***2021***



**计算机组成原理 ·实验报告·**

j0242087[1]

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 班 级： |  |
| 学 号： |  |
| 姓 名： | [作者] |
| 电 话： |  |
| 邮 件： |  |
| 完成日期： | 2021-12-18 |



目 录

[1 CPU设计实验（三级时序） 1](#_Toc90754913)

[1.1 设计要求 1](#_Toc90754914)

[1.2 方案设计 1](#_Toc90754915)

[1.3 实验步骤 4](#_Toc90754921)

[1.4 故障与调试 4](#_Toc90754922)

[1.5 测试与分析 4](#_Toc90754923)

[2 CPU设计实验（现代时序中断实现） 5](#_Toc90754924)

[2.1 设计要求 5](#_Toc90754925)

[2.2 方案设计 5](#_Toc90754926)

[2.3 实验步骤 9](#_Toc90754934)

[2.4 故障与调试 9](#_Toc90754935)

[2.5 测试与分析 10](#_Toc90754936)

[3 总结与心得 11](#_Toc90754937)

[3.1 实验总结 11](#_Toc90754938)

[3.2 实验心得 11](#_Toc90754939)

[参考文献 13](#_Toc90754940)

# CPU设计实验（三级时序）

## 设计要求

利用logisim平台构建一个变长指令周期三级时序单总线RISC-V CPU，可支持LW，SW，BEQ，ADDI，SLT指令，并支持自动读取存放于ROM中的指令以运行包含上述指令的简单程序。

## 方案设计

### RISC-V指令译码器设计

查找RISC-V指令码表可知，各个指令的op字段和funct3字段如下表所示。只需将输入指令与这些常量比较即可确认输入的指令。

表 1.1 指令与机器码的对应关系

|  |  |  |
| --- | --- | --- |
| 指令 | OP | Funct3 |
| ADDI | 0x4 | 0x0 |
| LW | 0 | 0x2 |
| SW | 0x8 | 0x2 |
| BEQ | 0x18 | 0x0 |
| SLT | 0xc | 0x2 |

得到电路图如图 1.1所示：

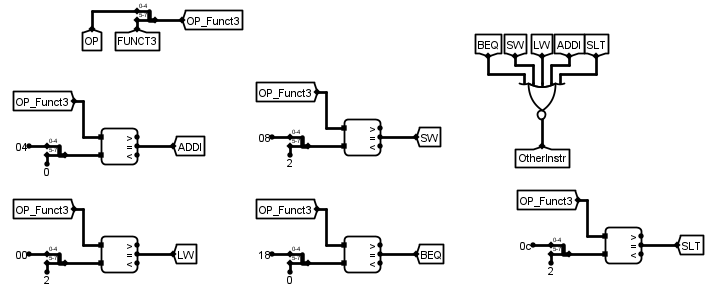


图 1.1 指令译码器电路

### 变长指令周期时序发生器FSM设计

根据状态图正确填写1号excel表格，如图 1.2得到逻辑表达式，用logisim的分析组合逻辑电路功能即可得到正确的电路。

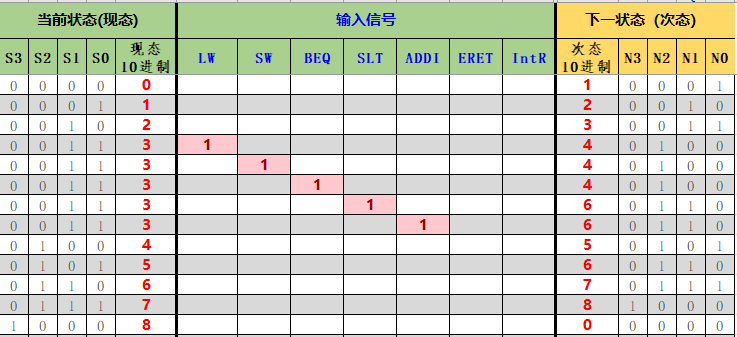


图 1.2 填写1号excel表格的结果

### 边长指令周期时序发生器输出函数设计

依然根据状态图，填写1号excel表中的输出函数真值表部分，如图 1.3，利用logisim分析组合逻辑电路功能即可得到正确电路。

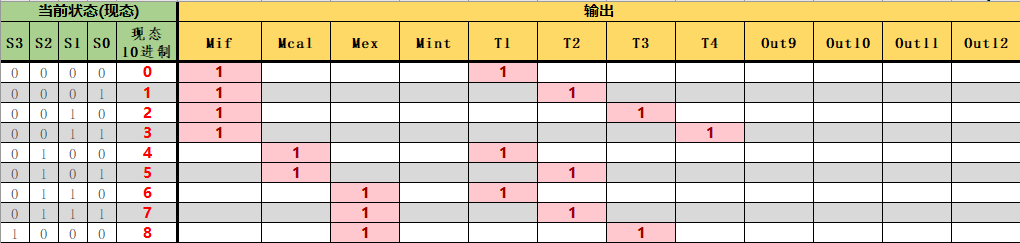


图 1.3 填写1号excel表格的结果

### 硬布线控制器组合逻辑单元设计

根据教材第六章（RISC-V）对这5种指令的介绍，正确填写2号excel表，如图 1.4，获得各输出信号表达式，利用logisim分析组合逻辑电路即可得到正确电路。

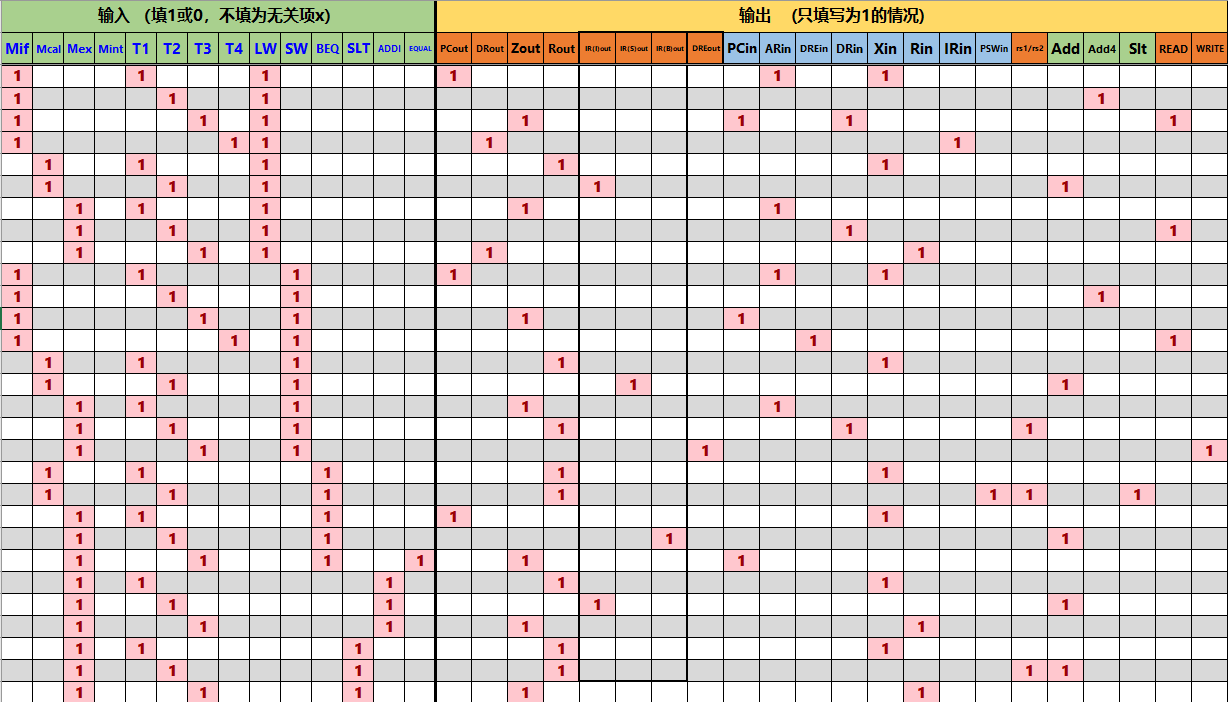


图 1.4 填写2号excel表格的结果

### 变长指令周期硬布线控制器设计

将状态机组件和输出函数组件选定为1.2.2和1.2.3中完成的组件，根据引脚标识连接状态机的LW，SW，BEQ，SLT，ADDL引脚。状态寄存器用于保存现态和接收次态，其输出端用分线器连接至状态机的S0~S3引脚和输出函数的S0~S3引脚，输入端用分线器与状态机的N0~N3输出引脚连接即可。电路结果见图 1.5。

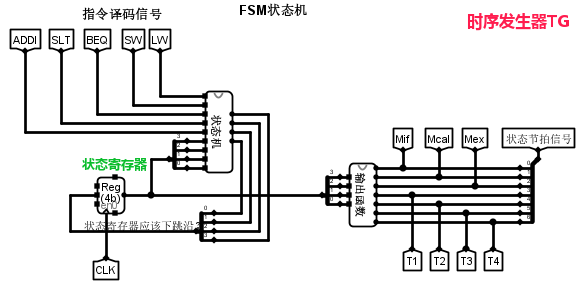


图 1.5 寄存器、状态机和输出函数的连接

## 实验步骤

按照1.2.1~1.2.6依次连接好电路和加载程序即可。

## 故障与调试

## 测试与分析

在连接好的RAM中加载sort-5-riscv.hex文件，自动运行，观察到程序停留在0x81d节拍，指令数为251，如图 1.6，RAM中0x80~0x87中的数按有符号从大到小排序正确，如图 1.7。

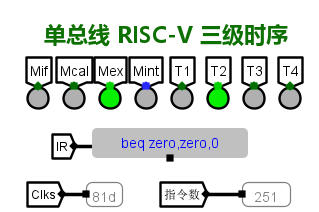


图 1.6 程序停在正确的位置

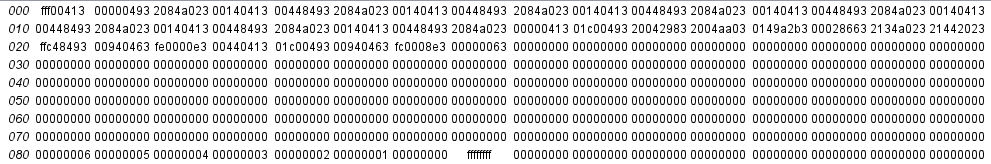


图 1.7 RAM中数字从大到小正确排序

# CPU设计实验（现代时序中断实现）

## 设计要求

利用logisim平台构建一个支持中断的现代时序RISC-V CPU，可支持LW，SW，BEQ，ADDI，SLT指令，支持中断机制，并支持自动读取存放于RAM中的指令以运行包含上述指令的简单程序。

## 方案设计

### RISC-V指令译码器设计

同[1.2.1](#_RISC-V指令译码器设计)

### 支持中断的微程序入口查找逻辑

根据状态图填写5号excel表格，得到组合逻辑表达式，利用logisim的组合逻辑电路分析功能即可得到正确的电路。填写结果见图 2.1

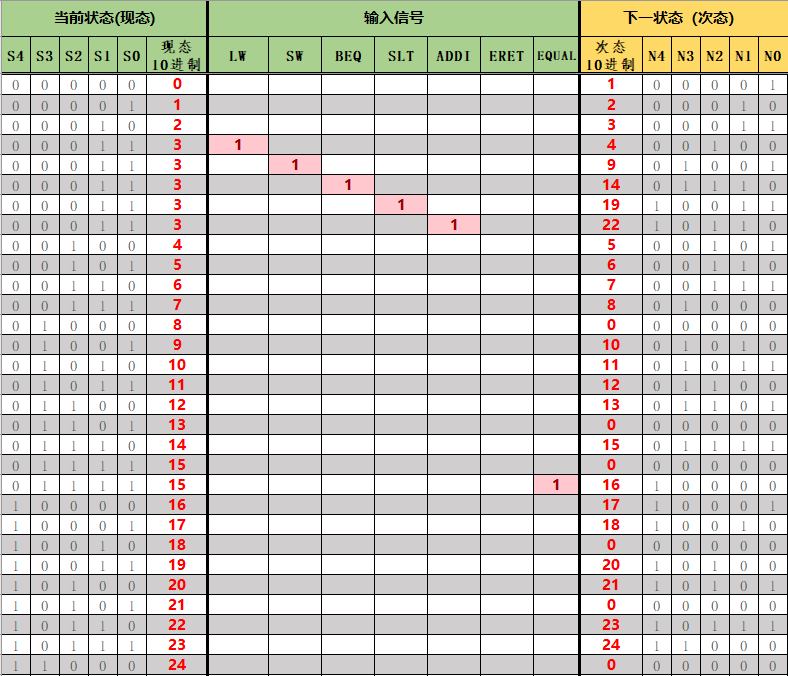


图 2.1 填写5号excel表格的结果

### 支持中断的微程序条件判别式测试逻辑

根据输入引脚P0，P1，P2，equal的含义，填写4号excel表格，如图 2.2得到逻辑表达式，利用logisim的分析组合逻辑电路功能即可得到正确电路。

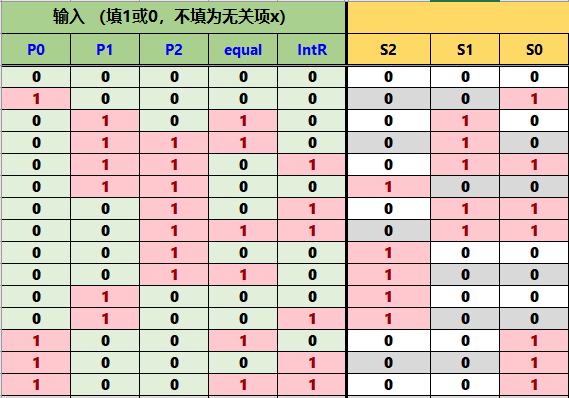


图 2.2 填写4号excel表的结果

### 支持中断的微程序控制器设计

在6号excel表中，根据教材对此部分的讲解，可以填写出现代时序的微程序控制器，如图 2.3。在此基础上，为了实现中断，引入了P2表示表明微程序是一条指令的最后一条微指令。因此需要在=将每条指令的最后一条微指令的P2置为1。另外，BEQ指令的第二条微指令为了实现分支跳转，也需要将P2置为1。ERET，中断指令1，2按照教材讲解填写即可。最终将生成的16进制微指令导入到控制存储器中。

连线部分，P0，P1，P2，equal，IntR起到判别测试的作用，实现程序的分支跳转。将其连接至判别测试组件，输出端通过分线器连接至多路选择器的选择端。入口查找与程序入口线相连。加法器实现指令地址加1，根据顺序地址的含义，加法器的输出端连接顺序地址线。当beq指令判定两操作数相等时，beq指令需跳转至beq的第三条微指令（因为beq默认在第二条微指令处结束），其地址为16，故中断入口为常量0x10。当P2=1且IntR=1时，程序跳转至中断分支，中断分支的第一条指令地址为26，故中断入口地址为常量0x1a。当P2=1且IntR=0时，指令正常结束，跳转至取指指令，其地址为0，则取指微程序入口为常量0x0。这样就完成了微程序控制器，电路图见图 2.4。

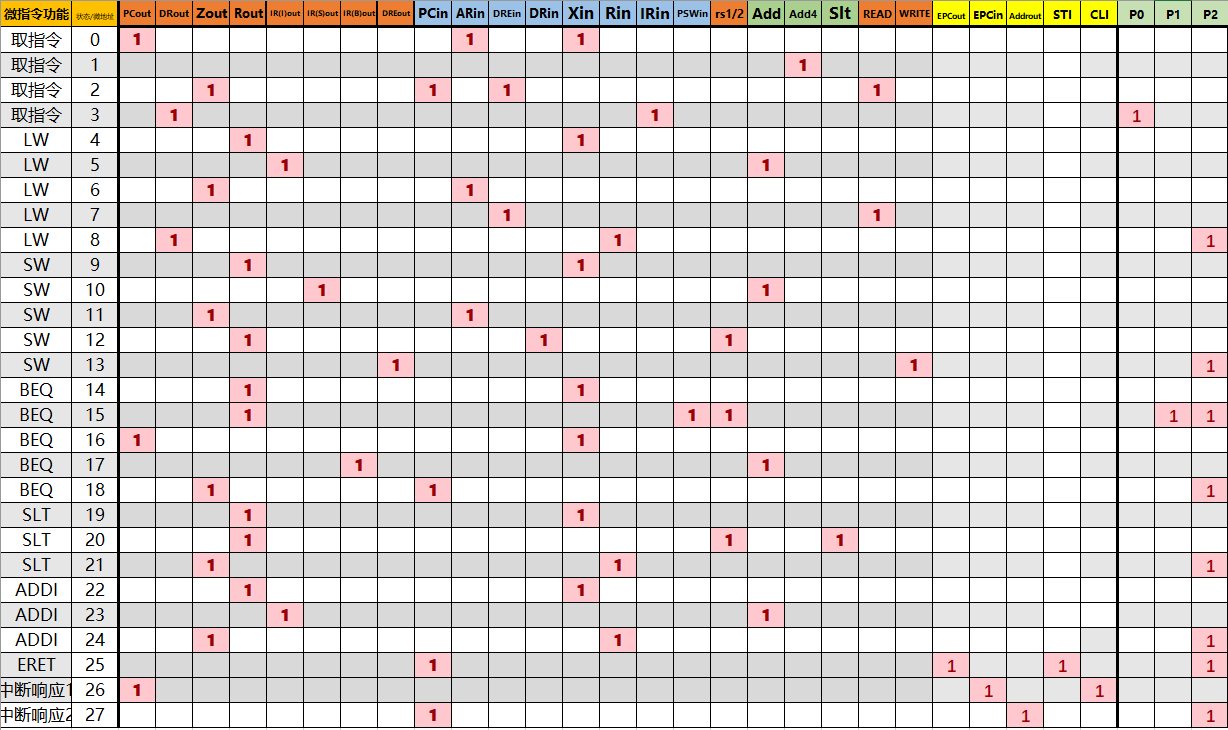


图 2.3 填写6号excel表的结果

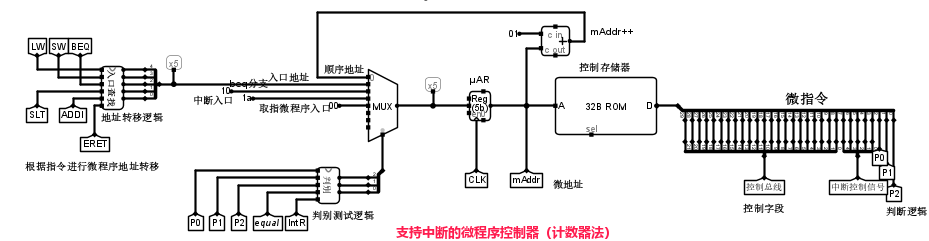


图 2.4 微程序控制器的连线结果

### 支持中断的微程序单总线CPU设计

CPU主要部分已经被完成，只需完成中断控制部分。mEPC用于保存断点，当收到mEPCin信号时，从内总线读取异常指令地址，收到mEPCout时将保存的地址输出到内总线。因此mEPCin连接寄存器的使能端，寄存器输出端通过eEPCout控制的三态门与内总线相连，同时寄存器的输入端与内总线相连。

中断控制器根据中断的类型，使程序运行不同的中断程序。则中断控制器的IntNo.端与一多路选择器的选择端相连，多路选择器的输入端为两中断程序入口地址。利用软件分析可知，这两个中断程序入口地址相对程序入口地址的偏移量分别为0xa4和0xec，这两个常量为多路选择器的输入端。

中断使能寄存器用于开关中断。当其处于开中断状态且有中断信号时，发出中断请求。根据D触发器的真值情况，将状态的非值与中断信号进行与操作，与门输出端即为中断请求。最终连接结果见图 2.5。

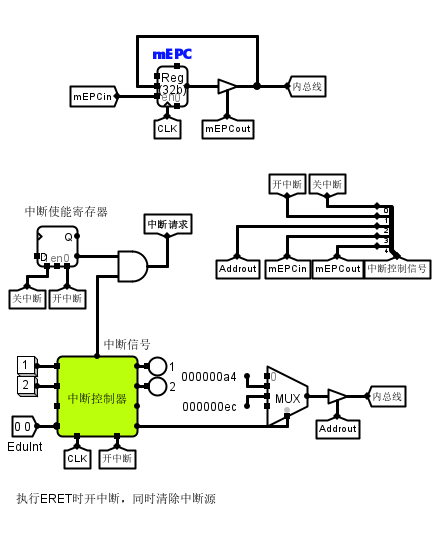


图 2.5 中断逻辑的实现

### 支持中断的现代时序硬布线控制器状态机设计

根据状态图正确填写5号excel表格，如图 2.6，得到逻辑表达式，利用logisim的分析组合逻辑电路即可得到正确的状态图。

### 支持中断的现代时序硬布线控制器设计

2.2.6设计的状态机输出端取决于现态与指令，则将寄存器的输出端通过分线器连接至状态机的现态输入端口，指令引脚与对应指令连接，状态机输出端通过分线器连接至寄存器的输入端即可，连接结果如图 2.7。

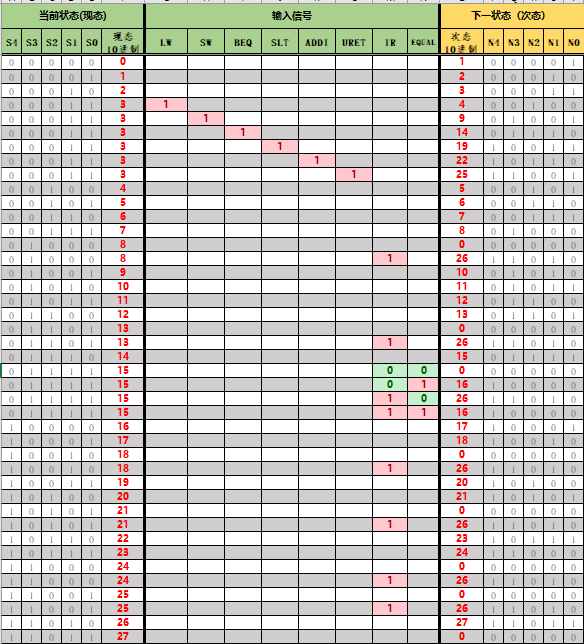


图 2.6 填写5号excal表的结果

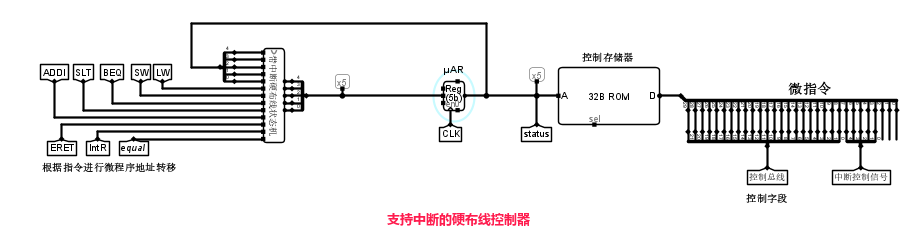


图 2.7 支持中断的硬布线控制器电路

## 实验步骤

根据2.2中描述方法，按顺序正确连接各个电路即可。

## 故障与调试

**故障现象：**在2.2.6中，状态机在现态为15时无法正确实现跳转。具体表现为当现态为15，equal=IR=1时，次态为26而不是16.

**原因分析：**这是由于在填写5号excel表时，此部分当IR或EQUAL为0时未填写相应表项。而在此表中，不填写视为与其无关，对于几个不会同时出现的变量，这样做不会造成问题，而当几个变量会同时出现时，如IR与EQUAL，则会造成逻辑错误。

**解决方案：**现态为15时，将IR与EQUAL的取值分为四种情况分别填写。

## 测试与分析

将Sort-5-int-riscv.hex加载到RAM中并运行，观察到程序进入死循环时节拍数为0x7c8，指令条数为251（图 2.8），内存中从0x80开始，按有符号数从大到小正确排列（图 2.9），表明电路正确且满足要求。

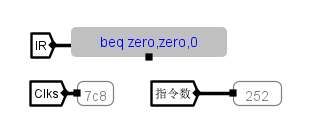


图 2.8 程序结束时的节拍数与指令条数

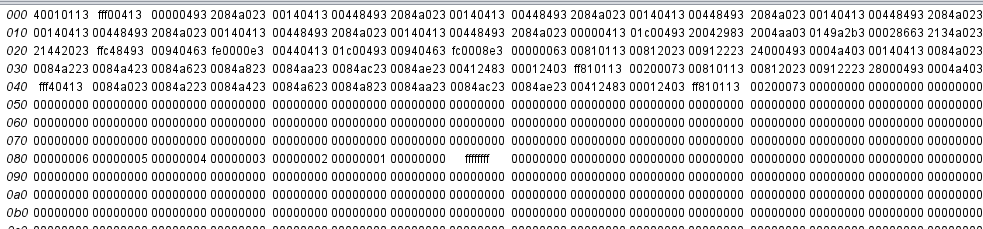


图 2.9 内存中的数按正确顺序排列

# 总结与心得

## 实验总结

1）完善了CPU电路设计：包括现代时序的指令译码器，微程序控制器、微程序入口查找、条件判别逻辑设计、微程序控制器和硬布线控制器；三级时序变长指令周期的时序发生器组合逻辑单元和硬布线控制器等。

2）进行了程序运行测试：主要为educoder每一关的自动测试以及最终的加载程序和运行的测试。

## 实验心得

CPU设计是组原学习和组原实验的重要内容。本次实验聚焦现代时序和三级时序这两种主要时序方式，涉及硬布线和微程序两种控制方式，也有中断功能的设计，是我对现代CPU运行的基本原理有所了解，加深了对教材第六章的理解。另外，excel表的填写与生成逻辑表达式和利用logisim分析组合逻辑电路的功能也让我感受到合理运用工具为学习过程提供的巨大便利。

# 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:** [作者] **嵌入签名图片** |
| 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:** |