

## 实验二 模型机组合部件的实现（一）

班级 计科 2305 姓名 刘良锐 学号 202308010524

### 一、实验目的

- 1.了解简易模型机的内部结构和工作原理。
- 2.熟悉译码器和运算器的工作原理。
- 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。

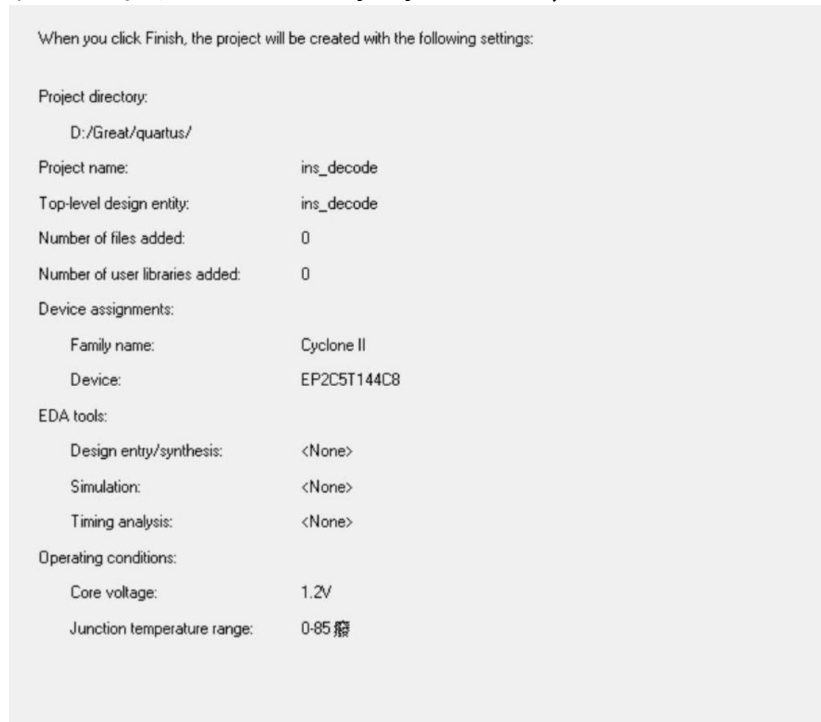
### 二、实验内容

- 1.使用 Verilog 语言设计指令译码器。
- 2.使用 Verilog 语言设计 AU 算术单元。

### 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）



## B) 编写源代码

```

1 module ins_decode (en,ir,mova,movb,movc,movd,add,sub,jmp,jg,ini,outl,movi,halt);
2     input en;
3     input [3:0] ir;
4     output mova,movb,movc,movd,add,sub,jmp,jg,ini,outl,movi,halt;
5
6     reg [11:0] status;
7
8     always @ (en,ir)
9     begin
10         if(en==0)
11             begin status<=12'b0000_0000_0000; end
12         else
13             begin
14                 if(ir==4'b0100)
15                     begin status<=12'b1000_0000_0000; end
16                 else if(ir==4'b0101)
17                     begin status<=12'b0100_0000_0000; end
18                 else if(ir==4'b0110)
19                     begin status<=12'b0010_0000_0000; end
20                 else if(ir==4'b0111)
21                     begin status<=12'b0001_0000_0000; end
22                 else if(ir==4'b1000)
23                     begin status<=12'b0000_1000_0000; end
24                 else if(ir==4'b1001)
25                     begin status<=12'b0000_0100_0000; end
26                 else if(ir==4'b1010)
27                     begin status<=12'b0000_0010_0000; end
28                 else if(ir==4'b1011)
29                     begin status<=12'b0000_0001_0000; end
30                 else if(ir==4'b1100)
31                     begin status<=12'b0000_0000_1000; end
32                 else if(ir==4'b1101)
33                     begin status<=12'b0000_0000_0100; end
34                 else if(ir==4'b1110)
35                     begin status<=12'b0000_0000_0010; end
36                 else if(ir==4'b1111)
37                     begin status<=12'b0000_0000_0001; end
38                 else
39                     begin status<=12'b0000_0000_0000; end
40             end
41         end
42     end
43     assign (mova,movb,movc,movd,add,sub,jmp,jg,ini,outl,movi,halt)=status;
44 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）

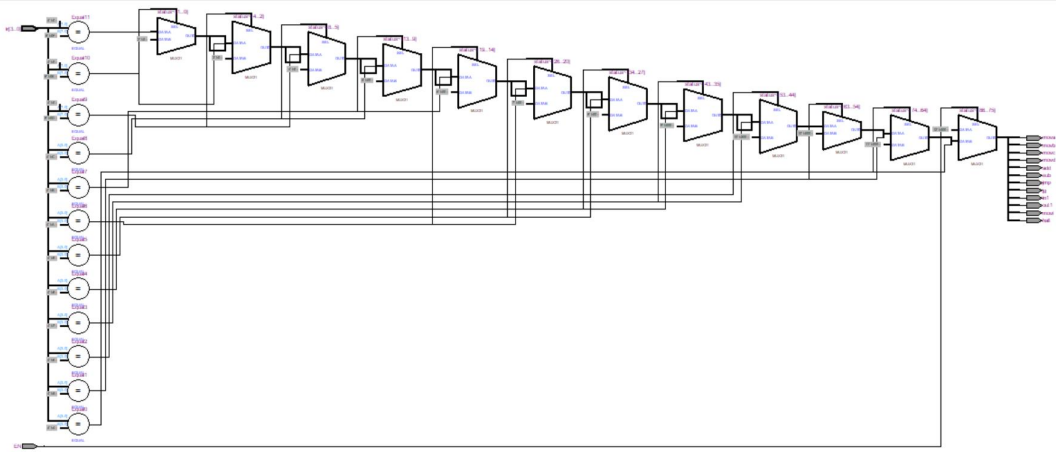
## 1. 对代码进行编译，出现 0 个错误、4 个警告。

Type	Message
Info	Info: Running Quartus II Classic Timing Analyzer
Info	Info: Command: quartus_tan --read_settings_files=off --write_settings_files=off ins_decode -c ins_decode --timing_analysis_only
Info	Info: Longest tpd from source pin "ir[1]" to destination pin "add" is 15.225 ns
Info	Info: Quartus II Classic Timing Analyzer was successful. 0 errors, 0 warnings
Info	Info: Quartus II Full Compilation was successful. 0 errors, 4 warnings
Info	Info: Running Quartus II Netlist Viewers Preprocess

## 2. 资源消耗

Flow Status	Successful - Fri Nov 22 16:39:52 2024
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	24 / 4,608 ( < 1 % )
Total combinational functions	24 / 4,608 ( < 1 % )
Dedicated logic registers	0 / 4,608 ( 0 % )
Total registers	0
Total pins	17 / 89 ( 19 % )
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

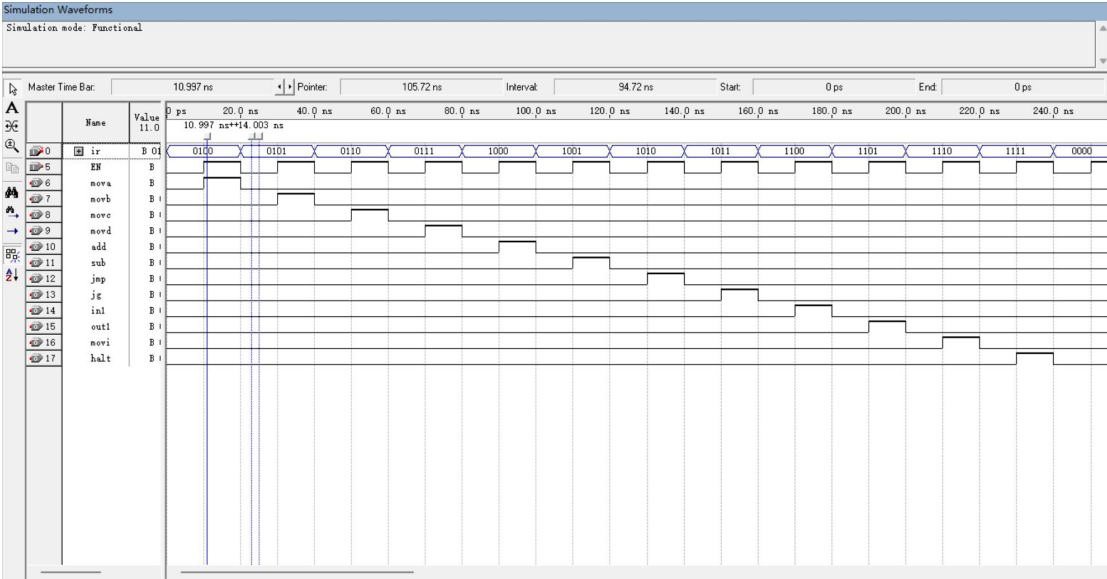


视图分析及结论

视图分析：通过观察 RTL 视图可知：左侧为输入，右侧为输出。位宽为 4 的输入控制通过多个比较器和多路复用器等元件控制 12 个输出，各个元件和引脚之间由导线相连。

结论：一个功能的实现需要多种元器件的彼此配合，简单的 Verilog 代码背后隐含复杂的执行电路。

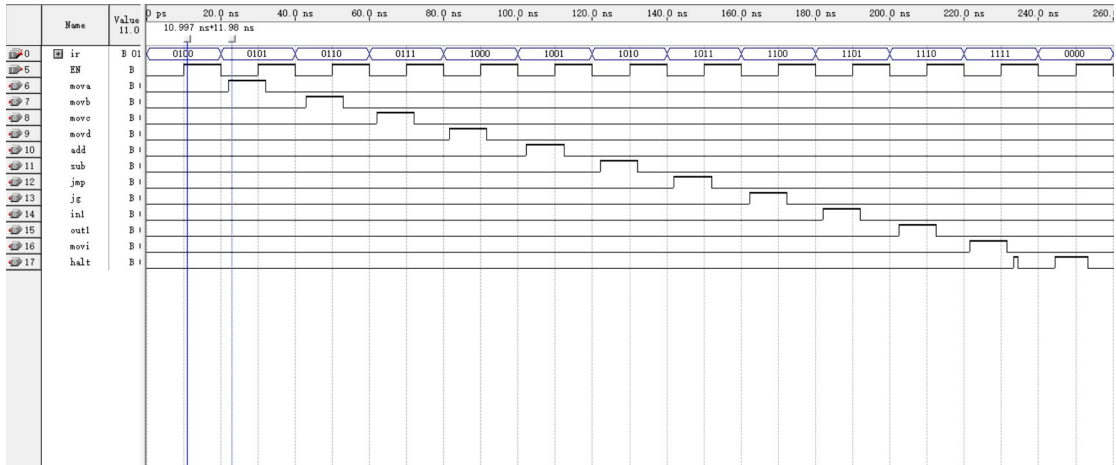
E) 功能仿真波形



结果分析及结论：对逻辑电路进行功能仿真，发现输出结果与真值表相同。功能仿真操作简单，能够验证电路基本功能的正确性。

F) 时序仿真波形

实验二 模型机组部件的实现（一）



结果分析及结论：时序仿真模拟了真实器件运行的仿真，不同元器件有不同的延迟，导致实际运行情况可能和功能仿真不同。由图可得，当输入状态改变时，输出结果并未同步改变，而是有一定的延迟。同时由于输入状态的改变，导致电路出现“冒险”现象，进一步使输出结果和预期结果有所出入。

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	15.757 ns	a[2]	movb	--	--	0	
2 Total number of failed paths								0	

tpd					
	Slack	Required P2P Time	Actual P2P Time	From	To
1	N/A	None	15.757 ns	i[2]	movb
2	N/A	None	15.603 ns	i[3]	movb
3	N/A	None	15.434 ns	i[0]	movb
4	N/A	None	15.279 ns	i[1]	movb
5	N/A	None	14.977 ns	i[3]	add
6	N/A	None	14.948 ns	i[2]	add
7	N/A	None	14.916 ns	i[3]	sub
8	N/A	None	14.884 ns	i[2]	sub
9	N/A	None	14.815 ns	i[2]	movb
10	N/A	None	14.738 ns	i[0]	add
11	N/A	None	14.676 ns	i[0]	sub
12	N/A	None	14.660 ns	i[3]	movb
13	N/A	None	14.638 ns	i[3]	jmp
14	N/A	None	14.623 ns	i[2]	jmp
15	N/A	None	14.613 ns	i[2]	in1
16	N/A	None	14.587 ns	i[1]	add
17	N/A	None	14.525 ns	i[3]	in1
18	N/A	None	14.521 ns	i[1]	sub
19	N/A	None	14.497 ns	i[0]	movb
20	N/A	None	14.475 ns	EN	halt
21	N/A	None	14.441 ns	i[0]	jmp
22	N/A	None	14.397 ns	i[2]	movi
23	N/A	None	14.342 ns	i[1]	movb
24	N/A	None	14.303 ns	i[1]	jmp
25	N/A	None	14.302 ns	i[3]	movi
26	N/A	None	14.289 ns	i[2]	movd
27	N/A	None	14.286 ns	i[0]	in1
28	N/A	None	14.229 ns	i[3]	halt
29	N/A	None	14.144 ns	i[1]	in1
30	N/A	None	14.132 ns	i[3]	movd
31	N/A	None	14.112 ns	i[0]	movi
32	N/A	None	14.002 ns	i[0]	movd
33	N/A	None	13.974 ns	i[1]	movi
34	N/A	None	13.857 ns	i[2]	out1
35	N/A	None	13.850 ns	i[1]	movd

结果分析及结论：由图可得，Timing Analyzer Summary 呈现了所有定时分析的结果，并向我们展示了每个定时特性的最坏情况定时。在实际情况下，不同元器件之间是存在时间延迟的，而且不同的元器件的延迟时间也不尽相同。

2、算术单元 AU

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

When you click Finish, the project will be created with the following settings:

Project directory:

D:/Great/au/

Project name:

au

Top-level design entity:

au

Number of files added:

0

Number of user libraries added:

0

Device assignments:

Family name:

Cyclone II

Device:

EP2C5T144C8

EDA tools:

Design entry/synthesis:

<None>

Simulation:

<None>

Timing analysis:

<None>

Operating conditions:

Core voltage:

1.2V

Junction temperature range:

0-85 癈

B) 编写源代码

```
1 module au(au_en, ac, a, b, t, gf);
2     input wire au_en;
3     input wire [3:0] ac;
4     input wire [7:0] a;
5     input wire [7:0] b;
6     output reg [7:0] t;
7     output reg gf;
8     always @ (au_en, ac, a, b)
9     begin
10         gf=1'b0;
11         t=8'b0000_0000;
12         if(au_en==1'b1)
13         begin
14             if(ac==4'b1000)
15             begin t=a+b; end
16             else if(ac==4'b1001)
17             begin
18                 t=b+(-a)+8'b0000_0001;
19                 if(t[7]==0)
20                 begin gf=1; end
21                 else if(t[7]==1)
22                 begin gf=1; end
23             end
24             else if(ac==4'b0100 || ac==4'b0101 || ac==4'b1101)
25             begin t=a; end
26             else
27             begin t=8'h22; end
28         end
29         else
30         begin t=8'h22; end
31     end
32 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）

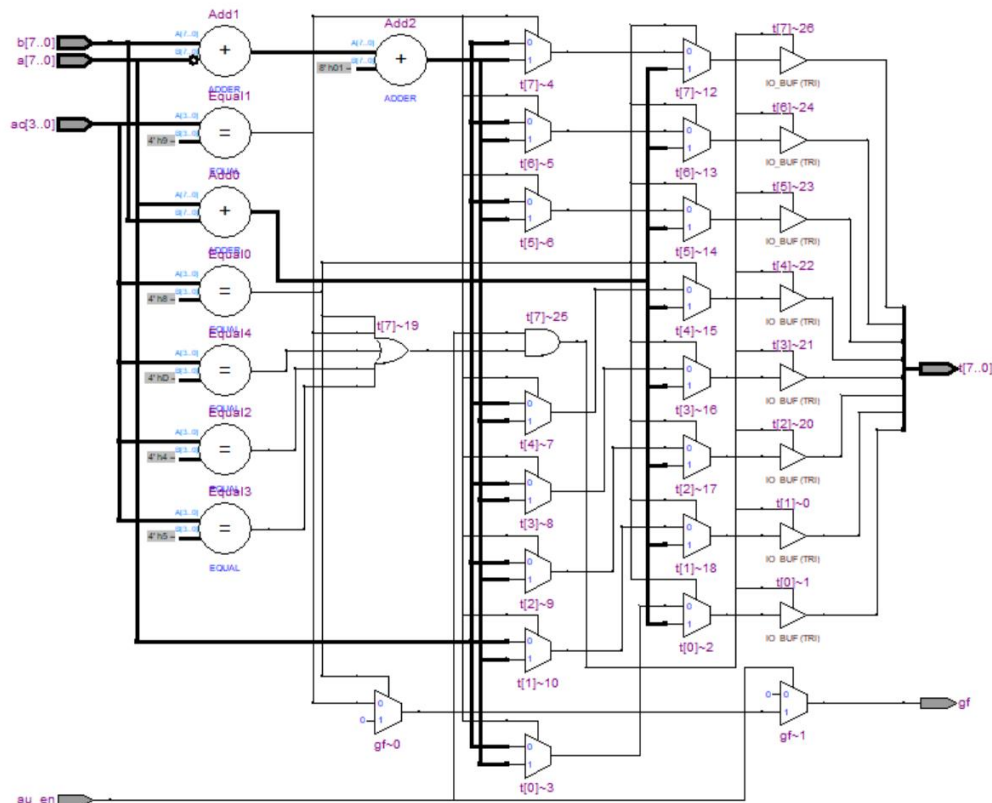
1. 对代码进行编译，出现 0 个错误、4 个警告。

Type	Message
	Info: Quartus II Assembler was successful. 0 errors, 0 warnings
	Info: *****
	Info: Running Quartus II Classic Timing Analyzer
	Info: Command: quartus_tan --read_settings_files=off --write_settings_files=off au -c au --timing_analysis_only
	Info: Longest tpd from source pin "ac[0]" to destination pin "t[7]" is 14.481 ns
	Info: Quartus II Classic Timing Analyzer was successful. 0 errors, 0 warnings
	Info: Quartus II Full Compilation was successful. 0 errors, 4 warnings

2. 资源消耗

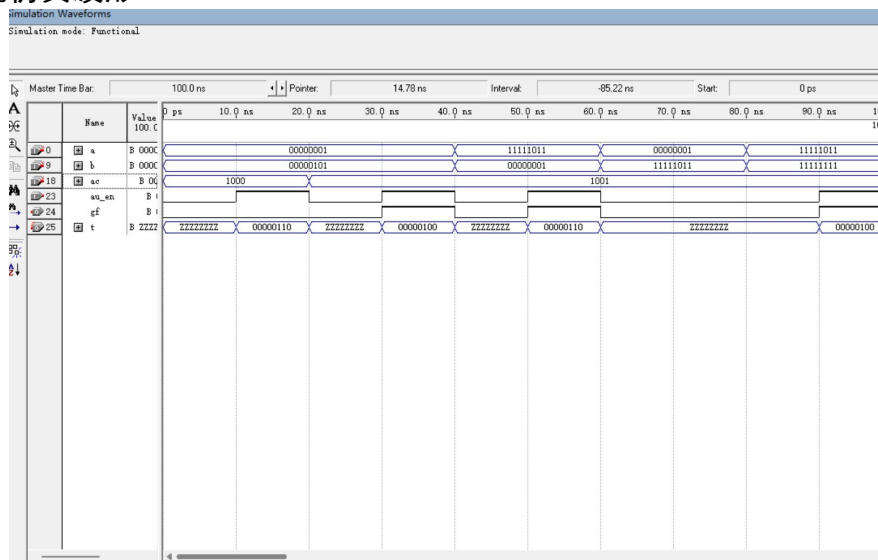
Flow Status	Successful - Fri Nov 22 21:32:31 2024
Quartus II Version	9.0 Build 184 04/29/2009 SP 1 SJ Web Edition
Revision Name	au
Top-level Entity Name	au
Family	Cyclone II
Met timing requirements	Yes
Total logic elements	30 / 4,608 ( < 1 % )
Total combinational functions	30 / 4,608 ( < 1 % )
Dedicated logic registers	0 / 4,608 ( 0 % )
Total registers	0
Total pins	30 / 89 ( 34 % )
Total virtual pins	0
Total memory bits	0 / 119,808 ( 0 % )
Embedded Multiplier 9-bit elements	0 / 26 ( 0 % )
Total PLLs	0 / 2 ( 0 % )
Device	EP2K5T144C6
Timing Models	Final

## D) RTL 视图



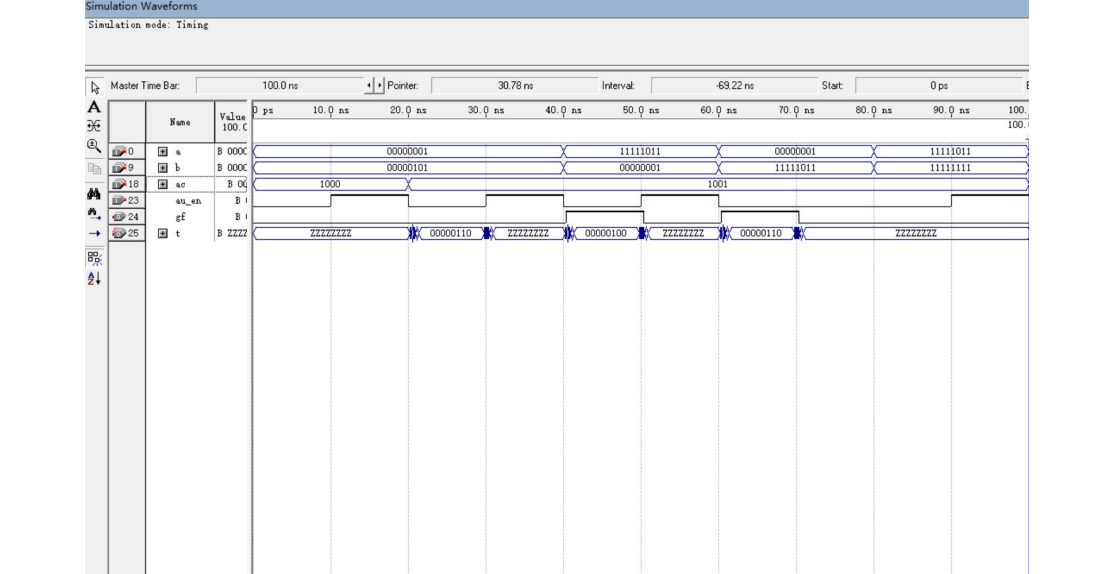
结果分析及结论：对 RTL 视图进行观察：左侧为输入引脚，右侧为输出引脚，电路中含有加法器、比较器、逻辑门、多路复用器等基本元器件。这表示一个简单的功能需要大量的基本元器件进行实现，原理十分复杂。

## E) 功能仿真波形



结果分析及结论：功能仿真是指在理想情况（不考虑元器件等的延时）下进行电路的仿真模拟。由仿真模拟波形图可得：对于输入状态的变化，其输出结果也实时变化，同时结果与设计的真值表相同。

F) 时序仿真波形



结果分析及结论：对电路进行时序仿真，观察波形图可以发现，在输入状态发生改变时，输出结果并未同步改变，而是有一定的延迟，并且出现了冒险现象。由此可知：元器件自身的延迟会影响整个电路的输出结果乃至功能。

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	14.481 ns	ac[0]	t[7]	--	--	0	
2 Total number of failed paths								0	



tpd					
	Slack	Required P2P Time	Actual P2P Time	From	To
1	N/A	None	14.481 ns	ac[0]	f[7]
2	N/A	None	14.332 ns	ac[3]	f[7]
3	N/A	None	14.214 ns	ac[1]	f[7]
4	N/A	None	14.145 ns	ac[2]	f[7]
5	N/A	None	13.746 ns	ac[0]	f[6]
6	N/A	None	13.653 ns	ac[0]	f[4]
7	N/A	None	13.597 ns	ac[3]	f[6]
8	N/A	None	13.579 ns	a[0]	f[7]
9	N/A	None	13.516 ns	b[0]	f[7]
10	N/A	None	13.504 ns	ac[3]	f[4]
11	N/A	None	13.479 ns	ac[1]	f[6]
12	N/A	None	13.454 ns	a[5]	f[7]
13	N/A	None	13.410 ns	ac[2]	f[6]
14	N/A	None	13.386 ns	ac[1]	f[4]
15	N/A	None	13.367 ns	a[3]	f[7]
16	N/A	None	13.353 ns	ac[0]	f[5]
17	N/A	None	13.317 ns	ac[2]	f[4]
18	N/A	None	13.267 ns	a[6]	f[7]
19	N/A	None	13.265 ns	a[2]	f[7]
20	N/A	None	13.238 ns	a[1]	f[7]
21	N/A	None	13.204 ns	ac[3]	f[5]
22	N/A	None	13.086 ns	ac[1]	f[5]
23	N/A	None	13.074 ns	ac[0]	f[3]
24	N/A	None	13.017 ns	ac[2]	f[5]
25	N/A	None	13.006 ns	a[7]	f[7]
26	N/A	None	12.990 ns	b[4]	f[7]
27	N/A	None	12.973 ns	a[4]	f[7]
28	N/A	None	12.958 ns	b[5]	f[7]
29	N/A	None	12.926 ns	b[1]	f[7]
30	N/A	None	12.925 ns	ac[3]	f[3]
31	N/A	None	12.920 ns	ac[0]	f[2]
32	N/A	None	12.862 ns	b[2]	f[7]
33	N/A	None	12.844 ns	a[0]	f[6]
34	N/A	None	12.813 ns	ac[0]	f[0]
35	N/A	None	12.807 ns	ac[1]	f[3]

结果分析及结论：由图可得，Timing Analyzer Summary 呈现了所有定时分析的结果，并向我们展示了每个定时特性的最坏情况定时。在实际情况下，不同元器件之间是存在时间延迟的，而且不同的元器件的延迟时间也不尽相同。

## 四、思考题

- 指令译码器必须要 12 个输出吗？可否将一些输出合并，哪些可以合并，为什么？  
比如 mova~movd 这四个指令，它们对应的输入信号都符合“01XX”的形式。可以为这四个指令单独构建一个 2-4 译码器，输出位宽为 4，有望减小电路成本。  
ADD 和 SUB 指令的输出也可以进行合并。两个指令的结果其实基本相同，只不过 SUB 指令附加了一个状态位 G。
- AU 中的 S[3..0]控制信号是来自哪里或者说与什么信息相同？  
来自于 Ir[3:0]。
- 为何 AU 算术单元不执行加、减运算和 MOVA、MOVB 和 OUT 指令的数据传送功能时，输出为高阻态？  
因为 AU 算术单元的输出端与其它输出单元共用一根总线，当 AU 在不执行相应的运算和数据传送功能时，输出高阻态从而与总线断开，避免对其它输出端造成影响。

## 五、实验总结、必得体会及建议

- 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。  
掌握了 quartus 软件的基本使用方法，能够使用 Verilog 语言编写简单的模块，实现简单的功能。了解了对电路进行仿真模拟的方法，同时理解了时序仿真和功能仿真的区别所在。（软件默认时序仿真，这一点让我印象深刻）
- 对本实验内容、过程和方法的改进建议（可选项）。  
很好，建议给 24 级加大强度。