

西安电子科技大学

微机原理与系统设计 课程实验报告

实验名称 实验一 汇编语言编程实验

计算机科学与技术 学院 2103015 班

姓名 李志豪 学号 21009290012

同作者

实验日期 2023 年 11 月 26 日

实验地点 E-II-311 实验批次 第 11 批

成 绩

指导教师评语：

指导教师：

____年____月____日

实验报告内容基本要求及参考格式

- 一、实验目的
- 二、实验所用仪器（或实验环境）
- 三、实验基本原理及步骤（或方案设计及理论计算）
- 四、实验数据记录（或仿真及软件设计）
- 五、实验结果分析及回答问题（或测试环境及测试结果）

实验一 汇编语言编程实验

1 实验目的

1. 掌握汇编语言的编程方法；
2. 掌握 DOS 功能调用的使用方法；
3. 掌握汇编语言程序的调试运行过程；

2 实验内容

1. 将指定数据区的字符串数据以 ASCII 码形式显示在屏幕上，并通过 DOS 功能调用完成必要提示信息的显示；
2. 在屏幕上显示自己的学号姓名信息；
3. 循环从键盘读入字符并回显在屏幕上，然后显示出对应字符的 ASCII 码，直到输入“Q”或“q”时结束；
4. 自主设计输入显示信息，完成编程与调试，演示实验结果；

3 实验步骤

1. 运行 QTHPCI 软件，根据实验内容，参考程序流程图编写程序；
2. 选择“项目”菜单中的“编译”或“编译连接”对实验程序进行编译连接；
3. 选择“调试”菜单中的“进行调试”，进入 Debug 调试，观察调试过程中传输指令执行后各寄存器及数据区的内容。按 F9 连续运行；

4 实验过程

4.1 系统功能调用

在本次实验中，针对总体的任务均需要进行汇编语言中 DOS 功能的调用，并利用中断请求指令 INT 21H 进行中断调用，所用到的 DOS 功能调用见表 1。

表 1: DOS 功能调用说明

AH 值	功能	调用参数	结果
1	键盘输入并回显		AL= 输出字符
2	显示单个字符 (带 Ctrl+Break 检查)	DL= 输出字符	光标在字符后面
6	显示单个字符 (无 Ctrl+Break 检查)	DL= 输出字符	光标在字符后面
8	从键盘上读一个字符		AL= 字符的 ASCII 码
9	显示字符串	DS:DX= 串地址, '\$' 为结束字符	光标跟在串后面
4CH	返回 DOS 系统		AL= 返回码

4.2 字符转换为 ASCII 码流程设计

根据任务要求, 需要对数据区的字符串数据以 ASCII 码形式显示在屏幕上, 以及涉及到将字符转化为 ASCII 码输出功能, 因此需要对汇编程序设计程序执行流程图。由 ASCII 码转换原理, 对于任意一个字符, 其对应的 ASCII 码值计算定理为:

引理 4.1. 对于存储在计算机中的字符的 ASCII 码, 其高四位与低四位分别记为 X,Y, 则有若 X 或 Y 值为 0~9H 时, 需要加上调整值 30H; 当 X 或 Y 值为 A~FH 时, 需要加上调整值 37H。

因此, 根据引理 4.1 设计程序流程图如图 1所示:

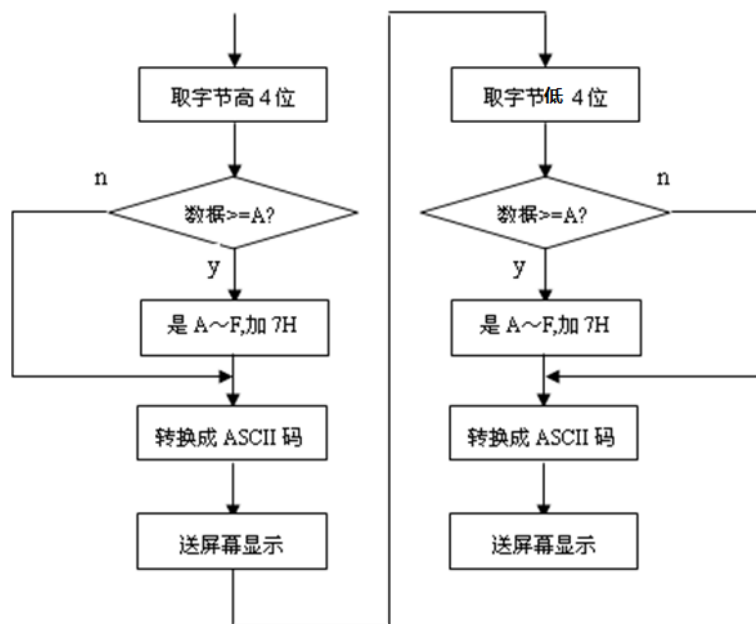


图 1: 字符转 ASCII 码流程图

4.3 任务一：以 ASCII 码形式输出数据区的字符串数据

在本次实验中，首先在数据区自定义待转换的字符串数据：“This is the first program of assembly language.”，然后编写子程序根据上述转换原理进行 ASCII 码转换，并将每个字符的转换结果以十六进制显示在屏幕上。

- 程序数据区定义

```
1 data segment
2 info db 'The following string needs to be convert to heximal format:',
3         ODH, 0AH, '$'
4 infoD db 'This is the first program of assembly language.', ODH, 0AH, '$'
5 infoC db 'Convert this string showing as follows:', ODH, 0AH, '$'
6 string db 'Name:Zhihao Li; ID:21009290012$', ODH, 0AH, '$'
7 tip db 'Input Single Character(enter q or Q to quit):$'
8 orgc db 'Inputed Character:$'
9 hex db 'Ouput Ascii:$'
10 data ends
```

- 任务一子程序

1 Task1:	21 mov bh, [si]
2 ; 输出提示信息	22 and bl, 0F0h ; 取高四位
3 lea dx, info	23 push cx
4 mov ah, 9	24 mov cl, 4
5 int 21h	25 shr bl, cl
6	26 pop cx
7 ; 输出待转换字符串	27 cmp bl, 9
8 lea dx, infoD	28 ja T1_alpha1
9 mov ah, 9	29
10 int 21h	30 T1_number1:
11	31 add bl, '0'
12 ; 输出转换提示信息	32 mov dl, bl
13 lea dx, infoC	33 mov ah, 2
14 mov ah, 9	34 int 21h
15 int 21h	35 jmp T1_Low4
16	36 T1_alpha1:
17 LEA si, infoD	37 add bl, 37h
18 mov cx, LENGTHOF infoD	38 mov dl, bl
19 T1_Transform:	39 mov ah, 2
20 mov bl, [si]	40 int 21h

<pre> 41 T1_Low4: 42 and bh, 0Fh ; 取低四位 43 cmp bh, 9 44 ja T1_alpha2 45 46 T1_number2: 47 add bh, '0' 48 mov dl, bh 49 mov ah, 2 50 int 21h 51 jmp T1_H 52 53 T1_alpha2: 54 add bh, 37h 55 mov dl, bh 56 mov ah, 2 </pre>	<pre> 57 int 21h 58 jmp T1_H 59 60 T1_H: 61 ; 输出 H 字符以及空格 62 mov dl, 72 63 mov ah, 2 64 int 21h 65 66 mov dl, 20h ; 换行符 67 mov ah, 2 68 int 21h 69 70 inc si 71 Loop T1_Transform 72 </pre>
--	--

- 任务一运行结果

在星研集成开发环境中，编译运行该汇编程序，得到如图 2 的运行结果：

```

D:\>D:\test
The following string needs to be convert to hexinal format:
This is the first program of assembly language.
Convert this string showing as follows:
54H 68H 69H 73H 20H 69H 73H 20H 74H 68H 65H 20H 66H 69H 72H 73H 74H 20H 70H 72H
6FH 67H 72H 61H 6DH 20H 6FH 66H 20H 61H 73H 73H 65H 6DH 62H 6CH 79H 20H 6CH 61H
6EH 67H 75H 61H 67H 65H 2EH 0DH 0AH 24H

```

图 2: 任务一程序运行结果

- 任务一结果分析

从运行结果可以看出，在必要信息的提示下，程序将数据区的整个字符串成功转换为十六进制表示的 ASCII 码输出，经过对比 ASCII 码表进行验证，所有字符均得到了正确的转换。

4.4 任务二：在屏幕上显示自己的学号姓名信息

对于这个任务，需要在数据区定义自己的学号与姓名信息，然后利用 DOS 功能调用即可输出到屏幕上。

- 程序数据区定义

```

1 data segment
2 info db 'The following string needs to be convert to heximal format:',
3     ODH, 0AH, '$'
4 infoD db 'This is the first program of assembly language.', ODH, 0AH, '$'
5 infoC db 'Convert this string showing as follows:', ODH, 0AH, '$'
6 string db 'Name:Zhihao Li; ID:21009290012$', ODH, 0AH, '$'
7 tip db 'Input Single Character(enter q or Q to quit):$'
8 orgc db 'Inputed Character:$'
9 hex db 'Ouput Ascii:$'
10 data ends

```

- 任务二子程序

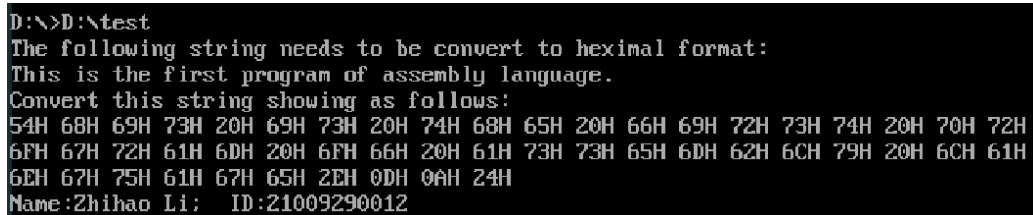
```

1     Task2:
2         ; 输出姓名学号
3         lea dx, string
4         mov ah, 9
5         int 21h

```

- 任务二运行结果

在整个程序中，该任务得到如图 3 的运行结果：



```

D:\>D:\test
The following string needs to be convert to heximal format:
This is the first program of assembly language.
Convert this string showing as follows:
54H 68H 69H 73H 20H 69H 73H 20H 74H 68H 65H 20H 66H 69H 72H 73H 74H 20H 70H 72H
6FH 67H 72H 61H 6DH 20H 6FH 66H 20H 61H 73H 73H 65H 6DH 62H 6CH 79H 20H 6CH 61H
6EH 67H 75H 61H 67H 65H 2EH 0DH 0AH 24H
Name:Zhihao Li; ID:21009290012

```

图 3: 任务二程序运行结果

4.5 任务三：循环从键盘读入字符并回显在屏幕上，然后显示出对应字符的 ASCII 码

在这项任务中，将读入的字符存入到 AL 寄存器中，通过设定退出程序的条件为“Q”或“q”实现输入结束。回显字符：调用 DOS 中断服务，将输入字符输出，字符转换为 ASCII 码：利用转换引理，编写程序，进行转换输出。

- 任务三子程序

<pre> 1 T3_cmpq: 2 lea dx, tip 3 mov ah, 9 4 int 21h 5 6 mov ah, 1 7 int 21h 8 cmp al, 'q' 9 ; 若等于 q/Q, 则跳转到 ending 10 je ending 11 cmp al, 'Q' 12 je ending 13 14 mov bl, al 15 mov bh, al 16 lea dx, orgc 17 mov ah, 9 18 int 21h 19 20 mov dl, bh 21 mov ah, 2 22 int 21h 23 mov dl, 9 24 mov ah, 2 25 int 21h 26 lea dx, hex 27 mov ah, 9 28 int 21h 29 30 and bl, 0F0h ; 取高四位 31 mov cl, 4 32 shr bl, cl 33 cmp bl, 9 34 ja T3_alph1 35 T3_numbl: 36 add bl, '0' 37 mov dl, bl 38 mov ah, 2 39 int 21h 40 jmp T3_low4 </pre>	<pre> 41 T3_alph1: 42 add bl, 37h 43 mov dl, bl 44 mov ah, 2 45 int 21h 46 T3_low4: 47 and bh, 0Fh ; 取低四位 48 cmp bh, 9 49 ja T3_alph2 50 T3_numbl: 51 add bh, '0' 52 mov dl, bh 53 mov ah, 2 54 int 21h 55 mov dl, 72 56 mov ah, 2 57 int 21h 58 jmp T3_cmpq 59 60 T3_alph2: 61 add bh, 37h 62 mov dl, bh 63 mov ah, 2 64 int 21h 65 mov dl, 72 66 mov ah, 2 67 int 21h 68 jmp T3_cmpq 69 70 ending: 71 mov dl, 0Dh 72 mov ah, 2 73 int 21h 74 mov dl, 0Ah 75 mov ah, 2 76 int 21h 77 78 ; 终止程序 79 mov ax, 4c00h 80 int 21h </pre>
---	--

- 任务三运行结果在整个程序中，该任务得到如图 4 的运行结果：

```
D:\>D:\test
The following string needs to be convert to heximal format:
This is the first program of assembly language.
Convert this string showing as follows:
54H 68H 69H 73H 20H 69H 73H 20H 74H 68H 65H 20H 66H 69H 72H 73H 74H 20H 70H 72H
6FH 67H 72H 61H 6DH 20H 6FH 66H 20H 61H 73H 73H 65H 6DH 62H 6CH 79H 20H 6CH 61H
6EH 67H 75H 61H 67H 65H 2EH 0DH 0AH 24H
Name:Zhihao Li; ID:21009290012
Input Single Character(enter q or Q to quit):
1
Inputed Character:1      Duput Ascii:31H
Input Single Character(enter q or Q to quit):
2
Inputed Character:2      Duput Ascii:32H
Input Single Character(enter q or Q to quit):
4
Inputed Character:4      Duput Ascii:34H
Input Single Character(enter q or Q to quit):
q
```

图 4: 任务三程序运行结果

- 任务三结果分析

从运行结果可以看出，在必要信息的提示下，程序对键盘输入的单个字符再次回显到“Inputed Character:”后，转换的 ASCII 码对应十六进制形式输出在“Output Ascii:”后；并且实现功能的循环操作，即在下一次再次通过“Input Single Character(enter q or Q to quit)”提示信息，等待用户输入下一个字符，最终通过控制字符“Q/q”退出程序。

5 实验心得与体会

通过本次实验，我将微机原理课程中学习到的理论知识成功应用到了实践当中，汇编编程不再是一个难题，尤其是在掌握了基础的星研软件编程和调试的基本用法后，自己在实验当中能够自主设计汇编程序，出色地完成实验任务。

西安电子科技大学

微机原理与系统设计 课程实验报告

实验名称 实验二 数码转换实验

计算机科学与技术 学院 2103015 班

姓名 李志豪 学号 21009290012

同作者

实验日期 2023 年 12 月 3 日

实验地点 E-II-311 实验批次 第 11 批

成 绩

指导教师评语：

指导教师：

____年____月____日

实验报告内容基本要求及参考格式

- 一、实验目的
- 二、实验所用仪器（或实验环境）
- 三、实验基本原理及步骤（或方案设计及理论计算）
- 四、实验数据记录（或仿真及软件设计）
- 五、实验结果分析及回答问题（或测试环境及测试结果）

实验二 数码转换实验

1 实验目的

1. 掌握不同进制数及编码相互转换的程序设计方法；
2. 掌握运算类指令编程及调试方法；
3. 掌握循环程序的设计方法；

2 实验内容

1. 重复从键盘输入不超过 5 位的十进制数，按回车键结束输入
2. 将该十进制数转换成十六进制数；结果以十六进制数的形式显示在屏幕上；
3. 如果输入非数字字符，则报告出错信息，重新输入；
4. 直到输入“Q”或“q”时程序运行结束；
5. 键盘输入一字符串，以空格结束，统计其中数字字符的个数，在屏幕显示；

3 实验原理

十进制数可以表示为： $D_n * 10^n + D_{n-1} * 10^{n-1} + \dots + D_0 * 10^0 = \sum_i^n D_i * 10^i$ ，其中 D_i 表示十进制数 0, 1, 2, 3, ..., 9，由此上式可以转换为：

$$\sum_i^n D_i * 10^i = (((((D_n * 10 + D_{n-1}) * 10) + D_{n-2}) * 10 + \dots + D_1) * 10 + D_0$$

由上式可归纳出十进制数转换为二进制数的方法：

引理 3.1. 从十进制数的最高位 D_n 开始做乘 10 加次位的操作，依此类推，则可求出二进制数结果；由求得的二进制数结果，在利用寄存器移位操作便很容易将其转换为十六进制数结果。

4 实验过程

4.1 十进制 ASCII 码转换为二进制数流程设计

由于用户通过键盘输入的字符数据读入到内存中，是以 ASCII 码形式存储的，为了获得对应的数字，必须先对其进行转换，然后根据上述引理求得进制转换后的结果，设计的转换流程图如图所示：

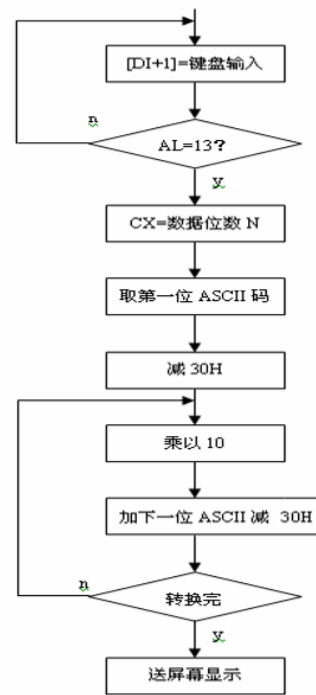


图 5: 十进制 ASCII 码转换为二进制数流程图

4.2 十进制数读入以及字符合法性判断

通过 `int 21H` 中断的 `0AH` 号功能 `Dos` 功能调用实现字符串的读入，对于该任务需要进行输入字符的合法性判断，以及退出条件检查。

- 合法性判断：当输入字符不合法、输入数字字符个数超出 5 位时进入异常处理模块
- 退出条件检查：当输入 “Q” 或 “q” 时会跳转至程序最后，执行退出程序模块
- 程序数据区定义

```

1 DATA SEGMENT
2     DBuffer DB 30 DUP(0)
3     InputInfo DB 0DH, 0AH, '-----',
4                 0DH, 0AH,
5                 'Please input the number in decimal format:', 0DH, 0AH,
6                 '> 1. The number must be less than or equal to 65535', 0DH, 0AH,
7                 '> 2. Enter q or Q to quit', 0DH, 0AH, '> Input number: ', '$'
8     ErrorInfo DB 0DH, 0AH, 'Sorry, the character you input has wrong form!',
9                 0DH, 0AH, '$'
10    ResultInfo DB 'Decimal format to binary format:', 0DH, 0AH,
11                '> Output number: $'
12    ExitInfo DB 0DH, 0AH, 'Congratulations, you have exit successful!',

```

```

13         0DH, 0AH, '$'
14     StringInfo DB '-----', 0DH, 0AH,
15         'Please input the string to be detect:', 0DH, 0AH,
16         '> Input string: ', '$'
17     SRInfo DB 0DH, 0AH, 'The number of digital characters in the string: ', '$'
18     SBuffer DB 28, 29 DUP(0)
19 DATA ENDS

```

- 任务子程序

<pre> 1 main PROC 2 START: 3 MOV AX, DATA 4 MOV DS, AX 5 JMP INITIAL 6 7 ERROR: 8 MOV AH, 09H 9 MOV DX, OFFSET ErrorInfo 10 INT 21H 11 12 INITIAL: 13 MOV AH, 09H 14 MOV DX, OFFSET InputInfo 15 INT 21H 16 17 ; initilize the memory 18 MOV DI, OFFSET DBuffer 19 20 ; the length of number 21 MOV CX, 0H </pre>	<pre> 22 INPUT: 23 MOV AH, 01H 24 INT 21H 25 ; detect if it is a carriage return 26 CMP AL, 0DH 27 JE TRANSFORM 28 CMP AL, 'Q' 29 JE FINAL 30 CMP AL, 'q' 31 JE FINAL 32 CMP AL, '0' 33 JB ERROR 34 CMP AL, '9' 35 JA ERROR 36 CMP CX, 5 ; CX: 0-4 37 JAE ERROR 38 ; store the right digital character 39 MOV [DI], AL 40 INC DI 41 INC CX 42 JMP INPUT </pre>
--	--

在初始化数字字符存储的数据区以及程序后，通过比较读入到 AL 的数字字符，如果是回车符说明，数字字符串读入完毕，直接进入译码转换模块；如果读入的字符“Q”或“q”，表示用户结束输入，程序进入退出模块；如果是非数字字符，则引起非法错误，进入异常处理模块；如果是正常数字字符，则将其存入到初始化的数据缓存区，并计算数字字符个数存入到 CX 中；如果数字字符超过 5 位数，则进入异常处理模块。

4.3 十六进制转换与输出

对于该项任务，首先编写十进制转二进制子程序，然后利用寄存器中二进制数据与十六进制的关系，以十六进制形式输出寄存器中内容，即可完成十进制数据向十六进制数据的转换。

- 任务子程序

43	TRANSFORM:	71	OUTPUT:
44	<i>; if no digital character is input</i>		PUSH CX
45	CMP CX, 0	73	MOV CL, 4
46	JE ERROR	74	ROL AX, CL
47	<i>; initialize</i>	75	POP CX
48	MOV DI, OFFSET DBuffer	76	
49	MOV AX, 0	77	MOV DL, AL
50	MOV BX, 10	78	AND DL, 0FH
51	LEAP:	79	ADD DL, 30H
52	MUL BX	80	
53	CLC	81	CMP DL, ':'
54	ADD AL, [DI]	82	JB NO
55	ADC AH, 0	83	ADD DL, 07H
56	<i>; transform the ascii code</i>	84	NO:
57	SUB AX, 30H	85	PUSH AX
58		86	MOV AH, 02H
59	INC DI	87	INT 21h
60	DEC CX	88	POP AX
61	JNZ LEAP	89	LOOP OUTPUT
62		90	MOV DL, 72
63	PUTRESULT:	91	MOV AH, 02H
64	PUSH AX	92	INT 21h
65	MOV AH, 09H	93	JMP INITIAL
66	MOV DX, OFFSET ResultInfo	94	
67	INT 21H	95	FINAL:
68	POP AX	96	MOV AH, 09H
69		97	MOV DX, OFFSET ExitInfo
70	MOV CX, 0004H	98	INT 21H

- 十进制向二进制转换

对于正常输入的字符数据，首先初始化数据结果寄存器 AX，基数寄存器 BX，然后通过

LEAP 模块使得 AX 寄存器不断乘以十进制计数 BX 并加上 ASCII 字符转换后的低位数据，直到所有的字符数据全部转换完毕，AX 寄存器中存储的便是转换后的二进制数据。

- 二进制向十六进制转换

为了进一步转换为十六进制数，利用寄存器的输出形式，即 OUTPUT 模块，对于 AX 中寄存器数据需要循环输出 4 次，先高位再低位输出；将二进制数据转换为十六进制的 ASCII 码字符输出到屏幕上。

4.4 字符串中数字字符个数统计

对于该项任务，首先在数据区定义字符串缓冲区，定义最大的字符读入数量，然后读入用户键盘输入的任意字符串，对每一个字符进行检测统计，如若是数字字符则进行计数并存储到 AX 中，最后以十六进制数将数字字符个数输出到屏幕上。

- 任务子程序

99	DetectString:	124	PUSH AX
100	MOV AH, 09H	125	MOV AH, 09H
101	MOV DX, OFFSET StringInfo	126	MOV DX, OFFSET SRInfo
102	INT 21H	127	INT 21H
103	; detect digital characters	128	POP AX
104	MOV AX, 0A00H	129	MOV CX, 0004H
105	MOV DX, OFFSET SBuffer	130	OUTPUT1:
106	INT 21H	131	PUSH CX
107		132	MOV CL, 4
108	MOV SI, DX	133	ROL AX, CL
109	MOV CX, 0	134	POP CX
110	MOV CL, [SI+1]	135	
111	ADD SI, 2	136	MOV DL, AL
112	MOV AX, 0	137	AND DL, 0FH
113	CHECK:	138	ADD DL, 30H
114	MOV BL, [SI]	139	
115	INC SI	140	CMP DL, ':'
116	CMP BL, 30H	141	JB NO1
117	JB NOTNUM	142	ADD DL, 07H
118	CMP BL, 39H	143	NO1:
119	JA NOTNUM	144	PUSH AX
120	INC AX	145	MOV AH, 02H
121		146	INT 21h
122	NOTNUM:	147	POP AX
123	LOOP CHECK	148	LOOP OUTPUT1

4.5 程序运行结果

- 正常进制转换结果

```
D:\>D:\test

-----
Please input the number in decimal format:
> 1. The number must be less than or equal to 65535
> 2. Enter q or Q to quit
> Input number: 65535
Decimal format to binary format:
> Output number: FFFFH

-----
Please input the number in decimal format:
> 1. The number must be less than or equal to 65535
> 2. Enter q or Q to quit
> Input number: 4096
Decimal format to binary format:
> Output number: 1000H

-----
Please input the number in decimal format:
> 1. The number must be less than or equal to 65535
> 2. Enter q or Q to quit
> Input number: _
```

图 6: 进制转换运行结果

- 异常处理及程序退出

```
-----
Please input the number in decimal format:
> 1. The number must be less than or equal to 65535
> 2. Enter q or Q to quit
> Input number: F
Sorry, the character you input has wrong form!

-----
Please input the number in decimal format:
> 1. The number must be less than or equal to 65535
> 2. Enter q or Q to quit
> Input number: 123456
Sorry, the character you input has wrong form!

-----
Please input the number in decimal format:
> 1. The number must be less than or equal to 65535
> 2. Enter q or Q to quit
> Input number: q
Congratulations, you have exit successful!

-----
```

图 7: 异常处理及程序退出运行结果

- 字符串数字字符的统计

```
-----
Please input the string to be detect:
> Input string: afba12425nfaig121424vag443
The number of digital characters in the string: 000EH

-----
```

图 8: 字符串数字字符的统计运行结果

5 实验心得与体会

通过本次实验，我掌握了基础的星研软件编程和调试的基本用法，为接下来的编程和硬件操作打下了基础。并且通过实验自主编写汇编程序，提高了自己的汇编编程能力，通过学习不同进制以及编码相互转换原理、运算类指令编程以及汇编调试方法，我的项目开发能力也得到了增强。

西安电子科技大学

微机原理与系统设计 课程实验报告

实验名称 实验三 基本 I/O 扩展实验

计算机科学与技术 学院 2103015 班

姓名 李志豪 学号 21009290012

同作者

实验日期 2023 年 12 月 10 日

实验地点 E-II-311 实验批次 第 11 批

成绩

指导教师评语：

指导教师：

____年____月____日

实验报告内容基本要求及参考格式

- 一、实验目的
- 二、实验所用仪器（或实验环境）
- 三、实验基本原理及步骤（或方案设计及理论计算）
- 四、实验数据记录（或仿真及软件设计）
- 五、实验结果分析及回答问题（或测试环境及测试结果）

实验三 基本 IO 扩展实验

1 实验目的

1. 了解 TTL 芯片扩展简单 I/O 口的方法;
2. 掌握数据输入输出程序编制的方法;

2 实验内容

本实验要求用 74LS244 作为输入口, 读取开关状态, 并将此状态通过 74LS273 连到发光二极管显示。具体实验内容如下:

1. 开关 Y_i 为低电平时对应的发光二极管亮, Y_i 为高电平时对应的发光二极管灭
2. 当开关 Y_i 全为高电平时, 发光二极管 Q_i 从左至右轮流点亮;
3. 当开关 Y_i 全为低电平时, 发光二极管 Q_i 从右至左轮流点亮;
4. 自主设计控制及显示模式, 完成编程调试, 演示实验结果;

3 实验原理

74LS244 是一种三态输出的 8 总线缓冲驱动器, 无锁存功能, 当 G 为低电平, A_i 信号传送到 Y_i , 当为高电平时, Y_i 处于禁止高阻状态; 74LS273 是一种带清除功能的 8D 触发器, $1D \sim 8D$ 为数据输入端, $1Q \sim 8Q$ 为数据输出端, 正脉冲触发, 低电平清除, 常用作 8 位地址锁存器。

4 实验过程

4.1 实验电路的连接

244 的 CS 接到 ISA 总线接口模块的 0000H, $Y_7 \sim Y_0$ ——开关 $K_1 \sim K_8$; 273 的 CS 接到 ISA 总线接口模块的 0020H, $Q_7 \sim Q_0$ ——发光二极管 $L_1 \sim L_8$; 该模块的 WR、RD 分别连到 ISA 总线接口模块的 IOWR、IORD; 该模块的数据 ($AD_0 \sim AD_7$) 连到 ISA 总线接口模块的数据 ($LD_0 \sim LD_7$)。

4.2 编写实验程序并编译链接运行

1	IO244 EQU 0230H;74LS244 端口地址	39	JNE R2L
2	IO273 EQU 0230H;74LS273 端口地址	40	JMP INPUT
3		41	
4	_STACK SEGMENT STACK	42	Q2:
5	DW 100 DUP(?)	43	MOV AX, 0FFFEH
6	_STACK ENDS	44	MOV DX, IO273
7		45	L2R:
8	_DATA SEGMENT WORD	46	CALL DELAY
9	_DATA ENDS	47	OUT DX, AX
10		48	ROR AX, 1
11	CODE SEGMENT	49	CMP AX, 0FFFEH
12	ASSUME CS:CODE, DS:_DATA, SS:_STACK	50	; 若相等, 说明一轮从左往右已经完成
13	START PROC NEAR	51	; 若不等, 则继续循环
14	MOV AX, _DATA	52	JNE L2R
15	MOV DS, AX	53	; 继续读入开关状态
16	INPUT:	54	JMP INPUT
17	MOV DX, IO244	55	Q3:
18	IN AX, DX	56	MOV AX, 0FFFEH
19	CMP AX, 0FFFFH ; 若开关全为低电平	57	MOV DX, IO273
20	JZ Q1 ; 从右往左依次点亮	58	CNT:
21	CMP AX, 0; 若开关全为高电平	59	CALL Delay
22	JZ Q2; 从左往右依次点亮	60	OUT DX, AX
23	CMP AX, 7FFEh	61	DEC AX
24	;JZ Q3	62	CMP AX, 0000H
25	MOV DX, IO273	63	JNE CNT
26	MOV AH, AL	64	JMP INPUT
27	;NOT AX ;AX 取非送给 273	65	Delay PROC NEAR; 延时子程序
28	; 点亮对应的二极管	66	Delay1:
29	OUT DX, AX	67	XOR CX,CX
30	JMP INPUT	68	LOOP\$
31	Q1:	69	RET
32	MOV AX, 7FFFH	70	Delay ENDP
33	MOV DX, IO273	71	START ENDP
34	R2L:	72	CODE ENDS
35	CALL DELAY ; 延时	73	END START
36	OUT DX, AX ; 送给 273, 点亮对应的二极管		
37	ROL AX, 1	75	
38	CMP AX, 7FFFH	76	

4.3 验证电路功能与结果分析

编译链接项目后，改变开关 Y_i 可观察到：

- 开关 Y_i 为低电平时对应的发光二极管亮， Y_i 为高电平时对应的发光二极管灭；
- 当开关 Y_i 全为高电平时，发光二极管 Q_i 从左至右轮流点亮；
- 当开关 Y_i 全为低电平时，发光二极管 Q_i 从右至左轮流点亮；

在完成基本实验的基础上，我还额外自主设计控制及显示模式，即循环熄灯功能：通过 74LS244 获取开关状态，当开关状态编码为 01H 时，设计电路从左至右轮流熄灯；当开关状态编码为 02H 时，设计电路从右至左轮流熄灯；经过在实验电路板上实际测试，验证了程序执行的正确性，并且经过实验室老师实际测试和检验。

5 实验心得与体会

在本次实验中，通过利用 74LS244 与 74LS273 芯片，掌握了计算机与外设通过接口进行通信的过程，在实际操作中，通过开关状态作为输入信息，控制二极管的显示，并且自主设计控制及显示模式。这极大地锻炼了我的汇编编程能力，为接下来的编程和硬件操作打下了基础。

西安电子科技大学

微机原理与系统设计 课程实验报告

实验名称 实验四 可编程并行接口实验

计算机科学与技术 学院 2103015 班

姓名 李志豪 学号 21009290012

同作者

实验日期 2023 年 12 月 17 日

实验地点 E-II-311 实验批次 第 11 批

成 绩

指导教师评语：

指导教师：

____年____月____日

实验报告内容基本要求及参考格式

- 一、实验目的
- 二、实验所用仪器（或实验环境）
- 三、实验基本原理及步骤（或方案设计及理论计算）
- 四、实验数据记录（或仿真及软件设计）
- 五、实验结果分析及回答问题（或测试环境及测试结果）

实验四 可编程并行接口实验

1 实验目的

1. 了解可编程并行接口 8255 的内部结构；
2. 掌握工作方式、初始化编程及应用；

2 实验内容

1. 流水灯实验：利用 8255 的 A 口、B 口循环点亮发光二极管；
2. 交通灯实验：利用 8255 的 A 口模拟交通信号灯；
3. I/O 输入输出实验：利用 8255 的 A 口读取开关状态，8255 的 B 口把状态送发光二极管显示；
4. 在完成 (1) 基础上，增加通过读取开关控制流水灯的循环方向和循环方式；
5. 在完成 (2) 基础上，增加通过读取开关控制交通红绿灯的亮灭时间；

3 实验原理

8255 是一个通用可编程并行接口电路。它具有 A、B、C 三个 8 位并行口。其中 C 口也可用作 A、B 口的联络信号及中断申请信号。通过编程，它可以被设置为基本输入输出、选通输入输出以及双向传送方式，对于 C 口还具有按位置 0、1 的功能，其状态控制字见表 2。

表 2: 8255 控制字

D7	D6	D5	D4	D3	D2	D1	D0
1(特征位)	A 组方式 (00= 方式 0,01= 方式 1,1X= 方式 2)	A 口 (0= 输出, 1= 输入)	C 口高 4 位 (0= 输出, 1= 输入)	B 组方式 (0= 方式 0, 1= 方式 1)	B 口 (0= 输出, 1= 输入)	C 口低 4 位 (0= 输出, 1= 输入)	
0(特征位)		不用		位选择 (000=PC0...111=PC7)			0= 复位, 1= 置位

4 实验过程

4.1 实验电路的连接

模块的 WR、RD 分别连到 ISA 总线接口模块的 IOWR、IORD。模块的数据（AD0～AD7）、地址线（A0～A7）分别连到 ISA 总线接口模块的数据（LD0～LD7）、地址线（LA0～LA7）。8255 模块选通线 CE 连到 ISA 总线接口模块的 0000H。8255 的 PA0～PA7 连到发光二极管的 L0～L7；8255 的 PB0～PB7 连到发光二极管的 L8～L15。

4.2 流水灯实验程序设计与分析

在流水灯实验中，我们使用 CS 和 A_0 、 A_1 选择 8255 内部的三个 8 位并行口和控制寄存器。根据 8255 端口地址可知，其 A、B、C 口及控制寄存器地址分别为 0270H、0271H、0272H、0273H。在本次实验中仅用到了 A 口作为数据传输端口，以及控制寄存器端口进行 8255 的配置。

- 在基本流水灯控制实验中，设置 8255 工作模式为模式 0。由于实验要求是 A 口作为数据输出口，我们将控制字设置为 10000000，即 80H；
- 控制 A 口送出流水灯的状态编码，并进行输出延迟，通过控制流水灯状态编码的循环移位就可以实现流水灯从中间向两边依次点亮的效果；
- 在基本实验的基础上，我还额外设计了由反向流水灯构成的循环流水灯电路，根据 8255 工作原理，将流水灯的控制状态进行反向输出，结合上一步骤中正向流水灯控制，就可以实现流水灯从中间向两边以及从两边回到中间循环往复点亮的效果。

4.3 编写实验程序并编译链接运行

1	I08255 equ 0270h	14	assume cs:code, ds:_data, ss:_stack
2	I08255A equ 0270h	15	start proc near
3	I08255B equ 0271h	16	mov ax, _data
4	I08255ctl equ 0273h	17	mov ds, ax
5		18	mov dx, I08255ctl
6	_stack segment stack	19	mov al, 10000000B
7	dw 100 dup(?)	20	out dx, al
8	_stack ends	21	q1:
9		22	mov ax, 00e7h
10	_data segment	23	mov dx, I08255A
11	_data ends	24	call delay
12		25	out dx, ax
13	code segment	26	

27	<code>mov ax, 00dbh</code>	51	<code>mov ax, 00dbh</code>
28	<code>mov dx, I08255A</code>	52	<code>mov dx, I08255A</code>
29	<code>call delay</code>	53	<code>call delay</code>
30	<code>out dx, ax</code>	54	<code>out dx, ax</code>
31		55	
32	<code>mov ax, 00bdh</code>	56	<code>mov ax, 00e7h</code>
33	<code>mov dx, I08255A</code>	57	
34	<code>call delay</code>	58	<code>mov dx, I08255A</code>
35	<code>out dx, ax</code>	59	<code>call delay</code>
36		60	<code>out dx, ax</code>
37	<code>mov ax, 007eh</code>	61	<code>jmp q1</code>
38	<code>mov dx, I08255A</code>	62	
39	<code>call delay</code>	63	<code>delay proc</code>
40	<code>out dx, ax</code>	64	<code>xor cx, cx</code>
41		65	<code>loop \$</code>
42	<code>mov ax, 007eh</code>	66	<code>ret</code>
43	<code>mov dx, I08255A</code>	67	<code>delay endp</code>
44	<code>call delay</code>	68	
45	<code>out dx, ax</code>	69	<code>start endp</code>
46		70	<code>code ends</code>
47	<code>mov ax, 00bdh</code>	71	<code>end start</code>
48	<code>mov dx, I08255A</code>	72	
49	<code>call delay</code>	73	
50	<code>out dx, ax</code>	74	

4.4 验证电路功能与结果分析

编译链接项目后，运行程序在实验箱上即可观察到程序控制的 8 个二极管，对称地循环从中间向两侧依次点亮，当达到最外侧时，又从两侧向中间依次点亮，这个实验结果验证了程序的正确性与可行性。

5 实验心得与体会

在本次实验中，我将 8255 可编程并行接口理论知识成功应用于实践当中，并在实验环境中检验了程序的正确性与可行性。通过本次实验，我加深了对 8255 芯片用法的使用，积累了汇编编程经验，以及微机系统与外设进行数据传输的过程。这有助于我以后开展硬件操作与接口设计工作，为个人专业发展打下了坚实的基础。