多周期MIPS处理器设计报告

1. 数据通路设计

参考讲义图片，完善以下数据通路以支持7条指令。

要求：

1. 必须自己绘制该数据通路，可以用铅笔在白纸上手绘拍照贴图，也可以画图工具直接绘制。数据通路中所标注的信号名称必须与代码保持一致（建议Verilog代码中所有信号使用小写字母，不要像下图中大小写混用，因为Verilog语言是区分大小写的，全部使用小写不容易出错）。
2. 使用最小的硬件，即一个指令/数据混合的存储器、一个ALU，除此以外不得有其他存储器、加法器。
3. 提示：每个周期计算结果使用寄存器保存，如取指保存到instruction reg，读取的源操作数保存到bus a、bus b，ALU计算结果保存到alu out。这样可以使得主频较高。

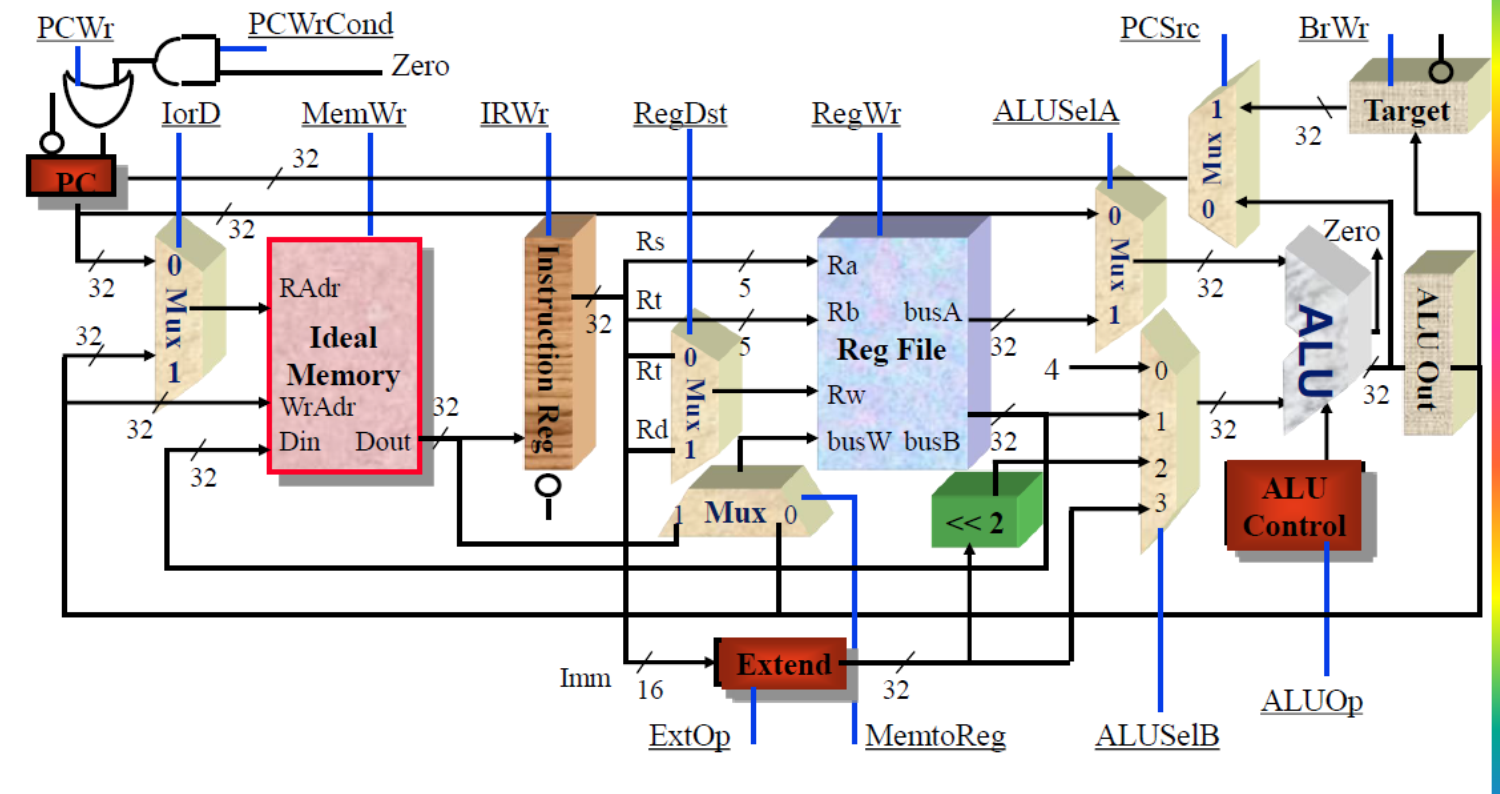


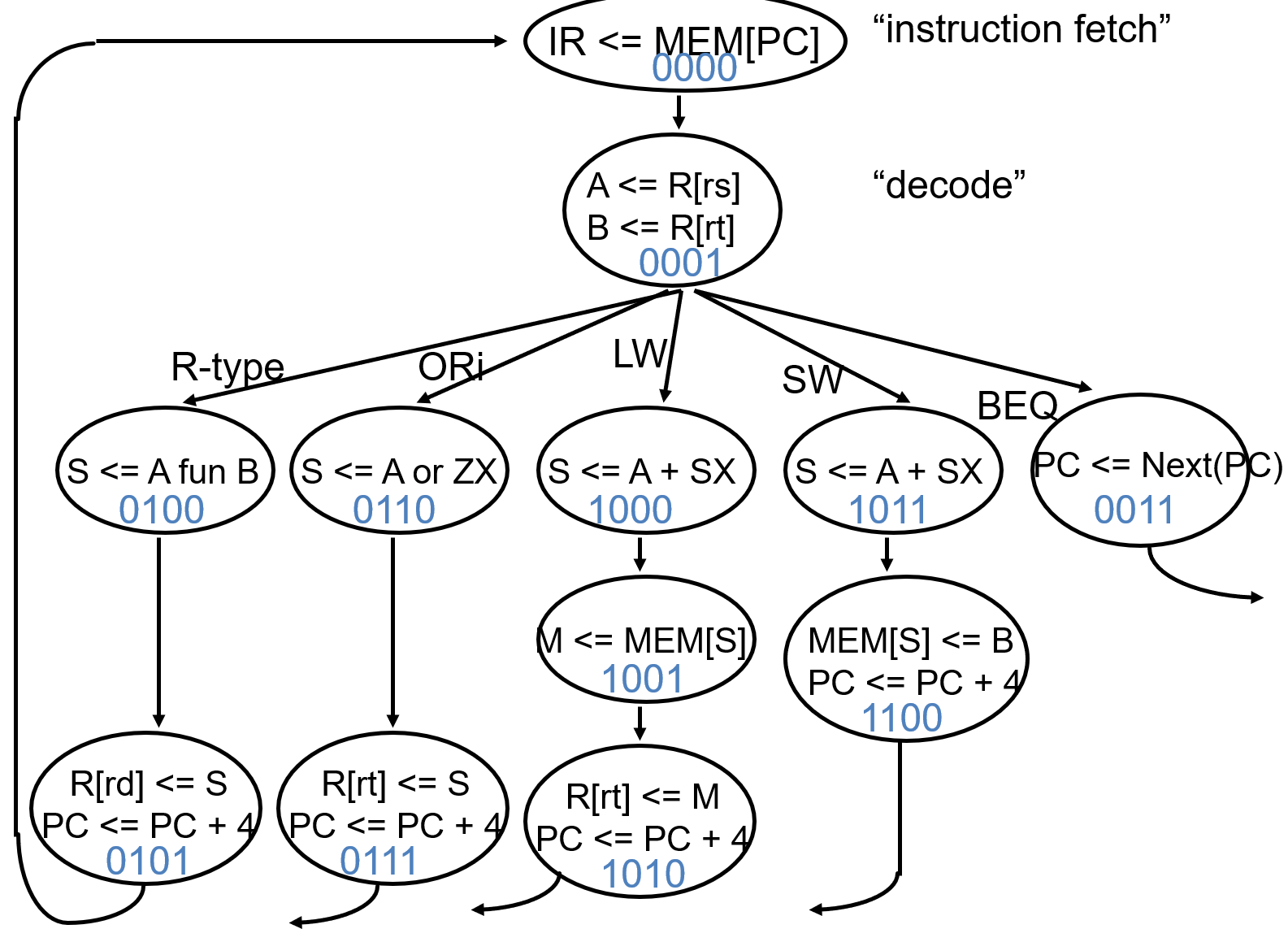
图 1 多周期MIPS处理器的数据通路

1. 控制通路设计

状态机设计：参考讲义图片，完善状态机以支持7条指令。

建议对状态机进行修改，将PC+4转到取指周期，并将写回结果放在ALU计算周期，这样可以省掉一个周期。

需要在状态机增加J指令的支持。建议和BEQ一样，增加一个J状态专门来处理J指令的执行。



按照数据通路中的控制点，补全下表说明每条状态的控制信号值。

表 1 每个状态下的控制信号取指

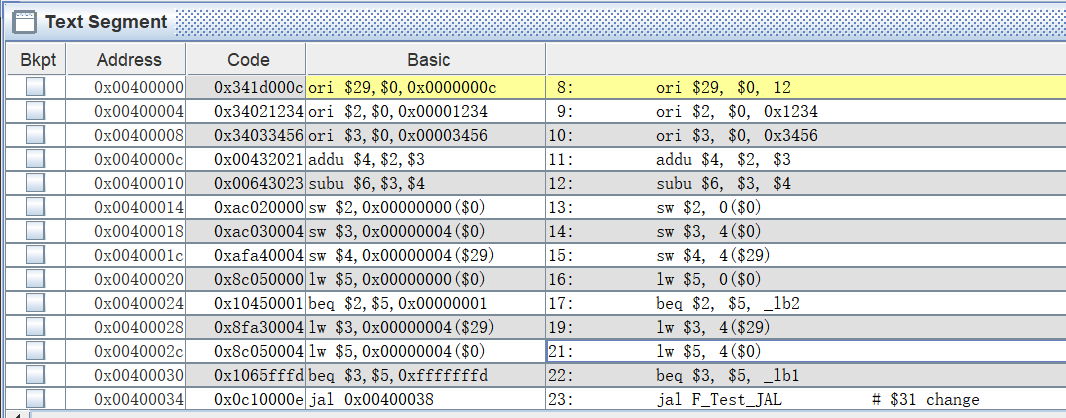
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 右边是状态名称  下面是控制信号名称 | Fetch | Decode | Rtype | Ori | lw | Sw | Beq | Rwrite | Oriwrite | Lwread | Lwwrite | Swwrite |
| pcwr |  |  |  |  |  |  |  |  |  |  |  |  |
| pcwrcond |  |  |  |  |  |  |  |  |  |  |  |  |
| iord |  |  |  |  |  |  |  |  |  |  |  |  |
| Memwr |  |  |  |  |  |  |  |  |  |  |  |  |
| 其他控制信号 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

1. 仿真结果
2. MARS编写的MIPS汇编程序

可以继续使用单周期MIPS中使用的汇编程序。

在MARS中汇编通过之后截屏如下：

必须自己按照7条指令自己编写汇编程序，不能完全照搬下面的示范！



1. 仿真波形

此处粘贴重要仿真结果波形。

1. 实现结果（可选）

Quartus软件熟悉的同学可以粘贴布局布线结果，包括资源占用率、最高主频等实现结果，建议与单周期实现结果对比，资源占用应该降低，最高主频应该提升。

|  |  |  |
| --- | --- | --- |
|  | 单周期实现结果 | 多周期实现结果 |
| ALUs |  |  |
| Registers |  |  |
| Fmax |  |  |

1. 评分标准

从设计思路、仿真结果、实现结果等方面进行综合评价，全部满分共计24分。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1、设计思路 | 10 | 8 | 6 | 4 | 0 |
| 查看设计文档思路是否清晰，描述是否完整 | 优秀，数据通路描述准确、控制通路表格正确 | 很好，文档较为清楚 | 及格，文档较完整 | 待改进，文档不清或错误较多 | 无文档 |
| 2、仿真结果 | 10 | 8 | 6 | 4 | 0 |
| 主要通过测试激励进行评估，可以用提交者的测试激励，也可以用自己的测试，验证功能是否正确。 | 优秀，7条指令100%验证通过（尽量用自己的测试程序，与提交的测试程序进行交叉验证） | 很好，1~2条错误 | 及格，3条指令错误 | 待改进，50%以上功能点错误 | 基本不对，编译不通过等 |
| 3、实现结果 | 10 | 8 | 6 | 4 | 0 |
| 是否有Quartus布线结果 | - | - | - | 有最高主频、面积占用率等信息 | 无 |

附录：提供代码的说明。

本练习提供了部分源代码，如果你能看懂，最好使用这些代码节省编程工作量。如果你不能看懂，那么可以自行编写代码。提供的代码仅供参考，里面有些代码是多余的功能（例如ALU模块），可以忽略，最终确保7条指令功能正确即可。

基本与单周期相同，主要变化如下：

1. 增加了部分寄存器，用于保存每个周期的计算结果；
2. 重点修改了mipv.v，其中增加了FSM优先控制状态机描述。
3. Mem.v：指令数据混合在一起的存储器，使用mem.txt进行初始化，其中前面一半用做代码存储，后面一半用做数据存储。当然不一定严格对半分，只要保证代码和数据地址区间分开即可。建议mem.txt主要对代码区进行初始化，数据区可以不初始化，测试时先用SW指令写入数据，然后再测试LW是否可以读出，如果读出结果与写入数据相同，说明LW和SW都是正确的。删除了dmem中大家困惑的be信号，此信号lw和sw指令是没用的，但实现lb、sb等指令时需要。
4. 其他说明请参考单周期MIPS处理器说明文档。