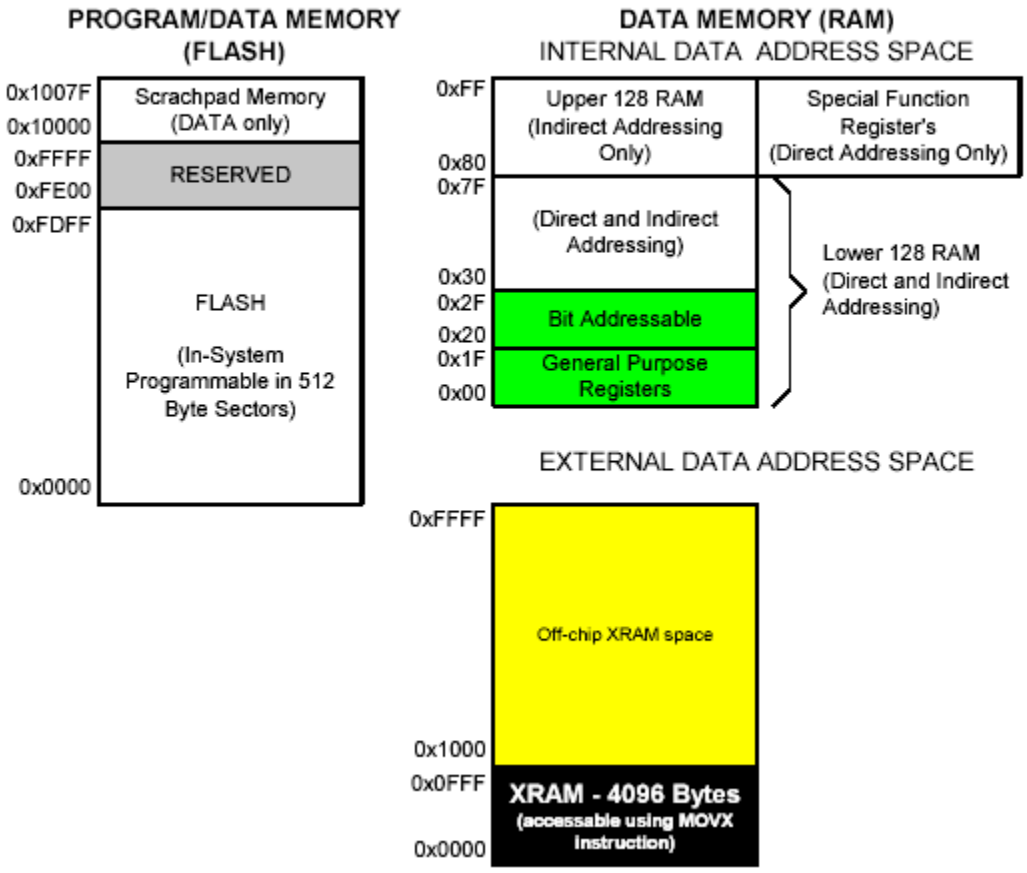
Registers and Memory

Figure 1.7. On-Chip Memory Map



Replace the "Main" routine of the blink program with the following code:

```
Main:; Disable the WDT. (IRQs not enabled at this point.)
; If interrupts were enabled, we would need to explicitly
disable
; them so that the 2nd move to WDTCN occurs no more than
four clock
; cycles after the first move to WDTCN.

mov WDTCN, #0DEh
mov WDTCN, #0ADh

; Set a register to a value and then move that value around
to free storage places
```

```
mov R0, #0x55      ; set register R0 to 55 hex
mov a, R0          ; move it to the Accumulator
mov b, a           ; move it to the b register
mov 0x35, a        ; move it to a byte in the scratchpad
[30h-7Fh]
mov dptr, #0x2000  ; set data pointer value
movx @dptr, a      ; move a to external memory at address
2000h
sjmp $             ; get stuck here at the end
;-------------------------------------------------------
-----
```

Save the program in the H: drive as **reg_mem.asm**

Add the program to the intro project, add it to the build, and remove the blink.asm from the build.

Open the following debug windows:

- 8051 (internal registers)
- Regs
- RAM
- External RAM (use View -Debug Windows - External Memory)

Assemble, build, and download the program.

Step through the program and verify that the data gets moved to the locations you expect.

Modify the program so that the data content being moved is 25 rather than 85 (55h).

Assemble, build, download and step through the program.