

**本科实验报告**

课程名称： 计算机组成原理实验

课程编号： 08060038

学生姓名： 邝庆璇

学号： 2016051598

学院： 信息科学与技术学院

系： 计算机系

专业： 计算机科学与技术

指导教师： 王传胜 梁倬骞

教师单位： 暨南大学计算机系

开课时间：2019~ 2020学年度 第1期

**暨南大学教务处**

**2019年12月05日**

数字逻辑实验 **课程实验项目目录**

学生姓名：邝庆璇 学号：2016051598

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 实验项目编号 | 实验项目名称 | \*实验项目类型 | 成绩 | 指导教师 |
| 1 | **0806003801** | **验证逻辑门** | **验证** |  | **王传胜 梁倬骞** |
| 2 | **0806003802** | **基本逻辑门电路之间的转换** | **验证** |  | **王传胜 梁倬骞** |
| 3 | **0806003803** | **超前进位全加器** | **验证** |  | **王传胜 梁倬骞** |
| 4 | **0806003804** | **译码器** | **验证** |  | **王传胜 梁倬骞** |
| 5 | **0806003805** | **数字比较器** | **验证** |  | **王传胜 梁倬骞** |
| 6 | **0806003806** | **七段数码管** | **验证** |  | **王传胜 梁倬骞** |
| 7 | **0712000807** | **触发器** | **验证** |  | **王传胜 梁倬骞** |
| 8 | **0712000808** | **计数器** | **验证** |  | **王传胜 梁倬骞** |
| 9 | **0712000809** | **寄存器** | **验证** |  | **王传胜 梁倬骞** |
| 10 | **0712000810** | **时序逻辑电路综合设计** | **验证** |  | **王传胜 梁倬骞** |
| 11 | **0712000811** | **VGA控制器** |  |  | **王传胜 梁倬骞** |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |
| 17 |  |  |  |  |  |
| 18 |  |  |  |  |  |

\*实验项目类型：演示性、验证性、验证性、设计性实验。

\*此表由学生按顺序填写。

**暨南大学本科实验报告专用纸**

课程名称 数字逻辑实验 成绩评定

实验项目名称 Verilog门级建模 指导教师 王传胜 梁倬骞

实验项目编号 0806003802 实验项目类型 验证 实验地点 N126

学生姓名 邝庆璇 学号 2016051598

学院 信息科学技术 系 计算机科学 专业 计算机科学与技术

实验时间 2019 年 9 月 30 日 上 午

一、实验目的

1、学会使用Verilog HDL进行门级建模；

2、初步学会怎么用Verilog HDL编写仿真程序；

3、初步学会怎么使用Vivado软件；

4、初步学会验证基本逻辑门的逻辑功能的方法；

5、初步学会验证怎么使用EG0-1实验板。

二、实验内容

编写Verilog程序，验证3输入的与门、与非门、或门、或非门、异或门、同或门

三、实验程序

exp1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/09/30 11:28:50

// Design Name:

// Module Name: exp-1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module exp1(

input in1,

input in2,

input in3,

output out1,

output out2,

//output out3,

output out4,

output out5,

output out6,

output out7

);

and funcAnd(out1, in1, in2, in3);

or funcOr(out2, in1, in2, in3);

xor funcXor(out4, in1, in2, in3);

nand funcNand(out5, in1, in2, in3);

nor funcNor(out6, in1, in2, in3);

xnor funcXnor(out7, in1, in2, in3);

endmodule

四、仿真程序

sim\_1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/09/30 11:57:26

// Design Name:

// Module Name: sim\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module sim\_1(

output out1,

output out2,

//output out3,

output out4,

output out5,

output out6,

output out7

);

reg[2:0] in;

initial

begin

in = 3'b00;

end

always #10

begin

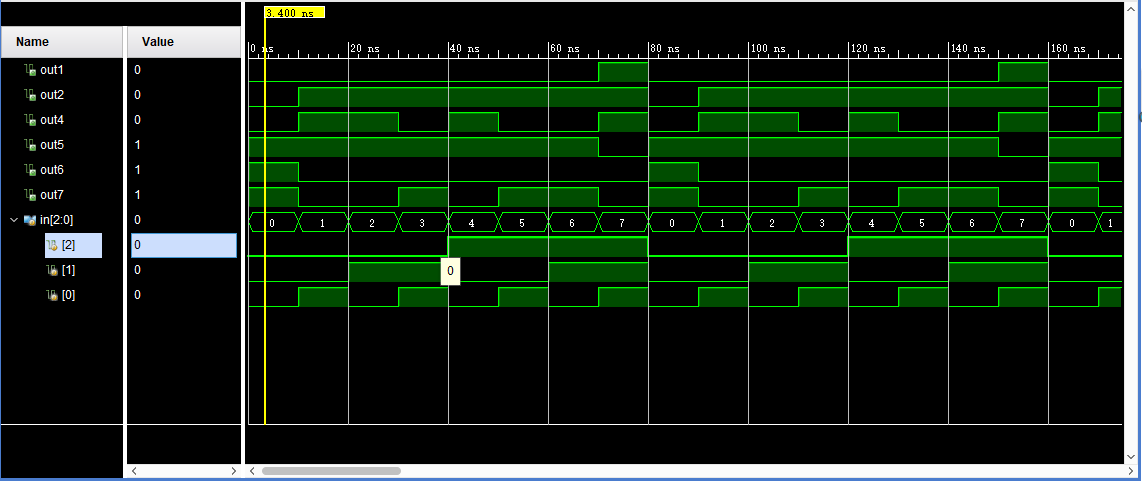
in = in + 1;

end

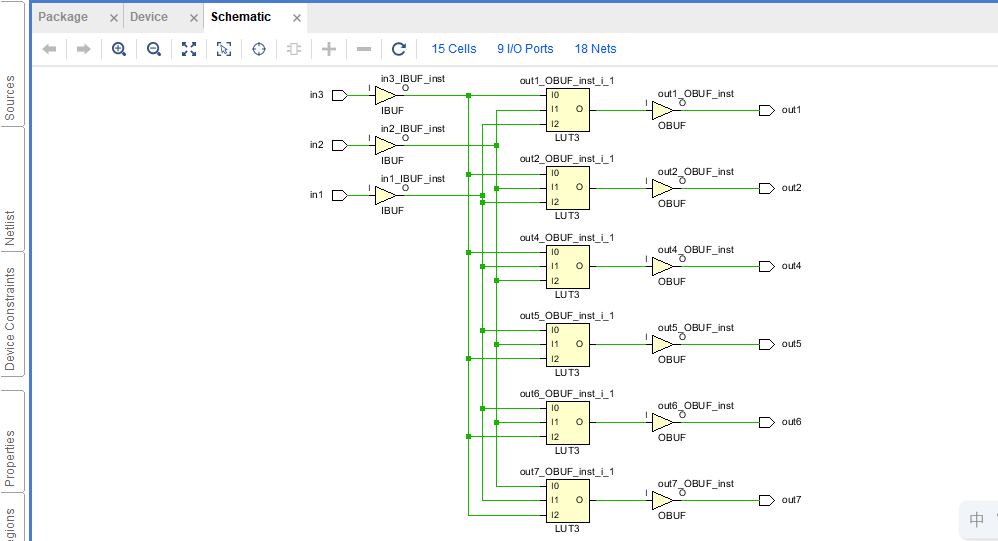
exp1 exp1(in[2],in[1], in[0], out1, out2, out4, out5, out6, out7);

endmodule

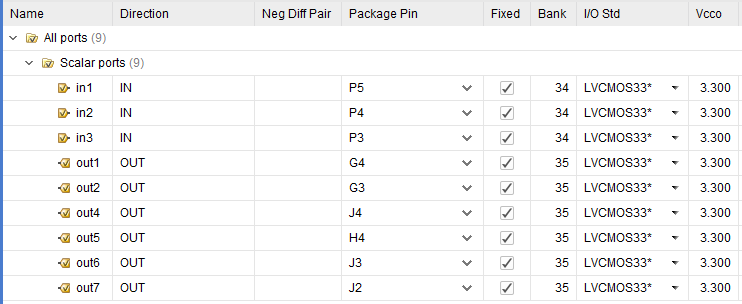
五、仿真结果



六、系统网表



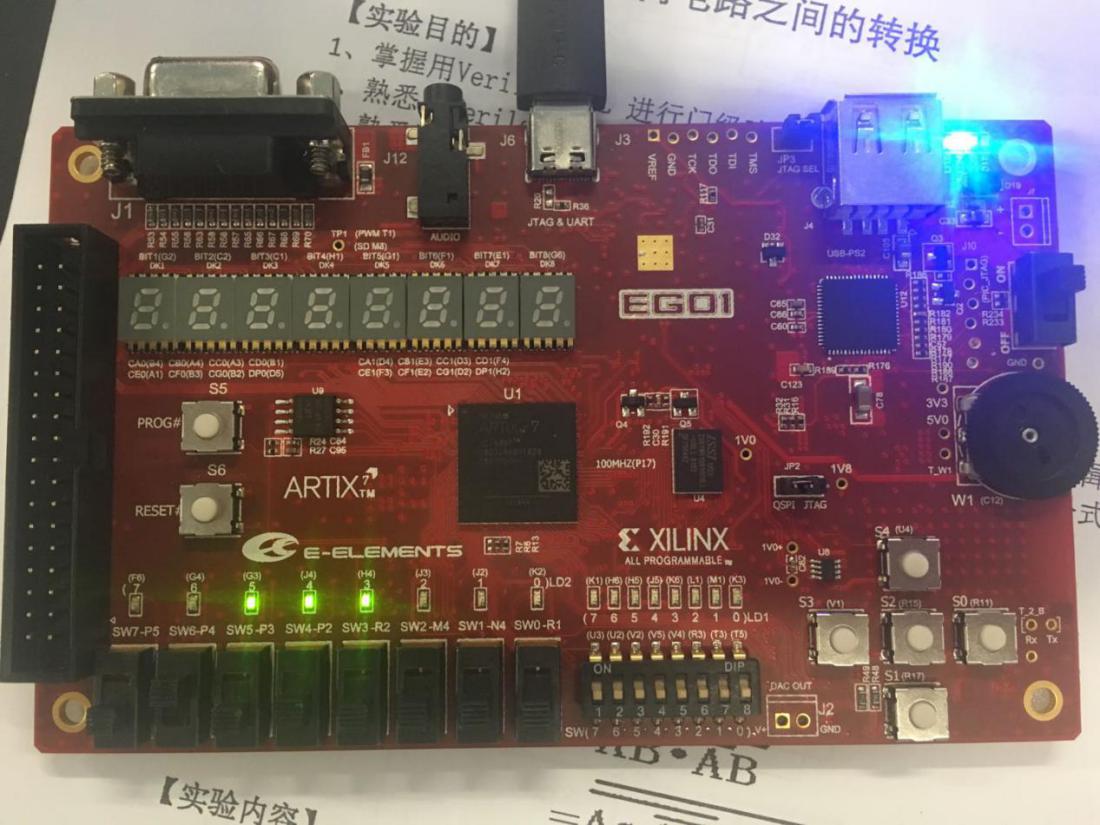
七、系统约束



八、实验结果

示例：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输入 | In1 | | In2 | | In3 | | |
| 输入端口 | P5 | | P4 | | P3 | | |
| 输入的值 | 0 | | 2 | | 0 | | |
| 输出 | out1 | out2 | out4 | out5 | | out6 | out7 |
| 输出端口 | G4 | G3 | J4 | H4 | | J3 | J2 |
| 逻辑运算 | And | Or | Xor | Nand | | Nor | xnor |
| 输出的值 | 0 | 1 | 1 | 1 | | 0 | 0 |



九、实验体会

此次试验属于初步的实践，学习了门级建模，熟悉了vivado软件的使用及verilog编程的语法，有效地为数字逻辑试验和学习作准备。感悟：刚开始使用时，使用软件进行建模的流程相对繁琐，但熟悉之后会比较顺利，毕竟初期的实验难度不大。

**暨南大学本科实验报告专用纸**

课程名称 数字逻辑实验 成绩评定

实验项目名称 基本逻辑门电路之间的转换 指导教师 王传胜 梁倬骞

实验项目编号 0806003802 实验项目类型 验证 实验地点 N126

学生姓名 邝庆璇 学号 2016051598

学院 信息科学技术 系 计算机科学 专业 计算机科学与技术

实验时间 2019 年 10 月 14 日 上 午

一、实验目的

掌握用Verilog HDL 进行门级建模；

熟悉用Verilog HDL写仿真程序；

熟悉使用Vivado软件；

学会验证基本逻辑门电路之间的转换方法；

熟悉使用EGO-1实验板。

二、实验内容

用与非门（NAND）来实现异或（XOR）运算

三、实验程序

exp1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/10/14 12:11:56

// Design Name:

// Module Name: exp\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module exp\_1(

input in1,

input in2,

output out

);

nand (temp1, in1, in2);

nand (temp2, in1, temp1);

nand (temp3, in2, temp1);

nand (out, temp2, temp3);

endmodule

四、仿真程序

sim\_1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/10/14 12:22:02

// Design Name:

// Module Name: sim\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module sim\_1(

output out

);

reg[1:0] in;

initial

begin

in = 3'b00;

end

always #10

begin

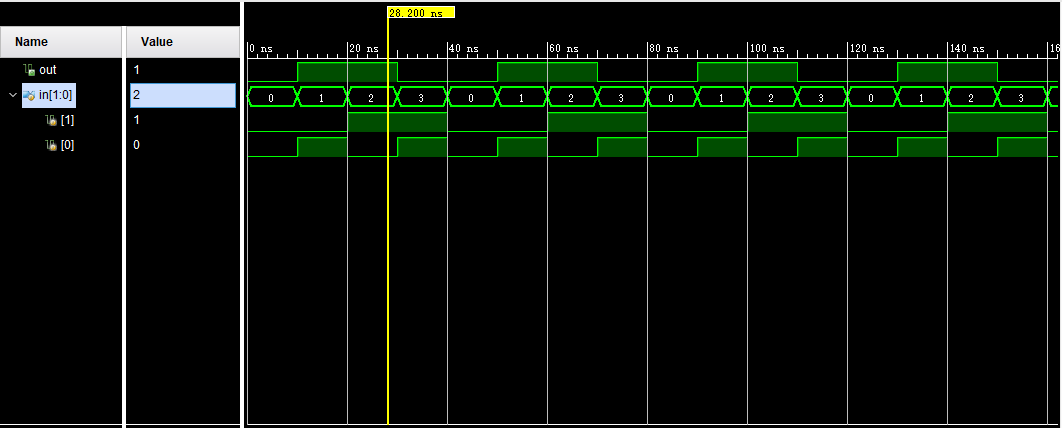
in = in + 1;

end

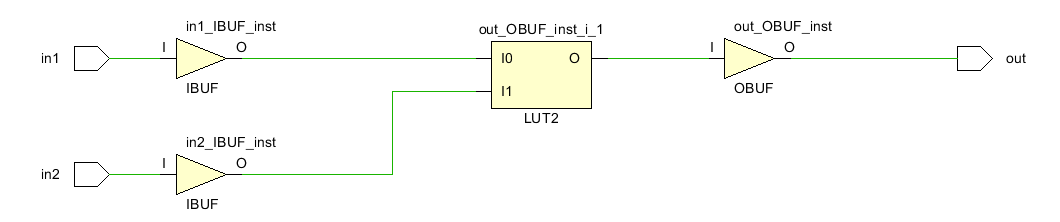
exp\_1 exp\_1(in[1], in[0], out);

Endmodule

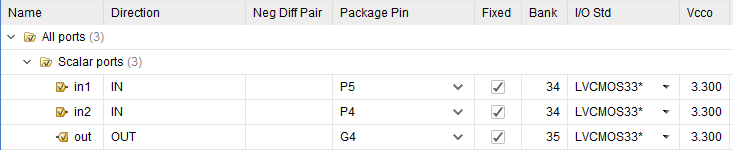
五、仿真结果



六、系统网表



七、系统约束



八、实验结果

示例：

|  |  |  |
| --- | --- | --- |
| 输入 | In1 | In2 |
| 输入端口 | P5 | P4 |
| 输入的值 | 0 | 1 |
| 输出 | out | |
| 输出端口 | G4 | |
| 输出的值 | 1 | |

**截图见下页**

**/////////////////////////////// TODO ! ///////////////////////////////////////**

九、实验体会

通过此次试验，进一步地联系了门级建模，也更加熟悉vivado的编程及使用流程（包括模拟simulation，综合synthesis，根据硬件设备指定I/O Ports，生产比特流generate bitstream，到最后把程序放到设备里面），还有对每一步结果分析。

**暨南大学本科实验报告专用纸**

课程名称 数字逻辑实验 成绩评定

实验项目名称 超前进位加法器 指导教师 王传胜 梁倬骞

实验项目编号 0806003803 实验项目类型 验证 实验地点 N126

学生姓名 邝庆璇 学号 2016051598

学院 信息科学技术 系 计算机科学 专业 计算机科学与技术

实验时间 2019 年 10 月 21 日 上 午

一、实验目的

学会用Verilog HDL 进行数据流建模；

掌握用Verilog HDL 写仿真程序；

掌握使用Vivado 软件；

学会超前进位全加器的原理；

掌握使用EGO-1实验板。

二、实验内容

用Verilog HDL 实现3位的超前进位加法器。

\*注意：必须使用超前进位实现！！！不能{c4,sum}=a+b+c0

三、实验程序

exp1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/10/21 12:43:16

// Design Name:

// Module Name: exp\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module exp\_1(

input [2:0] A,

input [2:0] B,

input C\_in,

output C\_out,

output [2:0] S

);

wire [2:0] G = A & B;

wire [2:0] P = A ^ B;

// 2为高位，0为低位, 210

wire C1, C2; // 从1开始的A1+B1, A2+B2所产生的进位

assign C1 = G[0] + (P[0] & C\_in); // 记得要加括号呀！

assign C2 = G[1] + (P[1] & C1);

assign C\_out = G[2] + (P[2] & C2);

assign S[0] = P[0] ^ C\_in;

assign S[1] = P[1] ^ C1;

assign S[2] = P[2] ^ C2;

endmodule

四、仿真程序

sim\_1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/10/21 12:58:48

// Design Name:

// Module Name: sim\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module sim\_1(

output C\_out,

output [2:0] S

);

reg [5:0] in;

reg [2:0] A;

reg [2:0] B;

reg inc;

initial

begin

in = 6'b000000;

assign A = in[5:3];

assign B = in[2:0];

assign inc = 1;

end

always #10

begin

in = in + 1;

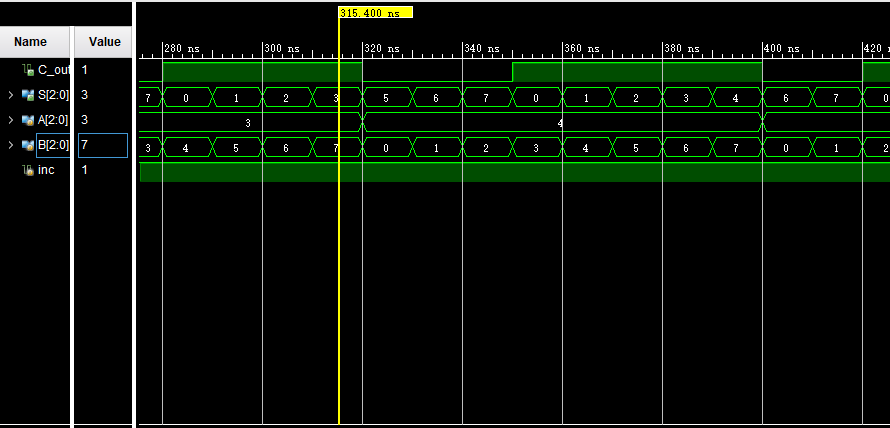
inc = inc + 1;

end

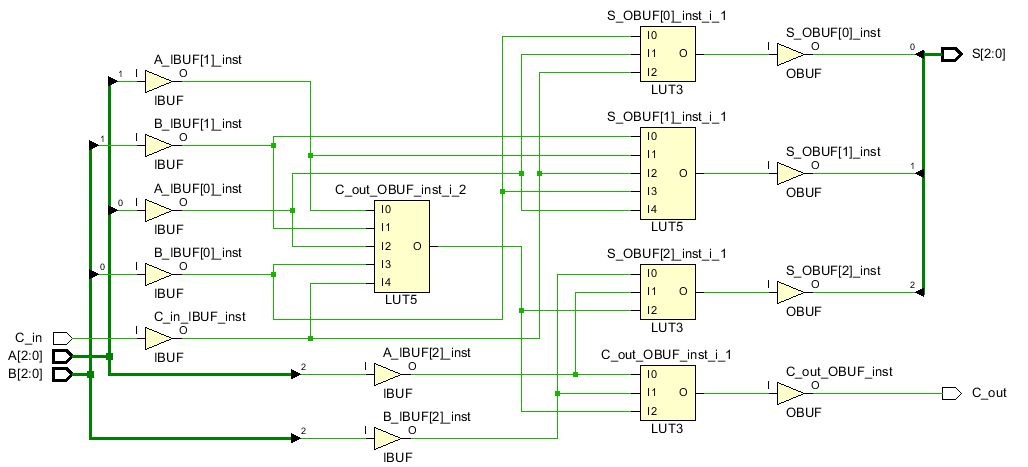
exp\_1 exp\_1(A, B, inc, C\_out, S);

endmodule

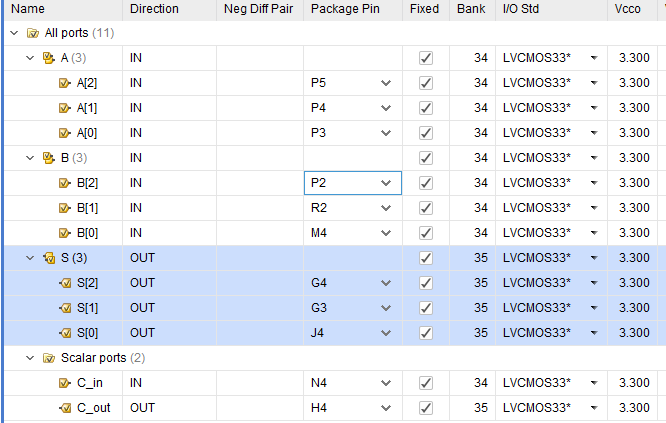
五、仿真结果



六、系统网表



七、系统约束



八、实验结果

示例（3+7+1=11=8+3）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入 | A (A2 A1 A0) | B (B2 B1 B0) | | C\_in |
| 输入端口 | P5 P4 P3 | P2 R2 M4 | | N4 |
| 输入的值 | 0 1 1 | 1 1 1 | | 1 |
| 输出 | S (S2 S1 S0) | | C\_out | |
| 输出端口 | G4 G3 J4 | | H4 | |
| 输出的值 | 0 1 1 | | 1 | |

**////////////////////////////////////////// TODO! //////////////////////////////////////////////////**

九、实验体会

通过此次试验，不但学习并练习了Verilog HDL进行数据流建模，熟悉了Verilog的各种操作符，而且还复习了超前进位全加器的原理。诚然，在编码过程当中，仍存在对Verilog语法及Vivado软件使用不熟练，导致走了一点弯路（比如没意识到实验程序exp1.v不能用while, for等语句，仿真文件里才能用），但在请教老师同学后迅速解决了问题，并完成了实验。

**暨南大学本科实验报告专用纸**

课程名称 数字逻辑实验 成绩评定

实验项目名称 译码器 指导教师 王传胜 梁倬骞

实验项目编号 0806003804 实验项目类型 验证 实验地点 N126

学生姓名 邝庆璇 学号 2016051598

学院 信息科学技术 系 计算机科学 专业 计算机科学与技术

实验时间 2019 年 10 月 28 日 上 午

一、实验目的

掌握用 Verilog HDL 进行数据流建模；

深入理解译码器的原理；

学会用 Verilog HDL 实现74LS138；

二、实验内容

用Verilog HDL 语言实现74LS138的功能。

三、实验程序

exp1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/10/28 12:13:17

// Design Name:

// Module Name: exp\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module exp\_1(

input [2:0] G,

input [2:0] A,

output [7:0] Y

);

wire E\_tmp = (G[0] & ~(G[1]) & ~(G[2]));

wire A0\_not = ~A[0];

wire A1\_not = ~A[1];

wire A2\_not = ~A[2];

assign Y[0] = ~(A0\_not & A1\_not & A2\_not & E\_tmp);

assign Y[1] = ~(A[0] & A1\_not & A2\_not & E\_tmp);

assign Y[2] = ~(A0\_not & A[1] & A2\_not & E\_tmp);

assign Y[3] = ~(A[0] & A[1] & A2\_not & E\_tmp);

assign Y[4] = ~(A0\_not & A1\_not & A[2] & E\_tmp);

assign Y[5] = ~(A[0] & A1\_not & A[2] & E\_tmp);

assign Y[6] = ~(A0\_not & A[1] & A[2] & E\_tmp);

assign Y[7] = ~(A[0] & A[1] & A[2] & E\_tmp);

endmodule

四、仿真程序

sim\_1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/10/28 12:16:20

// Design Name:

// Module Name: sim\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module sim\_1(

output [7:0] Y

);

reg [2:0] G;

reg [2:0] A;

initial

begin

assign A = 3'b000;

assign G = 3'b001;

end

always #20

begin

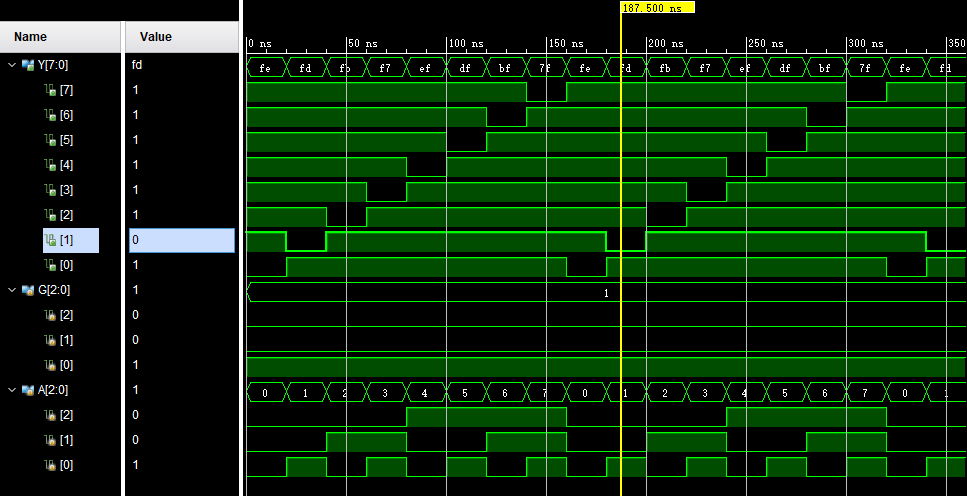
assign A = A + 1;

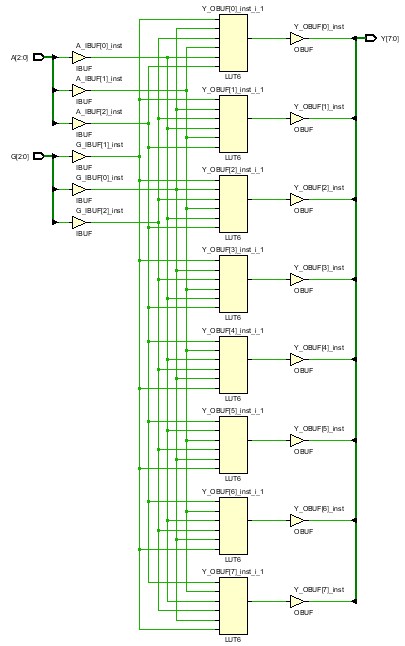
end

exp\_1 exp\_1(G, A, Y);

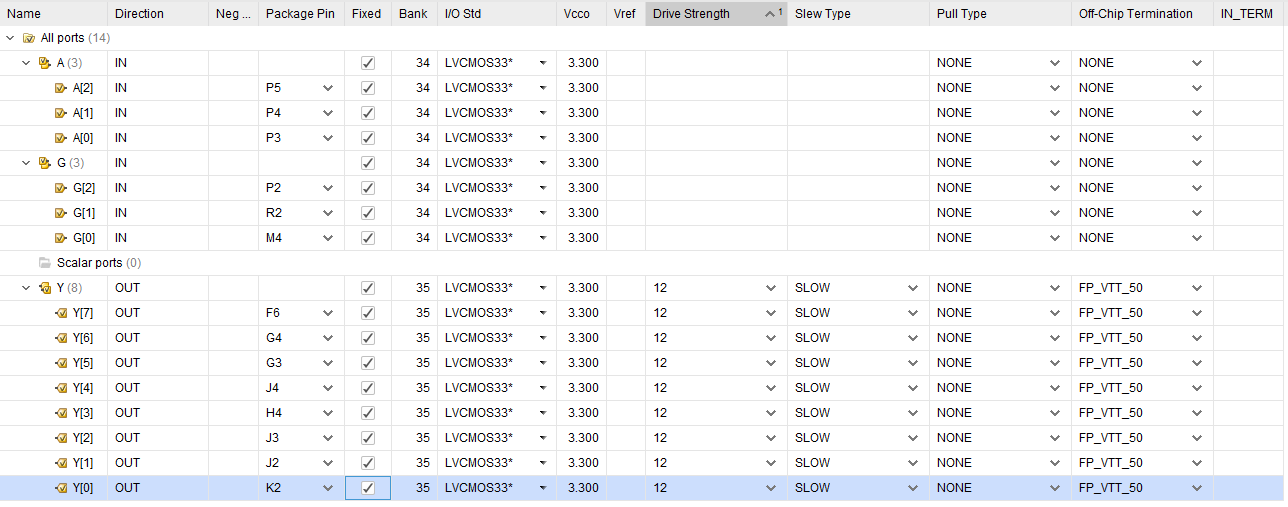
endmodule

五、仿真结果



六、系统网表

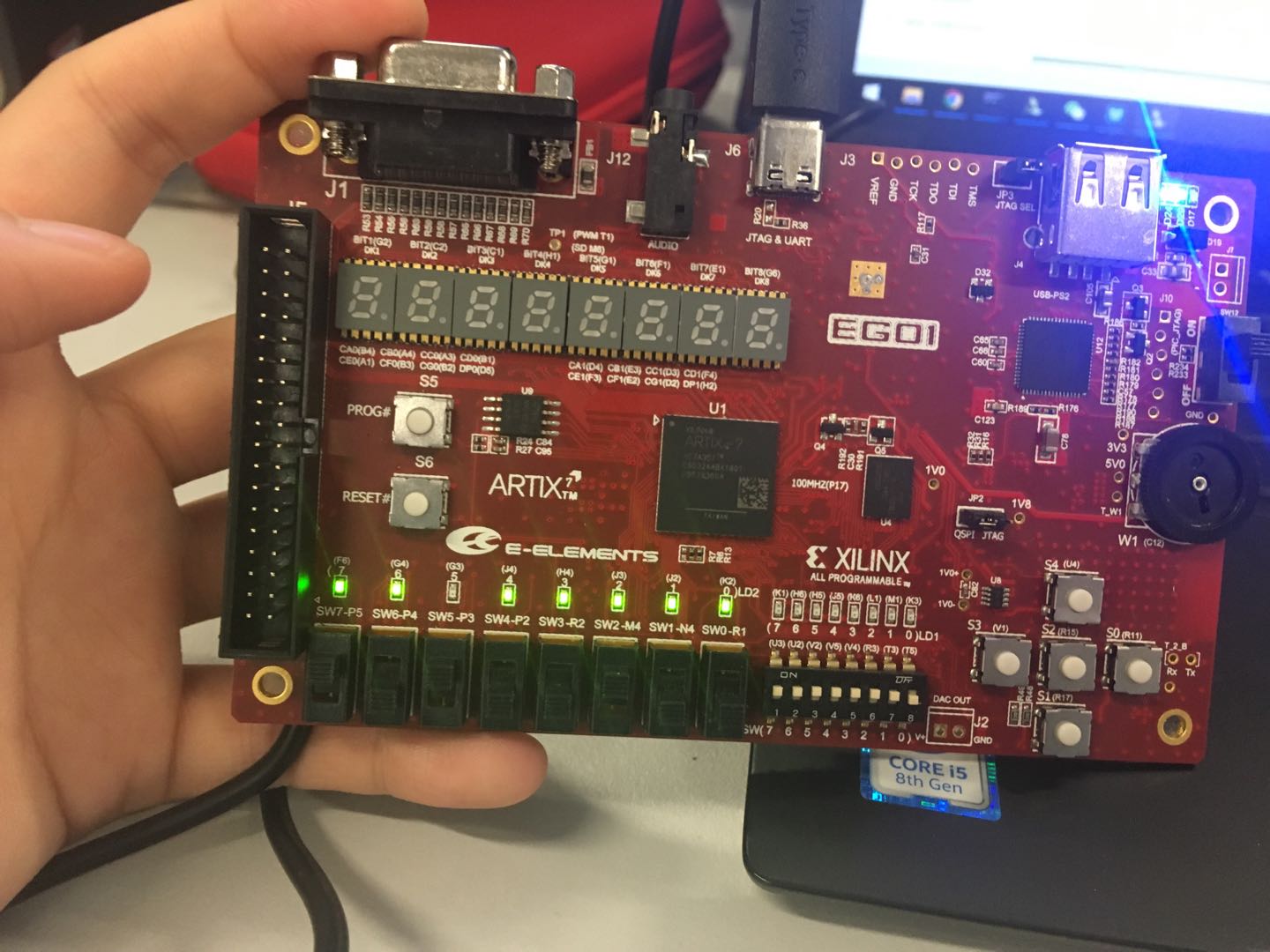
七、系统约束



八、实验结果

示例（A=101, G=001）：

|  |  |  |
| --- | --- | --- |
| 输入 | A (A2 A1 A0) | G (G3 G2 G1) |
| 输入端口 | P5 P4 P3 | P2 R2 M4 |
| 输入的值 | 1 0 1 | 0 0 1 |
| 输出 | Y (Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0) | |
| 输出端口 | F6 G4 G3 J4 H4 J3 J2 K2 | |
| 输出的值 | 1 1 0 1 1 1 1 1 | |



九、实验体会

在做之前的经验积累铺垫下，此次实验完成得比较顺利。但因为一开始的时候是照着图而不是真值表来编码，而图的线路比较复杂（看起来），所以很容易乱；然后改成看真值表，就容易找得到规律了。另外，要注意细心一点儿。

**暨南大学本科实验报告专用纸**

课程名称 数字逻辑实验 成绩评定

实验项目名称 译码器 指导教师 王传胜 梁倬骞

实验项目编号 0806003804 实验项目类型 验证 实验地点 N126

学生姓名 邝庆璇 学号 2016051598

学院 信息科学技术 系 计算机科学 专业 计算机科学与技术

实验时间 2019 年 10 月 28 日 上 午

一、实验目的

掌握用 Verilog HDL 进行数据流建模；

深入理解译码器的原理；

学会用 Verilog HDL 实现74LS138；

二、实验内容

用Verilog HDL 语言实现74LS138的功能。

三、实验程序

exp1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/10/28 12:13:17

// Design Name:

// Module Name: exp\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module exp\_1(

input [2:0] G,

input [2:0] A,

output [7:0] Y

);

wire E\_tmp = (G[0] & ~(G[1]) & ~(G[2]));

wire A0\_not = ~A[0];

wire A1\_not = ~A[1];

wire A2\_not = ~A[2];

assign Y[0] = ~(A0\_not & A1\_not & A2\_not & E\_tmp);

assign Y[1] = ~(A[0] & A1\_not & A2\_not & E\_tmp);

assign Y[2] = ~(A0\_not & A[1] & A2\_not & E\_tmp);

assign Y[3] = ~(A[0] & A[1] & A2\_not & E\_tmp);

assign Y[4] = ~(A0\_not & A1\_not & A[2] & E\_tmp);

assign Y[5] = ~(A[0] & A1\_not & A[2] & E\_tmp);

assign Y[6] = ~(A0\_not & A[1] & A[2] & E\_tmp);

assign Y[7] = ~(A[0] & A[1] & A[2] & E\_tmp);

endmodule

四、仿真程序

sim\_1.v

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/10/28 12:16:20

// Design Name:

// Module Name: sim\_1

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module sim\_1(

output [7:0] Y

);

reg [2:0] G;

reg [2:0] A;

initial

begin

assign A = 3'b000;

assign G = 3'b001;

end

always #20

begin

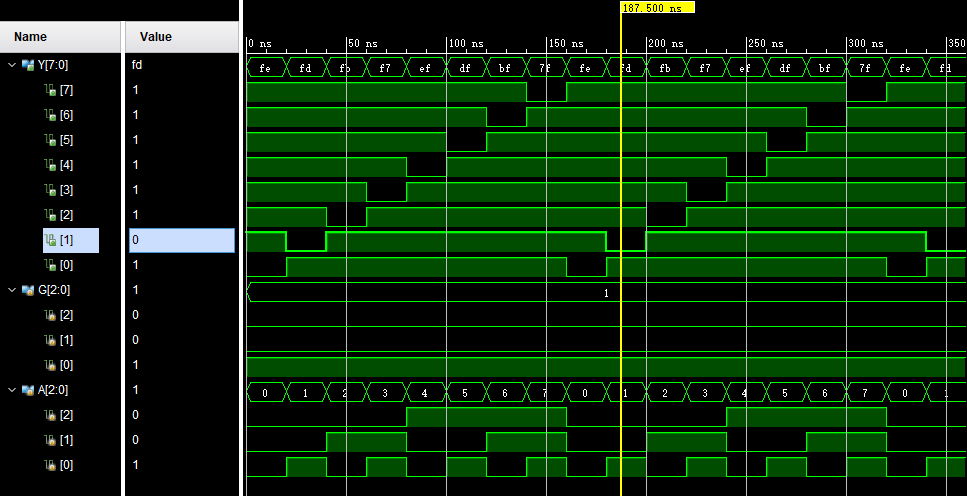
assign A = A + 1;

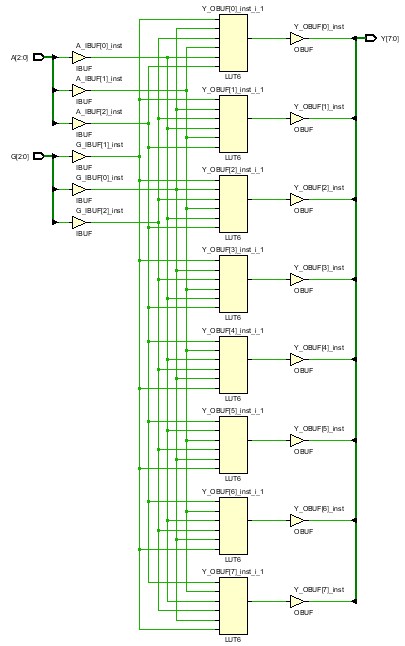
end

exp\_1 exp\_1(G, A, Y);

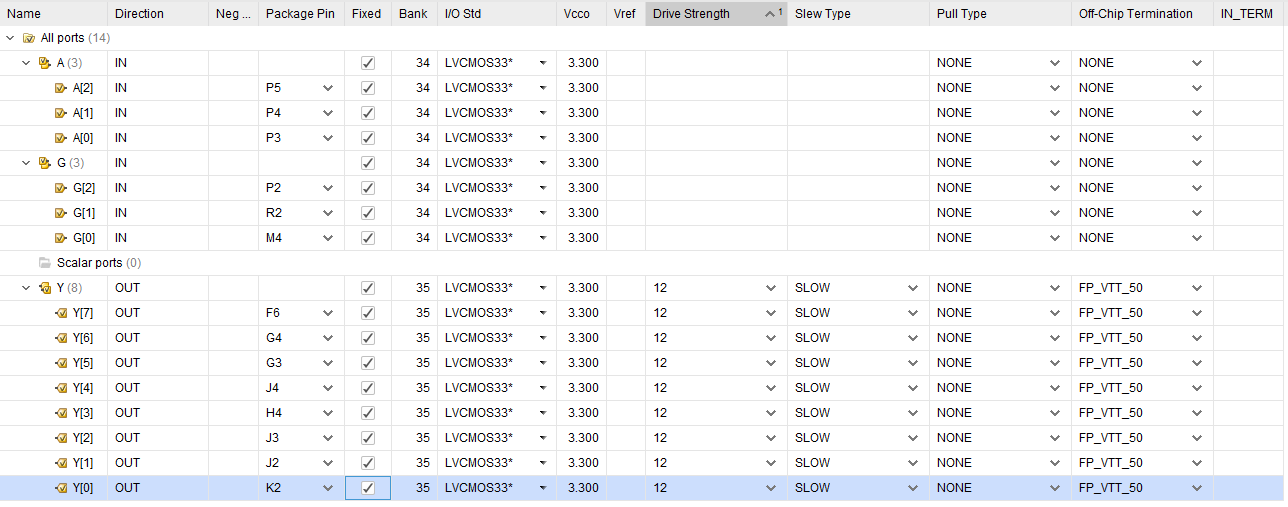
endmodule

五、仿真结果



六、系统网表

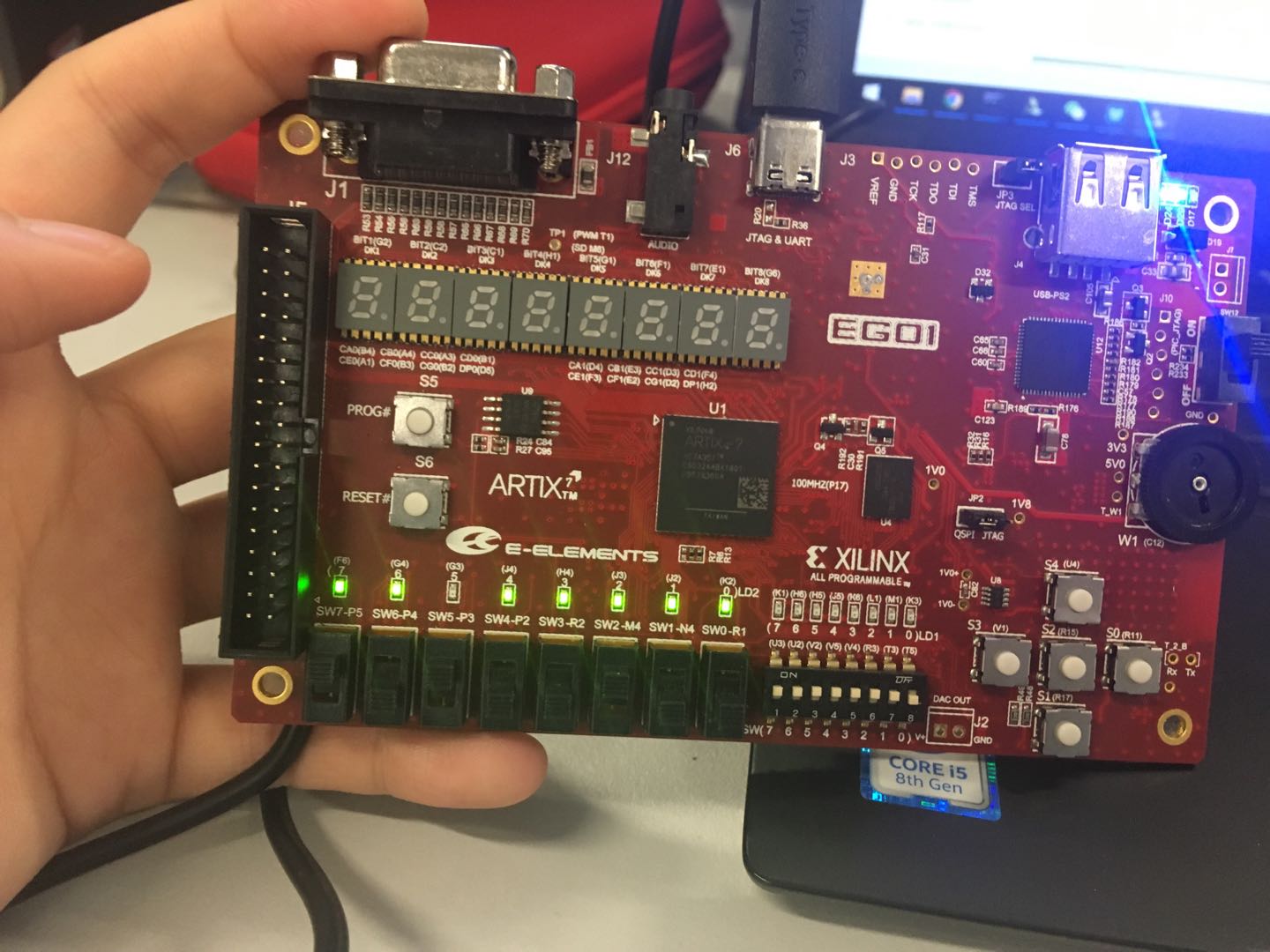
七、系统约束



八、实验结果

示例（A=101, G=001）：

|  |  |  |
| --- | --- | --- |
| 输入 | A (A2 A1 A0) | G (G3 G2 G1) |
| 输入端口 | P5 P4 P3 | P2 R2 M4 |
| 输入的值 | 1 0 1 | 0 0 1 |
| 输出 | Y (Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0) | |
| 输出端口 | F6 G4 G3 J4 H4 J3 J2 K2 | |
| 输出的值 | 1 1 0 1 1 1 1 1 | |



九、实验体会

在做之前的经验积累铺垫下，此次实验完成得比较顺利。但因为一开始的时候是照着图而不是真值表来编码，而图的线路比较复杂（看起来），所以很容易乱；然后改成看真值表，就容易找得到规律了。另外，要注意细心一点儿。