SACRAMENTO STATE

**Name: Karthik Shekarappa**
**Email: karthikms@csus.edu**
**Instructor: Dr. Behnam Arad**
**Course: EEE 273-Hierarchical Digital Design Methodology**
**Assignment Number: 4**
**Date Submitted: 4th March 2019**

```
module seq_detect(input x,clk,rst, output reg Z);
parameter[1:0] idle = 2'b00, s0 = 2'b01, s1 = 2'b10;
reg[1:0] current_state, next_state;
always @(posedge clk, negedge rst)
begin
      if(!rst)   current_state <= idle;
      else       current_state <= next_state;
end
always @(*)
begin
       next_state = idle ; Z = 1'b0;
case(current_state)
      idle:
            if(x) begin      next_state = s0; Z = 1'b0;   end
            else begin  next_state = idle; Z = 1'b0; end
      s0:
            if(!x) begin           next_state = s1; Z = 1'b0;   end
            else begin  next_state = s0; Z = 1'b0;   end
      s1:
            if(x) begin      next_state = s0; Z = 1'b1;   end
            else begin  next_state = idle; Z = 1'b0; end

      default:   begin  next_state = idle; Z = 1'b0;     end

endcase
end
endmodule
```

```
`include "seqdetect.v"
module seq_detect_fixture;
reg x,clk,rst;
wire Z;
integer i;
seq_detect u1(.x(x), .clk(clk), .rst(rst), .Z(Z));
initial
      $monitor($time,"   X = %b     Z = %b     reset = %b\n",x,Z,rst);
initial
      $vcdpluson;
initial
            begin x = 1'b0;rst = 1'b0; end
initial
begin
for(i=0;i<2;i=i+1)
      begin
      reset(0);
      seq_in(0,0,0);
      reset(1);
      seq_in(0,0,0);
      seq_in(0,0,1);
      seq_in(0,1,0);
      seq_in(1,0,1);
      seq_in(1,1,0);
      reset(0);
      reset(1);
      seq_in(1,0,1);
      reset(0);
      seq_in(1,0,1);
      reset(1);
      seq_in(0,0,1);
      seq_in(0,1,0);
      reset(0);
      reset(1);
      seq_in(1,0,1);
      seq_in(0,0,1);
      seq_in(0,1,0);
      end
end
task seq_in(input in1,in2,in3);
begin
      @(posedge clk)x = in1;
      @(posedge clk)x = in2;
      @(posedge clk)x = in3;
end
endtask
task reset(input r);
begin
      if (r)  #10 rst = 1;
      else  #10 rst = 0;
end
endtask
initial  begin clk = 1'b0;
forever #10 clk=~clk;
```

```
end
initial #1000 $finish;
endmodule
```

```
Chronologic VCS simulator copyright 1991-2014
Contains Synopsys proprietary information.
Compiler version I-2014.03-2; Runtime version I-2014.03-2;  Mar  4 19:04
2019
VCD+ Writer I-2014.03-2 Copyright (c) 1991-2014 by Synopsys Inc.
                   0   X = 0     Z = 0   reset = 0

                  60   X = 0     Z = 0   reset = 1

                 170   X = 1     Z = 0   reset = 1

                 190   X = 0     Z = 0   reset = 1

                 210   X = 1     Z = 1   reset = 1

                 230   X = 0     Z = 0   reset = 1

                 250   X = 1     Z = 1   reset = 1

                 270   X = 0     Z = 0   reset = 1

                 290   X = 1     Z = 1   reset = 1

                 310   X = 1     Z = 0   reset = 1

                 350   X = 0     Z = 0   reset = 1

                 360   X = 0     Z = 0   reset = 0

                 370   X = 0     Z = 0   reset = 1

                 390   X = 1     Z = 0   reset = 1

                 410   X = 0     Z = 0   reset = 1

                 430   X = 1     Z = 1   reset = 1

                 440   X = 1     Z = 0   reset = 0

                 470   X = 0     Z = 0   reset = 0

                 490   X = 1     Z = 0   reset = 0

                 500   X = 1     Z = 0   reset = 1

                 510   X = 0     Z = 0   reset = 1

                 550   X = 1     Z = 0   reset = 1

                 570   X = 0     Z = 0   reset = 1

                 590   X = 1     Z = 1   reset = 1

                 610   X = 0     Z = 0   reset = 1
```

```
                620   X = 0     Z = 0   reset = 0

                630   X = 0     Z = 0   reset = 1

                650   X = 1     Z = 0   reset = 1

                670   X = 0     Z = 0   reset = 1

                690   X = 1     Z = 1   reset = 1

                710   X = 0     Z = 0   reset = 1

                750   X = 1     Z = 0   reset = 1

                770   X = 0     Z = 0   reset = 1

                790   X = 1     Z = 1   reset = 1

                810   X = 0     Z = 0   reset = 1

                820   X = 0     Z = 0   reset = 0

                880   X = 0     Z = 0   reset = 1

                990   X = 1     Z = 0   reset = 1

$finish called from file "seqdetect_fixture.v", line 56.
$finish at simulation time                 1000
          V C S   S i m u l a t i o n   R e p o r t
Time: 1000
CPU Time:      0.340 seconds;       Data structure size:   0.0Mb
Mon Mar  4 19:04:12 2019
```