Ripple Counter

T -flipflop

module TFilpFlop(

input T,

input clk,

input reset,

output reg Q,

output reg Qbar

);

reg temp=0;

always@(posedge clk , reset)

begin

if(reset==0)

begin

temp<=0;

// Qbar<=1;

end

else if(T==1)

begin

temp<=~temp;

// Qbar<=~Qbar;

end

Q<=temp;

Qbar<=~temp;

end;

endmodule

RippleCounter

module rippleCounter(

input clk,

input enable,

input reset,

output [3:0] Q

);

wire [4:0]w;

assign w[0] =clk;

TFilpFlop uu0(enable,w[0],reset,Q[0],w[1]);

TFilpFlop uu1(enable,w[1],reset,Q[1],w[2]);

TFilpFlop uu2(enable,w[2],reset,Q[2],w[3]);

TFilpFlop uu3(enable,w[3],reset,Q[3],w[4]);

Endmodule

clockDivider

module ClockDivider(

input clkin,

output reg clkout

);

integer count=0;

reg temp=0;

always @(posedge clkin)

begin

if(count==49999999)

begin

temp<=~temp;

count=0;

end

else

count=count+1;

clkout<=temp;

end

endmodule

top

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 08:30:41 PM

// Design Name:

// Module Name: topLV

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module topLV(

input enable,

input clk,

input reset,

output [3:0] Q

);

wire clkout;

ClockDivider uu1 (clk,clkout);

rippleCounter uu (clkout,enable,reset,Q);

endmodule

testbench

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 08:33:11 PM

// Design Name:

// Module Name: TopLV\_tb

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module TopLV\_tb(

);

wire [3:0]Q;

reg enable,clk,reset;

rippleCounter uuu (clk,enable,reset,Q);

integer i=0;

initial

begin

clk=0;

for(i=0;i<4000;i=i+1)

begin

#10 clk<=~clk;

end

end

initial

begin

reset<=0;

enable<=0;

#100

enable<=1;

#100

reset<=1;

enable<=0;

#100

enable<=1;

#100

enable<=0;

#100

enable<=1;

end

endmodule

testbench

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 08:33:11 PM

// Design Name:

// Module Name: TopLV\_tb

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module TopLV\_tb(

);

wire [3:0]Q;

reg updown,clk,reset;

counter uuu (updown,reset,clk,Q);

integer i=0;

initial

begin

clk=0;

for(i=0;i<4000;i=i+1)

begin

#10; clk<=~clk;

end

end

initial

begin

reset<=0;

updown<=0;

#10;

updown<=1;

#10;

reset<=1;

updown<=0;

#10;

updown<=1;

#10;

updown<=0;

#10;

updown<=1;

end

endmodule

Syncrouns

Counter

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 09:03:19 PM

// Design Name:

// Module Name: counter

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module counter(

input upDown,

input reset,

input clk,

output reg [3:0] Q

);

always @(posedge clk, negedge reset)

begin

if(reset==0)

Q<=0;

else if(upDown==1)

begin

if(Q==9)

Q<=0;

else

Q<=Q+1;

end

else

begin

if(Q==0)

Q<=9;

else

Q<=Q-1;

end

end

endmodule

PB debounce

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 09:29:27 PM

// Design Name:

// Module Name: PBDebounce

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module PBDebounce(

input PB,

input clk,

output reg PB\_Debounce

);

integer max =1000000;

reg old=0;

reg ton=0;

integer count=0;

always @(posedge clk)

begin

if(ton==0)

begin

if(PB!=old)

begin

ton<=1;

count=0;

old=PB;

PB\_Debounce=~PB\_Debounce;

end

end

else

begin

count=count+1;

if(count==max)

ton=0;

end

end

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 09:42:30 PM

// Design Name:

// Module Name: top

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module top(

input PB,

input reset,

input updown,

input clk,

output [3:0] Q

);

wire PBOUT;

PBDebounce u (PB,clk,PBOUT);

counter uuu(updown,reset,PBOUT,Q);

endmodule

Part4

Counter

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 10:03:48 PM

// Design Name:

// Module Name: BCDcounter

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module BCDcounter(

input reset,

input updown,

input clk,

output reg [3:0] dig0,

output reg [3:0] dig1

);

reg temp1=0;

reg temp2=0;

always @ (posedge clk,negedge reset)

begin

if(reset==0)

begin

dig0<=0;

dig1<=0;

end

else

begin

if(updown==0)// o->up

begin

if(dig0==9)

begin

dig0<=0;

if(dig1==9)

dig1<=0;

else

dig1<=dig1+1;

end

else

dig0<=dig0+1;

end

else

if (dig0==0)

begin

dig0<=9;

if(dig1==0)

dig1<=9;

else

dig1<=dig1-1;

end

else

dig0<=dig0-1;

end

end

Endmodule

TDM:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 10:28:25 PM

// Design Name:

// Module Name: TDM

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module TDM(

input [3:0] dig0,

input [3:0] dig1,

input clk,

output reg [3:0] digout,

output reg selectDig

);

integer counter=0;

reg sel=0;

always @(posedge clk)

begin

if(sel==0)

digout<=dig0;

else

digout<=dig1;

counter<=counter+1;

if(counter==1000000)

begin

counter<=0;

sel<=~sel;

end

selectDig<=sel;

end

endmodule

Decoder:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 02/21/2024 10:34:40 PM

// Design Name:

// Module Name: BCDdecoder

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module BCDdecoder(

input [3:0] digit,

output reg AA,

output reg AB,

output reg AC,

output reg AD,

output reg AE,

output reg AF,

output reg AG

);

always @(digit)

begin

case(digit)

0:{AA,AB,AC,AD,AE,AF,AG}=7'b1111110;

1:{AA,AB,AC,AD,AE,AF,AG}=7'b0110000;

2:{AA,AB,AC,AD,AE,AF,AG}=7'b1101101;

3:{AA,AB,AC,AD,AE,AF,AG}=7'b1111001;

4:{AA,AB,AC,AD,AE,AF,AG}=7'b0110011;

5:{AA,AB,AC,AD,AE,AF,AG}=7'b1011011;

6:{AA,AB,AC,AD,AE,AF,AG}=7'b1011111;

7:{AA,AB,AC,AD,AE,AF,AG}=7'b1110000;

8:{AA,AB,AC,AD,AE,AF,AG}=7'b1111111;

9:{AA,AB,AC,AD,AE,AF,AG}=7'b1110011;

default:{AA,AB,AC,AD,AE,AF,AG}=7'b0000000;

endcase

end

endmodule