1.在v9-cpu中如何实现时钟中断的:

将trap设为FTIMER，iena设为0，然后调用'goto interrupt'触发中断。

2.v9-cpu指令，关键变量描述有误或不全的情况：

ssp：系统栈指针

usp：用户栈指针

cycle：循环计数

xcycle：4倍的循环计数

timer：中断计数器

timeout：中断计数器阈值

3.在v9-cpu中的跳转相关操作是如何实现的:

JMP/JMPI指令，在xpc上直接加上操作数，再跳转；

JSR/JSRA指令，先在栈中保存当前的pc值，再跳转；

BRANCH指令，条件判断结果为真跳转，反之继续执行。

4.在v9-cpu中如何设计相应指令，可有效实现函数调用与返回类似于MIPS或X86，在调用函数时保存旧的栈底指针地址，将参数压入栈，修改SP指针，指向调用的函数地址。函数返回时，弹出参数，修改SP指针，栈底指到旧的栈底地址。

5.emhello/os0/os1等程序被加载到内存的哪个位置,其堆栈是如何设置的：

程序被加载到内存的起始位置。堆栈从底向上，往高地址增长。

6. 在v9-cpu中如何完成一次内存地址的读写的：

先查看TLB(读的话看tr，写的话看tw)，看地址是否存在于TLB中。如果存在，则直接使用。如果不存在，则调用rlook函数(读)或wlook函数(写)将相应的地址读入TLB中，然后，先访问地址的高10位读出page directory entry，其对应着一个页表page table。再用地址的中间十位访问page table，读出page table entry，从这个entry才推算出物理地址。

7.在v9-cpu中如何实现分页机制：

在TLB中，设置了4个1MB大小页转换表（page translation buffer array） kernel read page translation table kernel write page translation table user read page translation table user write page translation table 有两个指针tr/tw, tw指向内核态或用户态的read/write　page translation table． tr/tw[page number]=phy page number //页帧号 还有一个tpage buffer array, 保存了所有tr/tw中的虚页号，这些虚页号是tr/tw数组中的index tpage[tpages++] = v //v是page number