

实验二 模型机组合部件的实现（一）（实验报告格式案例）

班级 信安 2101 姓名 杨华 学号 202108060121

一、实验目的

1. 了解简易模型机的内部结构和工作原理。
2. 熟悉译码器、运算器的工作原理。
3. 分析模型机的功能，设计指令译码器。
4. 分析模型机的功能，设计 ALU。

二、实验内容

1. 用 VERILOG 语言设计指令译码器；
2. 用 VERILOG 语言设计 ALU。

三、实验过程

1、指令译码器

A) 创建工程（选择的芯片为 family=Cyclone II; name=EP2C5T144C8）

左上角 file->New Project Wizard->选择工程位置和工程名->选择芯片 Cyclone II, available device 中选择 EP2C5T144C8->点击 next->最后点击 finish 完成创建工程

B) 编写源代码

```

1 module ins_decode(
2     input en,
3     input [7:0] ir,
4     output reg mova,movb,movc,add,sub,andl,notl,rsr,rsl,jmp,jz,jc,inl,outl,nop,halt);
5     parameter yes=1'b1,no=1'b0;
6     always @(*) begin
7         {mova,movb,movc,add,sub,andl,notl,rsr,rsl,jmp,jz,jc,inl,outl,nop,halt} = no;
8         begin
9             case (ir[7:4])
10                4'b1100 : begin
11                    if(ir[3:2]==2'b11) movb=en&yes;
12                    else if(ir[1:0]==2'b11) movc=en&yes;
13                    else mova=en&yes;
14                end
15                4'b1001 : begin
16                    add=en&yes;
17                end
18                4'b0110 : begin
19                    sub=en&yes;
20                end
21                4'b1011 : begin
22                    andl=en&yes;
23                end
24                4'b0101 : begin
25                    notl=en&yes;
26                end
27                4'b1010 : begin
28                    if(ir[1:0]==2'b00) rsr=en&yes;
29                    else if(ir[1:0]==2'b11) rsl=en&yes;
30                    //else begin end
31                end
32                4'b0011 : begin
33                    if(ir[3:0]==4'b0000) jmp=en&yes;
34                    else if(ir[3:0]==4'b0001) jz=en&yes;
35                    else if(ir[3:0]==4'b0010) jc=en&yes;
36                    //else begin end
37                end
38                4'b0010 : begin
39                    inl=en&yes;
40                end
41                4'b0100 : begin
42                    outl=en&yes;
43                end
44                4'b0111 : begin
45                    if(ir[3:0]==4'b0000) nop=en&yes;
46                    //else begin end;
47                end
48                4'b1000 : begin
49                    if(ir[3:0]==4'b0000) halt=en&yes;
50                    //else begin end;
51                end
52                default: begin end
53            endcase
54        end
55    end
56 endmodule
57

```

C) 编译与调试（包含编译调试过程中的错误、警告信息以及资源消耗）

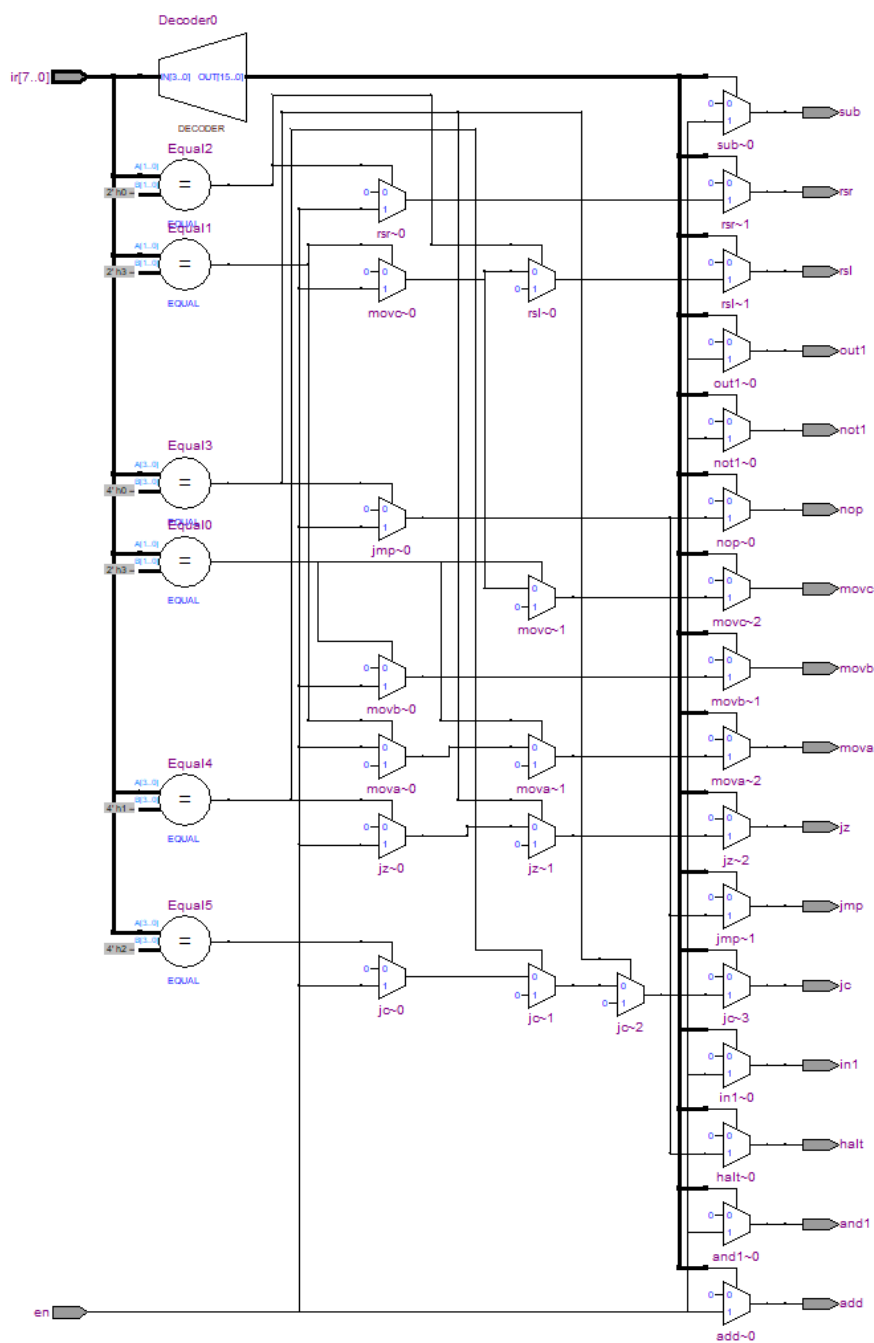
错误与警告信息：

Type	Message
Info	Longest tpd from source pin "ir[4]" to destination pin "jc" is 15.847 ns
Info	Quartus II Classic Timing Analyzer was successful. 0 errors, 0 warnings
Info	Quartus II Full Compilation was successful. 0 errors, 4 warnings
Warning	Feature LogicLock is not available with your current license
Warning	No exact pin location assignment(s) for 25 pins of 25 total pins
Warning	Found 16 output pins without output pin load capacitance assignment
Warning	The Reserve All Unused Pins setting has not been specified, and will default to 'As output driving ground'.

资源消耗:

```
Flow Status                Successful - Tue Nov 22 15:29:31 2022
Quartus II Version          9.0 Build 184 04/29/2009 SP 1 SJ Web Edition
Revision Name               ins_decode
Top-level Entity Name       ins_decode
Family                      Cyclone II
Device                      EP2C5T144C8
Timing Models               Final
Met timing requirements      Yes
Total logic elements         30 / 4,608 ( < 1 % )
    Total combinational functions  30 / 4,608 ( < 1 % )
    Dedicated logic registers      0 / 4,608 ( 0 % )
Total registers              0
Total pins                   25 / 89 ( 28 % )
Total virtual pins           0
Total memory bits            0 / 119,808 ( 0 % )
Embedded Multiplier 9-bit elements  0 / 26 ( 0 % )
Total PLLs                   0 / 2 ( 0 % )
```

D) RTL 视图

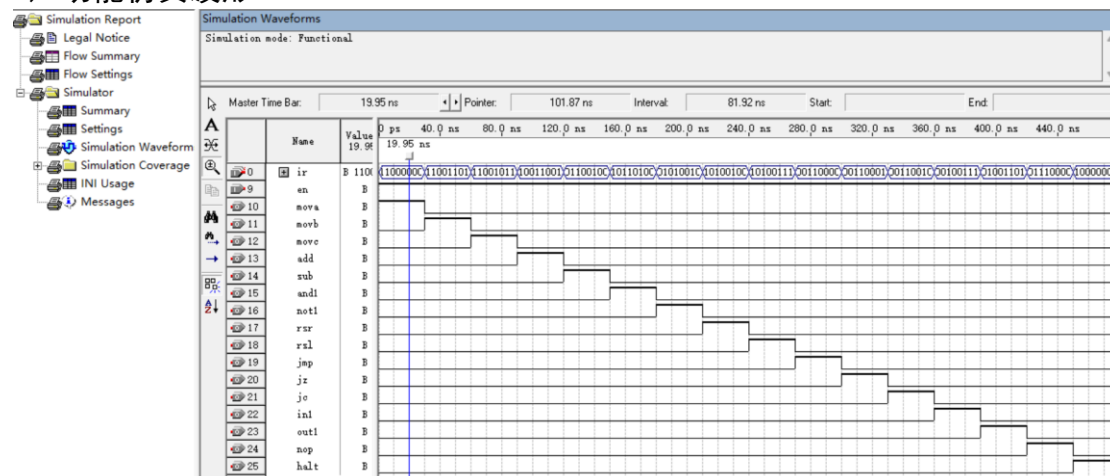


视图分析及结论：

分析：由视图分析可知，左侧为输入，右侧为输出。其中连接有一系列的元器件。比如比较器：当输入相等时输出 1，不相等时输出 0；还有大部分的 2-1 选择器构成，当控制信号为 0 时，输出第一位，控制信号为 1 时，输出第二位。图中输入信号为 *ir* 和 *en*，输出信号包 括 *add* 等 16 种情况。各个输出端口之间通过导线相连。

结论：电路结构比较复杂，最后的输出往往要经过多重门

E) 功能仿真波形



结果分析及结论：

结果分析：功能仿真也被称作 RTL 级行为仿真，前仿真，目的是分析设计电路逻辑关系的正确性。功能仿真不带有任何的门延时、线延时等等，只是理想情况下的仿真。

当 en 为 0 时，无论 ir 为何值时，16 个输出全为 0。

当 en 为 1 时，

ir=11000000 时，mova 输出为 1

ir=11001100 时，movb 输出为 1

ir=11000111 时，movc 输出为 1

ir=10011001 时，add 输出为 1

ir=01100100 时，sub 输出为 1

ir=10110100 时，andl 输出为 1

ir=01010010 时，notl 输出为 1

ir=10100100 时，rsl 输出为 1

ir=10100111 时，rsl 输出为 1

ir=00110001 时，jz 输出为 1

ir=00110010 时，jc 输出为 1

ir=00100111 时，inl 输出为 1

ir=01001101 时，outl 输出为 1

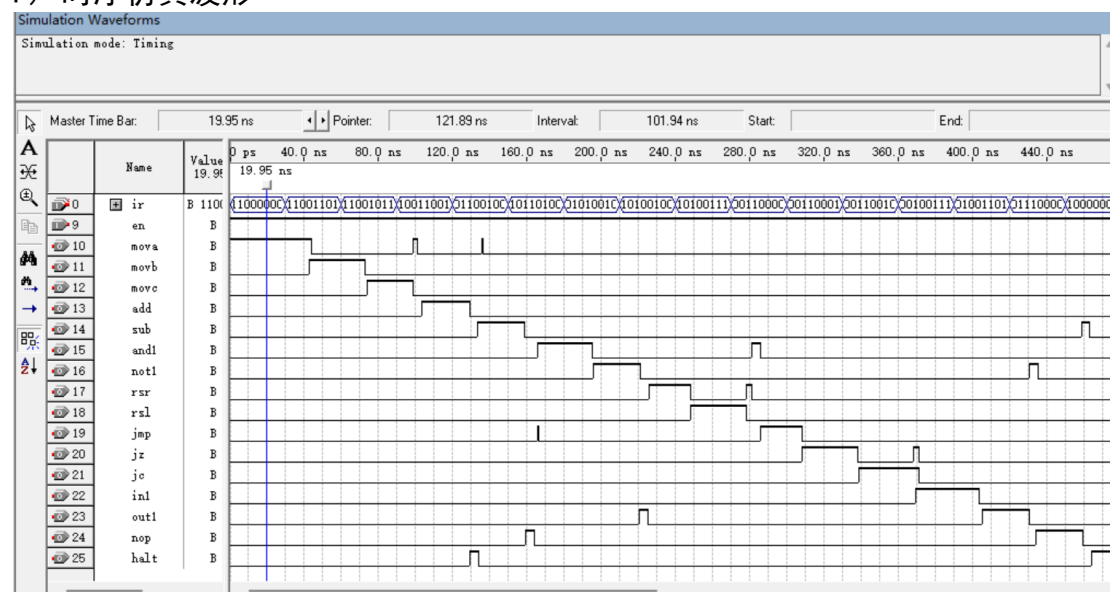
ir=01110000 时，nop 输出为 1

ir=00000001 时，halt 输出为 1

结论：

功能仿真操作简单，仿真速度快，但是功能仿真仅仅关注输出和输入的逻辑关系是否正确，不考虑时间延时信息，与实际情况可能会有一定误差。仿真结果与电路设计的真值表的结果相对应。

F) 时序仿真波形



结果分析及结论：

分析：时序仿真是在布线后进行，了解实现的功能是否满足真实器件运行的要求，与特定的器件有关。时序仿真输入改变后，输出并未立刻改变，而是出现了延迟。同时由于输入电路的改变，有可能会出现冒险，使得输入结果与实际输出结果有所不同。

结论：时序仿真不仅关注输出和输入的逻辑关系是否正确，同时还计算了时间延时信息。

G) 时序分析

操作方法是：编译后，在 compilation report 中选择【timing analysis】-【summary】和【tpd】

Timing Analyzer Summary										
	Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths	
1	Worst-case tpd	N/A	None	15.847 ns	ir[4]	jc	--	--	0	
2	Total number of failed paths								0	

实验一 译码器的实现

tpd					
	Slack	Required P2P Time	Actual P2P Time	From	To
1	N/A	None	15.847 ns	ir[4]	jc
2	N/A	None	15.785 ns	ir[4]	movc
3	N/A	None	15.700 ns	ir[6]	movc
4	N/A	None	15.615 ns	ir[4]	halt
5	N/A	None	15.593 ns	ir[6]	jc
6	N/A	None	15.480 ns	ir[5]	movc
7	N/A	None	15.440 ns	ir[4]	nop
8	N/A	None	15.433 ns	ir[6]	halt
9	N/A	None	15.419 ns	ir[5]	jc
10	N/A	None	15.360 ns	ir[4]	jmp
11	N/A	None	15.320 ns	ir[5]	halt
12	N/A	None	15.272 ns	ir[6]	nop
13	N/A	None	15.209 ns	ir[4]	rsr
14	N/A	None	15.096 ns	ir[6]	jmp
15	N/A	None	15.050 ns	ir[4]	movb
16	N/A	None	15.038 ns	ir[6]	rsr
17	N/A	None	15.009 ns	ir[5]	nop
18	N/A	None	14.984 ns	ir[4]	iz
19	N/A	None	14.965 ns	ir[6]	movb
20	N/A	None	14.932 ns	ir[5]	jmp
21	N/A	None	14.900 ns	ir[5]	rsr
22	N/A	None	14.834 ns	ir[1]	movc
23	N/A	None	14.745 ns	ir[5]	movb
24	N/A	None	14.742 ns	ir[4]	rsr
25	N/A	None	14.720 ns	ir[6]	iz
26	N/A	None	14.571 ns	ir[6]	rsr
27	N/A	None	14.556 ns	ir[5]	iz
28	N/A	None	14.433 ns	ir[5]	rsr
29	N/A	None	14.392 ns	en	jc
30	N/A	None	14.373 ns	ir[4]	movb
31	N/A	None	14.288 ns	ir[6]	movb
32	N/A	None	14.281 ns	en	halt
33	N/A	None	14.236 ns	ir[4]	not1
34	N/A	None	14.229 ns	ir[4]	out1
35	N/A	None	14.152 ns	ir[6]	out1
36	N/A	None	14.109 ns	ir[4]	in1
37	N/A	None	14.084 ns	ir[6]	not1
38	N/A	None	14.068 ns	ir[5]	movb
39	N/A	None	13.995 ns	en	movc
40	N/A	None	13.948 ns	en	nop
41	N/A	None	13.927 ns	ir[6]	in1
42	N/A	None	13.909 ns	en	jmp
43	N/A	None	13.884 ns	ir[5]	out1
44	N/A	None	13.884 ns	ir[5]	out1
45	N/A	None	13.855 ns	ir[4]	sub
46	N/A	None	13.803 ns	ir[5]	not1
47	N/A	None	13.778 ns	ir[6]	sub
48	N/A	None	13.777 ns	ir[1]	jc
49	N/A	None	13.765 ns	ir[5]	in1
50	N/A	None	13.749 ns	en	movb
51	N/A	None	13.733 ns	ir[1]	rsr
52	N/A	None	13.587 ns	ir[1]	halt
53	N/A	None	13.520 ns	en	iz
54	N/A	None	13.502 ns	ir[4]	and1
55	N/A	None	13.502 ns	ir[5]	sub
56	N/A	None	13.437 ns	en	rsr
57	N/A	None	13.384 ns	ir[1]	move
58	N/A	None	13.254 ns	ir[1]	nop
59	N/A	None	13.227 ns	ir[6]	and1
60	N/A	None	13.214 ns	ir[1]	jmp
61	N/A	None	13.185 ns	en	not1
62	N/A	None	13.168 ns	ir[1]	rsr
63	N/A	None	13.138 ns	en	in1
64	N/A	None	13.114 ns	ir[5]	and1
65	N/A	None	13.062 ns	ir[4]	add
66	N/A	None	13.027 ns	en	rsr
67	N/A	None	12.835 ns	ir[1]	iz
68	N/A	None	12.806 ns	ir[6]	add
69	N/A	None	13.027 ns	en	rsr
70	N/A	None	12.835 ns	ir[1]	iz
71	N/A	None	12.806 ns	ir[6]	add
72	N/A	None	12.679 ns	ir[5]	add
73	N/A	None	12.548 ns	en	movb
74	N/A	None	12.461 ns	en	and1
75	N/A	None	12.468 ns	en	out1
76	N/A	None	12.460 ns	en	sub
77	N/A	None	12.057 ns	en	add
78	N/A	None	11.031 ns	ir[7]	movc
79	N/A	None	10.980 ns	ir[7]	jc
80	N/A	None	10.874 ns	ir[7]	halt
81	N/A	None	10.555 ns	ir[7]	nop
82	N/A	None	10.493 ns	ir[7]	jmp
83	N/A	None	10.478 ns	ir[7]	rsr
84	N/A	None	10.450 ns	ir[2]	movc
85	N/A	None	10.296 ns	ir[7]	movb
86	N/A	None	10.117 ns	ir[7]	iz
87	N/A	None	10.041 ns	ir[3]	movc
88	N/A	None	10.011 ns	ir[7]	rsr
89	N/A	None	9.930 ns	ir[0]	movc
90	N/A	None	9.653 ns	ir[2]	jc
91	N/A	None	9.619 ns	ir[7]	movb
92	N/A	None	9.542 ns	ir[2]	halt
93	N/A	None	9.438 ns	ir[7]	out1
94	N/A	None	9.373 ns	ir[7]	not1
95	N/A	None	9.326 ns	ir[7]	in1
96	N/A	None	9.318 ns	ir[3]	jc
97	N/A	None	9.209 ns	ir[2]	nop
98	N/A	None	9.207 ns	ir[3]	halt
99	N/A	None	9.170 ns	ir[2]	jmp
100	N/A	None	9.124 ns	ir[0]	rsr
101	N/A	None	9.106 ns	ir[2]	movb
102	N/A	None	9.060 ns	ir[7]	sub
103	N/A	None	9.042 ns	ir[2]	movb
104	N/A	None	8.922 ns	ir[0]	jc
105	N/A	None	8.874 ns	ir[3]	nop
106	N/A	None	8.835 ns	ir[3]	jmp
107	N/A	None	8.804 ns	ir[0]	halt
108	N/A	None	8.781 ns	ir[2]	iz
109	N/A	None	8.694 ns	ir[3]	movb
110	N/A	None	8.659 ns	ir[7]	and1
111	N/A	None	8.633 ns	ir[3]	move
112	N/A	None	8.540 ns	ir[0]	rsr
113	N/A	None	8.480 ns	ir[0]	movb
114	N/A	None	8.477 ns	ir[0]	nop
115	N/A	None	8.446 ns	ir[3]	iz
116	N/A	None	8.431 ns	ir[0]	jmp
117	N/A	None	8.248 ns	ir[7]	add
118	N/A	None	8.050 ns	ir[0]	iz

结果分析及结论：由图可得，Timing Analyzer Summmary 总结所有经典定时分析的结果，并报告每个定时特性的最坏情况定时。

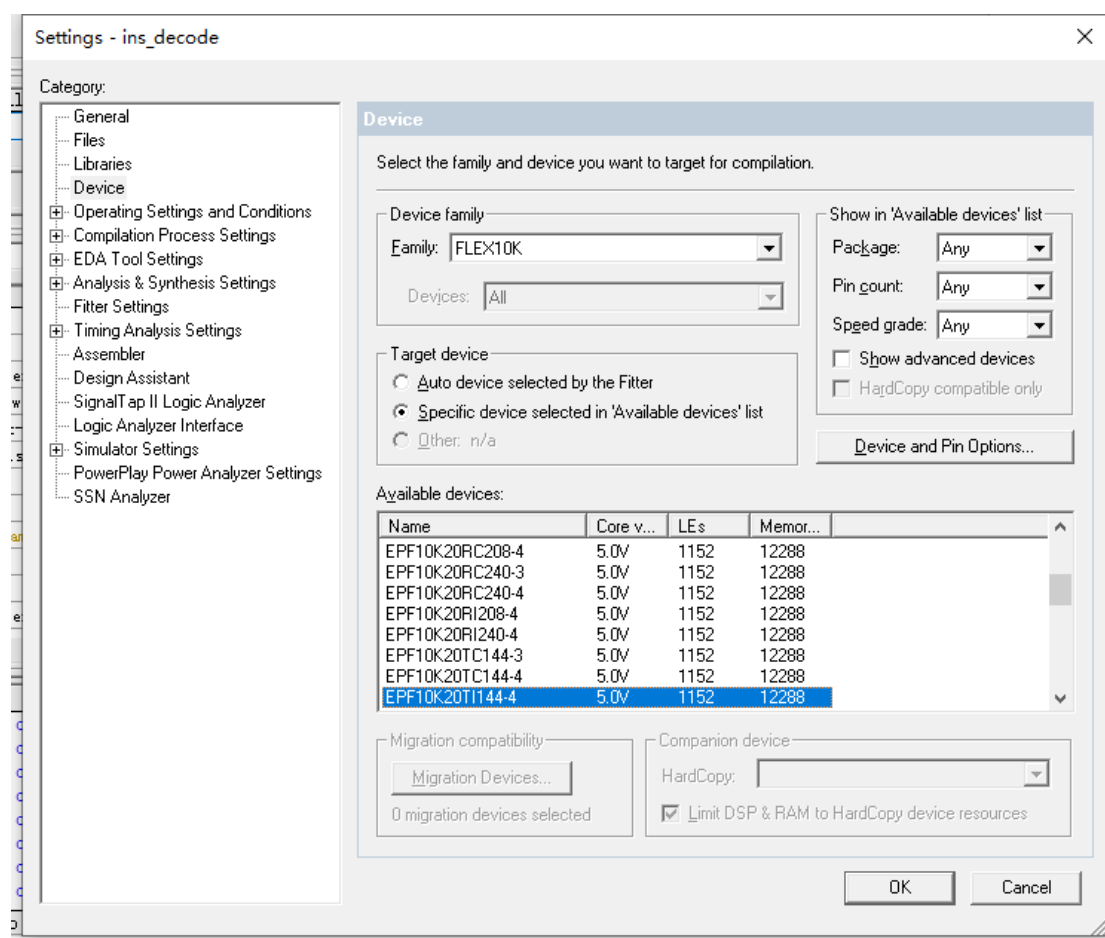
结论：实际连接图中个元器件连接之间是存在时间延迟的，而且不同的元器件之间的时间延迟也不相同。

2、算术逻辑单元 ALU

A) 创建工程（选择的芯片为 family=FLEX10K；name=EPF10K20T1144-4）

步骤：

左上角 file->New Project Wizard->选择工程位置和工程名->选择芯片 FLEX10K， available device 中选择 EPF10K20T1144-4->点击 next->最后点击 finish 完成创建工程



B) 编写源代码

```

module simple(m,s,a,b,t,cf,zf);
    input [7:0] a;
    input [7:0] b;
    input [3:0] s;
    input m;
    output [7:0] t;
    output cf,zf;
    reg cf,zf;
    reg [7:0] t;
    always@(m,s,a,b)
begin
    t=8'b0;
    cf=1'b0;
    zf=1'b0;
    if(m==1)
begin
    if(s==4'b1001)
begin
    {cf,t}=a+b;
    if(t==0)
    zf=1'b1;
    else
    zf=0;
    end
    else if(s==4'b0110)
begin
    {cf,t}=a-b;
    if(t==0)
    zf=1'b1;
    else
    zf=0;
    end
    else if(s==4'b1011)
    t=a&b;
    else if(s==4'b0101)
    t=~b;
    else if(s==4'b1010||s==4'b0100)
    t=b;
    else
    t=a;
    end
    else t=a;
    end
endmodule

```

C) 编译与调试（包含编译调试过程中的错误、警告信息以及资源消耗）
警告信息

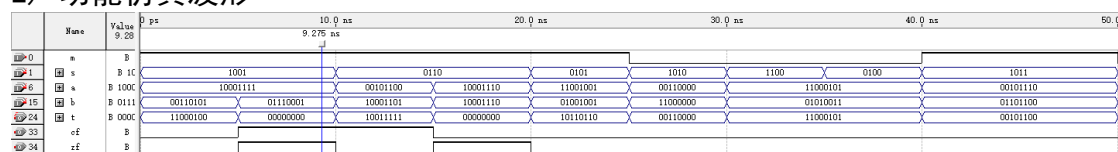
Type	Message
Info	Info: Quartus II Analysis & Synthesis was successful. 0 errors, 0 warnings
Info	Info: *****
Info	Info: Running Quartus II Fitter
Info	Info: Command: quartus_fit --read_settings_files=off --write_settings_files=off ins_decode -c ins_decode
Info	Info: Selected device EPF10K20T1144-4 for design "ins_decode"
Warning	Warning: Feature SignalProbe is not available with your current license
Info	Info: Fitter is using the Classic Timing Analyzer
Info	Info: Timing requirements not specified -- optimizing circuit to achieve the following default global requirements

输入相等时输出 1，不相等时输出 0；还有大部分的 2-1 选择器构成，当控制信号为 0 时，输出第一位，控制信号为 1 时，输出第二位。

结论：

一个功能实现需要经过多重门的处理，而且每个元件内部结构十分复杂

E) 功能仿真波形



结果分析及结论：

当 m 为 1 时：

s=1001, a=1000111, b=00110101, t=11000100, cf=0, zf=0;

s=1001, a=1000111, b=11000100, t=00000000, cf=1, zf=1;

s=0110, a=00101100, b=10001110, t=11000100, cf=0, zf=0;

s=0110, a=10001101, b=11000100, t=00000000, cf=1, zf=1;

s=0101, a=11001001, b=01001001, t=10110110, cf=0, zf=0;

s=1011, a=00101110, b=01101100, t=00101100, cf=0, zf=0;

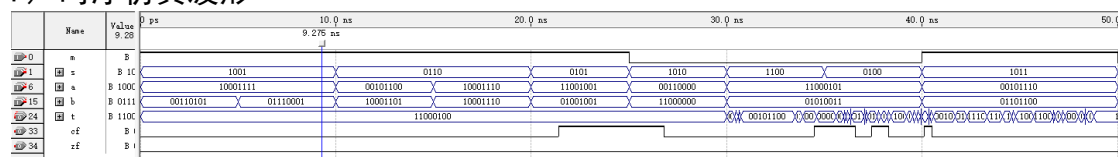
当 m 为 0 时：

s=0101, a=11001001, b=01001001, t=10110110, cf=0, zf=0;

s=1100, a=11000101, b=01010011, t=01010011, cf=0, zf=0;

s=0100, a=11000101, b=01010011, t=01010011, cf=0, zf=0;

F) 时序仿真波形



结果分析及结论：

结果分析：时序仿真是在布线后进行，了解实现的功能是否满足真实器件运行的要求，与特定的器件有关。时序仿真输入改变后，输出并未立刻改变，而是出现了延迟。同时由于输入电路的改变，有可能会出现冒险，使得输入结果与实际输出结果有所不同。

结论：时序仿真不仅关注输出和输入的逻辑关系是否正确，同时还计算了时间延时信息。

G) 时序分析

Timing Analyzer Summary										
	Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths	
1	Worst-case tpd	N/A	None	30.600 ns	s[0]	t[0]	--	--	0	
2	Total number of failed paths								0	

实验一 译码器的实现

ipd						
Slack	Regard	Actual PPT	From	To		
PPT Time		Time				
1	N/A	None	30.600 ns	403		
2	N/A	None	30.200 ns	403		
3	N/A	None	29.600 ns	403		
4	N/A	None	29.200 ns	403		
5	N/A	None	29.200 ns	403		
6	N/A	None	28.600 ns	403		
7	N/A	None	28.600 ns	402		
8	N/A	None	28.600 ns	403		
9	N/A	None	28.600 ns	402		
10	N/A	None	28.500 ns	401		
11	N/A	None	28.500 ns	402		
12	N/A	None	28.500 ns	401		
13	N/A	None	28.500 ns	403		
14	N/A	None	28.500 ns	401		
15	N/A	None	28.400 ns	402		
16	N/A	None	28.400 ns	403		
17	N/A	None	28.300 ns	401		
18	N/A	None	28.300 ns	403		
19	N/A	None	28.200 ns	402		
20	N/A	None	28.200 ns	403		
21	N/A	None	28.200 ns	403		
22	N/A	None	28.200 ns	403		
23	N/A	None	28.100 ns	401		
24	N/A	None	28.100 ns	401		
25	N/A	None	28.100 ns	403		
26	N/A	None	28.100 ns	401		
27	N/A	None	28.100 ns	403		
28	N/A	None	28.100 ns	402		
29	N/A	None	28.100 ns	402		
30	N/A	None	28.100 ns	401		
31	N/A	None	28.100 ns	403		
32	N/A	None	28.000 ns	403		
33	N/A	None	28.000 ns	401		
34	N/A	None	27.900 ns	401		
35	N/A	None	27.900 ns	403		
36	N/A	None	27.800 ns	402		
37	N/A	None	27.700 ns	403		
38	N/A	None	27.600 ns	403		
39	N/A	None	27.600 ns	403		
40	N/A	None	27.600 ns	401		
41	N/A	None	27.500 ns	401		
42	N/A	None	27.400 ns	402		
43	N/A	None	27.300 ns	401		
44	N/A	None	27.200 ns	401		
45	N/A	None	27.200 ns	402		
46	N/A	None	27.200 ns	401		
47	N/A	None	27.100 ns	402		
48	N/A	None	27.100 ns	403		
49	N/A	None	27.100 ns	402		
50	N/A	None	27.100 ns	402		
51	N/A	None	27.100 ns	403		
52	N/A	None	27.100 ns	401		
53	N/A	None	27.000 ns	404		
54	N/A	None	27.000 ns	404		
55	N/A	None	27.000 ns	403		
56	N/A	None	26.900 ns	404		
57	N/A	None	26.900 ns	404		
58	N/A	None	26.900 ns	402		
59	N/A	None	26.900 ns	404		
60	N/A	None	26.900 ns	401		
61	N/A	None	26.900 ns	404		
62	N/A	None	26.700 ns	404		
63	N/A	None	26.700 ns	402		
64	N/A	None	26.700 ns	403		
65	N/A	None	26.700 ns	401		
66	N/A	None	26.600 ns	402		
67	N/A	None	26.600 ns	403		
68	N/A	None	26.600 ns	403		
69	N/A	None	26.600 ns	403		
70	N/A	None	26.600 ns	402		
71	N/A	None	26.600 ns	404		
72	N/A	None	26.600 ns	402		
73	N/A	None	26.600 ns	402		
74	N/A	None	26.500 ns	402		
75	N/A	None	26.500 ns	404		
76	N/A	None	26.400 ns	402		
77	N/A	None	26.400 ns	403		
78	N/A	None	26.300 ns	402		
79	N/A	None	26.300 ns	402		
80	N/A	None	26.200 ns	403		
81	N/A	None	26.200 ns	403		
82	N/A	None	26.200 ns	404		
83	N/A	None	26.200 ns	401		
84	N/A	None	26.200 ns	402		
85	N/A	None	26.200 ns	403		
86	N/A	None	26.100 ns	403		
87	N/A	None	26.100 ns	402		
88	N/A	None	26.100 ns	401		
89	N/A	None	26.000 ns	403		
90	N/A	None	26.000 ns	403		
91	N/A	None	26.000 ns	403		
92	N/A	None	26.000 ns	403		
93	N/A	None	26.000 ns	402		
94	N/A	None	26.000 ns	403		
95	N/A	None	26.000 ns	403		
96	N/A	None	25.900 ns	402		
97	N/A	None	25.800 ns	403		
98	N/A	None	25.800 ns	403		
99	N/A	None	25.700 ns	403		
100	N/A	None	25.700 ns	403		
101	N/A	None	25.700 ns	401		
102	N/A	None	25.700 ns	401		
103	N/A	None	25.600 ns	402		
104	N/A	None	25.600 ns	403		
105	N/A	None	25.600 ns	403		
106	N/A	None	25.500 ns	403		
107	N/A	None	25.500 ns	402		
108	N/A	None	25.500 ns	403		
109	N/A	None	25.500 ns	403		
110	N/A	None	25.400 ns	403		
111	N/A	None	25.300 ns	401		
112	N/A	None	25.300 ns	403		
113	N/A	None	25.300 ns	402		
114	N/A	None	25.100 ns	401		
115	N/A	None	25.100 ns	402		
116	N/A	None	25.100 ns	403		
117	N/A	None	25.000 ns	401		
118	N/A	None	25.000 ns	403		
119	N/A	None	24.900 ns	403		
120	N/A	None	24.900 ns	401		
121	N/A	None	24.800 ns	403		
122	N/A	None	24.800 ns	403		
123	N/A	None	23.700 ns	401		
124	N/A	None	23.700 ns	402		
125	N/A	None	23.500 ns	403		
126	N/A	None	23.400 ns	403		
127	N/A	None	23.400 ns	403		
128	N/A	None	23.400 ns	401		
129	N/A	None	23.300 ns	401		
130	N/A	None	22.800 ns	402		
131	N/A	None	22.400 ns	404		
132	N/A	None	22.400 ns	403		
133	N/A	None	22.300 ns	402		
134	N/A	None	22.300 ns	404		
135	N/A	None	22.100 ns	404		
136	N/A	None	22.000 ns	402		
137	N/A	None	22.000 ns	401		
138	N/A	None	22.000 ns	403		
139	N/A	None	22.000 ns	403		
140	N/A	None	21.800 ns	403		
141	N/A	None	21.800 ns	403		
142	N/A	None	21.800 ns	403		
143	N/A	None	21.700 ns	403		
144	N/A	None	21.600 ns	402		
145	N/A	None	21.300 ns	403		
146	N/A	None	21.200 ns	403		
147	N/A	None	21.200 ns	402		
148	N/A	None	21.000 ns	402		
149	N/A	None	21.000 ns	401		
150	N/A	None	21.000 ns	403		
151	N/A	None	20.900 ns	403		
152	N/A	None	20.900 ns	403		
153	N/A	None	18.800 ns	403		
154	N/A	None	17.400 ns	403		

结果分析及结论：

由图可得，Timing Analyzer Summary 总结所有经典定时分析的结果，并报告每个 定时特性的最坏情况定时。如从 m 到 zf 最大延迟为 18.100ns，而 a[0]到 t[7]延迟则为 31.100ns。

结论：实际连接图中个元器件连接之间是存在时间延迟的，而且不同的元器件之间的时间延迟也不相同。

四、思考题

1. 指令译码器必须要 16 个输出吗？可否将一些输出合并，哪些可以合并，为什么？

不必。jmp 和 add 可以合并起来，因为 jmp 是将 add 后的结果写入 pc 中，则可以进 行 add 操作后直接进行写入操作。Add 和 sub 和 and 操作可以合并，因为这三个操作类似，且输出为使能信号，故可以用一个合并使能信号来作为三个输出的共同使能信号

2. ALU 中的 S[3..0]控制信号是来自哪里或者说与什么信息相同？

来自 ir 的前四位

3、为何 S[3..0]等于 1100 或 0100 时将输入 a 传给 t，S[3..0]等于 1010 时将输入 b 传给 t？

S[3..0]为控制信号，当 S[3..0]输入为 1100 时，控制输出 t 等于 a，S[3..0]等于 1010 或 0100 时，t 等于 b，此时 ALU 相当于选择器。

五、实验总结、必得体会及建议

1、从需要掌握的理论、遇到的困难、解决的办法以及经验教训等方面进行总结。

基本了解了简易模型机的内部结构，工作原理和功能。同时熟悉了译码器、ALU 的工作原 理。学会使用 Verilog 语言编写电路。

遇到的困难：不会使用波形功能仿真功能，通过百度解决，在编写代码的时候语言对 always, case 等语法不熟，通过学习代码解决 。

2、对本实验内容、过程和方法的改进建议（可选项）。

无