Joseph Thornton

Professor Faroughi

CSC 137 Section 3

Project 3

Sequence Recognizer

1. (A) (**FSD)**

X=1/z=1

X=0/z=0

X=0/z=0

X=1/z=0

Reset

X=0/z=0

X=0/z=0

X=1/z=0

X=1/z=0

**(Block diagram)**

Q0 d0

q(bar) r

Q1 d1

q(bar) r

Clk reset x

OG

NSG

Z

d0

d1

1. **NSG (truth tables)** Current State Next State

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Q1 | Q0 | D1 | D0 |
| 0 | 0 | 0 | **0** | **0** |
| 0 | 0 | 1 | **1** | **0** |
| 0 | 1 | 0 | **1** | **1** |
| 0 | 1 | 1 | **0** | **0** |
| 1 | 0 | 0 | **0** | **1** |
| 1 | 0 | 1 | **0** | **1** |
| 1 | 1 | 0 | **0** | **1** |
| 1 | 1 | 1 | **0** | **1** |

1. **OG** Current State Output

|  |  |  |  |
| --- | --- | --- | --- |
| X | Q1 | Q0 | Z |
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **0** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **0** |
| 1 | 1 | 0 | **0** |
| 1 | 1 | 1 | **1** |

**(Minimization)**

q1q0

1. **Logic expressions (**standard binary number dist = down (0,1) across (00,01,11,10)

x

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 |  | 1 |
| q1q0 |  |  |  |

D1=x̄q̄1q0+x̄q1q̄0

x

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | 1 |
| 1  q1q0 | 1 | 1 | 1 |

D0=x+q1q̄0

x

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  | 1 |  |

Z=xq1q0

//final circuit

D1

Z

Q1 d1

Q̄1 r

D0

Q0 d0

Q̄0 r

Clk reset x

// part 1 code

(B) 1. module meeleseq(

input x,reset,clock,

output z

);

wire [1:0] d;

reg [1:0] q;

assign d[1]=~x&~q[1]&q[0]|~x&q[1]&~q[0];

assign d[0]=x|q[1]&~q[0];

assign z=x&q[0]&q[1];

always@(posedge clock, posedge reset)

begin

if(reset==1)

q=0;

else

q=d;

end

endmodule

// part 2 code

1. module directdesign(

input x,reset,clock,

output reg z

);

parameter A=2'b00,

B=2'b01,

C=2'b10,

D=2'b11;

reg [1:0] currentstate,nextstate;

//NSG

always@\*

begin

casex (currentstate)

A: if(x==1)

nextstate=B;

else

nextstate=A;

B: if(x==1)

nextstate=B;

else

nextstate=C;

C: if(x==1)

nextstate=B;

else

nextstate=D;

D: if(x==1)

nextstate=B;

else

nextstate=A;

endcase

end

//OG

always@\*

begin

if(currentstate==D&x==1)

z=1;

else

z=0;

end

//flip flop same as other one

always@(posedge clock,posedge reset)

begin

if(reset==1)

currentstate<=A;

else

currentstate<=nextstate;

end

endmodule

(D). //output for both part 1 and part 2 are the same

Chronologic VCS simulator copyright 1991-2014

Contains Synopsys proprietary information.

Compiler version I-2014.03-2; Runtime version I-2014.03-2; Apr 19 14:58 2017

0: z=0

10: x=1

20: x=0

30: x=0

40: x=1

40: z=1

45: z=0

50: x=1

60: x=0

70: x=0

80: x=1

80: z=1

85: z=0

90: x=0

100: x=1

110: x=0

120: x=0

130: x=1

130: z=1

135: z=0

140: x=0

150: x=0

160: x=1

160: z=1

165: z=0

170: x=0

180: x=0

$finish called from file "proj3test2.v", line 36.

$finish at simulation time 190

V C S S i m u l a t i o n R e p o r t