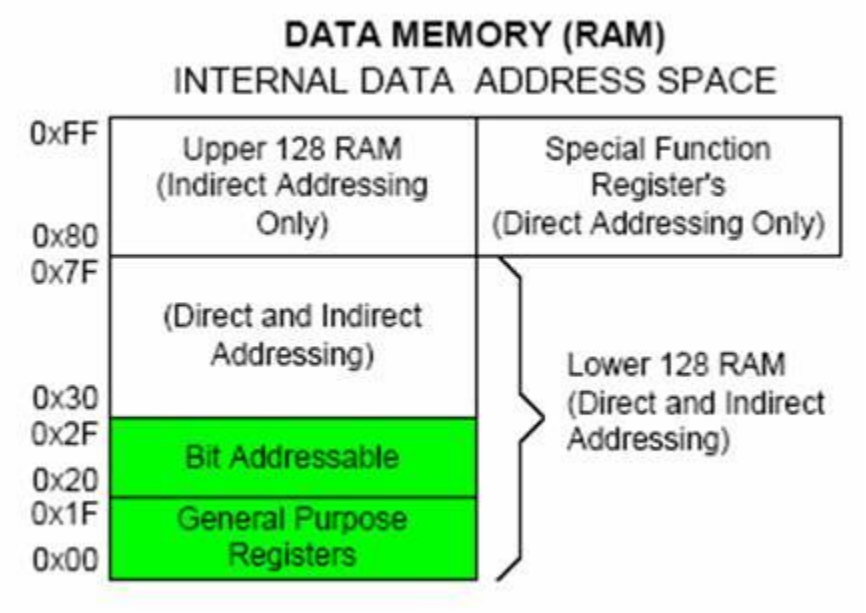
Accessing SFRs and RAM



The Watchdog Timer - a timer that resets the processor when it overflows unless the program resets the timer. Meant to prevent systems from running out of control.

- Can be disabled for debugging using the Watchdog Timer Control Register

Figure 13.3. WDTCN: Watchdog Timer Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								xxxxx111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xFF

Bits7-0: WDT Control
Writing 0xA5 both enables and reloads the WDT.
Writing 0xDE followed within 4 system clocks by 0xAD disables the WDT.
Writing 0xFF locks out the disable feature.

Bit4: Watchdog Status Bit (when Read)
Reading the WDTCN.[4] bit indicates the Watchdog Timer Status.
0: WDT is inactive
1: WDT is active

Bits2-0: Watchdog Timeout Interval Bits
The WDTCN.[2:0] bits set the Watchdog Timeout Interval. When writing these bits, WDTCN.7 must be set to 0.

```
; Disable watchdog timer
Main:     mov 0xFF, #0DEh
          mov 0xFF, #0ADh
```

The actual WDT is a 21-bit timer.

Exercise:

Write a program from scratch that:

1. Disables the watchdog timer.
2. Selects register bank 0.
3. Writes the following hex values to the upper bytes of RAM using register indirect mode.
   Address: Data:
   0x88 0x80
   0x89 0xFF
   0x8E 0x33
   Be sure NOT to write to the corresponding SFRs!

Save your program as ram_write.asm. Compile, build, and download the program.

Open the debug window for SFR register "Timers". (View - Debug Windows – SFR's - Timers) Note that the top SFRs have these addresses:

TCON 0x88
TMOD 0x89
CKCON 0x8E

Open the RAM debug window. Step through the program and verify that the RAM is being changed, and not the SFRs. Be sure you understand why!