# 实验七：VerilogHDL 开发流水线处理器

## 目标：

1. 使用Verilog开发基于流水线的处理器。
2. 使用Vivado对开发的处理器进行模拟仿真。
3. 编写测试用汇编，使用Mars工具汇编成为机器代码，并使用它调试测试你开发的处理器。

## 要求

1. 请不要抄袭，可以与同学讨论，但不要直接抄袭同学的代码和实验报告。
2. 请认真完成实验报告，并在你认为关键的位置插入屏幕截图。
3. 请在截至日期（初定为2019.12.29日23：55）前将**Verilog源代码和实验报告**提交至 Unicourse+ 上。

## 说明

1. 处理器应支持MIPS-Lite2指令集。
    - MIPS-Lite1 = { addu, subu, ori, lw, sw, beq, lui }
    - MIPS-Lite2 = { MIPS-Lite1, jal, jr， **j**}
2. 处理器为流水线设计。
3. 我为你提供了参考的流水线顶层设计视图：

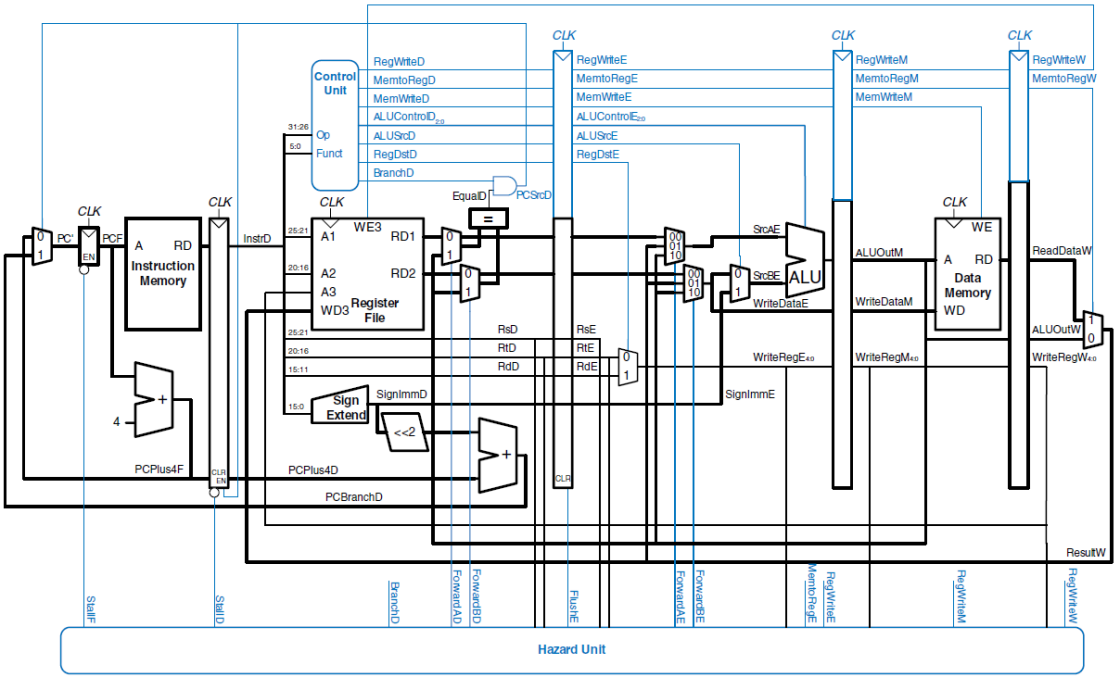


Figure 7.58 Pipelined processor with full hazard handling

    - 注意：本视图仅供参考，不能完整支持本次实验全部指令。
    - 注意：请将控制器（Control Unit）和冒险处理单元（Hazard Unit）分离。
    - *注意：lw和sw指令也存在数据冒险。*
4. 流水线 的设计以追求性能为第一目标，因此 的设计以追求性能为第一目标，因此 必须尽最大可能支持转发以解决数据冒险。
5. 为了解决数据冒险而设计的转发数据来源必须是**某级流水线寄存器**，不允许从功能部件的输出开始。

- 例如，ALU的输出不允许**直接**作为转发输入，转发的输入只能是是ALUOutM。

6. 基本模块的要求与定义规范与单周期设计相同。**注意命名规范**。

7. 测试说明

   - 你编写的汇测试程序必须确保**所有指令**都应被测试充分。
   - **相关性**是你的测试重点。测试程序必须去充分测试数据相关和分支相关。构造相关性测试的基本思路是对**各类**指令的组合（读操作数的位置，产生有效结果的位置）。
   - 下面是测试流水线addu指令相关性正确的一个例子 。(这个例子不能满足你全部的测试需求！)
     - 由于 addu最后必须读取操作数的位置在EX阶段， 那么与之相关的指令就只能是 N-1(MEM阶段 )，N-2(WB阶段 )。即需要测试让第 N-1 条指令或第 N-2 条指令的回写寄存器与addu的读取寄存器相同。
     - 能够产生有效结果的指令只是运算类和lw指令，因此对于addu指令来说**至少**(事实上还要更多的测试组合)包括如下测试用例 。

| 用例编号 | 测试类型 | 前序指令 | 冲突位置 | 冲突寄存器 | 测试序列 |
|---|---|---|---|---|---|
| 1 | R-M-RS | subu | MEM | rs | subu $1, $2, $3<br>addu $4, $1, $2 |
| 2 | R-M-RT | subu | MEM | rt | subu $1, $2, $3<br>addu $4, $2, $1 |
| 3 | R-W-RS | subu | WB | rs | subu $1, $2, $3<br>instru 无关<br>addu $4, $1, $2 |
| 4 | R-W-RT | subu | WB | rt | subu $1, $2, $3<br>instru 无关<br>addu $4, $2, $1 |
| 5 | I-M-RS | ori | MEM | rs | ori $1, $2, 1000<br>addu $4, $1, $2 |
| 6 | I-M-RT | ori | MEM | rt | ori $1, $2, 1000<br>addu $4, $2, $1 |
| 7 | I-W-RS | ori | WB | rs | ori $1, $2, 1000<br>instru 无关<br>addu $4, $1, $2 |
| 8 | I-W-RT | ori | WB | rt | ori $1, $2, 1000<br>instru 无关<br>addu $4, $2, $1 |
| 9 | LD-M-RS | | | | |
| 10 | LD-M-RT | | | | |
| 11 | LD-W-RS | | | | |
| 12 | LD-W-RT | | | | |

     - 当你按照上述方法构造测试用例时，其工作本质是进行覆盖性分析。做覆盖性分析时未必要逐条指令分析，可以按照类别来。指令分类不应是简单的R-I-J三类，而应该从指令功能来出发。
     - 你应该仿照上述思路自行构造测试asm，你应该先进行各类冲突的组合分析，并**在实验报告中给出类似上表的测试用例表**，然后才根据你的表格构造编写测试用例asm。
     - 注意：你应在实验报告中解释你的测试程序原理，预期输出与测试结果。对于函数相关指令也需要构造循环测试（与单周期相同）。

8. 50条指令指令集为MIPS-C3指令集。

   - MIPS-C3={ LB, LBU, LH, LHU, LW, SB, SH, SW, ADD, ADDU, SUB, SUBU, MULT, MULTU, DIV, DIVU, SLL, SRL, SRA, SLLV, SRLV, SRAV, AND, OR, XOR, NOR, ADDI, ADDIU, ANDI, ORI,

XORI, LUI, SLT, SLTI, SLTIU, SLTU, BEQ, BNE, BLEZ, BGTZ, BLTZ, BGEZ, J, JAL, JALR, JR, MFHI, MFLO, MTHI, MTLO }。

- 50条指令的其余详细技术说明会在下周之前给出。

9. 提示：在没有想清楚流水线的时候就在VerilogHDL层次进行编码是不明智的。前人大量经验与教训告诉我们， 对于大工程而言，在代码层次上做设计是短视的行为，表面看起来可以在早期迅速看到初步结果，但从整个项目的过程来看，开发人员将付出巨大的频繁试错的代价。我强烈建议你将设计与实现阶段分离。
    - 设计：在EXCEL之类的工具中进行详尽的设计并机型逻辑推演，是高效率的设计方法。
    - 实现：当你认为你自己的设计没有问题或基本没有问题时，再用VerilogHDL将你的设计描述出来。