***2021***



**计算机组成原理 ·实验报告·**

j0242087[1]

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术（ACM） |
| 班 级： | ACM1901 |
| 学 号： | U201914965 |
| 姓 名： | 卫云泽 |
| 电 话： | 18703866469 |
| 邮 件： | [1915981921@qq.com](mailto:1915981921@qq.com) |
| 完成日期： | 2021-12-19 |



目 录

[1 变长指令周期3级时序CPU设计实验 1](#_Toc90719935)

[1.1 设计要求 1](#_Toc90719936)

[1.2 方案设计 1](#_Toc90719937)

[1.3 实验步骤 5](#_Toc90719943)

[1.4 故障与调试 7](#_Toc90719949)

[1.5 测试与分析 8](#_Toc90719951)

[2 支持中断的现代时序CPU设计实验 9](#_Toc90719952)

[2.1 设计要求 9](#_Toc90719953)

[2.2 方案设计 9](#_Toc90719954)

[2.3 实验步骤 13](#_Toc90719961)

[2.4 故障与调试 14](#_Toc90719967)

[2.5 测试与分析 15](#_Toc90719970)

[3 总结与心得 17](#_Toc90719971)

[3.1 实验总结 17](#_Toc90719972)

[3.2 实验心得 17](#_Toc90719973)

[参考文献 18](#_Toc90719974)

# 变长指令周期3级时序CPU设计实验

## 设计要求

基于CPU实验电路框架和logisim平台中现有运算部件构建一个变长指令周期3级时序的单总线RISC-V CPU，实验中主要完成指令译码器、时序发生器、硬布线控制器的设计与实现。最终CPU支持5条典型RISC-V指令，能够运行内存冒泡排序程序。CPU支持的指令如表 1‑1所示。

表 1‑1 变长指令周期3级时序单总线RISC-V CPU指令描述

|  |  |
| --- | --- |
| 指令 | RTL功能描述 |
| lw | R[rd] ← M[R[rs1] + SignExt(imm)] |
| sw | M[R[rs1] + SignExt(imm)] ← R[rs2] |
| beq | if(R[rs1] == R[rs2]) PC ← PC + SignExt(imm) << 1 |
| addi | R[rd] ← R[rs1] + SignExt(imm) |
| slt | If (rs1 < rs2) R[rd] ← 1 else R[rd] ← 0 |

## 方案设计

本节采用从宏观到微观的设计思路，先介绍CPU数据通路基本框架，再介绍实验中需要设计的模块——硬布线控制器，最后分解介绍硬布线控制器中的关键部件：指令译码器、时序发生器状态机、时序发生器输出函数和硬布线控制器组合逻辑单元。

### 单总线CPU数据通路

单总线CPU数据通路框架已由实验材料给出，如图 1‑1所示。其中内总线与指令计数器PC、地址寄存器AR、数据寄存器DR、指令寄存器IR、运算逻辑单元ALU、寄存器堆等部件直接相连，内存RAM经由外总线与AR、DR相连，三级时序硬布线控制器与IR相连，同时接收程序状态字产生的状态（反馈）信号，产生每个节拍的控制信号，由控制总线按比特位分为若干控制信号（图中未展示控制总线分线逻辑），控制上述部件与总线连线的三态门或部件自身的状态。数据通路中所示的硬布线控制器需自行设计，包括指令译码器、时序发生器和硬布线控制器组合逻辑单元等。

图示, 示意图

描述已自动生成

图 1‑1 三级时序单总线RISC-V CPU数据通路

### 硬布线控制器

硬布线控制器输入为指令字和反馈信号（实验中为equal信号）和时钟信号，输出为每一个时钟节拍对应的控制总线信号和状态节拍信号（供调试）。

硬布线控制器由指令译码器、时序发生器和硬布线控制器组合逻辑单元三部分构成，其中时序发生器包含状态机与输出函数。各部分组件均需要自行设计实现。

硬布线控制器的数据通路设计如下：

1. 输入的指令字进入指令译码器，生成指令译码信号；

2. 时序发生器状态机根据指令译码信号和来自状态寄存器的现态产生次态，存入状态寄存器；

3. 时序发生器输出函数根据状态寄存器的现态产生对应周期信号M和节拍信号T，共同构成状态节拍信号；

3. 硬布线控制器组合逻辑根据指令译码信号、状态节拍信号和反馈信号（equal）信号产生控制总线信号。

硬布线控制器数据通路结构如图 1‑2 硬布线控制器数据通路所示。

图示

描述已自动生成

图 1‑2 硬布线控制器数据通路

### 指令译码器

实验需设计实现的指令译码器的输入为指令字，输出lw,sw,beq,addi,slt,OtherInstr等六条指令的译码。即若输入的指令字对应lw,sw,beq,addi,slt中其一指令，则将该指令的输出置一，其余输出置零；否则将OtherInstr置一，其余置零。

指令译码设计方案如下：由 RISC-V指令码表可知，RISC-V指令功能由func3字段（指令12-14比特）和opcode字段（指令2-6比特）共同指示，如表 1‑1所示。从指令字中提取func3与opcode对应字段，组合成为OP\_Funct3，再将OP\_Funct3和对应指令的相应字段的组合进行比较，若相等则输出该指令的译码。若OP\_Funct3与lw,sw,beq,addi,slt对应字段组合都不相等，则译码为OtherInstr。

表 1‑2 RISC-V指令操作码与指令对应关系（部分）

|  |  |  |
| --- | --- | --- |
| opcode | func3 | 指令 |
| 0x00 | 0x02 | lw |
| 0x08 | 0x02 | sw |
| 0x18 | 0x00 | beq |
| 0x04 | 0x00 | addi |
| 0x0c | 0x02 | slt |

### 变长指令周期时序发生器

变长指令周期时序发生器根据给定指令生成其执行周期M及对应时钟节拍T。时序发生器实现的状态转移逻辑图如图 1‑3所示。实验中实现的五条指令均包含取值周期Mif和执行周期Mex，Mif、Mex和Mcal分别包含四、二、三个时钟节拍。lw、sw和beq指令包含计算周期，addi和slt则没有计算周期。

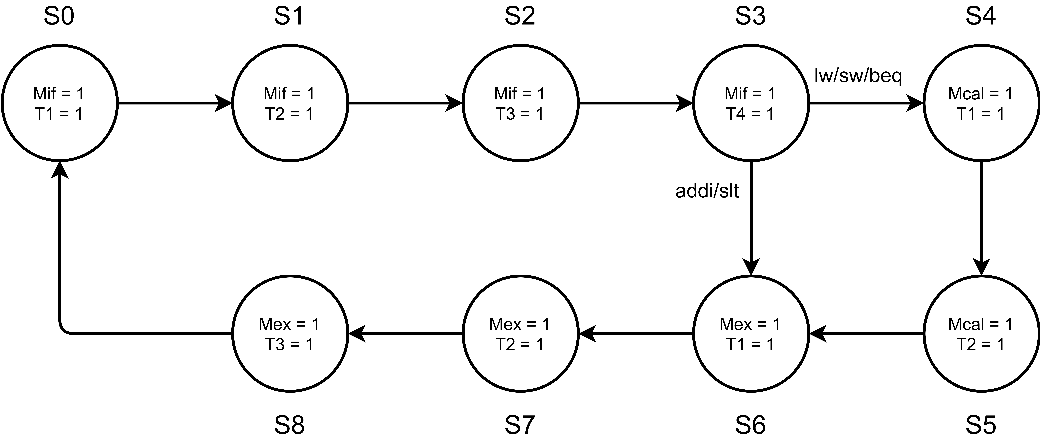


图 1‑3 变长指令周期时序发生器状态转移逻辑图

根据图 1‑3填写状态转换表，根据状态转换表生成的输出逻辑最小项之和表达式生成电路，即可实现**时序发生器状态机**（FSM），即实现不同指令下的状态转移逻辑。**时序发生器输出函数**以状态机生成的现态为输入，将每个状态“译码”为对应的机器周期M和时钟节拍T，如状态S0对应Mif=1、T1=1等。输出函数同样可以通过填写真值表的方式获得输出逻辑表达式，生成输出函数电路。

### 硬布线控制器组合逻辑单元

硬布线控制器组合逻辑单元的输入由四部分构成：指令译码信号、时序发生器产生的机器周期、节拍电位信号和反馈信号（equal），输出组合逻辑函数可表示为：

故只需填写每条指令的每个周期及反馈信号对应的输出控制信号真值表，生成相应输出信号的最小项之和的逻辑函数表达式，即可生成硬布线控制器组合逻辑单元电路。

## 实验步骤

本节采用从微观到宏观的实验流程，先分别实现指令译码器、时序发生器状态机、时序发生器输出函数和硬布线控制器组合逻辑单元，再组合实现硬布线控制器，最后载入内存冒泡排序程序进行测试。

### 指令译码器

（1）指令分线：指令分线要求将指令分出操作码OP、扩展操作码FUNCT3和FUNCT7，再将实验中实际使用的OP和FUNCT3字段组合形成OP\_Funct3，如图 1‑4。

图示

描述已自动生成

图 1‑4 指令分线与操作码重组

（2）生成译码信号：译码信号的生成只需要将OP\_Funct3字段和输出指令的相应字段比较即可，若相同则生成高电平，否则生成低电平。若lw、sw、beq、addi和slt信号均为低电平，则OtherInstr信号为高电平。查阅表 1‑2可获得操作码组合和不同指令的对应关系，以lw指令为例，其opcode字段为0x00，FUNCT3字段为0x02，其输出逻辑如图 1‑5所示。

图示

描述已自动生成

图 1‑5 lw指令译码信号生成逻辑

### 变长指令周期时序发生器状态机

（1）填写状态转换真值表：根据图 1‑3 变长指令周期时序发生器状态转移逻辑图填写状态转移真值表，如图 1‑6所示，其中状态3的次态为4（lw/sw/beq）或6（slt/addi/OtheInstr），其余状态的次态均为现态+1模9的值。

图表

描述已自动生成

图 1‑6 时序发生器状态转换真值表

（2）根据真值表生成的逻辑表达式生成电路：将真值表生成的逻辑表达式复制到Logisim电路分析功能对应输出的表达式中生成电路，即可实现时序发生器状态机。

### 变长指令周期时序发生器输出函数

（1）填写输出函数真值表：根据图 1‑3 变长指令周期时序发生器状态转移逻辑图填写相应状态的输出函数真值表，如图 1‑7所示。

游戏界面截图

描述已自动生成

图 1‑7 时序发生器输出函数真值表

（2）根据真值表生成的逻辑表达式生成电路：将真值表生成的逻辑表达式复制到Logisim电路分析功能对应输出的表达式中生成电路，即可实现时序发生器输出函数逻辑。

### 硬布线控制器组合逻辑单元

（1）根据指令操作流程及控制信号填写控制信号真值表：根据课本[3]中不同指令对应周期及节拍的输出控制信号填写真值表，如图 1‑8所示。

日历, 白板

描述已自动生成

图 1‑8 输出控制信号真值表

（2）根据真值表生成的逻辑表达式生成电路：将真值表生成的逻辑表达式复制到Logisim电路分析功能对应输出的表达式中生成电路，即可实现硬布线控制器组合逻辑单元。

### 硬布线控制器

将指令译码器、时序发生器状态机、时序发生器输出函数和硬布线控制器组合逻辑单元顺次连接形成数据通路。硬布线控制器数据通路实现见图 1‑2。

## 故障与调试

### 硬布线控制器时钟节拍问题

**故障现象：**硬布线控制器设计实验中，出现时钟节拍错位现象，如图 1‑9所示。比对cBus输出与预期输出，发现cBus可以对输入信号产生正确输出，数据通路连接正确，输出错误为时钟节拍错位。

图形用户界面, 文本

描述已自动生成

图 1‑9 硬布线控制器实验时钟节拍错位错误输出

**原因分析：**硬布线控制器产生的是控制总线信号。由于指令执行过程涉及的时钟信号均为上跳沿有效，故控制信号的变化应产生在时钟下跳沿。出现此错误的原因是没有将硬布线控制器的状态寄存器设置为下跳沿有效。

**解决方案：**将硬布线控制器状态寄存器设置为下跳沿触发即可。

## 测试与分析

本实验测试使用内存冒泡排序程序进行。内存冒泡排序的汇编代码见Sort-5-riscv.asm，实现字地址0x80开始的8个字单元的降序排序，编译后的机器指令见Sort-5-riscv.hex。将机器指令加载入CPU数据通路RAM中。加载的程序如图 1‑10前40个字所示。

程序加载完成后Ctrl+K运行程序，当指令数停留在251时表示程序运行完毕。程序运行后结果如图 1‑10字地址0x80到0x88处内存信息所示，输出结果为-1~6的降序排序结果。

图片包含 表格

描述已自动生成

图 1‑10 内存中的冒泡排序程序与运行结果

# 支持中断的现代时序CPU设计实验

## 设计要求

基于CPU实验电路框架和logisim平台中现有运算部件构建一个支持中断的现代时序的单总线RISC-V CPU，实验中主要完成指令译码器、支持中断的微程序入口查找逻辑、支持中断的微程序条件判别逻辑、支持中断的微程序控制器逻辑，此外还需要实现基于现代时序硬布线的CPU。最终CPU支持6条典型RISC-V指令（相比三级时序实验新增中断返回eret指令），能够运行内存冒泡排序，响应两个外部按键中断。CPU支持的指令如表 2‑1支持中断的现代时序单总线RISC-V CPU指令描述所示。

表 ‑支持中断的现代时序单总线RISC-V CPU指令描述

|  |  |
| --- | --- |
| 指令 | 指令功能描述 |
| lw | R[rd] ← M[R[rs1] + SignExt(imm)] |
| sw | M[R[rs1] + SignExt(imm)] ← R[rs2] |
| beq | if(R[rs1] == R[rs2]) PC ← PC + SignExt(imm) << 1 |
| addi | R[rd] ← R[rs1] + SignExt(imm) |
| slt | If (rs1 < rs2) R[rd] ← 1 else R[rd] ← 0 |
| eret | 恢复断点、开中断 |

## 方案设计

本节采用从宏观到微观的设计思路，先介绍支持中断的CPU数据通路基本框架，再介绍实验中需要设计的模块——微程序控制器和硬布线控制器，最后分解介绍微程序控制器中的关键部件：指令译码器、微程序入口查找逻辑、条件判别逻辑，以及硬布线控制器中的硬布线状态机的实现。

### 支持中断的单总线CPU数据通路

支持中断的现代时序CPU数据通路与图 1‑1所示的三级时序CPU数据通路较为相似，其区别在于控制信号改为由微程序控制器产生，增加了中断相关逻辑，这里主要介绍中断逻辑的设计。中断逻辑的输入来自微程序控制器产生的中断控制信号，其中开/关中断信号由中断使能寄存器IE保存，在开中断且有中断信号的情况下由中断控制器生成中断请求，中断请求作为反馈信号输入微程序控制器；中断控制器根据中断信号生成中断入口地址选择信号，由多路选择器选择相应中断程序入口地址输入内总线；中断逻辑的另一个重要部件是异常指令地址寄存器mEPC，负责保存中断执行之前的PC值，即本应当正常执行的下一条指令地址，mEPC与内总线相连，由mEPCin和mEPCout信号控制输入输出三态门的连接状态。支持中断的现代时序CPU数据通路如图 2‑1所示。

图示, 示意图

描述已自动生成

图 2‑1 支持中断的现代时序CPU数据通路

### 支持中断的微程序控制器

支持中断的微程序控制器输入信号包括指令字、反馈信号（equal和IntR）和时钟信号，输出信号为控制总线信号、中断控制信号和用于调试的微指令地址信号mAddr。与1.2.2节介绍的硬布线控制器实现相似，输入的指令字首先由指令译码器产生译码信号，输入入口查找逻辑获得相应指令的微程序入口地址，作为入口地址输入多路选择器。条件判别测试逻辑根据当前微指令的顺序控制字段和反馈信号生成多路选择器的选择信号，决定下一条微指令地址是顺序地址（mAddr+1）、入口地址、beq分支、中断入口还是取指微程序入口地址，微程序地址寄存器寄存多路选择器的输出地址，输出至控制存储器的地址输入端口，获得该地址对应的微指令。支持中断的微程序控制器如图 2‑2所示。

图示, 示意图

描述已自动生成

图 2‑2 支持中断的微程序控制器

### 指令译码器

支持中断的指令译码器与1.2.3节设计的指令译码器完全一致，其OtherInstr作为eret信号的译码输出。

### 支持中断的微程序入口查找逻辑

微程序入口查找逻辑基于图 2‑3 现代时序指令执行状态转换图。实验中微指令地址和状态号相同，从0开始顺次编号，故取指微程序指令入口地址为0，lw指令（计算周期开始）入口地址为4，等等。

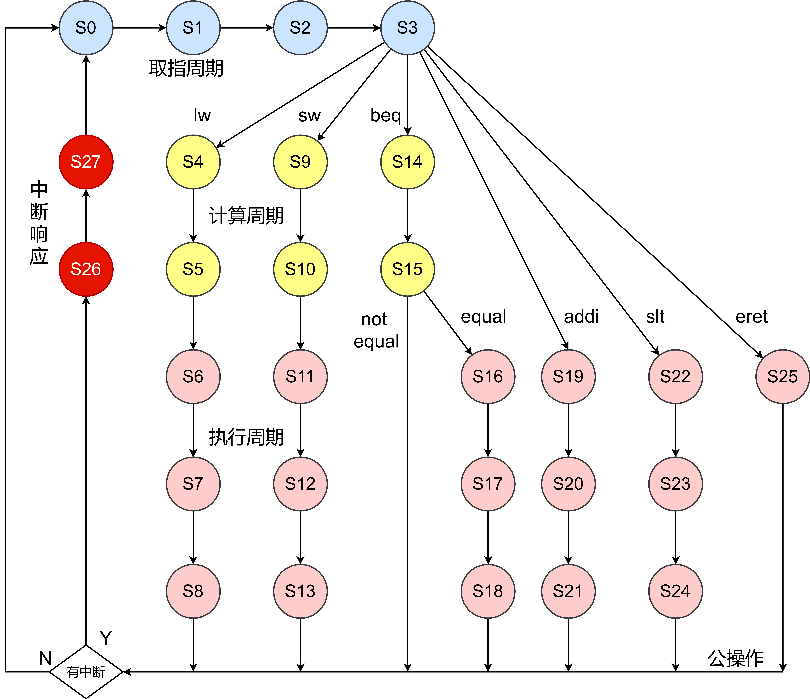


图 2‑3 现代时序指令执行状态转换图

### 支持中断的微程序条件判别测试逻辑

微程序条件判别测试逻辑的输入信号是微指令字中的判别测试逻辑和条件反馈信号，输出信号是后续地址的多路选择信号，其中多路选择信号对应后续地址如表 2‑2所示。

表 2‑2 多路选择信号、对应后续地址及判别测试逻辑对应关系

|  |  |  |
| --- | --- | --- |
| 多路选择信号 | 对应后续地址 | 判别测试逻辑（只填写为1的信号） |
| 0 | 顺序地址 | \ |
| 1 | 下一条指令入口地址 | P0 |
| 2 | beq分支地址 | P1 && equal |
| 3 | 中断入口地址 | P2 && IntR |
| 4 | 取指微程序入口地址 | (P1 && not equal || P2) && not IntR |

根据计数器法微指令顺序控制字段的定义，P0为1表示指令入口地址有效，故多路选择信号为1；P1为1表示当前微指令需要判断equal信号，若equal信号同时为1，则跳转至beq分支；P2为1指示当前为微程序的最后一条微指令，判断IntR是否为1：若IntR为1则跳转至中断入口地址，否则跳转至取指微程序入口地址。

### 支持中断的硬布线控制器

支持中断的硬布线控制器实验主要实现硬布线状态机部分，硬布线控制器组合逻辑单元功能由微程序方法模拟代替。硬布线状态机参照图 2‑3进行设计，支持中断的硬布线控制器数据通路如所示。

图示, 示意图

描述已自动生成

图 2‑4 支持中断硬布线控制器数据通路

## 实验步骤

本节采用从微观到宏观的实验流程，先分别实现指令译码器、支持中断的微程序入口查找逻辑、条件判别逻辑，再将上述部件组合连接形成支持中断的微程序控制器。此外，实验还实现了支持中断的硬布线控制器。最后实现支持中断的单总线CPU（主要是中断控制部分），并运行含有中断程序的内存冒泡排序程序进行测试。

### 指令译码器

支持中断的指令译码器与三级时序CPU指令译码器实现完全一致，见1.3.1节。

### 支持中断的微程序入口查找逻辑

（1）根据图 2‑3 现代时序指令执行状态转换图填写由机器指令译码信号到微程序入口地址转换的真值表，如图 2‑5所示。

图片包含 表格

描述已自动生成

图 2‑5 微程序入口地址查找真值表

（2）根据真值表生成的逻辑表达式生成电路，即可实现微程序入口查找逻辑。

### 支持中断的微程序条件判别测试逻辑

（1）根据表 2‑2填写条件判别测试逻辑真值表，如图 2‑6所示。

（2）根据真值表生成的逻辑表达式生成电路，即可实现条件判别测试逻辑。

表格, 日历

描述已自动生成

图 2‑6 条件判别测试逻辑真值表

### 支持中断的微程序控制器

（1）连接数据通路：根据2.2.2节的微程序控制器设计思路顺次连接指令译码器、地址转移逻辑、判别测试逻辑等，如图 2‑2所示。

（2）确定入口地址：查阅微程序地址表可知，beq分支、中断入口和取指微程序入口地址分别为0x10、0x1a和0x00，计数器法的顺序地址为微地址寄存器现态+1，按照此规则连接多路选择器的多路输入即可实现支持中断的微程序控制器。

### 硬布线控制器

基于图 2‑3图 2‑3 现代时序指令执行状态转换图填写硬布线控制器状态机真值表，生成硬布线控制器状态机组合逻辑电路，并完善硬布线控制器数据通路，如图 2‑4 支持中断硬布线控制器数据通路所示。

## 故障与调试

### 判断逻辑有误

**故障现象：**P1=P2=equal=1时，输出错为6；P1=IntR=1时，输出错为3，如图 2‑7。

电脑萤幕画面

描述已自动生成

图 2‑7 判断逻辑错误

**原因分析：**按照图 2‑6真值表逻辑，P1、P2和equal同时为1时会同时满足4/8两行条件，故组合结果为6，原因是P2和~IntR同时为真时也要判断是否满足beq分支的条件。至于P1=IntR=1时输出为什么应该是4，若从beq指令本身看，beq指令对应的第15条微指令判别控制字段P1和P2同时为1，并不会出现P1为1但P2不为1的情况，故该情况需要增加更为严格的逻辑判断。

**解决方案：**将图 2‑6第8行拆为图 2‑9中8/9两行，可解决第一个问题；将图 2‑6第6行拆为图 2‑9中的6/10两行，可解决第二个问题。

### 支持中断的微程序控制器设计实验中断响应输出有误

**故障现象：**equal信号和IntR信号均为1时，beq指令无法从0x0f跳转至0x10。

电脑萤幕画面

描述已自动生成

图 2‑8 支持中断的微程序控制器设计实验错误输出

**原因分析：**问题仍出现于判别测试逻辑中：当P1=P2=equal=IntR=1时，应当跳转至beq分支，但根据图 2‑6中4/7 两行，输出选择信号为3，即中断入口。

**解决方案：**将图2-6第7行拆为图2-9中7/11两行，增加对P1/equal的判断即可。

表格, 日历

描述已自动生成

图 2‑9 纠错的条件判别测试逻辑真值表

## 测试与分析

在RAM中加载Sort-5-int-riscv.hex，运行程序，按下2次中断信号1和3次中断信号2，程序运行内存结果如图2-10所示。字地址0x80之后的8个字实现了 -1~6的降序排序；字地址0x90之后的8个字均为2，即对2次中断信号1的响应结果；字地址0xa0之后的8个字均为-3，即对3次中断信号2的响应结果。

图片包含 文本

描述已自动生成

图 2‑10 含有中断响应的内存冒泡排序程序与运行结果

# 总结与心得

## 实验总结

本次实验主要完成了如下几点工作：

1. 完成方案总结：完成了变长指令三级时序CPU、现代时序CPU、支持中断的现代时序CPU实验；设计了指令译码器、变长指令周期时序发生器、硬布线控制器组合逻辑单元、微程序入口查找逻辑、微程序条件判别测试逻辑等部件；实现了硬布线控制器、微程序控制器等模块。
2. 功能总结：变长指令周期3级时序CPU和现代时序CPU实现了对5条典型RISC-V指令的支持，运行了内存冒泡排序程序；支持中断的现代时序CPU额外实现了对eret指令的支持，运行了含有中断程序的内存冒泡排序程序，实现了对程序中两种中断的正确响应。

## 实验心得

1. 本次实验收获颇丰：实验加深了我对CPU整体运行过程的理解与体悟。CPU设计实验主要关注了控制总线信号产生部分，即硬布线控制器或微程序控制器的实现。控制器是有机结合CPU中各组件的关键部件，从指令译码，到最终产生每个时钟节拍对应的控制总线信号，不同的实现方案有其各自的优劣，如硬布线控制器硬件成本较高，难以修改，但执行速度较快；微程序控制器修改灵活，代价是牺牲了一定的微指令执行速度。在不同控制器的设计与实现中，更加真切地体会到不同实现方式的特点。此外，实验的开展也加深了我对各种微命令的理解，实验后我可以为指令的不同执行周期设计合适的微命令组合。总而言之，本次实验是从理论到实践的一次很好的落地。
2. 实验体会：CPU实验的完成再度让我感受到处理器内部控制逻辑设计的精妙。实验中的每一次逻辑bug、考虑不周到而出现的设计漏洞，或许也正是CPU设计的先辈们曾遇到过的问题。与任何科研相同，CPU的设计与实现是艰辛的，但这也正是我国目前的科技缺口。我国半导体技术的发展路漫漫而修远，还有待我辈学成之后投身其中。

# 参考文献

1. DAVID A.PATTERSON(美).计算机组成与设计硬件/软件接口(原书第5版).北京:机械工业出版社.
2. David Money Harris(美).数字设计和计算机体系结构（第二版）. 机械工业出版社
3. 谭志虎,秦磊华,吴非，肖亮.计算机组成原理.北京:人民邮电出版社，2021年.
4. 谭志虎,秦磊华,胡迪青.计算机组成原理实践教程.北京:清华大学出版社，2018年.

|  |
| --- |
| 一、原创性声明 |
| 本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。  特此声明！  **作者签字:** |
| 二、对课程实验的学术评语（教师填写） |
|  |
| 三、对课程实验的评分（教师填写） |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 评分项目  （分值） | 课程目标1  工具应用  （10分） | 课程目标2  设计实现  （70分） | 课程目标3  验收与报告  （20分） | 最终评定  （100分） | | 得分 |  |  |  |  | |
| **指导教师签字:** |