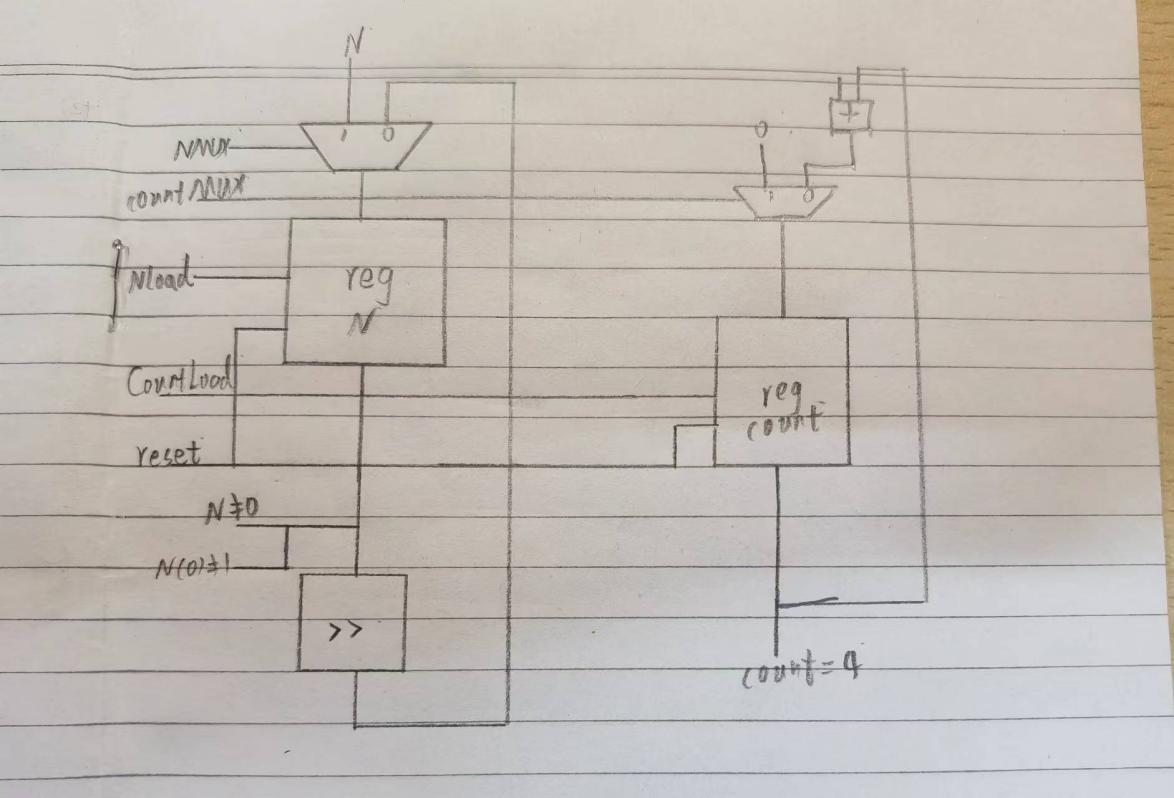
**Lab 5**

1. **电路图**

****

1. **代码**

**`timescale 1ns / 1ps**

**//////////////////////////////////////////////////////////////////////////////////**

**// Company:**

**// Engineer:**

**//**

**// Create Date: 2022/05/10 09:55:26**

**// Design Name:**

**// Module Name: dedicated\_micro**

**// Project Name:**

**// Target Devices:**

**// Tool Versions:**

**// Description:**

**//**

**// Dependencies:**

**//**

**// Revision:**

**// Revision 0.01 - File Created**

**// Additional Comments:**

**//**

**//////////////////////////////////////////////////////////////////////////////////**

**module adder(**

**input [3:0] s1,s2,**

**output [3:0] out**

**);**

**assign out = s1 + s2;**

**endmodule**

**module Mux8(**

**input [7:0] s1,**

**input [7:0] s2,**

**input NMUX,**

**output reg [7:0] out**

**);**

**always@(\*)**

**begin**

**if (NMUX == 0)**

**begin**

**out = s1;**

**end**

**else begin**

**out = s2;**

**end**

**end**

**endmodule**

**module Mux4(**

**input [3:0] s1,**

**input [3:0] s2,**

**input MUX,**

**output [3:0] out**

**);**

**assign out = (MUX == 0) ? s1: s2;**

**endmodule**

**module Move\_right(**

**input [7:0] N,**

**output [7:0] out**

**);**

**assign out = N>>1;**

**endmodule**

**module register8(**

**input clk, reset,NLoad,**

**input [7:0] in,**

**output reg [7:0] out**

**);**

**always@(posedge clk or posedge reset)**

**begin**

**if (reset == 1) out <= 0;**

**else if (NLoad == 1) out <= in;**

**end**

**endmodule**

**module register4(**

**input clk, reset,CountLoad,**

**input [3:0] in,**

**output reg [3:0] out**

**);**

**always@(posedge clk or posedge reset)**

**begin**

**if (reset == 1) out <= 0;**

**else**

**begin**

**if (CountLoad == 1) out <= in;**

**end**

**end**

**endmodule**

**module DataPath(**

**input clk, reset, NMUX, CountMUX,OutputMUX,NLoad,CountLoad,OE,**

**input [7:0] N,**

**output Nnotzero,Nnotone,CountFour,**

**output out, start**

**);**

**wire [7:0] N\_mux,N\_moved,N\_register;**

**wire [3:0] Count\_mux, Count\_added,Count\_register;**

**wire out1;**

**Mux8 m1(.s1(N\_moved),.s2(N),.NMUX(NMUX),.out(N\_mux));**

**register8 r1(.clk(clk),.reset(reset),.NLoad(NLoad),**

**.in(N\_mux),.out(N\_register));**

**assign Nnotzero = (N\_register == 0) ? 0:1;**

**assign Nnotone = (N\_register[0] == 1) ? 0:1;**

**Move\_right t1(.N(N\_register),.out(N\_moved));**

**Mux4 m2(.s1(Count\_added),.s2(0),.MUX(CountMUX),.out(Count\_mux));**

**register4 r2(.clk(clk),.reset(reset),.CountLoad(CountLoad),**

**.in(Count\_mux),.out(Count\_register));**

**assign CountFour = (Count\_register == 4)?1:0;**

**adder a1(.s1(1),.s2(Count\_register),.out(Count\_added));**

**Mux4 m3(.s1(0),.s2(1),.MUX(OutputMUX),.out(out1));**

**bufif1(out,out1,OE);**

**endmodule**

**module control(**

**input clk, reset, Nnotzero,Nnotone,CountFour,**

**output reg OE,**

**output reg NMUX,**

**output reg CountMUX,**

**output reg OutputMUX,**

**output reg NLoad,**

**output reg CountLoad**

**);**

**parameter S1=3'b000;**

**parameter S2=3'b001;**

**parameter S3=3'b010;**

**parameter S4=3'b011;**

**parameter S5=3'b100;**

**parameter S6=3'b101;**

**parameter S\_rst=3'b110;**

**reg [2:0] state,next\_state;**

**always@(\*)**

**begin**

**case(state)**

**S\_rst: next\_state = S1;**

**S1: next\_state = S2;**

**S2:**

**if(Nnotzero==1&& Nnotone==0) next\_state=S3;**

**else if(Nnotzero==1&&Nnotone==1) next\_state=S4;**

**else if(Nnotzero==0&&CountFour==1) next\_state=S6;**

**else if(Nnotzero==0&&CountFour==0) next\_state=S5;**

**S3:next\_state=S4;**

**S4:next\_state=S2;**

**S5:next\_state=S5;**

**S6:next\_state=S6;**

**endcase**

**end**

**always@(posedge clk or posedge reset)**

**begin**

**if(reset)**

**begin**

**state <= S\_rst;**

**end**

**else**

**begin**

**state <= next\_state;**

**end**

**end**

**always@(state)**

**begin**

**if(state==S1)**

**begin**

**NMUX=1;CountMUX=1;NLoad=1;CountLoad=1;OutputMUX=0;OE=0;**

**end**

**else if(state==S2)**

**begin**

**NMUX=0; CountMUX=0;NLoad=0;CountLoad=0;OutputMUX=0;OE=0;**

**end**

**else if(state==S3)**

**begin**

**NMUX=0;CountMUX=0; NLoad=0;CountLoad=1;OutputMUX=0;OE=0;**

**end**

**else if(state==S4)**

**begin**

**NMUX=0;CountMUX=0; NLoad=1;CountLoad=0;OutputMUX=0; OE=0;**

**end**

**else if(state==S5)**

**begin**

**NMUX=0;CountMUX=0;NLoad=0;CountLoad=0;OutputMUX=0;OE=1;**

**end**

**else if(state==S6)**

**begin**

**NMUX=0;CountMUX=0; NLoad=0; CountLoad=0;OutputMUX=1;OE=1;**

**end**

**end**

**endmodule**

**module dedicated\_micro(**

**input clk,reset,**

**input [7:0] N,**

**output led,**

**output [7:0] out**

**);**

**wire NMUX,CountMUX,OutputMUX,NLoad,CountLoad,OE;**

**wire Nnotzero, Nnotone, CountFour;**

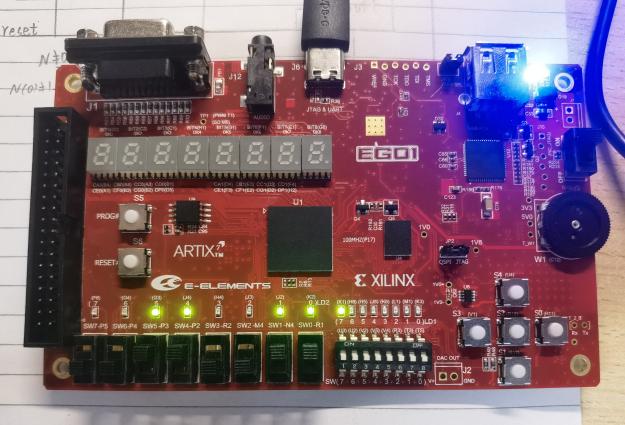
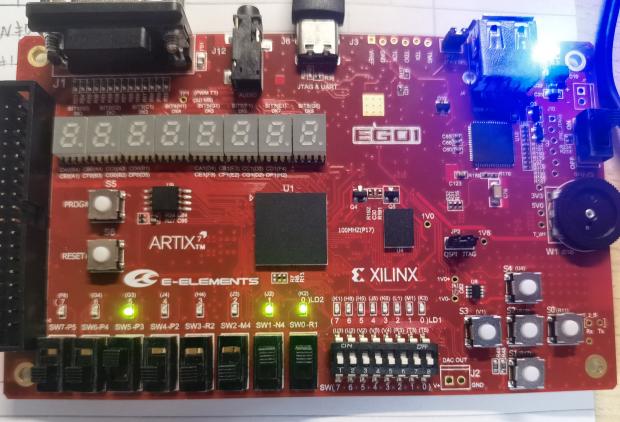
**assign out = N;**

**control c1(.clk(clk),.reset(reset),.Nnotzero(Nnotzero),.Nnotone(Nnotone),.CountFour(CountFour),.OE(OE),.NMUX(NMUX),.CountMUX(CountMUX),.OutputMUX(OutputMUX),.NLoad(NLoad),.CountLoad(CountLoad));**

**DataPath d1(.clk(clk),.reset(reset),.N(N),.NMUX(NMUX),.CountMUX(CountMUX),.OutputMUX(OutputMUX),.NLoad(NLoad),.CountLoad(CountLoad),.OE(OE),.Nnotzero(Nnotzero),.Nnotone(Nnotone),.CountFour(CountFour),.out(led));**

**endmodule**

1. **效果图**

****