

Project 3 VerilogHDL 完成五级流水线处理器开发

一、 设计说明

1. 完成以下指令集

初级：（80 以下）

a) 完成如下指令（若遗 proj2 的指令，自行补齐）：

算数运算指令：

add , addi , addu , addiu

sub , subu

slt , slti , sltu , sltiu

逻辑运算指令：

and , andi , or , ori

lui , nor , xor , xori

移位指令：

sll , sllv

srl , srlv , sra , srav

分支跳转指令：

beq , bne

bgez , blez

j , jr

访存指令：

lb , lh , lw

sb , sh , sw

a) b) 处理器支持溢出异常，保留指令例外。

高级：（95 分）

b) 57 条指令（参见文末附表）

算数运算指令：

add , addi , addu , addiu

sub , subu

slt , slti , sltu , sltiu

mult , multu , div , divu

逻辑运算指令：

and , andi , or , ori

lui , nor , xor , xori

移位指令：

sll , sllv

srl , srlv , sra , srav

分支跳转指令：

beq , bne

bgez , bgtz , blez , bltz , bgezal , bltzal

j , jr , jal , jalr

数据移动指令：

mfhi , mflo , mthi , mtlo

自陷指令：

break , syscall

访存指令：

lb , lbu , lh , lhu , lw

sb , sh , sw

特权指令:

eret , mfc0 , mtc0

- c) 处理器支持多种异常，要求完成整型溢出例外，系统调用例外，保留指令例外。
2. 处理器为流水线设计，支持各类冒险与转发。

二、 设计要求

3. 流水线处理器由 datapath(数据通路)和 controller(控制器)组成。
- a) 数据通路在单周期功能部件的基础上，增加了相应的流水寄存器。
 - b) IM: 容量为 8KB(32bit×2048 字)。
 - c) DM: 容量为 8KB(32bit×2048 字)。
4. Figure1 为供你参考的处理器架构图。
- a) 我们不确保 Figure1 是完全正确的；我们也不确保 Figure1 能够满足上述指令集。
 - b) 鼓励你从数据通路的功能合理划分的角度自行设计更好的数据通路架构。
 - c) 如果你做了比较大的调整，请务必注意不要与要求 5 矛盾。

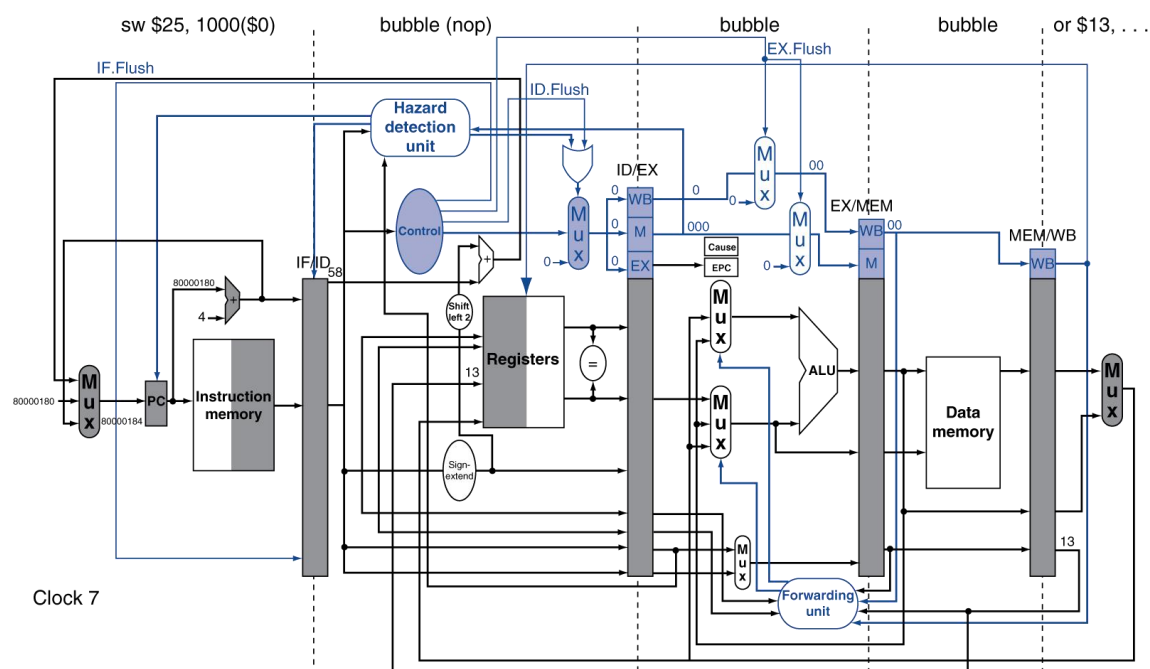


Figure1 数据通路(供参考)

5. 整个 project 必须采用模块化和层次化设计。
 - a) 顶层设计文件命名为 myCpu.v。
 - b) 建议 datapath 中的每个 module 都由一个独立的 VerilogHDL 文件组成。
 - c) 控制器的设计与单周期非常相似，除了要保留对应的控制信号到下一级流水阶段。也可以考虑用分布式控制器(参考高老师 ppt)
6. code.txt 中存储的是指令码

a) 用 VerilogHDL 建模 IM 时，必须以读取文件的方式将 code.txt 中指令加载至 IM 中。

b) code.txt 的格式如 Figure3 所示。每条指令占用 1 行，指令二进制码以文本方式存储。

```

1 34010001
2 34020008
3 34100000
4 34110008
5 3c12aabb
6 12200009

```

Figure3code.txt 文件格式

7. 为使得代码更加清晰可读，建议多使用宏定义，并将宏定义组织在合理的头文件中。
8. PC 复位后初值为 0x0000_3000，目的是与 MARS 的 Memory Configuration 相配合。

a) 教师用测试程序将通过 MARS 产生，其配置模式如 Figure4 所示。



Figure4MIPS 存储配置模式(MARS memory configuration)

三、 模块定义【WORD】

9. 仿照下面给出的 PC 模块定义，给出所有功能部件的模块定义。

10. PC 模块定义(参考样例)

(1) 基本描述

PC 主要功能是完成输出当前指令地址并保存下一条指令地址。复位后，PC 指向 0x0000_3000，此处为第一条指令的地址。

(2) 模块接口

信号名	方向	描述
NPC[31:2]	I	下条指令的地址
clk	I	时钟信号
Reset	I	复位信号。 1: 复位 0: 无效

PC[31:2]	O	30 位指令存储器地址(最低 2 位省略)
----------	---	-----------------------

(3) 功能定义

序号	功能名称	功能描述
1	复位	当复位信号有效时，PC 被设置为 0x0000_3000。
2	保存 NPC 并输出	在每个 clock 的上升沿保存 NPC，并输出。

11. 下列模块必须严格满足如下的接口定义：

a) 你必须在 VerilogHDL 设计中建模这 3 个模块。

b) 不允许修改模块名称、端口各信号以及变量的名称/类型/位宽。

文件	模块接口定义
mips.v	<pre>module mips(clk, rst) ; input clk ; // clock input rst ; // reset</pre>
im.v	<pre>im_4k(addr, dout) ; input [11:2] addr ; // address bus output [31:0] dout ; // 32-bit memory output reg [31:0] im[1023:0] ;</pre>
dm.v	<pre>dm_4k(addr, din, we, clk, dout) ; input [11:2] addr ; // address bus input [31:0] din ; // 32-bit input data input we ; // memory write enable input clk ; // clock output [31:0] dout ; // 32-bit memory output reg [31:0] dm[1023:0] ;</pre>

四、 测试要求

12. 所有指令都应被测试充分。

13. 详细说明你的测试程序原理及测试结果。【WORD】

a) 应明确说明测试程序的测试期望，即应该得到怎样的运行结果。

b) 每条汇编指令都应该有注释。

五、 其他要求

14. 打包文件：VerilogHDL 工程文件、code.txt、code.txt 所对应的汇编程序、项目报告。

15. 时间要求：待定！！

六、 成绩及实验测试要求

16. 实验成绩包括但不限于如下内容：初始设计的正确性、增加新指令后的正确性、实验报告等。
17. 实验测试时，你必须已经完成了处理器设计及开发。
 - a) 允许实验报告可以未完成。
18. 实验测试时，你需要展示你的设计并证明其正确性。
 - a) 应简洁的描述你的验证思路，并尽可能予以直观展示。
19. 实验指导教师会临时增加 1~2 条指令，你需要在**规定时间**内完成对原有设计的修改，并通过实验指导教师提供的测试程序。

七、 开发与调试技巧

20. 对于每条指令，请认真阅读《MIPS32® Architecture For Programmers Volume II: The MIPS32® Instruction Set》!
 - a) 如果测试时，你无法清楚的解释所要求的指令，测试成绩将减一档！
21. 建议先在 MARS 中编写测试程序并调试通过。注意 memory configuration 的具体设置。
 - a) 你应该加载 code.txt 至指令存储器以测试你的处理器设计。假设你的处理器设计是正确的。
 - b) 你需要**参照 Figure4 设置 MARS**，否则该程序将无法运行。
22. 利用\$readmemh 系统任务可以给存储器初始化数据。例如可以把 code.txt 文件中的数据加载至 my_memory 模块。

```
reg [31:0] my_memory[1023:0] ;
```

```
initial
```

```
$readmemh( "code.txt", my_memory ) ;
```

23. 有时我们需要较为集中的在顶层 testbench 中**观察甚至修改**下层模块的变量，那么你可以通过使用层次路径名来非常方便的达到这一目的。例如：

```
module testbench ;
```

```
    Ch1C1(...) ;
```

```
    $display(C1.Art) ;
```

```

endmodule

moduleChil(...) ;
    reg Art;

    ...
endmodule

```

完成的指令集：

表 3-1 算术运算指令

指令名称格式	指令功能简述
ADD rd, rs, rt	加（可产生溢出例外）
ADDI rt, rs, immediate	加立即数（可产生溢出例外）
ADDU rd, rs, rt	加（不产生溢出例外）
ADDIU rt, rs, immeidate	加立即数（不产生溢出例外）
SUB rd, rs, rt	减（可产生溢出例外）
SUBU rd, rs, rt	减（不产生溢出例外）
SLT rd, rs, rt	有符号小于置 1
SLTI rt, rs, immediate	有符号小于立即数设置 1
SLTU rd, rs, rt	无符号小于设置 1
SLTIU rt, rs, immediate	无符号小于立即数 ^[1] 设置 1
DIV rs, rt	有符号字除
DIVU rs,rt	无符号字除
MULT rs, rt	有符号字乘
MULTU rs, rt	无符号字乘

表 3-2 逻辑运算指令

指令名称格式	指令功能简述
AND rd, rs, rt	位与

指令名称格式	指令功能简述
ANDI rt, rs, immediate	立即数位与
LUI rt,immediate	寄存器高半部分置立即数
NOR rd, rs, rt	位或非
OR rd, rs, rt	位或
ORI rt, rs, immediate	立即数位或
XOR rd, rs, rt	位异或
XORI rt, rs, immediate	立即数位异或

表 3-3 移位指令

指令名称格式	指令功能简述
SLL rd, rt, sa	立即数逻辑左移
SLLV rd, rs, rt	变量逻辑左移
SRA rd, rt, sa	立即数算术右移
SRAV rd, rs, rt	变量算术右移
SRL rd, rt, sa	立即数逻辑右移
SRLV rd, rs, rt	变量逻辑右移

表 3-4 分支跳转指令

指令名称格式	指令功能简述
BEQ rs, rt, offset	相等转移
BNE rs, rt, offset	不等转移
BGEZ rs, offset	大于等于 0 转移
BGTZ rs, offset	大于 0 转移
BLEZ rs, offset	小于等于 0 转移
BLTZ rs, offset	小于 0 转移
BLTZAL rs, offset	小于 0 调用子程序并保存返回地址

指令名称格式	指令功能简述
BGEZAL rs, offset	大于等于 0 调用子程序并保存返回地址
J target	无条件直接跳转
JAL target	无条件直接跳转至子程序并保存返回地址
JR rs	无条件寄存器跳转
JALR rd, rs	无条件寄存器跳转至子程序并保存返回地址下

表 3-5 数据移动指令

指令名称格式	指令功能简述
MFHI rd	HI 寄存器至通用寄存器
MFLO rd	L0 寄存器至通用寄存器
MTHI rs	通用寄存器至 HI 寄存器
MTLO rs	通用寄存器至 L0 寄存器

表 3-6 自陷指令

指令名称格式	指令功能简述
BREAK	断点
SYSCALL	系统调用

表 3-7 访存指令

指令名称格式	指令功能简述
LB rt, offset(base)	取字节有符号扩展
LBU rt, offset(base)	取字节无符号扩展
LH rt, offset(base)	取半字有符号扩展
LHU rt, offset(base)	取半字无符号扩展
LW rt, offset(base)	取字
SB rt, offset(base)	存字节
SH rt, offset(base)	存半字

指令名称格式	指令功能简述
SW rt, offset(base)	存字

表 3-8 自陷指令

指令名称格式	指令功能简述
ERET	例外处理返回
MFC0	读 CP0 寄存器值至通用寄存器
MTC0	通用寄存器值写入 CP0 寄存器