

L A s s e m b l e r 8 0 8 6

Istruzioni per il Controllo del Processore

M. Rebaudengo - M. Sonza Reorda

Politecnico di Torino
Dip. di Automatica e Informatica

1

M. Rebaudengo, M. Sonza Reorda

Istruzioni per il Controllo del Processore

Servono a regolare il funzionamento del processore attraverso comandi software.

Si dividono nelle seguenti categorie:

- istruzioni per la gestione delle interruzioni
- istruzioni sui flag
- istruzioni di sincronizzazione
- istruzione nulla.

2

M. Rebaudengo, M. Sonza Reorda

Istruzioni di Interrupt

Permettono di gestire gli Interrupt *software*.

L'arrivo di un Interrupt causa:

- il salvataggio nello stack dei registri IP e CS
- il salvataggio, sempre nello stack, del Flag Register
- l'esecuzione di una routine di servizio dell'Interrupt (*Interrupt Service Routine* o *ISR*).

3

M. Rebaudengo, M. Sonza Reorda

Differenze Interrupt-Procedure

Il meccanismo di Interrupt è simile a quello di chiamata a procedura. Esistono tuttavia delle differenze:

- mentre l'esecuzione di una procedura può essere avviata solo via software, quella di una *ISR* può esserlo anche via hardware
- mentre una procedura può essere *NEAR* o *FAR*, una *ISR* è sempre *FAR*, in quanto la sua attivazione comporta comunque il caricamento di un valore in CS
- un Interrupt causa il salvataggio nello stack, oltre che dei registri CS e IP, anche del Flag Register.

4

M. Rebaudengo, M. Sonza Reorda

Interrupt Vector

L'indirizzo della routine di servizio dell'Interrupt viene ottenuto da una tabella allocata agli indirizzi bassi della memoria (locazioni da 0000:0000h a 0000:03FFh); ogni elemento di tale tabella, detta *Interrupt Vector*, contiene l'indirizzo di partenza, su 32 bit, della ISR relativa ad uno dei 256 Interrupt riconosciuti.

IP ₀	00000
CS ₀	
IP ₁	00004
CS ₁	
IP ₂	00008
CS ₂	

5

M. Rebaudengo, M. Sonza Reorda

Istruzione INT

Causa l'attivazione della routine di servizio relativa ad un certo Interrupt.

Formato

INT interrupt-type

dove *interrupt-type* identifica l'Interrupt, ossia il numero d'ordine nell'Interrupt Vector della ISR che si desidera attivare all'interno delle 256 ammesse dal sistema.

L'Indirizzo Fisico cui accedere all'interno dell'Interrupt Vector viene ottenuto semplicemente moltiplicando per 4 il parametro *interrupt-type*.

6

M. Rebaudengo, M. Sonza Reorda

INT (II)

Le operazioni causate da una INT sono, nel dettaglio, le seguenti:

- salvataggio nello stack del Flag Register
- azzeramento dei flag TF (Trap Flag) e IF (Interrupt Enable/Disable)
- salvataggio nello stack del registro CS
- caricamento in CS della seconda word dell'Interrupt Vector
- salvataggio nello stack del registro IP
- caricamento in IP della prima word dell'Interrupt Vector.

7

M. Rebaudengo, M. Sonza Reorda

Istruzione IRET

È l'istruzione conclusiva di una ISR.

Causa il ritorno del sistema nello stato precedente all'ultima INT e l'esecuzione delle istruzioni ad essa successive nel programma interrotto.

Non ha operandi.

Provoca le seguenti operazioni:

- preleva dallo stack il valore di IP
- preleva dallo stack il valore di CS
- preleva dallo stack il valore del Register Flag.

8

M. Rebaudengo, M. Sonza Reorda

Istruzione INTO

Equivale ad una INT 4 quando l'Overflow Flag (OF) è settato, altrimenti equivale ad un'istruzione nulla.

Non ha operandi.

Viene utilizzata quando è necessario controllare la validità del risultato di un'istruzione.

9

M. Rebaudengo, M. Sonza Reorda

Istruzioni di sincronizzazione

Permettono all' 8086/8088 di sincronizzarsi con dispositivi esterni:

- HLT
forza il processore nello stato *idle*, in cui non esegue nessuna istruzione e da cui esce solo quando riceve un Interrupt esterno
- WAIT
forza anch'essa il processore nello stato *idle*, ma con due differenze rispetto alla precedente:
 - al termine di eventuali routine di servizio dell'Interrupt, il processore torna nello stato *idle*
 - ogni 5 colpi di clock viene testata la linea TEST: quando questa diventa attiva il processore procede all'esecuzione dell'istruzione successiva alla WAIT.

10

M. Rebaudengo, M. Sonza Reorda

- **ESC**
usata per inviare istruzioni al coprocessore matematico 8087.
- **LOCK**
non è un'istruzione vera e propria, ma un prefisso da premettere ad una istruzione, quando si vuole che nessun altro dispositivo utilizzi il bus fin tanto che essa non è stata completamente eseguita. Viene utilizzata, ad esempio, in sistemi multi-processore per istruzioni che facciano accesso a risorse critiche (memorie condivise).

11

M. Rebaudengo, M. Sonza Reorda

Istruzione NOP

Non esegue nessuna operazione, se non quella di far avanzare l'*Instruction Pointer* all'istruzione successiva.

Sue possibili applicazioni sono:

- occupare il posto di una istruzione cancellata quando si vuole modificare il codice macchina senza doverlo ri-assemblare;
- introdurre un periodo di attesa di durata nota prima dell'esecuzione dell'istruzione successiva.

12

M. Rebaudengo, M. Sonza Reorda

Istruzioni sui Flag

Permettono di settare, resettare e complementare alcuni flag.

Sono:

- **STC, CLC, CMC:** settano, resettano e complementano il Carry Flag
- **STD e CLD:** settano e resettano il Direction Flag
- **STI e CLI:** settano e resettano l'Interrupt Flag.

13

M. Rebaudengo, M. Sonza Reorda

INT 21

Il DOS mette a disposizione del programmatore una serie di procedure di sistema per eseguire varie operazioni, tra cui l'input da tastiera e l'output su video di singoli caratteri.

L'accesso a tali procedure avviene attraverso l'istruzione **INT 21**. L'operazione richiesta viene specificata scrivendo un opportuno valore nel registro **AH**.

14

M. Rebaudengo, M. Sonza Reorda

Input da tastiera

- **AH=1:** aspetta che sulla tastiera venga battuto un tasto, lo visualizza, e ne ritorna il codice ASCII in **AL**
- **AH=6;** come il precedente, ma non è sensibile al **CTRL-BREAK**
- **AH=7:** come il primo, ma non è sensibile al **CTRL-BREAK**, ed il carattere non viene visualizzato.
- **AH=8:** come il precedente, ma è sensibile al **CTRL-BREAK**

15

M. Rebaudengo, M. Sonza Reorda

Input da tastiera (II)

- **AH=0AH:** legge una stringa (terminata da **<CR>**) e la mette in un buffer, il cui indirizzo è preso in **DS:DX**; nel primo carattere deve essere stata preventivamente scritta la lunghezza del buffer, nel secondo viene scritta la lunghezza della stringa letta.

16

M. Rebaudengo, M. Sonza Reorda

Output su video

- **AH=2:** visualizza il carattere il cui codice ASCII è contenuto in DL; esegue il controllo su CTRL-BREAK
- **AH=5:** scome il precedente, ma senza il controllo su CTRL-BREAK
- **AH=9:** visualizza la stringa il cui indirizzo è contenuto in DS:DX, e il cui termine è segnalato dal carattere '\$'.

17

M. Rebaudengo, M. Sonza Reorda

Data e Ora

- **AH=2AH:** acquisisce la data, scrivendo in CX il numero dell'anno, in DH quello del mese, in DL quello del giorno
- **AH=2CH:** acquisisce l'ora, scrivendo in CH l'ora, in CL i minuti, in DH i secondi, in DL i centesimi di secondo.

18

M. Rebaudengo, M. Sonza Reorda