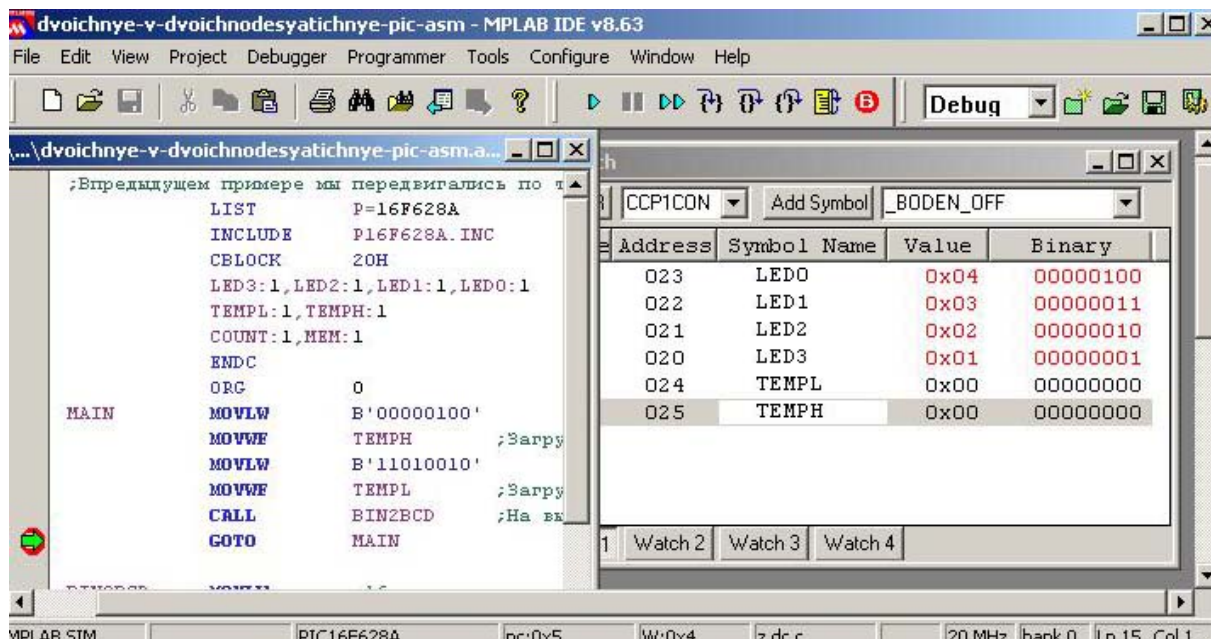


## Лабораторная работа №7 Двоичные в двоично-десятичные PIC ASM

**Цель:** Изучить процесс преобразования чисел в двоично-десятичный формат и вывода на семисегментный индикатор.

; В предыдущем примере мы передвигались по таблице числами от 0 до 9. Нам нужно преобразовать любое число в этот формат. Например, число 1234=0000 0100 1101 0010 - нужно сначала вывести 1 потом 2, 3 и 4.



```

LIST      P=16F628A
INCLUDE   P16F628A.INC
CBLOCK    20H
            LED3:1,LED2:1,LED1:1,LED0:1
            TEMPL:1,TEMPH:1
            COUNT:1,MEM:1
ENDC
ORG        0
MAIN MOV LW    B'00000100'
      MOV WF    TEMPH      ; Загружаем старший байт
      MOV LW    B'11010010'
      MOV WF    TEMPL      ; Загружаем младший байт
      CALL      BIN2BCD     ; На выходе в LED0-LED3 наше число в BCD
      GOTO      MAIN

BIN2BCD MOV LW    .16
      MOV WF    COUNT
      CLRF     LED0
      CLRF     LED1
  
```

```

        CLRF    LED2
        CLRF    LED3
LOOP8   RLF     TEMPL,F
        RLF     TEMPH,F
        RLF     LED0,F
        RLF     LED1,F
        DECFSZ  COUNT,F
        GOTO    ADJDEC

        SWAPF   LED1,W
        ANDLW   0FH
        MOVWF   LED3

        MOVFW   LED1
        ANDLW   0FH
        MOVWF   LED2

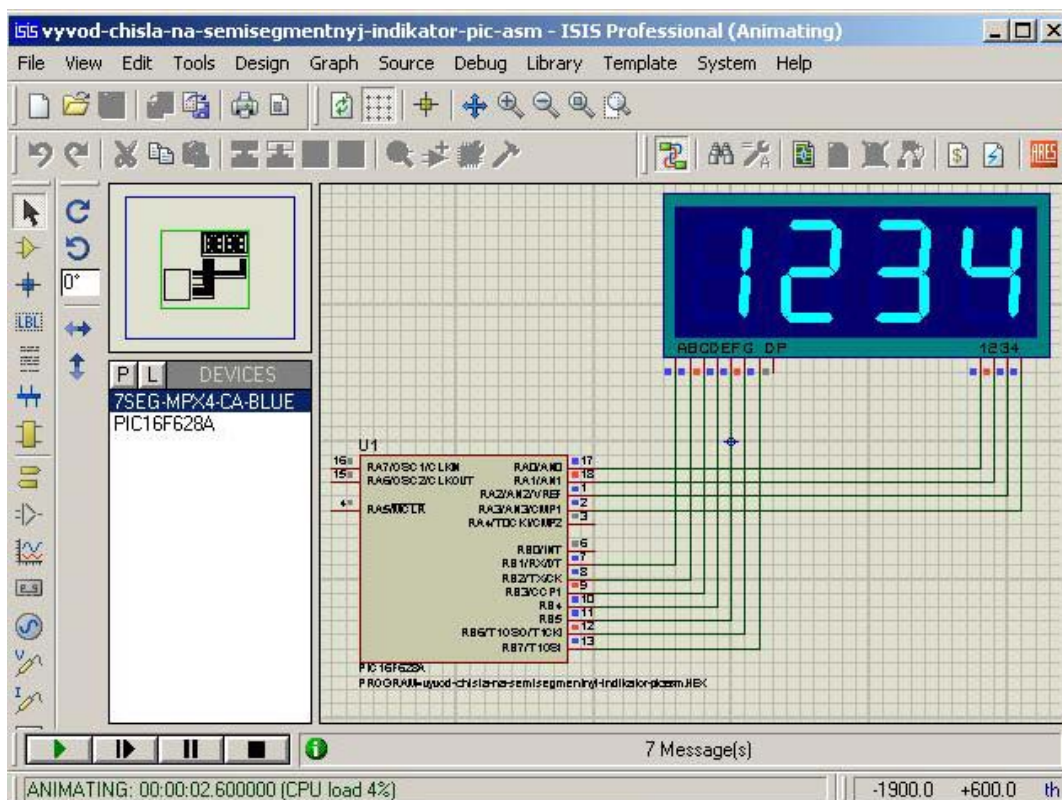
        SWAPF   LED0,W
        ANDLW   0FH
        MOVWF   LED1

        MOVFW   LED0
        ANDLW   0FH
        MOVWF   LED0
        RETURN

ADJDEC  MOVLW   LED0
        MOVWF   FSR
        CALL    ADJBCD
        MOVLW   LED1
        MOVWF   FSR
        CALL    ADJBCD
        GOTO    LOOP8
ADJBCD  MOVLW   3
        ADDWF   INDF,W
        MOVWF   MEM
        BTFSC   MEM,3
        MOVWF   INDF
        MOVLW   30H
        ADDWF   INDF,W
        MOVWF   MEM
        BTFSC   MEM,7
        MOVWF   INDF
        RETLW   0
        END
    
```

## Вывод числа на семисегментный индикатор PIC ASM

; Тут уже нормально выведем число на экран с помощью динамической индикации. То есть зажигаем один сегмент, тушим, зажигаем следующий словно кадры в кино. Можете в программе задержки заменить число 4 на 255 и увидите, как все на самом деле)



```

LIST      P=16F628A
INCLUDE   P16F628A.INC
__CONFIG __INTOSC_OSC_NOCLKOUT&_WDT_OFF&_PWRTE_ON&_M
CLRE_OFF&_LVP_OFF
ERRORLEVEL -302
CBLOCK    20H
    W_TEMP:1,STATUS_TEMP:1,FSR_TEMP:1
    J:1,K:1
    LED3:1,LED2:1,LED1:1,LED0:1
    TEMPL:1,TEMPH:1
    COUNT:1,MEM:1,CATOD:1,INDEX:1
ENDC
BANK0     MACRO
    BCF    STATUS,RP0
    BCF    STATUS,RP1
ENDM
BANK1     MACRO
    BSF    STATUS,RP0

```

```

        BCF      STATUS,RP1
ENDM
ORG      0
        GOTO     START
ORG      4
        GOTO     ISR
START    MOVLW    7
        MOVWF    CMCON
        CLRF     PORTB
        CLRF     PORTA
        BANK1
        MOVLW    B'11110000'
        MOVWF    TRISA
        MOVLW    B'00000001'
        MOVWF    TRISB
        BANK0

        MOVLW    B'00000100'
        MOVWF    TEMPH      ; Загружаем старший байт
        MOVLW    B'11010010'
        MOVWF    TEMPL      ; Загружаем младший байт
        CALL     BIN2BCD     ; На выходе в LED0-LED3 наше число

MAIN     MOVLW    LED3      ; Загружаем в FSR
        MOVWF    FSR        ; адрес последнего числа
        MOVLW    1          ; На 1 катод последнего сегмента
        MOVWF    CATOD      ; подаем напряжение, остальные
отключены
LOOP     MOVFW    INDF      ; Берем число по адресу загруженному в
FSR
        CALL     TABLE     ; Идем за кодом числа
        MOVWF    PORTB      ; Грузим его в порт
        MOVFW    CATOD      ; Берем значение Катода в W
        MOVWF    PORTA      ; Грузим в порт катода
        CALL     DELAY      ; Задержка
        CLRF     PORTA      ; Всё выключаем
        INCF     FSR,F       ; Переходим к следующему адресу
        RLF      CATOD,1     ; Двигаем 1 в право
        BTFSS    CATOD,4     ; Проверяем,у нас 4 катода
        GOTO     LOOP        ; Всё сначала 4 раза
        GOTO     MAIN

DELAY     MOVLW    .4
        MOVWF    J
JLOOP    MOVLW    .255

```

```

MOVWF    K
KLOOP    DECFSZ    K,F
GOTO     KLOOP
DECFSZ    J,F
GOTO     JLOOP
RETURN

```

```

TABLE    ADDWF     PCL,F
RETLW    B'10000000'
RETLW    B'11110010'
RETLW    B'01001000'
RETLW    B'01100000'
RETLW    B'00110010'
RETLW    B'00100100'
RETLW    B'00000100'
RETLW    B'11110000'
RETLW    B'00000000'
RETLW    B'00100000'

```

```

BIN2BCD  MOVLW     .16
MOVWF    COUNT
CLRF     LED0
CLRF     LED1
CLRF     LED2
CLRF     LED3
LOOP8    RLF       TEMPL,F
RLF      TEMPH,F
RLF      LED0,F
RLF      LED1,F
DECFSZ   COUNT,F
GOTO     ADJDEC

SWAPF    LED1,W
ANDLW    0FH
MOVWF    LED3

MOVFW    LED1
ANDLW    0FH
MOVWF    LED2

SWAPF    LED0,W
ANDLW    0FH
MOVWF    LED1

MOVFW    LED0

```

```
        ANDLW    0FH
        MOVWF    LED0
        RETURN
ADJDEC  MOVLW    LED0
        MOVWF    FSR
        CALL     ADJBCD
        MOVLW    LED1
        MOVWF    FSR
        CALL     ADJBCD
        GOTO     LOOP8
ADJBCD  MOVLW    3
        ADDWF    INDF,W
        MOVWF    MEM
        BTFSC    MEM,3
        MOVWF    INDF
        MOVLW    30H
        ADDWF    INDF,W
        MOVWF    MEM
        BTFSC    MEM,7
        MOVWF    INDF
        RETLW    0

ISR     MOVWF    W_TEMP
        SWAPF    STATUS,W
        BCF      STATUS,RP0
        MOVWF    STATUS_TEMP
        MOVFW    FSR
        MOVWF    FSR_TEMP

EXIT_ISR MOVFW    FSR_TEMP
        MOVWF    FSR
        SWAPF    STATUS_TEMP,W
        MOVWF    STATUS
        SWAPF    W_TEMP,F
        SWAPF    W_TEMP,W
        RETFIE
        END
```

### **Задания для выполнения:**

1. Изучить работу программ для перевода чисел в двоично-десятичную систему и вывода на семисегментный индикатор.
2. Добавить комментарии к каждой строке программ.
3. Смоделировать вывод чисел на семисегментный индикатор.

4. Оформить отчет о проделанной работе.