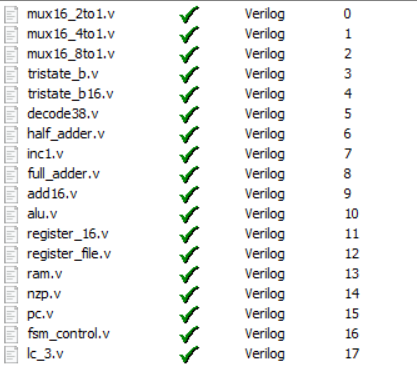
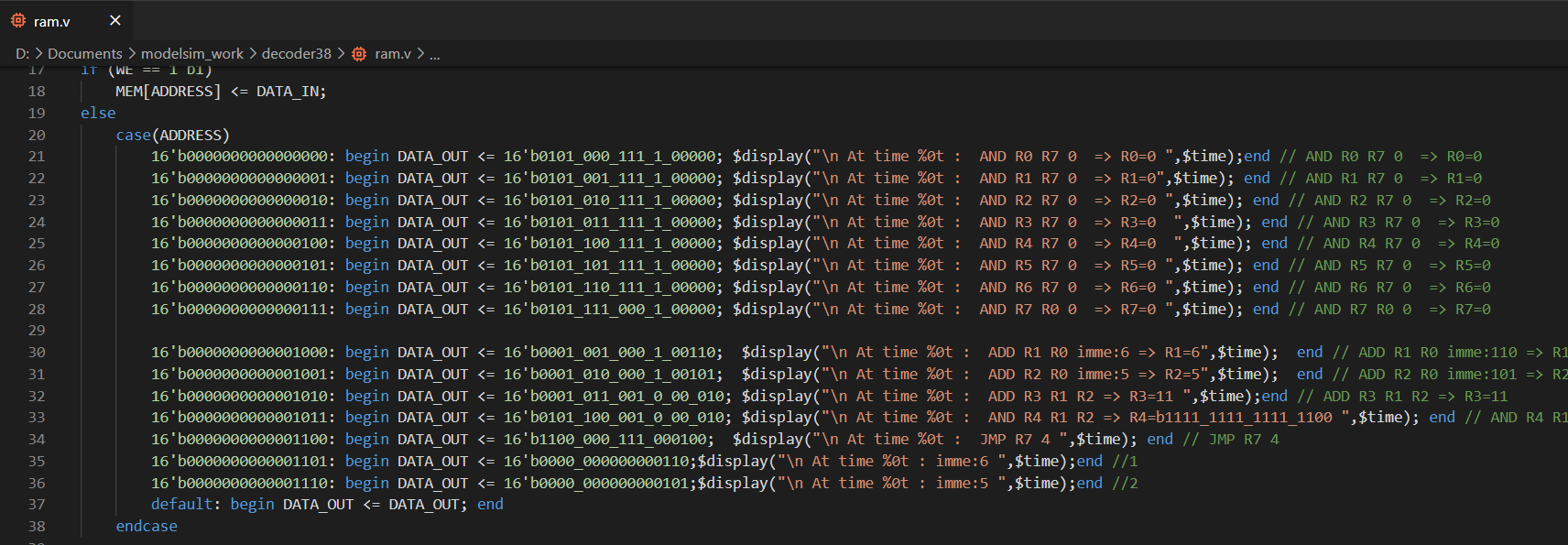
## LC3 Verilog readme

1、编译顺序（compile order）：

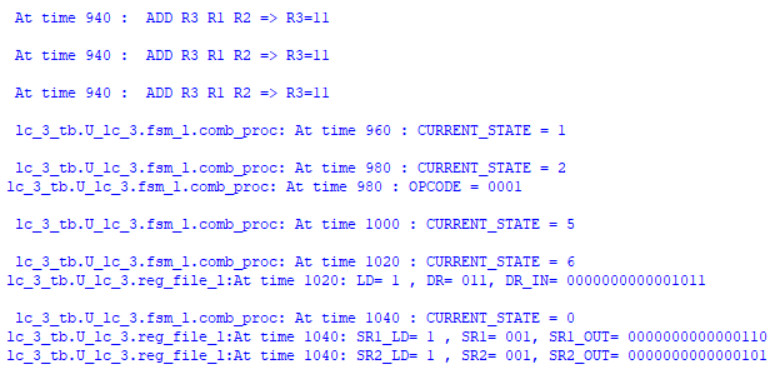


2、测试设置：

CPU仿真的testbench中，只有CLK和RESET。而CPU执行的指令应该提前放入到内存中（RAM）。这里使用了一个比较简单的方法去仿真CPU，但其实应该使用读取文本的方式对memory进行初始化处理，如 $readmemb 语句。



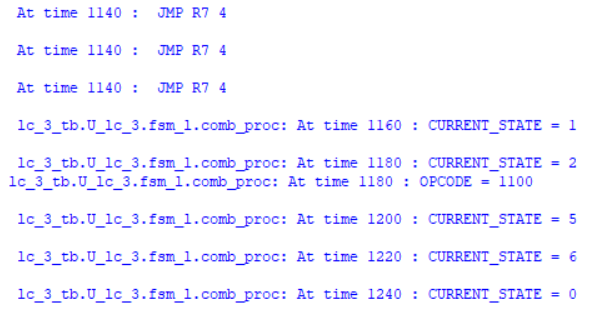
3、仿真结果



代码中，已经对一些关键的信息进行了$display输出。可以看到此时要执行的是“ADD R3 R1 R2”的指令，即R1+R2的值赋予R3，而最终的结果是R3=11。在CURRENT\_STATE=6（保存）的时候，reg\_file中，DR=’b011 (‘d3), DR\_IN=’b1011 (’d11) 。这个结果与所需结果一致。

4、需要改进

另外，本代码的JMP等PC地址跳转指令仍需要改进。



通过仿真可以看见，最后CPU一直在循环JMP指令，其原因是PC模块的连线并没有连接正确。打开“lc\_3.v”，寻找到:

    // PC and GatePC

    pc pc\_l(.CLK(CLK), .RESET(RESET),.LD(LDPC), .PCSEL(PC\_SEL), .OFFSET(PC\_OFFSET), .DIRECT(PC\_DIRECT), .PC\_OUT(PC\_OUT));

通过搜索可以发现，PC\_OFFSET 和 PC\_DIRECT 的连续并没完成。此部分在学习掌握LC-3架构后是很容易改进的，有兴趣的同学可以提前尝试一下。

完善的代码我们将在后面的实验中给出。