   为了降低实验难度，乘/除法运算的实现可以使用 VerilogHDL 的内置运算符，而不需要从门级开始建模乘法或除法的硬件算法。

        乘除法运算延迟。我们假定乘/除部件的执行乘法的时间为****5**** 个 cycle(包含写入内部的 ****HI**** 和 ****LO**** 寄存器)，执行除法的时间为 ****10**** 个 cycle。你在乘/除部件内部必须模拟这个延迟，即通过 ****Busy**** 输出标志来反映这个延迟。图2 给出了乘法的计算延迟。同时我们通过****只能有效 1 个 cycle****的****Start****信号来启动乘除法运算。

mult和div的波形样例如上图所示(右键点击查看大图)

       乘除法部件中内置了HI和LO两个寄存器，这两个寄存器同时也是与外界沟通的窗口。另外乘除部件有天然的运算功能，因此其其余的端口设计类似于GPR和ALU，此处就不再赘述了。

       乘除模块行为约定如下：

* 自 Start 信号有效后的第 1 个 clock 上升沿开始，乘除部件开始执行运算，同时 Busy 置位为 1。
* 在运算结果保存到 HI 和 LO 后，Busy 位清除为 0。
* 当 Busy 为 1 时，mfhi、mflo、mthi、mtlo、mult、multu、div、divu 均被阻塞，即被阻塞在 IF/ID。
* 数据写入 HI 或 LO，均只需 1 个 cycle。

SIGNALEXCEPTION：信号异常

ALUOutputSel增加1位选择乘除的输出

LOAD家族在W级寄存器后进行拓展

STORE家族在DM前进行拓展

B家族通过拼接去标记比较的结果，从而确定CMPOut

乘除家族要在E级控制器加控制信号，判断输出

2018.12.15

在SLL这里卡住，发现如果把位移当做是ALUB的话，但是要移位的也是ALUB。草

~~对于SLL直接在ALU乱搞，在ALU里面加入一个位移信号。~~

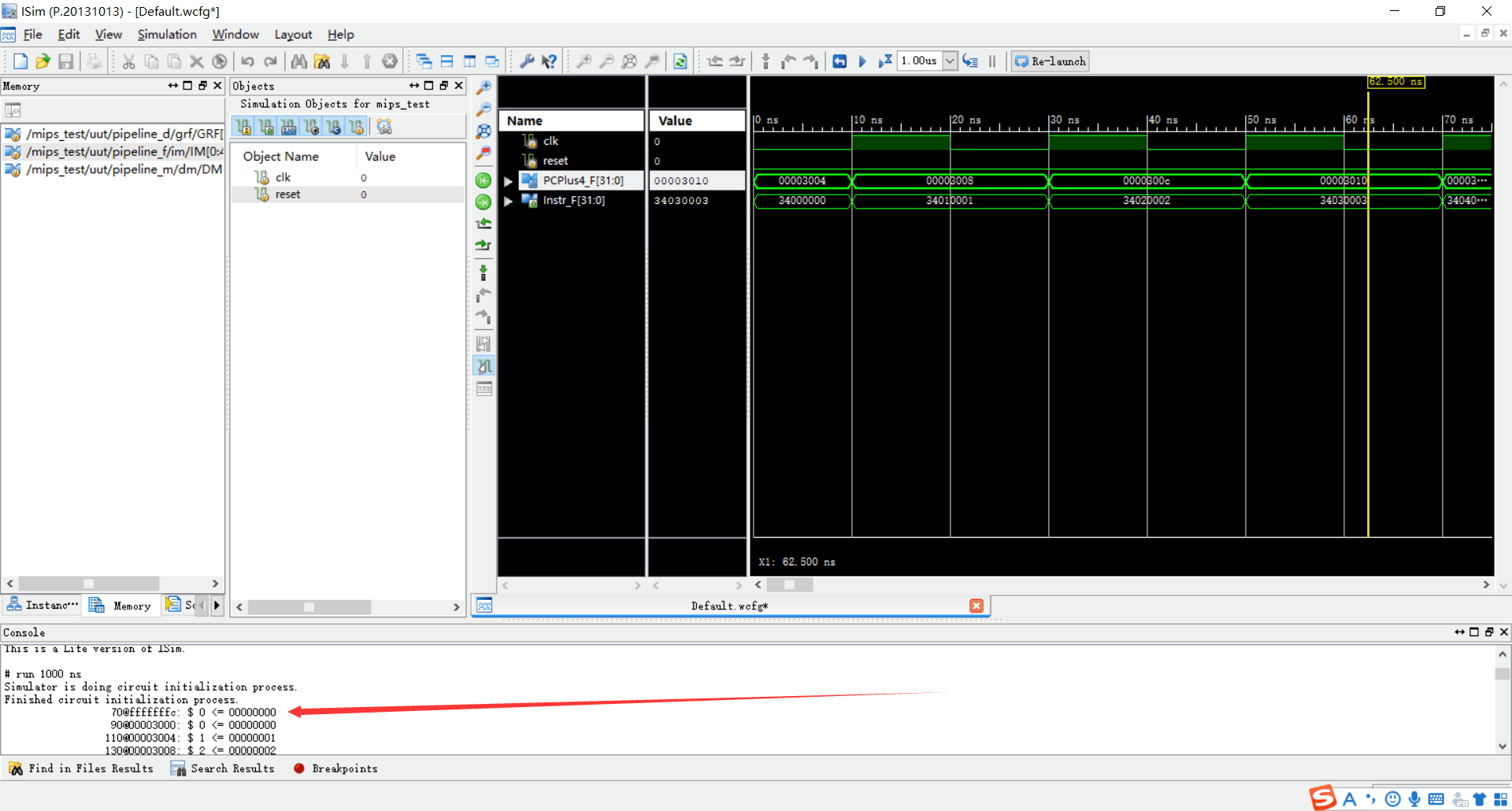
算了，还是增加一个ALUASEL乱搞吧

2018.12.16

注意要在IM那里扣掉PC的3000.

今天搞定SLL SLLV SRL SRLV SRA SRAV

注意到SLLV RD RT RS，所以MUXALUA是MFRSEOUT 和EXTOut



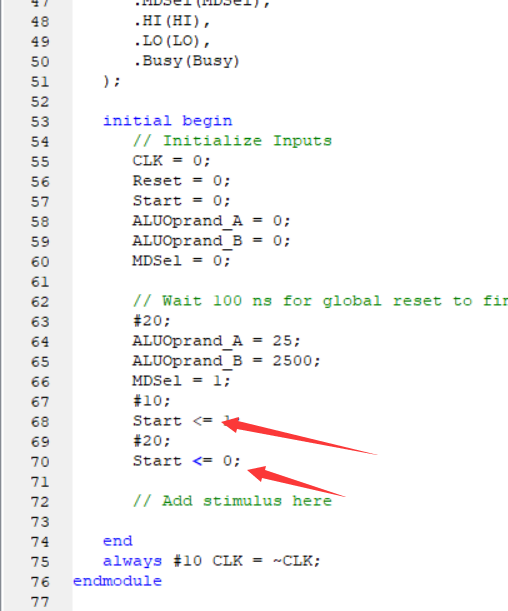
这个是一开始初始化之后对nop和SLL之间冲突的结果

改一下，写0就不要输出。

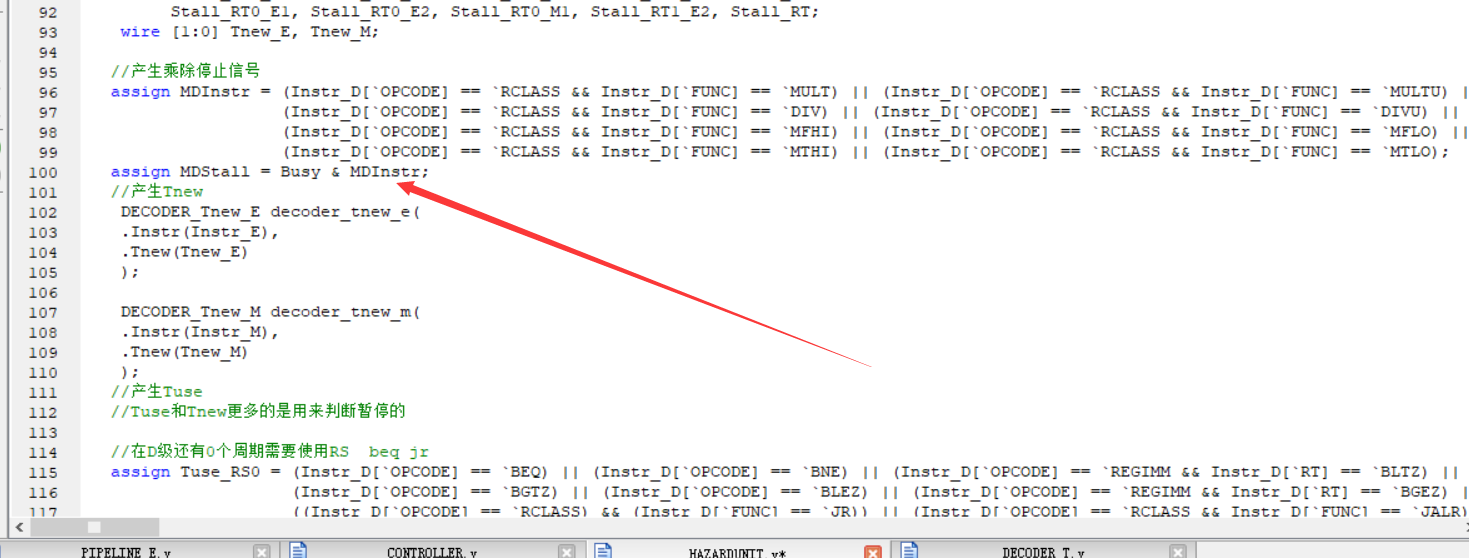
定义wire时千万别忘记加位宽

sb的字节的地址会有奇数，所以输出地址时得处理后两位

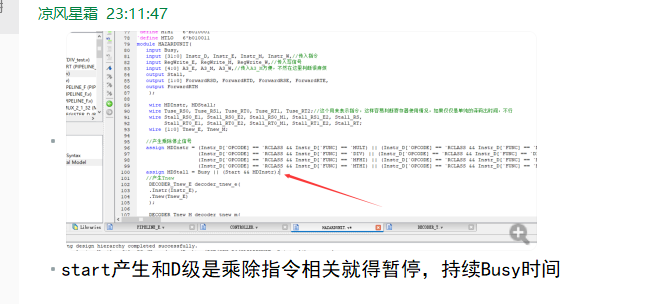
sb是存到那个字的某个位置，而其他位置不变，而不是全部赋0



在testbench里面Start要非阻塞赋值。。。。才能调出来那个波形



这里只能用单个&？？(wronG)



今天添加

SLL SLLV SRL SRLV SRA SRAV

还改了一下对0号寄存器的写入操作不输出

之前版本没有在ALU加AND XOR等操作

补上了ALU的缺漏，和加了SLT SLTI SLTU SLTIU

添加JALR

修复了SB输出地址的BUG

AND修复了SB SH存哪几个字节的BUG，主观臆测过于严重

2018.12.16 20：10

发现blez可能有BUG，发现bug了，看来不能用拼接去搞，得判断分开两个信号判断，当然可以当做是拼接之后当成一个信号传输，避免增加过多的传输线路。

2018.12.16 20：22

发现BLTZ有BUG，主要是两个判断信号的问题，应该是~CMPZEROOUT && ~CMPZEROOUT而不是或

2018.12.16 20:30

发现jalr有问题，人为地强行让写寄存器变成了31，修复了JALR的BUG，注意别主观臆测

2018.12.16 20：36

P6弱测过了，没加乘除指令

2018.12.16 20：48

发现我的算术右移符号错了，应该是>>>三个

P6.7准备搞乘除指令

BUSY传进HAZARD里面和STALL并

2018.12.17 22:42

加了乘除指令

暂停有BUG，好像没有能阻塞，导致CLK上升后

D级指令被传输了

1. 第一点，由于IM和DM的扩容，其输入地址的位数都增加了两位，与之对应的是PC的地址在用时要减去3000，PS 2018.12.20 DM的写入地址的位数未改！！！
2. 输出的MEMADDR应该是2-13位