

在线测试相关信息

提交要求

顶层模块

必须在 Verilog HDL 设计中建立这个**顶层模块**，不允许修改模块名称、端口各信号以及变量的名称 / 类型。

```
module mips(  
    input clk,  
    input reset  
);
```

GRF 模块

在 **GRF 模块**中，每个**时钟上升沿**到来时若要**写入数据**（即写使能信号为 1 且非 reset 时）则输出写入的位置及写入的值，格式（请注意空格）为：

```
$display("@%h: %d <= %h", WPC, Waddr, WData);
```

其中 `WPC` 表示相应指令的储存地址，从 0x00003000 开始；`Waddr` 表示输入的 5 位写寄存器的地址；`WData` 表示输入的 32 位写入寄存器的值。

例如，地址在 0x00003000 的指令向 3 号寄存器写入数据 0，则应该输出 `@00003000: $3 <= 00000000`，其中 `00003000` 应该是对应操作的指令的 16 位进制地址，不足 8 位需要补零。

添加输出语句的示例如下：

```
always @(posedge clk)  
    begin  
        // .....  
        if (RegWrite == 1)  
            begin  
                // .....  
                $display("@%h: %d <= %h", WPC, Waddr, WData);  
            end  
        end  
    end
```

DM 模块

在 **DM 模块**中，每个**时钟上升沿**到来时**若要写入数据**（即写使能信号为 1 且非 reset 时）则输出写入的位置及写入的值，格式（请注意空格）为：

```
$display("@%h: *%h <= %h", pc, addr, din);
```

其中 `pc` 是该操作对应的指令的地址，和 GRF 模块要求一致，`addr` 表示将要存入数据的 32 位地址，`din` 表示输入的 32 位写入 DM 的值。

例如，若地址在 0x00003004 的指令要在 0x00001004 地址写入数据 0，则应该输出
`@00003004: *00001004 <= 00000000`。

加输出语句的示例如下：

```
always @(posedge clk)
begin
    // .....
    if (MemWrite == 1)
    begin
        // .....
        $display("@%h: *%h <= %h", pc, addr, din);
    end
end
```

其它

IM 容量为 16KiB ($4096 \times 32\text{bit}$)，DM 容量为 12KiB ($3072 \times 32\text{bit}$)。

复位后，PC 指向 0x00003000，此处为第一条指令的地址，GRF 和 DM 中的所有数据清零。注意与 **MARS 中的设置**保持一致。在复位期间对存储单元进行操作请不要输出任何信息，以免影响评测结果。

除加减法外的指令均以指令集规定的行为为准，可参看 [MIPS-C 指令集](#)，加减法按无符号处理（不考虑溢出）。若对本次实验有任何疑问请及时在讨论区提出。

思考题

在相应的部件中，复位信号的设计都是**同步复位**，这与 P3 中的设计要求不同。请对比**同步复位**与**异步复位**这两种方式的 reset 信号与 clk 信号优先级的关系。

思考题

C 语言是一种弱类型程序设计语言。C 语言中不对计算结果溢出进行处理，这意味着 C 语言要求程序员必须很清楚计算结果是否会导致溢出。因此，如果仅仅支持 C 语言，MIPS 指令的所有计算指令均可以忽略溢出。请说明为什么在忽略溢出的前提下，`addi` 与 `addiu` 是等价的，`add` 与 `addu` 是等价的。提示：阅读《[MIPS32® Architecture For Programmers Volume II: The MIPS32® Instruction Set](#)》中相关指令的 Operation 部分（详见文档 page 34、page 35）。