

1. 实验报告和日志有模版吗？

有模板，请于乐学获取。

2. 数据存储器 and 指令存储器可以分开吗？

这两部分可以采用哈弗结构分开存放数据和指令，也可以采用冯·诺依曼结构放在一起，根据自己的思路进行设计，没有硬性要求

3. 仿真激励文件 sim 中下列模块中，调用的是 light_show，所以 .l_rst_n (rst) 语句要写.l_rst_n，而不是.l_rst，老师给的代码中写的是后者，会提示找不到.l_rst

```
light_show U_light_show
(
    .l_clk (clk),
    .l_rst_n (rst),
    .l_show_num (show_num),
    .O_led (led),
    .O_px (px)
);
```

4. Mips 汇编设计写完了要在单周期 CPU 上跑吗？如果不要，MiPS 汇编设计具体要求是什么？需要写到什么难度程度。

就是为自己的 CPU 写测试程序，只有最低要求，覆盖所有指令，没有难度上限。建议看下乐学上的计分要点，根据自己的实际展开设计。

CPU测试汇编代码和机器码

1. 提交文件格式为压缩文件 (.rar或者.zip)
2. 文件命名为学号，如1120178888.rar/zip
3. 截至时间：9.13日24:00
4. 需要点击提交按钮进行提交

MIPS汇编设计实验报告

1. 提交文件格式为word (.doc或者.docx)
2. 实验报告命名为学号，如1120178888.doc/docx
3. 截至时间：9.13日24:00
4. 需要点击提交按钮进行提交

MIPS汇编设计源代码

1. 提交文件格式为压缩文件 (.rar或者.zip)
2. 文件命名为学号，如1120178888.rar/zip
3. 截至时间：9.13日24:00
4. 需要点击提交按钮进行提交

这两个可以合二为一的，源代码部分可以一样。前一个框是为了 CPU 实验的完整性，方便复测。后一个框是要求实验报告，分析程序代码，功能描述等。

5. 汇编设计是自己设计一个？还是单纯的为测试 CPU 服务的？

可以理解为就是为自己的计算机，采用汇编语言写软件。

6. 请问有将 mips 汇编转为小端 16 进制数的便捷方法吗？最好是能直接被\$readmemh 读取的那种？

https://github.com/bit-mips/bitmips_experiments_doc/blob/master/others/cross_compiler.md#%E5%85%B3%E4%BA%8E%E6%8C%87%E4%BB%A4%E7%94%9F%E6%88%90 参考一下

7. 我目前需要实现的是一个单周期的 CPU，假设我把它分为取指周期，取数周期，执行周期。其中 ALU，指令访存，数据访存都是使用时序电路设计的。接受到信号就会做相应的事情。那如果我现在需要实现单周期的 CPU，只提供的一个时钟信号的话，是不是我自己需要一个分时，例如第 1，4，7，10 个时钟上升沿时给指令访存一个信号，第 2，5，8，11

个时钟上升沿时给数据访存一个信号，第 3，6，9，12 个时钟上升沿时给 ALU 一个信号这种。还是说有其他更好的方法实现？

这就类似多周期 CPU 了，在一个时钟上升沿的时候，所有模块都完成了当前指令的工作。

可是例如取数这种，你必须保证取指周期完成了才能取数，我怎么确定取指周期完成了呢？

并没有这两个周期。取指和取数是一个周期。如果用 reg 保存数据，给出地址就能返回数据。

例如 `add $1,$1,1` 这种指令。如果我有两条一模一样的指令，我怎么保证他会执行两次呢。

两条指令挨个执行就好。

例如 `add $1,$1,1` 这种，接收到时钟上升沿信号同步去执行读寄存器\$1,ALU 组合逻辑实现\$1+1,写寄存器\$1。如果这三件事都是通过同一个时钟上升沿信号执行的话能够保证执行正确吗？。

代码写对了就能。

每个上升沿，读到一条指令并解析它，分别执行 alu 等操作，这些都是组合逻辑。然后如果要写 regfile 就在下个上升沿写。

怎么确定 ALU 的状态已经传过去了呢？

实际上时钟沿上升的时候，各个电路内部的信号电压不是同时同幅度变化的。比如 ALU 依赖取指，那么 ALU 的逻辑只有取值的信号到了才会被驱动。写（入完成） reg 确实是下一个时钟的事情

8. 为什么仿真的时候只出现了时钟和复位信号？

需要仔细检查仿真文件，是不是其他信号没有引出来？时钟和复位能出来的话，你可以观察下其他想观测的信号和这两信号的设置区别。

仿真的时候，信号默认只显示仿真文件里的信号，如果要显示其他文件的信号，如果要显示其他文件的信号，要手动加进去。

9. IP 核封装有什么用处呢？如果不封装的话，不能够使用模块设计了么？

表现上类似于 npm 包，只暴露接口。Block design 的话，还是需要 IP 核。

可以直接调用，但是建议使用 IP 核。个性化指令都需要自己分析，为了统一难度，我们只补充了要求，不考虑溢出。

10. 提醒一下各位 Mars 编译会自动优化你的代码，所以可能和你想的不一样。Mars 是一个开源的工具，感兴趣的话可以自己修改，像对不上的情况的话，可以手工转一下机器码。

11. 我们要做小端的是吗？请问 Mars 生成的，操作码在最左边的那种是小端还是大端？

Mips 大端，小端都支持。印象中 Mars 给的是小端。实验没有要求一定大端小端，自己决定，报告里说明。

12. 汇编的个人实验，所涉及的指令可以是没有用 CPU 实现的吗？

可以用伪代码。你可以看下 mars 支持的伪代码，mars 可以将它们转化成机器指令。汇编个人实验的目的之一是为 cpu 提供测试程序。这个 cpu 可以是你个人设计的 cpu，也可以是团队设计的 cpu。在报告里说明就好。

13. 结果写回目标寄存器是在本时钟周期里写吗？还是要等到下一个时钟周期上升沿？

下个周期上升沿，你要是在本周期修改，比如 `addiu $10, $10, 1` 这条指令，因为是组合逻辑\$10 的值就会一直改变

14.，下一个时钟周期到来的时候，如果新的指令对应的寄存器堆写使能是 0 的话，会不会影响上一个周期产生的数据的写入呢？

不会的，触发器的基本特性。

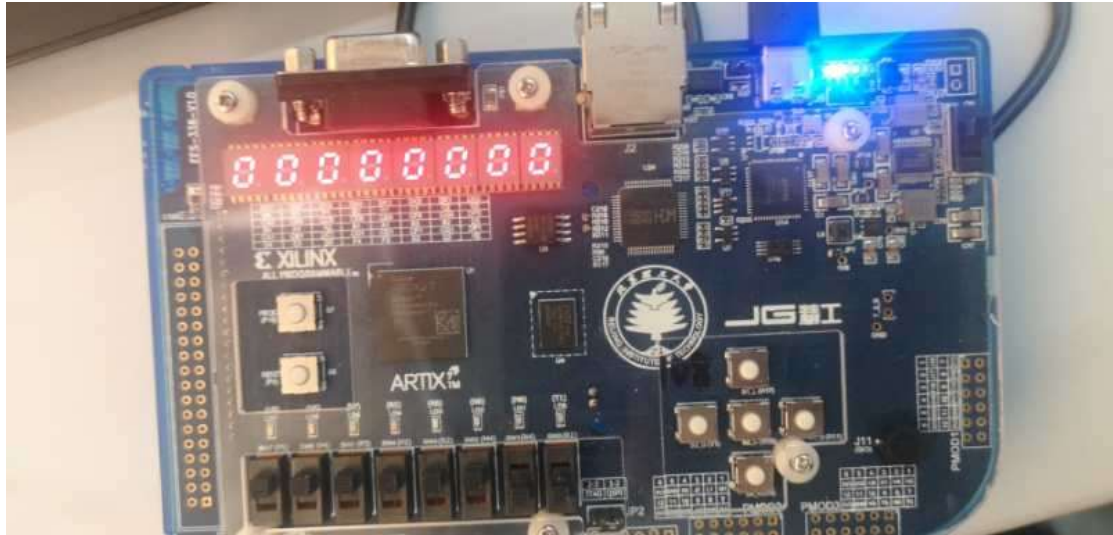
15. Testbench 有信号，顶部 cpu 模块没信号是怎么回事？

信号拉回 0，重新跑。

16. 怎么给一个 word 变量赋值成 32'h8000 0000 呢？

```
lui $t0 0x8000
```

17. 请问停在 0 不动是什么情况



如果是用的 soc，那就是第一个测试点之前就发生了错误，建议先仿真。仿真结束时，查看 confreg 模块 digital_num_v 变量，这个变量的最终值应该是 0x13000013，如果是，仿真应该就没问题。

18. VGA 怎么区分总线给的数据和要输出到屏幕的数据？

一般根据地址区分。

19. 下板之后，要怎么在指令内存里面写入编好的程序？

用 reg 写的 im，就只能通过 rst 在复位时候挨个初始化，你用 bram 就 coe 文件初始化。

20. 请问 coe 文件只读了第一个字可能是什么情况？

估计需要检查一下 PC，然后 PC 正常的话就顺着捋，直到找到不对劲的地方。（最后是通过去掉代码中的 rst 解决的）。

21. 请问有没有适用于咱们开发板的 VGA 的颜色和 RGB 向量的对比图，为啥我输入全 1 进去出来的是蓝色？

猜测是用 12 位显示颜色有问题，最终未解决。



22. 在 implementation 步骤总是 error？

随便加个输出试试，据说 vivado 会把没输出的模块优化成啥都没有。

23. Q: 命令行无法运行 xvlog

A: 把 xvlog 加到环境变量 Path 中

24. 指令 XORI (rt)=(rs) xor imm

rs 的值 xor 16 位立即数(0 扩展)，结果放到 rt

25. always@(*)和 assign 的区别

有时候两者会综合出相同的电路, `assign` 用于 `net` 类型, 能够生成一个逻辑电路, `always` 本质是设定在什么样的驱动下这个块才会重新执行, `*` 表示所有 `RHS`(Right Hand Side) 的信号都会作为驱动 (就像组合逻辑一样), 所以写法正确的情况下就能达到跟 `assign` 一个效果。因此从实践的角度来说, 如果能用 `assign` 请首先使用它。如果逻辑过于复杂, 那么小心使用 `always @(*)`。此外, 尽管 `always` 语句可以同时使用阻塞赋值 `=` 和非阻塞赋值 `<=`, 但是最佳实践是总是使用非阻塞复制。

26. 测试寄存器组的时候如何初始化寄存器组?

在 `initial` 里循环初始化

27. 仿真正常, RTL 分析正常, 综合的时候东西都被优化?

这是综合器内置的优化算法, 一般不需要管他, 只要仿真结果正确, 没问题就好。如果不想被优化, 在设置信号、变量和模块时可以指定属性参数, 提示综合器不被优化。设置属性的方法有很多种, 如设置 `don't touch` 属性等。

28. 接口部分是不是跟 `cpu` 和汇编分开的?

可以分开, 但鼓励集成。

29. 时钟频率设置为 25Hz, 但是板子接到显示器后显示超频?

检查 `VGA` 的分辨率设置是否符合显示协议要求。

30. `Data_sram_wen` 1 位和 4 位的区别?

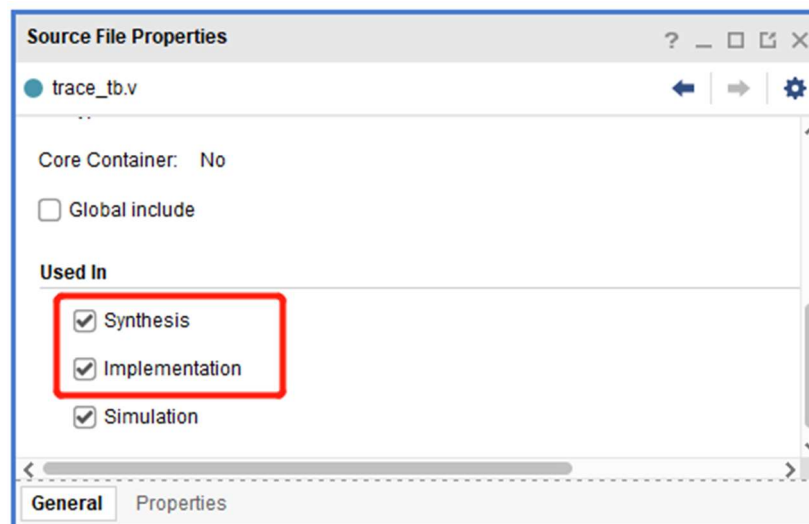
1 位是存储字整体读写, 4 位可以用每一位控制一个字节, 实现按字节读取, 32 位为 4 个字节。

31. 数组大小限制不能开显存, 有什么方式可以使用大内存?

用其他存储(`bram`), 降低分辨率再优化一下。

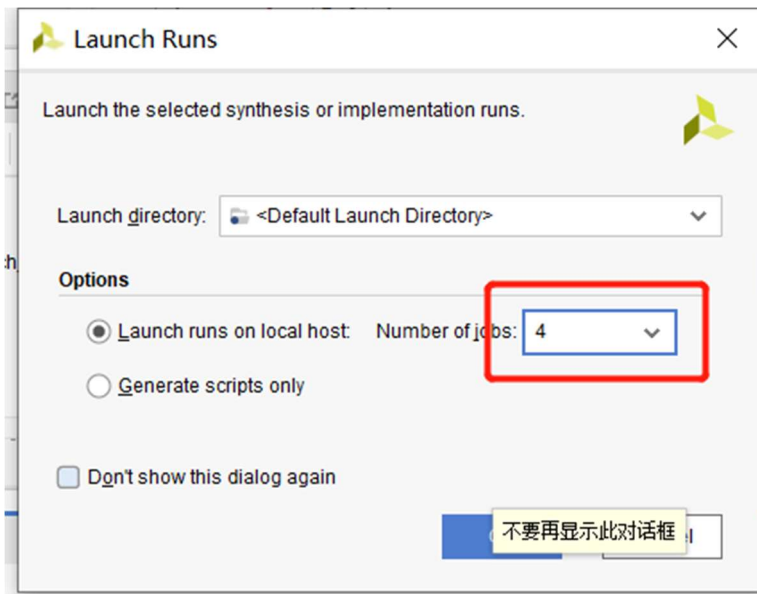
32. [Synth 8-3380] loop condition does not converge after 2000 iterations [trace_tb.v:182]

去掉红框中的选项



33. 综合时间太长解决方案?

开 4 线程



减少 reg 使用，reg 使用太多也会跑很长时间

34. bram 教程

https://github.com/bit-mips/bitmips_experiments_doc/blob/master/others/ip.md