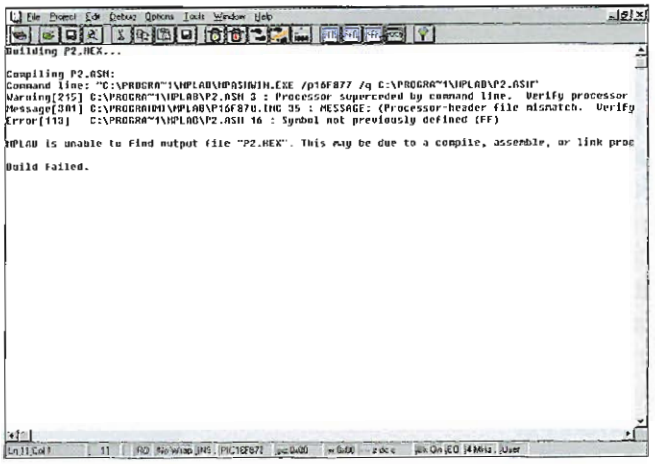
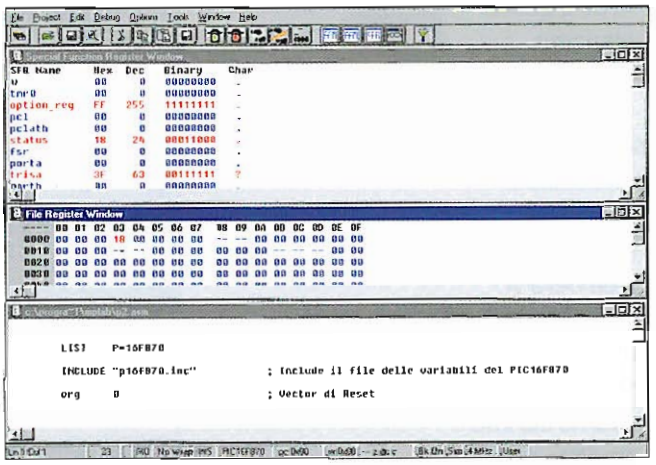
significato, ma per il momento vedremo solo i primi tre: 0, 1 e 2, che sono quelli coinvolti nelle operazioni aritmetico-logiche.

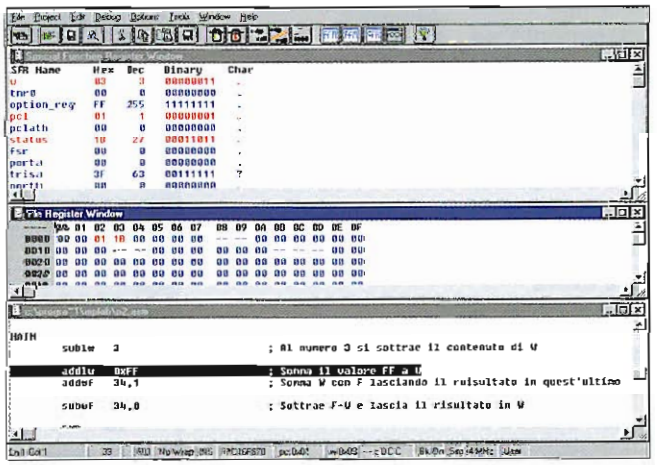
Il bit Z viene impostato a 1 quando il risultato dell'operazione, sia che si tratti di una somma o di una sottrazione, è 0. Invece il funzionamento degli altri due bit è differente a seconda se l'operazione è di somma o di sottrazione. Il bit C è il bit di carry o bit di riporto. Se l'operazione è di somma, questo bit si porrà a 1 quando c'è riporto, e si porrà a 0 se il risultato è contenuto in un registro da 8 bit o, in altre parole, se non c'è riporto. Se invece l'operazione è di sottrazione, questo bit funziona al rovescio; infatti se si vuole sottrarre A – B e A → B, ci sarà un 'avanzo', e il bit s'imposta a 1 e, se al contrario, A ← B il bit andrà a 0 perché ci sarà una 'mancanza'. Il bit DC funziona come C però per i numeri BCD.



Videata di errore se compiliamo il progetto senza cambiare i parametri.



Finestre da aprire per la simulazione.



F7 premuto la prima volta.

Operazioni multiple e complesse

In assembler, se si vuole sommare il valore di diversi registri bisogna farlo a poco a poco. Per prima cosa si muoverà un registro su W e si sommerà questo valore con un altro registro, memorizzando il risultato in W e sommando il registro successivo, e via di seguito.

La stessa cosa avviene con la sottrazione. Nel caso di operazioni più complesse di quelle della somma e della sottrazione, come possono essere quelle di moltiplicazione e divisione, non possiamo eseguirle direttamente perciò le dobbiamo scomporre in operazioni fondamentali. In questo modo, una moltiplicazione non è altro che un ciclo dove si ripete la somma del dato del moltiplicando per il numero di volte che indica il moltiplicatore. La stessa cosa si può fare per la divisione utilizzando sottrazioni invece di somme. Potremo applicare questo principio anche per altri tipi di operazioni, sebbene più complesse.

Prova delle espressioni in MPLAB

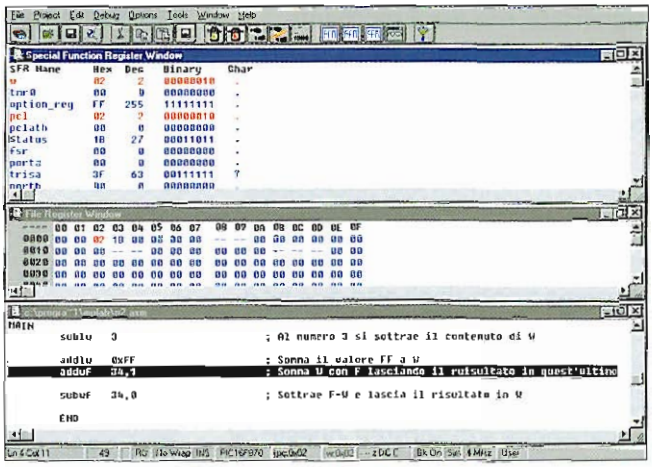
Verifichiamo ora le istruzioni viste in MPLAB. Copiate il programma della figura, create un progetto e fatene una coppia. Se ora compilate

il progetto vi apparirà una finestra come quella mostrata, dato che ci sono parametri che non abbiamo cambiato. Cambiate il modo di sviluppo, perché si adatti al PIC16F870 in simulazione. L'altro errore che appare è la confusione del valore "FF" esadecimale con un'etichetta non definita. Sostituitelo con 0xFF perché venga interpretato come un numero. Compilate nuovamente; a questo punto non dovrebbero più esserci errori. Per eseguire la simulazione aprite la finestra dei registri generali e quella dei registri specifici. Eseguiamo le istruzioni una alla volta verificando come si modificano i registri. L'opzione da scegliere è Debug →Run →Step, il cui tasto rapido è F7.

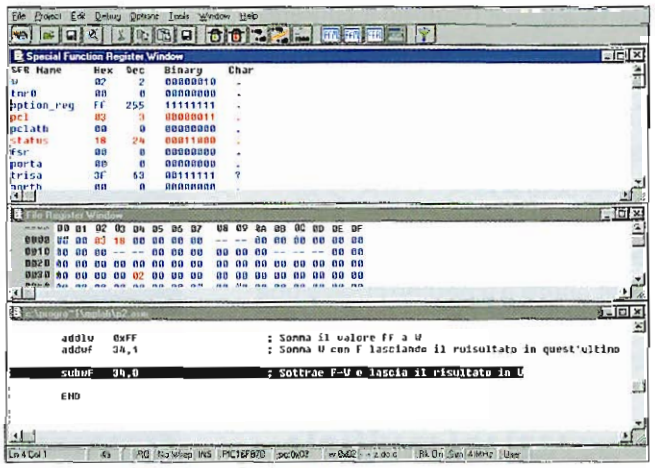
Premete F7: dovranno apparire i dati della videata. La linea nera vi segnala la prossima istruzione che verrà eseguita, quindi dovremo interpretare le istruzioni precedenti per vedere se comprendiamo ciò che è stato modificato. È stata fatta una sottrazione di 3-W e il risultato è stato lasciato in W. Effettivamente, se verifichiamo i registri che appaiono in rosso, ovvero quelli modificati con l'ultima istruzione, in W ora abbiamo un 3. Quindi, dato che non c'è riporto, il bit 0 (C) ha valore 1.

Premete nuovamente F7. Sommeremo FF a W (che aveva valore 3), produrremo come risultato 02 e avremo riporto, così come mostrano i registri. Il registro STATUS in questo caso non appare in rosso dato che conserva il valore. In seguito, premendo nuovamente F7, si sommerà il valore di W (02) con il contenuto del registro 34, che in precedenza aveva valore 0; il risultato verrà memorizzato nel registro 34. Nella finestra dei registri generali appare tutto questo. Non si produce riporto, quindi C torna a 0.

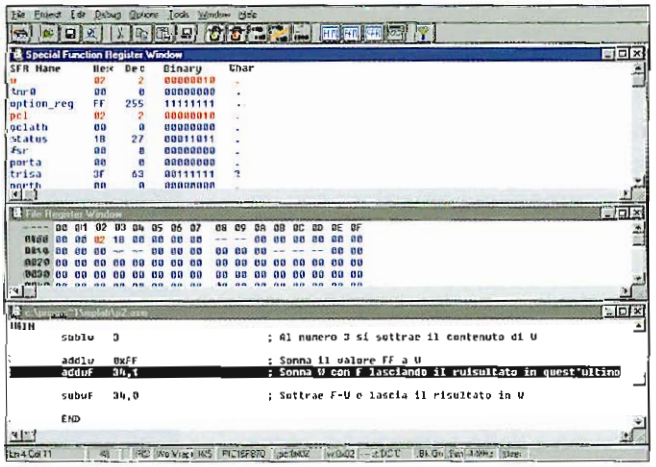
F7 premuto la seconda volta.



F7 premuto la terza volta.



F7 premuto l'ultima volta.



Inoltre, notate che W non perde il suo valore, e siccome non ha subito modifiche appare in azzurro. Premendo l'ultima volta F7, si sottrae il valore del registro 34 al valore di W; la linea nera che segnala le istruzioni non cambia perché non c'è più codice. Dato che W e il registro 34 hanno lo stesso dato (02) la somma dà come risultato 0, quindi il bit che ora si modifica è Z. Anche in questo caso non c'è riporto, C si pone a 1. W cambia di valore ma non il registro 34.