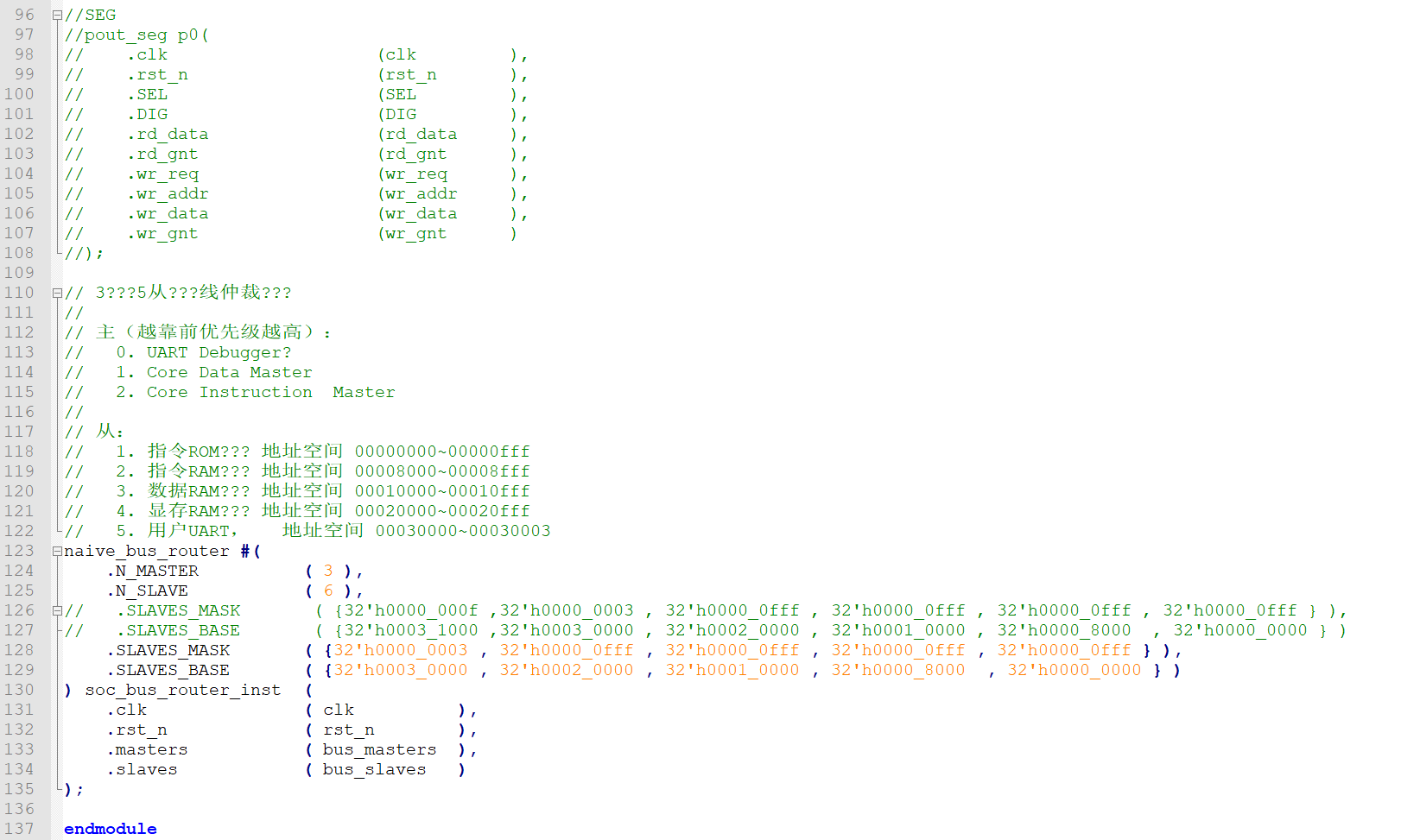
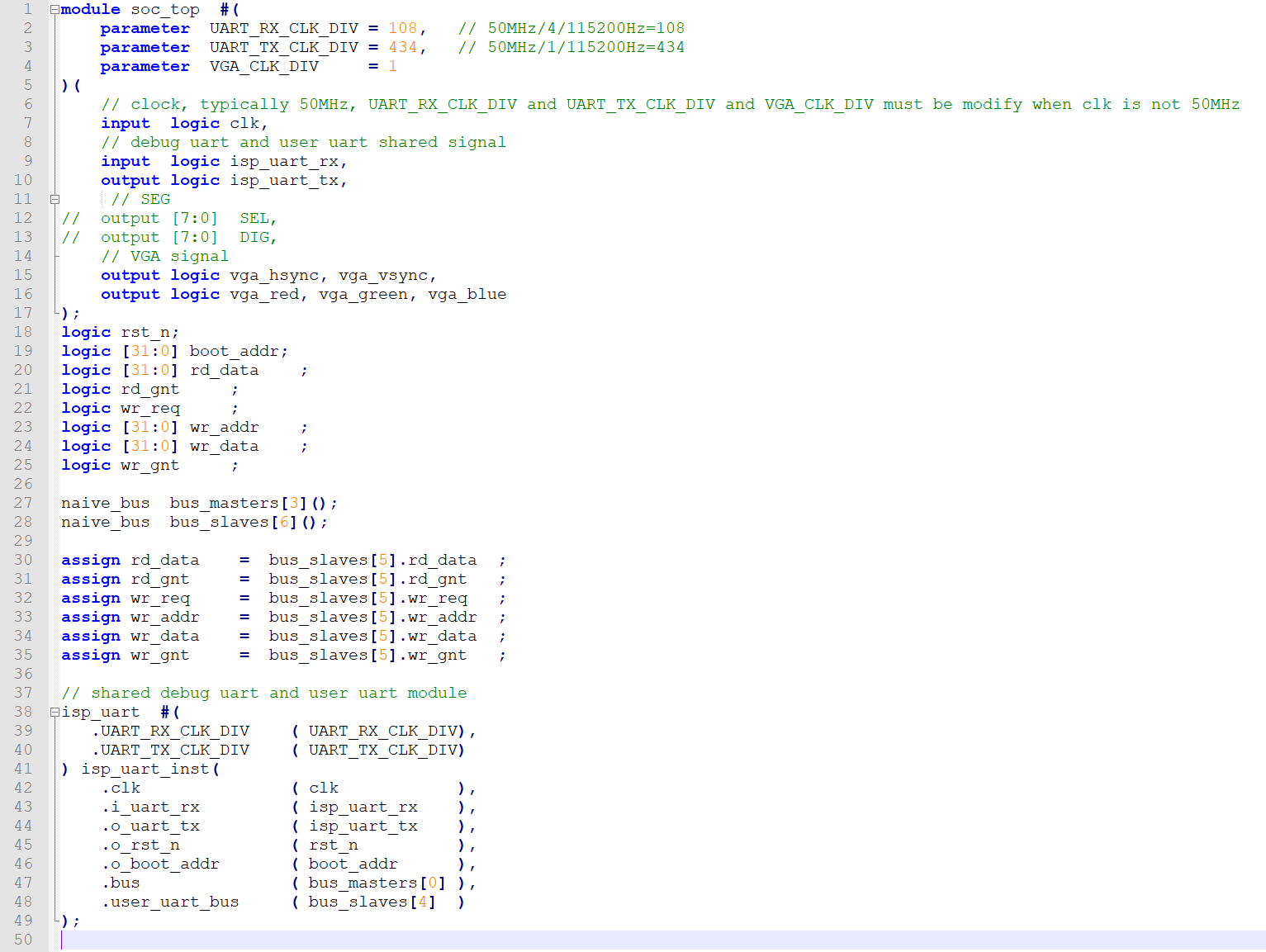
**简单32位RISC CPU设计**



为SEG分配空间，空间地址从32h’0003\_1000~32h’0003\_100f

Slave总线上的模块数，因为我们新加了一个SEG模块，因此要增加1

从设备增加一个，这里定义的总线也要相应的增加一个

把按键上的总线信号线连接到CPU总线上

Seg模块需要的总线上的信号

定义输出信号，其中SEL为位选，DIG为段选信号

对SEG模块进行例化，将slave总线连接到模块上，使模块能从总线上获取数据

#汇编代码

.org 0x0

.global \_start

\_start:

or t3, zero,zero

seg:

# 第一步：令t0寄存器=0x00031000，即seg外设的地址

or t0, zero,zero # t0 清零

lui t0, 0x00031 # t0 寄存器的高20bit=00031

# 第二步：s3递增，赋值给t0地址对应的空间

sb t3, (t0) # t1写入t0地址

addi t3, t3 , 0x1

# 第三步：延时，通过大空循环的方式

lui t2, 0x00c00 # t2 = 0x00c00000

big\_loop:

addi t2, t2, -1 # t2 = t2-1

bne t2, zero, big\_loop # if t2!=0, jmp to big\_loop

jal zero, seg # 大循环结束，跳到print\_hello，重复打印