

Anche i microprocessori hanno i loro «compatibili»

prima parte

È ormai ben noto che dall'Estremo Oriente provengono mille e mille tipi di personal computer compatibili con quelli dell'IBM: computer «made in Taiwan» oppure «in Philippines» nonché ovviamente «in Japan» sono ormai all'ordine del giorno, così come è tutt'altro che raro vedere monitor veri e propri dell'IBM con la fatidica targhetta «made in Taiwan».... Potenza delle ferree leggi di mercato! Ormai incontrare un personal computer «clone» o «cinese», come sinonimo di compatibile, non fa più tanto scalpore, e tutto sommato non ci dovremmo sorprendere tanto del fatto che si «cloni» anche un microprocessore

Già in passato era successo un qualcosa di analogo: l'8080 della Intel era stato copiato, espanso, potenziato ed abbellito dai progettisti della Zilog, i quali hanno fornito al mondo dell'informatica quel portento che era lo Z80: non si trattava di un clone, ma di un fratello maggiore, tutto sommato enormemente più potente.

Non stiamo qui a discutere sul fatto che effettivamente lo Z80 ha avuto un grande successo, diventando quasi uno standard nel campo dei microprocessori ad 8 bit.

Un fatto analogo, dicevamo, è successo per quel che riguarda il ben noto 8088, il microprocessore scelto dall'IBM come motore dei PC: dopo parecchi anni di utilizzazione di tale integrato, è apparsa sul mercato una serie di microprocessori solo in prima analisi «compatibili» con la serie Intel, ma largamente più potenti.

Ci occuperemo in questa sede del V20, che rappresenta un potenziamento dell'8088.

Differenze e miglioramenti

Prima ancora di parlare delle differenze, vediamo le analogie tra un V20 ed un 8088: innanzitutto a livello di piedinatura (perciò da un primo punto di vista «esterno») i due microprocessori sono del tutto identici, tanto che è possibile sostituire senza problemi un 8088 con un V20, con tutti i vantaggi che vedremo subito, a cominciare dal fatto che il V20 è fabbricato in tecnologia CMOS e come tale consuma molto di meno.

Dal punto di vista più interno il V20 possiede un set di istruzioni più ampio rispetto a quello dell'8088, ma che lo comprende in pieno: tutte le istruzioni dell'8088 hanno una immediata e identica rispondenza nel V20, senza alcuna eccezione, mentre una parte delle istruzioni aggiuntive del V20 lo rendono già in partenza un 80186.

Oltre a ciò alcune istruzioni sono sta-

te velocizzate all'interno del chip e ciò non può far che piacere visto che tra le istruzioni in esame ci sono la moltiplicazione e la divisione.

Perciò, rimanendo soltanto nell'ambito delle istruzioni «compatibili 8088» (ma si può dire «uguali a quelle dell'8088»), si ha un netto miglioramento nelle prestazioni del microprocessore, senza alcuna controindicazione di sorta: se invece in un programma si utilizzano le istruzioni aggiuntive del V20 (sia quelle analoghe a quelle dell'80186, sia quelle «nuove»), allora è ovvio che si perderà la compatibilità software su personal computer che girano normalmente sotto MS-DOS, dotati perciò di 8088.

La struttura interna del V20

Come si può vedere dalla figura 1, tratta dal data book della NEC, si può notare una ovvia rassomiglianza con la struttura dell'8088, riportata nel lontano numero 47 di MC, nell'ambito della rubrica «Assembler 8086/8088», ma ciò che più spicca è la differente nomenclatura delle varie parti. Ciò è una conseguenza ovvia del fatto che dovendo «copiare» un prodotto già affermato come l'8088 e sul quale sicuramente ci sono tutti i copyright possibili ed immaginabili, i progettisti della NEC si sono dovuti «inventare» nuovi nomi per le varie parti (a partire dai registri per arrivare ai blocchi funzionali), potendosi poi sbizzarrire per quel che riguarda le nuove caratteristiche. Vedremo in seguito che le differenze (sottolineiamo il fatto che sono solo differenze «formali», esteriori) potranno essere completamente annullate semplicemente «rileggendo» i nomi di registri ed istruzioni nel modo più consueto, confidando nella completa identità dei codici operativi, che è la cosa che più conta.

Iniziamo dunque dai registri del microprocessore: ferma restando la possibilità di suddividere i registri AX, BX, CX

e DX (che qui diventano AW, BW, CW e DW), nelle coppie AL, AH, BL, BH, ecc., abbiamo la seguente tabellina di corrispondenza tra i nomi del V20 e quelli dell'8088:

Registro V20	Registro 8088
AW	AX
BW	BX
CW	CX
DW	DX
IX	SI
IY	DI
BP	BP
SP	SP
PC	IP
PS	CS
SS	SS
DS0	DS
DS1	ES

Da questa tabella di corrispondenza vediamo dunque che i progettisti della NEC si sono ispirati (non a caso, ma di ciò parleremo in seguito) allo Z80 per i nomi dei registri IX ed IY e per il PC («Program Counter») al posto dell'«Instruction Pointer», nonché all'8085 per il registro dei flag, che è stato chiamato PSW («Program Status Word»).

A parte dunque lo Stack Pointer, lo Stack Segment ed il Base Pointer (per i quali non è semplice trovare sinonimi), ecco che compaiono il «PS» («Program Segment») ed i due «DS0» e «DS1», rispettivamente «Data Segment 0» e «Data Segment 1».

Mentre per tutti registri citati non c'è nulla di differente rispetto a quelli dell'8088 (a parte la PSW, della quale parleremo in seguito), notiamo la presenza di qualche registro in più tanto nell'EXU («EXecution Unit») che nella BCU («Bus Control Unit»).

Nella prima Unit sono presenti i seguenti registri innovativi, gestiti dalla CPU, ma non raggiungibili dall'esterno via software:

— **LC**, «Loop Counter», utilizzato nelle istruzioni di stringa, di I/O multiplo e nelle istruzioni di shift e/o rotate, fatto

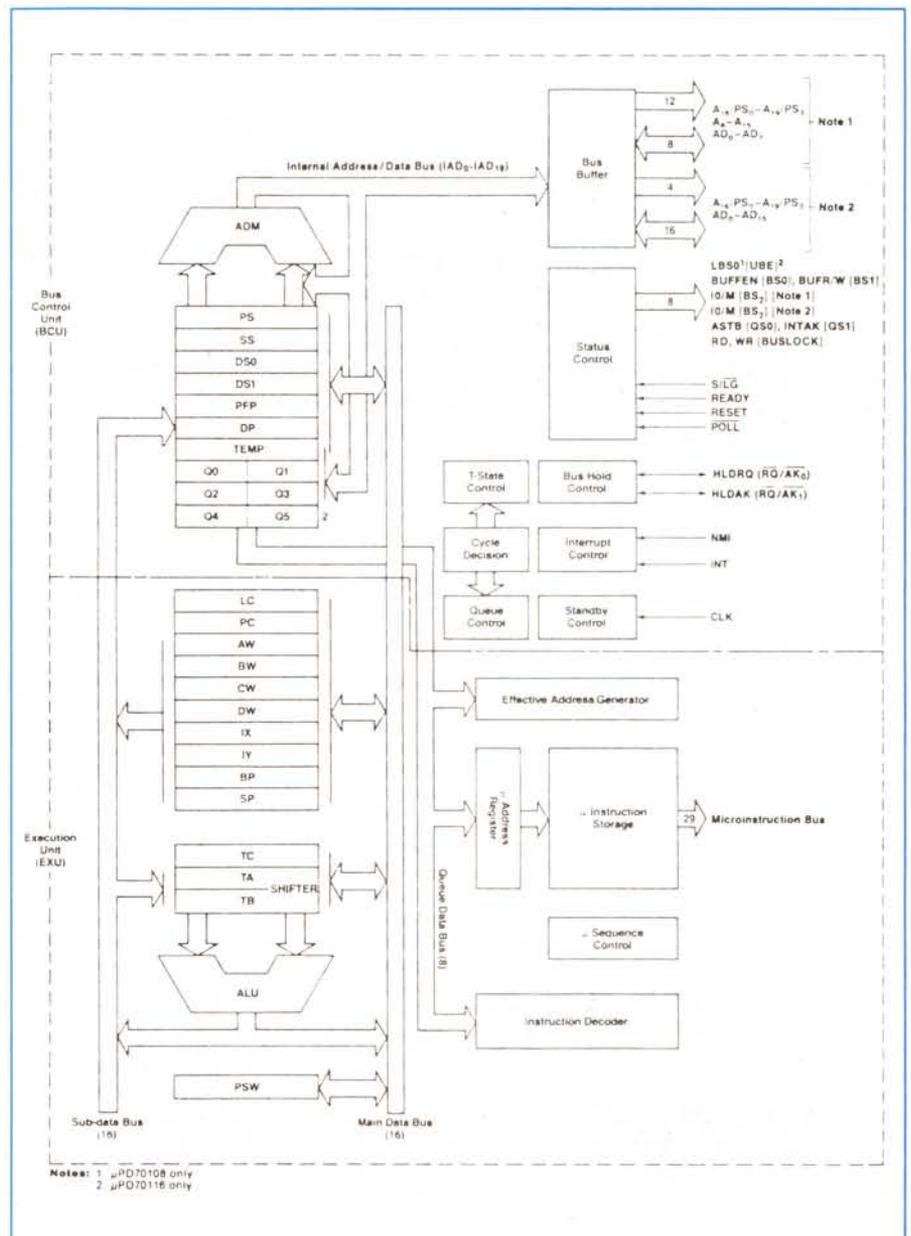


Figura 1 - Schema a blocchi del microprocessore NEC V20, indicante le sue parti funzionali ed i registri interni.

registro IX (=SI), incrementandone o decrementandone il contenuto di una o due unità (byte oppure word) a seconda se il flag di direzione sia resettato oppure settato.

Le istruzioni di moltiplicazione estesa sono del tipo:

MUL prodotto, fattore1, fattore2

e permettono di effettuare la moltiplicazione tra il «fattore1» (un registro a 16 bit oppure il contenuto a 16 bit di una locazione di memoria) ed un operando immediato («fattore2») di tipo byte o di tipo word, ponendo il risultato a 16 bit nel registro «prodotto», che può anche essere lo stesso di «fattore1» nel qual caso l'istruzione diventa del tipo

MUL prodotto, fattore2

Sono dunque possibili le 6 combinazioni:

— «MUL registroa,registrob,byte», ad esempio

MUL BW,CW,5

che effettua $CW * 5$ e pone il risultato in BW

— «MUL registro,byte», ad esempio

MUL AW,23

che moltiplica il contenuto di AW per 23

— «MUL registroa,registrob,word», ad esempio

MUL IX,IY,200H

che moltiplica per 200H il contenuto di IY per porre il risultato in IY

— «MUL registro,word», ad esempio

MUL IX,333H

che moltiplica il contenuto di IX per 333H

— «MUL registro,locazione,byte», ad esempio

MUL CW,ALFA16,12

che pone in CW il risultato della moltiplicazione del contenuto della word posta all'indirizzo ALFA16 per 12

— «MUL registro,locazione,word», ad esempio

MUL DW,[IY],2222

che pone in DW il prodotto tra la costante 2222 ed il contenuto della locazione di memoria il cui indirizzo è posto in IY.

Come si vede non si può fare in modo che il risultato rimanga in una cella di memoria, ma ciò non è un problema, dato che è semplice passare da un registro ad una locazione di memoria.

Altro tipo di istruzioni sono quelle di shift e di rotate (su registri o locazioni di memoria ad 8 o 16 bit) in cui il numero di volte che deve essere effettuata l'operazione è data da un operando immediato ad 8 bit: dato che si tratta di 7 istruzioni

base, abbiamo in totale altre 28 possibili istruzioni.

Le istruzioni di shift/rotate aggiuntive sono dunque

SHL operando, immediato

SHR operando, immediato

SHRA operando, immediato

ROL operando, immediato

ROR operando, immediato

ROLC operando, immediato

RORC operando, immediato

che rispettivamente effettuano uno shift verso sinistra, uno shift verso destra, uno shift verso destra ma con mantenimento del segno, una rotazione verso sinistra ed una verso destra nelle quali il carry ricopia il bit shiftato ed infine una rotazione verso sinistra ed una verso destra in cui viceversa il carry partecipa attivamente alla rotazione.

Come si vede anche queste sono istruzioni ben note e l'unica differenza riguarda il fatto che il numero di shift o di rotazioni è stabilito da un valore immediato e non più nel valore fisso «1» oppure nel valore contenuto nel registro CL, che così può attivamente essere utilizzato per shiftare il suo contenuto.

Altre istruzioni comuni all'80186 e successivi riguardano tre tipi di PUSH ed una POP: in particolare sono possibili le istruzioni duali

PUSH R

POP R

che servono rispettivamente a salvare nello stack e ripristinare da esso tutti i registri (le istruzioni omonime sono le PUSHA e POPA).

In particolare ricordiamo che la «PUSH R» effettua il salvataggio nello stack dei registri AW, CW, DW, BW, dell'SP iniziale, di BP, IX ed IY, mentre la «POP R» ripristina successivamente il contenuto dei registri IY, IX, BP, BW, DW, CW, AW ed infine di SP: come si vede il valore di SP viene calcolato per ultimo e corrisponde correttamente al valore che avrebbe se si fossero effettuate 8 istruzioni di POP. A nulla servirebbe ripristinare il valore di SP in corrispondenza del valore viceversa salvato al quarto posto in quanto con un facile ragionamento è possibile vedere che non rappresenterebbe il valore corretto per l'SP da ripristinare. Le altre due istruzioni di PUSH riguardano la possibilità di «push-are» nello stack un valore immediato ad 8 oppure a 16 bit, senza perciò richiedere l'intervento di un registro oppure di una locazione di memoria.

Altro ragionamento facile facile ci porta a scoprire la perfetta inutilità nonché impossibilità di una coppia di istruzioni duali alle precedenti, che ripristinerebbero il valore di una costante...

Le due istruzioni di interfacciamento verso linguaggi ad alto livello sono:

PREPARE immediato16,immediato8

e

DISPOSE

La prima serve a predisporre una zona dello stack dove linguaggi ad alto livello (tipo Pascal) possono allocare variabili locali o puntatori, da indirizzare per mezzo del registro BP: il valore «immediato16» specifica in questo caso l'ampiezza della zona all'interno dello stack da riservare, mentre «immediato8» indica il numero di volte che l'operazione precedente deve essere eseguita in base al livello lessicale della procedura interessata.

Tale istruzione corrisponde esattamente alla ENTER del 186 e seguenti.

Invece l'istruzione DISPOSE (sinonimo della LEAVE) serve a ripristinare lo stato dello stack al termine della procedura introdotta per mezzo di una PREPARE ed in particolare serve a liberare lo stack delle variabili locali, non più utilizzabili e che viceversa occuperebbero inutilmente spazio nello stack.

L'ultima istruzione che analizziamo in questa puntata è la

CHKIND registro16,memoria32

che serve a testare se il valore contenuto nel «registro16» (e che rappresenta un indice di un vettore) è posto all'interno dei valori minimo e massimo posti in memoria come double-word posta all'indirizzo indicato nell'istruzione stessa: se il valore dell'indice è al di fuori dei limiti, allora verrà generato un «BRK 5» che nel gergo del V20 corrisponde in tutto e per tutto all'«INT 5».

Tale istruzione è perfettamente identica alla «BOUND» di pertinenza dell'80186 e superiori. Con questo abbiamo terminato questa prima parte di analisi delle istruzioni del V20: nella prossima puntata parleremo delle ulteriori istruzioni presenti nel V20, ma che non trovano riscontro in analoghe del 186 e 286, ma che una certa rassomiglianza (non come codice operativo, purtroppo) con alcune di quelle del 386, delle quali parleremo nei prossimi numeri di MC.

Inoltre daremo un'occhiata ai miglioramenti ottenibili nei tempi di esecuzione di istruzioni particolari, quali le più volte citate moltiplicazione e divisione.

L'intelligenza (definizione)



F - 1000



F - 1200



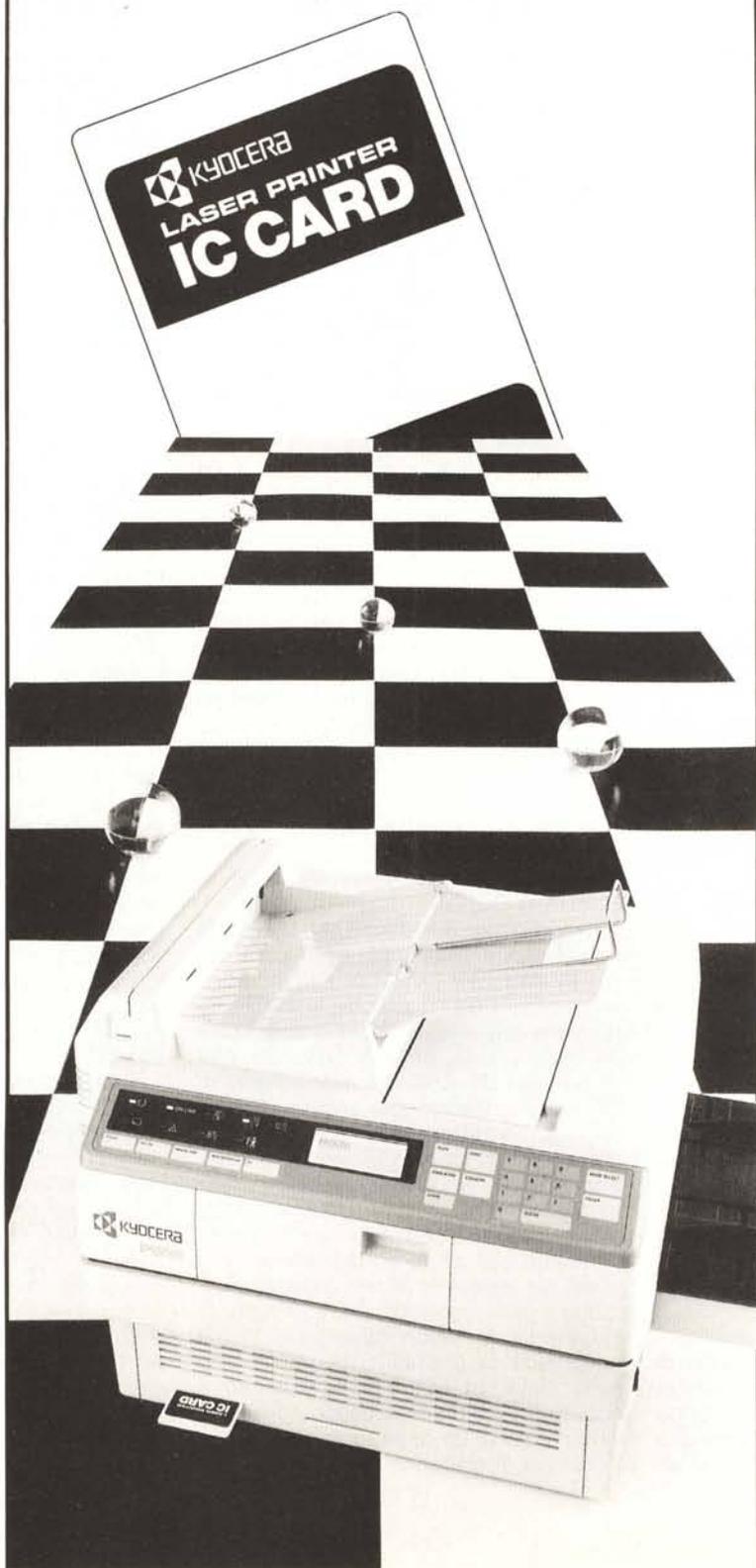
F - 2200



F - 3000



P - 2000



L'esperienza, la tecnologia, le prestazioni, la qualità: gli specialisti.

Il numero dei font residenti, le emulazioni standard, la velocità di stampa, l'espandibilità di memoria, le IC CARDS, l'interfaciabilità con mini e main frame: la versatilità.

Il linguaggio stampa "Prescribe", il linguaggio stampa "Postscript": l'intelligenza.

L'elaborazione testi, l'editoria da tavolo, la grafica professionale: l'altissima definizione.

L'affidabilità CBM, l'intelligenza Kyocera.

Soltanto gli specialisti potevano offrire intelligenza, definizione, in un panorama di opportunità: Kyocera le stampanti laser.

KYOCERA
LE STAMPANTI LASER

CBM SpA
DIVISIONE INFORMATICA

Distributore per l'Italia di KYOCERA stampanti laser.
Via Paolo Di Dono, 3/A - 00143 Roma - Tel. 50393.1 (R.A.)
Telex 611174 CBM SPA - Fax 50293205