

计算机组成原理

实验指导书

(Ver1.3)

计算机组成原理课程组 编著

重庆理工大学计算机学院

计算机科学与技术系

2024 年 12 月 3 日

目录

实验一 实验环境介绍-简单电路	2
一、 实验目的	2
二、 实验环境	2
三、 实验内容	2
四、 实验步骤	10
五、 实验报告	10
实验二 运算器组成实验.....	11
一、 实验目的	11
二、 实验环境	11
三、 实验内容	11
四、 实验步骤	15
五、 实验报告	15
实验三 存储器运算器综合实验	16
一、 实验目的	16
二、 实验环境	16
三、 实验内容	16
四、 实验步骤	17
五、 实验报告	17
实验四 控制器综合实验.....	18
一、 实验目的	18
二、 实验环境	18
三、 实验内容	18
四、 实验步骤	20
五、 实验报告	20
计算机组成原理课程设计	21
一、 课程设计目的	21
二、 课程设计环境	21
三、 课程设计内容	21
四、 课程设计步骤	22
五、 课程设计报告	22

实验一 实验环境介绍-简单电路

一、 实验目的

- (一) 熟悉和掌握 Quartus II 集成开发环境及使用方法;
- (二) 掌握基于图编辑器的顶层实体设计;
- (三) 掌握从设计输入到硬件配置的完整设计流程;
- (四) 掌握 Cyclone V 实验箱组成;
- (五) 熟悉使用 VHDL 硬件描述语言。

二、 实验环境

本实验的实验环境为 Altera Quartus II 软件和 Altera Cyclone V Starter 工具箱。实验平台要点如下：

- Cyclone V 是 Altera **FPGAs** 家族的一类，它面向低功耗应用。**Cyclone V GX 5CSEMA5F31C6** 芯片有不同的转换器，按钮，LED 灯，七段显示器，HDMI 接口，SD 卡读卡器以及其他链接等。**Cyclone V GX 5CSEMA5F31C6** 是 Starter 工具箱的重要组件。
- 测试台包含在测试设备（DUT）和用于仿真的模拟输入。这可以用于测试功能行为和计时行为。我们还可以采用测试台来分配 Starter 工具箱上相关信号的引脚名称，用 force 文件来包含仿真模拟输入。

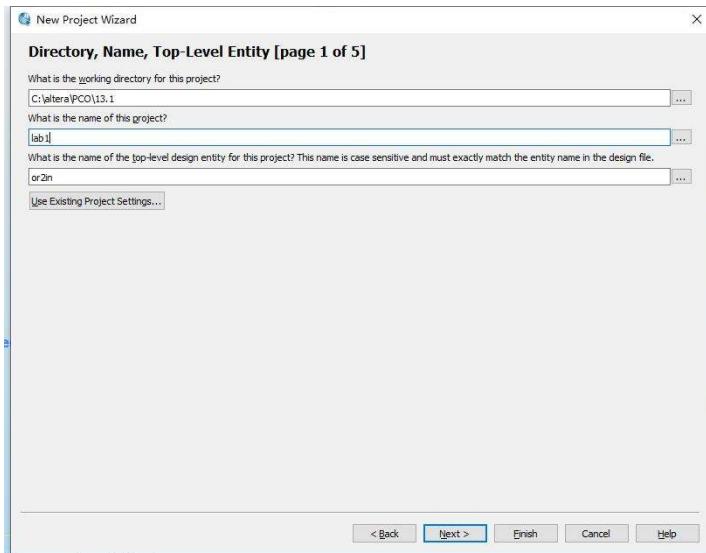
批注 [B1]: <https://baike.baidu.com/item/FPGA/935826?fr=aladdin>

三、 实验内容

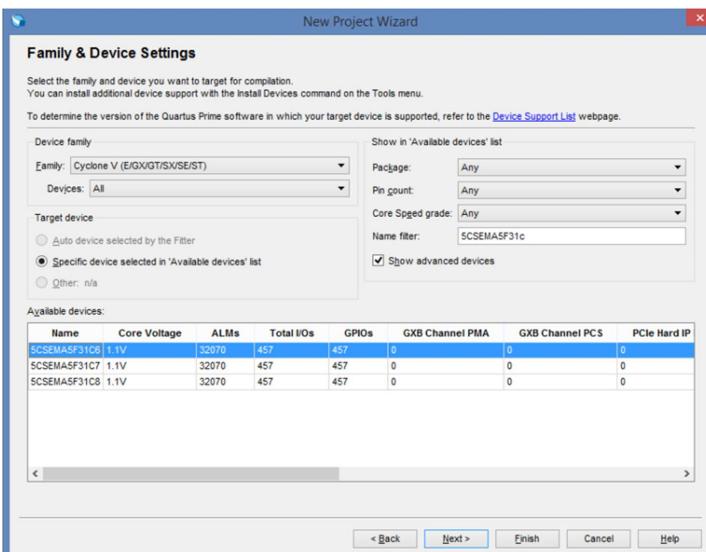
采用图编辑器设计实体：

在此过程中，你可以将各组件放在一张纸上，并将它们与虚拟电线连接起来以创建一个简单的数字电路。你可以仿真此电路来验证其行为是否正确。然后你可以在 Starter 工具箱上给 Cyclone V FPGA 板编程，最后演示电路的正确操作。

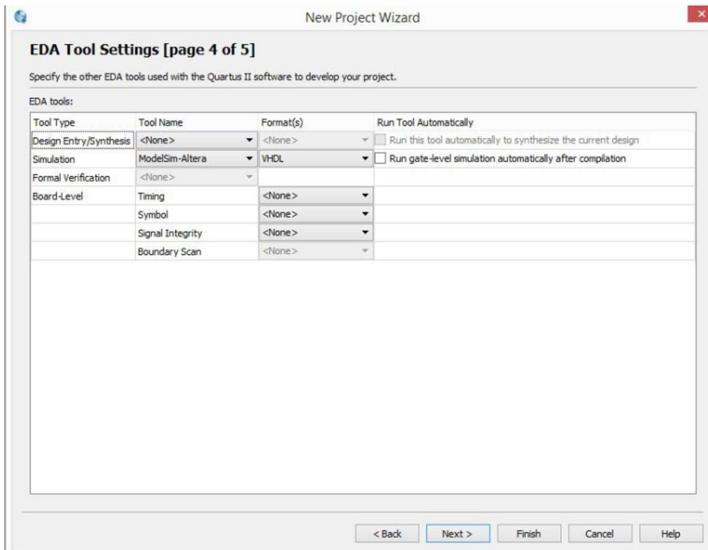
1. 完成以下功能：**ledr0 = sw0 OR sw1**
 - a) 创建一个工程
选择 File>New Project Wizard.... 创建工程，然后选择 Next>。
这个工程的工作目录是什么？ C:\altera\PCO\lab1
这个工程的名称是什么？ lab1
这个工程的顶层实体的名称是什么？ or2in



- b) 双击 Next 进入设备选择界面。在左边组件的组合框中，选择 Cyclone V，并在 Name filter 框的右侧填写你的 FPGA 板的正确设备名，即 **5CSEMA5F31C6**，再点击 Next。

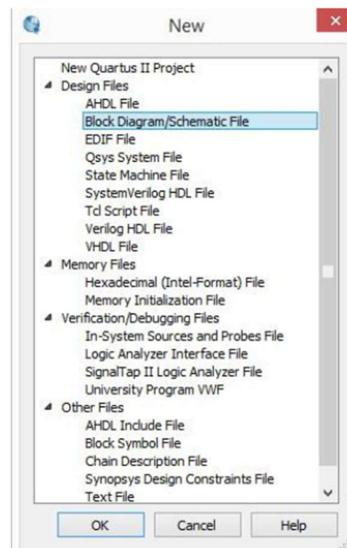


- c) 指定你要使用的 ModelSim 仿真器。在下一个窗口中，仿真器可以读取 Modelsim-Altera 和 VHDL。

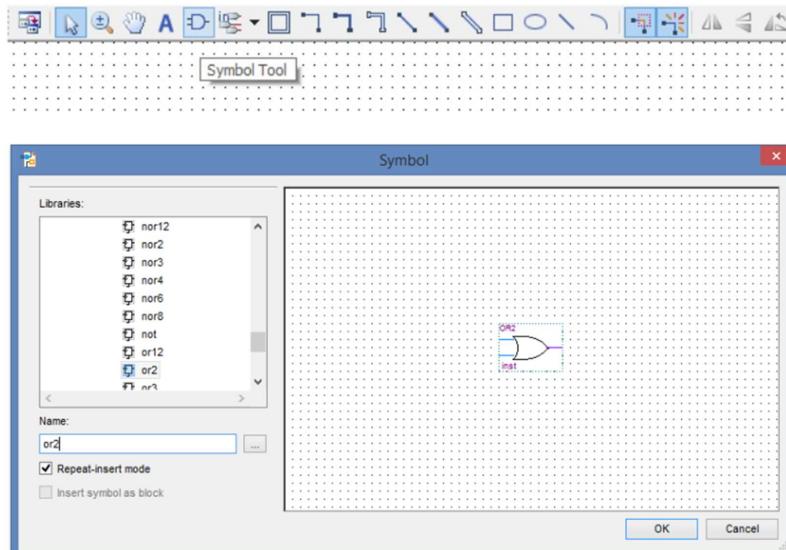


2. 进行设计

- a) 选择 File>New>Block Diagram/Schematic File 来创建一个新的 Block Diagram/Schematic 文件。



- b) 点击工作区左边的工具条上的添加符号图标添加一个新的门。你也可以在绘图区上通过双击来添加。在 Libraries: 下的列表框中，点击 C:/altera/quartus/libraries/左边的扩展符号 (+) 来展示可用库。先点开原始库，再点开逻辑库。



- c) 在逻辑库列表中选择 or2 符号，即可查看到 2 输入 OR 门，点击 OK。现在可以放置你的 or2 门。在工作区中点击某处来为你示意图中的 or2 门创建一个实例。
3. 为输入输出引脚命名
- 再次选择添加符号图标。打开 primitives/pin 库选择输出符号并点击 OK。将输出引脚放置到 or2 门的正确位置上。断开输出引脚的连接。重复上述步骤添加另外两个引脚。
 - 为了在 or2 符号上连接引脚符号，将鼠标放置到符号中某条电线的末尾，当光标变为+符号时，表明在光标的右下角有一个“+”。当光标显示为“+”时，点击并拖拽到你想连接的电线上。当你连接一条电线后，将会在电线的末尾显示一个小框。将光标放到小框中并释放鼠标按钮。采用这种方式，依次连接逻辑符号中的各个输入输出引脚。
 - 为了给引脚命名，右键点击引脚并选择 Properties。此时出现一个标记为 Pin name(s): 的文本框，你可以在此处输入引脚的名称。输入引脚名称后，点击 OK。
 - 当 switch0, switch1 某个或者都处于 on 位置时，你可以创建一个电路用于打开第一个红色发光二极管 ledr0。因此，将两个输入引脚的名称分别设置为 sw0 和 sw1，输出引脚设置为 ledr0。至此，你已完成了示意图的创建，将其保存为 or2in.bdf。下一步是将原理图中的引脚分配给 FPGA 上的对应物理引脚。
4. 编译设计
- 在编译之前，通过以下步骤“File→Create/update→Create HDL Design File from Current File”将 bdf 文件转换为 VHDL 文件。然后通过“Project→Add/remove files in project”来删除列表中的 or2in.bdf 文件，并添加生成的 or2in.vhd 文件。
 - 保存你的设计，然后使用 Processing>Start Compilation 命令将其编译。编译过程会花一些时间。你可以在左侧的状态窗口观察编译过程。当编译结束时

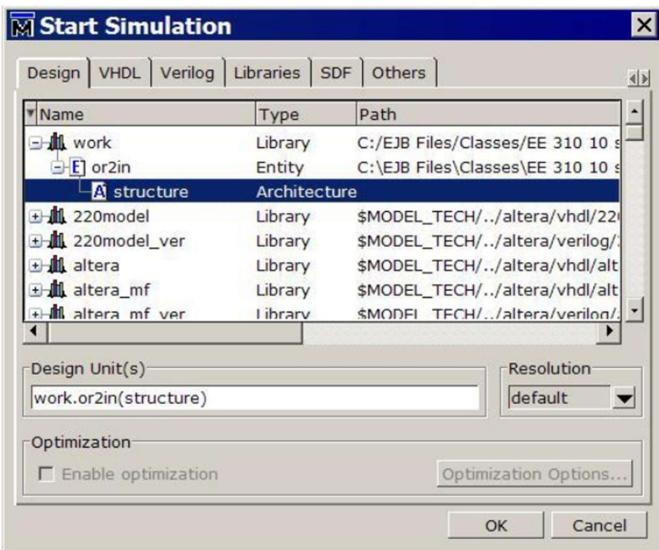
会弹出对话框。此时可以无视一切警告。

5. 硬件引脚分配

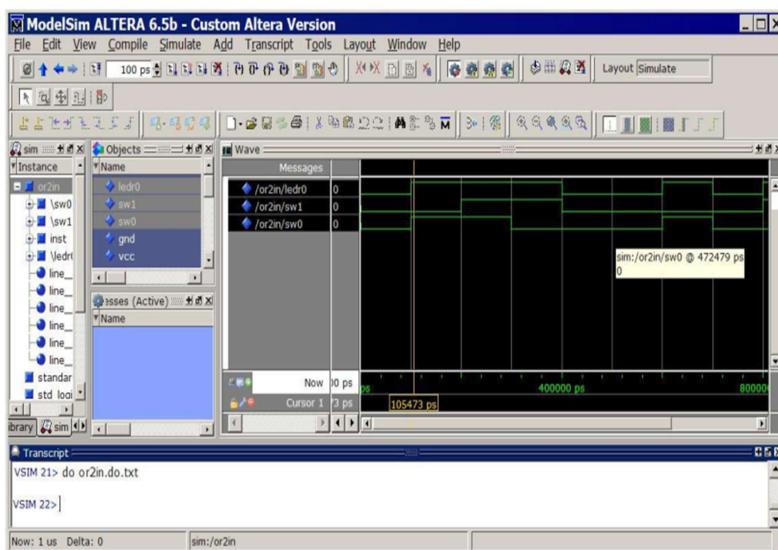
- a) 在菜单中选择 Assignments>Pin planner 打开引脚分配对话框。将引脚 PIN_AB12 和 PIN_AC12 分别分配给 sw0 和 sw1, PIN_V16 分配给 ledr0。分配过程如下：在列表中查找预先定义的符号引脚名（在 Node Name 列），双击各个引脚的 Location 列以打开可用的物理引脚下拉框。选择一个引脚会把它分配给它的右边的网。你也可以键入引脚名。现在关闭 Pin Planner 面板。
- b) 注意：在 BBLearn 界面中的文件 DE1-SoC_User_manual.pdf 中可查看引脚分配详细列表。
- c) 要点：在为各个节点设置恰当的物理引脚后，应该将其他所有引脚都设置为输入，这样它们就不会与板上的任何其他设备发生冲突。操作如下：选择 Assignments>Device...，然后点击 Device 和 Pin Options...选择 Unused Pins 选项卡。在组合框 Reserve all unused pins 中选择 As input, tri-stated。点击两次 OK 即可。
- d) 保存并再次编译。此时如无错误，有一个警告“Warning: Found 1 output pins without pin load capacitance assignment”（在编译完成后，你可以点击警告选项卡来检查），表明编译成功。下一步是仿真设计以验证设计的正确性。

6. 仿真设计

- a) 点击“Tools→Run simulation tool→RTL simulation”打开 ModelSim。在 ModelSim 主窗口，点击“File→Change directory”将目录切换到工程文件夹下，然后选择路径“C:\altera\PCO\lab1\simulation\modelsim”并点击 OK。
- b) 点击“Compile→Compile”，在 ModelSim 文件夹下选择 or2in.vhd（从 Quartus 目录复制过来）进行编译。
- c) 运行电路仿真时，需要为输入信号指定波形。该过程涉及以下简单命令：
 1. # force file for two-input logic gate
 2. # comment: need a # followed by a space
 3. force sw0 0;# comment: after a command, need a semicolon
 4. force sw1 0
 5. run 100ns
 6. force sw0 1
 7. run 100ns
 8. force sw1 1
 9. run 100ns
 10. force sw0 0
 11. run 100ns
 12. force sw1 0
 13. run 100ns
- d) 在 simulation\modelsim 目录下的文本文件 or2in.do 中输入上述命令。输入过程中请注意代码中注释的格式。
- e) 通过 Simulation>Start Simulation 进入仿真模式。在设计选项卡中，展开 work 扩展项下面的 or2in 实体下的 architecture structure 项，点击 OK。

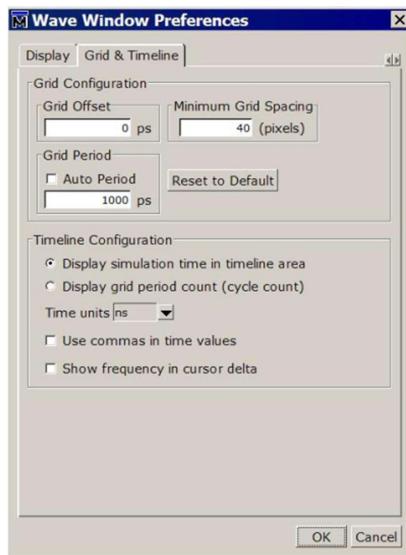


- f) 在 Objects 窗口中，选择 3 个引脚 (ledr0, sw0 和 sw1) 并拖拽到 Wave 窗口中。在命令窗中输入 do 或者 or2in.do 命令强制运行文件，即可在 Wave 窗口中看到波形。
- g) 注意：如果无法看到正确波形，右键点击波形窗口选择“Zoom full”或 按“F”键。
- h) 注意整个过程应无延迟，输入一变化输出就响应。这就是逻辑门被假定为理想的零延迟的函数仿真。



- i) 可以在 Wave 窗口中修改时间刻度。激活 Wave 框，选择 Wave>Wave

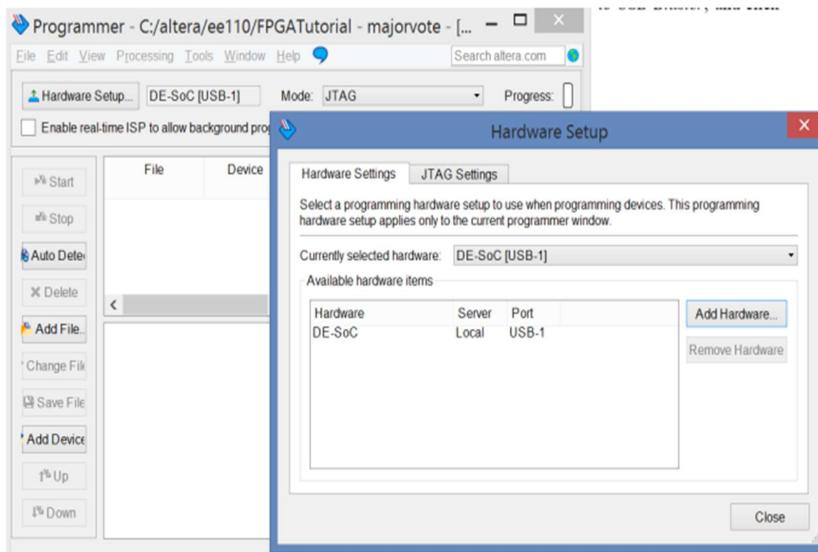
preferences...并选择 Grid&Timeline 选项卡，将时间单位变为 ns（纳秒）。点击 OK。



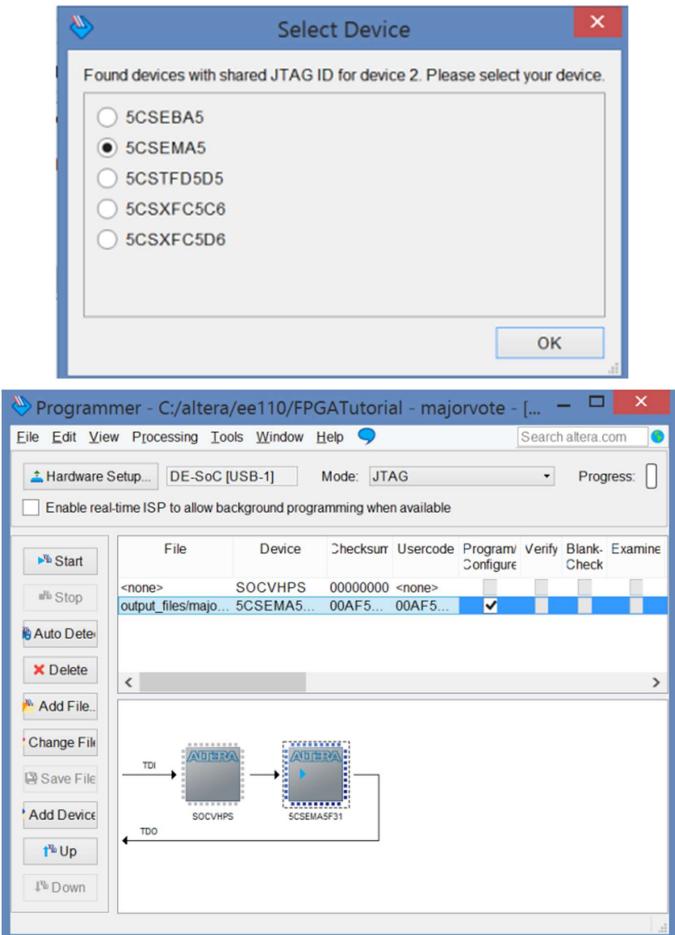
j) 在 Wave 窗口中保存波形，关闭 ModelSim。

7. 编程 FPGA 板

- 最后，采用在下拉菜单 Tools 下的 Programmer 工具对 FPGA 板进行编程。在编程过程中，请务必接入你的 FPGA 板并开启电源。
- 弹出 Programmer 窗口后，点击左上角的 Hardware Setup 按钮。然后将“Currently selected hardware”改变为 USB-Blaster，点击关闭。



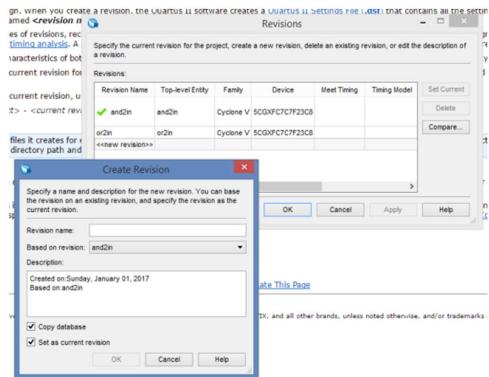
- c) 点击工具条左侧的 Auto Detect，选择 5CSEMA5 并点击 OK。右键点击 5CSEMA5，选择“Change File”，在路径“C:\altera\PCO\lab\output_files”下选择 or2in.sof 文件，并点击“Open”。最后，选择 Program/Configuration 框。



- d) 最后，点击 Start 按钮对设备进行编程，如此，就可以将你设计的电路编程到 FPGA 板上了！

8. 在工程中创建一个 revision

- a) revision 是一个设计中为设计文件定义的一组设置和分配。你可以采用不同的 revision 来定义不同的设置和分配以决策他们如何影响处理设计的结果。例如：你可以采用多个 revision 来为同一个设备编译设计，这个设备具有不同的默认逻辑选项和时间分析设置。或者，也可以采用两个 revision 来为两台保持所有其他设置和分配不变的不同的设备编译同一个设计。
- b) 为当前工程创建一个新的 revision，可以选择“Project->Revision”，在弹出的窗口中双击《new revision》。然后为创建的新 revision 选择基础 revision 并命名，点击 OK。



- c) 你可以按照上图选择当前 revision (绿色标记为当前选择的 revision)
- d) 为了仿真或下载当前 revision, 需要将当前 revision 设置为顶层实体。操作方法为: “Project->Set as top-level entity”。

四、 实验步骤

- (1) 阅读说明书 “Introduction to VHDL.doc” 和 “DE1-SoC_User_manual_rev.FG.pdf”，对 Quartus II 集成开发环境有基本的了解；
- (2) 利用图编辑器创建顶层实体 AND 门和 D 触发器，生成对应 VHDL 脚本；
- (3) 利用 VHDL 语言编写 AND 门和 D 触发器，生成相应图形块；
- (4) 全程编译，创建 VWF 文件或者编写 ModelSim 的 do 文件对电路进行仿真；
- (5) 根据 Cyclone V 指导书，为创建的电路分配物理引脚，锁定输入输出引脚，完成全程编译；
- (6) 编程下载。下载 SOF 文件至 FPGA，将你的电路下载到 FPGA 板中，并进行功能验证。

五、 实验报告

实验报告包含：

- (1) 实验目的；
- (2) 各模块的设计电路和系统的整体电路,对设计要进行详细的分析与说明；
- (3) 实验结果的记录与分析；
- (4) 列出操作步骤及顺序,标出重要的开关控制端；
- (5) 实验收获和体会；
- (6) 实验中碰到的问题和解决的方法。

实验二 运算器组成实验

一、 实验目的

1. 掌握运算器基本工作原理;
2. 掌握运算溢出检测的原理和实现方法;
3. 熟悉运算器的数据传输通路;
4. 利用 VHDL 语言实现 AC 寄存器;
5. 利用 VHDL 语言实现 74LS181 的算术运算及逻辑运算功能;
6. 掌握 Quartus II 的编译及仿真过程;
7. 在 Cyclone V FPGA 板上进行功能演示。

二、 实验环境

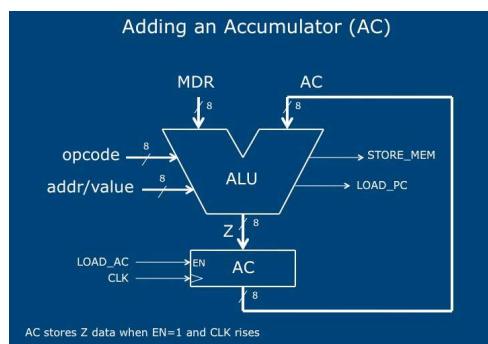
本实验的实验环境为 Altera Quartus II 软件和 Altera Cyclone V Starter 工具箱。实验平台要点如下：

- Cyclone V 是 Altera FPGAs 家族的一类，它面向低功耗应用。Cyclone V GX 5CSEMA5F31C6 芯片有不同的转换器，按钮，LED 灯，七段显示器，HDMI 接口，SD 卡读卡器以及其他链接等。Cyclone V GX 5CSEMA5F31C6 是 Starter 工具箱的重要组件。
- 测试台包含在测设备（DUT）和用于仿真的模拟输入。这可以用于测试功能行为和计时行为。我们还可以采用测试台来分配 Starter 工具箱上相关信号的引脚名称，用 force 文件来包含仿真模拟输入。

三、 实验内容

1. AC 寄存器实验

AC 寄存器可用来存储 ALU 的输入以实现两个操作数的运算操作，也可以用来存储来自 ALU 的结果，以将这些结果反馈给 ALU 以进行更多的处理。AC 寄存器和 ALU 的连接如下：



在本实验中，只需要使用寄存器来存储 ALU 的两个输入，ALU 的输出直接连接到数码管显示。从图中可以看出，AC 寄存器需要一条控制线：LOAD_AC。在此控制线的上升沿，触发将 Z 总线的数据存储到 AC 寄存器中的操作。AC 总线总是显示当前存储在 AC 寄存器中的值。CLK 为时钟输入。请使用 VHDL 语言创建 AC 寄存器。

2. 74LS181 试验

74181ALU 的运算逻辑如表 1 所示：

表.1 74181ALU 运算功能表

控制方式		M=1	M=0 算术运算	
S3	S2 S1 S0	逻辑运算	CN=1 (无进位)	CN=0 (有进位)
0 0 0 0		F=A	F=A	F=A 加 1
0 0 0 1		F=(A+B)	F=A+B	F=(A+B)加 1
0 0 1 0		F=(/A).B	F=A+/B	F=(A+/B)加 1
0 0 1 1		F=0	F=0 减 1	F=0
0 1 0 0		F=(A.B)	F=A 加 A.(/B)	F=A 加 A./B 加 1
0 1 0 1		F=/B	F=(A+B)加 A./B	F=(A+B)加 A/B 加 1
0 1 1 0		F=A \oplus B	F=A 减 B 减 1	F=A 减 B
0 1 1 1		F=A./B	F=A/(B)减 1	F=A/(B)
1 0 0 0		F=A+B	F=A 加 AB	F=A 加 AB 加 1
1 0 0 1		F=(A \oplus B)	F=A 加 B	F=A 加 B 加 1
1 0 1 0		F=B	F=(A+/B)加 AB	F=(A+/B)加 AB 加 1
1 0 1 1		F=AB	F=AB 减 1	F=AB
1 1 0 0		F=1	F=A 加 A	F=A 加 A 加 1
1 1 0 1		F=A+/B	F=(A+B)加 A	F=(A+B)加 A 加 1
1 1 1 0		F=A+B	F=(A+/B)加 A	F=(A+/B)加 A 加 1
1 1 1 1		F=A	F=A 减 1	F=A

(上表中的 “/” 表示求反)

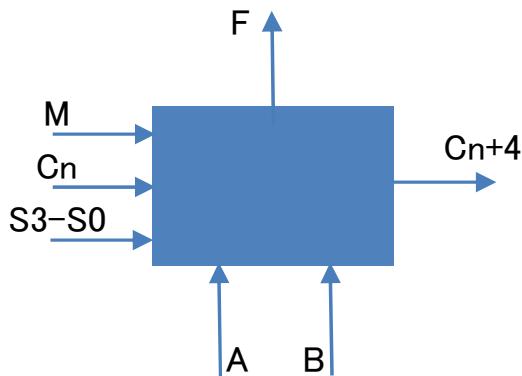
根据 74181ALU 的运算逻辑，本实验的 8 位 ALU 逻辑如表 2 所示：

表.2 实验二 ALU 运算功能表

方式	$M=1$	$M=0$
S3 S2 S1 S0	逻辑运算	CN=1 (无进位) 算术运算
0 0 0 0	$F=/A$	$F=A$
0 0 0 1	$F=/A+B$	$F=A+B$
0 0 1 0	$F=(/A).B$	$F=A+/B$
0 0 1 1	$F=0$	$F=0$
0 1 0 0	$F=(A.B)$	$F=A$ 加 $A.(/B)$
0 1 0 1	$F=/B$	$F=(A+B)$ 加 $A./B$
0 1 1 0	$F=A \oplus B$	$F=A$ 减 B
0 1 1 1	$F=A./B$	$F=A(/B)$
1 0 0 0	$F=/A+B$	$F=A$ 加 AB
1 0 0 1	$F=(A \oplus B)$	$F=A$ 加 B
1 0 1 0	$F=B$	$F=(A+/B)$ 加 AB
1 0 1 1	$F=AB$	$F=AB$
1 1 0 0	$F=1$	$F=A$ 加 A
1 1 0 1	$F=A+/B$	$F=(A+B)$ 加 A
1 1 1 0	$F=A+B$	$F=(A+/B)$ 加 A
1 1 1 1	$F=A$	$F=A$

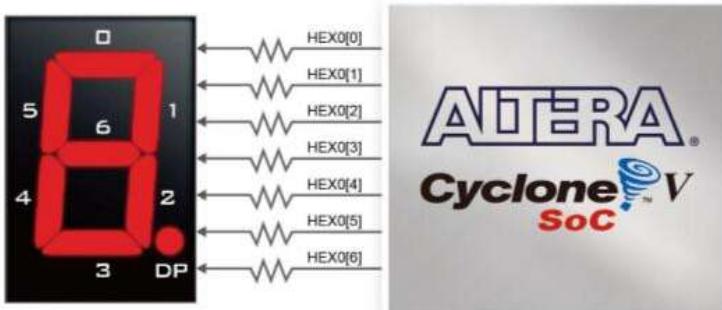
(注意: 上表中的“/”表示求反, “+”表示“或”, “加”表示“算数加”)

请根据上述表格, 在 Altera Quartus II 平台中利用 VHDL 语言, 构建一个 8 位运算器, 该运算器可以实现运算逻辑表的所有功能。其电路封装基本抽象如下图所示。X 和 Y 是两个操作数, Z 是运算结果。S3-S0 是运算操作符, 其组合与运算的对应关系见表 2.1。



3. 七段数码管实验

Cyclone V SoC FPGA 板上上有 3 对七段数码管用于数字显示。下图给出了七段数码管在 Cyclone V SoC FPGA 板上的引脚编号：



设计好的 8 位运算器的输出需要接到七段数码管来显示运算结果。8 位运算器的输出需要两个数码管来显示，每一个显示 4 位二进制数。请根据文档“DE1-SoC_User_manual_rev.FG.pdf”为每一个数码管的 7 个段分配物理引脚，分配时注意高低位顺序。

4. 仿真方法 2

新建 new→vector waveform file，出现的画面左侧空白双击，进入 insert node or bus →点击 node finder→pins : all→list 添加所有引脚。

设定各个输入端引脚值。(选择管脚→点击左侧带问号→ 设定各个端口的输入值。

设定仿真时间 edit→end time。

保存为 .vWF 文件。Simulation→Run Functional Simulation/Run Timing Simulation。

四、 实验步骤

- (1) 阅读说明书 “Introduction to VHDL.doc” 和 “DE1-SoC_User_manual_rev.FG.pdf”，对 Quartus II 集成开发环境有基本的了解；
- (2) 利用 VHDL 语言创建 AC 寄存器、8 位 ALU 和七段数码管；
- (3) 将创建的 AC 寄存器、8 位 ALU 和七段数码管转换为相应图形块，并进行连接；
- (4) 全程编译，创建 VWF 文件或者编写 ModelSim 的 do 文件对电路进行仿真；
- (5) 根据 Cyclone V 指导书，为创建的电路分配物理引脚，锁定输入输出引脚，完成全程编译；
- (6) 编程下载。下载 SOF 文件至 FPGA，将你的电路下载到 FPGA 板中，并进行功能验证。

五、 实验报告

实验报告包含：

- (1) 实验目的；
- (2) 各模块的设计电路和系统的整体电路,对设计要进行详细的分析与说明；
- (3) 实验结果的记录与分析；
- (4) 列出操作步骤及顺序,标出重要的开关控制端；
- (5) 实验收获和体会；
- (6) 实验中碰到的问题和解决的方法。

实验三 存储器运算器综合实验

一、 实验目的

1. 熟悉数据通路构成；
2. 了解 MegaWizard Plug-in Manager，学会用其创建元件；
3. 掌握 lpm_counter 的设置，作为计数器的工作特性和配置方法；
4. 掌握 lpm_latch 的设置，作为锁存器的工作特性和配置方法；
5. 掌握 ROM 的设置，作为只读存储器 ROM 的工作特性和配置方法；
6. 掌握用文本编辑器编辑 mif 文件，学习将程序代码以 mif 格式文件加载于 ROM 中；
7. 掌握 RAM 的功能、参数设置和使用方法。

二、 实验环境

本实验的实验环境为 Altera Quartus II 软件和 Altera Cyclone V Starter 工具箱。实验平台要点如下：

- Cyclone V 是 Altera FPGAs 家族的一类，它面向低功耗应用。Cyclone V GX 5CSEMA5F31C6 芯片有不同的转换器，按钮，LED 灯，七段显示器，HDMI 接口，SD 卡读卡器以及其他链接等。Cyclone V GX 5CSEMA5F31C6 是 Starter 工具箱的重要组件。
- 测试台包含在测设备（DUT）和用于仿真的模拟输入。这可以用于测试功能行为和计时行为。我们还可以采用测试台来分配 Starter 工具箱上相关信号的引脚名称，用 force 文件来包含仿真模拟输入。

三、 实验内容

1、FPGA 中 ROM 定制与读出实验

ALTERA 的 FPGA 中有许多可调用的模块库，可构成如 ROM、RAM、FIFO 等存储器结构。CPU 中的重要部件，如 RAM、ROM 可直接调用他们构成，因此在 FPGA 中利用 MegaWizard Plug-in Manager 可以构成各种结构的存储器，ROM 是其中的一种。由于 ROM 是只读存储器，所以它的数据口是单向的输出端口，ROM 中的数据是在对 FPGA 现场配置时，通过配置文件一起写入存储单元的。本实验只需要 ROM 有 3 组信号：clock —— 输入时钟脉冲；q[31..0] —— ROM 的 32 位数据输出端；address[7..0] —— ROM 的 8 位读出地址。实验中主要应掌握以下三方面的内容：

- (1) ROM 的参数设置，数据位宽和地址位宽；
- (2) ROM 中数据的写入，即 FILE 初始化文件的编写。编辑 ROM 配置文件（文件名.mif）。在这里预先给出后面将要用到的指令存储器初始化文件：ROM_RAM.mif。
- (3) ROM 的实际应用，将创建好的 ROM 连接到电路中。

2、FPGA 中 RAM 读写实验

在 FPGA 中利用 MegaWizard Plug-in Manager 可以构建 RAM 存储器，在 RAM 的左边有 8 位数据输入 $data[7..0]$ ，右边有 8 位数据输出 $q[7..0]$ ，数据位宽可以根据需要自行设置。 $wren$ 为读/写控制信号端。数据的写入：当输入数据和地址准备好以后， $clock$ 是地址锁存时钟，当信号上升沿到来时，地址被锁存，数据写入存储单元。数据的读出：从 $address[7..0]$ 输入存储单元地址，在 $clock$ 信号上升沿到来时，该单元数据从 $q[7..0]$ 输出。 $wren$ 读/写控制端，低电平时进行读操作，高电平时进行写操作； $clock$ —读/写时钟脉冲； $data[7..0]$ —RAM 的 8 位数据输入端； $address[7..0]$ —RAM 的读出和写入地址； $q[7..0]$ —RAM 的 8 位数据输出端。实验中主要掌握的内容与 ROM 类似。

四、实验步骤

- (1) 阅读说明书“Introduction to VHDL.doc”和“DE1-SoC_User_manual_rev.FG.pdf”，对 Quartus II 集成开发环境有基本的了解；
- (2) 利用 MegaWizard Plug-in Manager 创建计数器模块，锁存器模块，ROM 模块和 RAM 模块。
- (3) 利用实验二封装好的运算器、ROM 模块、RAM 模块、锁存器模块、计数器模块和显示模块模块构建一个运算电路，该电路由时钟驱动，可将 ROM 和 RAM 中存储的数据根据计数器自动完成相应运算，并将计算的结果根据锁存器提供的地址存放回 RAM 中。
- (4) 利用三组七段数码管分别显示当前 ROM 数据输出端的低八位内容、RAM 数据输出端的内容和运算结果（RAM 数据输入端的内容）。
- (5) 全程编译，创建 VWF 文件或者编写 ModelSim 的 do 文件对电路进行仿真；
- (6) 根据 Cyclone V 指导书，为创建的电路分配物理引脚，锁定输入输出引脚，完成全程编译；
- (7) 编程下载。下载 SOF 文件至 FPGA，将你的电路下载到 FPGA 板中，并进行功能验证；
- (8) 在系统读写。打开 QuartusII 的在系统存储模块读写工具 In-system Memory Content Editor，了解 FPGA 中 ROM 中的数据；
- (9) 利用系统读写 RAM 的工具对其中的数据进行读、写、修改、加载新的数据文件操作。

五、实验报告

实验报告包含：

- (1) 实验目的；
- (2) 各模块的设计电路和系统的整体电路，对设计要进行详细的分析与说明；
- (3) 实验结果的记录与分析；
- (4) 列出操作步骤及顺序，标出重要的开关控制端；
- (5) 实验收获和体会；
- (6) 实验中碰到的问题和解决的方法。

实验四 控制器综合实验

一、 实验目的

1. 理解总线系统的原理和作用。
2. 进一步加深对运算器、存储器及时序电路的理解。
3. 了解 MegaWizard Plug-in Manager，学会用其创建元件；
4. 掌握 lpm_mux 的设置，作为多路选择器的工作特性和配置方法；
5. 利用 VHDL 语言实现 PC 寄存器；
6. 利用 VHDL 语言实现译码器；
7. 掌握硬布线控制器设计原理；
8. 为整机实验以及课程设计做准备。

二、 实验环境

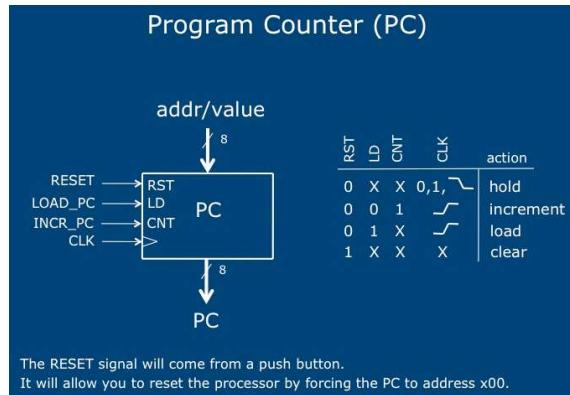
本实验的实验环境为 Altera Quartus II 软件和 Altera Cyclone V Starter 工具箱。实验平台要点如下：

- Cyclone V 是 Altera FPGAs 家族的一类，它面向低功耗应用。Cyclone V GX 5CSEMA5F31C6 芯片有不同的转换器，按钮，LED 灯，七段显示器，HDMI 接口，SD 卡读卡器以及其他链接等。Cyclone V GX 5CSEMA5F31C6 是 Starter 工具箱的重要组件。
- 测试台包含在测设备（DUT）和用于仿真的模拟输入。这可以用于测试功能行为和计时行为。我们还可以采用测试台来分配 Starter 工具箱上相关信号的引脚名称，用 force 文件来包含仿真模拟输入。

三、 实验内容

1. 程序计数器 PC 实验

程序计数器（PC）事实上是一个可以被清除（存储 0）8 位宽的顺序设备。当加载一个新的值或者增量（在当前值上加 1），由于 PC 包含指令的内存地址，我们将使用它的内容作为地址发送到内存。有 3 条新的控制线。RESET 线驱动处理器寻址 X00 并开始执行。因此，当 RESET 线被断言时，PC 必须马上被清除。如果未断言 RESET，则 PC 将保存其当前内容，直到 CLK 线的上升沿出现。在上升沿，如果 INCR_PC 线是 HIGH，如果 LOAD_PC 线被清除(0)，则 PC 增加。否则，如果 LOAD_PC 线被设置 (1)，则 PC 加载地址/值总线的数据。请参阅下图示中的函数表。



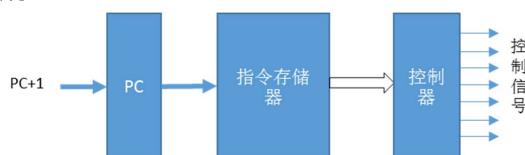
2. 译码器实验

构建如下指令格式的 32 位指令系统，设计一段测试程序存储在 ROM 存储器中，数据存放在 RAM 存储器中。

OP (4bits)		ADDR (28bits)
操作码	助记符	RTL 功能描述
0000	NOP	空操作
0001	JMP ADDR	无条件跳转
0010	SUB ADDR	$AC \leftarrow (AC) - (Mem[ADDR])$
0011	LAD ADDR	$AC \leftarrow (Mem[ADDR])$
0100	AND ADDR	$AC \leftarrow (AC) \& (Mem[ADDR])$
0101	OR ADDR	$AC \leftarrow (AC) + (Mem[ADDR])$
0110	ADD ADDR	$AC \leftarrow (AC) + (Mem[ADDR])$
0111	STD ADDR	$(Mem[ADDR]) \leftarrow AC$
...
XXXX		待扩展

3. 控制器实验

利用创建好的 PC、译码器和实验三的 ROM 存储器按照下图构建取指令通路（PC 寄存器 8 位），其中 PC 由时钟驱动，每个时钟自动完成取值以及 $PC=PC+1$ 的功能，控制存储器在后续步骤中实现。



将取指令数据通路与实验二封装的运算器、AC 寄存器和实验三封装的 RAM 连接起来可以利用电路从 ROM 中取值，从 RAM 中取数据，利用 ALU 进行运算，从而实现程序清单。结果可以用 3 组七段数码管显示。由于结果有不同的来源：RAM 或 AC，因此，可利用 MegaWizard Plug-in Manager 创建 lpm_mux 多路选择器，用于选取不同的数据来源。

四、 实验步骤

实验可按照自己设计的电路或参考电路按照搭积木的方式进行。先完成取指通路、再完成运算器通路部分，最后完成控制器，联合调试简单机器系统。

- (1) 阅读说明书 “Introduction to VHDL.doc” 和 “DE1-SoC_User_manual_rev.FG.pdf”，对 Quartus II 集成开发环境有基本的了解；
- (2) 利用 MegaWizard Plug-in Manager 创建计数器模块，锁存器模块，ROM 模块和 RAM 模块。
- (3) 利用实验二封装好的运算器、实验三封装的 ROM 模块、RAM 模块、AC 寄存器模块、PC 程序计数器模块、多路选择器模块和显示模块模块构建一个简单控制电路，该电路由时钟驱动，可从 ROM 中取值，按照程序清单自动执行相应功能，并将结果根据根据多路选择器的数据来源显示在七段数码管中。
- (4) 利用三组七段数码管分别显示当前 AC 寄存器的内容、RAM 数据输出端的内容和 ALU 的内容。
- (5) 全程编译，创建 VWF 文件或者编写 ModelSim 的 do 文件对电路进行仿真；
- (6) 根据 Cyclone V 指导书，为创建的电路分配物理引脚，锁定输入输出引脚，完成全程编译；
- (7) 编程下载。下载 SOF 文件至 FPGA，将你的电路下载到 FPGA 板中，并进行功能验证；
- (8) 在系统读写。打开 QuartusII 的在系统存储模块读写工具 In-system Memory_Content Editor，了解 FPGA 中 ROM 中的数据；
- (9) 利用系统读写 RAM 的工具对其中的数据进行读、写、修改、加载新的数据文件操作。

五、 实验报告

- (1) 实验目的；
- (2) 各模块的设计电路和系统的整体电路，对设计要进行详细的分析与说明；
- (3) 列出操作步骤及顺序，标出重要的开关控制端；给出各控制信号逻辑表达式以及电路。
- (4) 实验结果的记录与分析；
- (5) 列出操作步骤及顺序，标出重要的开关控制端；
- (6) 实验收获和体会；
- (7) 实验中碰到的问题和解决的方法。

计算机组成原理课程设计

一、课程设计目的

本课程设计综合利用计算机组成原理课程所学的理论知识，并结合其单元实验中所积累的计算机部件设计和调试方法，设计出一台具有自定义指令系统的简单计算机系统。所设计的系统能在基于 FPGA 的实验平台上运行一段程序，通过检查程序结果的正确性来判断所设计的计算机系统的正确性。

本课程设计属于设计型实验，不仅锻炼学生简单计算机系统的设计能力，而且通过进行单机系统底层电路的实现、故障分析与定位、系统调试等环节的锻炼，进一步提高学生分析和解决问题的能力。

课程设计要求学生具备设计数字电路的基本能力，并能较熟练地通过网络检索相关参考资料，了解计算机硬件系统设计的方法等。

二、课程设计环境

本课程设计的实验环境为 Altera Quartus II 软件和 Altera Cyclone V Starter 工具箱。实验平台要点如下：

- Cyclone V 是 Altera FPGAs 家族的一类，它面向低功耗应用。Cyclone V GX 5CGXFC5C6F27C7N 芯片有不同的转换器，按钮，LED 灯，七段显示器，HDMI 接口，SD 卡读卡器以及其他链接等。Cyclone V GX 5CGXFC5C6F27C7N 是 Starter 工具箱的重要组件。
- 测试台包含在测设备（DUT）和用于仿真的模拟输入。这可以用于测试功能行为和计时行为。我们还可以采用测试台来分配 Starter 工具箱上相关信号的引脚名称，用 force 文件来包含仿真模拟输入。

三、课程设计内容

课程设计的内容为设计与实现单周期 CPU。

单周期 CPU 的主要技术指标：

- 支持表 1 中至少 6 条指令。
- 能运行由自己所设计的指令系统构成的一段测试程序，测试程序应能涵盖所有指令，程序执行功能正确。

表 1 指令格式

#	指令	15~12	11~10	9~8	7~6	5~3	2~0	指令功能
1	or	0	rs	rt	rd	0	0	\$rd = \$rs \$rt (5~3位无用)
2	and	0	rs	rt	rd	0	1	\$rd = \$rs & \$rt 0000000111000001
3	add	0	rs	rt	rd	0	2	\$rd = \$rs + \$rt
4	sub	0	rs	rt	rd	0	3	\$rd = \$rs - \$rt

5	sllv	0	rs	rt	rd	0	4	\$rd = \$rs << \$rt
6	srlv	0	rs	rt	rd	0	5	\$rd = \$rs >> \$rt
7	sraw	0	rs	rt	rd	0	6	\$rd = \$rs >> \$rt 算术右移
8	slt	0	rs	rt	rd	0	7	\$rd = (\$rs < \$rt) ? 1 : 0
9	DISP	1	rs	rt	immediate-u	DISP[imm] = \$rs		
10	lui	2	0	rt	immediate-u	\$rt = imm << 8		
11	ori	3	rs	rt	immediate-u	\$rt = \$rs imm		
12	andi	4	rs	rt	immediate-u	\$rt = \$rs & imm		
13	addi	5	rs	rt	immediate-s	\$rt = \$rs + imm		
14	lw	6	rs	rt	immediate-s	\$rt = MEM[\$rs + imm]		
15	sw	7	rs	rt	immediate-s	MEM[\$rs+imm] = \$rt		
16	beq	8	rs	rt	offset-s	if (\$rs = \$rt) then PC = PC + 4 + SignExt[offset-s] * 4 else PC = PC + 4		
17	bne	9	rs	rt	offset-s	if (\$rs ≠ \$rt) then PC = PC + 4 + SignExt[offset-s] * 4 else PC = PC + 4		
18	bgt	10	rs	rt	offset-s	if (\$rs > \$rt) then PC = PC + 4 + SignExt[offset-s] * 4 else PC = PC + 4		
19	jump	11	target			PC = PC<31:29> 拼接target<25:0> 拼接“00”		
20	halt	12	0			halt (时钟暂停)		

四、课程设计步骤

1. 分析每条指令的功能，并用 RTL(Register Transfer Language)来表示。
2. 根据指令的功能给出所需的元件，并考虑如何将他们互连。
3. 确定每个元件所需控制信号的取值。
4. 汇总所有指令所涉及到的控制信号，生成指令与控制信号关系表。
5. 根据指令与控制信号关系表得到每个控制信号的逻辑表达式，据此设计控制单元电路。
6. 搭建仿真平台或 FPGA 实验环境，对数据通路和控制单元电路进行仿真验证。
7. 撰写课程设计报告，综合运用文字、图表等，对第 1-6 个阶段进行详细论述，同时，采集、整理实验数据，对实验结果进行分析，获取合理有效的结论。

五、课程设计报告

见“计算机组成原理课程设计报告模板”文件