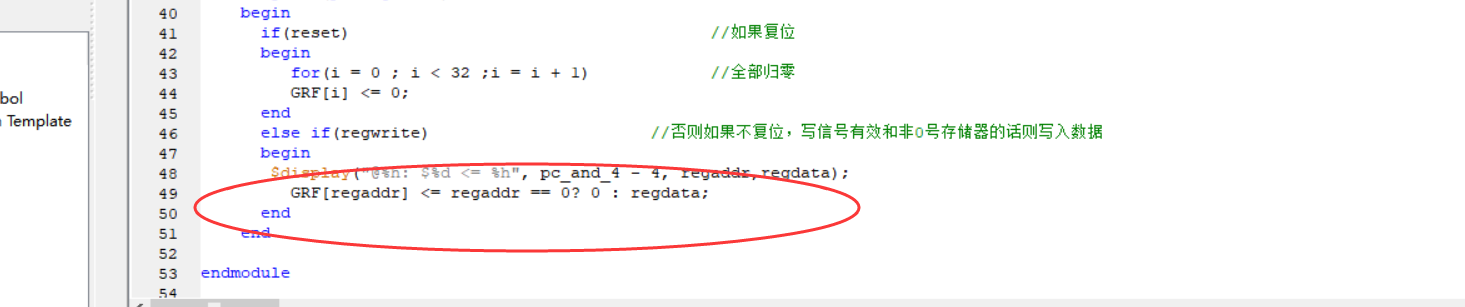


和我原电路不同的有

1. ALUstr改成了四位
2. IFU的IM变成了32bit×1024个字，所以PC取的位不同
3. 用wire在内部连线构造数据通路
4. 注意输出的MEMADDR是取2-11位
5. 三目运算符实现MUX
6. 注意l和1的区别，verilog里面特别严重，以及b和h都很容易搞错
7. 测试码要删除v2.0 raw
8. 注意要看波形debug
9. 纯组合逻辑，别搞reg输出
10. 以后begin别换行
11. 

这样写比原来好，因为是regwrite有效就display

1. 写MUX时从右到左用assign写
2. 注意有符号运算要另加$signed
3. 注意不要写错字母
4. beq bgez bgtz blez bltz bne仅仅修改CONTROLLER ALU IFU和MIPS
5. 在MIPS上千万别忘记增加wire
6. 与Beq共用npc\_sel有危险，比如

执行ifbgtz时，npc\_sel有效，但是结果为0而不是>0时，ifzero 和 npc\_sel会有效而终止程序!!!

1. 妙啊，加一个sign判断，就可以把所有b指令包括进去。sign zero eq\_or\_uneql 三个信号包括6个b指令。
2. 把rt还是instr传进controller
3. npc\_sel改成了判断其他+6条b指令的信号

000 else

001 beq

010 bne

011 bgtz

100 bgez

101 bltz

110 blez

1. 

不对就加括号

22、顶层别忘记加连线



连线出错

1. sw和sh和sb分组了
2. lw lh lhu lb lbu分组了
3. 判断32位只有一个1，把所有加起来就完事
4. 拼接成64位，然后左移就完事，或者手动生成constant
5. 运算符优先级问题，记得加括号
6. 还得防止课下指令op和func重复的问题
7. 记得交code

对于P4：

1. 主要参考远古的教程中“构造测试数据”一章，下面是我个人的一些经验
2. I型指令重点测试立即数为正负零的三种情况，无论何种扩展方式
3. B型跳转测试上跳、下跳、不跳三种情况
4. 测试0寄存器；所有寄存器的读写都要至少覆盖一次测试；JAL重点测试31号寄存器
5. 测试L、S指令时要注意它也是I型，所以offset也是正负零三种情况
6. 推荐的测试顺序：ORI、LUI、ADDU、SUBU、SW、LW、BEQ、J、JR、JAL

对于P5：

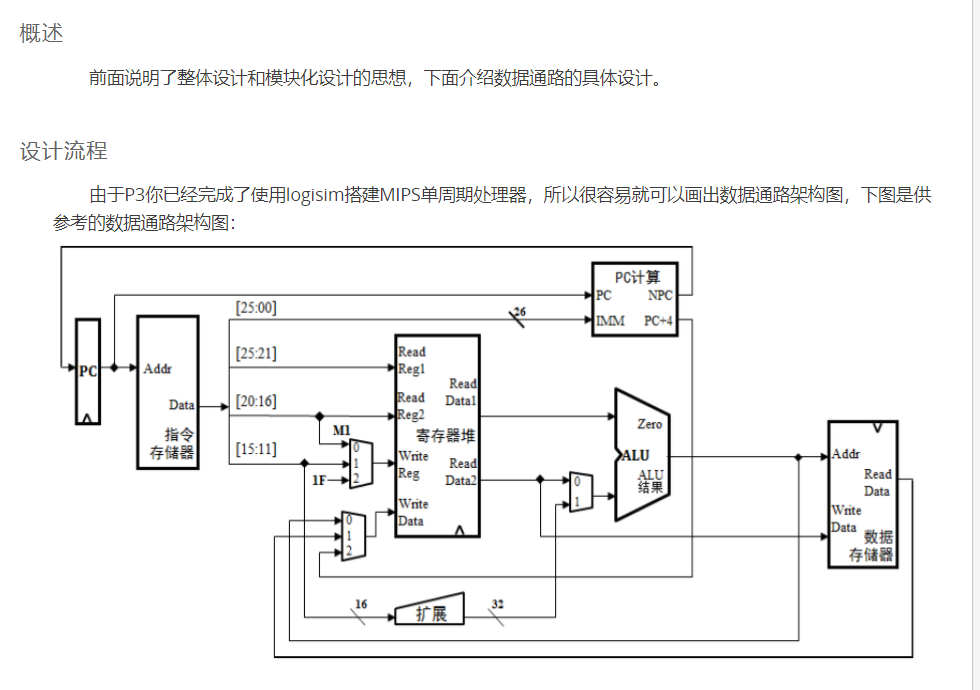
1. 在P4测试集基础上构建数据通路测试，简单说就是P4测试集中每条指令后面加5个nop情况，然后跳转后面加一个nop补延迟槽
2. 转发测试，构建方法参考教程，推荐善用macro宏套个模板
3. 把P4测试集补上延迟槽，然后拿来黑箱测试

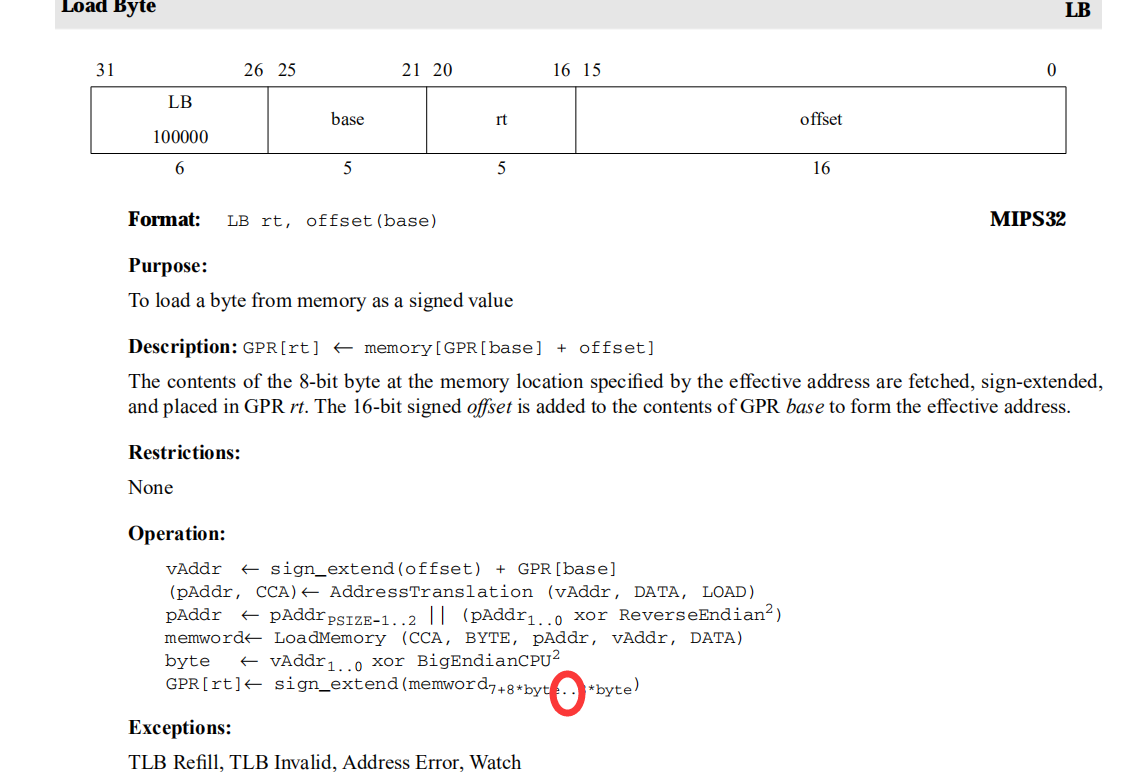
在上面的基础上，造大量随机数据弥补人脑疏漏

上面这些东西意义有限。****静态检查永远是最有效的****，推荐小黄鸭测试法和小黄鸭调试法（

1. 每一次更改尤其注意不要改变原来的地方
2. 利用mark\_all方便更改
3. offset\_10是用来判断sw sh sb lw lb lh等操作的

**一个Verilog HDL文件中可以定义多个module，因此建议所有mux（包括不同位数、不同端口数的所有MUX）都建模在一个mux.v中。**



1. 终于看懂了英文指令集

这个不是省略号，是逗号

1. verilog里面的拼接运算符一定要每个都加{}

{{16{1’b0}},{aluoprand\_b[15:0]}};

{{16{1’b0}},aluoprand\_b[15:0]};

不同！！！