**“计算机设计与实践”处理器实验设计报告**

姓名：闫纪良

班级：1503105

学号：1150310513

哈尔滨工业大学计算机学院

2017年7月

目录

[1. 整体设计 4](#_Toc488762875)

[1.2 整体设计框图 4](#_Toc488762876)

[1.2 整体设计思路 4](#_Toc488762877)

[2. 各模块接口框图以及信号定义 5](#_Toc488762878)

[2.1 时钟管理模块 5](#_Toc488762879)

[2.1.1 接口框图 5](#_Toc488762880)

[2.1.2 信号定义 5](#_Toc488762881)

[2.2 取指管理模块 6](#_Toc488762882)

[2.2.1 接口框图 6](#_Toc488762883)

[2.2.2 信号定义 6](#_Toc488762884)

[2.3 运算管理模块 7](#_Toc488762885)

[2.3.1 接口框图 7](#_Toc488762886)

[2.3.2 信号定义 7](#_Toc488762887)

[2.4 存储管理模块 8](#_Toc488762888)

[2.4.1 接口框图 8](#_Toc488762889)

[2.4.2 信号定义 8](#_Toc488762890)

[2.5 回写管理模块 9](#_Toc488762891)

[2.5.1 接口框图 9](#_Toc488762892)

[2.5.2 信号定义 9](#_Toc488762893)

[2.6 访存控制模块 10](#_Toc488762894)

[2.6.1 接口框架 10](#_Toc488762895)

[2.6.2 信号定义 10](#_Toc488762896)

[3 系统测试 11](#_Toc488762897)

[3.1 测试方案 11](#_Toc488762898)

[3.2 测试过程 11](#_Toc488762899)

[3.3 仿真波形 12](#_Toc488762900)

[4 各模块系统测试 14](#_Toc488762901)

[4.1 时钟管理模块 14](#_Toc488762902)

[4.1.1 测试过程 14](#_Toc488762903)

[4.1.2 仿真波形 14](#_Toc488762904)

[4.2 取指管理模块 14](#_Toc488762905)

[4.2.1 测试过程 14](#_Toc488762906)

[4.2.2 仿真波形 14](#_Toc488762907)

[4.3 运算管理模块 15](#_Toc488762908)

[4.3.1 测试过程 15](#_Toc488762909)

[4.3.2 仿真波形 15](#_Toc488762910)

[4.4 存储管理模块 15](#_Toc488762911)

[4.4.1 测试过程 15](#_Toc488762912)

[4.4.2 仿真波形 16](#_Toc488762913)

[4.5 回写管理模块 17](#_Toc488762914)

[4.5.1 测试过程 17](#_Toc488762915)

[4.5.2 仿真波形 17](#_Toc488762916)

[4.6 访存控制模块 17](#_Toc488762917)

[4.2.1 测试过程 17](#_Toc488762918)

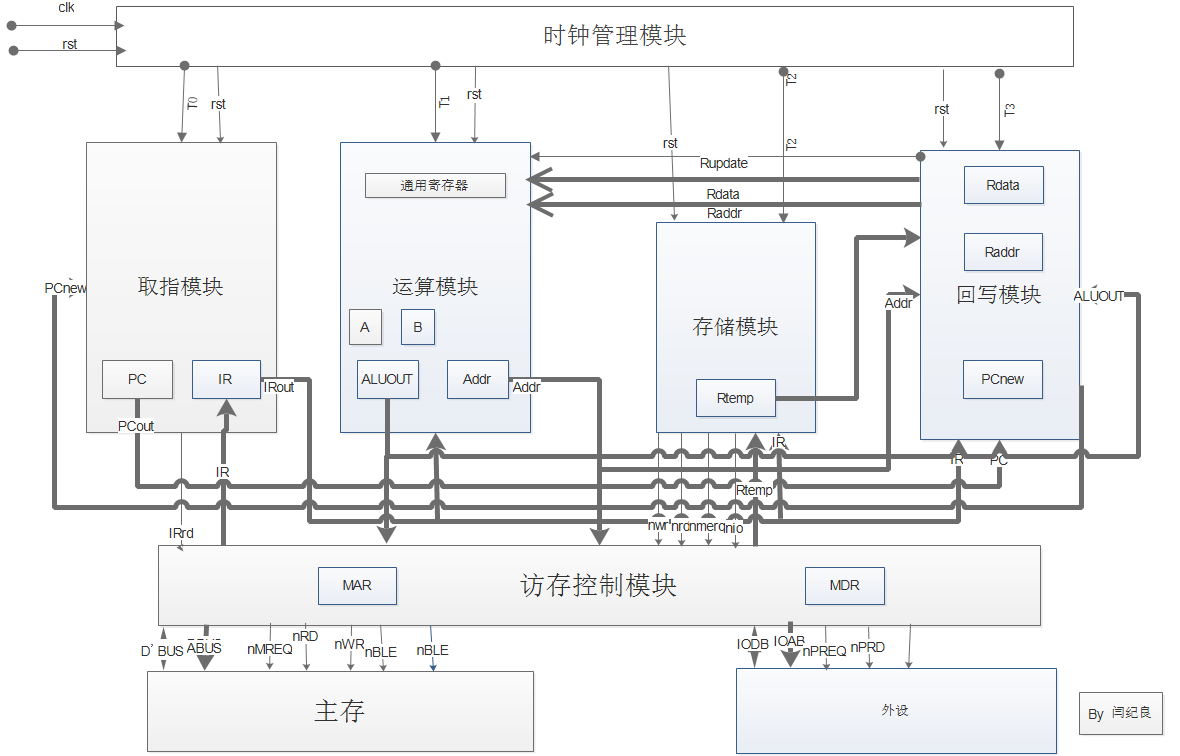
[4.2.2 仿真波形 18](#_Toc488762919)

[5. 处理器功能测试程序 18](#_Toc488762920)

[6 问题及解决办法 19](#_Toc488762921)

# 整体设计

## 1.2 整体设计框图



## 整体设计思路

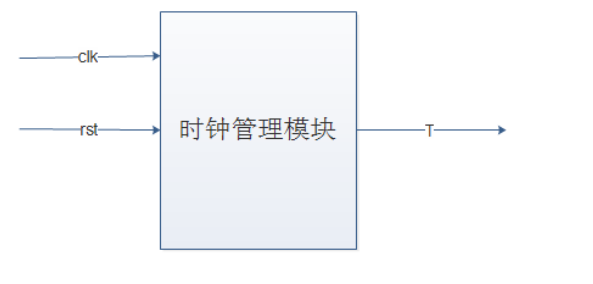
本CPU 设计可满足课程要求中的十条指令的正常运行。主要由六大模块组成，时钟管理模块，取指管理模块，运算管理模块，存储管理模块，回写管理模块，访存控制管理模块。时钟管理模块主要产生四个节拍，控制其他模块的执行。取指管理模块主要执行指令从内存通过访存管理模块的提取，以及将取回的指令送往运算、存储、回写模块以供后续使用。运算管理模块主要执行算术逻辑运算、地址的生成、寄存器的写入操作。存储管理模块主要根据指令操作码产生相应的访存控制信号。回写管理模块主要做了PC新地址的计算以及回写，寄存器的回写。访存控制模块主要和主存以及外设进行数据传输。

# 各模块接口框图以及信号定义

## 2.1 时钟管理模块

产生四个节拍，控制其余模块按相应节拍工作

### 2.1.1 接口框图



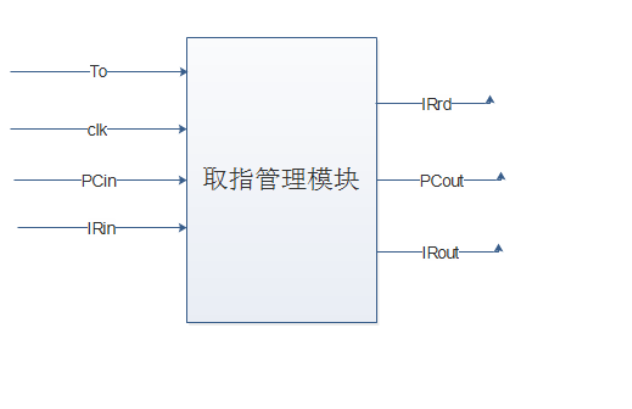
### 2.1.2 信号定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名 | 位数 | 方向 | 来源/去向 | 意义 |
| Clk | 1 | In | 处理器 | 系统时钟 |
| rst | 1 | In | 处理器 | 复位信号 |
| T | 4 | Out | 取指、运算、存储、回写 | 时钟信号 |

## 取指管理模块

取指管理模块主要执行指令从内存通过访存管理模块的提取，以及将取回的指令送往运算、存储、回写模块以供后续使用。

### 接口框图



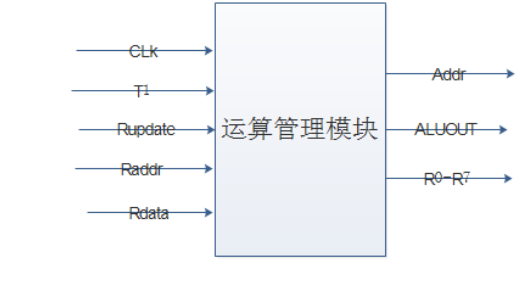
### 2.2.2 信号定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名 | 位数 | 方向 | 来源/去向 | 意义 |
| Clk | 1 | In | 处理器 | 系统时钟 |
| T0 | 1 | In | 时钟模块 | 时钟信号，控制工作 |
| PCin | 16 | In | 回写模块 | 更新PC值 |
| IRin | 16 | In | 访存控制模块 | 指令 |
| IRout | 16 | Out | 运算、存储、回写 | IR指令送往其他模块以供使用 |
| IRrd | 1 | Out | 访存控制模块 | 读取指令信号 |
| PCout | 16 | Out | 访存控制、回写 | 当前指令地址 |

## 运算管理模块

运算管理模块主要执行算术逻辑运算、地址的生成、寄存器的写入操作。

### 接口框图



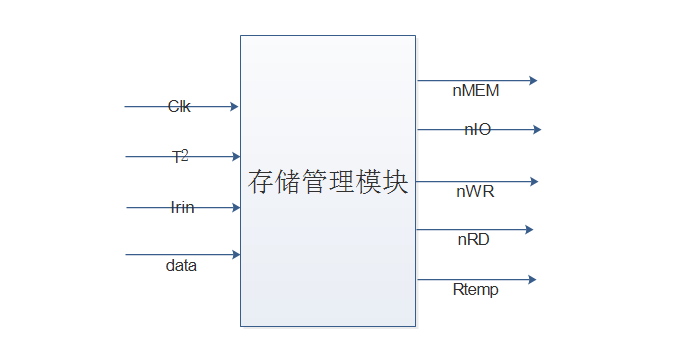
### 2.3.2 信号定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名 | 位数 | 方向 | 来源/去向 | 意义 |
| Clk | 1 | In | 处理器 | 系统时钟 |
| T1 | 1 | In | 时钟管理模块 | 时钟信号，控制工作 |
| Rupdate | 1 | In | 回写管理模块 | 寄存器会写命令 |
| Raddr | 3 | In | 回写管理模块 | 寄存器回写地址 |
| Rdata | 8 | In | 回写管理模块 | 寄存器回写数据 |
| Addr | 16 | Out | 访存控制、回写管理模块 | 有效地址 |
| ALUOUT | 8 | Out | 访存控制、回写管理模块 | 运算结果 |
| R0-R7 | 8 | Out | 处理器数码管 | 通用寄存器 |

## 存储管理模块

存储管理模块主要根据指令操作码产生相应的访存控制信号。

### 2.4.1 接口框图



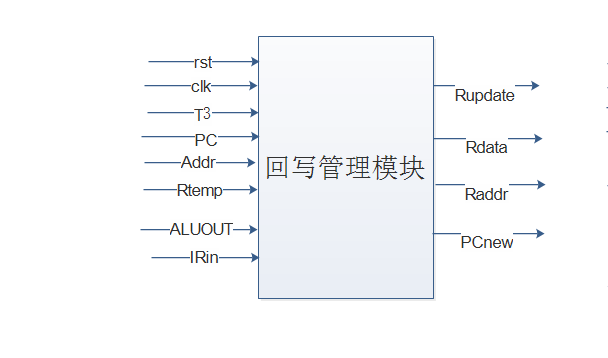
### 2.4.2 信号定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名 | 位数 | 方向 | 来源/去向 | 意义 |
| Clk | 1 | In | 处理器 | 系统时钟 |
| T2 | 1 | In | 时钟管理模块 | 时钟信号，控制工作 |
| Irin | 16 | In | 取指管理模块 | 指令码 |
| Data | 8 | In | 访存控制模块 | 内存、外设接收数据 |
| nMEM | 1 | Out | 访存管理模块 | 片选信号 |
| nIO | 1 | Out | 访存管理模块 | 外设片选信号 |
| nWR | 1 | Out | 访存管理模块 | 写有效 |
| nRD | 1 | Out | 访存管理模块 | 读有效 |
| Rtemp | 8 | Out | 回写管理模块 | 取到的数据 |

## 回写管理模块

回写管理模块主要做了PC新地址的计算及回写，寄存器的回写。

### 接口框图



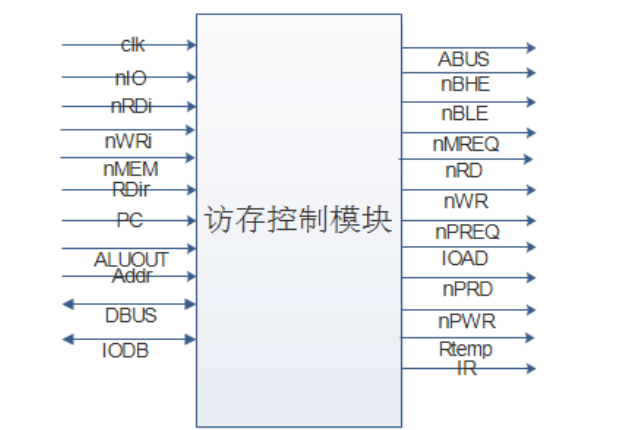
### 2.5.2 信号定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名 | 位数 | 方向 | 来源/去向 | 意义 |
| Clk | 1 | In | 处理器 | 系统时钟 |
| T3 | 1 | In | 时钟管理模块 | 时钟信号，控制工作 |
| Rst | 1 | In | 处理器 | 复位信号 |
| PC | 16 | In | 取指管理模块 | 当前指令地址 |
| Addr | 16 | In | 运算管理模块 | Pc待转移地址 |
| Rtemp | 8 | In | 存储管理模块 | 内存、外设所取数据 |
| ALUOUT | 8 | In | 运算管理模块 | 回写寄存器数据 |
| IRin | 16 | In | 取指管理模块 | 指令码 |
| Rupdate | 1 | Out | 运算管理模块 | 寄存器回写信号 |
| Rdata | 8 | Out | 运算管理模块 | 寄存器回写数据 |
| Raddr | 3 | Out | 运算管理模块 | 回写寄存器号 |
| PCnew | 16 | Out | 取指管理模块 | PC新地址 |

## 访存控制模块

访存控制模块主要和主存以及外设进行数据传输。

### 接口框架



### 信号定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名 | 位数 | 方向 | 来源/去向 | 意义 |
| Clk | 1 | In | 处理器 | 系统时钟 |
| nIO | 1 | In | 存储管理模块 | 访问IO |
| nRDi | 1 | In | 存储管理模块 | 读内存 |
| nWRi | 1 | In | 存储管理模块 | 写内存 |
| nMEM | 1 | In | 存储管理模块 | 访问主存信号 |
| RDIR | 1 | In | 取指管理模块 | 取指令信号 |
| PC | 16 | In | 取指管理模块 | 待取指令地址 |
| ALUOUT | 8 | In | 运算管理模块 | 待写入数据 |
| Addr | 16 | In | 存储管理模块 | 访存地址 |
| DBUS | 16 | Inout | 主存 | 数据总线 |
| ABUS | 16 | Out | 主存 | 地址总线 |
| nBHE | 1 | Out | 主存 | 访问高字节 |
| nBLE | 1 | Out | 主存 | 访问低字节 |
| nMREQ | 1 | Out | 主存 | 访问主存信号 |
| nRD | 1 | Out | 主存 | 读信号 |
| nWR | 1 | Out | 主存 | 写信号 |
| nPREQ | 1 | Out | IO外设 | 访问IO信号 |
| IODB | 8 | Inout | IO外设 | 数据总线 |
| IOAD | 2 | Out | IO外设 | 地址总线 |
| nPRD | 1 | Out | IO外设 | 读信号 |
| nPWR | 1 | Out | IO外设 | 写信号 |
| Rtemp | 8 | Out | 存储管理模块 | 取到的数据 |
| IR | 16 | Out | 取指管理模块 | 取到的指令 |

# 3 系统测试

## 3.1 测试方案

在cpu接口设计中，我除过将教材中给出的各种控制信号以及数据总线、地址总线输出外，还将8个通用寄存器、PC、IR输出，以便通过波形观察指令是否正确执行。指令执行中，T显示时钟节拍，DBUS显示数据总线，ABUS显示地址总线，nmreq显示存储器片选，IOAD、IODB显示外设端口、外设数据总线。依次执行设计要求中的十条指令，若结果均正确，证明系统可以正常执行。

## 3.2 测试过程

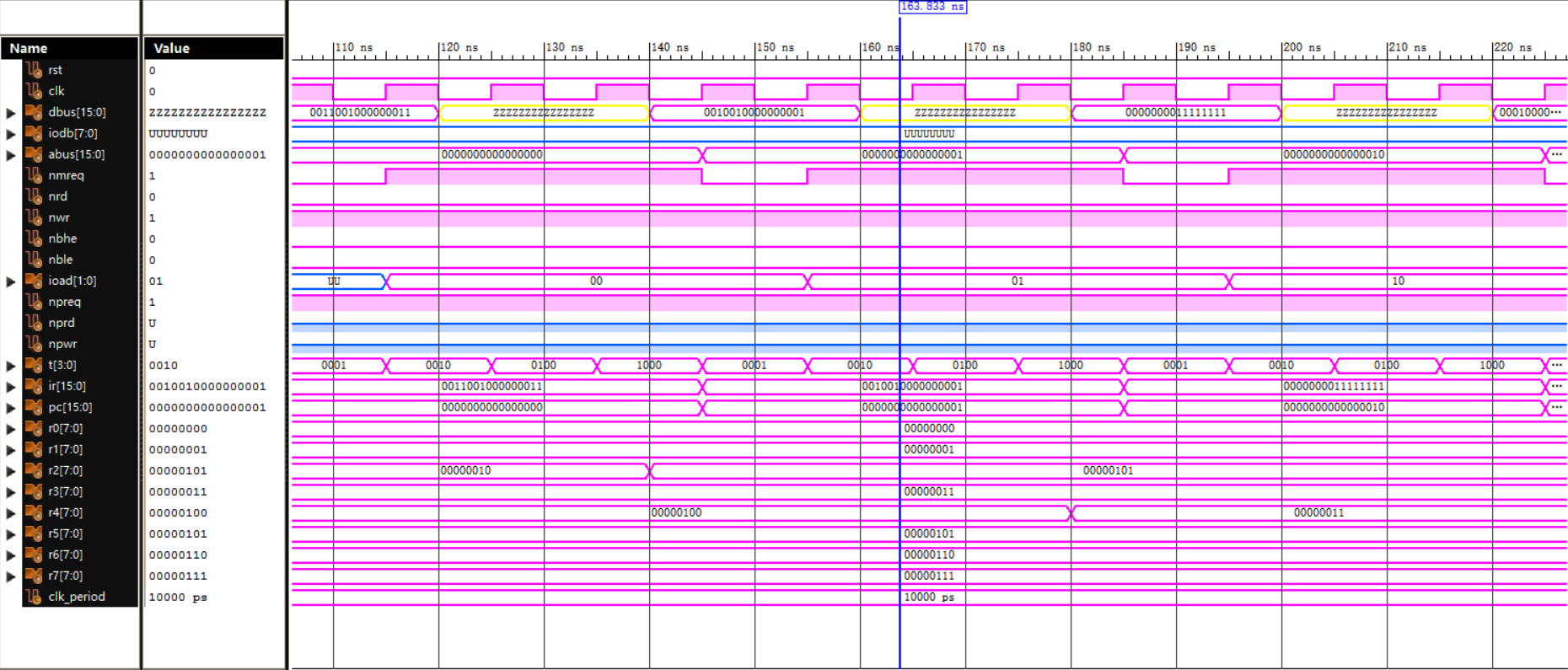
我在程序中就将通用寄存器初始化，故一开始寄存器里存在值，R0-R7依次为0-7；

依次执行十条指令：

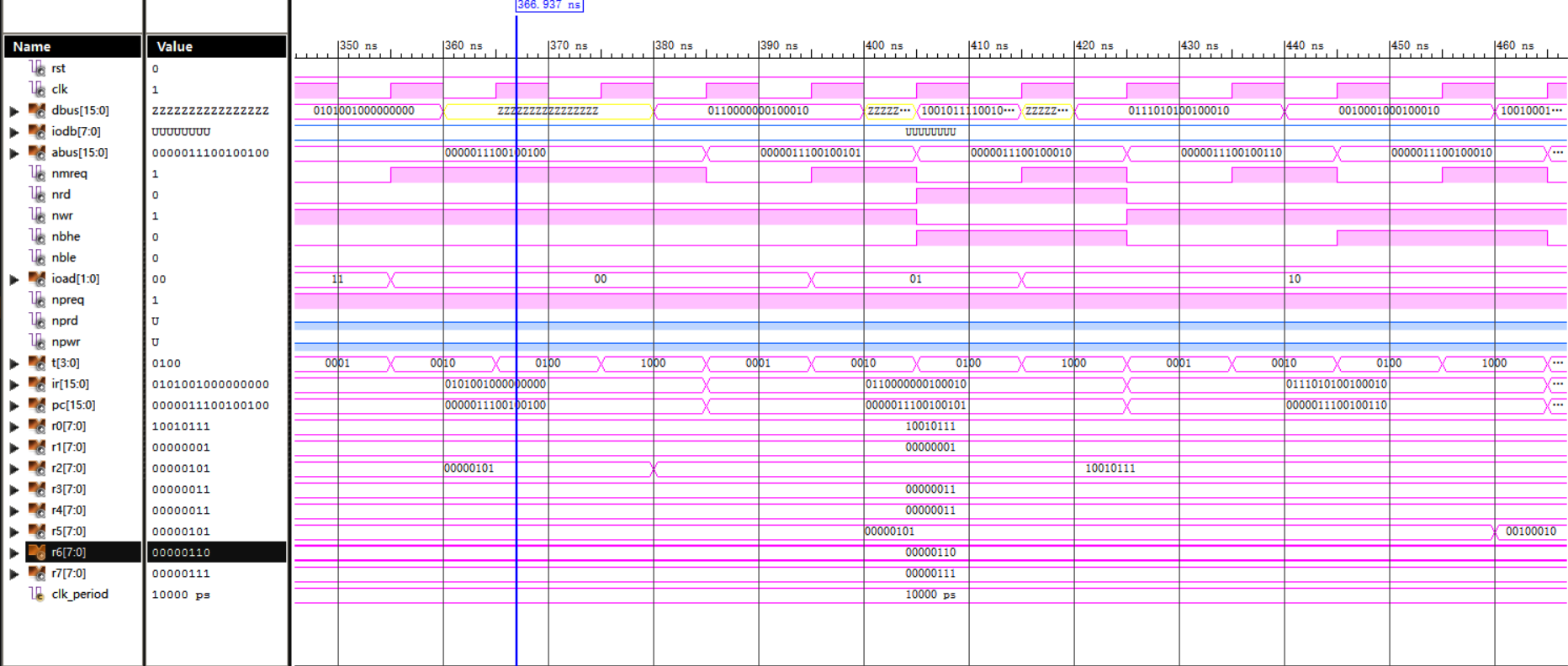
|  |  |  |
| --- | --- | --- |
| 指令 | 指令码 | 执行结果 |
| ADD R2 R3 | 0011001000000011 | R2=5 |
| SUB R4 R1 | 0010010000000001 | R4=3 |
| JMP FFH | 0000000011111111 | PC跳转到0000011111111111 |
| JZ R0 22 | 0001000000100010 | R0=0，故跳转 |
| JZ R1 22 | 0001000100100010 | R1！=0，不跳转，PC+1 |
| MVI R0 97 | 0100000010010111 | R0=97 |
| MOV R2 R0 | 0101001000000000 | R2=97 |
| STA R0 22 | 0110000000100010 | 将R0=97存到R7//Ad 地址单元 |
| LDA R5 22 | 0111010100100010 | 将R7//Ad地址单元取到R5，为22 |
| IN R1 01 | 1001000100000001 | 从外设端口01，输入33到R1 |
| OUT R2 02 | 1000001000000010 | 将R2输出到外设端口02 |

## 3.3 仿真波形

第1-3条指令：

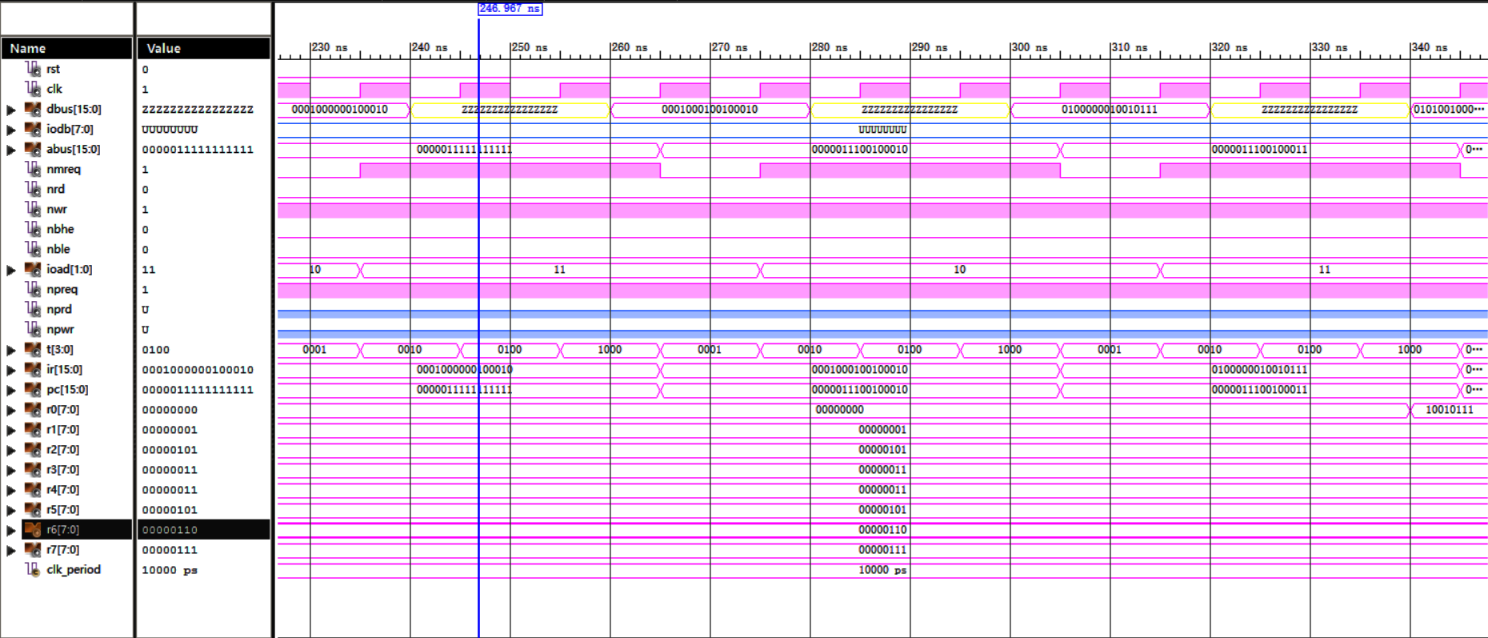


第4-6条指令：



第7-10条指令

：



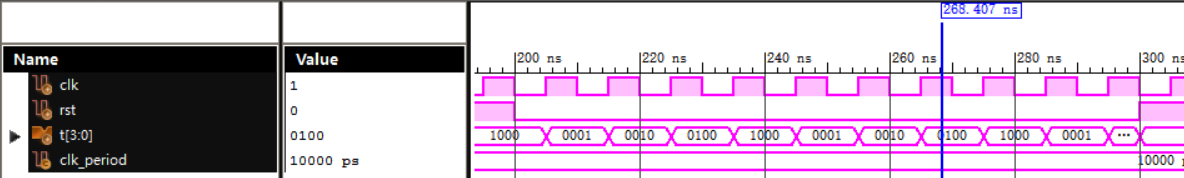
# 4 各模块系统测试

## 4.1 时钟管理模块

### 4.1.1 测试过程

先将Rst置为有效，时钟保持为1000，之后Rst置为无效，节拍发生器开始工作，10个节拍之后，再次Rst置为有效。

### 4.1.2 仿真波形

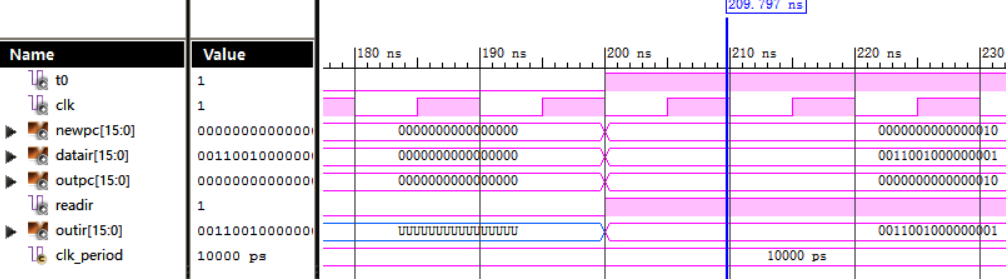


## 4.2 取指管理模块

### 4.2.1 测试过程

将 T0置为1，开始工作。给定新PC值 “0002”h，成功回写。给定IR指令“0011001000000001”，成功读入；

### 4.2.2 仿真波形



## 4.3 运算管理模块

### 4.3.1 测试过程

先将T1<=1，之后执行了Add R2 R1，两个周期后执行SUB R4 R1，两个周期之后，执行 JMP FFh，可以看到有效地址R7//Ad 计算为0000011111111111，正确。之后做了寄存器回写的验证，将“28”h回写到R6中。

### 4.3.2 仿真波形



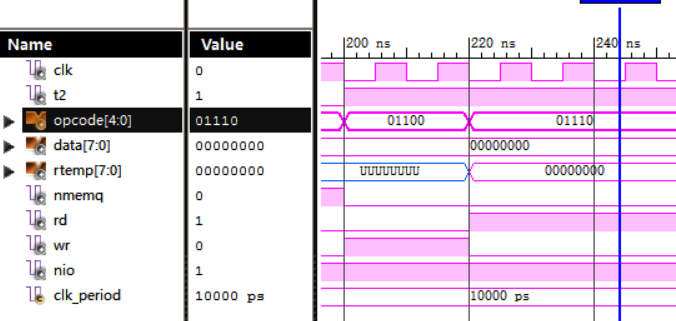
## 4.4 存储管理模块

### 4.4.1 测试过程

测试了STA 操作和LDA 操作，在执行STA操作时，成功发出主存片选有效，写有效信号。

在执行LDA操作时，成功发出主存片选有效，读有效信号。

### 4.4.2 仿真波形

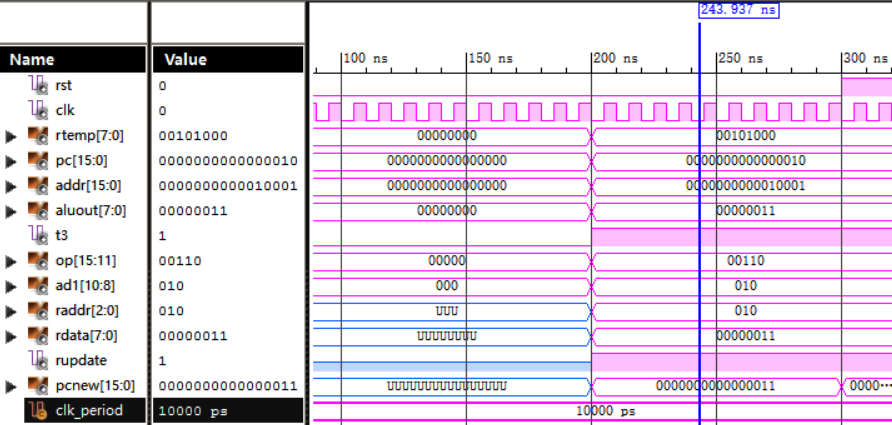


## 4.5 回写管理模块

### 4.5.1 测试过程

执行STA R2 “11” 指令，取回的数据为“03“h，观察波形，成功给出Pupdate信号，Raddr为010，Rdata 为03，PC执行+2操作成功。可验证回写管理模块能正确工作。

### 4.5.2 仿真波形



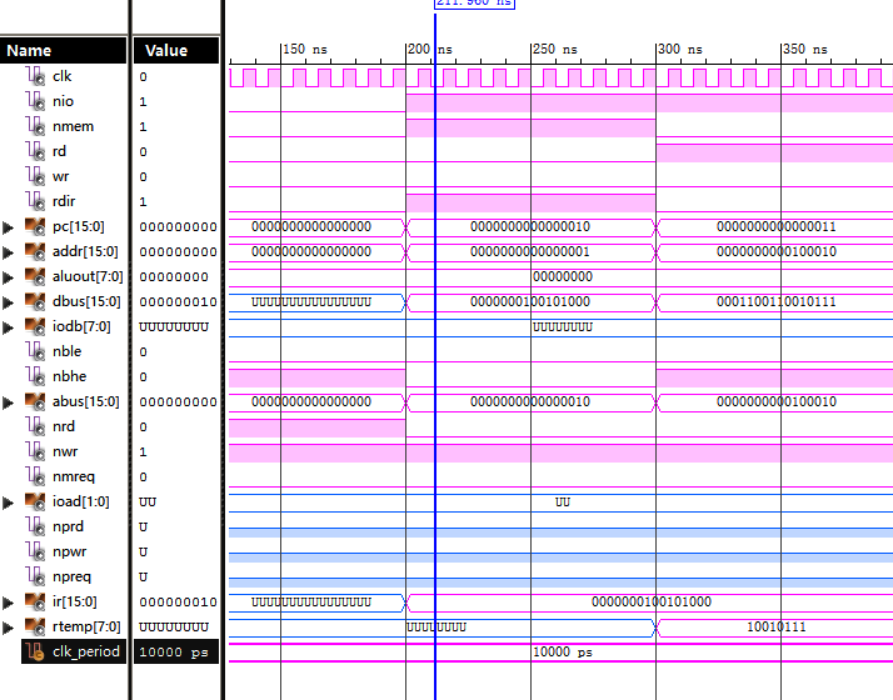
## 4.6 访存控制模块

### 4.2.1 测试过程

首先测试，在取指节拍时，Rdir有效，给入PC值“0002“h，成功取出指令”0128“h，并且各项内存控制信号正确给出。

再测试执行LDA指令时，nMEM有效，给入地址“0022“h，RD<=’1’,成功取回数据”1997“h。

### 4.2.2 仿真波形



# 5. 处理器功能测试程序

|  |  |  |  |
| --- | --- | --- | --- |
| 地址 | 指令 | 十六进制代码 | 寄存器/存储器值 |
| 0000 | MVI R7 X’0’ | 4700 | 0->R7 |
| 0001 | MVI R0 X’15’ | 4015 | 15->R0 |
| 0002 | LDA R1 X90 | 7190 | 3->R1 |
| 0003 | MOV R2 R1 | 5201 | 3->R2 |
| 0004 | ADD R0 R2 | 3002 | 18->R0 |
| 0005 | JMP X’10’ | 0010 | 跳转到0010 |
| 0010 | STA R0 X’91’ | 6091 | 18->[0091] |
| 0011 | SUB R0 R1 | 2001 | 15->R0 |
| 0012 | OUT R0 ‘00’ | 8000 | 5->K7-K4(四位指示灯) |
| 0013 | JZ R7 X’20’ | 1720 | R7=0,跳转 |
| 0020 | IN R3 ‘01’ | 9301 | 55->R3 |
| 0021 | OUT R3 ‘10’ | 8302 | 5->K7-K4(四位指示灯) |

# 6 问题及解决办法

实验过程中遇到了不少问题，现将主要问题记录如下：

1. 在取指管理模块取回IR之后送往其他模块之后，等待其他模块还未利用IR，IR总线上的指令就已经改变。解决办法是将Irout由边沿触发改为电平触发，使Irout总线上始终为当前指令码。
2. 在执行存数操作时，由于将地址和写指令一同放到存储管理模块，导致有时地址还未改变，就已执行写操作，导致写到前一个地址中。解决办法是，我将数据，地址一同放到运算管理模块给出，这样可以保证在写信号给出时，数据、地址都已准备就绪。
3. 在一些模块中，可能要在同一节拍中，做有时序关系的操作。我引入了时钟信号，充分利用一个节拍的时钟上下沿，来处理有时序关系的操作。
4. 在仿真中还遇到了尽管老师强调了很多次的三态门，在inout 总线上使用完之后需要置为高阻态。一开始只在代码中置为高阻，没有在测试中使用之后置为高阻，导致线变。查阅资料后，成功解决。
5. 下载到开发板时，发现没有充分的数码管来显示寄存器的值。解决办法是，利用K7-k4指示灯没有用到，将寄存器的第四位显示到其之上，达到验证的目的。