module REG\_74194 (CLR, S1, S0, CLK, SL, SR, A, B, C, D, QA, QB, QC, QD);

input CLR, S1, S0, CLK, SL, SR;

input A, B, C, D;

output QA, QB, QC, QD;

reg QA, QB, QC, QD;

always@(posedge CLK)

begin

if(!CLR)

begin

QA = 0;

QB = 0;

QC = 0;

QD = 0;

end

else

begin

case({S1,S0})

2'b00:begin QA = QA;QB = QB;QC = QC;QD = QD; end

2'b11:begin QA = A;QB = B;QC = C;QD = D; end

2'b01:

begin

case(SR)

1'b1:begin QD = QC; QC = QB; QB = QA; QA = 1; end

1'b0:begin QD = QC; QC = QB; QB = QA; QA = 0; end

endcase

end

2'b10:

begin

case(SL)

1'b1:begin QA = QB;QB = QC;QC = QD;QD = 1; end

1'b0:begin QA = QB;QB = QC;QC = QD;QD = 0; end

endcase

end

endcase

end

end

Endmodule

`timescale 1ns/1ps

module tb;

reg CLR, S1, S0, CLK, SL, SR;

reg A, B, C, D;

wire QA, QB, QC, QD;

always # 1 CLK = ~CLK;

initial

begin

CLK = 0;//clear

CLR = 0;

A = 1;B = 0;C = 1;D = 0;

S1 = 0;S0 = 0;

#2

CLR = 1;

S1 = 0;S0 = 0;//bao chi

#2

S1 = 1;S0 = 1; //zhi ru

#2

S1 = 0;S0 = 1;//you yi 0

SR = 0;

#2

S1 = 0;S0 = 1;//you yi 1

SR = 1;

#2

S1 = 1;S0 = 0;//zuo yi 0

SL = 0;

#2

S1 = 1;S0 = 0;//zuo yi 1

SL = 1;

#2

S1 = 0;S0 = 0;//bao chi

SR = 0;SL = 0;

end

REG\_74194 uut(.A(A),.B(B),.C(C),.D(D),.SR(SR),.SL(SL),.CLK(CLK),.CLR(CLR),.QA(QA),.QB(QB),.QC(QC),.QD(QD),.S0(S0),.S1(S1));

endmodule