Watchdog Timer Configuration

The WDT is used to generate a variety of output signals after a user programmable count. The WDT is suitable for use in the prevention of system lock-up, such as when software becomes trapped in a deadlock. Under these sort of circumstances, the timer will count to zero and the selected outputs will be driven. Under normal circumstance, the user will restart the WDT at regular intervals before the timer counts to zero.

SAMPLE CODE:

```
;[]==================================================
; Name: Enable_And_Set_Watchdog
; IN     : AL - 1sec ~ 255sec
; OUT  : None
;[]==================================================
Enable_And_Set_Watchdog  Proc     Near
            push    ax                      ;save time interval
            call    Unlock_Chip

            mov     cl, 2Bh
            call    Read_Reg
            and     al, NOT 10h
            call    Write_Reg               ;set GP24 as WDTO

            mov     cl, 07h
            mov     al, 08h
            call    Write_Reg               ;switch to LD8
            mov     cl, 0F5h
            call    Read_Reg
            and     al, NOT 08h
            call    Write_Reg               ;set count mode as second

            pop     ax
            mov     cl, 0F6h
            call    Write_Reg               ;set watchdog timer

            mov     al, 01h
            mov     cl, 30h
            call    Write_Reg               ;watchdog enabled

            call    Lock_Chip
            ret
Enable_And_Set_Watchdog  Endp
```

```
;[]===============================================
; Name: Disable_Watchdog
; IN      : None
; OUT  : None
;[]===============================================
Disable_Watchdog     Proc     Near
               call      Unlock_Chip

               mov     cl, 07h
               mov     al, 08h
               call      Write_Reg               ;switch to LD8

               xor      al, al
               mov     cl, 0F6h
               call      Write_Reg               ;clear watchdog timer

               xor      al, al
               mov     cl, 30h
               call      Write_Reg               ;watchdog disabled

               call      Lock_Chip
               ret
Disable_Watchdog     Endp
;[]===============================================
; Name           : Unlock_Chip
; IN      : None
; OUT  : None
;[]===============================================
Unlock_Chip   Proc     Near
               mov     dx, 2Eh
               mov     al, 87h
               out      dx, al
               out      dx, al
               ret
Unlock_Chip   Endp
;[]================================================
; Name           : Lock_Chip
; IN      : None
; OUT  : None
;[]================================================
Unlock_Chip   Proc     Near
               mov     dx, 2Eh
               mov     al, 0AAh
               out      dx, al
               ret
Unlock_Chip   Endp
;[]================================================
; Name: Write_Reg
; IN      : CL - register index
;           AL - Value to write
; OUT  : None
;[]==================================================
Write_Reg       Proc    Near
                push    ax
                mov     dx, 2Eh
                mov     al,cl
                out     dx,al
                pop     ax
                inc     dx
                out     dx,al
                ret
Write_Reg       Endp
;[]==================================================
; Name: Read_Reg
; IN      : CL - register index
; OUT  : AL - Value to read
;[]=====================================================
Read_Reg        Proc    Near
                mov     al, cl
                mov     dx, 2Eh
                out     dx, al
                inc     dx
                in      al, dx
                ret
Read_Reg        Endp
;[]==================================================
```