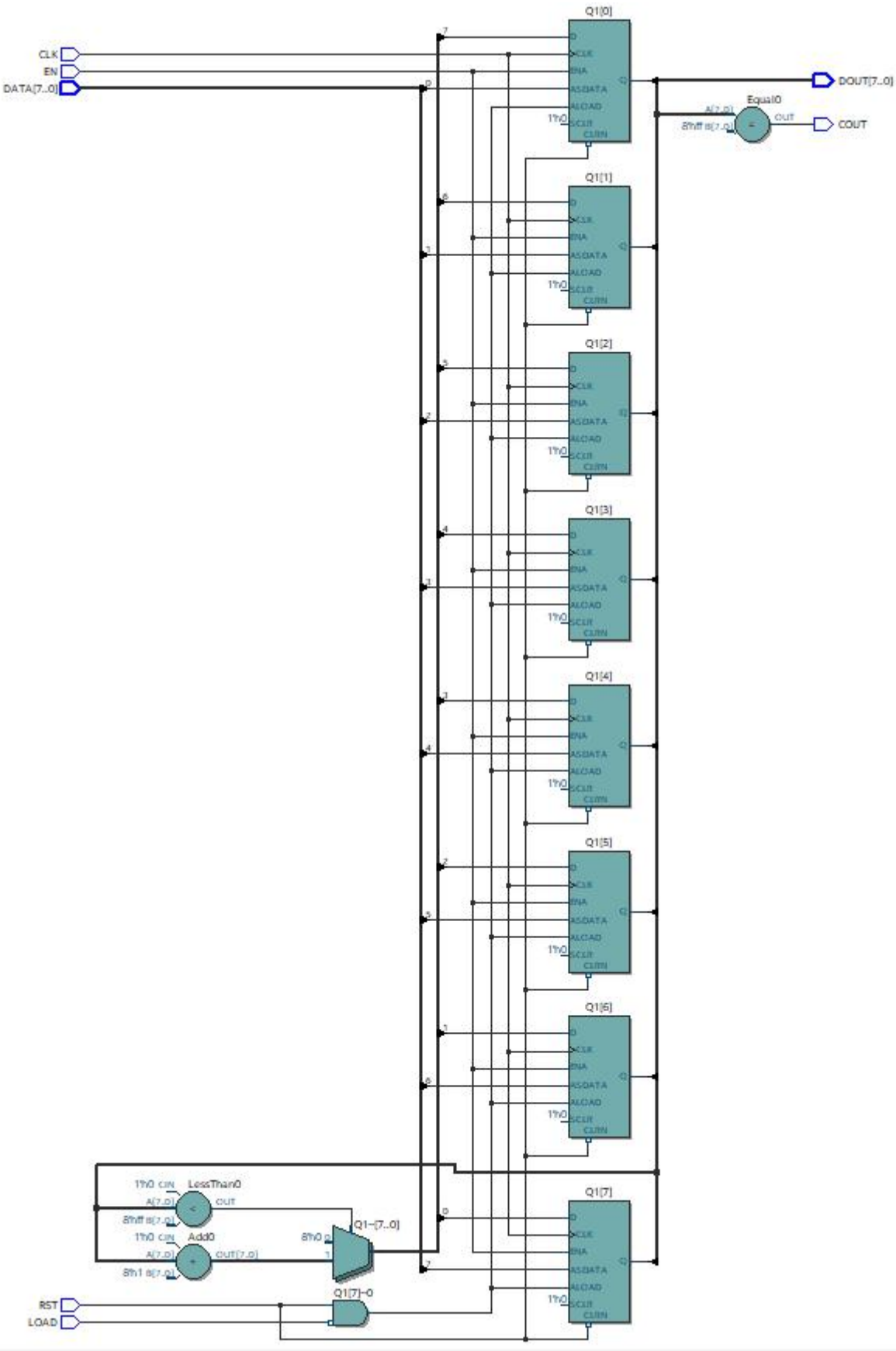
2020302181062
崔浩然
EDA 数字逻辑第五章作业

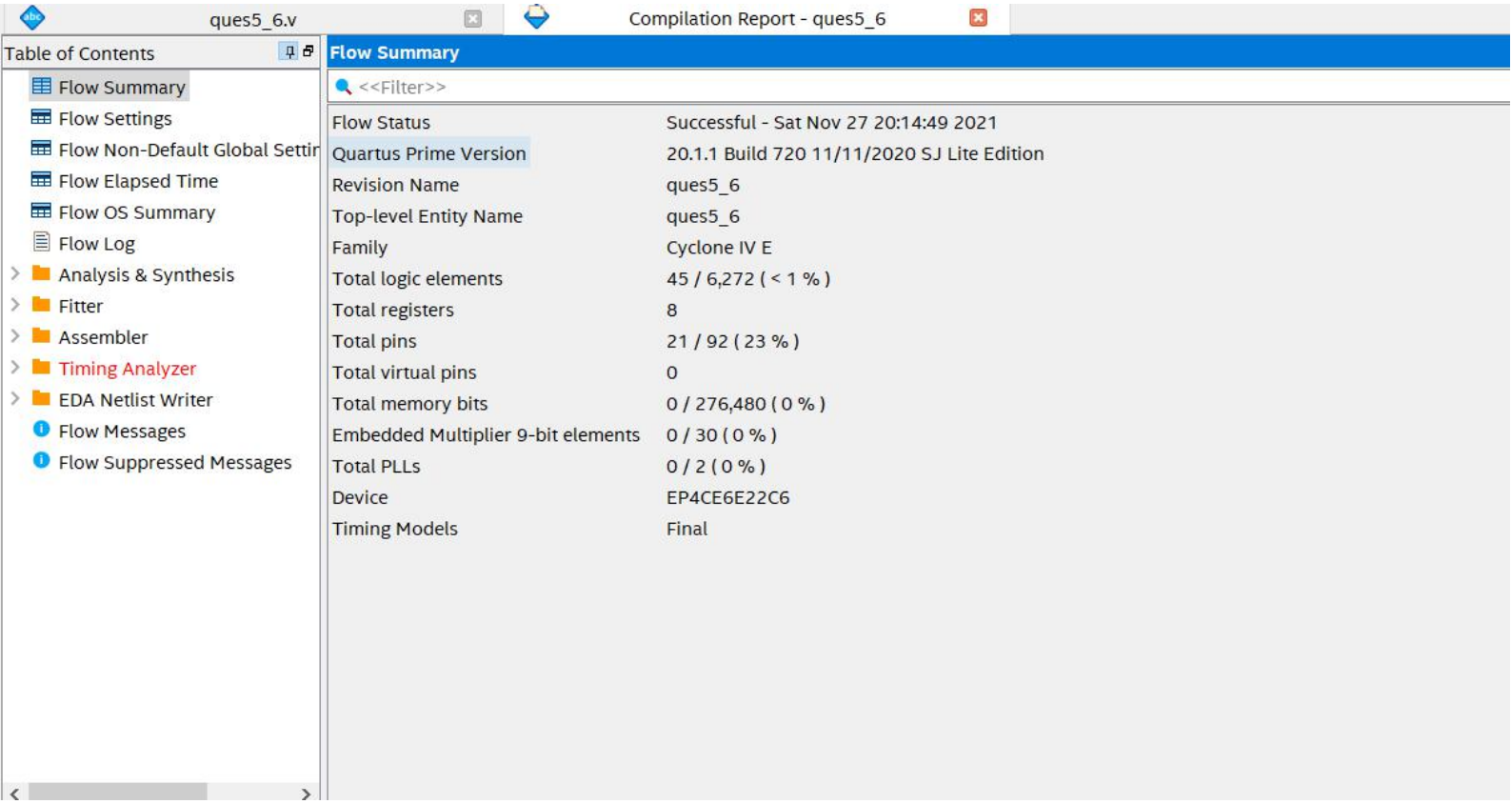
5-6：
代码文件：

```
module ques5_6 (CLK,RST,EN,LOAD,COUT,DOUT,DATA);
    input CLK,EN,RST,LOAD;
    input[7:0] DATA;
    output[7:0] DOUT;
    output COUT;
    reg[7:0] Q1;
    reg COUT;
    assign DOUT = Q1;
    always @(posedge CLK or negedge RST or negedge LOAD)
        begin
            if (!RST)
                Q1 <= 0;
            else if (!LOAD)
                Q1 = DATA;
            else if (EN) begin
                    if(Q1<255)
                        Q1 <= Q1+1;
                    else
                        Q1 <= 8'b00000000;
            end
        end
    always @(Q1)
        if (Q1==8'd255) COUT = 1'b1;
        else COUT=1'b0;
Endmodule
```

网表文件：


CLK
EN
DATA[7..0]
Q1[0]
Q1[1]
Q1[2]
Q1[3]
Q1[4]
Q1[5]
Q1[6]
Q1[7]
DOUT[7..0]
Equal0
COUT
LessThan0
Add0
Q1-[7..0]
Q1[7]~0
RST
LOAD

资源开销：



仿真文件：

```
`timescale 1 ps/ 1 ps
module ques5_6_vlg_tst();
// constants
// general purpose registers
//reg eachvec;
// test vector input registers
reg CLK;
reg [7:0] DATA;
reg EN;
reg LOAD;
reg RST;
// wires
wire COUT;
wire [7:0]  DOUT;

// assign statements (if any)
ques5_6 i1 (
// port map - connection between master ports and signals/registers
    .CLK(CLK),
    .COUT(COUT),
    .DATA(DATA),
    .DOUT(DOUT),
    .EN(EN),
    .LOAD(LOAD),
    .RST(RST)
);
initial
begin
    CLK=0;
    DATA=8'b00100100;
    EN=1;
```

```
    LOAD=1;
    RST=1;
    //COUT=0;
    //DOUT=000;
    //#100 LOAD=0;
        #100
        LOAD = 0;
        #5
        LOAD = 1;

// code that executes only once
// insert code here --> begin

// --> end
$display("Running testbench");
end

//always
// optional sensitivity list
// @(event1 or event2 or .... eventn)
//begin
always #10 CLK=~CLK;



//always  #100 LOAD=~LOAD;
// code executes for every event on sensitivity list
// insert code here --> begin

//@eachvec;
// --> end
//end
endmodule
```

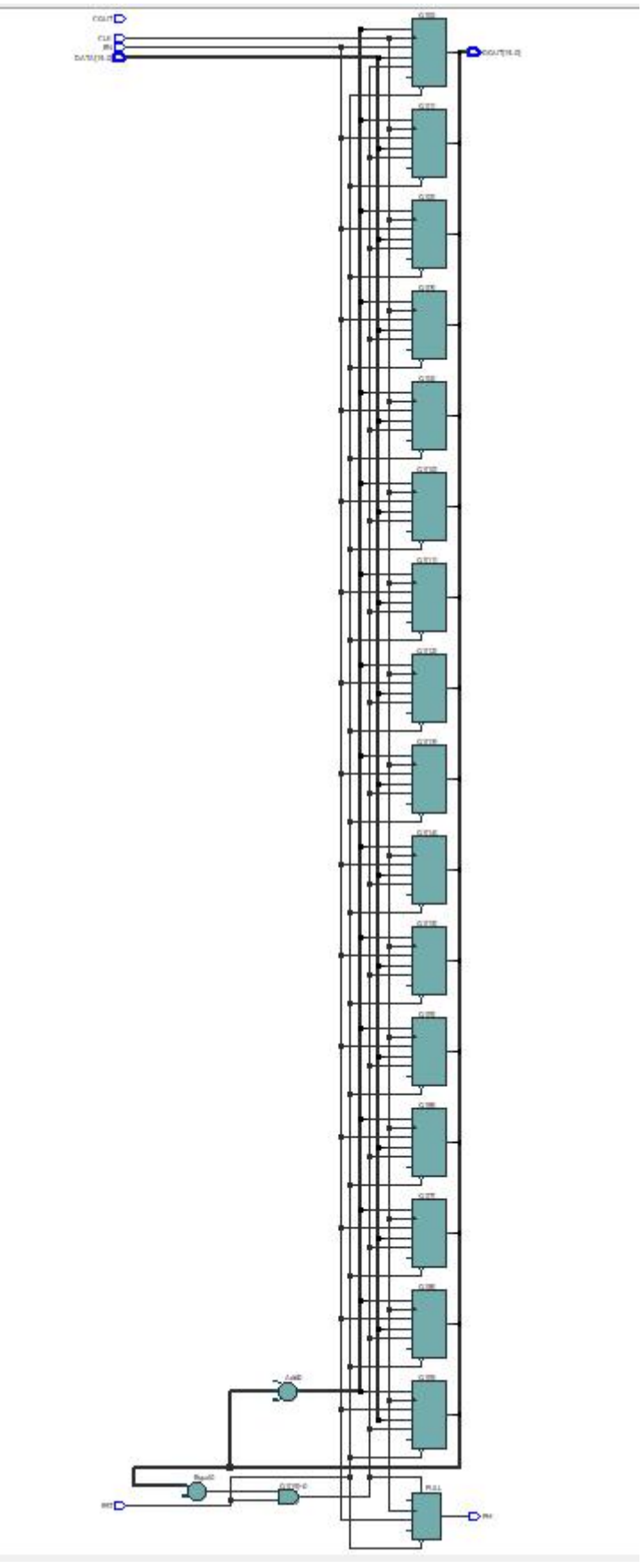
5-7：

代码文件：

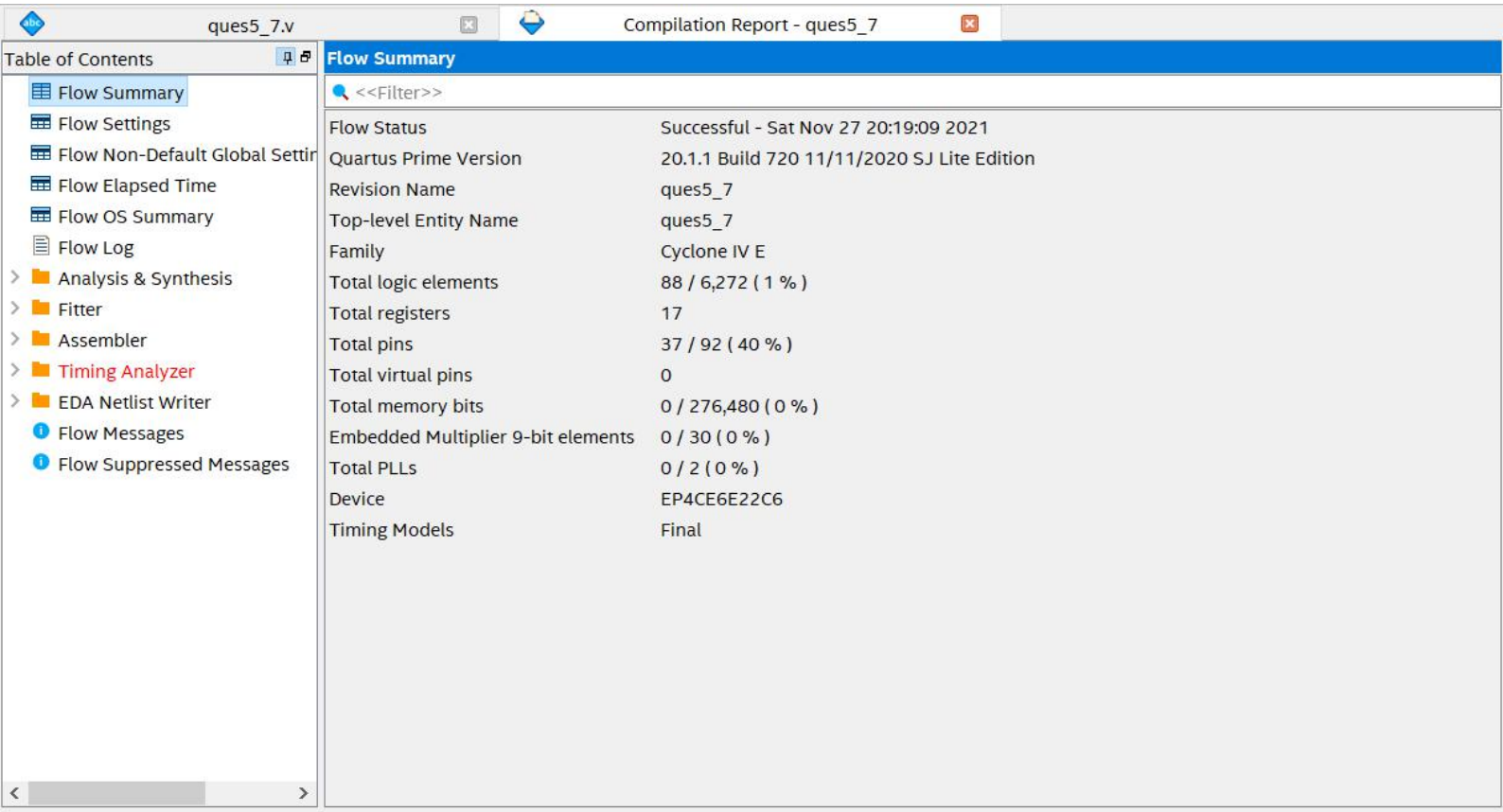
```
module ques5_7 (CLK,RST,DATA,COUT,DOUT,EN,PM);
    input CLK,RST,COUT,EN;
    input[15:0] DATA;
    output[15:0] DOUT;
    output PM;
    reg FULL;
    reg[15:0] Q1;
    wire LOAD;
    always @(posedge CLK or negedge RST or posedge LOAD)
        begin
            if(!RST)
                begin
                    Q1<=0;FULL<=0;
                end
            else if(LOAD)
                begin
                    Q1<=DATA;FULL<=1;
                end
```

```
            else if(EN) begin
                Q1<=Q1+1;FULL<=0;
                end
        end
    assign LOAD=(Q1==16'd16);
    assign PM=FULL;
    assign DOUT=Q1;
Endmodule
```

网表文件：



资源开销：



仿真文件：

```
`timescale 1 ps/ 1 ps
module ques5_7_vlg_tst();
// constants
// general purpose registers
//reg eachvec;
// test vector input registers
reg CLK;
reg COUT;
reg [15:0] DATA;
reg EN;
reg RST;
// wires
wire [15:0]  DOUT;
wire PM;

// assign statements (if any)
ques5_7 i1 (
// port map - connection between master ports and signals/registers
    .CLK(CLK),
    .COUT(COUT),
    .DATA(DATA),
    .DOUT(DOUT),
    .EN(EN),
    .PM(PM),
    .RST(RST)
);
initial
begin
    CLK=0;
    COUT=0;
```

```
    DATA=16'd6;
    EN=1;
    RST=1;
// code that executes only once
// insert code here --> begin

// --> end
$display("Running testbench");
end
always  #10 CLK=~CLK;
// optional sensitivity list
// @(event1 or event2 or .... eventn)
always@(RST)
begin
    #100 RST=~RST;
    #5 RST=1;
// code executes for every event on sensitivity list
// insert code here --> begin

//@eachvec;
// --> end
end
endmodule
```

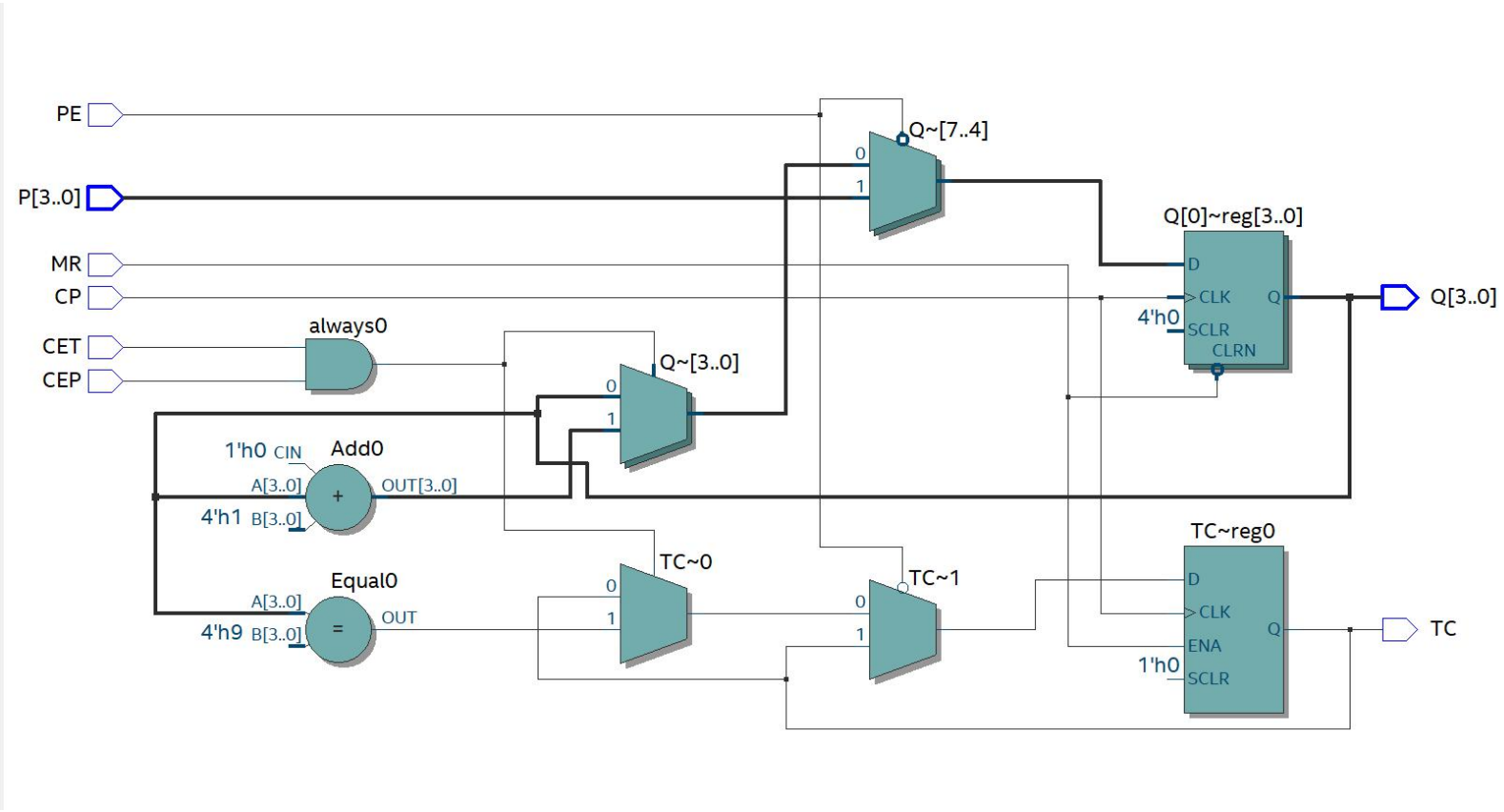
5-10：

代码文件：

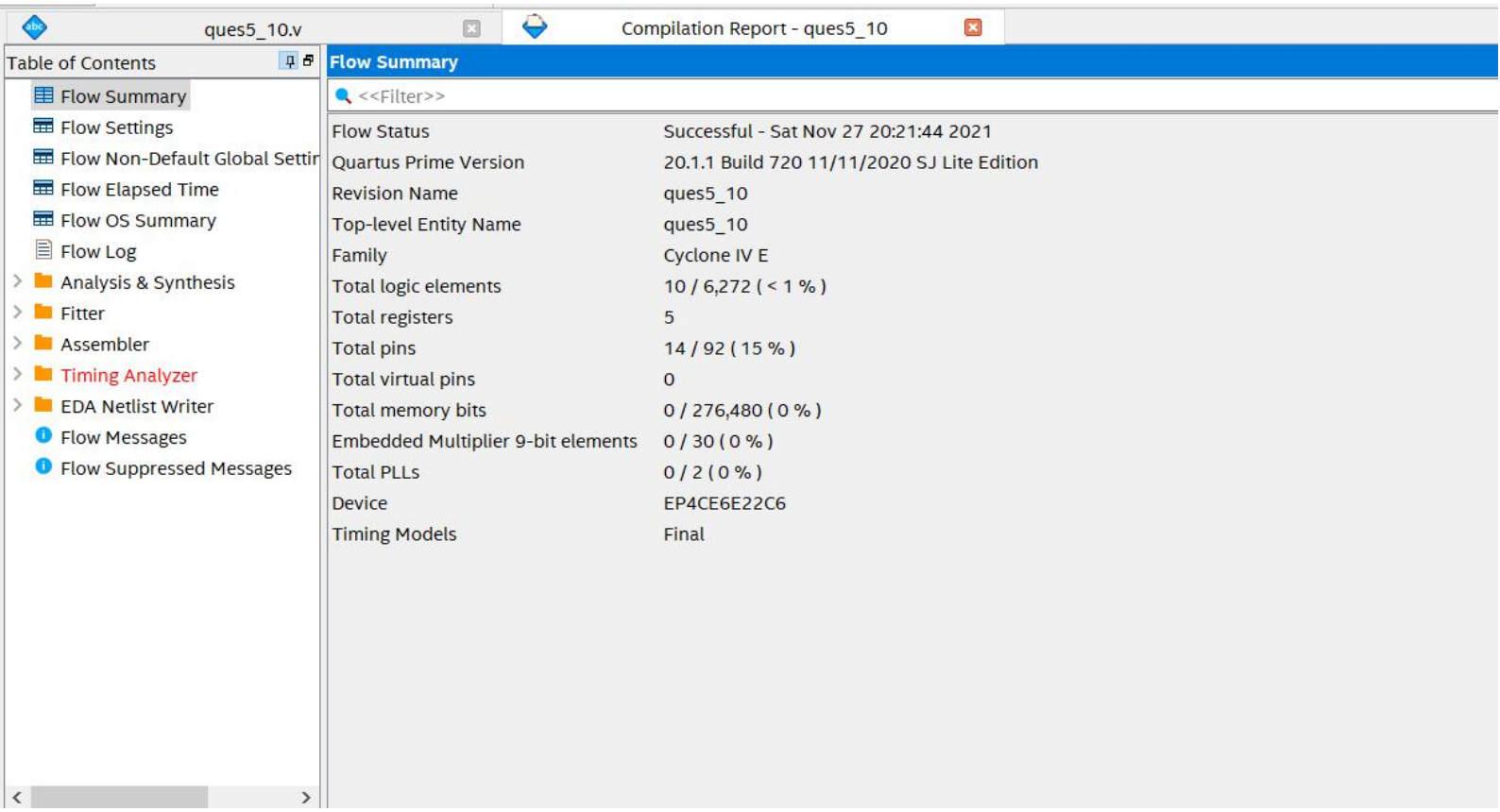
```
module ques5_10(CP,PE,MR,CET,CEP,TC,Q,P);
    input CP,PE,MR,CEP,CET;
    input[3:0] P;
    output[3:0] Q;
    output TC;
    reg TC;
    reg[3:0] Q;
    always @(negedge MR or posedge CP) begin
        if(!MR) begin
            Q<=0;
        end
        else if(!PE) begin
            Q<=P;
        end
        else if(CEP&CET) begin
            Q<=Q+1;
            if(Q==4'd9) begin
                TC<=1;
            end
            else begin
                TC<=0;
            end
        end
        else begin
            Q<=Q;
        end
    end
```

```
endmodule
```

网表文件：



资源开销：



仿真文件：

```
`timescale 1 ps/ 1 ps
module ques5_10_vlg_tst();
// constants
// general purpose registers
//reg eachvec;
// test vector input registers
```

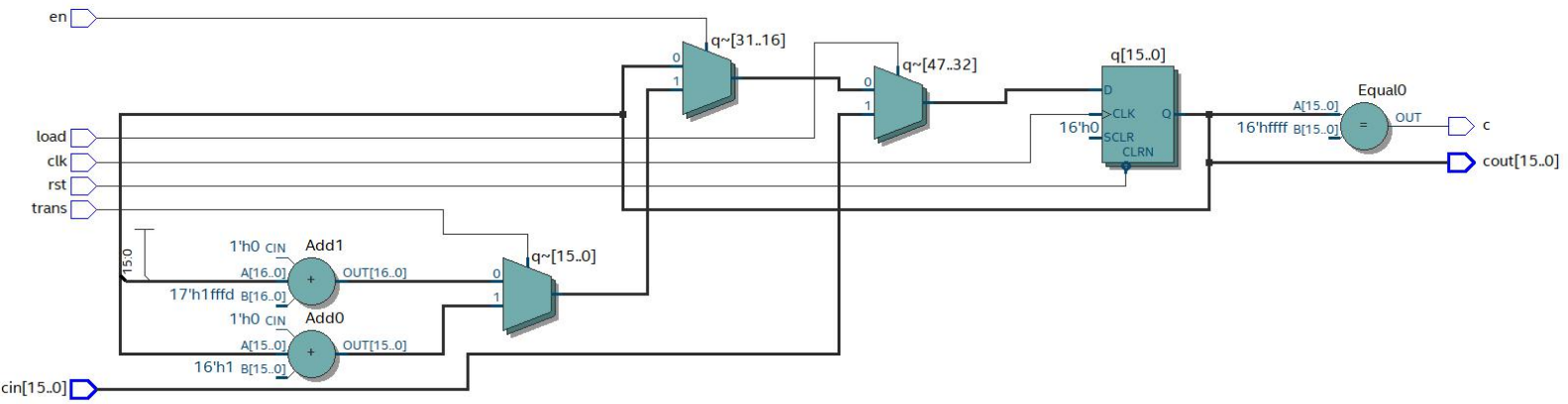
```
reg CEP;
reg CET;
reg CP;
reg MR;
reg [3:0] P;
reg PE;
// wires
wire [3:0]  Q;
wire TC;

// assign statements (if any)
ques5_10 i1 (
// port map - connection between master ports and signals/registers
   .CEP(CEP),
   .CET(CET),
   .CP(CP),
   .MR(MR),
   .P(P),
   .PE(PE),
   .Q(Q),
   .TC(TC)
);
initial
begin
   CEP=1;
   CET=1;
   CP=0;
   MR=1;
   P=4'b0010;
   PE=1;
// code that executes only once
// insert code here --> begin

// --> end
$display("Running testbench");
end
always #10 CP=~CP;
// optional sensitivity list
// @(event1 or event2 or .... eventn)
always @(MR)
begin
// code executes for every event on sensitivity list
// insert code here --> begin
#128 MR=~MR;
#133 MR=1;
#64 PE=~PE;
#8 PE=1;
//@eachvec;
// --> end
end
endmodule
```

5-11：

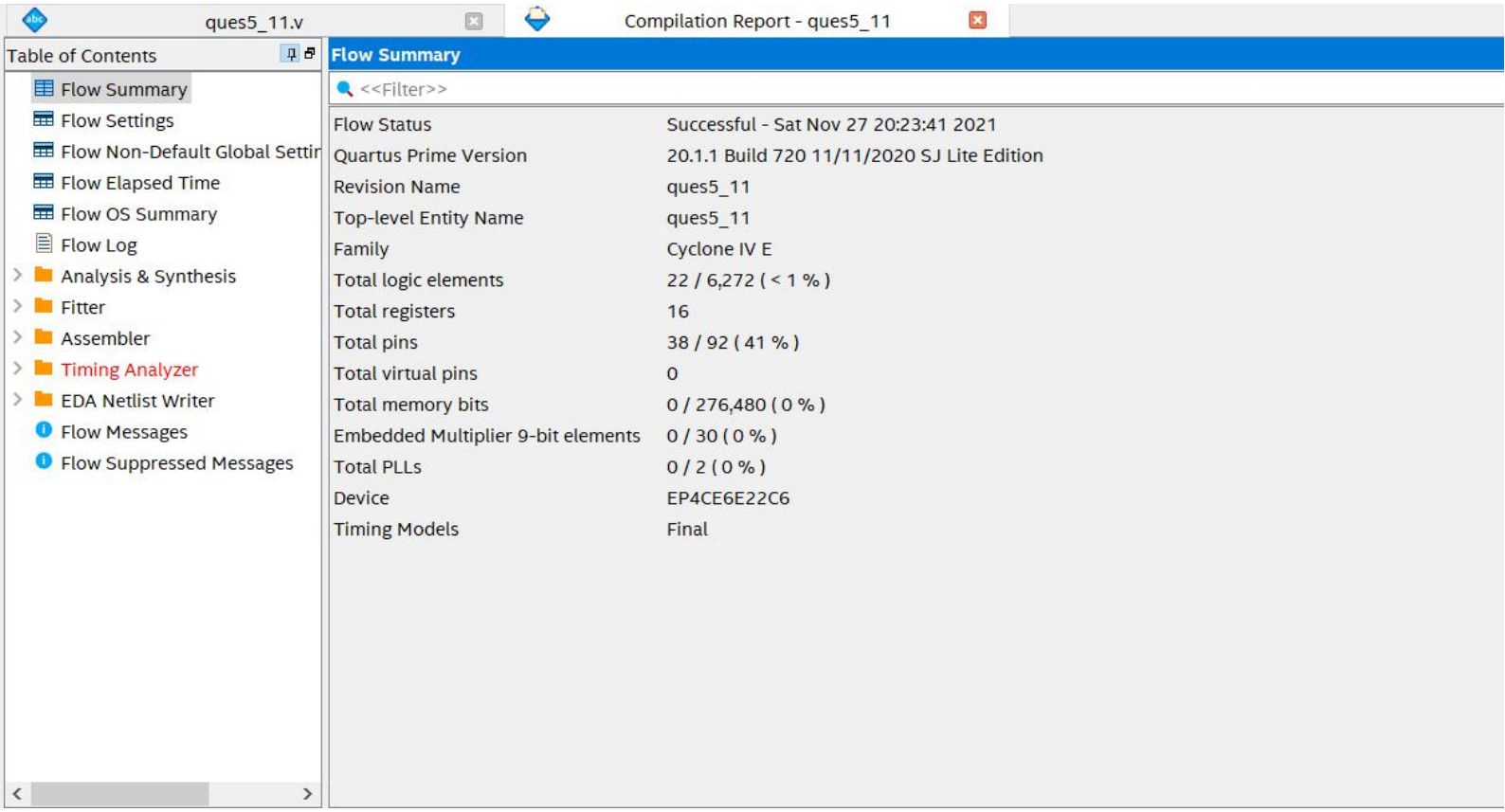
代码文件：

```
module ques5_11(rst,en,clk,trans,load,cin,cout,c);
    input rst,en,clk,trans,load;
    input[15:0] cin;
    output[15:0] cout;
    output c;
    reg[15:0] q;
    reg c;
    always @(posedge clk or negedge rst) begin
        if(!rst) begin
            q<=0;
        end
        else if(load)
            q<=cin;
        else if(en) begin
            if(trans) begin
                q<=q+1;
            end
            else
                q<=q-1;
        end
    end
    always @(q) begin
        if(q==16'd65535)
            c=1'b1;
        else
            c=1'b0;
    end
    assign cout=q;
Endmodule
```

网表文件：



资源开销：



仿真文件：

```
`timescale 1 ps/ 1 ps
module ques5_11_vlg_tst();
// constants
// general purpose registers
//reg eachvec;
// test vector input registers
reg [15:0] cin;
reg clk;
reg en;
reg load;
reg rst;
reg trans;
// wires
wire [15:0]  cout;
wire c;
// assign statements (if any)
ques5_11 i1 (
// port map - connection between master ports and signals/registers
    .cin(cin),
    .clk(clk),
    .cout(cout),
    .en(en),
    .load(load),
    .rst(rst),
    .trans(trans),
    .c(c)
);
initial
begin
// code that executes only once
// insert code here --> begin
```

```
cin=16'd24;
clk=0;
en=1;
load=0;
rst=1;
trans=1;
// --> end
$display("Running testbench");
end
always #10 clk=~clk;
// optional sensitivity list
// @(event1 or event2 or .... eventn)
always@ (load or trans)
begin
// code executes for every event on sensitivity list
// insert code here --> begin
#24 load=~load;
#8 load=0;
#64 trans=~trans;
#64 trans=~trans;
//@eachvec;
// --> end
end
Endmodule
```

数字逻辑作业

2020302180062
崔浩然
数逻

武漢大學
WUHAN UNIVERSITY
Wuhan 430072, Hubei, P.R.China 中国·武汉 Tel.(027)

习题五

5.6

写输出函数与激励函数表达式，该电路为Mealy型电路

$Z=X\bar{y}_1\bar{y}_2+Xy_1y_2$　　$J_1=K_1=1$，$J_2=K_2=X\oplus y_1$

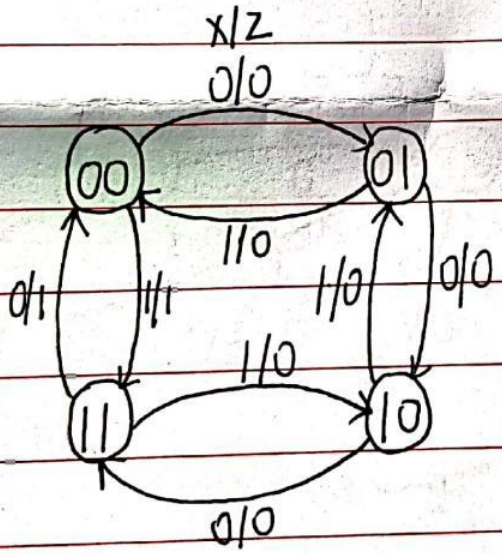
次态真值表

输入	现态		激励函数				次态	
X	y_2	y_1	J_2	K_2	J_1	K_1	y_2^{n+1}	y_1^{n+1}
0	0	0	0	0	1	1	0	1
0	0	1	1	1	1	1	1	0
0	1	0	0	0	1	1	1	1
0	1	1	1	1	1	1	0	0
1	0	0	1	1	1	1	1	1
1	0	1	0	0	1	1	0	0
1	1	0	1	1	1	1	0	1
1	1	1	0	0	1	1	1	0

状态表

现态	次态 $y_2^{n+1}y_1^{n+1}$ / 输出Z	
y_2y_1	X=0	X=1
00	01/0	11/1
01	10/0	00/0
10	11/0	01/0
11	00/1	10/1

状态图

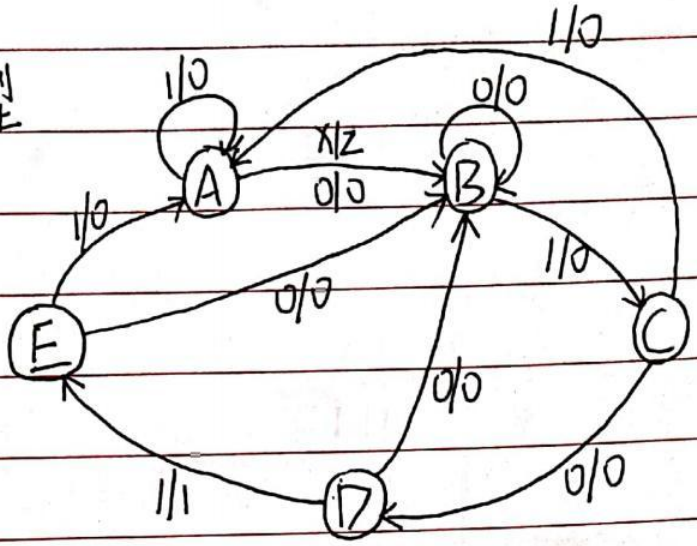


由状态图与状态表可知，该电路为一个模四可逆计数器，X=0时进行加1计数，输出Z为进位信号，当X=1时实现减1计数，输出Z为借位信号，即Z=0不借位，Z=1借位

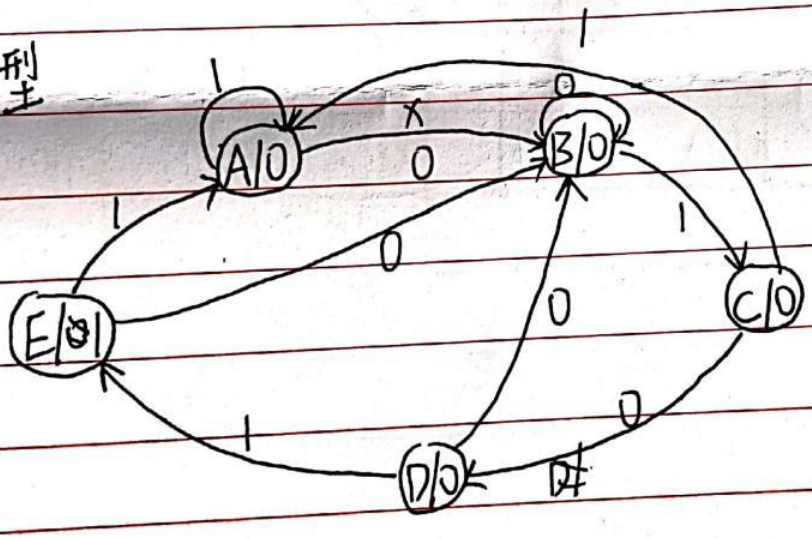
武漢大學
WUHAN UNIVERSITY
Wuhan 430072, Hubei, P.R.China 中国·武汉 Tel.(027)

题5.7

Mealy型

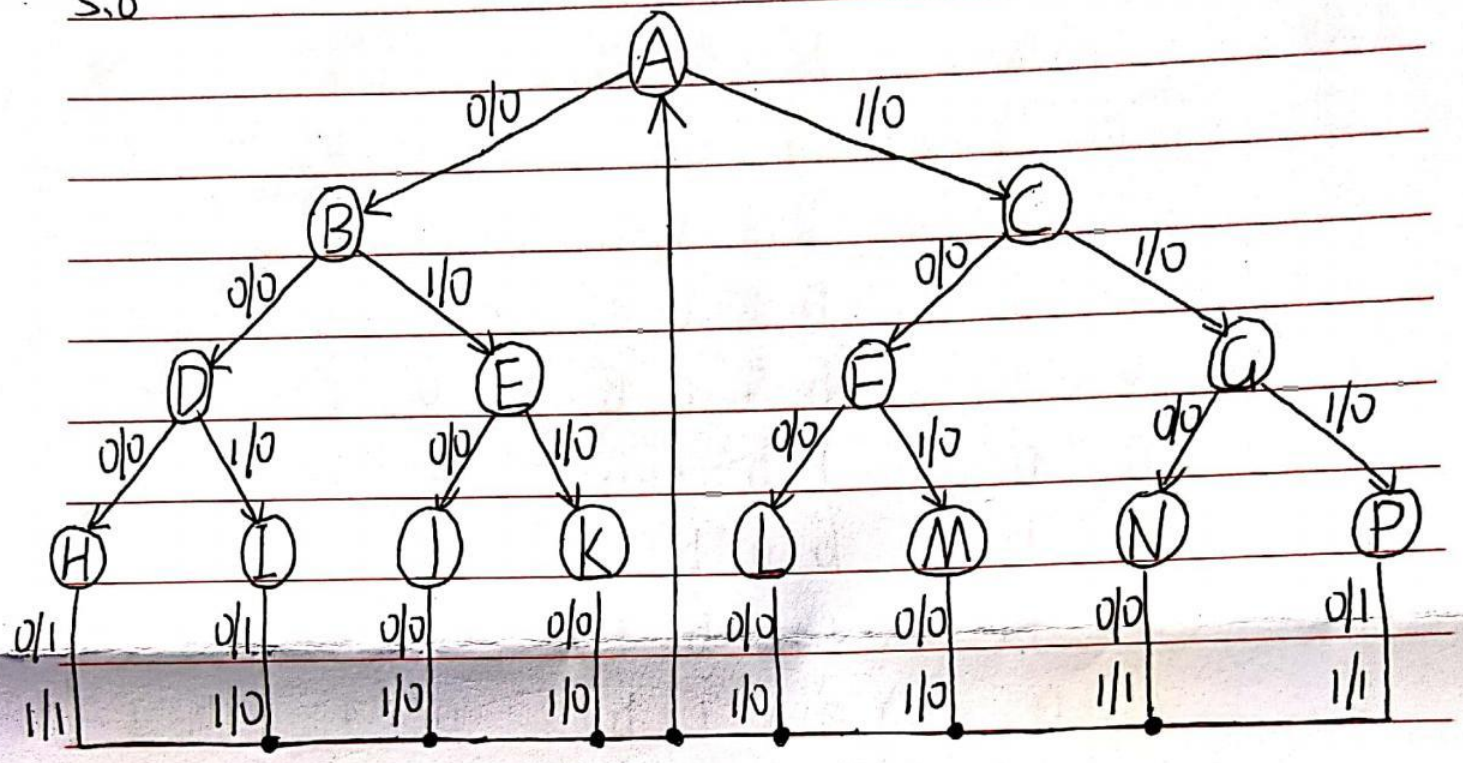


Moore型



武漢大學
WUHAN UNIVERSITY
Wuhan 430072, Hubei, P.R.China 中国·武汉 Tel.(027)

5.8



武漢大學
WUHAN UNIVERSITY
Wuhan 430072, Hubei, P.R.China 中国·武汉 Tel.(027)

5.12

输入	现态	激励函数				次态		输出
X	y_2	y_1	J_2	K_2	J_1	K_1	y_2^{n+1}	y_1^{n+1}	Z
0	0	0	0	d	1	d	0	1	0
0	0	1	1	d	d	0	1	1	0
0	1	0	d	0	0	d	1	0	1
0	1	1	d	1	d	1	0	0	1
1	0	0	1	d	0	d	1	0	0
1	0	1	1	d	d	1	1	0	0
1	1	0	d	1	1	d	0	1	0
1	1	1	d	0	d	0	1	1	1

y_1 \ Xy_2	00	01	11	10
0	0	d	d	1
1	1	d	d	1

J_2

$J_2 = X + y_1$

y_1 \ Xy_2	00	01	11	10
0	d	0	1	d
1	d	1	0	d

K_2

$K_2 = \bar{X}y_1 + X\bar{y}_1 = X \oplus y_1$

y_1 \ xy_2	00	01	11	10
0	1	0	1	0
1	d	d	d	d

J_1

$J_1 = x\bar{y}_2 + xy_2 = x \odot y_2$

y_1 \ xy_2	00	01	11	10
0	d	d	d	d
1	0	1	0	1

K_1

$K_1 = \bar{x}y_2 + x\bar{y}_2 = x \oplus y_2$

y_1 \ xy_2	00	01	11	10
0	0	1	0	0
1	0	1	1	0

Z

$Z = \bar{x}y_2 + y_1y_2$

∴激励函数：$J_2 = x + y_1$　　$K_2 = x \oplus y_1$

$J_1 = x \odot \bar{y}_2$　　$K_1 = x \oplus y_2$

输出函数：$Z = \bar{x}y_2 + y_1y_2$

2020302181062
崔浩然
数逻

武漢大學
WUHAN UNIVERSITY
Wuhan 430072, Hubei, P.R.China 中国·武汉 Tel.(027)

习题七

7.10

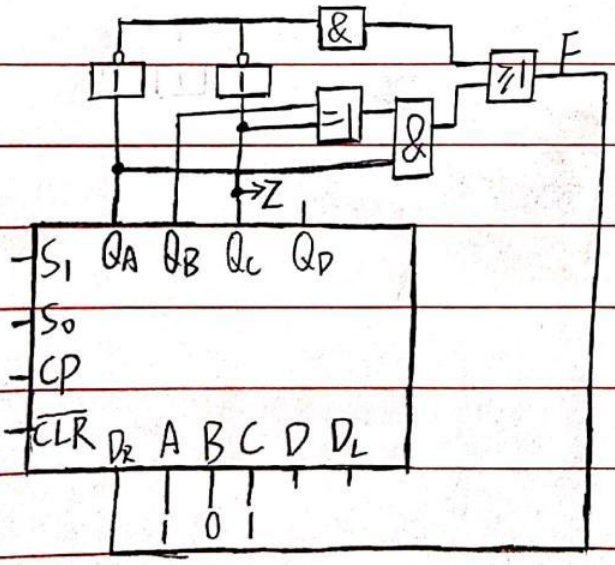
CP	F(DR)	QA	QB	QC
0	1	1	0	1
1	1	1	1	0
2	0	1	1	1
3	0	0	1	1
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	1	0	1	0

QC \ QAQB	00	01	11	10
0	1	1	1	0
1	0	0	0	1

F

$F = \bar{Q}_A\bar{Q}_C + Q_AQ_B\bar{Q}_C + Q_A\bar{Q}_BQ_C$

$= \bar{Q}_A\bar{Q}_C + Q_A(Q_B \oplus Q_C)$



在S_1S_0控制下，先置寄存器74194的初始状态为$Q_AQ_BQ_C=101$，然后令其工作在右移串行输入状态，即可在时钟脉冲下从Z端产生所需要的脉冲串序列。