

P4 思考题

1. 根据你的理解，在下面给出的 DM 的输入示例中，地址信号 `addr` 位数为什么是 `[11:2]` 而不是 `[9:0]`？这个 `addr` 信号又是从哪里来的？

| 文件 | 模块接口定义 |
|------|---|
| dm.v | <pre>dm(clk, reset, MemWrite, addr, din, dout); input clk; //clock input reset; //reset input MemWrite; //memory write enable input [11:2] addr; //memory's address for write input [31:0] din; //write data output [31:0] dout; //read data</pre> |

地址在 mips 里面都是 4 的倍数，而在 Verilog 中是 32 位存入的，所以要是 `[11:2]`，`addr` 信号是从上一个模块 ALU 计算得出的

2. 在相应的部件中，**reset** 的优先级比其他控制信号（不包括 `clk` 信号）都要高，且相应的设计都是**同步复位**。清零信号 `reset` 是针对哪些部件进行清零复位操作？这些部件为什么需要清零？

IFU, DM, GRF。

这些部件中，如果清零，那么寄存器堆，数据存储器，PC 都要变成开始的部分，所以都要清零。

3. 列举出用 Verilog 语言设计控制器的几种编码方式（至少三种），并给出代码示例。

格雷码，独热码，二进制编码。

格雷码：00 01 11 10

独热码：000 010 100

二进制编码：00 01 10 11

4. 根据你所列举的编码方式，说明他们的优缺点。

格雷码便于化简，独热码每次只有一个数为 1，比较时只需要 1 个 bit，二进制编码比较符合顺序逻辑，使用的触发器较少。

5. 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。

Add 和 addi 都是考虑溢出的，而 addu 和 addiu 都是不考虑溢出的，所以如果不考虑溢出，他们是等价的。

6. 根据自己的设计说明单周期处理器的优缺点。

优点：没有冲突，每次只运行一条指令，所以如果当前指令写的没有问题，就不会运行错误。

缺点：运行慢。

7. 简要说明 jal、jr 和堆栈的关系。

这两个指令可以实现递归功能，如果实现递归的话，就要用到栈，这个时候，就要把我们回来要用的值都存进栈中，包括\$ra。