

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

其中 \$time 表示当前的时间，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 100 时刻向 3 号寄存器写入数据 0，则应该输出 100@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。

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

其中 \$time 表示当前的时间，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 100 时刻要在 0x00001004 地址写入数据 0，则应该输出 100@00003004: \*00001004 <= 00000000。

## 其它

IM 容量为 16KB (32bit × 4096 字)，DM 容量为 12KB (32bit × 3072 字)。

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](#)（该指令集描述针对单周期，注意单周期和流水线有些许指令行为描述不同，但最终结果是一致的），加减法按无符号处理（不考虑溢出）。

无需考虑当写通用寄存器和内存同时发生时，指令输出的先后顺序问题。

若对本次实验有任何疑问请及时在讨论区提出。

## 额外说明

由于本次设计是流水线 CPU，对各种指令间存在的**数据冲突**和**控制冲突**，应尽量使用**转发**来解决，不能使用转发解决的情况下才选择使用暂停解决。因此，CPU 设计时解决冲突的方案不同，跑完测试程序所需要的周期数也不同。越好的设计方案，所需要的周期数相对而言也就越少。

对于我们的实验而言，我们不要求大家设计的都非常完美，但至少常见的能使用转发解决的冲突都要用转发来解决。课上测试时我们会根据测试程序的不同设置不同的**周期范围**，你的 CPU 要在这个范围内跑完整个测试程序，否则即使结果正确也无法通过评测。课上测试时我们会**重点**提问你的设计中对于冲突的解决方案，成绩的很大一部分也取决于你的 CPU 所跑的**周期数**与**问答情况**。

对于指令的行为，Mars 和指令集并不是完全相同的，建议大家添加指令前先简单的试试新增指令在 Mars 中是如何运行的，这个并不麻烦。引用一位同学的话——怠惰会导致惨剧。