**实验报告**

1. 使用到的部件

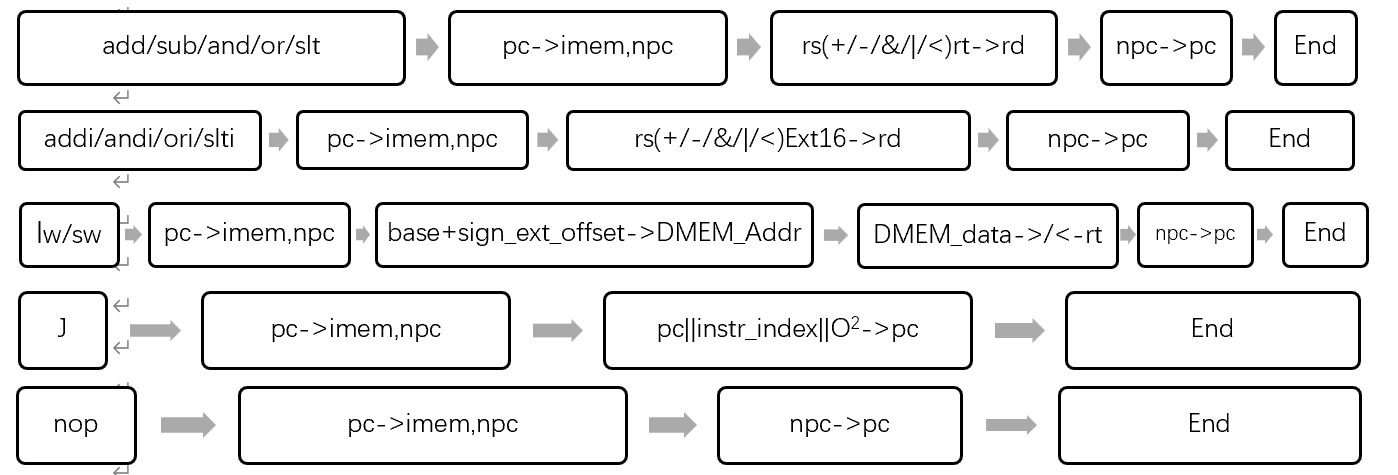
|  |  |
| --- | --- |
| 部件 | 用途 |
| PC | 存储下一条指令地址 |
| NPC | 使指令地址+4 |
| Imem | 指令寄存器 |
| RegFile | 寄存器堆 |
| ALU | 算数逻辑单元 |
| DMEM | 数据存储器 |
| EXT16，S\_EXT16 | 位扩展 |
| || | 拼接部件 |

二、数据通路的设计

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令 | pc | npc | imem | RegFile | ALU | | Ext16 | DMEM | | | S\_Ext16 | || | |
| rd | A | B | Addr | | Data | A | B |
| add | npc | pc | pc | ALU | rs | Rt |  |  |  | |  |  |  |
| sub | npc | pc | pc | ALU | rs | Rt |  |  |  | |  |  |  |
| and | npc | pc | pc | ALU | rs | Rt |  |  |  | |  |  |  |
| or | npc | pc | pc | ALU | rs | Rt |  |  |  | |  |  |  |
| slt | npc | pc | pc | ALU | rs | Rt |  |  |  | |  |  |  |
| addi | npc | pc | pc | ALU | rs | Ext16 | Imm16 |  |  | |  |  |  |
| slti | npc | pc | pc | ALU | rs | Ext16 | Imm16 |  |  | |  |  |  |
| andi | npc | pc | pc | ALU | rs | Ext16 | Imm16 |  |  | |  |  |  |
| ori | npc | pc | pc | ALU | rs | Ext16 | Imm16 |  |  | |  |  |  |
| lw | npc | pc | pc | Data | rs | S\_Ext16 |  | ALU |  | | offset |  |  |
| sw | npc | pc | pc |  | rs | S\_Ext16 |  | ALU | Rt | | offset |  |  |
| j | || | pc | pc |  |  |  |  |  |  | |  | PC  31-28 | Imem  25-0 |
| nop | npc | pc | pc |  |  |  |  |  |  | |  |  |  |

三、控制部件设计

1. 指令流程图



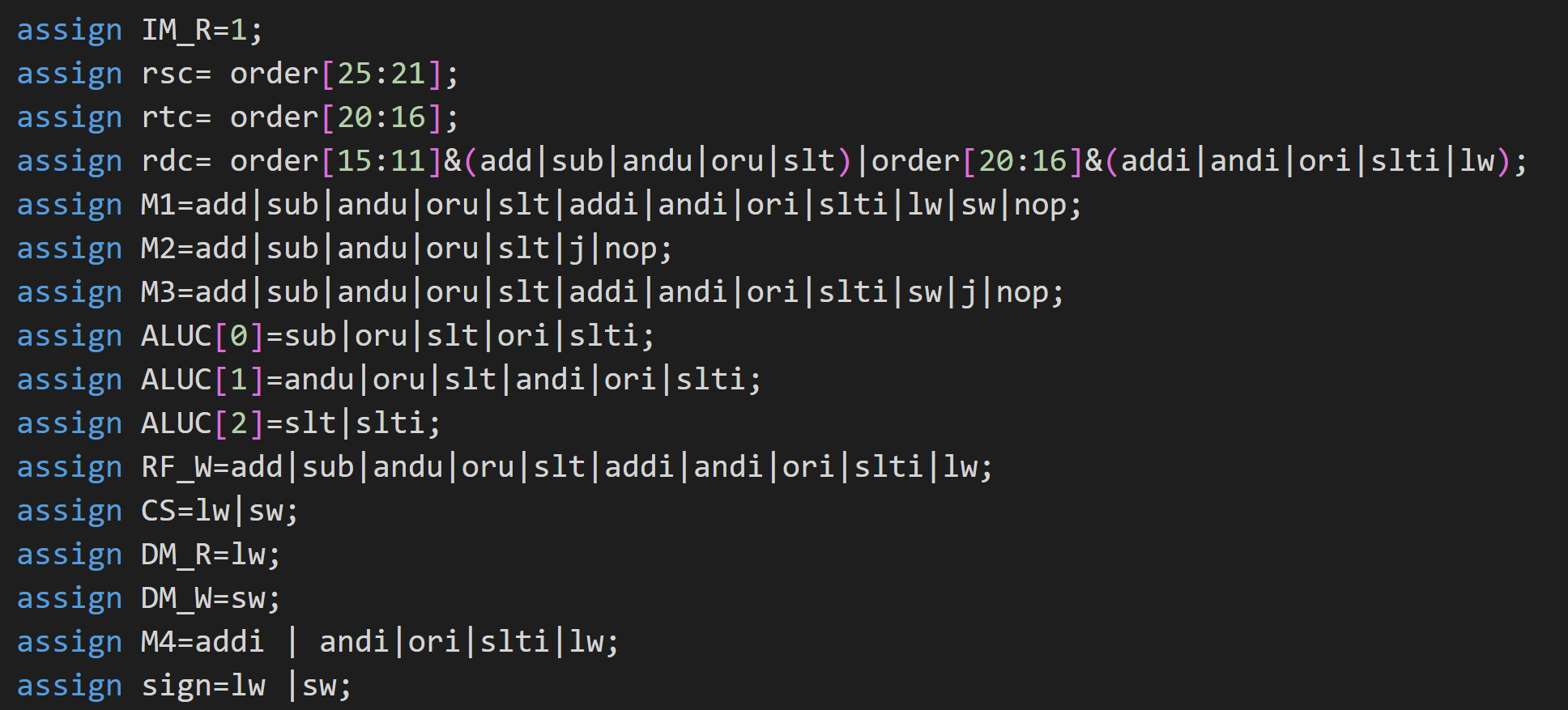
1. 指令操作时间表

|  |  |  |  |
| --- | --- | --- | --- |
| ALU功能控制 | | | |
| 操作 | ALUC2 | ALUC1 | ALUC0 |
| + | 0 | 0 | 0 |
| - | 0 | 0 | 1 |
| & | 0 | 1 | 0 |
| | | 0 | 1 | 1 |
| < | 1 | 0 | 0 |

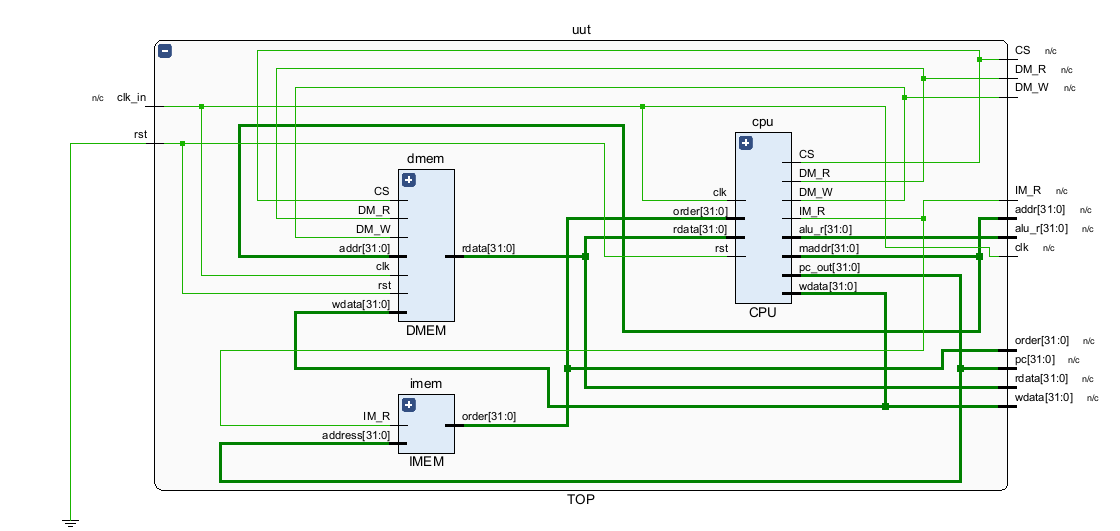
|  |  |
| --- | --- |
| 控制信号表 | |
| 信号 | **功能说明** |
| PC\_CLK | CPU工作主频 |
| IM\_R | Imem读信号 |
| rsc[4:0] | rs寄存器选择输入控制端 |
| rtc[4:0] | rt寄存器选择输入控制端 |
| rdc[4:0] | rd寄存器选择输入控制端 |
| M1 | MUX1选择器控制端 |
| M2 | MUX2选择器控制端 |
| M3 | MUX3选择器控制端 |
| ALUC0 | ALU控制端0 |
| ALUC1 | ALU控制端1 |
| ALUC2 | ALU控制端2 |
| RF\_W | RegFile写信号 |
| RF\_CLK | RegFile时钟 |
| CS | Dmem片选信号 |
| DM\_R | Dmem读信号 |
| DM\_W | Dmem写信号 |
| sign | Ext16有无符号 |

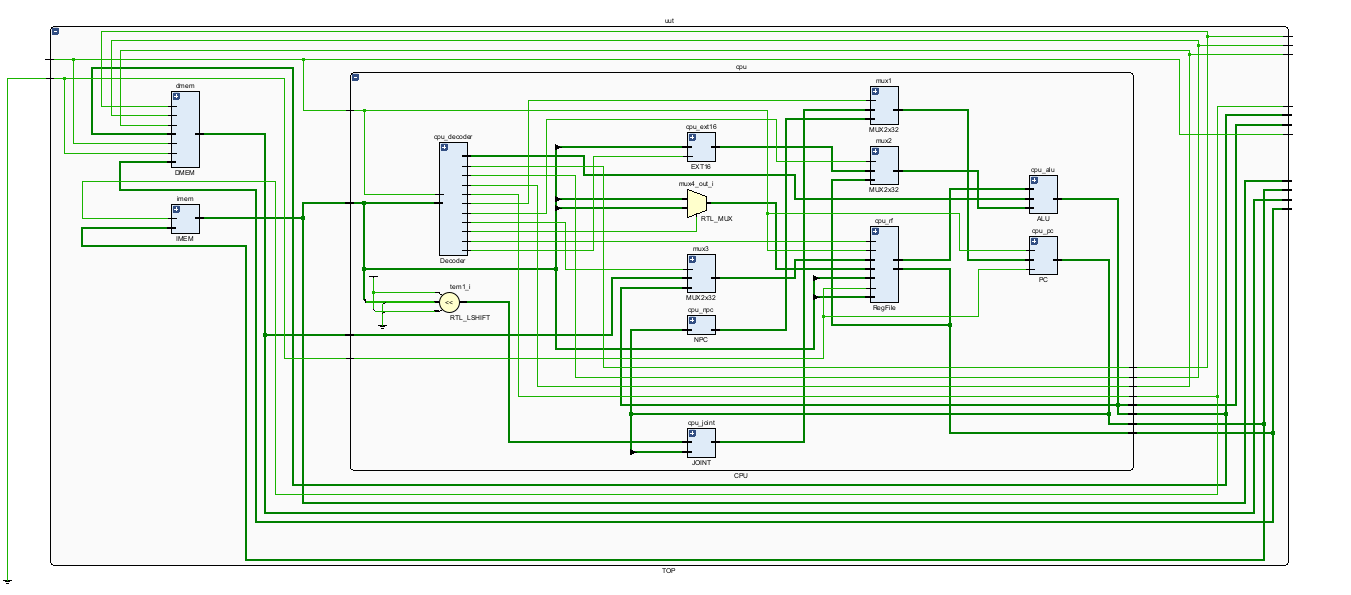
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 操作时间表 | | | | | | | | | | | | | |
| 信号 | **add** | **sub** | **and** | **or** | **slt** | **addi** | **andi** | **ori** | **slti** | **lw** | **sw** | **j** | **nop** |
| PC\_CLK | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| IM\_R | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| rsc[4:0] | IM25-21 | | | | | | | | | | |  |  |
| rtc[4:0] | IM20-16 | | | | |  |  |  |  |  | IM20-16 |  |  |
| rdc[4:0] | IM15~11 | | | | | IM20-16 | | | | |  |  |  |
| M1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| M2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| M3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| ALUC0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| ALUC1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| ALUC2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| RF\_W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| RF\_CLK | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| CS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| DM\_R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| DM\_W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Sign | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| M4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

1. 综合信号



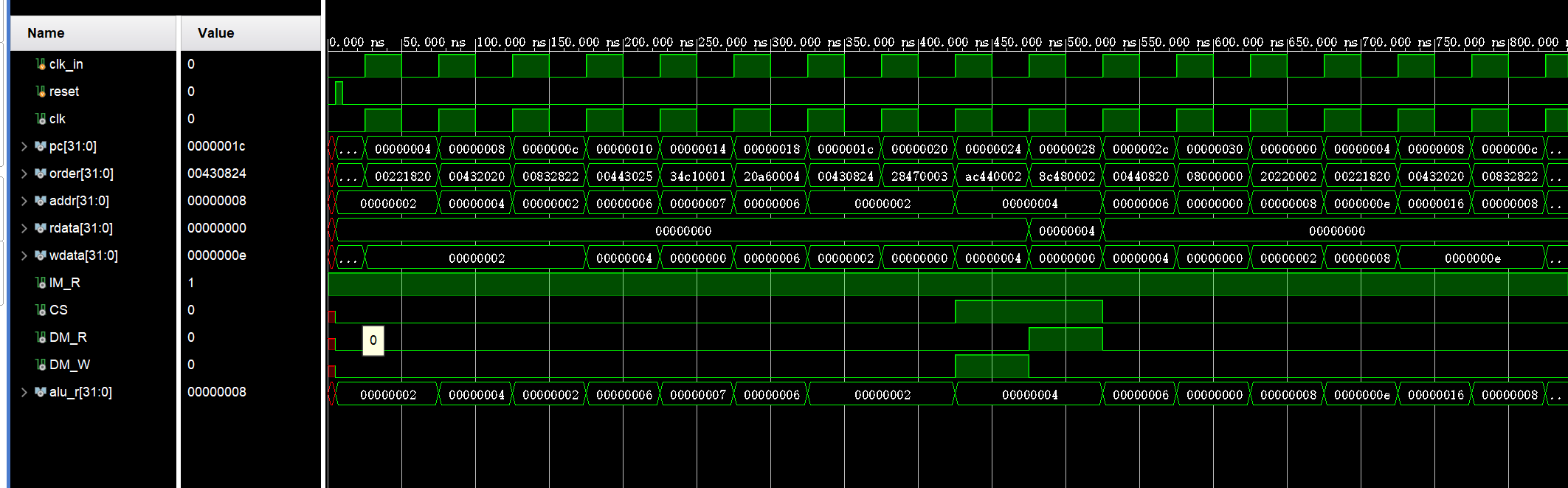
1. 电路图





1. 波形图

|  |  |  |
| --- | --- | --- |
| 测试指令: | Hex | 测试结果 |
| Addi %s2 %s1 2 | 20220002 | 成功 |
| Add %s3 %s1 %s2 | 00221820 | 成功 |
| Add %s4 %s2 %s3 | 00432020 | 成功 |
| Sub %s5 %s4 %s3 | 00832822 | 成功 |
| Or %s6 %s2 %s4 | 00443025 | 成功 |
| Ori %s1 %s6 1 | 34c10001 | 成功 |
| Andi %s6 %s5 4 | 20a60004 | 成功 |
| And %s1 %s2 %s3 | 00430824 | 成功 |
| Slti %s7 %s2 3 | 28470003 | 成功 |
| Sw %s2 %s4 2 | ac440002 | 成功 |
| Lw %s2 %s8 2 | 8c480002 | 成功 |
| Slt %s1 %s2 %s4 | 00440820 | 成功 |
| J | 08000000 | 成功 |



指令

ALU输出

指令地址

执行j命令成功跳转

Lw命令时成功读取数据

1. 感悟

这是一个非常有趣的项目。我学习到了如何实现指定MIPS指令功能的简单单周期CPU的电路设计。同时也需要我们对数字电路设计有一定的了解。希望捞捞。