

Университет ИТМО

## **Лабораторная работа №3**

Выполнил:  
студент III курса группы 3125  
Припадчев Артём

Проверит:  
Тропченко А.А.

Санкт-Петербург  
2014

Символьное (в ASCII) преобразование десятичного числа в шестнадцатеричное  
“590” → “0x24e”

### Программа в C51 Прямая адресация

```
#include <reg51.h>
int x; //переменная в регистровой памяти данных, имя
переменной подразумевает значение
char code y[ ]= "590"; //символьная константа в программной
памяти
char xdata yy[3]; //результат преобразования в расширенной
памяти

main()
{
    char i; //переменная в регистровой памяти
    x=0;
    for (i=0; i<3; i++) //перевод в двоичную
        x=x*10+(y[i]&0x0f);
    for(i=2;i>=0; i--)
    {
        yy[i]= (x&0x000f)+0x30; //преобразование в символы
        x=x>>4;
    }
    while(1); //динамический останов
}
```

### Листинг программы

```
0000 E4          CLR    A
0001 F500        R      MOV    x,A
0003 F500        R      MOV    x+01H,A
                                ; SOURCE LINE # 9
0005 F9          MOV    R1,A
                                ; SOURCE LINE # 10
0006 E9          MOV    A,R1
0007 FF          MOV    R7,A
0008 33          RLC    A
0009 95E0        SUBB   A,ACC
000B FE          MOV    R6,A
000C 7400        R      MOV    A,#LOW y
000E 2F          ADD    A,R7
000F F582        MOV    DPL,A
0011 7400        R      MOV    A,#HIGH y
0013 3E          ADDC   A,R6
0014 F583        MOV    DPH,A
0016 E4          CLR    A
0017 93          MOVC   A,@A+DPTR
0018 540F        ANL    A,#0FH
001A FB          MOV    R3,A
001B 33          RLC    A
001C 95E0        SUBB   A,ACC
001E FA          MOV    R2,A
001F AE00        R      MOV    R6,x
0021 AF00        R      MOV    R7,x+01H
0023 7C00        MOV    R4,#00H
0025 7D0A        MOV    R5,#0AH
0027 120000      E      LCALL  ?C?IMUL
002A EF          MOV    A,R7
002B 2B          ADD    A,R3
```

```

002C F500      R      MOV      x+01H,A
002E EE        MOV      A,R6
002F 3A        ADDC     A,R2
0030 F500      R      MOV      x,A
0032 09        INC      R1
0033 B903D0    CJNE     R1,#03H,?C0001
                                   ; SOURCE LINE # 11
0036 7902      MOV      R1,#02H
                                   ; SOURCE LINE # 12
                                   ; SOURCE LINE # 13
0038 E500      R      MOV      A,x+01H
003A 540F      ANL     A,#0FH
003C 2430      ADD     A,#030H
003E FF        MOV     R7,A
003F E9        MOV     A,R1
0040 FD        MOV     R5,A
0041 33        RLC     A
0042 95E0      SUBB    A,ACC
0044 FC        MOV     R4,A
0045 7400      R      MOV     A,#LOW yy
0047 2D        ADD     A,R5
0048 F582      MOV     DPL,A
004A 7400      R      MOV     A,#HIGH yy
004C 3C        ADDC    A,R4
004D F583      MOV     DPH,A
004F EF        MOV     A,R7
0050 F0        MOVX    @DPTR,A
                                   ; SOURCE LINE # 14
0051 E500      R      MOV     A,x+01H
0053 AE00      R      MOV     R6,x
0055 7804      MOV     R0,#04H
0057 CE        XCH    A,R6
0058 A2E7      MOV     C,ACC.7
005A 13        RRC     A
005B CE        XCH    A,R6
005C 13        RRC     A
005D D8F8      DJNZ   R0,?C0010
005F F500      R      MOV     x+01H,A
0061 8E00      R      MOV     x,R6
                                   ; SOURCE LINE # 15
0063 19        DEC     R1
0064 C3        CLR     C
0065 E9        MOV     A,R1
0066 6480      XRL    A,#080H
0068 9480      SUBB   A,#080H
006A 50CC      JNC    ?C0004
                                   ; SOURCE LINE # 16
006C 80FE      SJMP   ?C0007

```

Занимаемый размер 108 байт

### Программа на C51 – косвенный доступ через адрес-указатель

```

#include <reg51.h>
unsigned int x; //переменная в Data
unsigned char i;
char code * y="590"; //указатель на текстовую константу,
//имя переменной обозначает адрес

char xdata * yy; //указатель на текстовую переменную

```

```

main()
{
  for (i=0; i<3; i++)
    x=x*10+(*y++&0x0f);
  for(i=2;i>=0; i--)
  {
    *yy+= (x&0x000f)+0x30;
    x=x>>4;
  }
  while(1); //динамический останов
}

```

### Листинг программы

0000	E4		CLR	A
0001	F500	R	MOV	i,A
0003	0500	R	INC	y+01H
0005	E500	R	MOV	A,y+01H
0007	AE00	R	MOV	R6,y
0009	7002		JNZ	?C0010
000B	0500	R	INC	y
000D	14		DEC	A
000E	F582		MOV	DPL,A
0010	8E83		MOV	DPH,R6
0012	E4		CLR	A
0013	93		MOVC	A,@A+DPTR
0014	540F		ANL	A,#0FH
0016	FB		MOV	R3,A
0017	33		RLC	A
0018	95E0		SUBB	A,ACC
001A	FA		MOV	R2,A
001B	AE00	R	MOV	R6,x
001D	AF00	R	MOV	R7,x+01H
001F	7C00		MOV	R4,#00H
0021	7D0A		MOV	R5,#0AH
0023	120000	E	LCALL	?C?IMUL
0026	EF		MOV	A,R7
0027	2B		ADD	A,R3
0028	F500	R	MOV	x+01H,A
002A	EE		MOV	A,R6
002B	3A		ADDC	A,R2
002C	F500	R	MOV	x,A
002E	0500	R	INC	i
0030	E500	R	MOV	A,i
0032	B403CE		CJNE	A,#03H,?C0001
				; SOURCE LINE # 10
0035	750002	R	MOV	i,#02H
				; SOURCE LINE # 11
				; SOURCE LINE # 12
0038	E500	R	MOV	A,x+01H
003A	540F		ANL	A,#0FH
003C	2430		ADD	A,#030H
003E	FF		MOV	R7,A
003F	0500	R	INC	yy+01H
0041	E500	R	MOV	A,yy+01H
0043	AC00	R	MOV	R4,yy
0045	7002		JNZ	?C0011
0047	0500	R	INC	yy

```

0049 14          DEC      A
004A F582        MOV      DPL,A
004C 8C83        MOV      DPH,R4
004E EF         MOV      A,R7
004F F0         MOVX     @DPTR,A
                                ; SOURCE LINE # 13
0050 E500        R        MOV      A,x
0052 C4          SWAP     A
0053 F8          MOV      R0,A
0054 54F0        ANL      A,#0F0H
0056 C8          XCH      A,R0
0057 68          XRL      A,R0
0058 F500        R        MOV      x,A
005A E500        R        MOV      A,x+01H
005C C4          SWAP     A
005D 540F        ANL      A,#0FH
005F 48          ORL      A,R0
0060 F500        R        MOV      x+01H,A
                                ; SOURCE LINE # 14
0062 1500        R        DEC      i
0064 E500        R        MOV      A,i
0066 C3          CLR      C
0067 9400        SUBB     A,#00H
0069 50CD        JNC      ?C0004
                                ; SOURCE LINE # 15
006B 80FE        SJMP     ?C0007

```

Занимаемый размер 107 байт

### Программа на Ассемблере

Dseg at 8 ; сегмент данных в Data

Xseg at 0 ; сегмент данных в Xdata

Yy: ds 3 результат преобразования во внешней памяти

Cseg at 0 ; начало программного сегмента, после сброса или включения питания

Jmp start команда старта на начало программы

Y: db "590" ; текстовая константа в ASCII-коде

```

start:  clr a
        Mov DPL,#y
        clr a
        movc a,@a + dptr
        Anl a,#0x0f
        jz Mark1
        Mov r0,a

```

```

Cikl1:  clr a
        Add a,r3
        Add a, #100
        Mov b,a
        Anl a,#0x0f
        Mov r3,a
        Mov a,b
        Anl a,#0xf0
        Swap a
        Add a,r2

```

```
Mov b,a
Anl a,#0x0f
Mov r2,a
Mov a,b
Anl a,#0xf0
Swap a
Add a,r1
Mov r1,a
Djnz r0,cikl1
```

```
Mark1:
clr a
Inc dptr
movc a,@a + dptr
Anl a,#0x0f
jz Mark2
Mov r0,a
```

```
Cikl2:
clr a
Add a,r3
Add a, #10
Mov b,a
Anl a,#0x0f
Mov r3,a
Mov a,b
Anl a,#0xf0
Swap a
Add a,r2
Mov b,a
Anl a,#0x0f
Mov r2,a
Mov a,b
Anl a,#0xf0
Swap a
Add a,r1
Mov r1,a
Djnz r0,cikl2
```

```
Mark2:
clr a
Inc dptr
movc a,@a + dptr
Anl a,#0x0f
jz Mark3
Mov r0,a
```

```
Cikl3:
clr a
Add a,r3
Add a, #1
Mov b,a
Anl a,#0x0f
Mov r3,a
Mov a,b
Anl a,#0xf0
Swap a
```

```
Add a,r2
Mov b,a
Anl a,#0x0f
Mov r2,a
Mov a,b
Anl a,#0xf0
Swap a
Add a,r1
Mov r1,a
Djnz r0,cikl3
```

```
Mark3:
Mov DPL,#yy
Mov a,r1
movx @dptr,a
Inc dptr
Mov a,r2
movx @dptr,a
Inc dptr
movx @dptr,a
```

end

Занимаемый размер 128 байт