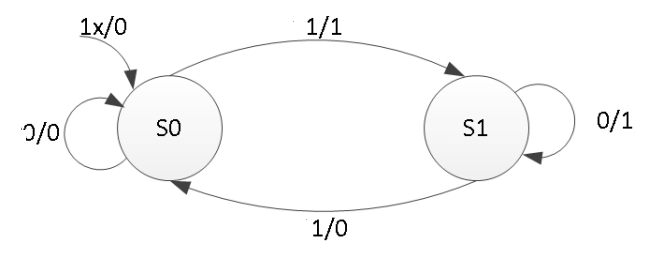
VERILOG CODE FOR A STATE MACHINE

1. Verilog code for MEALY FSM



module mealy\_machine (input clk, input reset, input x, output reg Z);

reg state, nextstate;

parameter S0=0, S1=1;

always @(posedge clk or posedge reset) // always block to update state

if (reset)

state <= S0;

else

state <= nextstate;

always @(state or x) // always block to compute both output & nextstate

begin

case(state)

S0:

if(x)

begin

Z = 1;

nextstate = S1;

end

else

begin

Z = 0;

nextstate = S0;

end

S1:

if(x)

begin

Z = 0;

nextstate = S0;

end

else

begin

Z= 1;

nextstate = S1;

end

default:

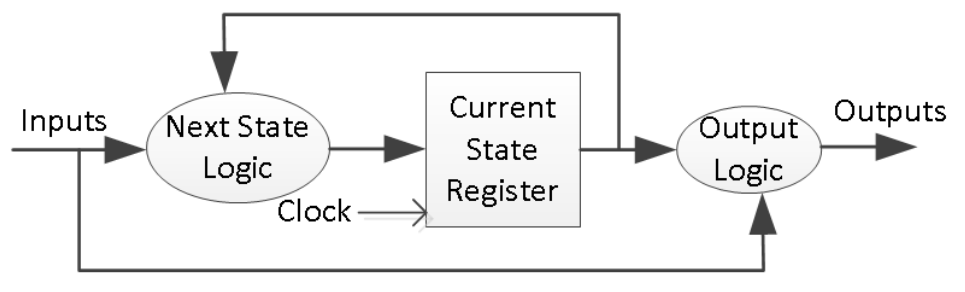
nextstate = S0;

endcase

end

endmodule

or, using 3 blocks



module mealy\_machine (input clk, input reset, input x, output reg Z );

reg state, nextstate;

parameter S0=0, S1=1;

always @(posedge clk or posedge reset) // always block to update state

if (reset)

state <= S0;

else

state <= nextstate;

always @(state or x) // always block to compute output

begin

z = 1'b0;

case(state)

S0: if(x)

Z = 1;

S1: if(!x)

Z = 1;

endcase

end

always @(state or x) // always block to compute nextstate

begin

nextstate = S0;

case(state)

S0: if(x)

nextstate = S1;

S1: if(!x)

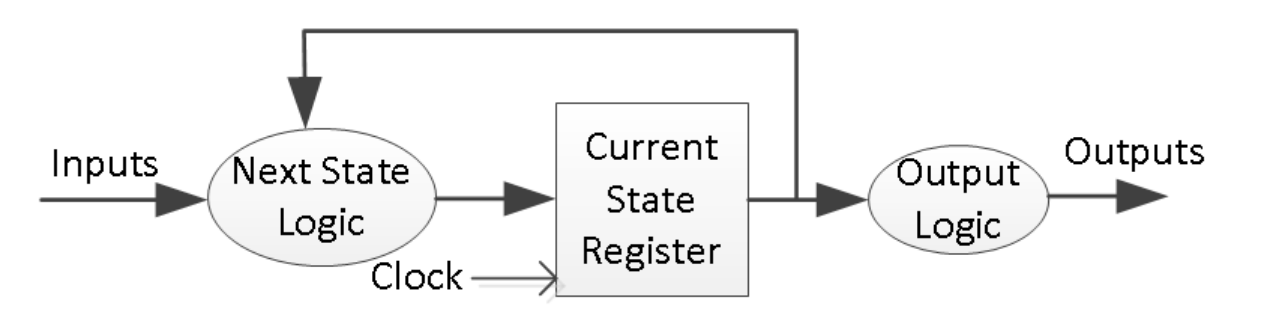
nextstate = S1;

endcase

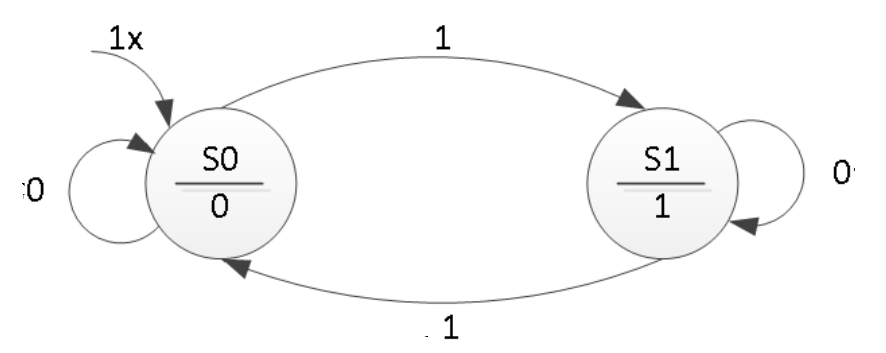
end

endmodule

1. Verilog code for MOORE FSM



a)

****

module moore\_machine (input clk, input reset, input x, output reg Z);

reg state, nextstate;

parameter S0=0, S1=1;

always @(posedge clk or posedge reset) // always block to update state

if (reset)

state <= S0;

else

state <= nextstate;

always @(state) // always block to compute output

begin

case(state)

S0: Z = 0;

S1: Z = 1;

endcase

end

always @(state or x) // always block to compute nextstate

begin

nextstate = S0;

case(state)

S0: if(x)

nextstate = S1;

S1: if(!x)

nextstate = S1;

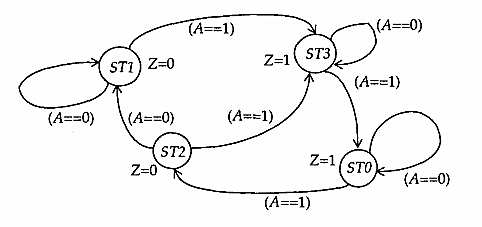
endcase

end

endmodule

B)

|  |  |  |  |
| --- | --- | --- | --- |
| P.S | N.S | | OUTPUT |
| A=0 | A=1 |
| ST0 | ST0 | ST2 | 1 |
| ST1 | ST1 | ST3 | 0 |
| ST2 | ST1 | ST3 | 0 |
| ST3 | ST3 | ST0 | 1 |



State diagram of a Moore machine

moduleMoore\_FSM (A, Clock, Z);

input A, Clock;

output Z;

reg Z;

parameter STO = 0, STl = 1, ST2 = 2, ST3 = 3 ;

reg [0:1] Moore\_State;

always @ (negedgeClock)

case (Moore\_State)

STO:

begin

Z = 1;

if (A)

Moore\_State = ST2;

end

ST1 :

begin

Z = 0;

if (A)

Moore\_State = ST3;

end

ST2:

begin

Z = 0;

if (~A)

Moore\_State = ST1;

else

Moore\_State= ST3;

end

ST3:

begin

Z = 1;

if (A)

Moore\_State = STO;

end

endcase

endmodule