

ASSEMBLER ASSEMBLER ASSEMBLER ASSEMBLER

8086 8088

di Pierluigi Panunzi

La struttura interna dei due microprocessori

A dispetto del titolo della rubrica, in questa puntata parleremo ancora poco di Assembler, ma ciò è spiegabile in quanto desideriamo far conoscere ai lettori questa coppia di microprocessori «gemelli» a cominciare dalle loro caratteristiche interne: sappiamo di soddisfare così i desideri dei lettori interessati a questioni vicine all'hardware. I programmatori troveranno «pane per i loro denti» a cominciare dalla prossima puntata.

L'8086: uno sguardo al suo interno

Visto da fuori, se non ci fosse la dicitura che lo identifica, l'8086 si potrebbe benissimo confondere con un qualsiasi altro integrato a 40 piedini, di quelli che, in numero di tre-quattro, troviamo all'interno del nostro personal computer ed in genere di prestazioni nettamente differenti per non dire minori. Nella figura 1 abbiamo riportato il significato dei vari piedini del circuito integrato ed ora passeremo ad analizzarli brevemente. Molti di questi sono ben noti, in quanto presenti praticamente in ogni microprocessore che si rispetti: nel nostro caso troveremo viceversa nuovi pin, in virtù delle innovazioni apportate dal nostro microprocessore.

Ecco che sul lato sinistro del micro troviamo una lunga sequenza di pin indicati come «AD»: si tratta, come è facilmente intuibile, del bus degli indirizzi e dei dati (AD sta per Address-Data), che già sappiamo essere per l'appunto posti in piedini comuni: ricordiamo che questa condivisione («multiplexing») è solo a livello fisico, in quanto «temporalmente» e cioè in istanti differenti si ha una totale differenza di comportamento ed i pin risultano a tutti gli effetti o solo Address

Bus o solo Data Bus. All'interno dell'8086 ci sono infatti degli appositi selettori che permettono di «affacciare» allo stesso pin due circuiterie logiche aventi funzioni nettamente differenti: basti pensare che i piedini possono essere, in istanti di tempo differenti, o di sola uscita (Address Bus) oppure bidirezionali (Data Bus), fatto che già da solo implica una complessità circuitale non indifferente. Evidentemente questa «suddivisione temporale» dei compiti viene segnalata all'esterno da un certo numero di pin, importantissimi in quanto permettono di pilotare senza errori i componenti esterni i quali non si accorgono nemmeno di quanto sta accadendo all'interno del microprocessore.

I pin in questione sono: \overline{RD} , \overline{WR} , M/\overline{IO} , DT/\overline{R} , DEN e ALE .

In particolare quest'ultimo è quello che serve, per mezzo di appositi circuiti esterni, a «congelare» l'indirizzo contenuto in un ben determinato istante nell'Address-Data Bus (il nome ALE sta infatti per «Address Latch Enable»); viceversa il segnale DEN («Data ENable») indicherà ai circuiti esterni che il Bus in quell'istante contiene un dato a 16 bit: a seconda dello

stato degli altri segnali, tale dato verrà trasmesso o ricevuto dall'8086 (pin DT/\overline{R} , «Data Transmit/Receive») dalla memoria o da una porta di I/O (pin M/\overline{IO} , «Memory/Input Output») per un'operazione di lettura (pin \overline{RD} , «ReaD») o di scrittura (pin \overline{WR} , «WRite»).

Il tutto, come già abbiamo detto nelle precedenti puntate, prende le mosse dal microprocessore 8085, nel quale tali pin erano praticamente già presenti ed al quale i progettisti dell'Intel hanno preso spunto.

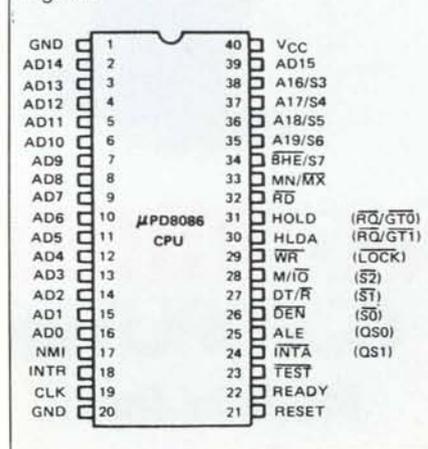
A differenza però dell'8085 (ricordiamoci che l'8086 è un «16 bit») questa procedura di scambio di segnali da e verso l'esterno è attivata solo se sul pin MN/\overline{MX} («MiNimum/MaXimum mode») è presente un livello logico «1», ad indicare che si opera in «modo minimo». Al contrario, con un livello «0» nel pin indicato, il microprocessore può funzionare nel cosiddetto «Maximum mode» (ovviamente assente nell'8085), grazie al quale è svincolato dall'onere di gestire i vari segnali, demandando tale compito ad un integrato di supporto (l'8288 dell'Intel) al quale invia semplicemente l'informazione dello «stato» in cui si trova, grazie a tre appositi pin ($S0$, $S1$ ed $S2$).

Sarà appunto l'8288 a decodificare tale stato ed a generare gli opportuni segnali di interscambio.

Dalla figura 1 vediamo dunque che in «Maximum mode» alcuni piedini cambiano totalmente significato: un esempio l'abbiamo già visto con i tre segnali di stato ed ora ci occuperemo brevemente degli altri.

Ciò che dobbiamo avere ben presente in questo momento è che l'8086 può già lavorare in un sistema multiprocessore, in cui vi sono appunto più processori affacciati su uno stesso Bus logico: ora, secondo ben determinate strategie, i processori possono a turno «guadagnare» l'accesso a tale Bus, fat-

Figura 1



to del quale l'8086 viene a conoscenza in base allo stato dei pin $\overline{RQ}/\overline{GT0}$ e $\overline{RQ}/\overline{GT1}$; viceversa l'8086 può effettuare una «chiusura» del bus agli altri processori (in termine tecnico «lock»), grazie ad un'istruzione particolare (la LOCK) strettamente legata all'omonimo pin, per mezzo del quale si può segnalare all'esterno la «presa di potere» da parte dell'8086.

Proseguendo dunque nell'analisi dei piedini, troviamo altri quattro bit di indirizzo (A16... A19), con i quali si ottiene la possibilità di indirizzare fino ad 1 Mbyte di memoria: tralasciamo a questo punto il fatto che anche questi piedini possiedono una seconda funzione...

Altri cinque pin svolgono altrettante funzioni fondamentali e presenti per l'appunto in tutti i microprocessori:

— CLK è il pin al quale viene connesso il circuito di temporizzazione esterno («CLock»);

— INTR, «INTerrupt Request», è il pin al quale vengono convogliate le richieste di interrupt da parte di componenti esterni, secondo una particolare procedura sulla quale ritorneremo a tempo debito;

— INTA, «INTerrupt Acknowledge», è il pin di uscita per mezzo del quale l'8086 comunica all'esterno la sua disponibilità ad essere «interrotto»;

— RESET è il pin tramite il quale si resetta il microprocessore, forzandolo ad eseguire una certa routine posta all'indirizzo FFFF0H;

— NMI, «Non Maskable Interrupt», è un pin presente anche in molti altri processori e si riferisce alla possibilità dall'esterno di inviare un «interrupt non mascherabile» (è perciò in un certo senso «inevitabile» per il microprocessore), in generale utilizzato per la gestione di particolari eventi esterni (ad esempio l'abbassamento della tensione di alimentazione, ecc.).

Infine particolari funzioni sono svolte dai due pin TEST e READY, entrambi in input, che permettono la sincronizzazione dell'8086 con particolari eventi esterni: il primo agisce in combinazione con una istruzione assembler, la WAIT, la quale interrompe l'elaborazione del programma in cui viene posta, facendo entrare il processore in «stato di wait», fino a che il segnale presente sul pin TEST non ritorna a «0».

Il secondo pin, READY, serve invece ai dispositivi di memoria o di I/O particolarmente lenti, per segnalare al microprocessore che l'operazione loro assegnata non è ancora finita: classico è l'esempio dell'8086 con clock molto elevato e con la disposizione RAM ad accesso viceversa lento. In questo caso si può «rallentare» l'8086 solo laddove le temporizzazioni sono critiche e

cioè durante gli accessi alla memoria, soprattutto nella fase di lettura, il cui tempo di esecuzione è strettamente legato al componente impiegato per implementare la RAM.

Detto questo abbiamo terminato l'analisi dei pin, incompleta nel senso che ne abbiamo volutamente trascurati alcuni per non appesantire ulteriormente l'articolo, ed andiamo ad analizzare la struttura interna del nostro microprocessore a 16 bit.

La struttura logica interna dell'8086

Facendo riferimento alla figura 2, possiamo vedere che il microprocessore è suddiviso in tre parti logiche fondamentali: l'«Execution Unit», «Bus Interface Unit» e «Control & Timing», rispettivamente identificati nel seguito con le loro sigle, EU, BIU e CT. Cominciamo dunque dalla prima, l'EU, fondamentalmente la stessa che si trova in tutti i microprocessori, variando in pratica a seconda del numero di registri e a seconda del parallelismo del processore.

In particolare nell'EU troviamo due blocchi logici: il «Register File» e l'«ALU». Quest'ultima (ALU sta per Arithmetic Logic Unit) è praticamente il «cuore» del microprocessore ed è il blocco logico capace di effettuare tutti i calcoli aritmetici e logici, nel nostro caso su quantità a 16 bit come a pure soli 8 bit: anticipiamo che tra le varie operazioni eseguibili vi sono

— spostamenti di dati da e verso memoria;

— tutte le operazioni algebriche, comprese la moltiplicazione e la divisione;

— le funzioni logiche quali AND, OR, XOR, NOT, comparazioni ecc. con la notevole possibilità, sulla quale ritorneremo più volte, di operare su dati ad 8 bit e a 16 bit, come visto, nonché a 32 bit (nel caso di moltiplicazione e divisione), secondo una struttura detta «a matrice», in cui ogni operazione può essere effettuata praticamente su qualsiasi tipo di operando, registro o memoria che sia.

Ma ne ripareremo in seguito, in quanto questa è una caratteristica fondamentale sulla quale si basa praticamente tutta la programmazione in Assembler dell'8086.

Ovviamente in base al risultato delle operazioni verranno settati gli opportuni flag, alcuni dei quali sono i soliti (Carry, Zero, Parity, Overflow) mentre altri sono nuovi: tra questi spicca il flag di Trace, che consente l'esecuzione in «Single Step» da programma, senza perciò necessità di circuiti esterni. Altro flag importante è il DF (Direction Flag) indispensabile, come vedremo nelle prossime puntate, nelle

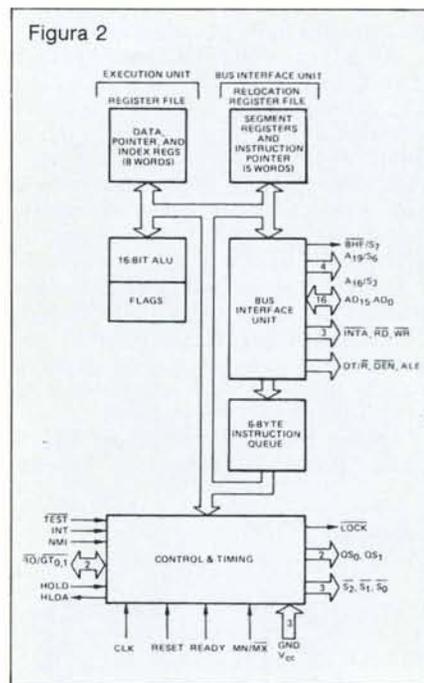
operazioni che coinvolgono blocchi di memoria (stringhe formate da byte o da word).

L'altra parte logica costituente l'EU è, come è detto, il «Register File», costituito dall'insieme dei registri attorno ai quali gravitano tutte le operazioni dell'8086.

Tali registri si suddividono, per le loro caratteristiche, in due gruppi logici e data la loro importanza inizieremo già da questa puntata a conoscerli, anche se poi li riincontreremo praticamente in ogni istruzione: cominciamo dal primo gruppo al quale appartengono i registri «general purpose».

Fanno parte di tale gruppo i registri AX, BX, CX, DX: sono registri gestibili singolarmente, se considerati a 16 bit, oppure possono essere «spezzati» ognuno in due registri ad 8 bit, chiamati rispettivamente AL, AH, BL, BH, CL, CH, DL e DH.

L'altro gruppo è formato da 4 regi-



stri, questa volta di uso particolare: si tratta di SP, il ben noto Stack Pointer, SI e DI, due registri con funzione di «indice», ed infine BP, il «Base Pointer», usato in particolari operazioni concernenti lo Stack.

Comunque, se si desidera, anche questi 4 registri possono essere usati nelle operazioni logico-aritmetiche: forse solo lo Stack Pointer (SP) deve essere usato con le dovute cautele dal momento che la sua funzione è ben nota e soprattutto non ben note conseguenze di operazioni errate su di esso.

Passiamo ora alla BIU (Bus Interface Unit) della quale fanno parte: «Relocation Register File», la «Bus Interface Unit» vera e propria e la «Instruction Queue».

Alla prima parte appartengono 5 registri a 16 bit di uso particolare.

Ricordando quanto accennato la scorsa puntata, ma rimandando alla prossima per un'analisi più dettagliata, ai quattro segmenti in cui è diviso il «mondo» dell'8086 corrispondono altrettanti registri a 16 bit, che prendono il nome di:

- CS per il Code Segment;
- DS per il Data Segment;
- SS per lo Stack Segment;
- ES per l'Extra Segment.

Strettamente legato a questi, ma non direttamente alterabile da programma, se non con le istruzioni di salto, c'è l'importantissimo IP (Instruction Pointer), corrispondente solo alla lontana al ben noto PC (Program Counter) dei micro ad 8 bit, simile in quanto è lui ad essere incrementato ogni volta che è terminata la fase di fetch di un'istruzione, ma fondamentalmente diverso dal PC in quanto deve far sempre riferimento al CS: ne riparleremo in dettaglio nella prossima puntata.

All'interno della BIU vera e propria invece troviamo tutti i circuiti logici che consentono al microprocessore la gestione del Bus multiplexato e dei segnali ad esso associati: anticipiamo i tempi dicendo che in questo blocco logico esiste un particolare «sommatore» tramite il quale vengono generati gli indirizzi fisici a 20 bit, da porre sull'Address Bus, secondo una prassi particolare dell'8086, della quale bisognerà sempre tener conto quando programmeremo in Assembler, anche se risulta praticamente trasparente (e perciò invisibile) per l'utente.

Ultima parte all'interno del BIU è, come detto, l'«Instruction Queue»

(IQ), un'ennesima caratteristica dell'8086 assente in altri microprocessori, ma presente anche nell'8088 e nei successivi 801XX, 802XX, eccetera: si tratta di una «coda» di 6 byte dove troveranno posto altrettante istruzioni del nostro programma, già fetch-ate e quindi già pronte per essere eseguite quand'è il loro momento.

Con questa particolare struttura, quando si va a leggere una certa istruzione dalla memoria, in realtà vengono letti 6 byte consecutivi: dal momento che l'8086 è a 16 bit e come tale ha la maggior parte delle istruzioni aventi op-code a 16 bit, ecco che i 6 byte corrispondono in media a 3 istruzioni.

In tal modo si ottiene un notevole risparmio di tempo nell'esecuzione di un programma in quanto si risparmia sul tempo di accesso alla memoria: viceversa si ha uno «spreco di fetch» nel caso in cui la prima istruzione della coda sia un salto ad un'altra locazione. In tal caso i byte successivi presenti nella coda non saranno mai utilizzati e perciò dovranno essere scartati.

Anche questo fatto però, dal punto di vista programmatico, è del tutto trasparente per il programmatore il quale, a meno che non utilizzi un «Analizzatore di Stati Logici», non si accorge mai della situazione: a tutti gli effetti è come se il microprocessore leggesse un byte alla volta, come siamo abituati a vedere con altri processori.

Detto quindi della BIU, rimane da accennare al blocco logico «Control & Timing»: con un'occhiata alla figura 2 si comprenderà facilmente che tale blocco è preposto alla gestione di praticamente tutti i segnali di controllo, sia di Input che di Output, attraverso

so i quali l'8086 comunica con il mondo esterno.

Non ci soffermiamo oltre, in quanto praticamente andremmo a ripetere quanto già detto nella descrizione dei segnali di controllo stessi.

Ora parliamo dell'8088

Facendo riferimento alle figure 3 e 4, parliamo ora delle differenze tra l'8088 ed il suo «fratello a 16 bit».

Per quanto riguarda i pin, innanzitutto ritroviamo il fatto che l'Address Data Bus va questa volta solo da AD0 a AD7, mentre da A8 a A15 si parla solo di Address Bus: sappiamo ormai benissimo che l'8088 ha un Data Bus ad 8 bit, per cui necessita di 2 accessi alla memoria nel caso di istruzioni con op-code a 16 bit.

Anche analizzando la figura 4 si nota, a parte una differente rappresentazione (in questo caso più dettagliata), un'unica differenza, data dall'«Instruction Queue» ora di soli 4 byte invece dei 6 dell'8086: dato che, come detto, in genere le istruzioni sono con op-code a 16 bit, ecco che, anche con una «queue» più piccola, al termine di un'istruzione troviamo già pronti due byte dell'istruzione successiva.

Evidentemente, così come succedeva per l'8086, ancora una volta questi bei discorsi cadono nel caso della presenza di istruzioni di salto, nonché per le altre istruzioni (peraltro ai limiti della generalità) che spaziano da 1 a 7-8 byte, queste ultime essendo formate da «prefissi», dall'op-code e da indirizzi «estesi»: niente paura, ne riparleremo con più calma e dettaglio nel seguito.

Figura 3

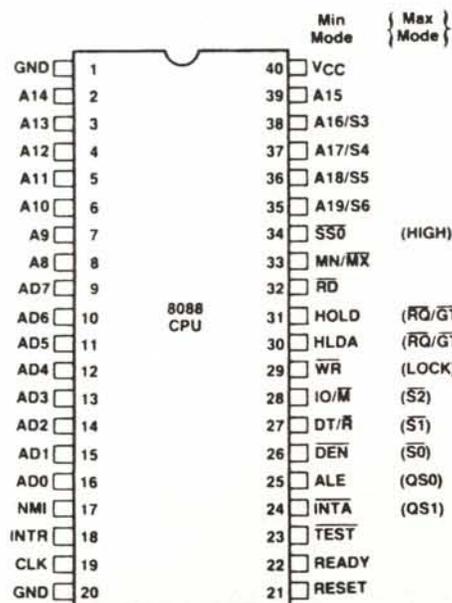
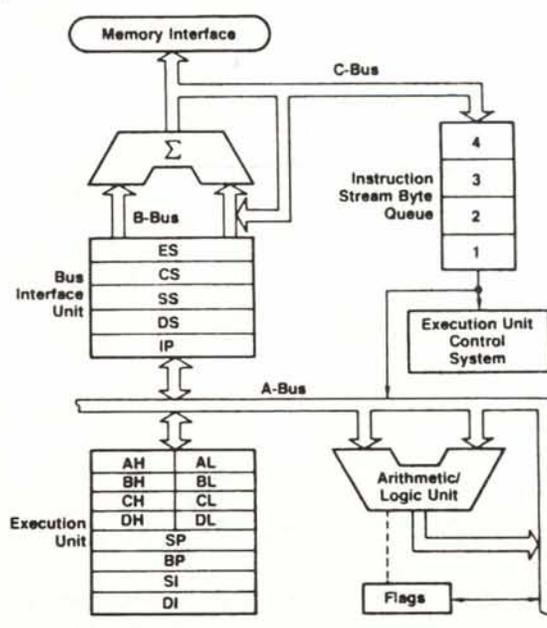
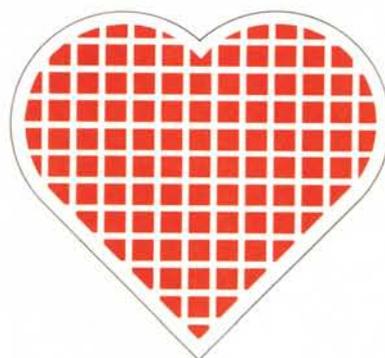


Figura 4



DA CHI E' LEADER NEL MONDO IN TECNOLOGIE DEL FUTURO - SANYO 555 IL COMPUTER A DIMENSIONE UOMO

wallcovering



CERCASI CONCESSIONARI PER ZONE LIBERE



Una tecnologia complessa che, invisibile all'utente finale, è stata sfruttata appieno per aumentare le prestazioni del sistema in termini di velocità, affidabilità e versatilità. SANYO 555, potente e compatto, è in grado di adeguarsi ad ogni necessità di elaborazione.

Funzioni di guida e menu consentono un approccio semplificato al lavoro. Lavorare è quindi facile, e la facilità d'uso costituisce uno stimolo ad imparare. Il sistema SANYO 555 è immediatamente utilizzabile da qualsiasi utente, senza la necessità di precedenti esperienze.

Il sistema SANYO 555 consente di elaborare sia i dati che i testi e sono disponibili molti pacchetti applicativi sia per l'automazione dell'ufficio sia per la gestione dell'azienda. Un'ampia scelta di linguaggi consente di ottenere la soluzione più adeguata ad ogni singolo problema.

l'Amico personal

COMPUTER
 **SANYO**

Sanyo Italiana Spa 20124 Milano - Viale V. Veneto, 22
Tel. (02) 6557762-3-4-5-6 con ric. automatica