Gagan Gupta

06/03/19

ELEN 21L F 2:15pm

Prelab#9

1. Next State table and state transition diagram

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Present State |  | Next State input Dependence |  | Outputs | | | | | |
|  |  |  |  | LCT | RCT | LSI | RSI | S[1] | S[0] |
| sIDLE |  | if(iRSRV==1)NSTATE=sRSRV;  else if(iLSRV==1)NSTATE=sLSRV;  else NSTATE=sIDLE; |  | 0 | 0 | 0 | 0 | 1 | 1 |
| sRSRV |  | if(iRESET==1)NSTATE=sIDLE;  else NSTATE=sMOVL; |  | 0 | 0 | 1 | 0 | 1 | 0 |
| sMOVL |  | if(iRESET==1)NSTATE=sIDLE;  else if(QLEFT==1 && iLPAD==0)NSTATE=sENDL;  else if(QLEFT==0 && iLPAD==1)NSTATE=sMISL;  else if(QLEFT==1 && iLPAD==1)NSTATE=sMOVR;  else NSTATE=sMOVL; |  | 0 | 0 | 0 | 0 | 1 | 0 |
| sENDL |  | NSTATE=sIDLE; |  | 0 | 1 | 0 | 0 | 0 | 0 |
| sMISL |  | if(iRESET==1)NSTATE=sIDLE;  else if(iQLEFT==1)NSTATE=sENDL;  else NSTATE=sMISL |  | 0 | 0 | 0 | 0 | 1 | 0 |
| sLSRV |  | if(iRESET==1)NSTATE=sIDLE;  else NSTATE=sMOVR; |  | 0 | 0 | 0 | 1 | 0 | 1 |
| sMOVR |  | if(iRESET==1)NSTATE=sIDLE;  else if(QRIGHT==1 && iRPAD==0)NSTATE=sENDR;  else if(QRIGHT==0 && iRPAD==1)NSTATE=sMISR;  else if(QLEFT==1 && iLPAD==1)NSTATE=sMOVL;  else NSTATE=sMOVR; |  | 0 | 0 | 0 | 0 | 0 | 1 |
| sENDR |  | NSTATE=sIDLE |  | 1 | 0 | 0 | 0 | 0 | 0 |
| sMISR |  | if(iRESET==1)NSTATE=sIDLE;  else if(iQRIGHT==1)NSTATE=sENDR;  else NSTATE=sMISR; |  | 0 | 0 | 0 | 0 | 1 | 0 |

1. Verilog for pong\_controller2

Module pong\_controller2(QLEFT, QRIGHT, iRESET, iRSRV. iLSRV, iRPAD, iLPAD, VCLK, PSTATE, S, LCT, RCT, LSI, RSI);

input QLEFT, QRIGHT, iRESET, iRSRV. iLSRV, iRPAD, iLPAD, VCLK;

output reg LCT, RCT, LSI, RSI;

output reg [9:1]PSTATE;

output reg [1:0]S;

parameter [9:1]sIDLE=9'b100000000, sRSRV=9'b000000001, sMOVL=9'b000000010, sENDL=9'b000000100, sMISL=9'b000001000, sLSRV=9'b000010000, sMOVR=9'b000100000, sENDR=9'b001000000, sMISR=9'b010000000;

parameter reg [9:1] NSTATE;

always@(\*)

begin

case(PSTATE)

sIDLE:begin

LCT=0;

RCT=0;

LSI=0;

RSI=0;

S[1]=1;

S[0]=1;

if(iRSRV==1) NSTATE = sRSRV;

else NSTATE = sIDLE;

end

sRSRV:begin

LCT=0;

RCT=0;

LSI=1;

RSI=0;

S[1]=1;

S[0]=0;

if(iRESET==1) NSTATE = sIDLE;

else NSTATE = sMOVL;

end

sMOVL:begin

LCT=0;

RCT=0;

LSI=0;

RSI=0;

S[1]=1;

S[0]=0;

if(iRESET==1) NSTATE =sIDLE;

else if(Q1[3]==1)NSTATE = sENDL;

else NSTATE = sMOVL;

end

sENDL:begin

LCT=0;

RCT=1;

LSI=0;

RSI=0;

S[1]=0;

S[0]=0;

NSTATE = sIDLE;

end

default:NSTATE = sIDLE;

endcase

end

always@(posedge Clock)

begin

if(iRESET==1)PSTATE<=sIDLE;

else PSTATE<=NSTATE;

end

1. Test plan
2. Verilog for upcount4

module upcount4(CLK, iRESET, CT, count);

input CLK, iRESET, CT;

output reg [3:0]count;

always@(posedge CLK)

begin

if(CT==1)count<=count+1;

else if(iRESET==1)count<=4'b0000;

else count<=count;

end  
endmodule