MAJOR PROJECT DOCUMENT BY K.NIKITHA ON DESIGNING AND VERIFYING A STATE MACHINE

Design and Verify the a state machine for the coin-operated electronic newspaper vending machine and draw its ASM chart. Assume the newspaper costs 15 cents, The machine accepts only nickels(N) and dimes(D) and exact change must be provided, The machine does not return the extra money. Valid combinations including order of coins are one nickel and one dime, three nickels, or one dime and one nickel. Two dimes are valid but the machine does not return money. Write its HDL and verify using a test bench.

- When each coin is inserted, a 2-bit signal coin [1:0] is sent to the digital circuit. The signal is asserted at the next negative edge of a global clock signal and stays up for exactly 1 clock cycle.
- The output of the digital circuit is a single bit. Each time the total amount inserted is 15 cents or more, an output signal newspaper goes high for exactly one exactly one clock cycle and vending machine door is released.
- A reset signal can be used to reset the finite state machine. We assume synchronous reset .

- Input

  2 bit, coin [1:0]-no coin x0=2’b00
  Nickle x5=2’bo1,
  Dime x10=2’b10

- Output

  1 bit, newspaper-release=1’b1

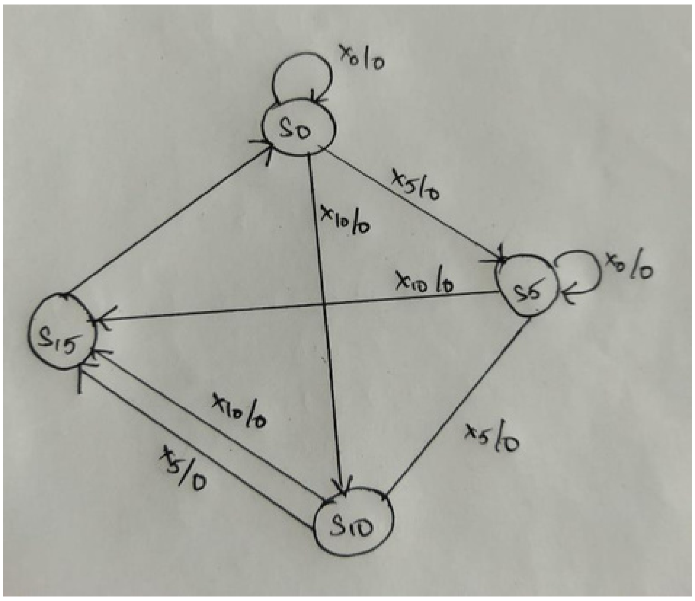
STATES

S0-0 cents
S5-5 cents
S10-10 cents
S15-15 cents

❖ STATE DIAGRAM

STATE	INPUT	NEXT STATE	OUTPUT
S0	X0 X5 X10	S0 S5 S10	0 0 0
S5	X0 X5 X10	S5 S10 S15	0 0 0
S10	X0 X5 X10	S10 S15 S15	0 0 0
S15	---	S0	1

❖ ASM CHART



❖ PROGRAM

```
Module void (coin, clock, reset, newspaper);
Input [1:0] coin;
Input clock;
Input reset;
Output newspaper;
Wire newspaper;
Wire [1:0] next_state
reg [1:0] press_state

parameter s0 =2’b00;
paremeter s5 =2’b01;
parameter s10 =2’b10;
parameter s15 =2’b11;
function [2:0] fsm;
input [1:0] fsm_coin;
input [1:0] fsm _pres_state;
begin
case(fsm_pres_state)
s:0
begin
if (fsm_coin ==2’b10)
begin
fsm_newspaper=1’b0;
fsm_next_state=s10;
end
else if (fsm_com==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:
begin
if (fsm_coin==2’b10)
begin
fsm_newspaper=1’b10:
fsm_NEXT_state=s15;
end
else if (fsm_coin=2’bo1)
begin
fsm_newspaper=1’b0;
fsm_NEXT_state=s10;
end
else
begin
fsm_newspaper=1’bo;
fsm_NEXT_state=s10;
end
else
begin
fsm_newspaper=1’bo;
fsm_NEXT_state=s5;
end
s10:
begin
```

```
if (fsm_coin=2’b10)
begin
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;
begin
fsm_newspaper=1’b1;
fsm_next_state=s0;
end
end case
fsm={fsm_newspaper,fsm_next_state};
end
end function
assign {newspaper,nextstate}=fsm(coin,pres,state);
always@(posedge clock)
begin
if(reset=1’b1)
pres_state <=50;
else
pres_state<=next_state;
```

```
end
endmodule
module stimules
reg clock;
reg [1:0] coin;
reg reset;
wire newspaper;
vend vends (coin,clock,reset,newspaper);
intial
begin
&display (“\t\t time reset news paper\n”);
&monitor(“%d%d%d”,& time,reset,newspaper);
End
Intial
Begin
Clock=0
Coin=0
Reset=1
# 50 reset=0;
@(negedage clock);
// one nickle and one dime
# 180 coin=1; # 40 coin=0;
# 80 coin=2; # 40 coin=0;
// 3nickles
# 80 coin=1; # 40 coin=0;
# 80 coin=1; # 40 coin=0;
# 80 coin=1; # 40 coin=0;
// 2 dimes,machine does not return a nickle to get newspaper
# 180 coin=2; # 40 coin=0;
# 80 coin=1; # 40 coin=0;
end
```

```
always

begin

# 20 clock =~clock

End

endmodule
```