P7 提交要求

整体要求

- MIPS 处理器须为流水线设计，MIPS 微系统须支持中断和异常。
- 除本文明确的规范和补充声明外，MIPS 微系统设计以《See MIPS Run Linux》（下文简称《SMRL》）作为标准。《SMRL》的标准与 MARS 的行为存在一定差异，在测试时不以 MARS 为准。
- P7 较前几个 Project 为同学预留了更多需自主设计的内容，最终 P7 的实现因人而异。只要满足所给出的设计约束、行为规范和 MIPS 基本设计规范，任何设计都被认为是正确的。
- **此章节主要包含实现细节与评测要求，一些基本概念或定义请结合前面的教程理解。**

顶层模块接口

- MIPS 微系统接口（请顶层模块严格满足该要求）：

```
module mips(
    input clk,                      // 时钟信号
    input reset,                    // 同步复位信号
    input interrupt,                // 外部中断信号
    output [31:0] macroscopic_pc, // 宏观 PC

    output [31:0] i_inst_addr,    // IM 读取地址（取指 PC）
    input  [31:0] i_inst_rdata,   // IM 读取数据

    output [31:0] m_data_addr,    // DM 读写地址
    input  [31:0] m_data_rdata,   // DM 读取数据
    output [31:0] m_data_wdata,   // DM 待写入数据
    output [3 :0] m_data_byteen,  // DM 字节使能信号

    output [31:0] m_int_addr,     // 中断发生器待写入地址
    output [3 :0] m_int_byteen,   // 中断发生器字节使能信号

    output [31:0] m_inst_addr,    // M 级 PC

    output w_grf_we,              // GRF 写使能信号
    output [4 :0] w_grf_addr,     // GRF 待写入寄存器编号
    output [31:0] w_grf_wdata,    // GRF 待写入数据

    output [31:0] w_inst_addr     // W 级 PC
);
```

- 相较于 P6 的顶层模块新增了以下 4 个接口：

- 相较于 P6 的顶层保持不变，个别接口
  - `interrupt`：外部中断信号。由中断发生器产生，每次中断信号会持续到处理器响应该信号。请注意，处理该中断信号的方式应和处理 Timer 产生的中断不完全相同，具体见前面的教程。
  - `macroscopic_pc[31:0]`：详细概念见宏观 PC。我们保证评测过程中宏观 PC 仅用于定位指令，作为产生外部中断信号的条件。
  - `m_int_addr[31:0]`：中断发生器待写入地址。当该信号命中中断发生器响应地址，且字节使能信号有效时，视为响应外部中断。
  - `m_int_byteen[3:0]`：中断发生器字节使能信号，当该信号任意一位置位时视为有效。

硬件约束

- 顶层模块中应该至少包含 CPU、Bridge、Timer0、Timer1 四个功能部件。
- 地址空间：见系统桥。
- 主要部件：
  - CPU：在 P6 基础上进行增量开发，增加 CP0 协处理器，支持异常和中断等。
  - CP0：见CP0约束。
  - Bridge：须作为独立的 module，不包括在 CPU 中；访问外设均须通过系统桥。
  - IM：容量为 **16KiB**（**4096** × 32bit）。
  - DM：容量为 **12KiB**（**3072** × 32bit）。
  - Timer：定时器官方源代码已经给出，无需自行设计实现。
  - 中断发生器：
    - 中断信号依据宏观 PC 产生，依据相应的待写入地址和字节使能信号关闭，具体实现参考下发的 tb。
    - 由于其内部并没有真正的存储单元，我们规定读出的数据始终保持 0，且写入时除了响应中断外不会产生其他影响。

CP0 约束

- 协处理器位置：不作明确要求，自行设计。
- 输出要求：写入时无需 display。
- 为了支持异常和中断，必须实现的寄存器包括：**SR、CAUSE、EPC**。
- 寄存器规范：
  - CP0 寄存器的初始值均为 0，未实现位始终保持 0。
  - 当进入中断或异常状态时，均需要将 EXL 置为 1，用以屏蔽中断信号（注意《SMRL》中并没有指定进入中断时 EXL 的值）；当退出中断或异常状态时，也均需要将 EXL 置

为 0，取消屏蔽中断信号。

- Cause 寄存器的 IP 域每周期写入 HWint 对应位的值。
- 当进入中断或异常状态时，需要将受害指令的 PC 写入 EPC。

指令约束

- 处理器应支持如下指令集：

```
nop, add, sub, and, or, slt, sltu, lui
addi, andi, ori
lb, lh, lw, sb, sh, sw
mult, multu, div, divu, mfhi, mflo, mthi, mtlo
beq, bne, jal, jr,
mfc0, mtc0, eret, syscall
```

- 在 P6 基础上新增了 `mfc0, mtc0, eret, syscall` 四条新指令。
- `eret` 具有跳转的功能但是没有延迟槽，你的设计应该保证 `eret` 的后续指令不被执行。
- `syscall` 指令行为与 MARS 不同，无需实现特定的输入输出功能，只需直接产生异常并进入内核态。

中断异常约束

- 异常入口：《SMRL》的表 5.1 中定义了 MIPS 的异常入口，但考虑到简化设计以及与 MARS 保持一致，我们只支持 0x4180 一个入口地址，所有异常与中断都将从这里进入。
- 嵌套中断异常：本实验不要求支持中断异常嵌套的情况。
- 优先级：中断优先级高于异常优先级，即当有异常提交至 CP0 寄存器时，若有中断发生，则硬件应先响应中断，并重新执行受害指令及其后续指令；若没有中断发生，则处理异常。
- 精确异常：
  - 除下面的情况外，对所有中断异常的处理都应遵循精确异常的处理规则。
  - 在进入中断或异常状态时，如果受害指令及其后续指令**已经改变**了 MDU 的状态，则无需恢复。假设 CP0 在 M 级，MDU 在 E 级，考虑以下情况：
    - `mult` 在 E 级启动了乘法运算，流水到 M 级时产生了中断，此时无需停止乘法计算，其它乘除法指令同理。
    - `mthi` 在 E 级修改了 HI 寄存器，流水到 M 级时产生了中断，此时无需恢复 HI 寄存器的值，`mtlo` 同理。
    - `mult` 在 E 级，受害指令在 M 级，此时还未改变 MDU 状态，不应开始乘法计算，其它乘除法指令同理。

    - `mthi` 在 E 级，受害指令在 M 级，此时还未改变 MDU 状态，不应修改 HI 寄存器的值，`mtlo` 同理。
- 中断规范：
    - Timer0 输出的中断信号接入 HWInt[0] (最低中断位)，Timer1 输出的中断信号接入 HWInt[1]，来自中断发生器的中断信号接入 HWInt[2]。
    - 规定中断产生时的受害指令为宏观 PC 对应的指令，此时应将宏观 PC 写入 EPC。
- MIPS 微系统需要支持的异常：

ExcCode	助记符	描述
0	Int	中断。
4	AdEL	取数或取指时地址错误。
5	AdES	存数时地址错误。
8	Syscall	系统调用。
10	RI	不认识的（或者非法的）指令码。
12	Ov	自陷形式的整数算术指令（例如 `add`）导致的溢出。

- 补充说明：
    - 分支跳转指令无论跳转与否，延迟槽指令为受害指令时 `BD` 均需要置位。
    - 发生取指异常或 `RI` 异常后视为 `nop` 直至提交到 CP0。
    - 跳转到不对齐的地址时，受害指令是 PC 值不正确的指令（即需要向 EPC 写入不对齐的地址）。
    - 对于未知指令的判断仅需考虑 opcode（和 R 型指令的 funct），且仅需判断是否出现在 P7 要求的指令集中，同时保证未知指令的测试用例中 opcode 和 funct 码的组合一定没有在 MARS 的基本指令集中出现。

官方测试说明

- 为便于进行测试，我们允许从 0x417C 直接前进到 0x4180，此种情况下 CPU 行为与 P6 一致，不应有中断响应等其他行为。
- 测试数据规范：
    - 测试时不会出现跳转到未加载指令的位置的情况。

    - `eret` 只会出现在中断处理程序中，后可能紧跟另一条非 `nop` 的指令。
    - 测试程序保证不会写入 Cause，但可能写入 SR 和 EPC。
    - 测试程序只会通过指令 `sb $0, 0x7f20($0)` 访问中断发生器（响应中断），且只会在中断处理程序中访问。
    - 中断处理程序会对寄存器和内存进行读写来验证 CPU 的正确性。
    - 中断处理程序执行过程中保证不出现异常，且不会产生中断。
- 官方 tb 示例：
    - 外设不给予中断时，使用的 tb 为：下载链接。
    - 评测机通过检测同学们的宏观 PC 给予中断信号并对中断进行测试，例如此下载链接中的 tb 会在处理器的宏观 PC 第一次到达 0x3010 时给予 CPU 一个中断信号。

官方 Mars

- 课程组修改了 Mars，增加了输出运行信息等功能，支持课程 P7 要求的异常和定时器中断，供同学们本地测试，下载链接。