

2008 试题

一、填空题

1. ASCII 码是用 8 位二进制数表示一个字符, 字符 '0' 的 ASCII 码用十六进制数表示为 30、字符 'A' 的 ASCII 码用十进制数表示为 65d、ASCII 码 00001101B 表示回车 ASCII 码的最低位称为 D0 位, 则所有 ASCII 码的 D7 位都为 0。

2. 设 N 为有符号数, 及指令 MOV AL, N。N 的取值范围用十进制数表示为 -128~127; 若 N=-1, 则指令执行后, AL 中存储的二进制数为 11111111; 若指令执行后, AL 中存储的二进制数为 10000000B, 则 N=-128。

3. 8086 系统工作在最小模式下时, 需要 3 片 8282 地址锁存器 (每片含有 8 个 D 锁存器), 8086 用于锁存地址的控制信号为 ALE, 需要锁存的信号为三类: 地址、BHE 和 。

4. 设变量定义伪指令中

```
DATA SEGMENT
X1    DB  +1, 0, -1,
X2    DW  12H, X1
COUNT EQU  $-X1
DATA ENDS
```

该 DATA 段总共为 7 字节, 位于奇存储体中的最后 1 个数据为 00H,
\$=0007H。若 8086 执行:

```
MOV  AX, DATA
MOV  DS, AX
MOV  AX, [2]
MOV  BX, X2
```

则 AX=0000H, BX=0003H, COUNT=07H。

5. 设有如下指令序列:

```
1234H: 190H    NEXT1: .....
                                MOV  AL, 0A0H
                                CMP  AL, 50H
1234H: 200H    JG    NEXT1
1234H: 202H    NEXT2: .....
```

当指令 CMP AL, 50H 执行完时, AL 寄存器的值为 0A0H, 标志寄存器 FR 的 OF 位为 1, SF 位为 0, ZF 位为 0, CF 位为 0。

指令 JG NEXT1 的代码为 2 字节代码, 指令代码的第 2 字节为 8E, NEXT1 的逻辑地址是 1234H:190H, 当 JG NEXT1 指令执行完后,

下一条应执行的指令的物理地址是 12542H。

6. 8086 CPU 执行指令的部件称为 EU，指令队列最少能放下 1 条指令。输入输出指令必须使用的 8 位寄存器为 IP。一般情况下，查询输入方式要占用 2 个端口地址，要先检查外设的 状态 信息，再输入数据。此时 $\overline{M/\overline{IO}}$ 信号为 低 电平， \overline{RD} 信号为 低 电平。

7. 在可屏蔽中断、非屏蔽中断和指令中断之中，8086 在复位后处于关闭状态的中断是指令中断，与外部触发信号无关的中断是 指令中断，优先级最高的是 指令中断，非屏蔽中断 中断需要 8086 输出中断响应信号，该中断响应信号为 1 个总线周期的低电平。设 8259A 工作在完全嵌套方式，若中断服务寄存器 $ISR=00000101B$ ，则 IR2 引脚对应的中断服务被打断进入中断嵌套，转入执行 IR0 引脚的中断服务程序。

8. RS-232-C 是（异步还是同步）异步 串行通信标准，若采用全双工方式传送数据时，最少需要 2 条线，其中 TxD 引脚发送数据，RxD 引脚接收数据。若发送与接收方计算机约定采用相同的校验方式，当发送计算机发送字符 ‘A’ 时，接收计算机收到的字符却为 ‘C’（未发生帧出错），这时会发生 溢出 出错。MODEM 的中文名称为 调制解调器，在串行发送数据时其作用是 数字→频率，在串行接收数据时其作用是 频率→数字。

二、判断题

1. （ n ） 中断向量表中存放的是中断服务程序。 地址
2. （ y ） 端口地址为奇地址的外设通过数据总线高 8 位与 8086 CPU 进行数据传送。
3. （ n ） IP 寄存器中存放的是当前正在执行的指令的地址。 T1
4. （ n ） I/O 指令寻址方式是指 IO 地址与存储器地址统一编址。
5. （ y ） 8086 系统中每个存储单元都有一个唯一的物理地址。
6. （ y ） 当产生可屏蔽中断时，中断类型号由硬件系统提供。
7. （ n ） 执行 MOVSB 指令后，DI 和 SI 都加 1。Df 控制
8. （ n ） AD 转换器的分辨率是指模拟量和转换后数字量之间的误差。
9. （ n ） 可执行程序的扩展名是 OBJ。
10. （ n ） 8086 指令机器码一般由操作码和操作数构成。

三、简答题

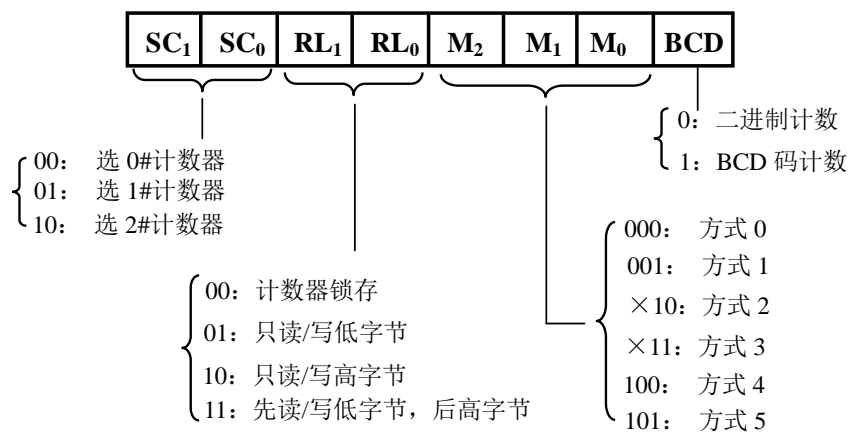
1. 设 $(BX) = 100H$ ， $(DI) = 200H$ ，写出下列指令中源操作数的寻址方式，以及完成该指令需用的总线周期数。

① MOV AL, 12H

② ADD [BX+DI], AX

③ MOV AX, [BX+1H]

2. 设有 2 片 8259A 接成主从级联方式，主片设为特殊完全嵌套方式，从片设为完全嵌套方式，从片的 INT 引脚接至主片的 IR4 引脚，写出主、从片 8259A 中断请求引脚的优先级顺序。若主片、从片均为完全嵌套方式，从片 IR5 对应的中断服务程序正在执行，从片 IR3 管脚的中断请求是否能打断 IR5 中断服务的执行？
3. 设某片 8253 的片选地址为 280H—287H，8253 的 A1、A0 引脚分别与 8086 CPU 的 A1、A0 相连，0#计数器的输入时钟频率为 1MHz，工作于 10 分频方式（速率发生器方式），其输出端 OUT0 与 1#计数器的 CLK1 端相连，要求 1#计数器的 OUT1 端输出周期为 100ms 的方波，写出相应的初始化 8253 的指令序列，并画出 OUT1 端输出波形的示意图。附 8253 控制字格式：



4. 编写一段指令序列，定义一个具有 200H 个字节的堆栈段，并将其内容全部初始化为 0。若 AX=1234H，画出执行 PUSH AX 指令前、后堆栈段的存储状况示意图。（给出堆栈指针及堆栈内容的变化情况。）
5. 设有如下数据段定义及段内数组求和子程序，希望该子程序能完成对数据段中定义的 100 个字数据的求和，不考虑溢出。该子程序中存在若干处错误，请予以改正。

```

DATA    SEGMENT
D1      DB      100    DUP ( ? )
DATA    ENDS
SUM     PROC
        PUSH    AX
        PUSH    BX
        MOV     DS, DATA
        MOV     BX, D1

```

```

MOV    CX, 100
MOV    AX, 0
ADD:   ADD    AL, [BX]
INC     BX
LOOP   ADD
POP     AX
POP     BX
RET
SUM    END

```

四、

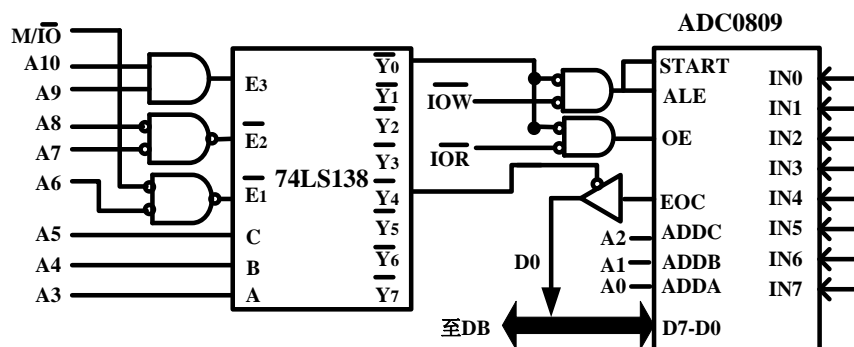
从键盘输入 2 个数字字符（非数字字符不接收），将其转换成压缩 BCD 码的形式存在数据段中，先输入的数字作为高位，后输入的数字作为低位。

五、硬件编程题

设电路如图所示（见下页）。

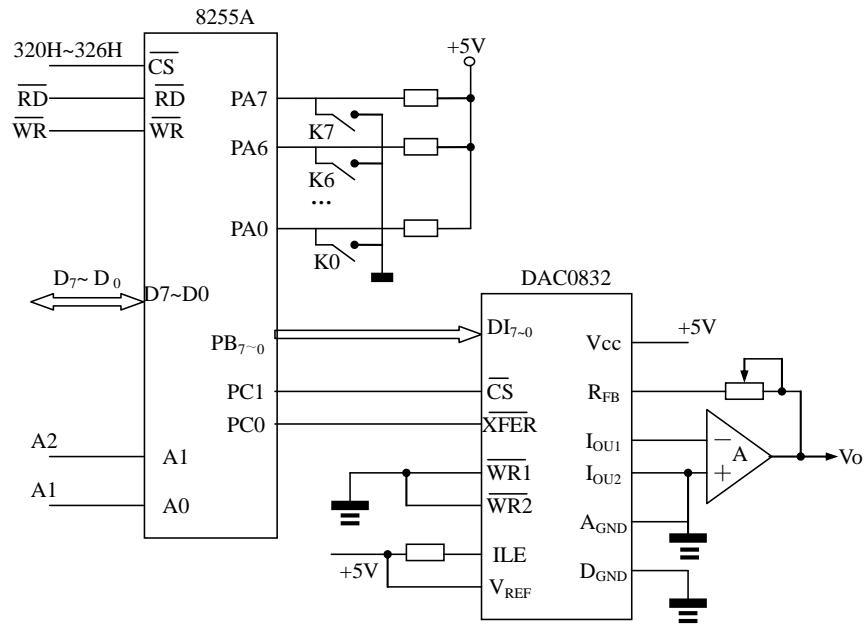
1) 在图中标出 $\overline{Y_0}$ 、 $\overline{Y_4}$ 对应的地址范围（设未用到的高位地址为 0）（3 分）；

2) 完成汇编程序的编写：要求从 ADC0809 的 8 个通道分别采集 1 个数据后顺序存放在数据段偏移地址为 STRBUF 的缓冲区中。当某个通道采集的数字量大于 7FH 时，在屏幕显示 “The Voltage is too High!”。对 8 个通道采集 256 次数据后退出现程序（12 分）。

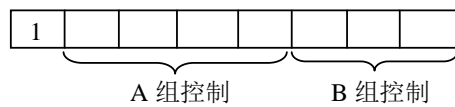


六、硬件编程题..... (10 分)

电路如图所示，写出程序，完成 8255A 初始化，并循环完成如下操作：从 8255A 的 PA 口读入开关状态，将读入的数据直接输出到 DAC0832 进行 DA 转换。其中 DAC0832 需要设置为直通工作方式。



附表：8255A 工作方式控制字格式



$D_6D_5 = \begin{cases} 00 & \text{A 组工作在方式 0} \\ 01 & \text{A 组工作在方式 1} \\ 1X & \text{A 组工作在方式 2} \end{cases}$	$D_2 = \begin{cases} 0 & \text{B 组工作在方式 0} \\ 1 & \text{B 组工作在方式 1} \end{cases}$
$D_4 = \begin{cases} 0 & \text{PA 口为输出} \\ 1 & \text{PA 口为输入} \end{cases}$	$D_1 = \begin{cases} 0 & \text{PB 口为输出} \\ 1 & \text{PB 口为输入} \end{cases}$
$D_3 = \begin{cases} 0 & \text{PC 口高 4 位为输出} \\ 1 & \text{PC 口高 4 位为输入} \end{cases}$	$D_0 = \begin{cases} 0 & \text{PC 口低 4 位为输出} \\ 1 & \text{PC 口低 4 位为输入} \end{cases}$