

Project8 VerilogHDL 开发 MIPS 微系统(支持设备与中断)

V1.3@2014.02.21

高小鹏

V1.3: 修改说明 (刘铮同学)

- 1) 第 16 项: 修改了 DM 地址范围
- 2) 第 25 项: 进一步明确了中断服务程序加载方法
- 3) 第 38 项: 如何在 MARS 中设置异常处理程序地址

V1.2: 修改说明 (感谢 ChengChung 同学)

- 1) 第 16 项: 增加了定时器的地址

V1.1: 修改说明 (感谢 ChengChung 同学)

- 2) 第 2 项: 增加了 SLT 和 SLTU
- 3) 第 15 项: 修改 DM 容量。

一、 设计说明

1. MIPS 微系统应包括: MIPS 处理器、系统桥和 1 个定时器。
2. MIPS 处理器应支持 MIPS-C5 指令集。
 - a) 除 BREAK 和 SYSCALL 外, MIPS-C5 包括了 MIPS-C 的绝大多数指令。
 - b) MIPS-C5={ 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、ERET、MFC0、MTC0}。
 - c) 所有运算类指令均可以不支持溢出; 所有指令均可以不支持异常。
3. MIPS 处理器为多周期设计。
4. MIPS 微系统支持定时器硬件中断。

批注 [GXP1]: 1.支持了这些指令, 基本上就可以用 C 语言编写简单的测试程序, 然后用 GCC 编译 C 程序产生 MIPS 汇编了。具体内容参见第 27 条。

批注 [gxp2]: 【补充】之前遗漏

批注 [gxp3]: 【补充】之前遗漏

二、 系统桥与设备

5. 为了支持设备, MIPS 微系统需要配置系统桥。系统桥的相关内容请参考《L15.1-支持 IO.pptx》。

- a) 只需要支持 1 个设备，即定时器即可。
- 6. MIPS 微系统中只包括 1 个设备，即定时器。
 - a) 定时器的设计规范请参看《COCO 定时器设计规范.docx》。

三、 中断机制

- 7. 为了支持异常和中断，处理器必须实现 0 号协处理器(CP0)。为此，必须实现的 CP0 寄存器包括：SR、CAUSE、EPC、PrID。关于这几个寄存器，请大家阅读《L15-MIPS 系统结构.pptx》中的相关内容。
- 8. 除了阅读 PPT，你还应该阅读《See MIPS Run Linux》的第 3 章和第 5 章内容。由于这 2 章内容较多，因此建议：
 - a) 第 3 章：以 SR 为例，并非所有内容都需要学习。你应先阅读完 PPT 中的 SR 后，再以 PPT 内容为索引再详细阅读第 3 章中的相关内容。其他寄存器类似。**不需要阅读**的内容包括：3.4。
 - b) 第 5 章**建议阅读**的内容包括：5.1~5.5，5.7。
 - c) 所有与下列关键词相关的内容都**不需要阅读**：64 位、MMU、TLB、Cache、软中断、虚拟地址等。
- 9. 虽然《See MIPS Run Linux》的表 5.1 定义了 MIPS 的异常入口，但考虑到简化以及与 MARS 模拟器一致，我们将只支持 0x00004180 这个入口地址，即所有的异常与中断都从这里进入。
 - a) 你需要修改 NPC 模块，以确保当异常/中断发生时，NPC.NPC 输出 0x00004180。
- 10. 由于本系统只有 1 个中断/异常入口地址，因此只要求支持设备中断。MIPS 内部异常(如指令错误)则不被考虑。

四、 微系统设计

- 11. MIPS 处理器需要增加接口信号。以下为参考设计(只列出了新增的信号)：

信号名	方向	描述
PrAddr[31:2]	O	32 位地址总线(最低 2 位忽略)
BE[3:0]	O	4 位字节使能
PrDIn[31:0]	I	从 Bridge 模块读入的数据
PrDOut[31:0]	O	输出至 Bridge 模块的数据
Wen	O	时钟信号

HWInt[7:2]	I	6 个硬件中断请求
------------	---	-----------

12. 多周期处理器由 datapath(数据通路)和 controller(控制器)组成。

- a) 数据通路应至少包括如下 module: PC(程序计数器)、NPC(NextPC 计算单元)、GPR (通用寄存器组, 也称为寄存器文件、寄存器堆)、ALU(算术逻辑单元)、EXT(扩展单元)、IM(指令存储器)、DM(数据存储器)、乘除单元、CP0 等。

13. 为了支持 CP0, Figure1 为供你参考的数据通路架构图。

- a) 该图更多的是让你对所要设计的多周期处理器的数据通路有所认识, 无法保证该设计是 100% 正确和合理的。
- b) 在这个参考设计中需要增加部件和调整部件功能, 包括乘除部件、CP0 和多路选择器。
- c) 如果你做了比较大的调整, 请注意务必不要与要求错误!未找到引用源。矛盾。

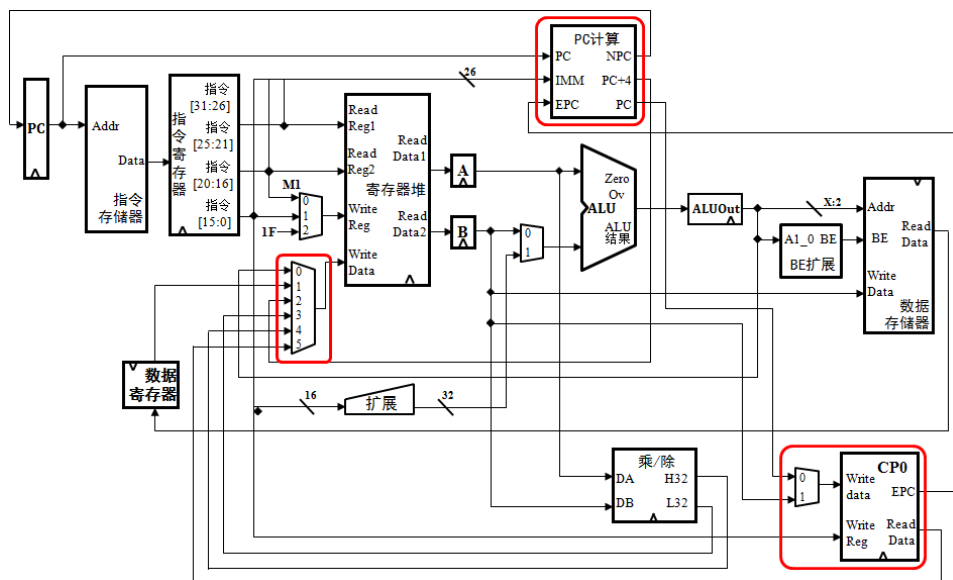


Figure1 多周期数据通路(供参考)

14. MIPS 处理器如何与 Bridge 模块及设备, 请参考《L15.1-支持 IO.pptx》。

15. 多周期数据通路应至少包括 PC、NPC、IM、DM 这些。其中:

- a) IM: 容量为 8KB(32bit×2048 字)。
- b) DM: 容量为 12KB(32bit×3072 字)。

批注 [gxp4]: 【修改】存储器容量

16. 微系统的地址图必须严格按照下图设计。其中重要的地址范围和入口请见下

表。



图 1 MIPS 地址图

	地址或地址范围	备注
数据存储器	0x0000_0000 至 0x0000_2FFF	
指令存储器	0x0000_3000 至 0x0000_4FFF	
PC 初始值	0x0000_3000	
Exception Handler 入口地址	0x0000_4180	
定时器寄存器地址	0x0000_7F00 至 0x0000_7F0B	定时器 3 个寄存器

- a) 需要注意的是，本 project 与《See MIPS Run Linux》和 ppt 中给出的 MIPS 的地址范围是不同的，而与 MARS 是相同的。这主要是为了能够让你能更好的验证设计。
- b) Exception handler 的代码属于指令存储器。注意 handler 在指令存储器中的地址位置以及正确编写 modelsim 仿真时指令的初始化文件。

17. 定时器的 ISR 请接入 MIPS 处理器的 HWInt[2]，即最低中断。

五、 文件、层次及模块实例化命名要求

18. 目前正在开发新的测试服务器。当开发完毕后，你的设计文件将统一提交测试服务器进行自动化测试并据此给出测试成绩。为此，请严格遵循本节要求。
19. 本 project 的顶层文件命名为：mini_machine.v。
20. 模块层次命名如下：

第 1 层：	module mini_machine(...) ;
--------	----------------------------

顶层模块名规范	
第2层: mini_machine 模块对 CPU、桥、定时器的实例化规范	<pre>xxx U_MIPS (...) ; xxx U_BRIDGE (...) ; xxx U_TIMER (...) ;</pre>
第3层: mips 处理器对 PC、RF、IM、DM 的实例化规范	<pre>YYY U_PC (...) ; YYY U_RF (...) ; YYY U_IM (...) ; YYY U_DM (...) ;</pre>
第4层: PC、IM、DM 模块内部变量命名规范	<pre>PC 模块: reg [31:2] PC ; RF 模块: reg [31:0] gpr[31:1] ; IM 模块: reg [31:0] im[1023:0] ; DM 模块: reg [31:0] dm[2047:0] ;</pre>

六、 模块定义【WORD】

21. 本部分要求与 Project6 完全相同，请参考之。

七、 测设要求

22. 请开发一个主程序以及定时器的 exception handler。整个系统完成如下功能：

- ①定时器在主程序中被初始化为模式 0；②定时器倒计时至 0 产生中断；
- ③handler 重置初值寄存器从而再次启动定时器的计数器。②及③被无穷重复。

23. 主程序在初始化时将定时器初始化为模式 0，设定初值寄存器的初值为 1000。

- a) 注意：主程序可能需要涉及对 CP0.SR 的编程。

24. 编写一个定时器中断服务程序(ISR)。ISR 完成如下功能：①保存现场②加载初值寄存器③回复现场④异常返回。

25. 虽然 MARS 只能验证异常而不能验证中断，但你可以在 MARS 中先验证你的中断/异常处理框架是否正确。方法是：你让某条指令出错，然后看 MARS 能否进入你写的 exception handler。至于你如何处理这个错误，则是次要问题。提示：你可以考虑产生一个除零错误。

- a) 按照图 2 添加你的 exception handler。

批注 [gxp5]: 之所以定义比较小的值，目的在于快速仿真。

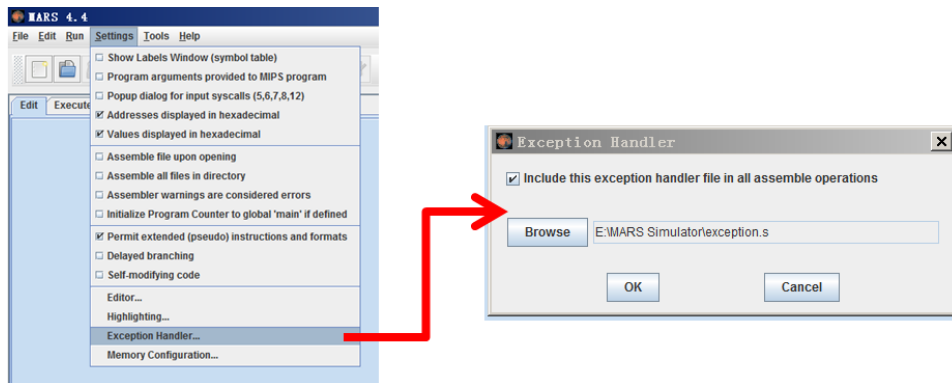


图 2 在 MARS 中验证 Exception handler 机制

b) 主程序可能需要编程 CP0.SR。

八、 问答【WORD】

26. 第 23 项定义的初值寄存器初值不能低于多少？为什么？

27. 有待补充。

九、 成绩及实验测试要求

28. 实验成绩包括但不限于如下内容：设计的正确性、实验报告等。

29. 实验测试时，你需要重点解读乘除法实现

a) 解读不仅应准确，而且应力求简洁。

30. 考查时，教师将可能用专用 testbench 和 code.txt 检测代码执行情况。

十、 其他要求

31. 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。

32. 自动测试：自动测试服务器正在开发中。具体开放时间另行通知。

33. 时间要求：开学第 1 周提交，具体时间另行通知。

34. 现场考查：开学第 1 周通知，具体时间另行通知。

35. 本实验要求文档中凡是出现了【WORD】字样，就意味着该条目需要在实验报告中清晰表达。

36. 实验报告请按照《计算机组成原理实验报告撰写规则.doc》要求排版。

十一、 开发与调试技巧

37. 请参考 Project6 的内容。

38. 如何让中断服务程序的地址恰好在某个特定地址呢？

- a) 假设中断服务程序文件为 `exception.s`，内容如下。在这个样例中，只包括了 1 个 `eret` 指令。

```
1 .ktext 0x00004180
2      eret
```

- b) `Line1` 定义了异常程序的起始地址为 `0x00004180`。注意这个地址在图 1 中可以看到（从上面属第 7 项）。