

全国计算机技术与软件专业技术资格（水平）考试

2007 年下半年 嵌入式系统设计师 下午试卷

（考试时间 14:00~16:30 共 150 分钟）

请按下述要求正确填写答题纸

1. 在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。
2. 在答题纸的指定位置填写准考证号、出生年月日和姓名。
3. 答题纸上除填写上述内容外只能写解答。
4. 本试卷共 4 道题，全部是必答题，满分 75 分。
5. 解答时字迹务必清楚，字迹不清时，将不评分。
6. 仿照下面例题，将解答写在答题纸的对应栏内。

例题

2007 年下半年全国计算机技术与软件专业技术资格（水平）考试日期是 (1) 月 (2) 日。

因为正确的解答是“11 月 3 日”，故在答题纸的对应栏内写上“11”和“3”（参看下表）。

例题	解答栏
(1)	11
(2)	3

试题一（共 20 分）

阅读下面有关某项目自动化配送中心处理机控制系统的叙述，回答问题 1 至问题 3，将答案填入答题纸的对应栏内。

【说明】

某企业大型配送控制中心通常采用自动化方式存取仓库内的货物，配送控制中心对货物的存取由两部分组成：其一，配送控制中心管理和调度多台配送车；其二，配送车根据指令从仓库货架上存取货物。

图 1-1 为某企业大型仓库货物存取示意图。图中配送车上安装有智能控制设备，通过视频接口接收图像数据，实施对货架位置的定位识别，然后将识别信息发送到配送控制中心。配送控制中心向配送车发送控制命令，将配送车上的货物放置到指定的货架位置，或从指定的货架位置上取出货物，装载到该配送车。

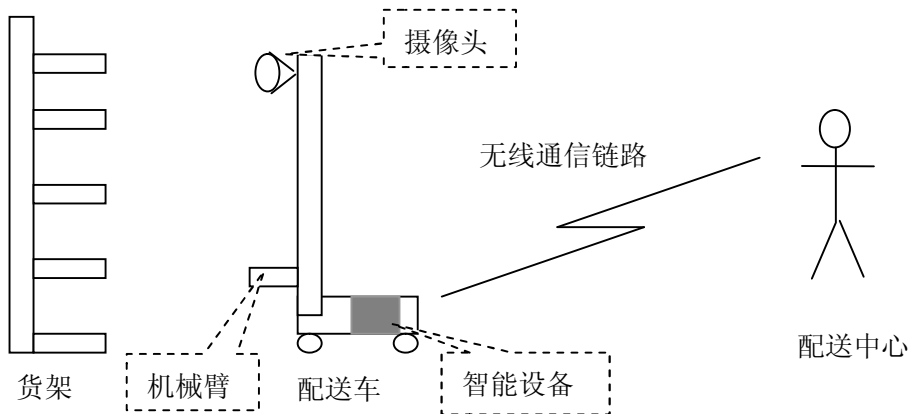


图 1-1 大型仓库货物存取示意图

图 1-2 所示为智能设备控制配送车从货架上取货的工作过程示意图。

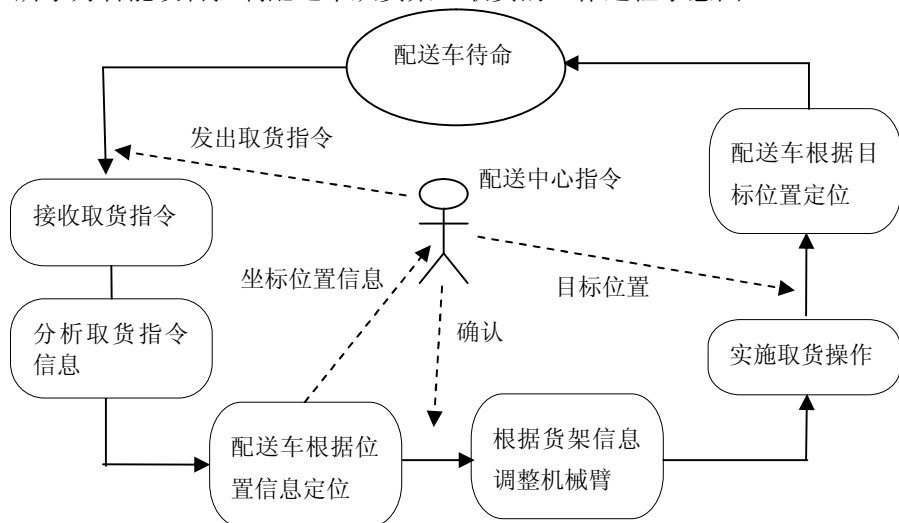


图 1-2 配送车取货的工作过程示意图

【问题 1】(9 分)

假设图 1-1 中的智能设备采用 8051 微处理器，该微处理器的定时器主频为 12MHz。该智能设备中的数据采样周期分别为 10ms、15ms、1s，请给出：

- (1) 设备中定时器应设置的最大计时单位；
- (2) 8051 定时器计数寄存器的初始值；
- (3) 若 8051 采用外部时钟接入方式，请完成图 1-3 中的连接方式。



图 1-3 8051 外部时钟接入示意图

【问题 2】(6 分)

根据图 1-2 所示的配送车取货的工作过程示意图，得到图 1-4 所示的配送车取货软件流程图，请在图中的 (n) 处填入适当的内容。

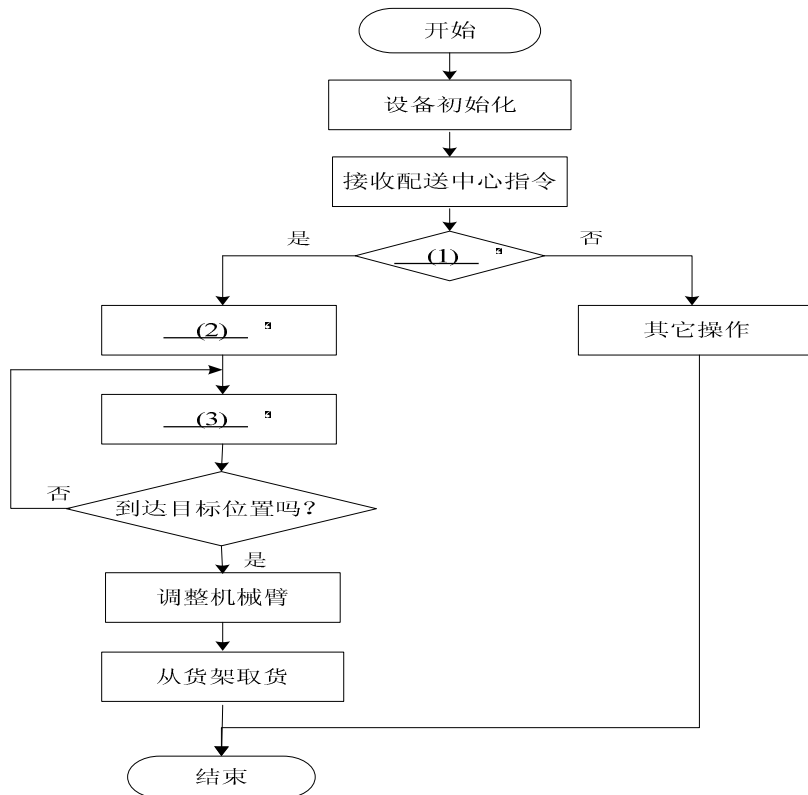


图 1-4 配送车取货的软件流程图

【问题 3】(5 分)

按照你对 8051 微处理机的理解, 填写表 1-1 中的 (1) ~ (5) 空格, 完成表中给出的 5 种寻址方式的指令格式。

表 1-1 8051 寻址方式

寻址方式	汇编指令
直接寻址	MOV A, <u>(1)</u>
寄存器寻址	MOV A, <u>(2)</u>
寄存器间接寻址	MOV A, <u>(3)</u>
立即寻址	MOVX A, <u>(4)</u>
变址寻址	MOVX A, <u>(5)</u> +PC

注: MOV: 将存储器的内容取到累加器中

A: 累加器

Ri: 通用寄存器

PC: 指令寄存器

DPTR: 间址寄存器

试题二(共 15 分)

阅读以下关于嵌入式软件测试的叙述，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某嵌入式软件主要用于控制飞机起落架。飞机起落架的可靠性直接关系着机载人员的人身安全。根据载机设备对软件可靠性要求，一般将软件分为 3 级：关键级软件，主要级软件和一般软件。由于该嵌入式软件被定义为关键软件，要求按关键级软件进行测试。

【问题 1】(5 分)

请根据测试要求，简要说明语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖的含义。

【问题 2】(7 分)

根据本题所示的软件关键级别，回答该软件需要做哪几项覆盖测试？要求的覆盖率指标是多少？如果是一般级软件，应做哪几项覆盖测试？

【问题 3】(3 分)

在软件单元测试中，主要测试对象是软件模块，如果被测程序中有多处调用了其它过程代码，测试中应如何处理这些功能的引用？软件的性能测试在测试工作的哪个阶段进行？

试题三（共 20 分）

阅读以下关于中断的叙述，回答问题 1 至问题 3，将答案填入答题纸的对应栏内。

【说明】

某计算机中断系统有 4 级中断 I1, I2, I3 和 I4，中断响应的优先次序为 I1→I2→I3→I4，即先响应 I1，再响应 I2，……，最后响应 I4。每级中断对应一个屏蔽码，屏蔽码中某位为“1”表示禁止中断（关中断），若为“0”则表示允许中断（开中断）。各级中断处理程序与屏蔽码的关系如表 3-1 所示。

表 3-1 中断处理程序与屏蔽码的关系

中断处理程序	屏蔽码			
	I1 级	I2 级	I3 级	I4 级
I1 级	1	1	1	1
I2 级	0	1	1	1
I3 级	0	0	1	1
I4 级	0	0	0	1

【问题 1】（6 分）

若 t_i 时刻 I1、I2 和 I4 级同时产生中断，在各中断处理完成后， t_j ($t_i < t_j$) 时刻发出 I3 级中断申请，CPU 为 I3 服务时，I1 级发出请求，在 CPU 为 I1 服务时，I2 级发出请求。请参照图 3-1 所示的格式，画出 CPU 的运行轨迹。

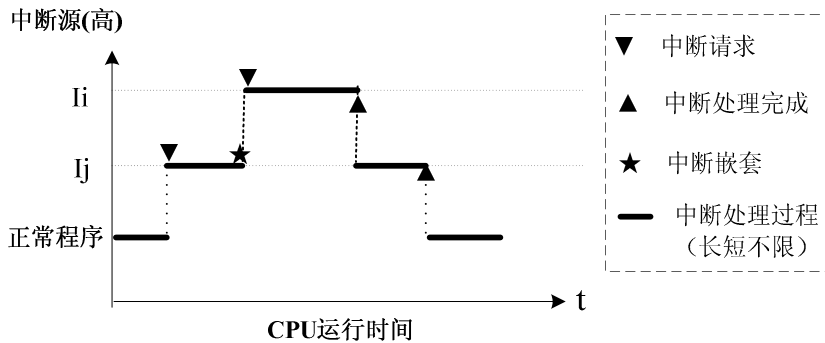


图 3-1 CPU 的运行轨迹示意图

【问题 2】(8 分)

若将中断优先次序设置为 I1→I4→I3→I2，即先响应 I1，再响应 I4，……，最后响应 I2。请重新设置各级的屏蔽码，填写表 3-2。

表 3-2 中断处理程序与屏蔽码的关系

中断处理程序	屏蔽码			
	I1 级	I2 级	I3 级	I4 级
I1 级				
I2 级				
I3 级				
I4 级				

【问题 3】(6 分)

设中断优先次序为 I1→I4→I3→I2。若 t_i 时刻 I1、I2 和 I4 级同时产生中断，在各中断处理完成后， t_j ($t_i < t_j$) 时刻发出 I3 级中断申请，CPU 为 I3 服务时，I1 级发出请求，在 CPU 为 I1 服务时，I2 级发出请求。请画出这种情况下 CPU 的运行轨迹。

试题四(共 20 分)

阅读以下说明和 C 语言代码,回答问题 1 至问题 3,将解答填入答题纸的对应栏内。

【说明】

在实时系统中,许多控制软件需要将数据封装到一个数据结构中,以节省存储空间。对于位操作,使用汇编语言实现其访问比较容易,但会增加编程难度,因此现在普遍采用 C 语言实现。使用高级语言编程要特别注意结构的存储格式以及编译器的特性。本题所使用的编译器对变量按声明顺序分配地址。分析图 4-1 所示的 C 语言代码,回答以下问题。

```
1:  typedef struct
2:      { int A : 16; //按 16 位字对齐;
3:        char B : 8;
4:        char C : 8;
5:        char D : 8;
6:        char E : 8;
7:        int F; //占 16 位并按 16 位字对齐;
8:      }radartype;
9:  typedef struct
10:     { unsigned int X; //占 16 位并按 16 位字对齐;
11:       unsigned int Y;
12:       unsigned int Z;
13:       unsigned int U;
14:     } datatranstype;
15:  radartype myRadarData[2]=
16:     {{1, 'a', 'b', 'c', 0, 512}, {2, 'x', 'y', 'z', 0, 1024}};
17:
18:  void main(void)
19:  { radartype *p;
20:    datatranstype *q;
21:    p = myRadarData;
22:    q = (datatranstype *)p;
23:    q++;
24:  }
```

图 4-1 C 语言程序代码

【问题 1】(8 分)

如果处理机按 16 位以大端方式 (big_endian) 编址, 请在图 4-2 所示的存储器图表中填入 myRadarData 数据的存储内容(十六进制表示)。

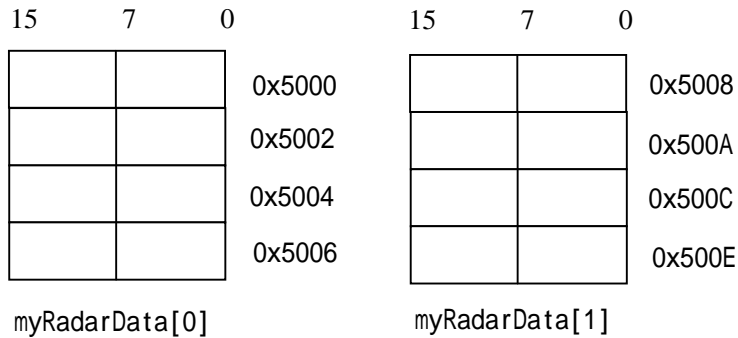


图 4-2 myRadarData 数据存储格式

【问题 2】(8 分)

在图 4-1 所示的程序中, 第 22 行的语句执行完成后, 下列语句的结果是多少? 请将应填入 (n) 处的内容写在答题纸的对应栏中。

q->X = (1)

q->Y = (2)

q->Z = (3)

q->U = (4)

若再执行一次 q++, 则下列语句的结果又是多少? 请将应填入 (n) 处的内容写在答题纸的对应栏中。

q->X = (5)

q->Y = (6)

q->Z = (7)

q->U = (8)

【问题 3】(4 分)

内存空间常划分为代码段 (text)、数据段 (data)、bss 段 (bss)、堆区 (heap) 和栈区 (stack), 那么图 4-1 中 myRadarData 数组的存储空间应分配在哪个段中? 指针变量 p、q 应分配在哪个段中?