

一. 两种工作状态是 ARM 和 Thumb 状态。

二. 7 种运行模式。中断、快中断、复位中断、软中断异常、预取指令中止异常、数据中止异常、未定义指令异常。

三. (1) 并不是每条 ARM 指令都是单周期，特定指令的周期数可变。

(2) 内嵌的桶形移位器产生了更为复杂的指令，扩展了指令的功能，因此改善了内核的性能。

(3) 支持 16 位的 Thumb 指令集，提高了代码密度。

(4) 支持条件执行：每条指令都可以设置一个执行条件，只有条件满足时才执行。

(5) 增强指令：一些功能强大的数字信号处理指令被加入到 ARM 指令集中。

四. 取指：指令从存储器中取出，放入指令流水线；

译码：指令译码；

执行：把一个操作数移位，产生 ALU 的结果。如果指令是 Load 或 Store，在 ALU 中计算存储器的地址；

缓存/数据：如果需要，则访问数据存储器；否则，ALU 的结果只是简单地缓冲一个时钟周期，以便使所有指令具有同样的流水线流程；

回写：将指令产生的结果写回到寄存器堆，包括任何从寄存器读

出的数据。

五. (1)在适当的 LR 中保存断点的地址。

(2)把当前程序状态寄存器 (CPSR) 中的内容保存到模式私有寄存器 SPSR 中;

(3)将寄存器 CPSR 中的 MODE 域设置为中断 (异常) 应进入的运行模式;

(4)对 CPSR 的 I 位和 F 位进行相应的设置, 以防止再次响应同一个中断请求。

(5)强制 PC 从相关的异常向量处取指, 即到中断向量表中获取中断向量, 转向用户所编写的中断 (异常) 服务程序。

六. (1) 专门配置了较多的私有寄存器, 从而可使中断服务程序有足够的寄存器来使用, 不必与被中断服务程序使用同一组寄存器, 这样就免去了因寄存器冲突而必需的保护及恢复现场工作。

(2) ARM 把 FIQ 的中断向量放在了中断 (异常) 向量表末尾 0X0000001C 处, 因此它后面没有其它中断向量, 允许用户将中断服务程序直接放在这里。

七. 1.B、D 2.A、B、C、D 3.B、C、D

八. × × √ √ × × × √ √ ×

九. 1.A 2.C 3.B 4.C 5.B 6.C 7.C 8.B 9.D 10.D 11.D