**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

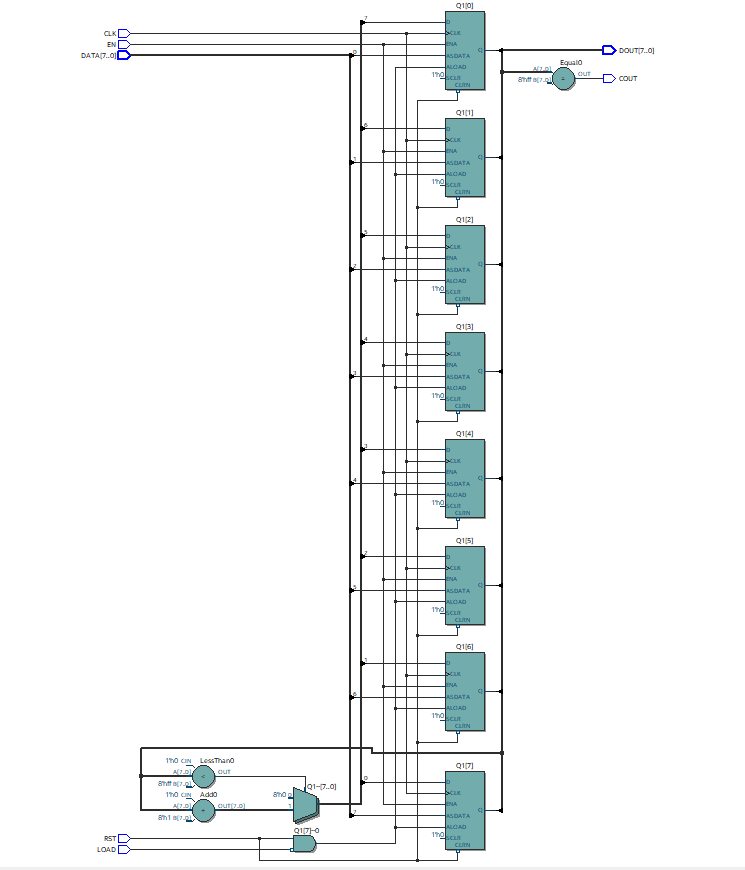
always @(Q1)

if (Q1==8'd255) COUT = 1'b1;

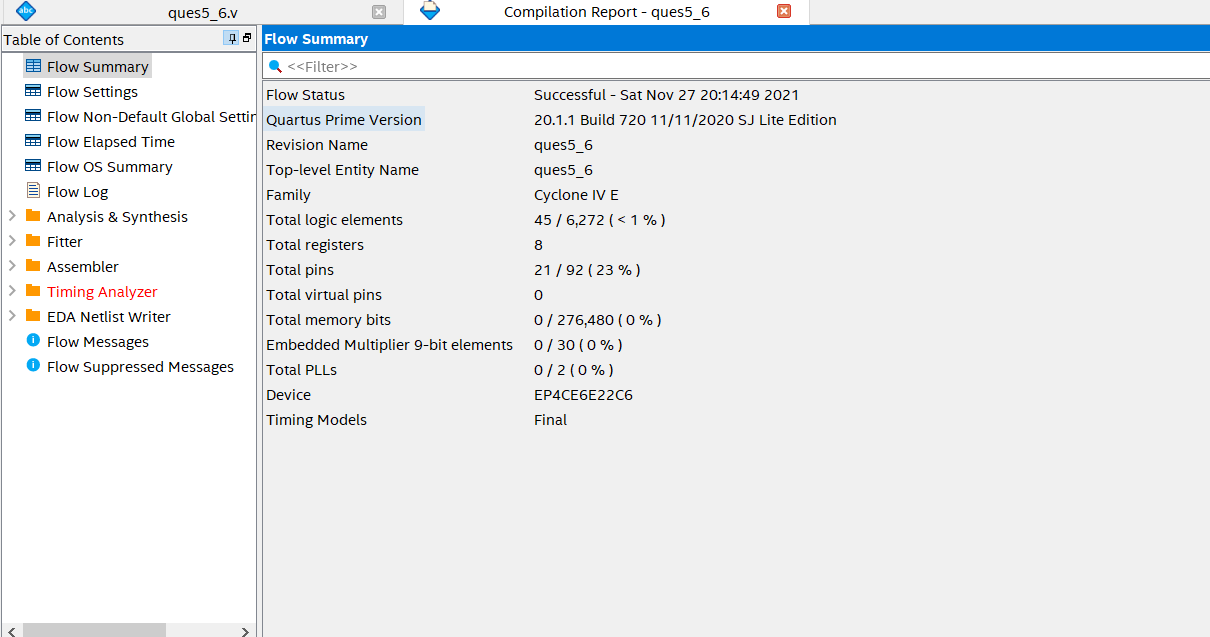
else COUT=1'b0;

Endmodule

**网表文件：**



**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

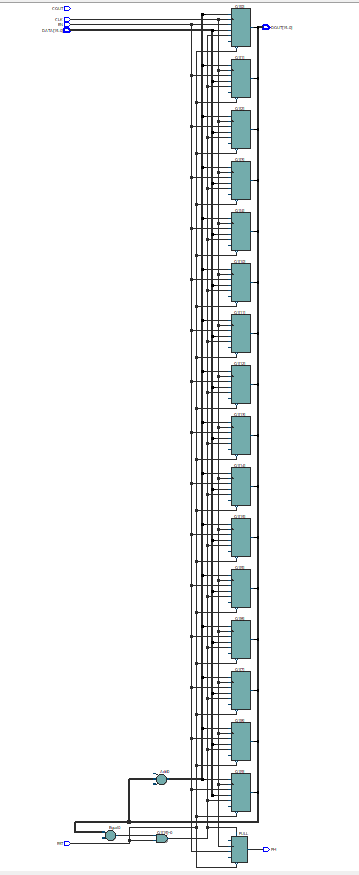
assign LOAD=(Q1==16'd16);

assign PM=FULL;

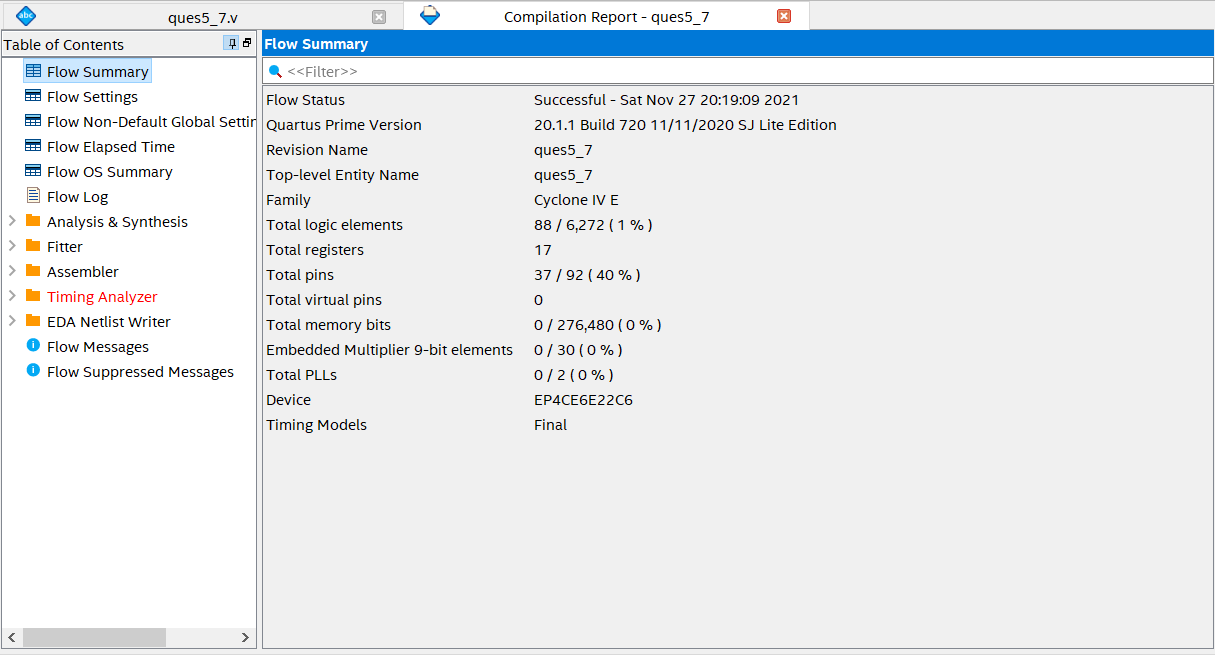
assign DOUT=Q1;

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

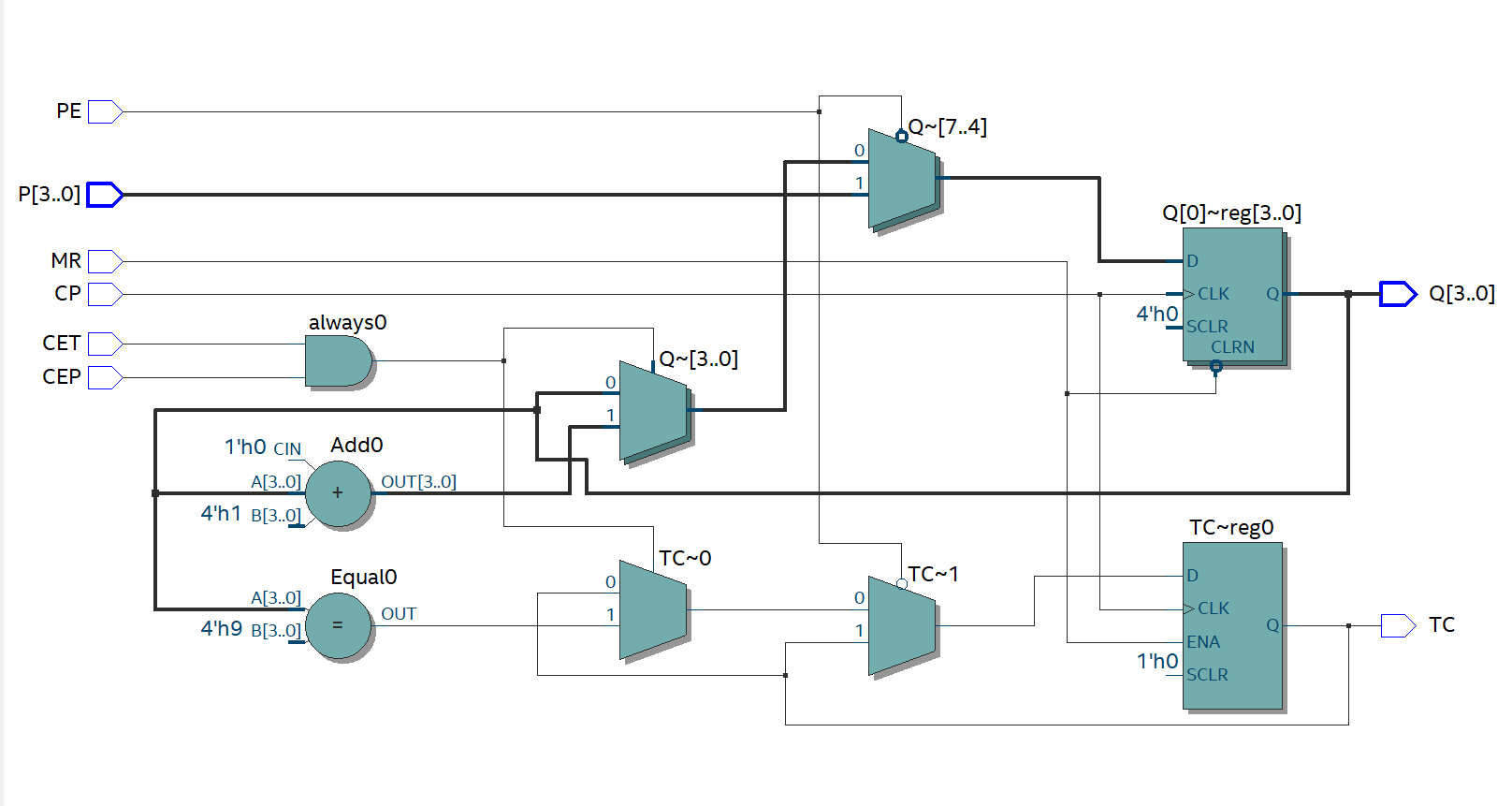
Q<=Q;

end

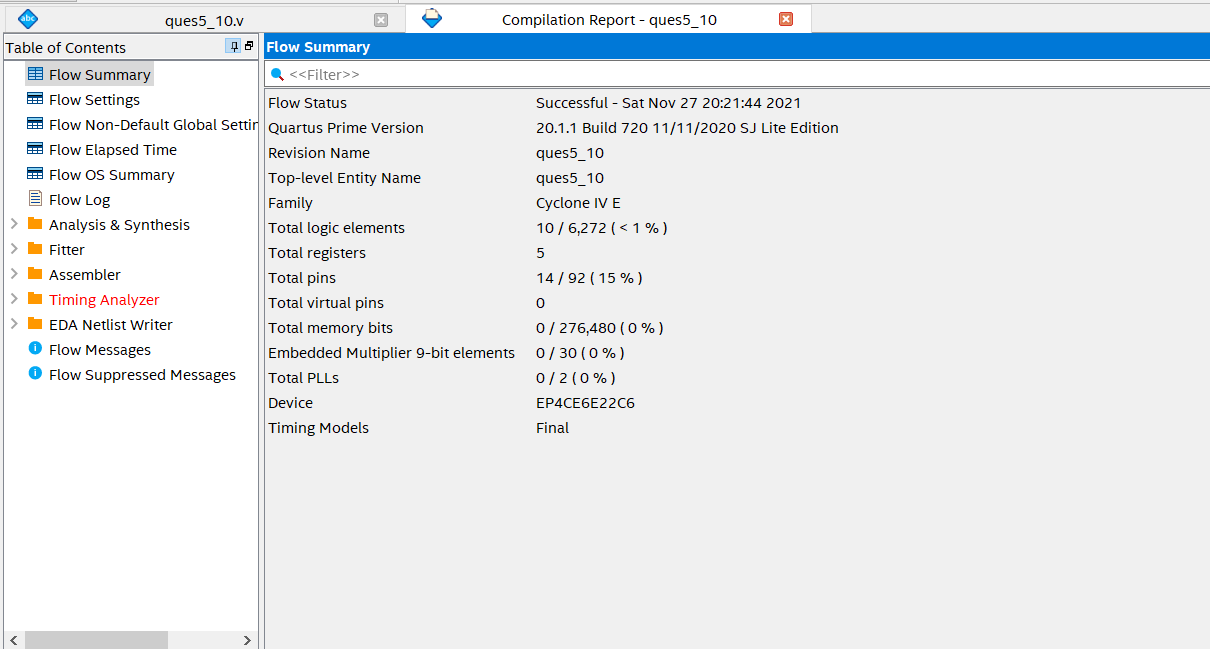
end

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

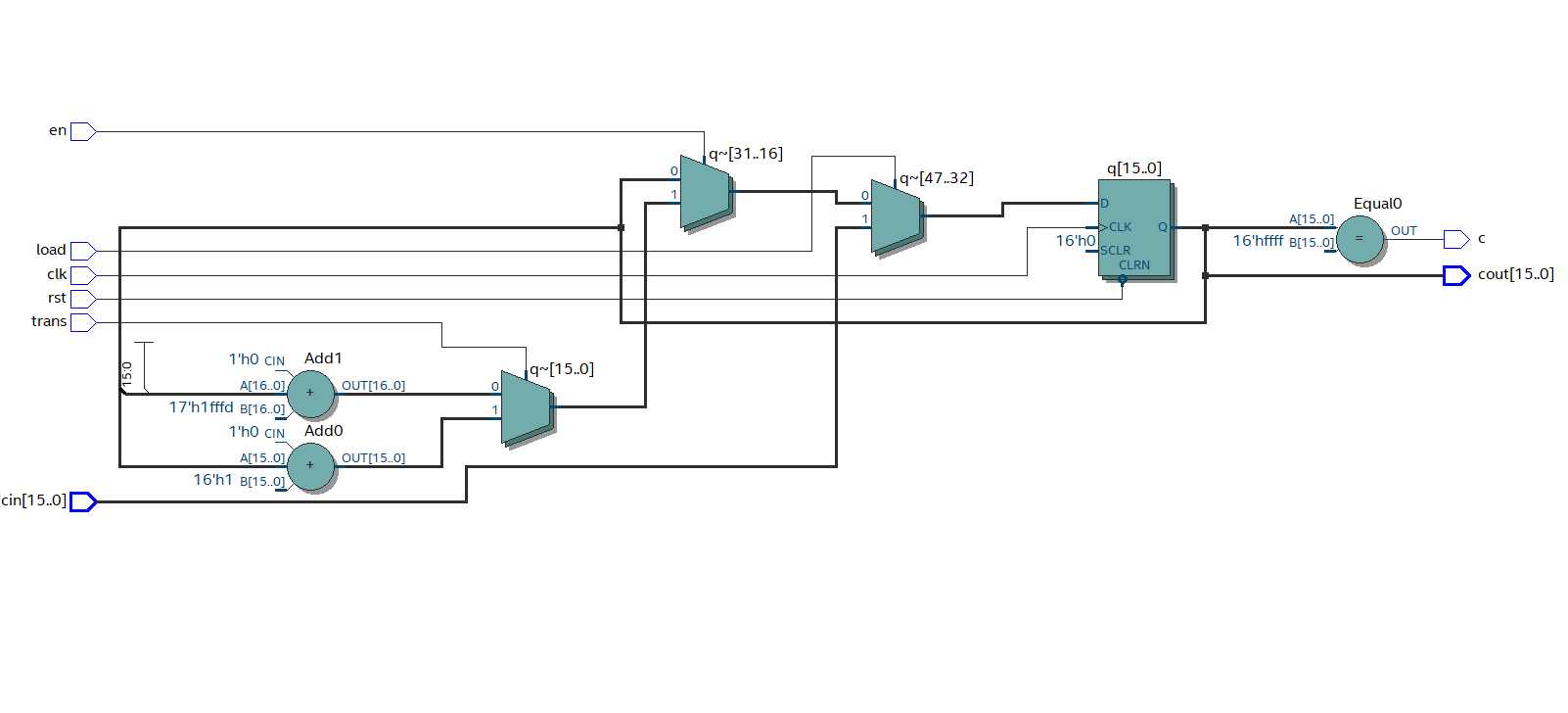
c=1'b0;

end

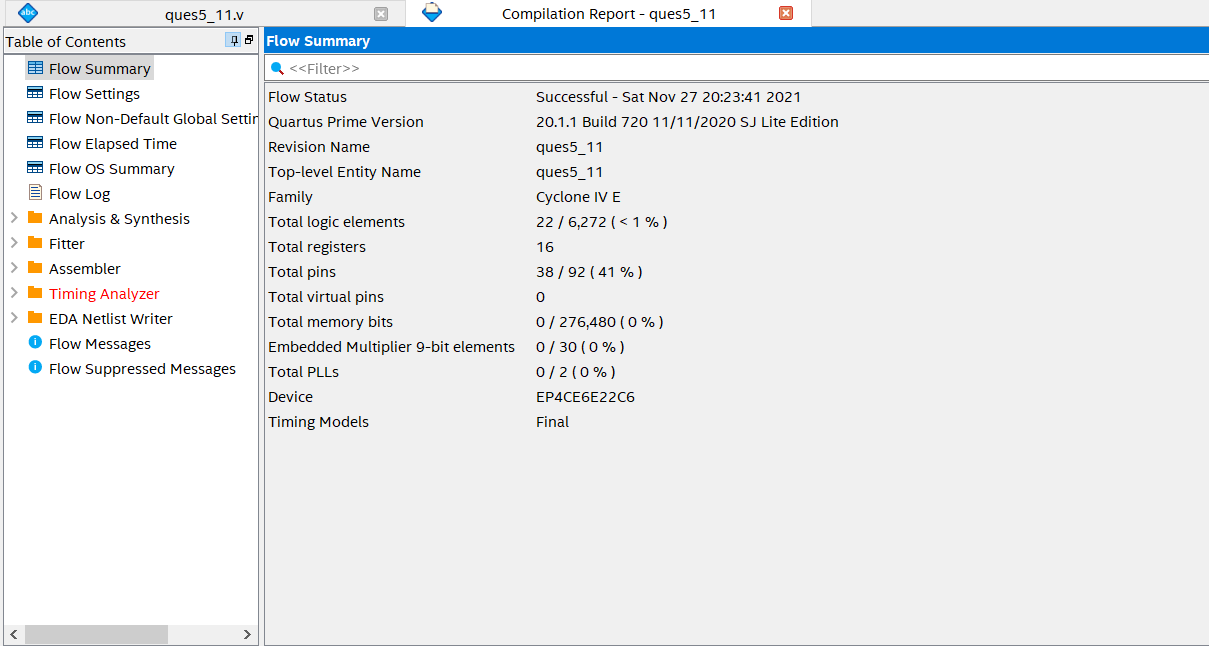
assign cout=q;

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;

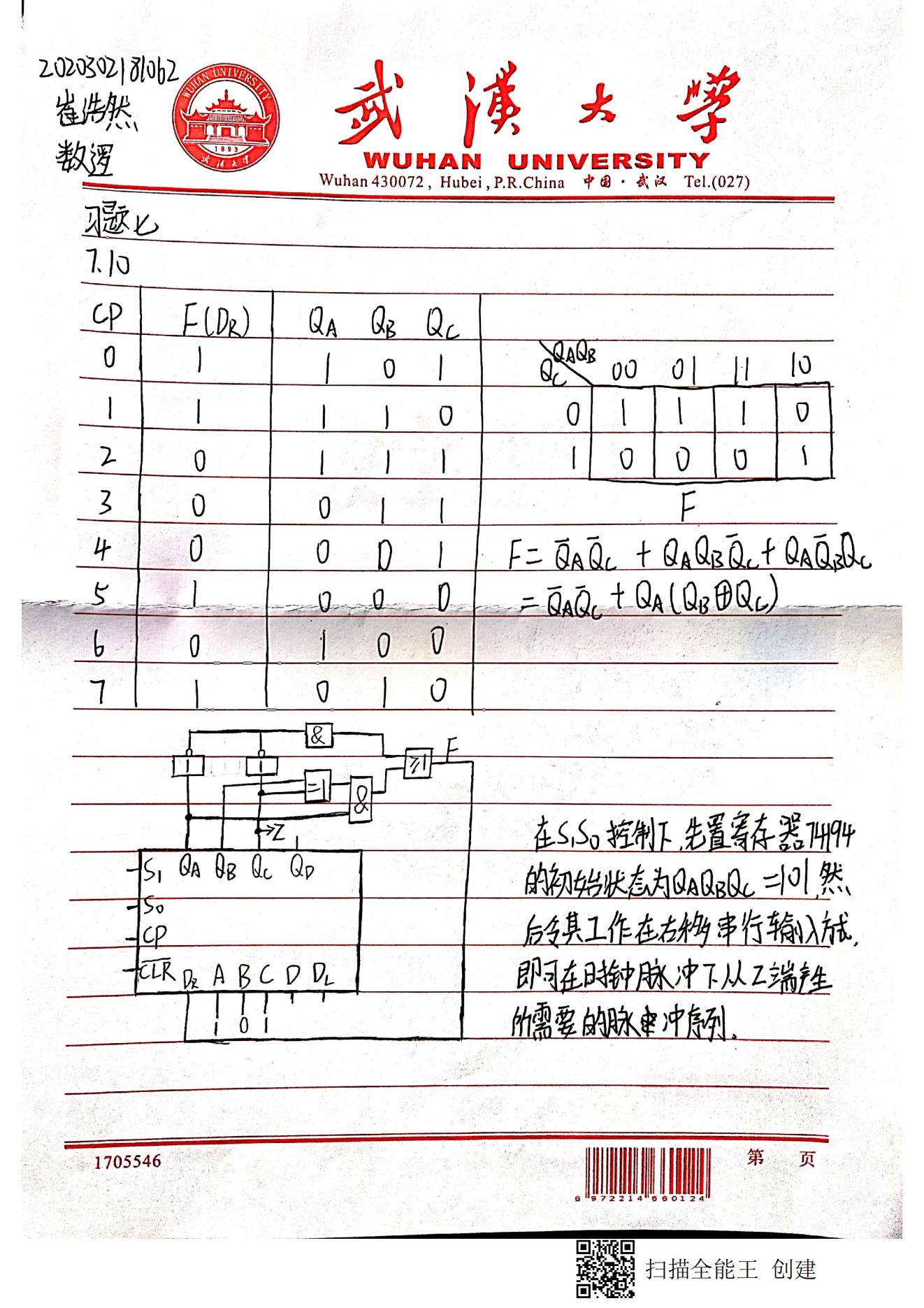
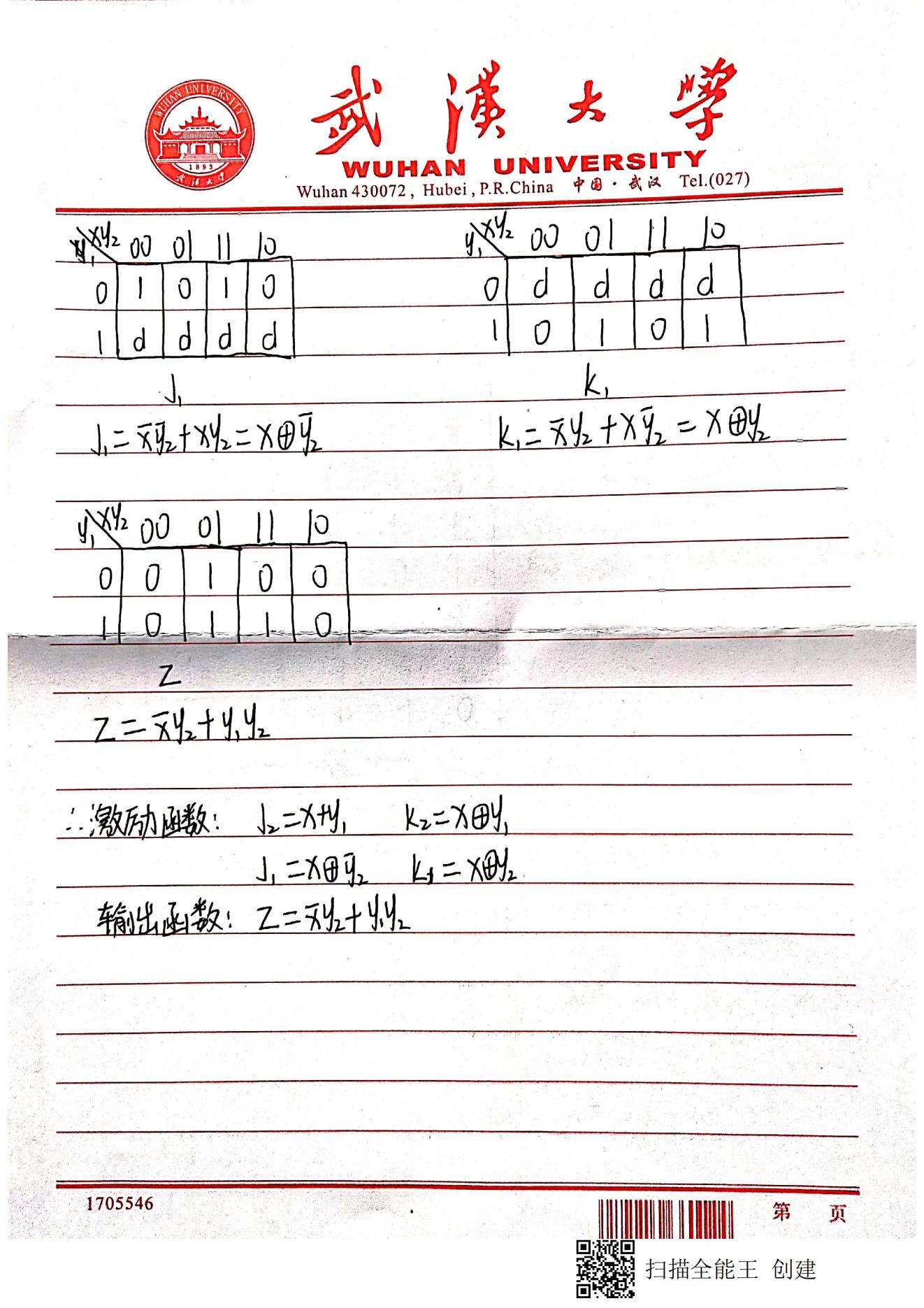
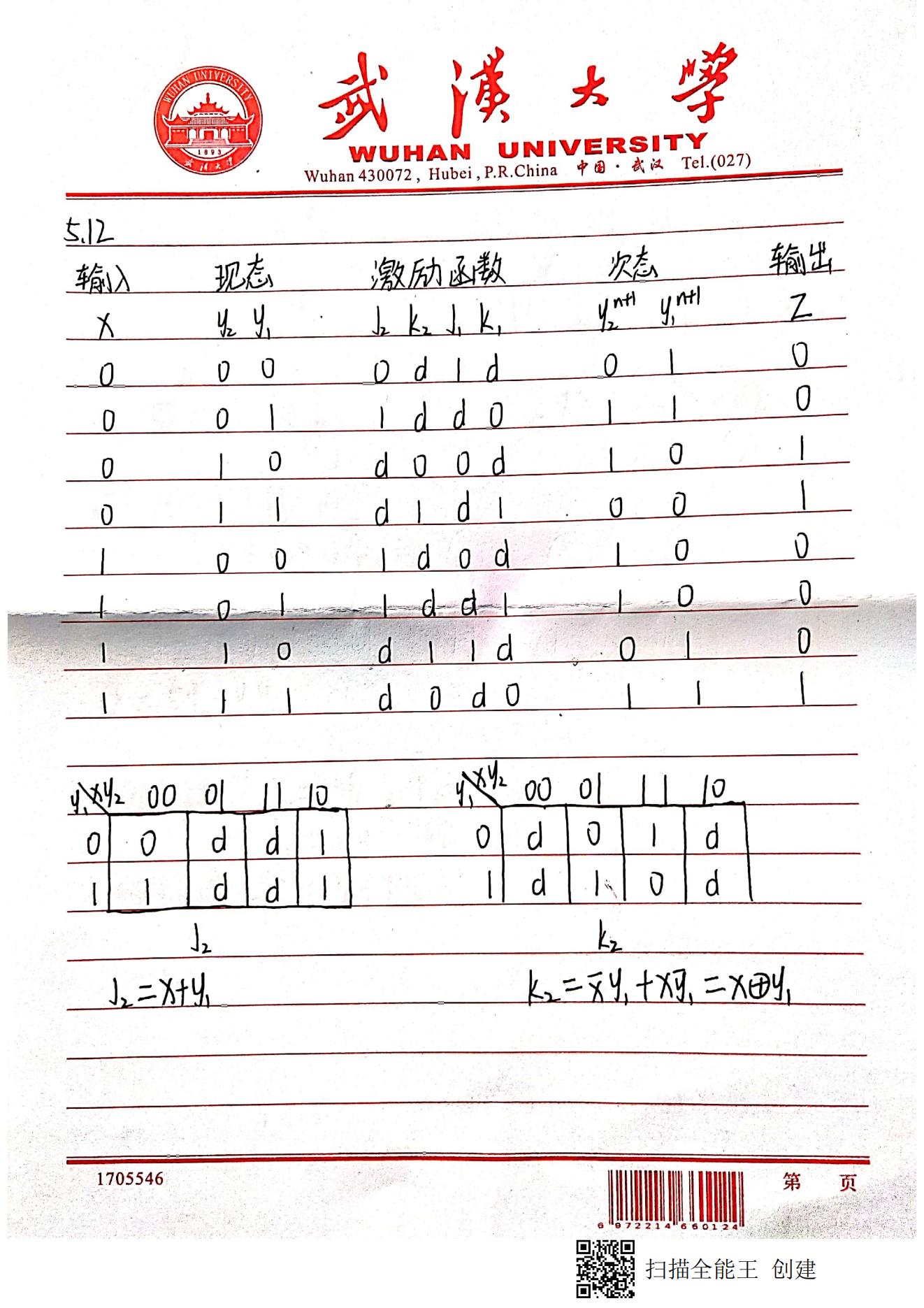
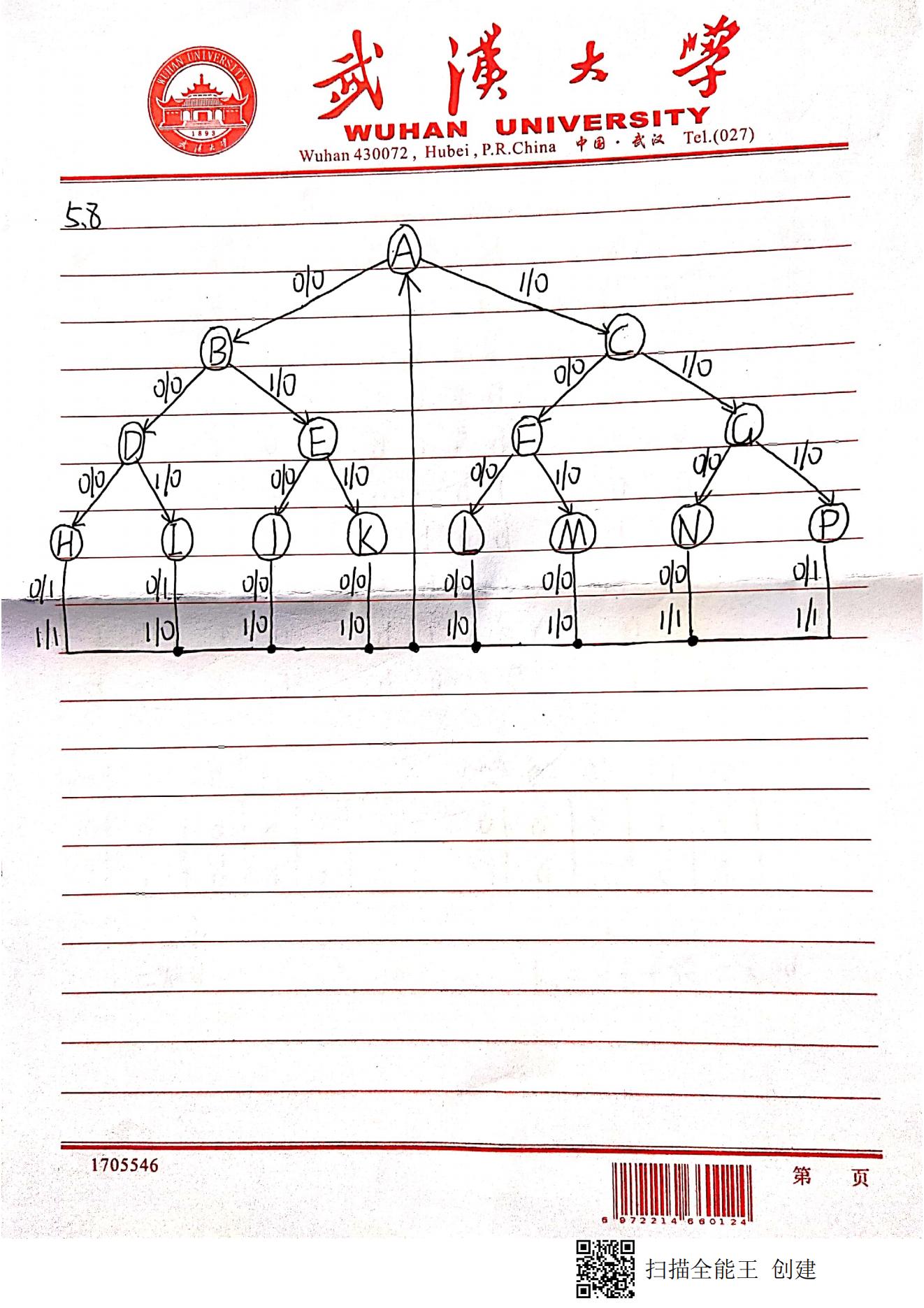
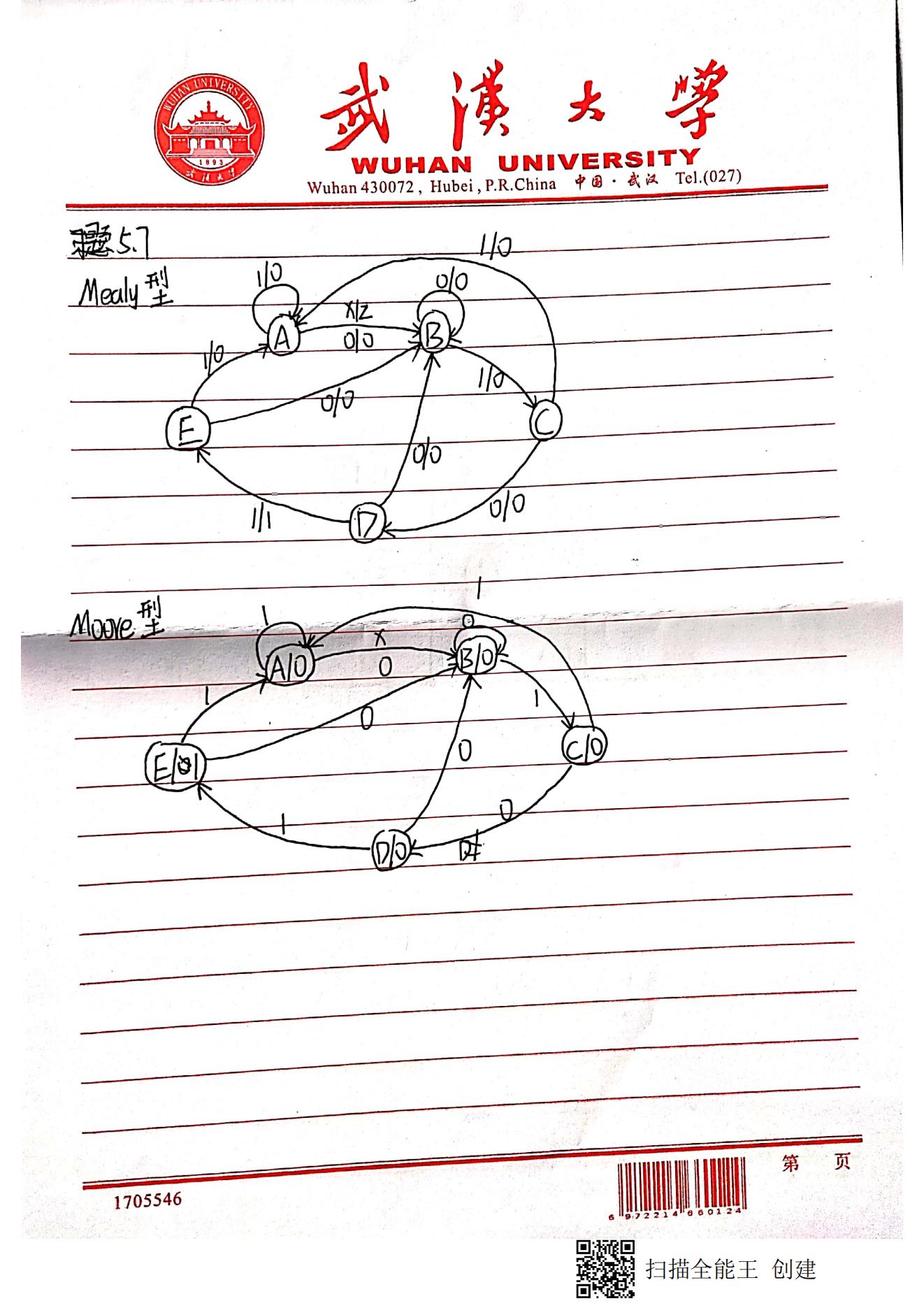
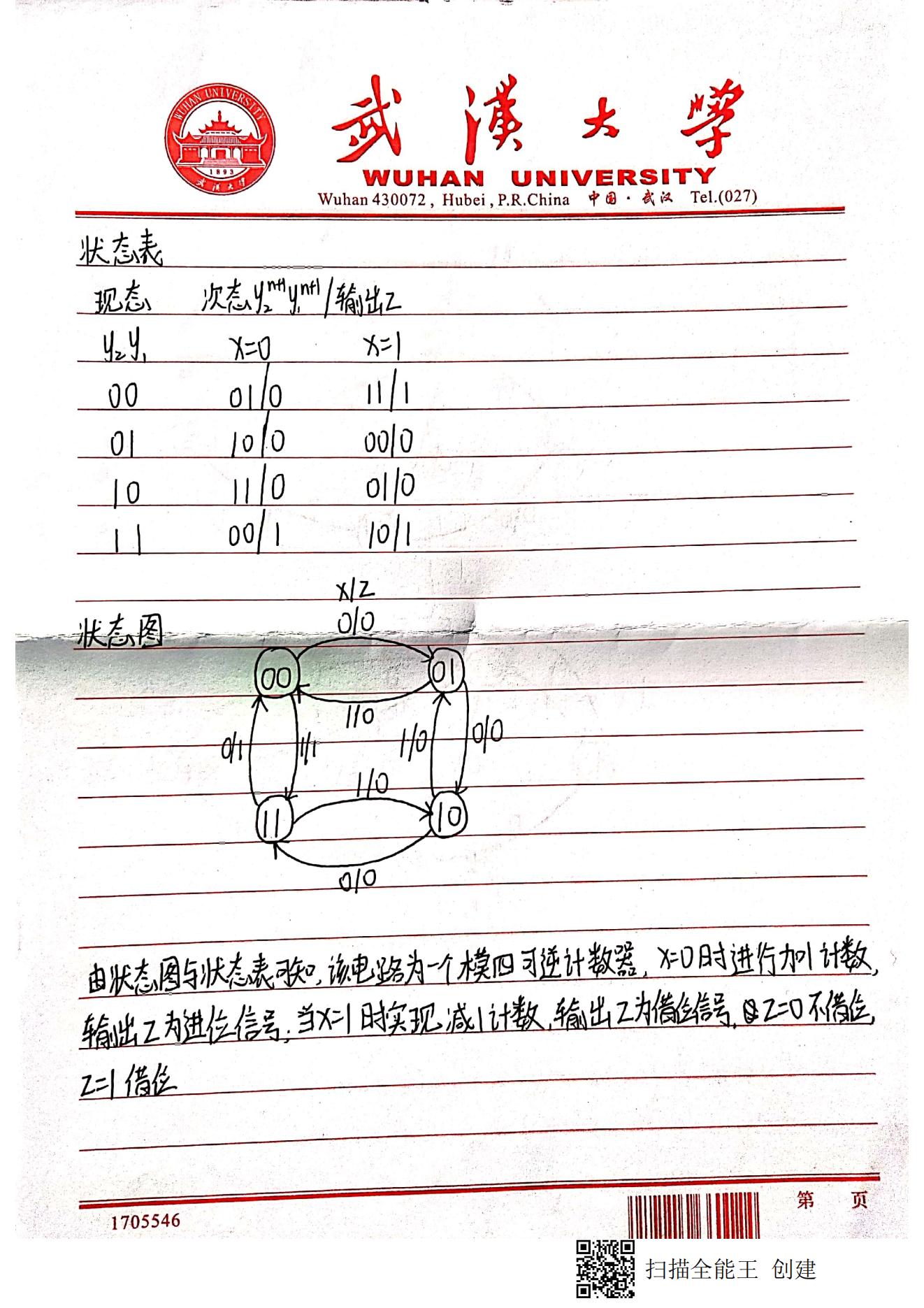
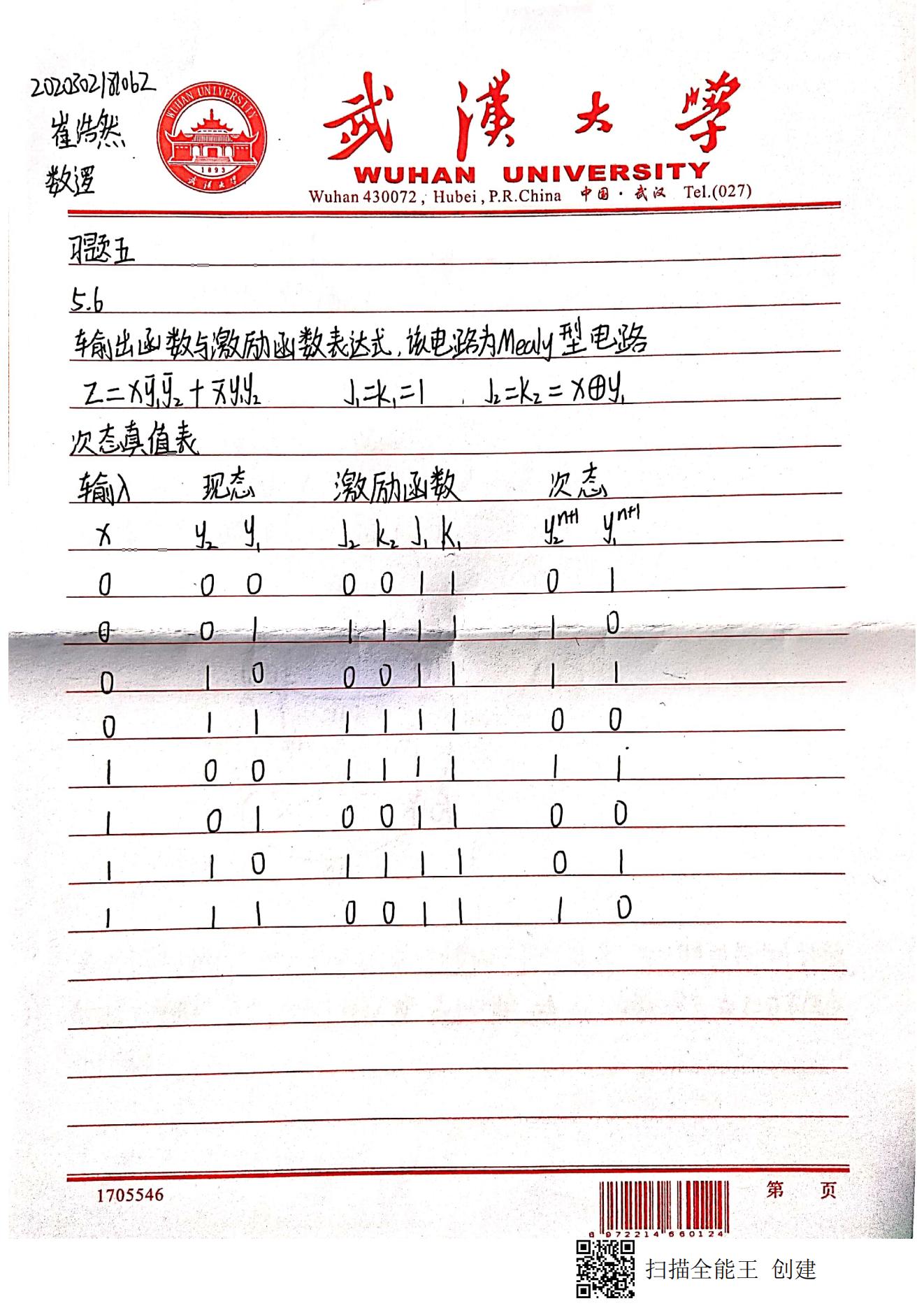
//@eachvec;

// --> end

end

Endmodule

**数字逻辑作业**



**第一页**

**第二页**

**第三页**

**第四页**

**第五页**

**第六页**

**第七页**