

# 嵌入式系统 随堂测试

1、将高速缓存分为指令缓存（I Cache）和数据缓存（D Cache）的体系结构（ B ）。

A、普林斯顿结构

B、哈佛结构

C、RISC

D、CISC

2、Cortex-A8处理器采用（ B ）：

A、ARM v8-A架构

B、ARM v7-A架构

C、ARM v9-A架构

D、ARM v4-A架构。

3、存储一个32位数0x32f8a7c到4000H~4003H四个字节单元中，若以大端模式存储，则4000H存储单元内容为（ D）

A、 0x32 B、 0x8a C、 0x7c D、 0x03

4、指令“LDMIA R0!, {R1, R2, R3, R4}”的寻址方式为：（ C ）

A、立即数寻址 B、寄存器间接寻址

C、多寄存器寻址 D、堆栈寻址

5、某ARM指令完成的功能是"如果相等，则进行带进位的加法"，该指令是( D )。

A、adcne r1,r2,r3

B、addeq r1,r2,r3

C、andeq r1,r2,r3

D、adceq r1,r2,r3

6、以下哪一项不属于ARM异常中断（ C ）。

A、复位

B、IRQ

C、条件转移

D、数据终止

7、 对于**ARM7**三级流水线，当一条指令被译码时，上一条指令正被（ **D** ）

A、取指。

B、译码。

C、PC值递增。

D、执行

8. 执行软中断指令后，**ARM**处理器处于（ **B** ）模式。

A. USER

B. SVC

C. SYSTEM

D. ARM

9. 寄存器R14除可做通用寄存器外，还可以做（ B ）

A、程序计数器

B、链接寄存器

C、栈指针寄存器

D、基址寄存器

10. 不属于ARM数据处理指令中第2操作数寻址方式的是（ B ）。

A、立即数寻址    B、变址寻址

C、寄存器寻址    D、寄存器移位寻址

11. MLA R2,R0,R1,R3的运行结果是（ D ）。

- A.  $R2 = R0 * R1 * R3$
- B.  $R2 = R0 + R1 + R3$
- C.  $R2 = R0 + R1 * R3$
- D.  $R2 = R0 * R1 + R3$

12. 对寄存器R1的内容除以4的正确指令是： D

- A. LSR R1, #2
- B. LSL R1, #2
- C. MOV R1, R1, LSL #2
- D. MOV R1, R1, LSR #2。

13. 已知mem32[0xE0010]=0x01,  
mem32[0xE0014]=0x02

LDR R1, =#0x000E0010

LDR R0, [R1,#4]!

执行上述指令序列后，R0,R1寄存器中的值是( C )。

A. 0x01,0xE0010

B. 0x02,0xE0010

C. 0x02,0xE0014

D. 0x01,0xE0014

14. R0=0x05, 执行ORR R0,R0,#3的运算后，R0结果是( D )

A. 0x03

B. 0x05

C. 0x06

D. 0x07



15. `.fill repeat {, size}{, value}`伪操作的含义是（ A ）

- A. 分配一段内存单元，用size长度value填充repeat次。
- B. 分配一段字节内存单元，用value填充size个字节。
- C. 分配一段字节内存单元，并用0填充内存。
- D. 分配一段字节内存单元，用value将内存单元初始化。

16. 多寄存器存取指令中，`LDMFD sp!,{r4-r7,pc}`等于下述哪条指令（ B ）。

- A. `LDMDA sp!,{r4-r7,pc}`
- B. `LDMIA sp!,{r4-r7,pc}`
- C. `LDMDB sp!,{r4-r7,pc}`
- D. `LDMIB sp!,{r4-r7,pc}`

17.将ARM处理器R0中的二个字节的数据，存入由R1+4指示的内存区域,且地址自动更新,则使用的ARM指令是(B) .

A. STR R0, [R1,#4]

B. STRH R0, [R1, #4]!

C. STRH R0, [R1,#4]

D. STRB R0,[R1, #4] !

18.写一条 ARM 指令，将R1所表示的内存单元中的字数据装载到R0，然后将R2中的字数据保存到R1所表示的内存单元中。

```
swp r0,r2,[r1]
```

19.编制程序，求两个数组 DATA1 和 DATA2 对应的数据之和，并把和数存入新数组 SUM 中，计算一直进行到两数之和为零时结束，并把新数组的长度存于 R0 中。

```
.data
data1: .word 2,5,0,3,-4,5,0,10,9      ;数组 data1
data2: .word 3,5,4,-2,0,8,3,-10,5     ;数组 data2
sum:    .word 0,0,0,0,0,0,0,0,0       ;数组 sum

.text
.global _start
_start: ldr      r1,=data1
        _____(1)____ldr r2,=data2_____
        ldr      r3, =sum
        mov      r0, #0
loop:   ldr      r4, [r1],#04
        ldr      r5, [r2],#04
        adds     r4, r4, r5
        add      r0, r0, #1
        _____(2)____str r4,[r3],#04_____
        bne loop
.end
```