Breakpoint in ASIM

Breve guida all'utilizzo dei breakpoint in ASIM per il testing del software

Raffaele del Gaudio

preparato per il corso di CSD del prof. Nicola Mazzocca

L'obiettivo di questa guida è illustrare come sia possibile inserire dei "breakpoint" nel software scritto per il simulatore ASIM. La necessità dei breakpoint sovviene quando si vuole effettuare testing di architetture software con più nodi considerando che il simulatore ASIM non è capace di mandare avanti l'esecuzione di una macchina alla volta.

Con questo metodo sarà possibile emulare il comportamento di un breakpoint reale, riuscendo a mettere in pausa (e a far ripartire) l'esecuzione di specifici nodi software dell'architettura.

1 Cosa aggiungere nel software

Il meccanismo software consiste di una variabile byte e di una funzione che ritorna solo quando la variabile in questione contiene 0 al suo interno.

Il seguente codice va inserito in tutti i nodi software che si intende abilitare al debugging mediante breakpoint.

Listing 1: Strutture software necessarie per il debugging

```
1                   ORG     $9900
2   CONTINUE        EQU     0
3   STOP            EQU     1
4   debug_var       DC.B    STOP
5
6                   ORG     $9902
7   BREAKPOINT
8                   CMPI.B  #CONTINUE,debug_var
9                   BNE     BREAKPOINT
10                  RTS
```

In seguito a questa aggiunta è possibile inserire i breakpoint nel proprio codice da testare sottoforma di chiamata alla routine `BREAKPOINT`. Di seguito un esempio di breakpoint inserito alla riga 2 del programma.

Listing 2: Esempio di utilizzo nel codice del breakpoint

```
1   MOVE.B  PIA1_DATA,D2
2   JSR     BREAKPOINT * <- BREAKPOINT SULLA VARIABILE debug_var
3   MOVE.B  (A0,D1),PIA1_DATA
4   ADDQ.W  #1,D1
5   BRA     WAIT_SEND
```

2 Cosa fare durante l'esecuzione

Come è intuibile, alla chiamata della funzione `BREAKPOINT` il nodo software rimane in un ciclo finché la variabile `debug_var` non contiene il valore 0. Segue che l'utilizzatore dovrà manualmente modificare la memoria alla locazione `$9900`[1] dello specifico nodo inserendo 0 quando si vuole far continuare l'esecuzione ed 1 quando si vuole che l'esecuzione si blocchi alla prossima chiamata alla funzione `BREAKPOINT`.

In fig. 1 viene mostrato dove è presente il menù per modificare una locazione di memoria durante un'esecuzione mentre in fig. 3 cosa si deve settare per rendere bloccante il breakpoint e in fig. 2 come fare per far proseguire l'esecuzione.

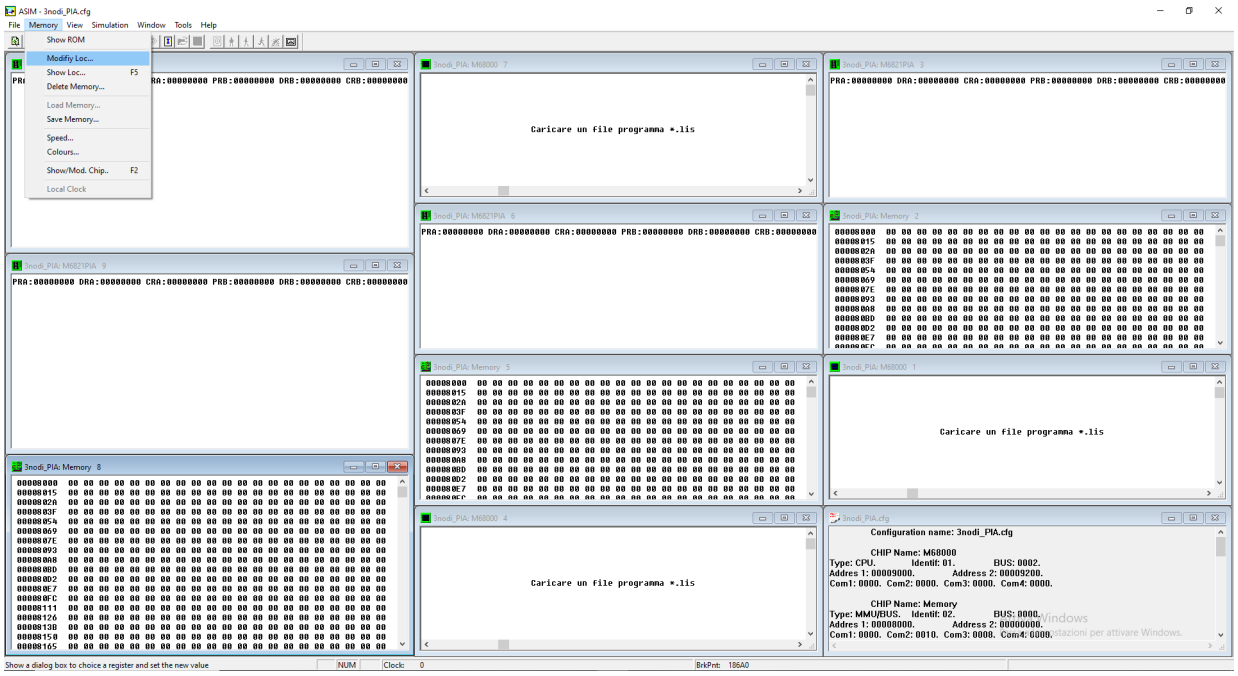

Figure 1: Menù modifica memoria

[1]Qualora venga modificato l'indirizzo di memoria in cui viene posizionato il codice del Listing 1, anche questo indirizzo cambierà.

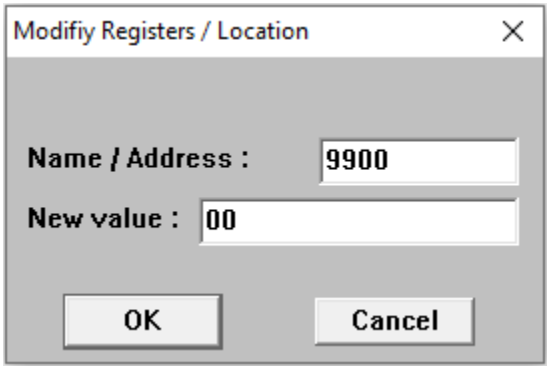

Figure 2: Setting della memoria a 0 per far proseguire l'esecuzione

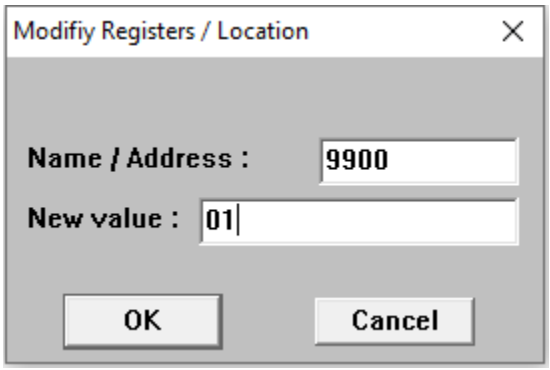

Figure 3: Setting della memoria a 1 per mettere in pausa l'esecuzione