0	0
780	0	1
820	0	0
1100	0	1
1140	0	0
1460	0	1
1500	0	0