|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X\YZ | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |

Y’Z + XZ + X’YZ’

wire or1, and1, and2, noty, notx, notz;

not (noty, y);

not (notx, x);

not(notz, z);

or (or1, x, noty);

and(and1, or1, z);

and(and2, notx, y, notz);

or (F, and1, and2);

2) wire data1, data2, data21, data3, data4;

Mux M1( a,b,c,d,Ctrl, data1);

Half\_adder HA1( data4, data1, Ctrl, data21);

Mux M2(data4, data2, data21, 1’b0, Ctrl, data3);

Register R1( data3, Clk, data4);

assign Sys\_out = data4;

3)

S0

S1

a

b

c

d

a

b

c F

4)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | X | A+ | B+ | Z |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

reg[1:0] state\_reg, state\_next;

parameter S0 = 2’b00;

parameter S1= 2’b01;

parameter S2=2’b10;

parameter S3=2’b11;

always @ (posedge clk, posedge Reset)

if (Reset) state\_reg <= S0;

else state\_reg <= state\_next;

always @(\*)

case (state\_reg)

S0: if(x) state\_next = S1;

else state\_next = S0;

S1: if(x) state\_next = S2;

else state\_next = S0;

S2: if(x) state\_next = S2;

else state\_next = S3;

S3: if(x) state\_next = S1;

else state\_next =S0;

default: state\_next = S0;

endcase

if (state\_reg == S3)

if (x == 1’b1)

z = 1’b1;

else

z = 1’b0;

else z = 1’b0;