

上海大学1998 年攻读 硕 士学位研究生

入学考试试题

通信与电子系统 无线电电子学

信号与信息处理 生物医学工程

招生专业 考试课目 微型计算机原理及应用

一. 填空题 (每空1分 共26分)

1. 8088有 20 根地址线, 对内存寻址的物理地址范围是 00000H~FFFFFH, 对I/O端口寻址范围是 0000H~0000H 个。
2. 用8位二进制码表示的补码, 其整数范围中最大值是 7 H, 最小值是 -8 H。
3. 8088中 $(DS)=6000H$, 则数据段的物理地址中, 首地址是 60000H 末地址是 6000FH。
4. 在PC微机系统中, 调试运行一个完整的8088汇编语言程序至少要用到下列工具文件: (1) LINK (2) DEBUG (3) ASM (4) OBJ。
5. 在PC微机系统中, $INT\ 21H$ 是 系统调用。它的中断程序入口地址存放在地址为 00000H 的中断向量表中, 共占用 4 个字节内存单元。
6. CPU经I/O口与外设进行信息交换的数据传送方式有以下四种: 并行传送, 串行传送, 直接存储器访问, 间接存储器访问。
7. 异步串行传送ASCII码 'A', 采用奇校验, 停止位为1位。串行口发送一帧信息为 10 位。写出该信息位数据格式 (低) (高)。
8. RS-232C串行通信接口标准规定, 逻辑"1"的电平范围为 -3V~-15V, 逻辑"0"的电平范围为 +3V~+15V。因此, RS-232C接口信号和TTL电平信号匹配必须通过 缓冲器 和 驱动器 芯片来实现。
9. 8088系统中各种中断的优先次序为(高) 0, 1, 2, 3, 4, 5, 6, 7 (低)。
- 80386系统中各种中断的优先次序为(高) 0, 1, 2, 3, 4, 5, 6, 7 (低)。(两空任选一空填入相应内容)

若两空皆填,则第一个空的内容有效)

10. 8088系统上电复位后, CPU执行的第一条指令首字节存放在地址为_____的存储单元内。8031系统上电复位后, CPU执行的第一条指令, 首字节存放在地址为_____的程序存储器单元中。(答题要求同第9题)

二. 程序分析 (每一题中, 可在a, b两小题中任选一题解答。若两题都答, a题的答案有效) (每空两分, 共20分)

1. a) 8086/8088 程序

```
DATA SEGMENT
    X    DW  9F4B
    ONE  DW  ?
DATA    ENDS
CODE    SEGMENT
START   PROC FAR
    ASSUME CS:CODE, DS:DATA
BEGIN:  PUSH  DS
        MOV   AX, 0
        PUSH  AX
        MOV   AX, DATA
        MOV   DS, AX
        MOV   CX, 10H
        MOV   BX, 0
        MOV   DX, 1
        MOV   AX, X
        OR    AH, AL
        JZ    DONE
        FIND: MOV   AX, X
                AND   AX, DX
                JZ    NEXT
                INC   BX
```

```
        NEXT: ROL   DX, 1
                LOOP  FIND
        DONE: MOV   ONE, BX
                RET
        START   ENDP
        CODE    ENDS
        END     BEGIN
```

执行结果:

(AX) = _____

(CX) = _____

(DX) = _____

(DS:0000H) = _____

(DS:0002H) = _____

b) MCS-51 程序

```

        ORG 0200H
        MOV DPTR, #DAT
        MOV R0, #0
        MOV R1, #1
        MOV R2, #2
        CLR A
        MOVC A, @A+DPTR
        ORL A, R3
        JZ HERE
        MOV DPTR, #DAT
Loop1:  MOV R3, #8
Loop2:  CLR A
        MOVC A, @A+DPTR
        ANL A, R1
        MOV R4, A
        JZ NEXT
        INC R0
NEXT:   MOV A, R1
        RL A
        MOV R1, A
        DJNZ R3, Loop2

```

```

        INC DPTR
        DJNZ R2, Loop1
        MOV A, R0
HERE:   STMP HERE
        ORG 1000H
        DAT DB 4BH, 9FH
        END

```

程序运行后

```

(A) = _____ H
(R1) = _____ H
(R3) = _____ H
(R4) = _____ H
(1000H) = _____ H

```

2. a) 8086/8088 程序

```

DATAAREA SEGMENT
    BLOCK DB 04H, 21H, 95H, 79H, 32H, 0FFH, 85H, 53H
    COUNT EQU $-BLOCK
DATAAREA ENDS

STACK SEGMENT PARA STACK 'STACK'
    DB 100 DUP(?)
STACK ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATAAREA, SS:STACK
START PROC FAR
    BEGIN: PUSH DS
           MOV AX, 0

```



```

    PUSH    AX
    MOV     AX, DATAEA
    MOV     DS, AX
    MOV     DX, 2
COUNT1: MOV     CX, COUNT
    MOV     BX, CX
    MOV     SI, CX
AGAIN:  MOV     AL, BLOCK[SI-1]
    CMP     AL, BLOCK[SI-2]
    JGE     NEXT
    XCHG    AL, BLOCK[SI-2]
    MOV     BLOCK[SI-1], AL
    MOV     BX, CX
NEXT:   DEC     SI
    DEC     CX

```

```

    CMP     CX, DX
    JGE     AGAIN
    CMP     BX, COUNT
    JE      DONE
    INC     DX
    JMP     COUNT1
DONE:   RET
START   ENDP
CODE    ENDS
        END    BEGIN

```

执行结果:

(BX) = _____
 (DX) = _____
 (CX) = _____
 (DS:0000H) = _____
 (DS:0007H) = _____

b) MCS-51 程序

```

ORG     0100H
BLOCK   EQU     8000H
COUNT  EQU     8
ADDR    EQU     40H
    MOV     R7, #COUNT
    MOV     R1, #ADDR
    MOV     R5, #1
    MOV     DPTR, #BLOCK
LOOP:   MOVX    A, @DPTR
    MOV     @R1, A
    INC     DPTR
    INC     R1
    DJNZ    R7, LOOP
LOOP2:  MOV     R7, #COUNT
    MOV     A, R7
    MOV     R6, A
    INC     R5

```

```

    MOV     R1, #47H
    MOV     R0, #46H
LOOP3:  CLR     C
    MOV     A, @R1
    SUBB    A, @R0
    JB      ACC.7, SIGN
    JB      PSW.2, EXCH
    STMP    UNCH
SIGN:   JB      PSW.2, UNCH
EXCH:   MOV     A, @R0
    XCH     A, @R1
    MOV     @R0, A
    MOV     A, R7
    MOV     R6, A

```

(转下页)

(接上页)

```

UNCH: DEC R0
      DEC R1
      DEC R7
      MOV A, R7
      CLR C
      SUBB A, R5
      JNC LOOP3
      MOV A, R6
      CJNZ A, #COUNT, LOOP2
    
```

```

      MOV R7, #COUNT
      MOV R1, #ADDR
      MOV DPTR, #BLOCK
Loop4: MOV A, @R1
      MOVB @DPTR, A
      INC DPTR
      INC R1
      DJNZ R7, LOOP4
HERE: STMP HERE
      END
    
```

程序执行前在数据区8000H开始的单元中存放有如下原始数据:

04H, 21H, 95H, 79H, 32H, FFH, 85H, 53H

则程序执行后: R6 = ____ H 数据区(8000H) = ____ H
 R7 = ____ H (8007H) = ____ H
 R5 = ____ H

三. 阅读下列说明和流程图, 把应填入其中①—⑤处的字句写在答卷的对应栏内.

(每题2分, 共10分)

[说明] 斐波那契数列 $\{T_k\}$ 的定义如下:

$$T_0 = 0$$

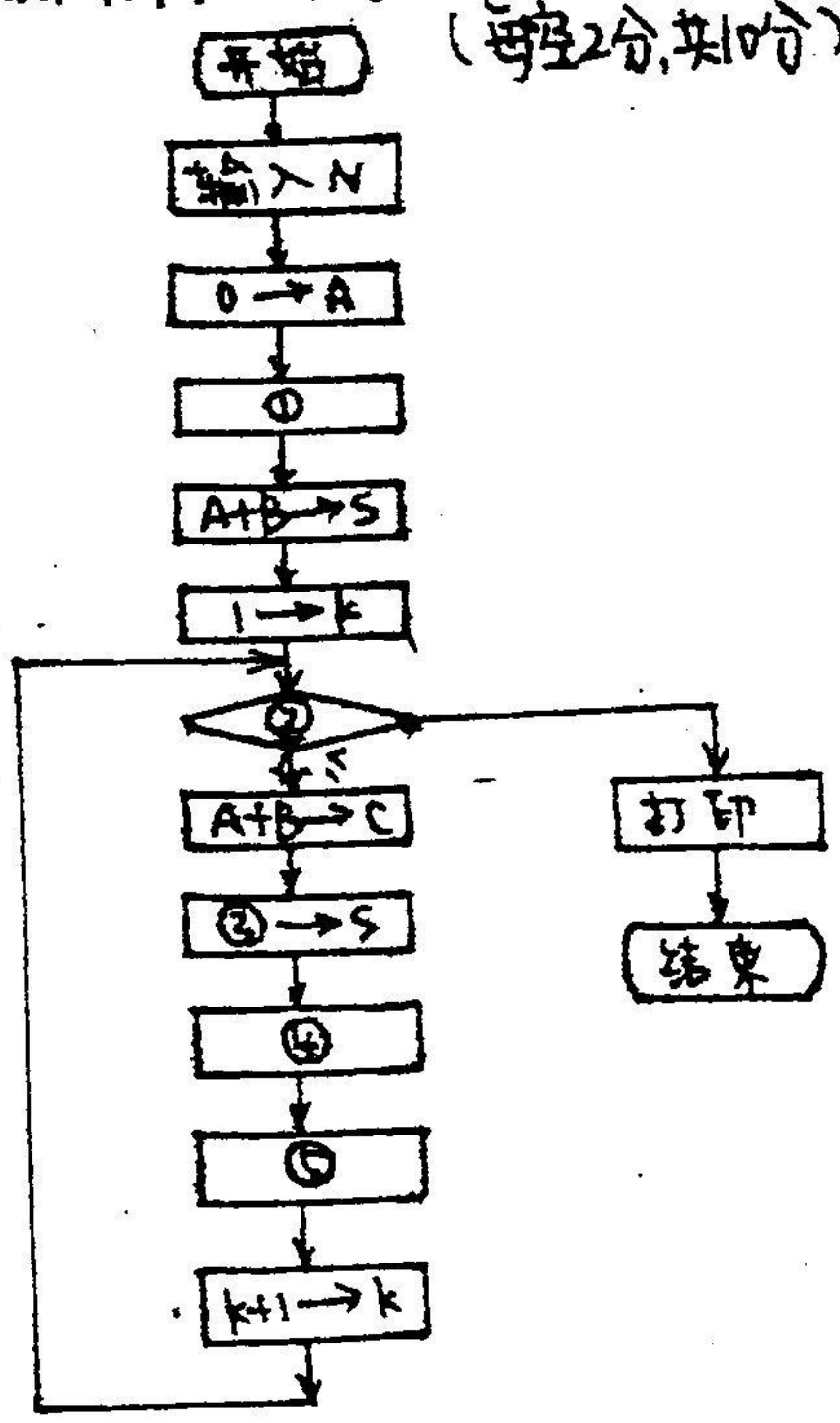
$$T_1 = 1$$

$$T_{k+1} = T_k + T_{k-1} \quad (k = 1, 2, 3, \dots)$$


本流程图描述计算斐波那契数列前N (N≥2) 项之和 ($S = T_0 + T_1 + \dots + T_{N-1}$) 的算法


[答案]

- ① _____
- ② _____
- ③ _____
- ④ _____
- ⑤ _____



四. 将下列RAM芯片6264(8K×8), 8255芯片, 8251芯片与系统三总线相连. 要求6264的地址为0000H~01FFFH, 8255的端口地址为90H~93H, 8251的端口地址为9FH. 8251芯片和CPU采用中断方式联结. (可根据需要加入地址译码器) (14分)

数据总线D₀~D₇ 


地址总线A₀~A₁₉ 

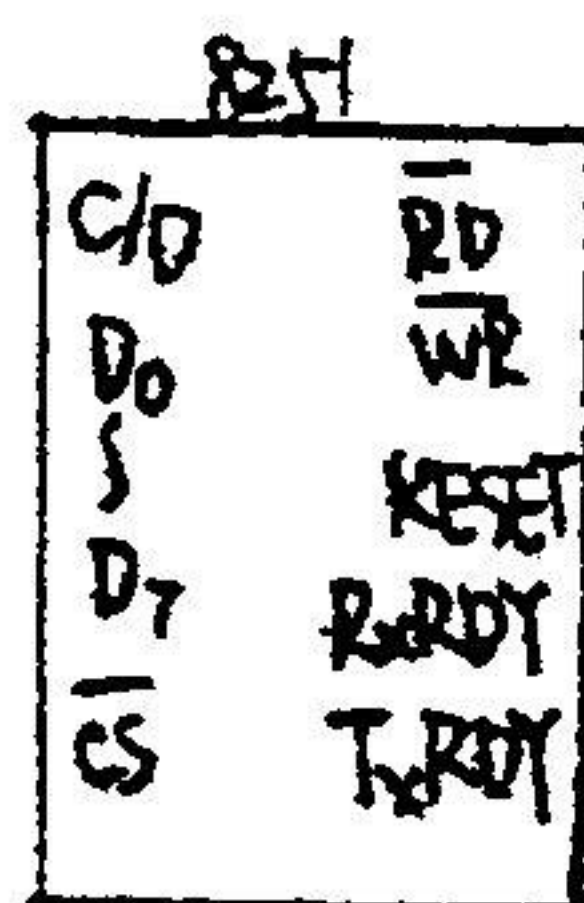
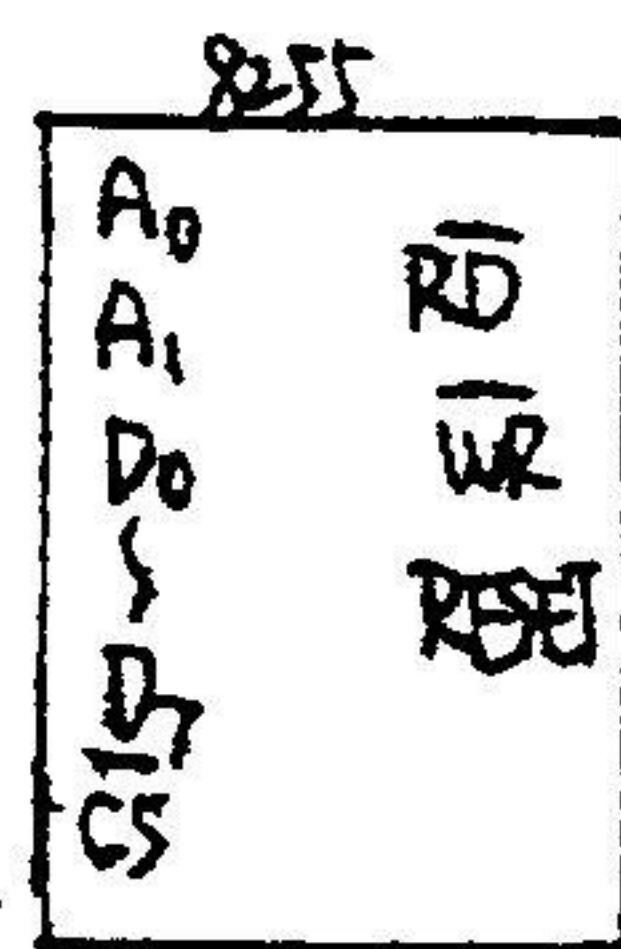
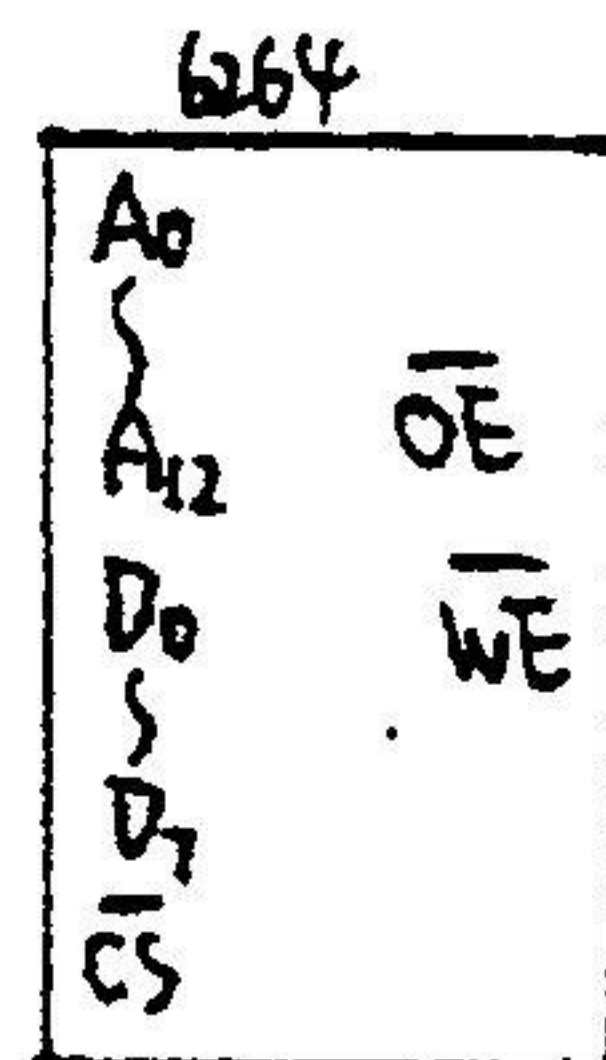
IO/ \overline{M} 

\overline{RD} 

\overline{WR} 

RESET 

INTR 



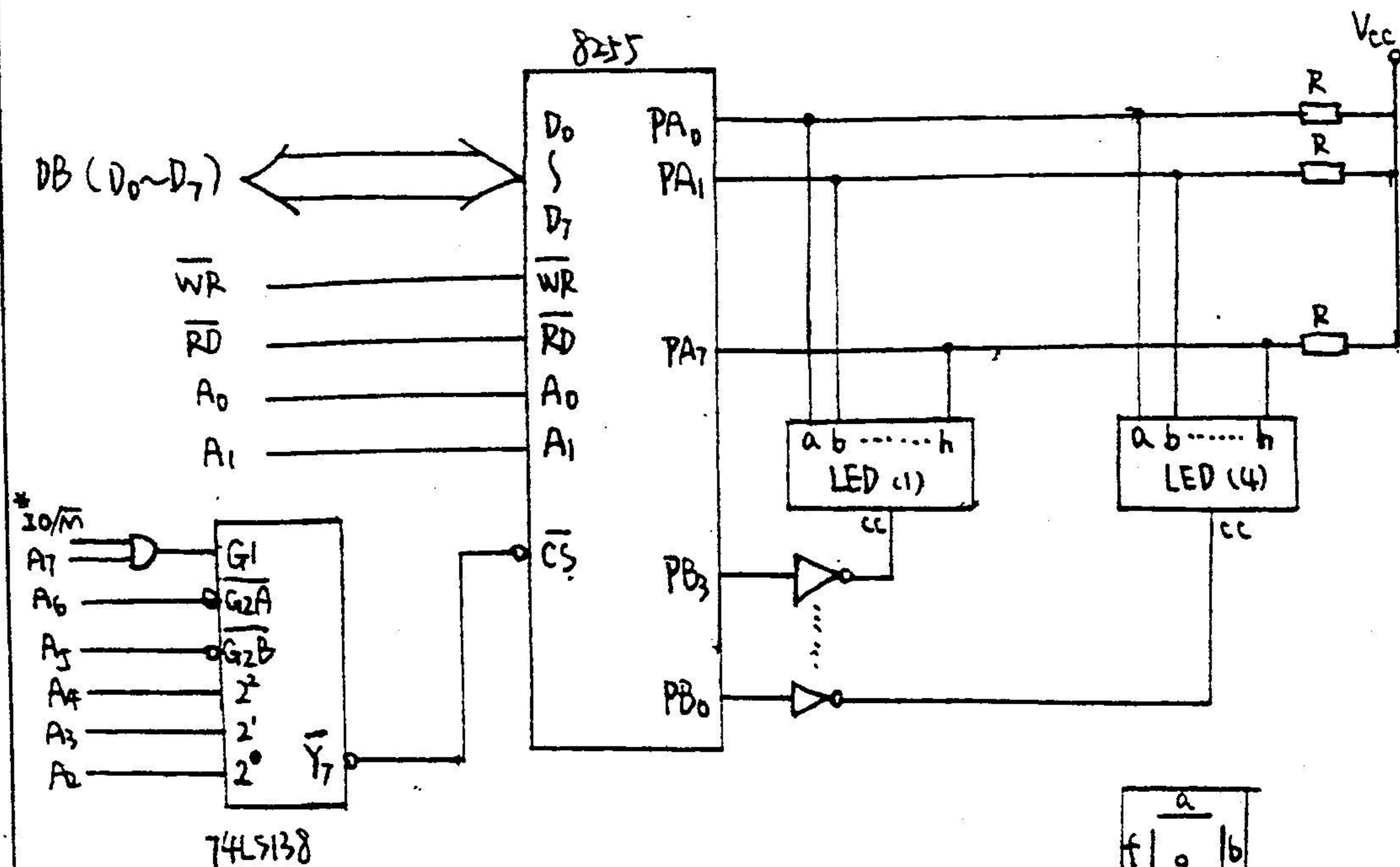
五. 8255芯片与CPU三总线的连接方式如图所示(图见下页) (15分)

1. 确定端口地址 PA口 = _____ PB口 = _____
PC口 = _____ 控制寄存器 = _____ (每空1分)

2. 4位LED动态显示数据, 字节. PAD为字型位, PBD中的PB₃~PB₀作字位控制. 要求四位LED动态显示内容为P123, 试写出字形的位显示编码.

显示内容	a	b	c	d	e	f	g	h
P								
1								
2								
3								

(每个位显示编码1分)



*当CPU为8031时,该信号固定接高电平

3. 用汇编语言编写程序,完成对8255的初始化及动态显示P12.3. (7分)

说明:①假定延时子程序的标号为DLM,可以直接调用,不必重新设计

② 8255 控制字格式

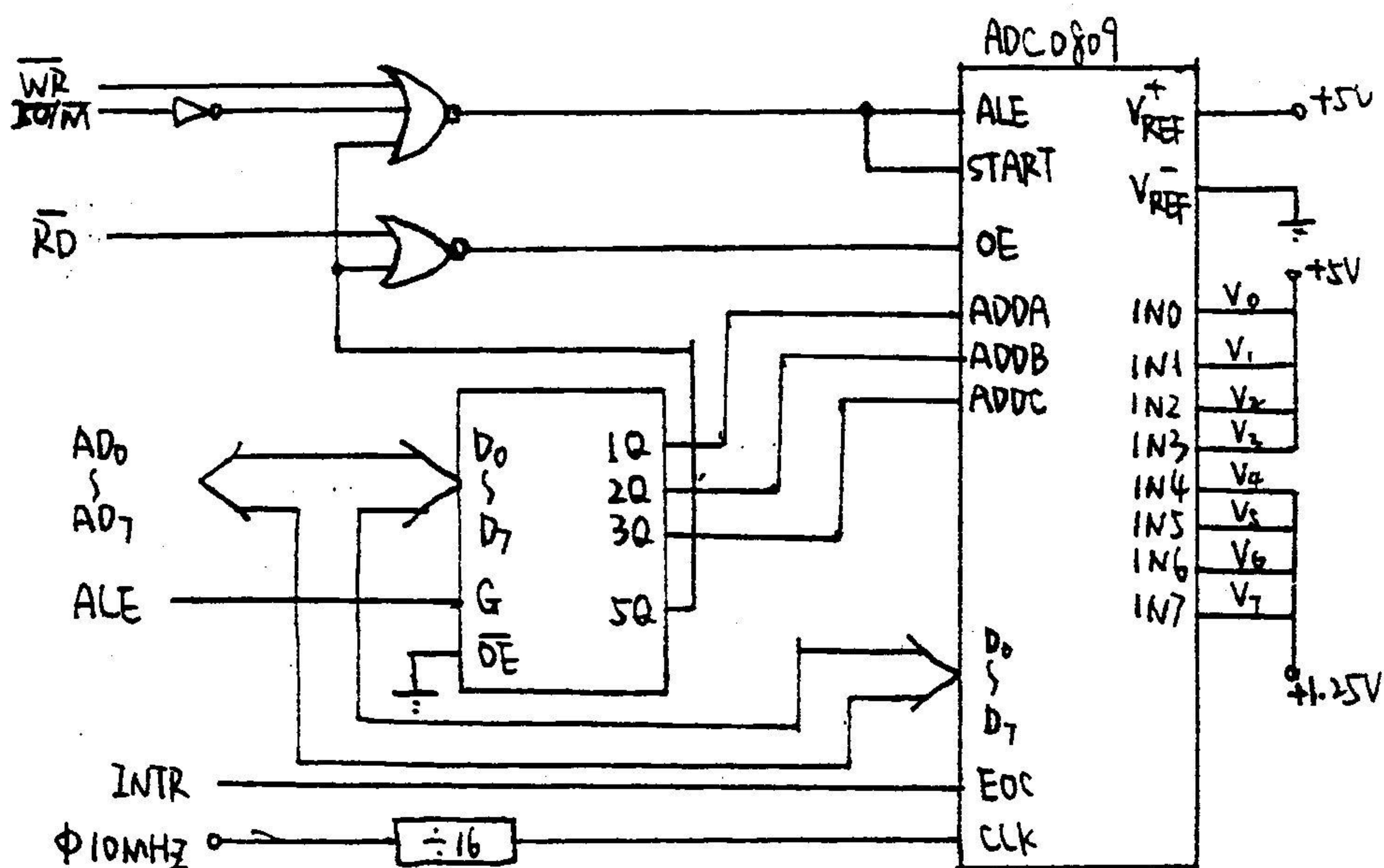
与B组有关
由方式有关

	I	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
A组方式								
方式0		0	0	与A组有关 由方式决定				B组方式
方式1		0	1					方式0
方式2		1	x					方式1

六 ADC 0809和CPU的连接框图如下 (若采用8031系统解题,请参阅题后说明)

要求将0809的8路模拟量 V_0, V_1, \dots, V_7 依次采样,这样的采样过程循环操作16次。A/D转换结束后,将采样结果存放在首地址为DS:2000H的RAM中

1. 若要在主程序中进行A/D通道的切换及启动操作,在中断服务程序中进行采样操作。试编写相应的程序段,完成上述功能。 (7分)



2. 根据基准电压 V_0, V_1, \dots, V_7 的值写出内存单元 DS:2000H ~ DS:2007H 的内容。(8分)

(若采用 8031 CPU 来完成此题, 请阅读下列说明:

1. IO/M 改为固定的电平
2. EOC 输出信号经反相后送 CPU 的 INTO 引脚
3. 数据存放在 2000H 开始的单元中。)

ADDA、ADDB、ADDC 与 8 路输入 INX 的关系真值表:

IN	0	1	2	3	4	5	6	7
ADDA	0	1	0	1	0	1	0	1
ADDB	0	0	1	1	0	0	1	1
ADDC	0	0	0	0	1	1	1	1