# 嵌入式系统

NO. DATE.

## 第一章 嵌入式计算

Q1-1 需求是从客户那里收集来的对系统的非形式描述，客户不是嵌入式系统设计人员，因此其描述通常不使用专业术语表达。而规格说明要精确反应客户的需求并且作为设计时必须遵循的要求，用系统设计者的专业术语描述。

Q1-2 规格说明不讲系统如何做，而是讲系统做什么；体系结构描述了如何实现那些功能。

## 第二章 指令系统

Q2-1 低序：字最低位存放在最低位字节
高序：字最高位存放在最高位字节

Q2-2 冯诺依曼结构：数据和指令存放在同一存储器。
哈佛结构：数据和指令分别存放在各自独立存储器。

Q2-3 a. 16个通用寄存器 R0-R15
b. CRSR是当前程序状态寄存器
c. Z表示零指令执行完后结果为0，若全0则置1
d. PC(程序计数器)保存在R15.

Q2-4 a. 1000
b. 1010
c. 0010

Q2-5 a. EQ：等于零 b. NE：不等于零 c. MI负数
d. VS：溢出 e. GE：有符号大于或等于 f. LT：有符号小于

NO. DATE.

Q2-6 BL指令引导处理器转移到子程序处开始执行.
在子程序跳转前,会把下一条指令的地址存储器.
存储到R14(LR)之中.然后将目标地址存储到R15中.

Q2-7 MOV r15 r14

Q2-8. 略.

Q2-9. NEON指令集是一组适用于ARM处理器的单指令多数据(SIMD)指令.用于支持向量和多媒体处理.

Jazelle指令集允许直接执行8位Java字节,因此执行Java程序时不需要使用字节码解释器.

第三章 CPU

Q3-1 内存映射I/O的优点如下:
①省略了I/O操作的复杂逻辑,易实现,耗费低.
②可以利用丰富的内存寻址实现灵活的I/O操作.

Q3-2. 忙等I/O效率低,当I/O操作未完成时,CPU除了反复测试设备什么都不能做,而CPU需要很多操作与I/O事务并行.

Q3-3 假设存储单元ds1处寄存器地址为0x2000,代码如下:

```
#define ds1 0x2000
while(*ds1 == 0)
```

NO. DATE.

Q3-4 假设设备(dev1)中有两个寄存器 ds1和dd1, dev1的地址为0x1000. ds1的偏移量为0，则ds1地址为0x1000. 同理dd1的地址为 0x1004. 代码如下.

```
#define based_addr 0x1000
#define ds1 (based_addr+0)
#define dd1 (based_addr+4)
int data_dd1;
while ((peek(ds1)&0x001)==0)
    data_dd1=peek(dd1);
```

Q3-9. CPU没有并行的操作事务，仅有一项I/O事务.

Q3-10 选择中断优先级。中断优先级允许多处理设备到中断线路上，并且允许CPU在处理重要请求时忽略不重要的中断请求。而中断向量只提供灵活性，使中断设备可以指定为它服务的中断服务程序.

Q3-21 CPU异常的3个典例
0除除数，未定义的Resets指令和非法的内存访问.

Q2-22 陷阱，软件中断，是一种显式产生异常状态的指令，最通用的用法是进入管态.

Q2-24 a.强制未命中：发生在单元第一次被访问时
b.容量未命中：工作容量集大于高速缓存容量
c.冲突未命中：两个地址映射到高速缓存同一单元.

NO. DATE.

Q3.25. $t_{av} = ht_{cache} + (1-h)t_{main}$ $t_{av} = 608ns$

Q.3-26 $65ns = h t_{cache} + (1-h) t_{main}$ 得 $h = 98\%$

Q3-27 LRU 替换策略.

将地址的最后一位作为组的索引

存取001后

| 组 | 块0标记 | 块0数据 | 块1标记 | 块1数据 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | 00 | 1111 | | |

存取010后.

| 组 | 0标. | 0数 | 1标. | 1数 |
|---|---|---|---|---|
| 0 | 01 | 0000 | | |
| 1 | 00 | 1111 | | |

存取100后

| 组 | 0标. | 0数 | 1标. | 1数 |
|---|---|---|---|---|
| 0 | 01 | 0000 | 10 | 1000 |
| 1 | 00 | 1111 | 01 | 0110 |

存取101后

| 组 | 0标. | 0数 | 1标. | 1数. |
|---|---|---|---|---|
| 0 | 01 | 0000 | 10 | 1000 |
| 1 | 10 | 0001 | 01 | 0110 |

存取111后.

| 组 | 0标. | 0数 | 1标 | 1数. |
|---|---|---|---|---|
| 0 | 01 | 0000 | 10 | 1000 |
| 1 | 10 | 0001 | 01 | 0100 |

NO. DATE.

习题3-28

a. 每条指令执行一次，到 Bloop 高速缓存的状态

| 块 | 标记 | 指令 |
|---|---|---|
| 0 | 110 | Bloop |
| 1 | 101 | ADD r0, r0, #1 |

循环执行完，高速缓存的状态

| 块 | 标记 | 指令 |
|---|---|---|
| 0 | 011 | BEG loopend |
| 1 | 010 | CMP r0, r1 |

b. 每条指令执行1次到 Bloop，高速缓存的状态

| 块 | 标记 | 指令 |
|---|---|---|
| 00 | 11 | B loop |
| 01 | 10 | MUL r4, r4, r6 |
| 10 | 10 | ADD r2, r2, r4 |
| 11 | 10 | ADD r0, r0, #1 |

循环执行完，高速缓存的状态

| 块 | 标记 | 指令 |
|---|---|---|
| 00 | 11 | B loop |
| 01 | 10 | CMP r0, r1 |
| 10 | 10 | BEG loopend |
| 11 | 10 | ADD r0, r0, #1 |

c. LRU替换原则

每条指令执行1次到 Bloop，高速缓存的状态

| 组 | 块0标记 | 块0指令 | 块1标记 | 块1指令 |
|---|---|---|---|---|
| 0 | 110 | B loop | 101 | ADD r2,r2,r4 |
| 1 | 100 | MUL r4,r4,r6 | 101 | ADD r0,r0,#1 |

NO. DATE.

Q4-28 $T_{CPU} <= 1/44.1MHz = 2.3\times10^{-5}s$

$200MHz = 5\times10^{-8}s$ 执行指令条数 $n = 2.3\times10^{-5} / 5\times10^{-8} + 100$

$= 360$

Q4-29 如何下次发生的中断优先级低于或等于前次，则正在执行的中断服务子程序继续执行；

若下次发生的中断优先级高于前次，则正在执行的中断服务子程序被打断，中断服务程序优先处理发生在中断之后，然后再处理前次中断。

## 第五章 程序设计与分析

Q5-2 用数组作为缓冲区

```
#define
int CMAX;
int circ[CMAX];
int pos;  int i=0, sum;
int circ_avg(int xnew){
    pos = ((pos == CMAX-1)? 0 :(pos+1));
    circ[pos] = xnew;
    for(i=0; sum=0; i=pos-1; i++)
    {sum=sum+circ[i] }
     aver = sum / pos;
return aver;
}
```

NO. DATE.

☆Q5-6

a. 符号表：

| | |
|---|---|
| P1 | 200 |
| P2 | 228 |

c. 符号表

| | |
|---|---|
| S1 | 200 |
| S2 | 208 |

b. 符号表：

| | |
|---|---|
| P1 | 100 |
| P2 | 108 |
| P3 | 116 |

Q5-7 可以，外部引用是入口点的子集

Q5-8 a可以重入 b,c不可重入

可重入函数：可以被中断的函数。在多任务处理中，可以在函数执行的任何时刻中断它；不可中断重入的函数由于使用了一些系统资源，比如全局变量区，中断向量表，被中断后会出现问题

Q5-9. 使用和改变全局变量数组，不是可重入函数

Q5-12 a. 两次

```
for(i=0; i<5 ; i++)
{
  x[i*2] = a[i*2] * c[i*2];
  x[i*2+1] = a[i*2+1] * c[i*2+1];
}
```

b. 四次

```
for(i=0; i<8 i++){
x[i*4] = a[i*4] * c[i*4];
x[i*4+1] = a[i*4+1] * c[i*4+1];
x[i*4+2] = a[i*4+2] * c[i*4+2];
x[i*4+3] = a[i*4+3] * c[i*4+3];
}
```

NO. DATE.

Q5-13 a. for (i=0; i<N; i++) {
z[i] = a[i] + b[i];
d[i] = a[i] - b[i]; }

Q5-14 不能使用代码移出；可以使用归纳变量消除.
zbinduct = 0;
ainduct = 0;
for(i=0; i<N; i++)
for(j=0; i<M; i++)
*(zptr + zbinduct) = *(aptr + ainduct)*(
bptr + zbinduct);
zbinduct++; }
ainduct++; }

Q5-15 a. 最少4个 b. 最少4个 c. 最少4个 d. 最少6个

Q5-16 a. 将第二句与第三句对换，则最少需要4个寄存器为好.
b, c, d 已为最少，不需改动

Q5-17 该高速缓存共有256行，每个高速缓存行长为4个字，数组每个元素大小为1个字
a. a[0][0] 与 x[0][0] 的低12位地址相同
b. a[0][0] 与 x[0][0] 的低12位地址相比小1个字.
c. a[0][0] 与 x[0][0] 的低12位地址相差大于等于4个字.

NO. DATE.

Q6-6 a. (10+100) /1000 = 0.11

b. (100+75+50) / 400 = 0.56

c. (1+2+20) /10 = 0.55

Q6-9 就绪态表示该进程已行可以执行，等待CPU调度。等待态表示该进程的执行条件不满足，还在等待I/O或其他进程的数据，或触发其运行的定时器尚未期满等。

Q6-12 进程超周期 a. 10 ; b. 20 ; c. 60

Q6-13 进程超周期为200

增加一个P1示例后的CPU利用率 = 4/200+4/200+1/10

+2/40+6/50 = 0.3

远没有达到RMS的CPU利用率上限0.69，故可以增加

Q6-14 进程超周期为100

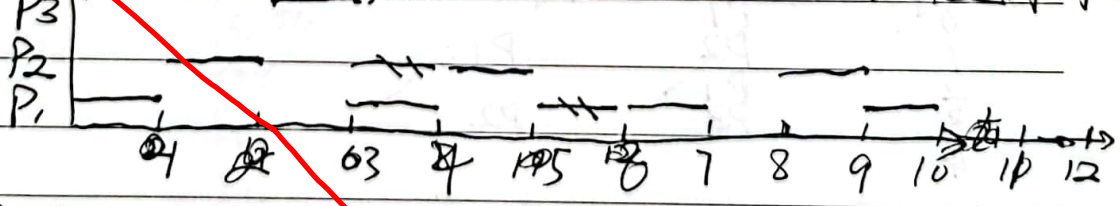
CPU利用率：0.1+0.18+0.1+0.1+x/25

5个任务的RMS利用率的最小上界为 $5(2^{1/5}-1) \approx 74.3\%$

故 $x \leq 6.5$ P5的最大执行时间为6

Q6-17 1. 根应用RMS调度的时序图

根据RMS原则，优选级 P1 > P2 > P3，进程超周期为12.



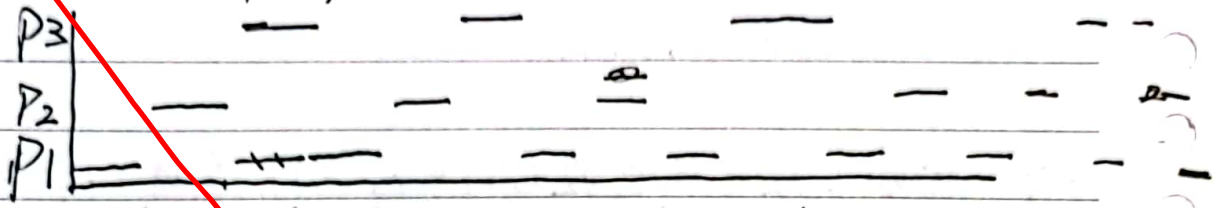

2. EDF调度与RMS调度相同

结束

NO. DATE.

Q6-18 进程超周期为24.

根据RMS原则，优先级 P1>P2>P3




利用EDF调度，若进程距离截止时间一样近，则优先级按P1>P2>P3

具体调度如下：

| 时间 | 运行中的进程 | 截止时限 |
|---|---|---|
| 0 | P1 | |
| 1 | P2 | |
| 2 | P3 | P1 |
| 3 | P1 | P2 |
| 4 | P2 | |
| 5 | P3 | P1 |
| 6 | P1 | |
| 7 | idle | P2,P3 |
| 8 | P2 | P1 |
| 9 | P1 | |
| 10 | P3 | |
| 11 | P3 | P1, P2 |
| 12 | P1 | |
| 13 | P2 | |
| 14 | idle | P1 |
| 15 | P1 | P2, P3 |
| 16 | P2 | |
| 17 | P3 | P1 |
| 18 | P1 | |
| 19 | P3 | P2 |
| 20 | P2 | P1 |
| 21 | P1 | |
| 22 | idle | |
| 23 | idle | P1P2P3 |

进程超周期为30

Q6-19 应用RMS调度，根据RMS原则，优先级 P1>P2>P3

| 时间 | 运行中的进程 | 截止时限 |
|---|---|---|
| 0 | P1 | |
| 1 | P2 | P1 |
| 2 | P1 | |
| 3 | P3 | P2 |
| 4 | | |

NO. DATE.

Q6-22 应用RMS调度，周期为100

根据RMS原则，优先级 P1 > P2 > P3 > P4 > P5.

| 时间 | 运行中的进程 | 时间 | 运行中进程 |
|---|---|---|---|
| 0 | P1 | 60 | P1 |
| 1 | P2 | 61 | P2 |
| 2 | P3 | 62 | P3 |
| 3 | P3 | 63 | P3 |
| 4 | P4 | 64 | P4 |
| 5 | P1 | 65 | P1 |
| 6 | P4 | 66 | P4 |
| 7 | P4 | 70 | P1 |
| 8 | P4 | 71 | P2 |
| 9 | P4 | 75 | P1 |
| 10 | P1 | 80 | P1 |
| 11 | P2 | 81 | P2 |
| 12 | P4 | 82 | P3 |
| 13 | P4 | 83 | P3 |
| 14 | P4 | 85 | P1 |
| 15 | P1 | 90 | P1 |
| 16 | P4 | 91 | P2 |
| 17 | P5 | 95 | P1 |
| 18 | P5 | 100 | P1 |
| 19 | P5 | | |
| 20 | P1 | | |
| 21 | P2 | | |
| 22 | P3 | | |
| 23 | P3 | | |
| 24 | P5 | | |
| 25 | P1 | | |
| 26 | P5 | | |
| 27 | P5 | | |
| 28 | P5 | | |
| 29 | idle | | |
| 30 | P1 | | |
| 31 | P2 | | |
| ⋮ | | | |
| 35 | P1 | | |
| 40 | P1 | | |
| 41 | P2 | | |
| 42 | P3 | | |
| 43 | P3 | | |
| 45 | P1 | | |
| 50 | P1 | | |
| 51 | P2 | | |
| 52 | P4 | | |
| 53 | P4 | | |
| 54 | P4 | | |
| 55 | P1 | | |
| 56 | P4 | | |
| 57 | P4 | | |
| 58 | P4 | | |
| 59 | P4 | | |

NO. DATE.

在用EDF调度.

| 时间 | 运行中的进程 | 截止时限 |
|---|---|---|
| 0 | P1 | |
| 1 | P2 | |
| 2 | P3 | |
| 3 | P3 | |
| 4 | P4 | P1 |
| 5 | P1 | |
| 6 | P4 | |
| 7 | P4 | |
| 8 | P4 | |
| 9 | P4 | P1, P2 |
| 10 | P1 | |
| 11 | P2 | |
| 12 | P4 | |
| 13 | P4 | |
| 14 | P4 | P1 |
| 15 | P1 | |
| 16 | P5 | |
| 17 | P5 | |
| 18 | P5 | |
| 19 | P5 | P1, P2, P3 |
| 20 | P1 | |
| 21 | P2 | |
| 22 | P3 | |
| 23 | P3 | |
| 24 | P5 | P1 |
| 25 | P1 | |
| 26 | P5 | |
| 27 | P5 | |
| 28 | P5 | |
| 29 | idle | P1, P2 |
| 30 | P1 | |
| 31 | P2 | |
| 32 | idle | P1 |
| 33 | P1 | |
| 34 | idle | P1, P2, P3 |
| 35 | P1 | |
| 36 | P2 | |
| 37 | P3 | |
| 38 | P3 | P1 |
| 39 | P5 | |
| 40 | idle | P1 P2 P3 P4 |
| 41 | P1 | |
| 42 | P2 | |
| 43 | P3 | |
| 44 | P3 | |
| 45 | idle | P4 |
| 46 | P1 | |
| 47 | P2 | |
| 48 | P3 | |
| 49 | P3 | |
| 50 | idle | P1 |
| 51 | P4 | |
| 52 | P4 | |
| 53 | P4 | |
| 54 | P4 | |
| 55 | P4 | |
| 56 | P4 | |
| 57 | P4 | |
| 58 | P4 | |
| 59 | P4 | P1, P2, P3, P4 |
| 60 | P1 | |
| 61 | P2 | |
| 62 | P3 | |
| 63 | P3 | |
| 64 | P4 | P1 |
| 65 | P1 | |
| 66 | P4 | |
| 67 | P4 | |
| 68 | P4 | |
| 69 | idle | P1, P2 |
| 70 | P1 | |
| 71 | P2 | |
| 72 | idle | |
| 73 | idle | P1 |
| 74 | P1 | |
| 79 | idle | P1, P2, P3 |
| 80 | P1 | |
| 81 | P2 | |
| 83 | P3 | |
| 84 | P3 | P1 |
| 85 | P1 | |
| 86 | idle | |
| 89 | idle | P1, P2, P3 |
| 90 | P1 | |
| 91 | P2 | |
| 92 | idle | |
| 94 | idle | P1 |
| 95 | P1 | |
| 99 | idle | P1 P2 P3 P4 P5 |
| 100 | P1 | |

EDF与RMS所需的上下文切换次数相同