

课后答案网，用心为你服务！



[大学答案](#) --- [中学答案](#) --- [考研答案](#) --- [考试答案](#)

最全最多的课后习题参考答案，尽在课后答案网（www.khdaw.com）！

Khdaw团队一直秉承用心为大家服务的宗旨，以关注学生的学习生活为出发点，

旨在为广大学生朋友的自主学习提供一个分享和交流的平台。

爱校园（www.aixiaoyuan.com） 课后答案网（www.khdaw.com） 淘答案（www.taodaan.com）

1-4题 答案

1) 方括号法

二进制数: $[101]_2$

十进制数: $[101]_{10}$

16进制数: $[101]_{16}$

2) 字母法

二进制数: 101B

十进制数: 101D

16进制数: 101H

1-5题 答案

(1) 87H

(2) 0. AH

(3) 2F. BH

(4) 0. FH

(5) 6F. 1CH

(6) 7CB. 1FH

1-6题 答案

- | | |
|-----------------|----------|
| (1) 214 | D6H |
| (2) 823 | 337H |
| (3) 0. 6875 | 0. BH |
| (4) 0. 59765625 | 0. 99H |
| (5) 11. 6875 | B. BH |
| (6) 3855. 84375 | F0F. D8H |

1-7题 答案

- (1) $[AAH]_2 = 1010 \ 1010B$
 $[AAH]_{10} = 170$
- (2) $[BBH]_2 = 1011 \ 1011B$
 $[BBH]_{10} = 187$
- (3) $[C. CH]_2 = 1100. 11B$
 $[C. CH]_{10} = 12. 75$
- (4) $[DE. FCH]_2 = 1101 \ 1110. 111111B$
 $[DE. FCH]_{10} = 222. 984375$
- (5) $[ABC. DH]_2 = 101010111100. 1101B$
 $[ABC. DH]_{10} = 2748. 8125$
- (6) $[128. 08H]_2 = 100101000. 00001B$
 $[128. 08H]_{10} = 296. 03125$

1-8题 答案

- 1) 先把大于1的数按比例缩小成小于1的数进行运算。
- 2) 再把结果按同样比例扩大成原数进行输出。

1-9题 答案

16位(二进制)原码数的范围:

$$-2^{15} \sim +2^{15}$$

16位浮点数(e=5和s=9)的范围:

$$-2^{31} \sim +2^{31}$$

1-10 题答案

- (1) $97H+0FH=A6H$
 $97H-0FH=88H$
- (2) $A6H+33H=D9H$
 $A6H-33H=73H$
- (3) $F3H+F4H=1E7H$
 $F3H-F4H=FFH$
- (4) $B6H+EDH=1A3H$
 $B6H-EDH=F9H$

1-11题 答案

- (1) $110011BX101101B=8F7H$
 $110011B/101101B=1B. 110B$
- (2) $111111BX1011B=2B5H$
 $111111B/1011B=101B. 1000B$

1-11题 答案

- (1) $110011B \times 101101B = 8F7H$
 $110011B / 101101B = 1B \dots\dots 110B$
 (2) $111111B \times 1011B = 2B5H$
 $111111B / 1011B = 101B \dots\dots 1000B$

1-12题 答案

- (1) $33H \wedge BBH = 33H$ (2) $ABH \wedge 7FH = 2BH$
 $33H \vee BBH = BBH$ $ABH \vee 7FH = FFH$
 $33H \oplus BBH = 88H$ $ABH \oplus 7FH = D4H$
 (2) $CDH \wedge 80H = 80H$ (4) $78H \wedge 0FH = 08H$
 $CDH \vee 80H = CDH$ $78H \vee 0FH = 7FH$
 $CDH \oplus 80H = 4DH$ $78H \oplus 0FH = 77H$

1-13题 答案

- 1) $k_1=1$, 其余至少有一个是 '1'
 2) $k_3=1$, 其余任取 '0' 或 '1'

1-14题 答案

$k_6=0$ 和 $k_7=0$

1-15题 答案

- 1) $X=+38$ (2) $X=+76$
 $[X]_{\text{原}} = 0010 \ 0110B$ $[X]_{\text{原}} = 0100 \ 1100B$
 $[X]_{\text{原}} = 0010 \ 0110B$ $[X]_{\text{原}} = 0100 \ 1100B$
 $[X]_{\text{原}} = 0010 \ 0110B$ $[X]_{\text{原}} = 0100 \ 1100B$
 3) $X=-54$ (4) $X=-115$
 $[X]_{\text{原}} = 1011 \ 0110B$ $[X]_{\text{原}} = 1111 \ 0011B$
 $[X]_{\text{原}} = 1100 \ 1001B$ $[X]_{\text{原}} = 1000 \ 1100B$
 $[X]_{\text{原}} = 1100 \ 1010B$ $[X]_{\text{原}} = 1000 \ 1101B$

1-16题 答案

- 1) $[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 00101110B + 00110111B = 65H$
 $[X+Y]_{\text{真}} = +01100101B$
- 2) $[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 01001110B + 00001111B = 5DH$
 $[X+Y]_{\text{补}} = +0101 1101B$
- 3) $[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 01110000B + 10101101B = 1DH$
 $[X+Y]_{\text{补}} = +00011101B$
- 4) $[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 11001101B + 01100001B = 2EH$
 $[X+Y]_{\text{补}} = +00101110B$

1-17题 答案

OV的真值表为:

Xn	Yn	Sn	OV(溢出情况)
0	0	1	1 (正溢出)
0	1	0	0 (不溢出)
1	0	1	0 (不溢出)
1	1	0	1 (负溢出)

1-18题 答案

- 1) $X=53, Y=-33$
 $[X+Y]_{\text{变补}} = 00, 0010100B$
 运算结果: 正确
- 2) $X=120, Y=38$
 $[X+Y]_{\text{变补}} = 01, 0011110B$
 运算结果: 正溢出
- 2) $X=-115, Y=-36$
 $[X+Y]_{\text{变补}} = 10, 1101001B$
 运算结果: 负溢出
- 4) $X=-50, Y=-70$
 $[X+Y]_{\text{变补}} = 11, 0001000B$
 运算结果: 正确

1-19题 答案

- 1) $[89]_{\text{补}} = 01011001B = 59H$
 $[89]_{\text{补}} = 0059H$
- 2) $[96]_{\text{补}} = 01100000B = 60H$
 $[96]_{\text{补}} = 0060H$
- 3) $[-39]_{\text{补}} = 11011001B = D9H$
 $[-39]_{\text{补}} = FFD9H$
- 4) $[-113]_{\text{补}} = 10001111B = 8FH$
 $[-113]_{\text{补}} = FF8FH$

1-20题 答案

- 1) $[47]_{\text{BCD}} = 47H$
- 2) $[59]_{\text{BCD}} = 59H$
- 3) $[1996]_{\text{BCD}} = 1996H$
- 4) $[1997.6]_{\text{BCD}} = 1997.6H$

1-22题 答案

一级字库中汉字: 3755个
 二级字库中汉字: 3008个

1-21题 答案

- (1) $[AB8]_{ASCII} = 41\ 42\ 38H$
- (2) $[STUDENT]_{ASCII} = 53\ 54\ 55\ 44\ 45\ 4E\ 54H$
- (3) $[COMPUTER]_{ASCII} = 43\ 4F\ 4D\ 50\ 55\ 54\ 45\ 52H$
- (4) $[GOOD]_{ASCII} = 47\ 4F\ 4F\ 44H$

1-23题 答案

1 方法:

- 1) 区、位号分别转换成16进制数;
- 2) 分别加上20H。

2 举例:

厂 19 07 $\rightarrow +20H=27H$
 $\downarrow$
 $13H+20H=33H$
 即: 厂的国标码为 3327H

1-24 题答案

奇偶校验码: 1位冗余

ASCII码的

海明码: 4位冗余

1-25 题答案:

- 1) 根据权值为 2^0 、 2^1 、 2^2 和 2^3 的位为校验位的规定, 海明码的校验位应为第1位、第2位、第4位和第8位
- 2) 'B' 的ASCII码为100 0010B, 填入海明码相应位后为:

1	2	3	4	5	6	7	8	9	10	11
		1		0	0	0		0	1	0

- 3) 若采用偶校验, 根据书中表1-5计算出:

$P_1=1$ (奇数个1) $P_2=0$ (偶数个1)

$P_3=0$ (偶数个1) $P_4=1$ (奇数个1)

- 4) 填入 2) 中的1、2、4、8位后得 'B' 的海明码为:

1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	0	0	0	1	0	1	0

1-28 答案案

- 4位机: CPU一次只能处理4位二进制数, 如MCS-48系列机;
8位机: CPU一次能处理8位二进制数, 如MCS-51系列机;
16位机: CPU一次能处理16位二进制数, 如M68HC16系列机;
32位机: CPU一次能处理32位二进制数, 如M69300系列机。

三、填空题(1-20题)

- 1-1 十进制整数转换为二进制数的方法是_____法, 十进制整数转换为十六进制数的方法为_____法; 十进制小数转换为二进制小数的方法为_____法, 十进制小数转换为16进制小数的方法是_____法。
- 1-2 在定点整数机中, 一个二进制16位的原码数N的可表示范围为_____, 在浮点机中, 同样一个二进制16位(阶码5位和尾数9位)数N的可表示范围要大得多, 通常可写为_____。
- 1-3 在补码运算中, 正数的原码、反码和补码是同一种形式, 负数的补码是它的_____, 负数反码的数值位是其原码数值位的_____。
- 1-4 一个n位补码数扩展为2n位补码数时, 只要把_____位向左扩展n位, 其值不变。
- 1-5 $[X]_{\text{补}}$ 连同符号位一起右移1位(符号位不变), 其值变为_____。
- 1-6 在补码的加减运算中, 只要运算结果_____, 其和/差总是正确的。
- 1-7 一个8位二进制补码数的范围是_____。
- 1-8 一个n位二进制补码数的范围是_____。
- 1-9 在二进制8位补码数的运算中, 机器判断运算结果溢出的算式是OV=_____。
- 1-10 变形补码和补码类似, 变形补码比补码多一位_____位。
- 1-11 在变形补码运算中, 若两数和的变形补码的符号位变为_____, 则表示运算过程中产生了正溢出; 若两数和的变形补码的符号位变为_____, 则表示运算过程中产生了负溢出。
- 1-12 由于BCD加法是一种十进制加法, 其低4位向高4位进位是逢十进一的, 但单片机的加法器是一种二进制的加法器, 其低4位向高4位是按_____的, 故在BCD加法程序设计中必须在加法指令后跟一条BCD调整指令。

参考
答案

三、填空题(1-20题)

- 1-1 十进制整数转换为二进制数的方法是 除2取余 法, 十进制整数转换为十六进制数的方法为 除16取余 法; 十进制小数转换为二进制小数的方法为 乘2取整 法, 十进制小数转换为十六进制小数的方法是 乘16取整 法。
- 1-2 在定点整数机中, 一个二进制16位的原码数N的可表示范围为 $-2^{15} \leq N \leq 2^{15}-1$ 。在浮点机中, 同样一个二进制16位(阶码5位和尾数9位)数N的可表示范围要大得多, 通常可写为 $-2^{31} \leq N \leq 2^{31}$ 。
- 1-3 在补码运算中, 正数的原码、反码和补码是同一种形式, 负数的补码是它的 反码加1, 负数反码的数值位是其原码数值位的 按位取反。
- 1-4 一个n位补码数扩展为2n位补码数时, 只要把 符号 位向左扩展n位, 其值不变。
- 1-5 $[X]_{\text{补}}$ 连同符号位一起右移1位(符号位不变), 其值变为 $[\frac{X}{2}]_{\text{补}}$ 。
- 1-6 在补码的加减运算中, 只要运算结果 不溢出, 其和/差总是正确的。
- 1-7 一个8位二进制补码数的范围是 $-128 \sim +127$ 。
- 1-8 一个n位二进制补码数的范围是 $-2^{n-1} \sim 2^{n-1}-1$ 。
- 1-9 在二进制8位补码数的运算中, 机器判断运算结果溢出的算式是 $OV = \text{Cs} + \text{Cp}$ 。
- 1-10 变形补码和补码类似, 变形补码比补码多一位 符号 位。
- 1-11 在变形补码运算中, 若两数和的变形补码的符号位变为 01B, 则表示运算过程中产生了正溢出; 若两数和的变形补码的符号位变为 10B, 则表示运算过程中产生了负溢出。
- 1-12 由于BCD加法是一种十进制加法, 其低4位向高4位进位是逢十进一的, 但单片机法器是一种二进制的加法器, 其低4位向高4位是按 逢16进一 原则进位的, 故BCD加法程序设计中必须在加法指令后跟一条BCD调整指令。

三、填空题(13-20题)

- 1-13 0~9的ASCII码和它的值之间相差 30H。
- 1-14 一级汉字有 3755 个; 二级汉字有 3008 个。
- 1-15 “厂”的国标码是 3327H; “厂”的区位码是 1907H。
- 1-16 奇偶校验码编码和解码只能发现代码在传输过程中有无错误, 但海明码编码和解不仅可以发现错误, 而且还可以 纠正 代码在传输过程中产生的错误。
- 1-17 海明码编码后有 11 位, 其中有效位可以有 7 位; 校验位有 4 位。
- 1-18 如果说海明码编码是形成校验位, 那么海明码解码应该是计算 纠错 位。
- 1-19 除CPU外, 单片机内部还包括 RAM、ROM、I/O接口、定时器/计数器 和 中断系统 等电路。
- 1-20 若某单片机有20条地址线和16条数据线, 则该单片机的寻址能力应该是 1MB 字长为 16 位, 该单片机属于 16 位的单片机。

、选择题 (1-10题)

- 1-1 如下数制中的数, _____ 的值最大; _____ 的值最小。
A、568H B、2555
C、110 1010 1101B D、ABCH
- 1-2 设: $X=ABH$ 和 $Y=0FH$, 现欲从 X 中取出低4位(高4位变为0), 则 X 和 Y 两数应采用 _____ 运算。
A、逻辑加 B、逻辑乘
C、异或 D、取反
- 1-3 设: $X=ABH$ 和 $Y=FOH$, 现欲使 X 的高4位取反和低4位不变, 则 X 和 Y 两数应采用 _____ 运算。
A、逻辑加 B、逻辑乘
C、异或 D、取反
- 1-4 $[-0]_{反}$ 是 _____; $[-0]_{补}$ 是 _____。
A、0000 0000B B、1000 0000B
C、1111 1111B
- 1-5 设: $X=-127$ 和 $Y=-8$, 在 $[X+Y]_{变补}$ 运算后, 符号位变为 _____ 表明运过程中产生了 _____。
A、00B B、01B
C、10B D、11B
- 1-6 如下 _____ 不是BCD码。
A、1001 1001B B、0110 1001B
C、0001 0101 0111B D、1010 1000B

7-10 题

参考
答案

四、选择题(1-10题)

- 1-1 如下数制中的数，D 的值最大；A 的值最小。
A、568H B、2555
C、110 1010 1101B D、ABCH
- 1-2 设：X=ABH和Y=0FH，现欲从X中取出低4位（高4位变为0），则X和Y两数应采用B 运算。
A、逻辑加 B、逻辑乘
C、异或 D、取反
- 1-3 设：X=ABH和Y=FOH，现欲使X的高4位取反和低4位不变，则X和Y两数应采用C 运算。
A、逻辑加 B、逻辑乘
C、异或 D、取反
- 1-4 $[-0]_{反}$ 是 C； $[-0]_{补}$ 是 A。
A、0000 0000B B、1000 0000B
C、1111 1111B
- 1-5 设：X=-127和Y=-8，在 $[X+Y]_{变补}$ 运算后，符号位变为 C 表明运过程中产生了 负溢出。
A、00B B、01B
C、10B D、11B
- 1-6 如下 D 不是BCD码。
A、1001 1001B B、0110 1001B
C、0001 0101 0111B D、1010 1000B

四、选择题(7-10题)

- 1-7 如下 B 不是BCD码。
A、23H
C、123H
B、A8H
D、95H
- 1-8 MCS-51是属于 B 单片机。
A、4位
C、16位
B、8位
D、32位
- 1-9 在分布式控制系统中，单片机主要用在 C 系统中
A、MIS级
C、DDC级
B、SCC级
- 1-10 以下单片机中， D 是Motorola公司的单片机。
A、8751
C、AT89C51
B、PIC16C61
D、MC6805

2-1题 答案

工艺上单片机可分为:

1) HMOS型

特点:(1)对电平要求低

高电平--大于2.0V

低电平--小于0.8V

(2)功耗大

2) CMOS型

特点:(1)对电平要求高

高电平--大于4.5V

低电平--小于0.45V

(2)电流小,功耗小(10 μ A)

2-2题 答案

程序计数器 PC

16位计数器

它是SFR中的一个

2-4题 答案

堆栈---符合"先进后出"或"后进先出"的RAM区

容量---128字节(8031/8051)

SP---8位二进制计数器

作用---存放栈顶/栈底地址
初始化后, SP=07H

2-5题 答案

数据指针DPTR:

二进制16位计数器

作用:

存放外部RAM/
ROM地址。

2-6题 答案

MCS-51寻址能力: 2X64KB

寻址范围:

ROM--0000H-FFFFH

RAM--0000H-FFFFH

00H-FFH(内部)

用户最多可配容量:

ROM--0000H-FFFFH

RAM--0000H-FFFFH

00H-FFH(内部)

2-7题 答案

8051内部RAM有128字节

内部RAM分为3个区:

1) 工作寄存器区

0#区---00H-07H

1#区---08H-0FH

2#区---10H-17H

3#区---18H-1FH

2) 位寻址区

20H-2FH

3) 便笺区

30H-7FH

2-8题 答案

SFR个数:

8031/8051--21个

8032/8052--26个

可位寻址有11个, 如右图所示。

寄存器号	D7	D6	D5	D4	D3	D2	D1	D0	字节地址
B	F7	F6	F5	F4	F3	F2	F1	F0	FOH
ACC	E7	E6	E5	E4	E3	E2	E1	E0	EOH
PSW	D7	D6	D5	D4	D3	D2	D1	D0	DOH
IP	-	-	-	BC	BB	BA	B9	B8	B8H
P3	B7	B6	B5	B4	B3	B2	B1	B0	BOH
IE	AF	-	-	AC	AB	AA	A9	A8	A8H
P2	A7	A6	A5	A4	A3	A2	A1	A0	AOH
SCON	9F	9E	9D	9C	9B	9A	99	98	98H
P1	97	96	95	94	93	92	91	90	90H
TCON	8F	8E	8D	8C	8B	8A	89	88	88H
P0	87	86	85	84	83	82	81	80	80H

2-9题 答案

P0、P1、P2和P3等属于SFR
物理地址为:

P0--80H

P1--90H

P2--A0H

P3--B0H

作用:

1) 存放I/O数据

2) 控制相应引脚电平。

2-10题 答案

8051主要有以下5部分组成:

1) CPU结构

ALU ——算术/逻辑运算

定时控制部件—由OSC和IR等组成

专用寄存器组—PC、A、B、PSW、SP和DPTR等。

2) 存储器结构

片内ROM—0000H~0FFFFH (4KB)

片外ROM—0000H~FFFFH (最多64KB)

片内RAM—00H~7FH (128B)

片外RAM—0000H~FFFFH (64KB)

SFR——21个

3) I/O端口

并行口——P0、P1、P2和P3

串行口——串行发送/接收

4) 定时/计数器

T0/T1 (2X16) 有定时和计数两种模式

5) 中断电路 (5级)

T0、T1、INT0、INT1和串口中断

2-11题 答案

P2口——传送片外ROM/RAM
的高8位地址

P0口——传送片外ROM/RAM
的低8位地址

采用片外地址锁存器的原因:

因为P0口要先传送低8位地址, 后
传送I/O数据。

2-12题 答案

ALE用于锁存P0口低8位地址。

ALE上脉冲频率在不访问片外RAM/ROM时
为 $f_{osc}/6$ 。

ALE上系列脉冲可作为外部时钟源使用。

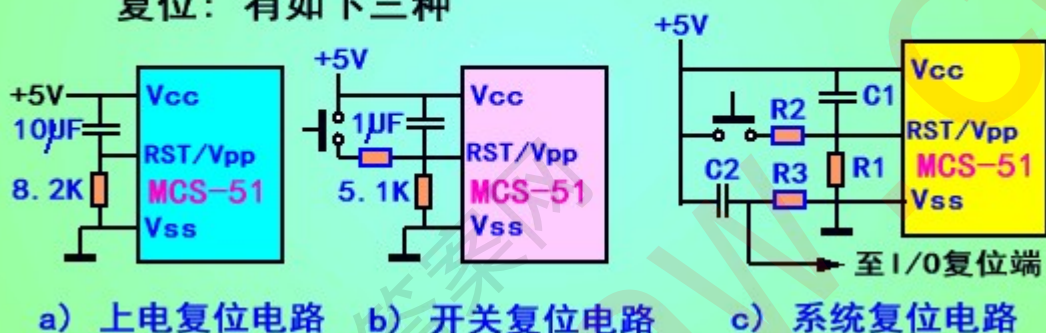
2-13题 答案

$\overline{\text{PSEN}}$ ——片外ROM的选通线
 $\left. \begin{array}{l} \overline{\text{RD}} \\ \overline{\text{WR}} \end{array} \right\}$ ——片外RAM的选通线

2-15题 答案

作用：令CPU处于待命状态

复位：有如下三种



2-17题 答案

空闲方式：一种节电方式。CPU不工作，但中断、定时器、串口电路正常，RAM和SFR功能保留。

功耗：1.7mA~5mA (正常11mA~20mA)

进入方法：PCON. 0 $\leftarrow$ 1

退出方法：(1) 令 $\overline{\text{INT0}}/\overline{\text{INT1}} = \text{高电平}$

(2) RST $\leftarrow$

2-18题 答案

掉电状态：fosc停振，串口、中断功能消失，但内部RAM和SFR功能保留。

功耗：5μA~50μA

进入方法：PCON. 1 $\leftarrow$ 1

退出方法：RST $\leftarrow$

2-19题 答案

节电型器件：CHMOS型

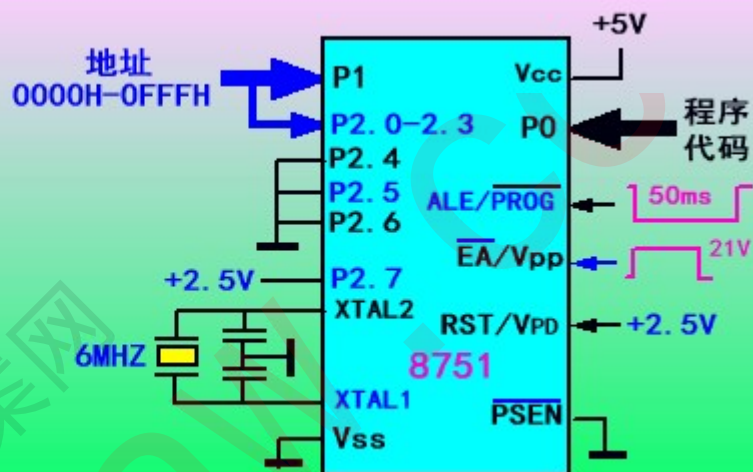
编程/校验型器件：EPROM型

2-20题 答案

所需电路如右图。

所需时间为：

$$T=4096*(50\text{ms}+\text{校验时间})$$

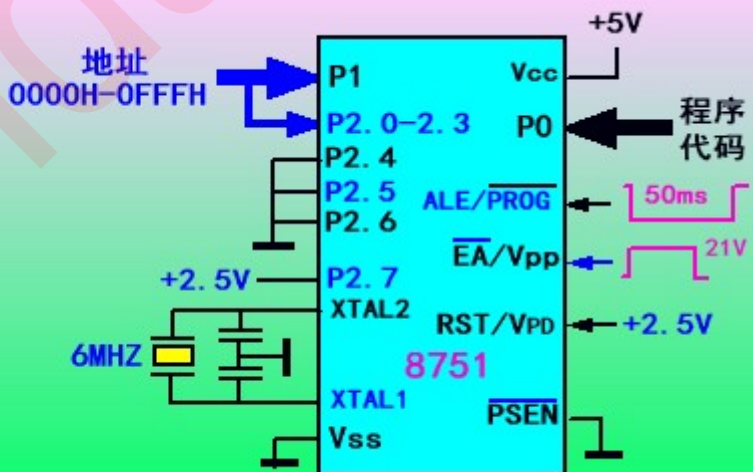


2-20题 答案

所需电路如右图。

所需时间为：

$$T=4096*(50\text{ms}+\text{校验时间})$$



2-21题 答案

保密编程——程序代码写入时保密位也被写入，用于保护软件版权。

异 同：

编程——写入的程序代码允许执行，也允许读出/再编程。

保密编程——写入的程序代码允许执行，但不可读出/再编程且能禁止对外部ROM的访问。

引脚连接：P2. 6应接TTL高电平。

2-22题 答案

时 钟 周 期：最小时间单位， $T = 1/f_{osc}$

机 器 周 期：实现特定功能所需时间。

指 令 周 期：执行一条指令所需时间。

对于MCS-51：1个机器周期=12T

三、填空题(1-12题)

- 2-1 MCS-51的CPU有三部分电路组成：一是 ALU；二是 定时控制部件；三是 专用寄存器组。
- 2-2 MCS-51内部有一个程序计数器PC，它实际上是一个有 16 个触发器组成的计数器，能自动加“1”，用来存放 将要执行指令 的地址。
- 2-3 MCS-51内部有一个寄存器，该寄存器在除法指令执行前用来存放被除数，在除法指令执行后用来存放两数之商，该寄存器通常称为 累加器A。
- 2-4 堆栈指示器（又称堆栈指针）SP是一个有二进制8位的计数器，用来存放堆栈的底或栈顶地址。若堆栈中无数，则SP中的地址必然是 栈底 地址；若堆栈中压入数据，则SP中的地址称为 栈顶 地址。
- 2-5 DPTR称为数据指针，由 DPH 和 DPL 拼装而成，可以用来存放程序存储器ROM或外部RAM中数据的地址。
- 2-6 当MCS-51欲寻址内部ROM工作时， $\overline{EA}$ 引脚应接 +5V 电平。
- 2-7 8751内部有一个4KB容量的EPROM存储器，它的地址范围是 000H—FFFH。
- 2-8 MCS-51内部有一称为Rn(n为0—7)的工作寄存器组，其物理地址区为 00H—1FH。
- 2-9 MCS-51和外部存储器相连时，其CPU先是要通过P0口输出外部存储器的低8位地址，后要传送外部存储器的读/写数。因此，外部存储器低8位地址在P0口的保持时间较短，通常要把它锁存到外部地址锁存器74LS373中。该锁存器由CPU的 ALE 引上信号的下降沿加以选通。
- 2-10 MCS-51通过 TXD 引脚可以串行发送数据，通过 RXD 引脚可以串行接收数据。
- 2-11 MCS-51内部有两个二进制 16 位的定时器，它可以设定为定时器模式，也可以设定为计数器模式。
- 2-12 MCS-51有两个外部中断输入引脚，它们是 INT0 和 INT1。

三、填充题(13-20题)

- 2-13 MCS-51和外部存储器进行硬件连接时，CPU送给外部存储器的16位地址码，其高位由 P2 口输出和低8位由 P0 口输出。
- 2-14 MCS-51和外部存储器硬件相连时，RD 和 WR 引脚可以用来选通外部RAM工作。
- 2-15 8031可以用来指示地址总线上的地址码是送给外部ROM的引脚是 PSEN，该引脚上输出低电平有效。
- 2-16 给8031的RST引脚送一个宽度大于24T的正脉冲可以使8031处于复位状态。此时，PC中内容为 0000H，SP中内容为 07H。
- 2-17 欲使8751从掉电状态中退出，则应给 RST 引脚上送一个足够宽的正脉冲。
- 2-18 若设8051主频 $f_{osc}=12\text{MHz}$ ，则8051最短指令的执行时间为 1 μs ；最长指令的行时间为 4 μs 。
- 2-19 MCS-51执行MOVX A, @DPTR指令需要两次访问存储器：第1次是根据 PC 中内容址的，目的是取指令；第2次是根据 DPTR 中内容寻址的，目的是执行指令。
- 2-20 若对8751的编程脉冲是50MS宽的负脉冲，采用边编程和边校验方式，则8751完编程和校验大约需要 245.76 秒钟。

四、选择题(1-7题)

- 2-1 下列单片机中, D 是EPROM型的。
A、8051 B、8031
C、8032A D、8751
- 2-2 MCS-51的下列指针寄存器中, C 是堆栈指针寄存器。
A、PC B、DPTR
C、SP D、R0
- 2-3 MCS-51的堆栈区在 A 中。
A、内部RAM B、外部RAM
C、内部ROM D、外部ROM
- 2-4 MCS-51有4个三种不同类型的存储器。其中, B 的寻址范围是00H—7FH/FFH
A、内部ROM B、内部RAM
C、外部ROM D、外部RAM
- 2-5 对8751编程时, 其片内所需12位地址码是由 C 端口输入的。
A、P1. 3—P1. 0和P0 B、P2. 3—P2. 0和P0
C、P2. 3—P2. 0和P1 D、P3. 3—P3. 0和P2
- 2-6 对8751校验时, 其读出数据由 A 端口输出。
A、P0 B、P1
C、P2 D、P3
- 2-7 在8031的CPU中, 驱动能力最大的端口是 A。
A、P0 B、P1
C、P2 D、P3

四、选择题(8-12题)

- 2-8 8031执行MOVX A, @DPTR指令时, 需要两次访问内存, 访问次序为 B。
- A、外部RAM和外部RAM B、外部ROM和外部RAM
C、外部RAM和外部ROM D、外部ROM和外部ROM
- 2-9 8031在执行MOVC A, @A+PC指令时, 需要两次访问内存, 访问次序是 D。
- A、外部RAM和外部RAM B、外部ROM和外部RAM
C、外部RAM和外部ROM D、外部ROM和外部ROM
- 2-10 如下引脚中, C 是外部ROM存储器的选通线。
- A、RD和WR B、EA
C、PSEN D、ALE
- 2-11 如下两条指令执行后, 累加器A和PSW中内容是 A。
- MOV A, #19H
ADD A, #66H
- A、7FH和01H B、85H和01H
C、85H和03H D、7FH和03H
- 2-12 MCS-51可以用来控制它进入节电工作方式的特殊功能寄存器是 C。
- A、TCON B、SCON
C、PCON D、TMOD

3-7题 答案

- (1) 7450H 50H为立即数
E550H 50H为字节地址
755020H 50H为字节地址
A250H 50H为位地址
852050H 50H为字节地址
(2) 7666H 66H为立即数
7E66H 66H为立即数
756645H 66H为字节地址
9266H 66H为位地址
8966H 66H为字节地址

3-8题 答案

- (1) 7460H 结果为:
7840H A=40H
F6H R0=60H
8841H (40H)=60H
C8H (41H)=40H
(2) 902003H 结果为:
7418H DPTR=2003H
752038H A=38
7820H (20H)=18
C6H R0=20H

3-9题 答案

- (1) MOV A, R1
MOV R0, A
(2) MOV 30H, 20H
(3) MOV R0, #20H
MOVX A, @R0
MOV 20H, A
(4) MOV DPTR, #2000H
MOVX A, @DPTR
MOV 20H, A
(5) MOV A, #00H
MOV DPTR, #2000H
MOVC A, @A+DPTR
MOV 20H, A
(6) MOV A, #00H
MOV DPTR, #2000H
MOVC A, @A+DPTR
MOV DPTR, #3000H
MOVX @DPTR, A

3-10题 答案

```
ORG 0500H
MOV DPTR, #2050H
MOVX A, @DPTR      ; A ← (2050H) = X
MOV R0, A           ; 存入R0
MOV DPTR, #2060H
MOVX A, @DPTR      ; A ← (2060H) = Y
XCH A, R0           ; A ↔ R0
MOVX @DPTR, A       ; 2060H ← X
MOV A, R0           ; A ← Y
MOV DPTR, #2050H
MOVX @DPTR, A       ; 2050H ← Y
SJMP $
END
```

3-11题 答案

(1)



(2)



3-12题 答案

(1) (20H)=Z
(21H)=Y
(22H)=X

(2) (20H)=X
(21H)=Y
(22H)=Z



3-13题 答案

(1) A=FCH PSW=C0H
(2) A=36H PSW=84H

3-14题 答案

(1) A=20H (2) A=D5H
(30H)=A6H (30H)=不变
PSW=C1H PSW=85H
R0=不变 R0=不变

(3) A=49H (4) A=49H
(30H)=不变 (30H)=A4H
PSW=01H PSW=01H
R0=2FH R0=不变

3-15题 答案

```

ORG 0500H
MOV R0, #00H ; R0清零
MOV A, M1 ; X1送A
ADD A, M2 ; R1 ← X1+X2
MOV R1, A ; Cy ← 进位位
MOV A, R0
ADDC A, #00H ; R0 ← R0+Cy
MOV R0, A
MOV A, R1 ; R1 ← X1+X2+X3
ADD A, M3 ; Cy ← 进位位
MOV R1, A ; 送入R1
MOV A, R0
ADDC A, #00H ; R0 ← R0+Cy
MOV R0, A
SJMP $
END
    
```

3-17题 答案

```

ORG 0800H
MOV R0, #20H ; 20H送R0
MOV A, 20H
MOV B, 21H
DIV AB ; A/B=A...B
MOVX @R0, A ; 商送20H
INC R0
MOV A, B
MOVX @R0, A ; 余数送21H
SJMP $
END
    
```

3-18题 答案

```

ORG 0900H
MOV R0, #30H ; R0指向30H
CLR C ; 清零Cy
MOV A, #5DH
SUBB A, #0B4H ; 5DH-B4H送A
MOV @R0, A ; 送回30H
INC R0
MOV A, #6FH ; 6FH送A
SUBB A, #13H ; 6FH-13H-Cy送A
MOV @R0, A ; 送回31H
SJMP $
END
    
```

3-19题 答案

- (1) (30H)=AB (2) (40H)=AAH
 A=E1H A=AAH
 (3) (30H)=7AH (4) A=F0H
 A=10H

3-20题 答案

- (1) ANL 20H, #3FH
 (2) ORL 20H, #0C0H
 (3) XRL 20H, #0C0H
 (4) XRL 20H, #0FFH

3-21题 答案

```

(1)  ORG    0200H
      MOV    DPTR, #2000H
      MOVX   A, @DPTR           ; X送A
      MOV    20H, A             ; 送入20H单元
      RL     A                   ; 2X送A
      ADD    A, 20H              ; 3X送A
      MOV    20H, A             ; 送回20H
      INC    DPTR
      MOVX   A, @DPTR           ; Y送A
      RL     A                   ; 2Y送A
      ADD    A, 20H              ; 3X+2Y送A
      MOV    20H, A             ; 送入20H单元
      SJMP   $
      END
  
```

3-22题 答案

```

(1)      :
      CJNE  A, #14H, LOOP
LOOP: JNC   LABEL
      SJMP  $
LABEL:   :
      :
      END

(2)      :
      CJNE  A, #14H, LOOP
LOOP: JC    LABEL
      SJMP  $
LABEL:   :
      :
      END

(3)      :
      CJNE  A, #11, LOOP
LOOP: JC    LABEL
      SJMP  $
LABEL:   :
      :
      END

(4)      :
      CJNE  A, #10, LOOP
LOOP: JC    LABEL
      SJMP  $
LABEL:   :
      :
      END
  
```

3-23题 答案

```
ORG 1000H
MOV DPTR, #DATA1 ; DPTR指向DATA1
MOV R1, #30H ; R1指向内部30H
MOV R2, #14H ; 块长20送R2
LOOP: MOVX A, @DPTR ; 源数送A
      MOV @R1, A ; 送往内部RAM
      INC DPTR
      INC R1
      DJNZ R2, LOOP ; 未完, 则LOOP
      SJMP $
      END
```

3-24题 答案



机器将调用3456H处子程序

3-25题 答案



栈底地址=栈顶地址
所以堆栈为空
PC=2348H

3-26题 答案

若 SUBPR=4060H, 则有

- (1) 是错误的, 因MA和4000H不在同一个2KB范围内
- (2) 正确
- (3) 最好, 因指令字节数最少

若SUBPR=2060H, 则

- (2) 是错误的, 因ACALL 指令和目标地址SUBPR不在同一个2KB范围内。
- (1) 最好, 因为能生成浮动代码, 且指令字节数最少。
- (3) 正确

马丙展 23:32:44

3-27题 答案

```
ORG 1100H
MOV R0, #20H      ; R0指向20H
MOV R2, #20        ; 20送R2
MOV DPTR, #2000H   ; DPTR指向2000H
LOOP: MOVX A, @DPTR ; 源数送A
      JB ACC. 7, NEXT ; 若负, 则NEXT
      MOV @R0, A      ; 否则, 送内部RAM
      INC R0          ; R0加1
NEXT: INC DPTR        ; DPTR加1
      DJNZ R2, LOOP    ; 未完, 则LOOP
      SJMP $
      END
```

3-28题 答案

```
ORG 1200H
CLR C ; Cy清零
MOV A, 20H
SUBB A, 21H ; 两数差送A
JNB ACC. 7, NEXT ; 若正, 则NEXT
CPL A ; 若负, 则求补
INC A
NEXT: MOV 30H, A ; 送回绝对值
SJMP $
END
```

三、填空题(1-11题)

- 3-1 MCS-51指令系统中的指令采用单地址指令格式, 每条指令均由 操作码 和 操作数/操作数地址 等字段组成。其中, 操作码 字段是必选段; 下地址字段隐含在 程序计数器PC 中。
- 3-2 指令码中含有操作数本身的指令, 其寻址方式必为 立即型 寻址。
- 3-3 指令码中含有操作数所在寄存器号的指令, 其源操作数的寻址方式一定为 寄存器间接 寻址。
- 3-4 在SJMP rel 指令中, 以其在只读存储器中始址为参考点可以在 -126~+129 范围内转移。
- 3-5 MCS-51有两条查表指令, 指令的通式是 MOVC A, @A+DPTR 和 MOVC A, @A+PC。
- 3-6 只能读外部RAM零页单元中内容的指令是 MOVX A, @Ri。
- 3-7 除INC SP指令外, 还有 PUSH direct 和 调用 指令执行时可以使SP中内容加“1”。
- 3-8 除DEC SP指令外, 还有 POP direct 和 RET/RETI 指令可以使SP中内容减“1”。
- 3-9 不带Cy的移位指令有两条: 一条是 RL A; 另一条是 RR A。
- 3-10 能对PC中内容运算处理为目的的指令, 它必然属于 控制转移 类指令。
- 3-11 MCS-51有两条可以根据累加器A中内容为判断条件的转移指令, 这两条指令是 JZ rel 和 JNZ rel。

三、填空题(12-17题)

- 3-12 MCS-51有两条减1条件转移指令，通常在循环程序设计中使⤵用。该指令通式是 DJNZ Rn, rel 和 DJNZ direct, rel。
- 3-13 调用指令的作用有两个：一个是 断点地址入栈；另一个是 子程序地址送PC。
- 3-14 返回指令的作用是：断点地址(栈内)恢复到PC。
- 3-15 MCS-51有两条可以根据Cy中内容为条件的转换指令，它们是 JC rel 和 JNC rel。
- 3-16 MCS-51有三条可以根据位地址中内容为判断条件的转移指令。它们是：JB bit, rel、JNB bit, rel 和 JBC bit, rel。
- 3-17 某单片机主频 $f_{osc}=12\text{MHz}$ ，CPU执行如下子程序可以延时 20.003 MS。

	ORG 1000H	机器周期数
DELAY:	MOV R2, #250	1
DELAY1:	MOV R3, #38	1
	NOP	1
DELAY2:	DJNZ R3, DELAY2	2
	DJNZ R2, DELAY1	2
	RET	2

四、选择题(1-7题)

- 3-1 MCS-51无 D 指令。
A、单字节
B、双字节
C、三字节
D、四字节
- 3-2 MCS-51无 C 指令。
A、单机器周期
B、双机器周期
C、三机器周期
D、四机器周期
- 3-3 如下指令中的源操作数, B 是直接寻址。
A、MOV A, #20H
B、MOV A, 20H
C、MOV A, R0
D、MOV A, @R0
- 3-4 如指令中的源操作数, A 是位寻址。
A、SETB 20H
B、MOV A, 20H
C、MOV A, #20H
D、DJNZ 20H, rel
- 3-5 如下指令中, C 指令的源操作数寻址方式不属于变址寻址。
A、MOVC A, @A+PC
B、MOVC A, @A+DPTR
C、MOVX A, @DPTR
D、JMP @A+DPTR
- 3-6 如下指令中, D 是合法指令
A、MOV R1, R0
B、MOV @R0, R5
C、MOV #20H, A
D、MOV 20H, #20H
- 3-7 C 指令的转移范围最小。
A、LJMP addr16
B、AJMP addr11
C、SJMP rel
D、JMP @A+DPTR

四、选择题(8-12题)

3-8 如下指令中, B 指令是以判断Cy中内容为条件的转移指令。

- | | |
|---------------|--------------|
| A、JZ rel | B、JC rel |
| C、JB bit, rel | JBC bit, rel |

3-9 A 是字节型指令。

- | | |
|-----------|----------|
| A、CLR A | B、CLR C |
| C、CLR 20H | D、SETB C |

3-10 以下指令中, C 指令执行后均能使bit中内容始终为零。

- | | |
|----------------|----------------|
| A、JB bit, rel | B、JNB bit, rel |
| C、JBC bit, rel | |

3-11 如下指令中, D 指令的寻址方式是变址寻址。

- | | |
|---------------|---------------|
| A、LJMP addr16 | B、AJMP addr11 |
| C、SJMP rel | D、JMP @A+DPTR |

3-12 如下指令中, B 类指令执行时均能使SP内容加“1”。

- | | |
|------------------|---------------|
| A、XCH A, R2 | B、PUSH ACC |
| RL A | LCALL SQR |
| C、DJNZ R2, LOOP | D、ADD A, #20H |
| CJNE A, R2, NEXT | SUBB A, R2 |

4-1题 答案

程序设计语言有如下三类:

1、机器语言

- (1) 二进制形式——能直接为CPU识别和执行
- (2) 十六进制形式——能为监控程序翻译成二进制, 供CPU执行。

2、汇编语言(助记符形式)

特点:

- (1) 由人工/机器汇编成机器码供CPU执行, 因机器而异。
- (2) 直接操作到寄存器或内部RAM等。

3、高级语言(符号语言)

特点:

- (1) 由解释/编译程序翻译成机器码, 供CPU执行。
- (2) 面向过程和问题, 独立于机器, 接近于自然语言和数学表达式。
- (3) 语句和机器码间无——对应关系。

汇编语句有如下两类:

1、指令性语句

汇编程序识别后能产生机器码, 供CPU执行。

2、指示性语句

汇编程序识别后能起某种控制作用, 但不会产生机器码。

4-3题 答案

常用伪指令有如下8条。

定义汇编的开始和终止: **ORG/END**

定义字符名称: **EQU/DATA**

定义字节/字/存储空间: **DB/DW/DS**

定义位地址: **BIT**(具体作用略)。

4-6题 答案



相应程序如下:

		字节数	周期数
ORG	0500H		
MOV	A, 20H ; A ← 源数	2	1
MOV	B, #16 ; B ← 16	3	2
DIV	AB ; A/B=A←B	1	4
ORL	A, #30H	2	1
MOV	21H, A ; 21H ← BCDH的ASCII	2	1
ORL	B, #30H	3	2
MOV	22H, B ; 22H ← BCDL的ASCII	3	2
SJMP	\$	2	2
END		18	15

4-7题 答案

```

ORG 0100H
MOV A, #00H      ; A ← 0
MOV R0, #20H     ; R0 ← 20H
XCHD A, @R0      ; } B [0000 | 0000 | BCD1 | BCD0] 20H
MOV B, A          ; }
MOV A, @R0        ; } A ← [0000 | BCD1]
SWAP A            ; }
MUL AB            ; A ← BCD1 * BCD0
MOV B, #0AH       ; B ← 10
DIV AB            ; BCD1 * BCD0 / 10 = A...B
SWAP A            ; }
ORL A, B          ; } [十位BCD | 个位BCD] A
MOV 21H, A        ; 送回21H单元
SJMP $
END

```

4-8题 答案

```

ORG 0200H
LEN DATA 20H
NUM DATA 21H
BLOCK DATA 23H
MOV NUM, #00H      ; 清零NUM单元
MOV NUM+1, #00H    ; 清零NUM+1单元
MOV R0, #BLOCK     ; R0 ← 块始址
LOOP: MOV A, @R0    ; A ← 源数
JNB ACC. 7, POSI   ; 若正, 则POSI
INC NUM+1          ; 负数个数加1
SJMP NEXT
POSI: INC NUM       ; 正数个数加1
NEXT: INC R0        ; 源指针加1
DJNZ LEN, LOOP     ; 若未完, 则LOOP
SJMP $
END

```

4-9题 答案

```

    ORG 0200H
    VAR DATA 20H
    FUNC DATA 21H
    MOV A, VAR ; A ← (20H)
    CJNE A, #50, NEXT1 ; 建立 Cy
NEXT1: JNC DONE ; 若 X ≥ 50, 则 DONE
    CJNE A, #20, NEXT2 ; 建立 Cy
NEXT2: JC NEXT3 ; 若 X < 20, 则 NEXT3
    RL A ; 若 X > 19, 则 A = 5X
    RL A
    ADD A, 20H ; A ← 5X
    SJMP DONE
NEXT3: RL A ; A ← 2X
DONE: MOV FUNC, A ; 21H ← A
    SJMP $
    END

```

4-10题 答案

```

    ORG 0300H
    MOV A, R3
    RL A
    ADD A, R3
    MOV DPTR, #BRTAB ; DPTR ← BRTAB
    JMP @DPTR
    ORG 0400H
BRTAB: LJMP ROUT0
    LJMP ROUT1
    :
    LJMP ROUT84
ROUT0: :
    :
ROUT84: :
    END

```

共有85个分支, 可以在64KB范围内转移。

4-13题 答案

```

ORG    0500H
LEN    DATA 40H
SOUCE  DATA 41H
DIST   DATA 20H
MOV     R1, #DIST    ; A ← 目标地址
MOV     R0, #LEN
MOVX    A, @R0        ;
MOV     R2, A        ; } R2 ← 块长
MOV     R0, #SOUCE   ; R0 ← 源指针
LOOP:  MOVX  A, @R0    ; A ← 源数
      JNB   PSW. 0, NEXT ; 若偶数, 则NEXT
      MOV   @R1, A    ; 若奇数, 则送存
      INC   R1        ; 修改目的指针
NEXT:  INC   R0        ; 修改源指针
      DJNZ  R2, LOOP  ; 未完, 则LOOP
      SJMP  $
      END

```

4-11题 答案

```

ORG    0500H
SOUCE  DATA 30H
DIST   DATA 20H
MOV     R0, #SOUCE   ; R0 ← 源地址
MOV     R1, #DIST    ; R1 ← 目标地址
LOOP:  MOVX  A, @R0    ; A ← 源数
      CJNE  A, #24H, NEXT ; 若A<>$, 则NEXT
      MOV   @R1, A    ; 传送$符
      SJMP  $
NEXT:  MOV   @R1, A    ; 传送数据
      INC   R0        ; 源指针加1
      INC   R1        ; 目的指针加1
      SJMP  LOOP
      END

```

4-15题 答案

```
ORG 0500H
MOV DPTR, #3000H; 目标地址送3000H
PUSH DPL          ; 目标地址送栈
PUSH DPH
MOV DPTR, #2000H; 源地址送DPTR
MOV R2, #01H      ; 块长送 R3R2
MOV R3, #01H
LOOP: MOVX A, @DPTR ; 源数送 A
INC DPTR
MOV R5, DPH        ; 源针加1后送R5R4
MOV R4, DPL
POP DPH            ; 目标地址送DPTR
POP DPL
MOVBX @DPTR, A     ; 送数
INC DPTR           ; 目标地址加1
PUSH DPL           ; 送栈
PUSH DPH
MOV DPH, R5        ; 源针送DPTR
MOV DPL, R4
CLR C              ; Cy清零

MOV A, R2          ; R2-1送R2
SUBB A, #01H
MOV R2, A
MOV A, R3          ; R3-Cy送R3
SUBB A, #00H
MOV R3, A
ORL A, R2          ; R3VR2 送 A
JNZ LOOP          ; R3VR2<>0, 则LOOP
SJMP $            ; 否则, 停机
END
```

4-16题 答案(流程图略)

```

ORG    0700H
PAPE    DATA    20H
NAOE    DATA    21H
FIRST   DATA    22H
MOV     PAPE, #00H
MOV     NAOE, #00H
MOV     R0, #FIRST+1 ; R0 ← 块始址
LOOP:   MOV     A, @R0 ; A ← 源数
        JNB     ACC. 7, POSI ; 正数, 则POSI
        JNB     PSW. 0, NEXT ; 负偶数, 则NEXT
        INC     NAOE ; 负奇数, NAOE加1
        SJMP    NEXT
POSI:   JB      PSW. 0, NEXT ; 正奇数, 则NEXT
        INC     PAPE ; 正偶数, PAPE加1
NEXT:   INC     R0 ; 源指针加1
        DJNZ    FIRST, LOOP ; 未完, 则LOOP
        SJMP    $
        END
    
```

4-17题 答案

```

ORG     0800H
MIN     DATA    20H
BLOCK   DATA    21H
MOV     R0, #BLOCK ; R0 ← 块始址
MOV     R2, #100 ; R2 ← 100
MOV     MIN, #0FFH ; MIN ← 最大数
LOOP:   MOV     A, @R0 ; A ← 源数
        CJNE    A, MIN, NEXT1 ; 形成Cy
NEXT1:  JNC     NEXT2 ; (MIN)数小, NEXT2
        MOV     MIN, A ; 否则, MIN ← 小数
NEXT2:  INC     R0
        DJNZ    R2, LOOP ; 未完, 则LOOP
        SJMP    $
        END
    
```

4-18题 答案

```

ORG 0900H
MOV R0, #FIRST ; FIRST送R0
MOV R1, #SECOND ; SECOND送R1
MOV PSW, #08H ; 进1#工作区
MOV R0, #SUM ; SUM送R0
CLR PSW. 3 ; 返0#工作区
MOV R2, #06H ; 组数送R2
LOOP2: CLR C
MOV R3, #04H ; 字节数送R3
LOOP1: MOV A, @R0 ; 求和
        ADDC A, @R1
        SETB PSW. 3 ; 进入1#工作区
        MOV @R0, A ; 存和
        INC R0
        CLR PSW. 3 ; 返回0#工作区
        INC R0 ; 修改源指针
        INC R1 ; 修改源指针
        DJNZ R3, LOOP1 ; 本组未完, LOOP1
    
```

```

        DJNZ R2, LOOP2 ; 6组未完, LOOP2
        SJMP $
    END
    
```

4-19题 答案

```

ORG 0A00H
LEN DATA 20H
BLOCK DATA 21H
MOV R0, #BLOCK ; R0 ← 块始址
LOOP: MOV A, @R0 ; A ← 源数
        CLR C ; Cy ← 0
        SUBB A, #0AH ; 建Cy标志
        JNC NEXT1 ; A>9, 则NEXT1
        MOV @R0, #00H ; 存0
        SJMP NEXT2
NEXT1: ADD A, #07H ; 修正A
        MOVC A, @A+PC ; 查表得ASCII码
        MOV @R0, A ; 送回
NEXT2: INC R0 ; 源指针加1
        DJNZ LEN, LOOP ; 未完, 则LOOP
        SJMP $
        DB 41H, 42H, 43H, 44H, 45H, 46H
    END
    
```

4-20题 答案

```

ORG    0A00H
MOV     R0, #20H ; A ← 外20H
MOV     A, 20H   ; A ← 源数
RL      A
MOV     B, A
ADD     A, #0BH  ; 修改A
MOVC    A, @A+PC ; A ← 平方值L
MOVX    @R0, A   ; 送存
INC     R0       ; 目标地址加1
MOV     A, B
ADD     A, #03H  ; 修改A
INC     A        ; 指向结果H地偏量
MOVC    A, @A+PC ; 查表得平方值H
MOVX    @R0, A   ; 送存
SJMP    $
SQR TAB: DW    0, 1, 4, 9, ... 10000
END

```

4-21题 答案

```

ORG     0B00H
MOV     R0, #BLOCK ; R0 ← BLOCK
MOV     A, #00H    ; A ← 0
MOV     R2, #0AH   ; R2 ← 10
MOV     SUM+1, #00H ; SUM+1 ← 0
LOOP:   ADD     A, @R0 ; }
        DA      A      ; SUM ← BCD和
        MOV     SUM, A ; }
        JNC     NEXT   ; 无进位, NEXT
        MOV     A, SUM+1 ; A ← (SUM+1)
        ADDC    A, #00H ; }
        DA      A      ; A ← 高位BCD和
        MOV     SUM+1, A ; 送回SUM+1单元
        MOV     A, SUM  ; A ← 低位BCD
NEXT:   INC     R0      ; 源指针加1
        DJNZ    R2, LOOP ; 未完, 则LOOP
        SJMP    $
END

```

4-22题 答案

```

ORG    0B00H
MOV     R0, #BLOCK; 块始址送R0
MOV     A, #00H    ; 清零A
MOV     R2, #0AH   ; 10送R2
MOV     SUM+1, #00H; (SUM+1) 清零
LOOP:  ADD     A, @R0    ; 形成低位BCD和
        DA      A
        MOV     SUM, A    ; 送回SUM
        MOV     A, SUM+1
        INC     R0
        ADDC    A, @R0
        DA      A        ; 形成高位BCD和
        MOV     SUM+1, A  ; 送入SUM+1
        INC     R0
        MOV     A, SUM    ; (SUM) 送入A
        DJNZ    R2, LOOP  ; 若未完, 则LOOP
        SJMP    $
        END
    
```

4-23题 答案

```

ORG 0C00H
MDA DATA 20H
MDB DATA 21H
MDC DATA 22H
MOV MDC+1, #00H; 清零MDC+1
MOV A, MDA      ; a 送 A
ACALL SQR       ; a2 送A
MOV MDC, A      ; 存入MDC
MOV A, MDB      ; b 送 A
ACALL SQR       ; b2 送 A
ADD A, MDC      ; a2+b2 送A
MOV MDC, A      ; 存入MDC
MOV A, MDA      ; a 送 A
MOV B, MDB      ; b 送 B
RL A            ; 2a 送 A
MUL AB         ; 2ab 送 A
    
```

```

        ADD  A, MDC      ;结果 送 A
        MOV  MDC, A      ;存入MDC
        JNC  NEXT       ;若无进位, 则NEXT
        INC  MDC+1       ;否则, (MDC) 加1
NEXT: SJMP $
SQR: ADD  A,
      MOVC A, @A+PC
      RET
SQRTAB: DB 0, 1, 4, 9, ... 81
      END

```

注意:

因为 $a < 10$ 和 $b < 10$
 所以 $2ab$ 必无进位
 $a^2 + b^2$ 必无进位

马丙展 23:40:47

三、填空题(1-18题)

- 4-1 机器语言有两种形式:一种是 二进制 形式;另一种是 16进制 形式, 二进制 形式的机器语言可为CPU直接执行。
- 4-2 汇编语言是一种符号语言,因机器而异,所编程序需汇编成机器语言才能直接为CPU执行,其优点是可直接操作到机器内部的 寄存器 和 RAM 单元。
- 4-3 MCS-51的汇编语句采用典型的四分段格式:标号段、操作码段、操作数段和注释段。其中, 操作码段 是必选段,用来指示CPU执行 何种操作。
- 4-4 注释段以 分号 开头,指示指令段或程序段的作用;标号段是以大写英文字母开头的字母数字串(最多8个),标号段中禁止使用 保留字 和 寄存器号 作标号。
- 4-5 MCS-51宏汇编中,有两条可以为变量赋值的伪指令,它们是 EQU 和 DATA/XDATA。
- 4-6 如下程序为汇编程序识别和执行后, $(1500H) =$ 00H、 $(1501H) =$ 45 和 $(1502H) =$ 41H。

```

      ORG 1500H
TAB: DW 45
      DB 'A'

```

- 4-7 可以定义存储空间的伪指令是 DS。
- 4-8 可以定义位地址的伪指令是 BIT。
- 4-9 用户程序编写好后,必须汇编成机器码才能为机器执行。MCS-51源程序的汇编有手工和机器两种,但原理完全相同。机器汇编需要汇编程序对用户程序进行两次扫描才能完成:第1次扫描主要是确定每条指令的 指令码;第2次扫描是计算指令码的 标号值 和 地址偏移量。
- 4-10 在人工汇编的第2次扫描中,标号值或地址偏移量的计算公式是 地址偏移量 = 目标地址 - 转移指令始址 - 转移指令字节数。

三、填空题(11-18题)

- 4-11 简单程序又称顺序程序或直线程序,有时还称为无分支程序,其特点是程序中不含有 转移 指令。
- 4-12 分支程序有时也称为散转程序,其特点是程序中含有 无条件转移 指令。
- 4-13 MCS-51循环程序有循环初始化、循环体和循环结束处理等三部分组成。其中,循环初始化程序段中通常要给 循环变量(地址指针) 和 循环计数器 赋初值;循环体中则必定含有一条 减“1”条件转移 指令。
- 4-14 循环程序有两种类型:一是 先判断后处理;二是 先处理后判断。
- 4-15 查表指令有两条:一条是用PC作为基址来查表,另一条是用DPTR作为基址来查表,但查表原理是相同的。后者在查表前DPTR中应放被查表的始址,前者在查表时PC中的基址不等于被查表始址,故它在查表指令前必须有一条 ADD A, #data 指令,以便使PC+data=被查表始址。
- 4-16 子程序设计时应注意四点:一是子程序前要冠以标号,标号应以 子程序任务 命名;二是主程中要有一条 调用 指令才能转入子程序执行;三是子程序中必须安排一条 返回 指令才能使CPU回到主程序执行;四是CPU转入子程序或从子程序返回主程序前都需要有参数传递。
- 4-17 带符号补码数加/减运算的算法:一是 先对两数进行符号数扩展;二是 对符号扩展后的两数进行加/减运算。
- 4-18 带符号补码数的乘除运算的算法:一是 符号位单独处理;二是 把参加运算的数变成绝对值;三是 对绝对值进行乘除运算;四是 把运算结果变成补码。

四、指出程序运行结果(1-10)

- 4-1 已知: $(20H) = X$ ($X < 50$) 和 $A = 00H$, 如下程序为CPU执行后DA中内容是什么

```
ORG 0100H
MOV DPTR, #1000H
JMP @A+DPTR
DONE: MOV DA, A
      SJMP $
      ORG 1000H
      MOV A, 20H
      RL A
      SJMP DONE
      ORG 1080H
      MOV A, 20H
      RL A
      RL A
      SJMP DONE
      END
```

(DA) = 2X

四、指出程序运行结果(1-10)

4-2 上例中, 若累加器A中内容为80H, 则重新执行上述程序后DA中内容是什么

```
ORG 0100H
MOV DPTR, #1000H
JMP @A+DPTR
DONE: MOV DA, A
      SJMP $
ORG 1000H
MOV A, 20H
RL A
SJMP DONE
ORG 1080H
MOV A, 20H
RL A
RL A
SJMP DONE
END
```

(DA) = 4X

马丙展 23:42:14

四、指出程序运行结果(1-10)

4-3 已知: (71H) = X、SP=71H和外部RAM的(2000)=Y。试问如下程序执行后外部RAM的2500H单元中内容是什么(X和Y均不超过63)?

```
ORG 0300H
MOV DPTR, #2000H
MOVX A, @DPTR
MOV R2, A
POP ACC
ADD A, R2
RL A
MOV DPTR, #2500H
MOVX @DPTR, A
SJMP $
END
```

(2500H) = 2(X+Y)

四、指出程序运行结果(1-10)

4-4 已知：a和b是两个小于8的二进制数，分别放在DA和DB单元。试问如下程序执行后DC单元中内容是什么？

```
ORG 0400H
MOV A, DA
ACALL SQR
MOV R1, A
MOV A, DB
ACALL SQR
ADD A, R1
MOV R1, A
MOV A, DA
MOV B, DB
MUL AB
RL A
ADD A, R1
MOV DC, A
SJMP $
SQR: INC A
      MOVC A, @A+PC
      RET
SQR TAB: DB 0, 1, 4, 9, 16, 25, 36, 49, 64, 81
END
```

$(DC) = a^2 + 2ab + b^2$

四、指出程序运行结果(1-10)

4-5 如下程序执行后累加器A和70H单元中内容是什么？

```
ORG 0500H
MOV A, #0AAH
MOV DPTR, #0070H
MOVX @DPTR, A
MOV R0, #70H
MOV @R0, A
MOV A, #0BBH
XCHD A, @R0
SJMP $
END
```

$A = BAH$ $(70H) = ABH$

四、指出程序运行结果(1-10)

4-6 已知:外部RAM(1000H)=X, 如下程序执行后20H和SP单元中内容是什么(X <85)?

```
ORG 0600H
MOV SP, #60H
MOV DPTR, #1000H
MOVX A, @DPTR
MOV B, A
RL A
ADD A, B
MOV 61H, A
INC SP
POP 20H
SJMP $
END
```

(20H) = 3X SP = 60H

四、指出程序运行结果(1-10)

4-7 已知:内部RAM(20H)=FFH(-1), 如下程序执行后30H单元中内容是什么(X <127)?

```
ORG 0700H
MOV A, 20H
JNB ACC.7, NEXT
MOV 30H, #X
SJMP DONE
NEXT: MOV 30H, #2X
DONE: SJMP $
END
```

(30H) = X

四、指出程序运行结果(1-10)

4-8 已知:内部RAM (20H) = 90H, 如下程序执行后30H单元中内容是什么 ($X < 127$) ?

```
ORG 0800H
MOV A, 20H
ADD A, #80H
JNB ACC. 7, NEXT
MOV 30H, #X
SJMP DONE
NEXT: MOV 30H, #2X
DONE: SJMP $
END
(30H) = 2X
```

四、指出程序运行结果(1-10)

4-9 如下程序执行到MOV 30H, #X时, 71H、72H和SP单元中内容是什么?

```
ORG 0900H
MOV SP, #70H
LCALL SQR
SJMP $
SQR: MOV 30H, #X
RET
END
```

(71H) = 断点地址低8位
(72H) = 断点地址高8位
SP = 72H

四、指出程序运行结果(1-10)

4-10 如下程序执行后会产生什么效果？为什么？

```
ORG 0A00H
MOV DPTR, #2000H
MOV R0, #20H
LOOP: MOV R2, #0AH
      MOVX A, @DPTR
      MOV @R0, A
      INC DPTR
      INC R0
      DJNZ R2, LOOP
      SJMP $
      END
```

该程序执行后会进入死循环，因为R2中内容始终不会为零。

5-3题 答案

通常有4部分组成:

- 1) 地址寄存器和译码器
对地址线上地址译码, 产生字选择线, 选中某存储单元工作
- 2) 存储阵列
由基本存储电路组成存储单元, 存储单元组成存储阵列
- 3) 三态输出缓冲器
8 位寄存器, 暂存从存储阵列中读出或写入的信号
- 4) 控制电路
控制存储器的读/写过程
若地址线有N条, 则字线= 2^N 条

5-4题 答案

单译码编址存储器的地址码存放在字地址寄存器中, 经字译码器译码产生字选择线, 选中某存储单元工作。

双译码编址存储器的地址线被分成X和Y两组, X组地址线上地址码经X地址译码器译码后在X地选线上产生X 向选择电平, Y组地址码经Y 译码器译码后在Y地选线上产生Y 向选择电平, X和Y地选线共同选中的存储单元才是所要选中的存储单元

双译码编址存储器所需地址选择线条数比单译码的少很多, 故它的存储容量可以做得很大。

5-5题 答案

- 512*4 地址线9条, 数据线4条
1K*4 地址线10条, 数据线4条
2K*8 地址线11条, 数据线8条
4K*1 地址线12条, 数据线1条
4K*4 地址线12条, 数据线4条
16K*1 地址线14条, 数据线1条
32K*4 地址线15条, 数据线4条
64KB 地址线16条, 数据线8条
128K*8 地址线17条, 数据线8条
512KB 地址线19条, 数据线8条
4MB 地址线22条, 数据线8条

5-6题 答案

对双译码编址方式

假设 X 向地址线8条

Y 向地址线8条

则有 X 地选线= $2^8=256$ 条

Y 地选线= $2^8=256$ 条

共需地选线= $2*256$
= 512 条

所需位线=8条

每条位线应和65536个
基本存储电路相连。

对单译码编址方式

所需字线= $2^{16}=65536$ 条

5-7题 答案

掩膜ROM

有管：表示存"0"

无管：表示存"1"

适合批量生产

熔丝PROM

熔丝通：存"0"

熔丝断：存"1"

适合批量生产

UVEPROM

浮置栅内有电荷：存"0"

浮置栅内无电荷：存"1"

适合小批量/研制阶段。

5-9题 答案

读出一正常模式下的读出

校验—编程模式下的读出

引脚信号（编程时）：

$\overline{\text{PGM}}$ —编程负脉冲

V_{pp} —+21V

$\overline{\text{CE}}$ —TTL低电平

$\overline{\text{OE}}$ —TTL低电平

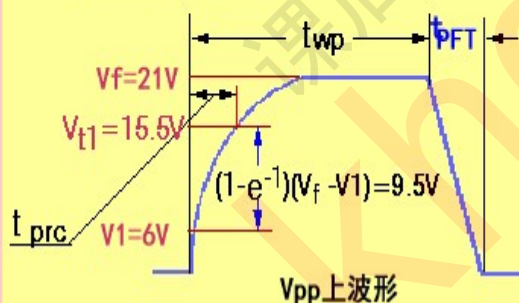
5-10题 答案

2764—紫外光擦除（20-30分钟）

2815—片擦除：一个脉冲擦除1片

字节擦除：一个脉冲擦除1字节

V_{pp} 波形如下图所示。



5-11题 答案

SRAM：基本存储电路是触发器。

DRAM：基本存储电路是存储电容

C_g —无电荷，存"0"

C_g —有电荷，存"1"

SRAM的写入过程（略）

读出过程（略）

DRAM的写入过程（略）

读出过程（略）

5-12题 答案

62128的存储容量是16KB

基本存储电路个数：

$$16 \times 1024 \times 8 = 131072$$

所需MOS管：

$$131072 \times 6 = 786432 \text{只}$$

5-13题 答案

读出过程（略）

存储单元个数：

$$16K \times 1 = 16384$$

基本存储电路个数：

$$16K \times 1 = 16384$$

5-14题 答案

2114的存储容量是 $1K \times 4$,
故需2块芯片才能为CPU提供1KB主存储器。因此,
32KB容量的主存需要2114
 $= 32 \times 2 = 64$ 块
若采用2116($16K \times 1$), 则至少
要8块2116才能组成16KB
主存, 以供CPU使用。因此
32KB内存共需2116
 $= 2 \times 8 = 16$ 块

5-15题 答案

CPU和存储器工作时
存储器工作速度↓
CPU等待时间↑
∴ 存储器速度应和CPU速度
度相同步。
CPU分给存储器地址不同,
CPU和外部存储器的连接
也不同。

5-16题 答案

CPU和存储器连接时, 有如下
三种译码方式:

- 1、全译码(需译码器多)
所有地址是唯一的, 地址
无重叠。
- 2、部分译码(译码器少)
当有片选地址线悬空时,
地址会有重叠。
- 3、线选法(无译码器)
当有片选地址线悬空时,
地址会有重叠。

5-17题 答案

SRAM (NMOS)

功耗最大

因基本电路是触发器

SRAM (CMOS)

功耗低

因为是CMOS元件

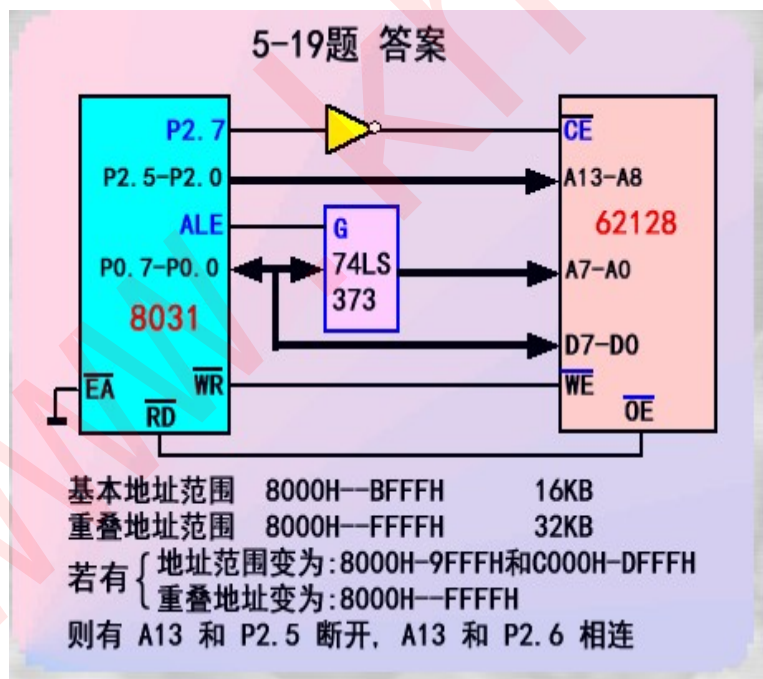
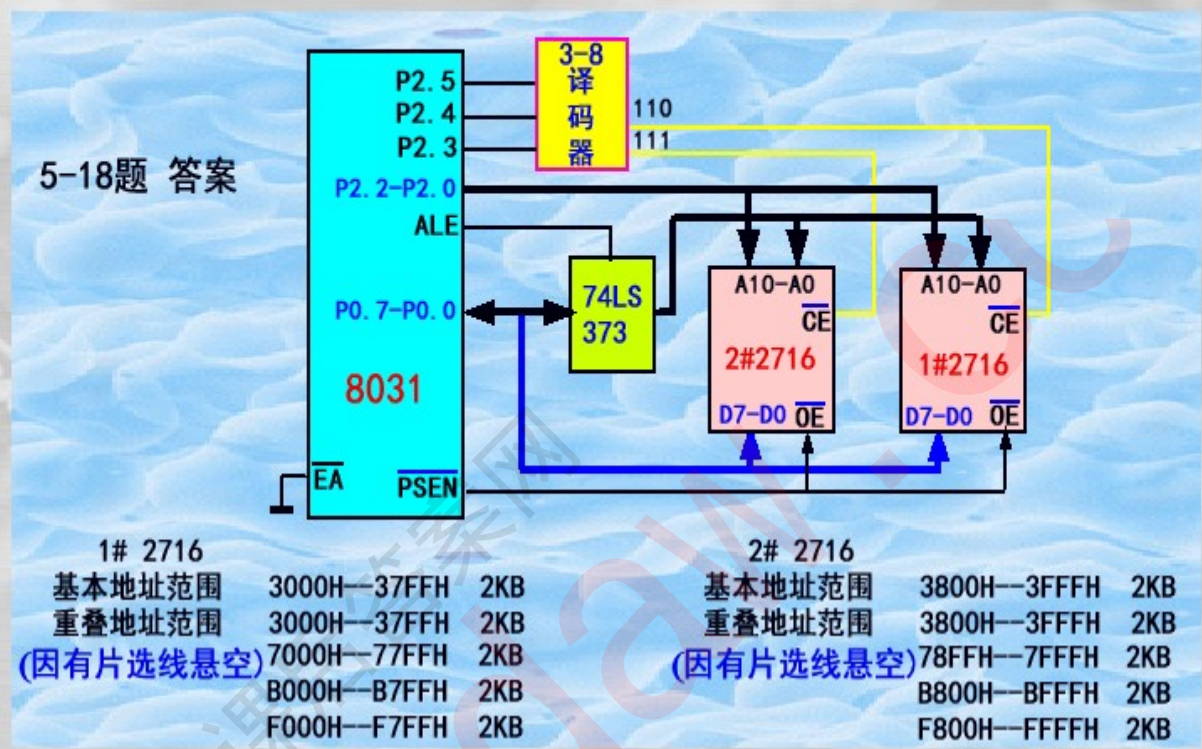
DRAM (NMOS)

功耗最低

因为靠电容存储信息

二、问答题2(18-20题)

5-18 试画出8031和2716的连线图，要求用三-八译码器，8031的P2.5, P2.4和P2.3参加译码，基本地址范围为3000H-3FFFH，该2716有没有重叠地址？根据是什么？若有，则写出每片2716的重叠地址范围。



5-20题 答案

1#2114的基本地址范围:

F000H—F3FFH (1KB)

2#2114的基本地址范围:

F400H—F7FFH (1KB)

装满64KB共需2114

2*64=128块

2-4译码器=16块

三、填空题(1-17题)

- 5-1 存储容量为1MB的DRAM存储器有 1048576 个存储单元, 每个存储单元有 8 位。需要二进制 20 位对它进行编码。
- 5-2 半导体存储器将沿着 提高集成度、提高存取速度 和 降低工作电压 等三个方面高速发展。
- 5-3 半导体存储器内部有两种基本结构: 一是 单译码编址; 二是 双译码编址。其中, IBM-PII系统机中所用DRAM存储器通常属于 双译码编址 的存储器。
- 5-4 一个64KB的单译码编址存储器, 其内部字选择线有 65536 条, 一个同样存储容量的双译码编址存储器(X向地址线有8条), 其内部地址选择线只有 256X2 条。
- 5-5 掩膜ROM存储器存储一位二进制信息的原理是: 有管表示存“0”和无管表示存“1”(或者相反); 熔丝式PROM存储器存储一位二进制信息的原理是: 熔丝通表示存“0”和熔丝断表示存“1”(或者相反); UVEPROM依靠FAMOS管来存储一位二进制信息, 浮置栅中无电荷表示存 1。
- 5-6 ROM存储器正常工作时只能读不能写, 停电后再加电信息不丢失, 故又称为 程序存储器; RAM存储器正常工作时既能读又能写, 停电后再加电其中信息会丢失, 故用来存放实时数据。 FLASH 是一种新型存储器, 正常工作时既能读又能写, 停电后再加电信息不丢失(保留最后一次写入的信息), 故它可以在U盘中使用。
- 5-7 2764在编程时PGM引脚应接 50ms宽的负脉冲, V_{pp} 应接 21V; 读出校验时OE要接 负脉冲。
- 5-8 六管静态RAM的“基本存储电路”是 触发器, 动态RAM的“基本存储电路”有一个存储电容 C_g 和一只MOS管组成。如果一片容量为64KX8的动态RAM的存储阵列至少需要 $64K \times 8 = 524288$ 只MOS管, 那么同样容量的静态RAM的存储阵列至少需要 3145728 只MOS管。

三、填充题(9-17题)

- 5-9 动态RAM的缺点是需要刷新，静态RAM的优点是工作可靠和集成度高。
- 5-10 2186全集成化DRAM采用内部电路进行刷新时，RDY/REFEN变为输出状态线。若RDY/REFEN=0电平，则表示2186正在刷新，MCS-51不能对它进行读写。
- 5-11 在MCS-51对存储器进行硬件连接时，ROM的容量是根据用户程序长短决定的，用户必须从地址和字长两方面满足MCS-51的要求，RAM的容量取决于实时数据的数量。
- 5-12 MCS-51有16条地址线(P2口8条和P0口8条)，故它对外部存储器的寻址范围是0000H~FFFFH。MCS-51通过PSEN引脚来控制外部ROM存储器是否从地址总线上接收地址；通过RD和WR引脚来选通外部RAM工作。
- 5-13 MCS-51地址总线既要同外部ROM/RAM相连，又要同所有外部设备相接。但MCS-51在某一瞬时送到地址总线上的地址是唯一的，要么是外部ROM的地址，要么是外部RAM的地址，要么是外部设备的地址。在MCS-51和外部存储器的连接中，为了弄清每片存储器的基本地址范围和重叠地址范围，根据MCS-51高位地址的连接方式，共有3种地址译码方法：一是全译码；二是部分译码；三是线选法。
- 5-14 如果说MCS-51的片选地址线是用来选中存储系统中哪些芯片工作，那么片内地址线选中的是被选存储芯片中的地址单元。
- 5-15 在MCS-51和外部存储器的连接中，基本地址范围是指：CPU悬空片选线为0时的地址范围；重叠地址范围是指：CPU的悬空片选线从全0变到全1时的地址范围。
- 5-16 在采用线选法的存储器系统中，如果MCS-51有n条片选地址线悬空，则一个实际的存储单元就会对应 2^n 个CPU地址（即：存储单元的CPU地址不是唯一的）。
- 5-17 在图5-25中，2[#]插座和3[#]插座上可以插2764只读存储器是因为PSEN参加了选通；2[#]和3[#]插座上可以插6264 RAM存储器是因为RD和WR参加了选通。

四、选择题(1-6)

- 5-1 如下 D 是外存储器。
A、光盘驱动器 B、硬盘驱动器
C、软盘驱动器 D、光盘
- 5-2 以下存储器中，读写速度最快的是 A。
A、RAM存储器 B、光盘存储器
C、软盘存储器 D、硬盘存储器
- 5-3 在下面四种类型的存储器芯片中，A 最可靠；D 可用来在线编程。
A、掩膜ROM B、熔丝式PROM
C、UVEPROM D、FLASH存储器
- 5-4 以下四种UVEPROM中，D 容量最大。
A、2764 B、27128
C、27256 D、27512
- 5-5 综合起来，6264有以下四种工作方式。B 方式下功耗最大；D 方式下功耗最小。
A、禁止 B、读/写
C、选通 D、未选通
- 5-6 以下三种地址译码方式中，A 方式可以使MCS-51选中的存储单元的
地址是唯一的。
A、全译码 B、部分译码
C、线选法

6-1题 答案

定义：CPU暂时中止主程序执行转而为外设服务，服务完后又返回主程序执行的过程。

分类：按中断的功能分为

- 1) 可屏蔽中断
- 2) 非屏蔽中断
- 3) 软件中断。

好处：1) CPU的效率高
2) 提高实时数据的处理时效
3) 提高故障处理的时效。

6-2题 答案

中断源：引起中断的原因

- 种类：1) 外部设备中断源
用于CPU和外设间数据传送
2) 控制对象中断源
用于CPU对控制对象的监控
3) 故障源中断源
用于CPU对故障的处理。
4) 定时脉冲源中断源
用于CPU的定时控制。

6-3题 答案

定义：高优先级中断嵌入低优先级中断的过程。

中断系统：能实现中断功能的那部分电路。

功能：

- 1) 响应中断(关中断、断点地址入栈、转入中服和撤除中断等)
- 2) 中断优先权排队(先响应高优先级中断, 后响应低优先级中断)
- 3) 实现中断优先级嵌套(高优先级中断能自动嵌入低优先级中断)
- 4) 中断返回(断点地址恢复到PC)。

6-4题 答案

INT0中断——IE0, 位地址89H, 因有中断而置位, 因中断被响应而复位。

INT1中断——IE1, 位地址8BH, 因有中断而置位, 因中断被响应而复位。

T0/T1中断——TF0/TF1, 位地址为8DH/8FH, 因中断而置位, 因响应而复位

串口中断——TI/RI, 位地址99H/98H, 因有中断而置位, CPU在中断响应后进入中断服务程序, 在中断服务程序中由软件复位。

6-5题 答案

定义： EX0——INT0中断允许位
ET0——T0中断允许位
EX1——INT1中断允许位
ET1——T1中断允许位
ES——串口中断允许位
EA——中断总控位

允许T1中断的指令为：

MOV IE, #88H

采用位操作指令为：

SETB EA

SETB ET1

6-6题 答案

相应初始化程序如下：

```
SETB EA
SETB EX0
SETB EX1
MOV IP, #05H
...
```

若INT0和INT1引脚同时有中断请求输入, 则MCS-51先响应INT0上中断请求。这是因为INT0上固有的中断优先级比INT1的要高。

6-7题 答案

MCS-51响应中断有三种情况：

- 1) 若CPU正执行主程序, 且现行中断开放, 则CPU现行指令执行完后响应中断。
- 2) CPU正响应某个中断, 若又有更高优先权中断进来, 则CPU响应之。
- 3) 若CPU正执行如下指令中的一条

RETI

访问 IE/IP 指令

则CPU需等到该类指令的下指令执行完后才响应这个中断。

6-8题 答案

中断入口地址如下:

INT0—0003H

T0—000BH

INT1—0013H

T1—001BH

串行口—0023H

响应中断的最短时间为:

3个机器周期(48T)。

6-9题 答案

自动撤除的中断有:

INT0 边沿触发方式

INT1 边沿触发方式

T0 溢出中断

T1 溢出中断

需用户撤除的中断有:

串行口中断, 用如下指令撤除。

CLR TI

CLR RI

INT0/INT1电平触发中断, 采用软件和硬件方法撤除。

6-10题 答案

中断初始化程序如下:

SETB EA ;开所有中断

SETB EX0 ;开INT0中断

SETB IT0 ;令INT0为边沿触发

6-11题 答案

IRR(中断请求寄存器)

若IR_n上有中断请求, 则IRR_n=1 (IRR_n在8259 因响应CPU中断而收到第3个 INTA负脉冲后复位)。

IMR(中断屏蔽寄存器)

若软件使IMR_n=1, 则IRR_n上中断被屏蔽。

ISR(现行服务寄存器)

若IR_n上中断被响应, 则相应ISR_n=1。

PR(优先级分析器)

把已被CPU响应中断(由ISR指示)同现行中断(由IRR指示, 未被屏蔽)进行优先级比较。若现行中断比ISR所指中断的优先级高, 则8259就响应, 并在收到CPU第3个 INTA脉冲后令ISR_n=1 (ISR_n在中断服务程序中由软件复位)。

6-12题 答案

当8259的ISR0=1和ISR3=1时, 说明CPU正响应IR0上来的中断请求, 因为8259内部对IR_n上来的中断优先级的排序是:

高—————>低
IR0 IR1 IR2 IR3 IR4 IR5 IR6 IR7

6-14题 答案

8259的选口线是: CS和A0

CAS2~CAS0作用是:

若8259为主片

则CAS2~CAS0为输出线
(选中某从片工作)。

若8259为从片

则CAS2~CAS0为输入线
(接收主片CAS_n上信号)。

6-15题 答案

一片8259可接八个中断源, 每个中断源都需要一个中断入口地址, 该中断入口地址的高10位(间址8)/高11位(间址4)是由CPU通过ICW2和ICW1在初始化时送给8259的, 而ICW1中的低6位(间址8)/低5位(间址4)由8259在响应中断时自动形成。8259在收到CPU的三个INTA负脉冲后自动把中断入口(矢量)地址送给CPU, 以便CPU据此而进入相应中断服务程序执行。

ICW2—中断矢量地址高8位

ICW1—高2位/3位为中断矢量低字节的高2位/3位。

6-17题 答案

由ICW4格式可知, 如下情况需要ICW4:

- 1) 8086/8088系统
- 2) 特殊全嵌套模式
- 3) 缓冲方式
- 4) 自动结束中断方式

6-18题 答案

OCW2所含命令字有:

指定EOI命令 } 作用: 令ISRn=0
非指定EOI命令 }

EOI时自动轮转(置位)命令

EOI时自动轮转(复位)命令

作用: 1) 令ISRn=0

2) IRn为最低优先级, 其它轮转升级。

EOI时指定轮转命令

作用: 1) 指定ISRn=0

2) IRn为最低优先级, 其它轮转升级。

优先级置位命令

作用: 指定中断的优先级变为最低。

OCW3所含命令字有:

中断屏蔽字

查票模式字

状态读取模式字

} 用于优先级轮转模

6-19题 答案

8259初始化后, 不写任何OCW命令字就可进入全嵌套中断模式。当8259因响应IRn上中断而进入全嵌套中断模式后相应ISRn=1, 故在其中断服务程序末尾必须采用指定/非指定EOI命令, 使得相应ISRn=0

6-20题 答案

使8259进入中断优先级轮转模式的OCW命令有如下5条:

EOI时自动轮转(置位)命令

EOI时自动轮转(复位)命令

EOI时指定轮转命令

EOI时非指定轮转命令

优先级置位命令。

6-22题 答案

五片8259级联最多可以构成36级中断系统

6-21题 答案

CPU对8259查票的目的是弄清 IR_n 上是否有中断请求。

对8259的查票过程如下：

- (1) 查票命令字 $\rightarrow$ 8259 (要求查票)
- (2) 产生如下格式的查票字



- (3) 如下指令可读得查票字

```
MOV R0, #PT59A
MOVX A, @R0
```

6-23题 答案

MCS-51扩展外部中断源的方法有三种：

- 1) 借用T0/T1扩展外部中断源
扩展外部中断源的个数只有2个
- 2) 用查询法扩展外部中断源
扩展外部中断源个数多时, 查询时间太长。
- 3) 采用8259扩展外部中断源
响应快, 1片可扩展8个外中断源

6-24题 答案

T1作为外中断源的初始化程序为：

```
MOV TMOD, #60H ; 令T1为方式2
MOV TL1, #0FFH ; } FFFFH  $\rightarrow$  T1
MOV TH1, #0FFH ; }
SETB EA ; 开所有中断
SETB ET1 ; 开T1中断
SETB TR1 ; 启T1工作
```

6-25题 答案

查询法外扩中断源的优缺点：

- 1) 无需硬件, 程序简单
- 2) 外扩中断个数多时, 查询程序太长, 中断响应速度太慢。

用8259外扩中断源的优缺点：

- 1) 属于硬件外扩中断源
- 2) 中断响应速度快。

6-26题 答案

8259是为8086/8088系统设计。因为8259A的八个外中断源需要8个中断入口地址, 故CPU响应从8259A来的中断请求时须从8259提取一条CALL XXXX 指令。即：CPU响应8259A发来的中断请求时只有连续执行三条MOVX A, @Ri 指令才能在INTA上产生三个负脉冲, 以便能从8259A获取CALL指令的操作码和中断矢量地址。

三、填空题(1-15题)

- 6-1 中断的定义是暂时停止主程序执行而转为外部设备服务,服务完后又自动返回原程序执行的过程。这句话的含义有三点:
(1) CPU暂停主程序执行是由中断源的中断请求引起的,中断请求是随机的。
(2) CPU应能自动找到引起中断的中断源的中断服务程序始址。
(3) 执行完中断服务程序后要能自动返回主程序执行。
- 6-2 中断是计算机发展史上的一个里程碑。现代计算机都具有中断功能是因为中断有个作用:一是能提高 CPU的工作效率;二是能提高 CPU对故障处理的时效。
- 6-3 MCS-51有五级中断。其中,外部中断有 INT0 和 INT1;内部中断有 T0、T1、和 串行口 等。
- 6-4 INT0和INT1有两种触发中断的方式:一是 负边沿 触发中断;二是 低电平 触发中断。T0和T1中断是在 全“1”变为全“0”溢出 时触发中断的;串行口中断是在串行口接收到/发送完一个字符后产生的。
- 6-5 MCS-51响应中断时要做三件事。这三件事:一是 断点地址入栈;二是 关中断;三是按 中断矢量 转入相应中断服务程序执行。
- 6-6 MCS-51响应中断的最长时间是 8 个机器周期;最短时间是 3 个机器周期。
- 6-7 若8259A的SP接+5V,则该8259A被设定为主片,其CAS2~CAS0为代码 输出 线。若8259A的SP接地,则CAS2~CAS0变为代码 输入 线。
- 6-8 在8259A内部IRRn用于记录IRn(n的取值是0~7)上哪些外设中断请求;ISRn用于记录IRn上哪些中断请求已被CPU响应;IMRn用于控制IRn上哪些中断需要被屏蔽。若某8259A的IR5上中断已被CPU响应,现在IR0上又来了一个中断。此时,IRRn中 IRRO 被置位,ISRn中 ISR5 被置位。

三、填空题(9-15题)

- 6-9 当8259A的IRn上某个中断被CPU响应后,该8259A通过INTA引脚上接收CPU送来的三个负脉冲就能把一条 CALL XXXXH 指令(机器码)送给CPU,以便转入相应中断服务程序执行。
- 6-10 8259A有ICW1、ICW2、ICW3和ICW4等四个初始化命令字,用于对8259A初始化。其中, ICW1 和 ICW2 用于8259A在响应中断时给CPU反送一个中断矢量字。
- 6-11 一片主8259A和三片从8259A可以把MCS-51的中断扩展为 29 级。
- 6-12 已知:8259A的IR0上中断嵌套在IR5的中断中,且CPU正在IR0的中断服务程序中执行。此时,ISR0= 1 和 ISR5= 1。
- 6-13 借用MCS-51的T0/T1来扩展外部中断源的方法十分简单,但要求T0/T1必须工作在计数器模式下的 Mode 2 方式。
- 6-14 采用查询法扩展外部中断源方法的缺点是 中断源个数多时查询时间太长。
- 6-15 在图6-20中,8259A的INTA负脉冲是由MCS-51执行 MOVX A, @DPTR 指令产生的。

四、选择题(1-10)

- 6-1 如下四种中断源中，MCS-51的T0/T1溢出中断属于 D 中断源；串行口中断属于 A 中断源。
- A、外部设备
B、控制对象
C、故障对象
D、定时脉冲
- 6-2 中断按中断功能可以分为如下三类。MCS-51的五级中断属于 A 中断。
- A、可屏蔽
B、非屏蔽
C、软件
- 6-3 MCS-51有5级中断和6个中断标志，它们分布在TCON和SCON中。其中，串行口中断标志是 C。
- A、IE0和IE1
B、TF0和TF1
C、TI和RI
- 6-4 如下指令执行后，
- SETB EA
SETB ET0
- MCS-51便能自动响应 A 中断。
- A、INT0
B、T0
C、INT1
D、T1
- 6-5 如下指令执行后，B 中断的优先级最高；C 中断的优先级最低。
- SETB PT1
SETB PT0
- A、INT0
B、T0
C、INT1
D、T1

四、选择题 (6-10)

- 6-6 T1中断的入口地址是 D。
A、00023H B、000BH
C、0013H D、001BH
- 6-7 以下中断标志中， C 需要软件撤除。
A、IE0/IE1 B、TF0/TF1
C、TI/RI
- 6-8 CPU把 C 初始化命令字送给8259A后，就可决定8259A的工作方式。
A、ICW1 B、ICW2
C、ICW3
- 6-9 如下操作命令字中， C 可用来设定8259A的中断模式。
A、OCW1 B、OCW2
C、OCW3
- 6-10 借用T0来扩展外部中断源时，该外部中断请求信号应由 B 引脚输入。
A、 $\overline{\text{INT0}}$ B、T0
C、 $\overline{\text{INT1}}$ D、T1