di Pierluigi Panunzi

L'evoluzione della specie

Al termine di una più che doverosa serie di articoli (in un certo senso propedeutici) riguardanti i microprocessori della Intel, l'8086 prima, l'80286 dopo, ecco che dunque siamo più che pronti ad affrontare il grande passo che ci porterà a conoscere più da vicino l'80386, che si sta sempre più affermando come il «cuore», il «motore» dei PC più moderni, senza con questo voler assolutamente minimizzare l'operato dei microprocessori precedenti, semmai un tantino sottoutilizzati (il 286 in special modo)

È un «32 bit»!

Qualche anno fa andava di moda alle TV una reclame che pubblicizzava un prodotto, diciamo così, «medicinale», per acquistare il quale bastava citarne il nome; allora era nato lo slogan «Basta la parola!».

Allo stesso modo, se si vuole sintetizzare la potenza intrinseca di un 386, basta la parola (o meglio due...): 32 bit. Appunto il titolo del paragrafo.

Ricordano i lettori i tempi in cui esistevano i microprocessori ad 8 bit?! Una decina d'anni sembra invece un'eternità. E i microprocessori a 16 bit? Quanta potenza e che meraviglia di computer sono nati...

Ora che stanno sempre più prendendo piede i personal computer dotati di microprocessore a 32 bit cosa dovremmo dire?

Ci possiamo in pratica solo lamentare che ancora non esistono seri e veri sistemi operativi multitasking, che viceversa sfrutterebbero l'incredibile potenzialità di tali mostri-millepiedi.

Comunque già possiamo abbondantemente accontentarci di vedere i nostri programmi correre come non avevano mai fatto: lo stesso WordStar Professional 4 con cui il redattore sta scrivendo questo articolo è diventato una «scheggia» (come si dice a Roma)! Grazie: in un PC dotato di 386, di clock a 25 MHz, e di hard disk di tipo «fast» non ci si accorge nemmeno che il WS ha una schermata iniziale!

Per non parlare poi dei giochi: è diventato impossibile pilotare la navetta spaziale tra i piloni elettrificati del buon «Buck Rogers»...

Ma non divaghiamo, stavamo parlando di cose serie.

Dicevamo dunque della velocità: il 386 ha in pratica la stessa velocità del 286 in termini di tempi di esecuzione delle istruzioni (moltiplicazioni tra word in 24 cicli di clock massimi), mentre ovviamente permette (essendo a 32 bit) di lavorare su quantità appunto a 32 bit (doubleword). Da notare fin d'ora, ma ritorneremo sull'argomento nel seguito, che una moltiplicazione tra due quantità a 32 bit (un registro ed una locazione di memoria), per dare un risultato a 64 bit, impiega al massimo 41 cicli di clock!!! Considerato poi che il microprocessore può essere clock-ato alla bellezza di 25 MHz (e nelle ultime versioni addirittura a 33 MHz), ciò dà un'idea della sua potenza in soli termini «temporali».

A proposito, visto che ci siamo, (ma poi passiamo senz'altro al 386 vero e proprio) il computer di cui sopra dà come indice di «relative speed (orig PC = 100%)» del PCTOOLS la bellezza del 1035%, mentre il programma «System Info» (SI) di Norton fornisce un «Computing Index (CI) relative to IBM/XT» pari a 22.0: non male, vero?!

Tornando al titolo, i «32 bit» sono veramente la sintesi della potenza «bruta» del 386: una volta conosciutolo meglio ci si accorgerà che tutto sommato la potenza del 386 va ben oltre i 32 bit più e più volte citati.

Un'occhiata alle sue caratteristiche

Dicevamo dunque che il 386 possiede registri a 32 bit, ma può lavorare indifferentemente su quantità a 16 e ad 8 bit, tanto registri che locazioni di memoria: per quanto riguarda quest'ultima, c'è subito da dire che la quantità di memoria indirizzabile è veramente enorme.

Dal momento che il bus degli indirizzi è a 32 bit, ecco che, in «Real Mode» il 386 può indirizzare 2^32 locazioni, cioè 4 Gbyte (Gigabyte!), valore che si ritrova anche come massima ampiezza di un singolo segmento (!), mentre in modo virtuale può indirizzare qualcosa come 64 Tbyte, dove la «T» sta per «Tera», che in genere rappresenta «1000 Giga», ma che in termini informatici vale in realtà 1024 Giga.

I 64 Tbyte (2^46 e cioè un valore maggiore di 70 e 12) in definitiva sono 64*1024*1024 Mbyte di memoria: interessante è il confronto con quel Megabyte di memoria, vanto di parecchi personal, che viceversa l'MS-DOS si ostina a non vedere nemmeno per intero.

Tanta memoria virtuale, così come quella fisica, è interamente gestibile «on chip», grazie ad una «Memory Management Unit» (MMU) integrata.

Per quanto riguarda la programmazione, alla quale dedicheremo ampio spazio nel seguito, accenniamo già al fatto

che il 386 è già completamente compatibile («verso il basso») con il 286, per quel che riguarda il modo protetto ed i livelli di privilegio, per non parlare della compatibilità del codice oggetto che è ovviamente totale sia verso il 286 che verso il capostipite 8086: ci sono in realtà delle minime differenze «di comportamento» nell'esecuzione di talune istruzioni, delle quali parleremo a tempo debito, così come già accadeva con il 286.

A tutto questo l'80386 aggiunge ovvie estensioni, sia per quel che riguarda le protezioni (si ricordano i lettori dei campi dei vari descriptor che bisognava lasciare vuoti per compatibilità verso il 386? In questa sede ne conosceremo il significato), sia per quel che riguarda il set di istruzioni che è ulteriormente arricchito.

Su tutte le nuove istruzioni spicca finalmente una «Test and Set», della quale si ha necessità per la gestione corretta di «semafori» nella gestione di risorse condivise in sistemi multi-task, ma forse è meglio non correre troppo...

Altra novità del 386 è la possibilità notevolissima di operare in «Virtual 8086 Mode» creando apposite «isole» logiche in cui funziona in tutto e per tutto come un 8086, il tutto supervisionato da un sistema operativo che gira viceversa in modo protetto, come un 386 puro.

Attenzione: un programma multi-tasking come il «Windows» non realizza quanto detto, dal momento che vede tutte le applicazioni (i programmi che girano ognuno in una «window») come task da attivare in sequenza, proprio come già era capace di fare il 286 (ed infatti «Windows» gira sotto 286).

Invece sotto «Virtual 8086 Mode», come avremo modo di vedere in seguito nel dettaglio, si crea un vero e proprio ambiente 8086 in cui si può far girare addirittura un sistema operativo (ad esempio proprio l'MS-DOS), senza che questo possa interferire con il sistema operativo «supervisore»: in tal modo non si parla più di «lanciare più applicazioni» (task) in un unico ambiente, ma si potrà parlare di «lancio di sistemi operativi dotati di applicazioni», senza dover necessariamente riscrivere «in 386» i sistemi operativi stessi.

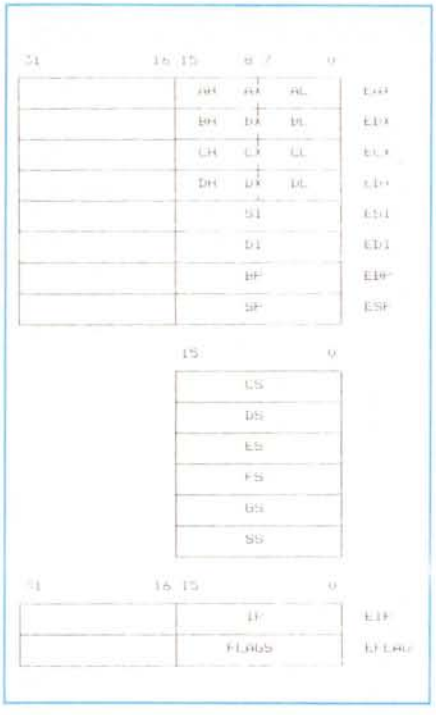

Figura 1 - Questi sono i registri interni dell'80386: quasi tutti sono un'estensione a 32 bit dei registri ben noti. Nuovi sono i due registri di segmento FS e GS, a tutti gli effetti simili ad ES.

Suggestivo è a tal proposito l'uso del «Turbo Debugger» (della Borland, anche se non c'era bisogno di ricordarlo...) nella versione 386 (il programma TD386) il quale si va ad installare nel secondo Mbyte di memoria e mostra il primo Mbyte (e cioè l'usuale ambiente DOS, seppur ridotto ai canonici 640 Kbyte) come «Virtual 8086 Mode»: lanciando da tale debugger il file "COMMAND.COM" si entra appunto in un DOS posto in un'«isola», all'interno del quale si possono ovviamente eseguire tutti i programmi che vogliamo...

Comunque ne riparleremo: gli orizzonti che si aprono sono veramente vasti...

Concludiamo l'elenco brutale e non certo completo delle innovazioni introdotte con il 386 dicendo che il «piccolo mostro» è dotato di una serie di registri nuovi che consentono il debugging di programmi via hardware (meraviglioso! Il bello è che il Turbo Debugger di cui sopra già gestisce queste novità), mentre «last but not least» ha la possibilità di gestire, oltre al costosissimo 80387 (coprocessore matematico), anche il più economico (!) ma più lento 80287, per la gioia degli utenti che non vogliono ulteriormente dissanguarsi...

I registri del 386

Nella figura 1 abbiamo riportato tutti i registri utilizzabili da un programma applicativo: mancano alcuni registri di cui parleremo nel seguito e che non sono di immediata utilità ed utilizzazione da parte dell'utente medio.

In pratica si tratta di tre gruppi di registri:

— 8 registri «general purpose»;
— 6 registri di segmento;
— 2 registri speciali.

I primi 8 registri sono in particolare un'estensione verso i 32 bit dei registri già ben noti dall'8086 e dall'80286: al loro nome è stata aggiunta una lettera «E» che sta per «Extended» ed hanno tutti la possibilità di essere visti tanto a 32 bit che a 16 bit.

In tutti i casi dunque la parte meno significativa di un registro «Extended» (ad esempio ESI) corrisponde al registro «vecchio» (nel caso, SI), mentre la parte più significativa è del tutto nuova: in particolare quando il 386 lavora in modo reale (ad esempio in un qualunque programma MS-DOS) allora utilizza di ogni registro esteso solo la parte meno significativa, anche perché è ovvio che i programmi scritti per 8086 o 80286 non potevano prevedere l'uso di registri «Extended», altrimenti non avrebbero funzionato.

Inoltre, come è ovvio aspettarsi, i primi quattro registri del gruppo sono ulteriormente scindibili in coppie di registri ad 8 bit, così come siamo abituati a fare con i precedenti microprocessori.

In definitiva il 386 vede EAX come un

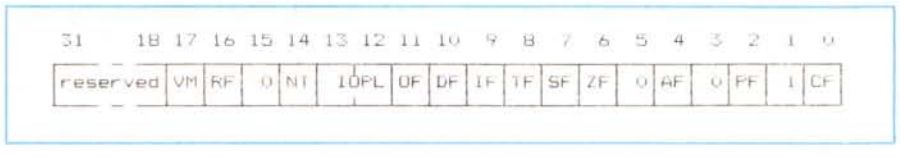

Figura 2 - Il registro di flag «Extended FLAGS» (EFLAGS) contiene, oltre ai soliti bit già presenti nel 286, due nuovi bit chiamati «VM» e «RF», legati rispettivamente al «Virtual 8086 Mode» ed al debugging tramite appositi registri interni del 386.

registro a 32 bit, ma contemporaneamente ne può vedere i 16 bit inferiori come l'usuale AX, che infine può ancora una volta vedere (sempre contemporaneamente e non in alternativa) come AH ed AL: in un programma per 386 si può dunque caricare tutto EAX e successivamente compiere operazioni con AH ottenendo un nuovo valore per EAX.

Per quanto riguarda i compiti dei registri, vediamo che è stato mantenuto quanto si aveva nell'86 e nel 286, ma con un'elasticità quasi completa: da un lato cioè sono rimasti i ruoli «canonici» per i vari registri:

— EAX, AX, AH ed AL hanno ancora il ruolo di «accumulatore»;

— EBX e BX hanno il ruolo di registro «base» (in special modo nell'istruzione XLAT);

— ECX e CX fungono ancora da registro «contatore» per tutte le istruzioni tipo LOOP nonché per la ripetizione di istruzioni di stringa (prefisso REP), mentre CL è ancora usato per operazioni di shift multiple in cui non è coinvolto un valore immediato;

— DX ha mantenuto la funzione di selettore della porta di I/O nelle funzioni di Input ed Output: dato che le porte indirizzabili sono, come vedremo, rimaste 65536, ecco che tale funzione non è stata estesa ad EDX tanto che le istruzioni di IN ed OUT non possono avere come operando EDX;

— ESI, SI, EDI e DI hanno ancora il ruolo rispettivamente di «sorgente» e «destinazione» nelle operazioni di stringa, le quali ora, oltre che a dati ad 8 e 16 bit, possono far riferimento a dati a 32 bit, cambiando semplicemente il codice mnemonico, ma non l'«opcode»;

— ESP ed SP rimangono ovviamente gli «Stack Pointer» in segmenti a 32 e 16 bit rispettivamente, senza altre funzioni aggiuntive, che tutto sommato sarebbero solo di disturbo alle normali attività dello stack...

— EBP e BP infine sono ancora una volta i «Base Pointer» per operazioni di accesso a locazioni appartenenti allo stack.

L'elasticità completa è viceversa dovuta al fatto che, oltre a quanto detto, tutti i registri possono essere usati come accumulatori (per carità lasciamo da parte ESP!!), cosa che peraltro già accadeva con l'8086, ma in più si ha la possibilità di utilizzare tutti gli otto registri nell'indirizzamento indiretto di locazioni di memoria, come vedremo nel seguito.

I registri di segmento, l'IP e i flag

È il secondo gruppo di registri della figura 1: oltre ai quattro registri ben noti, che sono CS, DS, ES ed SS, rispettivamente «Code Segment», «Data Segment», «Extra Segment» e «Stack Segment», appaiono due nuovi registri a 16 bit, chiamati «FS» e «GS», in pratica due utili aggiunte al registro ES, adatti in quei programmi che fanno accesso a dati appartenenti a più segmenti di dati differenti.

Il loro nome è semplicemente una prosecuzione «alfabetica» della serie «CS-DS-ES» ed in generale quando si useranno nei programmi compariranno sotto forma di «segment override»; a differenza di ES invece non hanno alcuna funzione dedicata. Un esempio di loro applicazione può essere in un programma che preveda la gestione della memoria video in routine grafiche: mentre DS supervisionerà i dati e le variabili interne del programma ed ES comprenderà come segmento la memoria video, ad FS potremo utilmente associare il segmento posto nella «parte bassa» della memoria, laddove si trovano le variabili di sistema, oltreché gli interrupt vector, per non essere co-

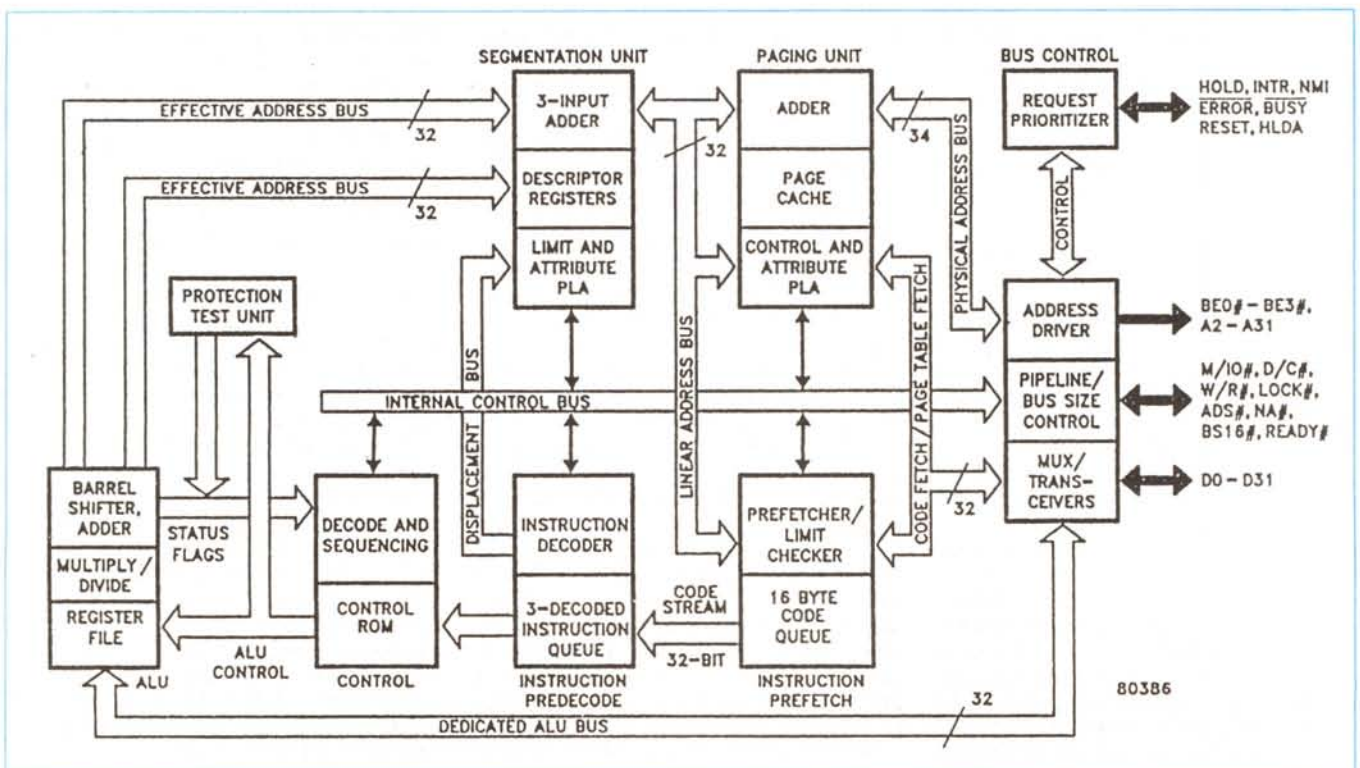

Figura 3 - L'architettura interna dell'80386 è particolarmente complessa: si tratta di una struttura a «pipeline» a 32 bit effettivi che tali rimangono nell'interfacciamento esterno: solo nella versione 80386SX, di recente immessa sul mercato, i 32 bit diventano 16 nell'interfacciamento esterno, per favorire l'utilizzo di circuiterie finora utilizzate nell'ambito dei 16 bit e soprattutto abbassando il costo effettivo del chip.

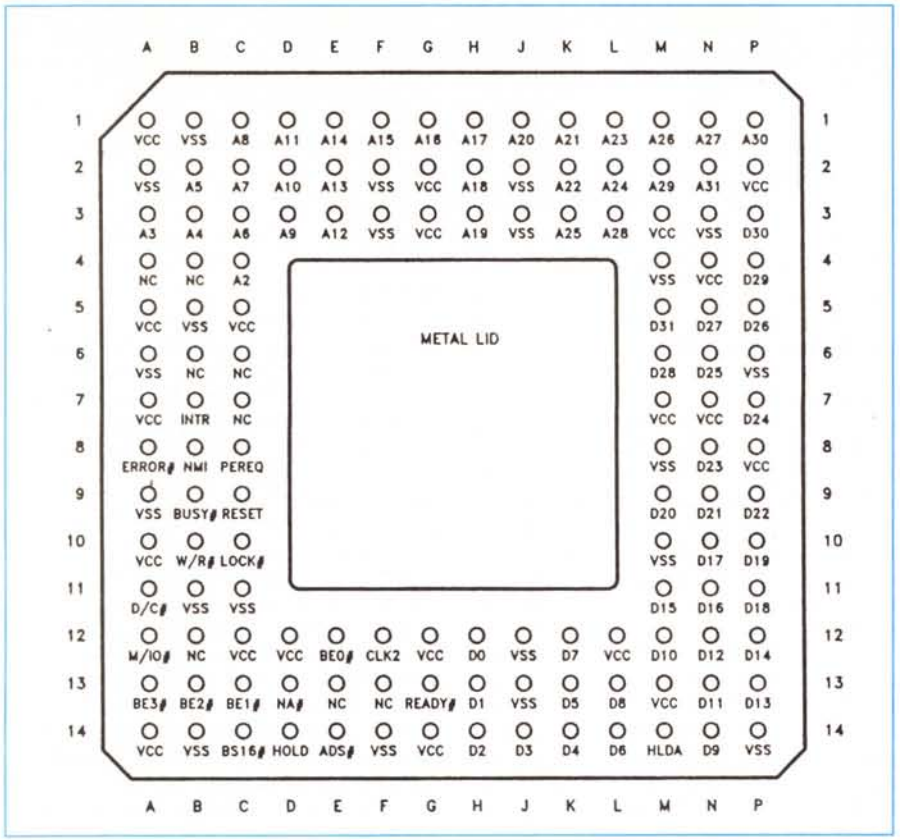

Figura 4 - Anche questa figura è tratta, come la precedente, dalla letteratura Intel: si tratta della piedinatura del 386, che prevede l'uso di ben 132 pin di connessione dei quali 41 solo per l'alimentazione e la massa.

stretti a salvare e cambiare il valore di DS ogni volta che dobbiamo ad esempio andare a leggere il «modo video corrente».

Come già accadeva nel 286, nel 386 in «Real Mode» tali registri contengono proprio il valore del segmento di memoria a partire dal quale è posto il «segmento» (pare sempre un gioco di parole...).

Invece ci aspettiamo già (ed infatti così è) che in «Protected Mode» diventino dei «selector» a rispettive tabelle di descrizione dei segmenti: nel seguito avremo modo di analizzare i dettagli, per vedere soprattutto quali sono le innovazioni apportate nel 386.

Il terzo gruppo di figura 1 è rappresentato dall'«Instruction Pointer» e dal registro dei flag.

L'IP è anche lui stato esteso a 32 bit (EIP) ed è indispensabile nel caso di utilizzazione del 386 «alla piena potenza», allorché si deve poter indirizzare un'istruzione posta fino al limite di 4 Gbyte: lavorando a 16 bit (vedremo nel seguito cosa significa «lavorare a 32 bit» contrapposto a «lavorare a 16 bit») invece è ancora una volta la parte meno significativa a rappresentare il ben noto IP.

Infine per quanto riguarda il registro dei flag, c'è da dire che è stato anche lui esteso a 32 bit, però in realtà sono stati aggiunti solo due flag, lasciando i rimanenti ad implementazioni future (80486, 80586, ecc.). In figura 2 vediamo dunque che nella parte più significativa di «EFLAGS» sono stati aggiunti i flag «VM» («Virtual Mode») e «RF» («Resume Flag») sulla cui utilizzazione torneremo a tempo debito: per ora diciamo che si tratta, il primo, del flag che serve a switch-are dal modo protetto al «Virtual 8086 Mode», del quale abbiamo già dato un cenno alquanto vago, mentre il secondo serve abbinato all'uso dei registri di breakpoint utili per il debugging.

Sembra un ritornello: anche di questo parleremo più in là. D'altro canto, come ci si può aspettare, gli argomenti e le novità sono veramente molteplici per cui è logico che si segua un certo ordine di presentazione altrimenti si rischia di confondere ancor più le idee: specie con il 386, alcuni argomenti saranno alquanto complicati, ma il fatto di aver già affrontato certe tematiche per il 286 ci farà affrontare con spirito migliore taluni punti complessi.

A tale scopo (lo diciamo ora, ma dovrebbe essere chiaro già in partenza) già fin da questa puntata consigliamo i lettori interessati a rileggersi gli articoli sul 286 in quanto parecchie volte faremo riferimento ad essi per non dover ripetere concetti che, tutto sommato, abbiamo già analizzato anche in più puntate.

Però, per non «perdere per strada» i lettori, tenderemo comunque a ripetere concetti particolarmente importanti: a tal proposito ad esempio ci accorgiamo del fatto che abbiamo dato quasi per scontato che il 386 può lavorare in modo «Reale» ed in modo «Protetto», dal momento che questa non è altro che una diretta conseguenza del fatto che tale microprocessore prende le mosse dal 286 e ci sarebbe sembrato strano dover dire all'inizio che «il 386 può lavorare in due modi...».

Il tutto ovviamente accade perché il software, dell'8086 prima e dell'80286 dopo, possa girare senza problemi in un computer dotato di 80386.

Chiudiamo questa puntata dando un'occhiata alla figura 3 (tratta dalla documentazione Intel), nella quale è riportata la struttura interna, per blocchi funzionali, del 386: la prima tentazione è di andare ad analizzare tale struttura a «pipeline» fortemente ottimizzata ed elaborata per seguire il percorso che compie un'istruzione dalla sua lettura alla sua esecuzione, però ci rendiamo conto che il discorso si farebbe ancora più complesso di quello che già risulta, considerando il 386 come una «scatola nera» (o «Black Box», BB, come si dice in gergo). Semmai, laddove possa interessare, potremo dedicare qualche spazio anche ad un'analisi più «microscopica» del microprocessore: a proposito, dimenticavamo di dire che il 386 si presenta come un chip quadrato dotato di «appena» 132 piedini (diconsi centotrentadue).

Tra questi, come si vede dalla figura 4, spiccano ben 20 pin da connettere all'alimentazione e 21 alla massa, oltre ai più di 32 pin per l'Address Bus ed i 32 per il Data Bus: in particolare la proliferazione di pin di alimentazione e di massa è spiegata dall'alta complessità di integrazione del chip e soprattutto dalle alte frequenze in gioco che avvicinano sempre più i computer a dei trasmettitori in «CB» («Citizen Band») da radioamatori: senza opportune connessioni di massa interne il processore irradierebbe ben oltre il consentito e viceversa verrebbe disturbato da campi elettrici vicini. Ma tanto anche con un «vecchio PC» a 4.77 MHz acceso nelle vicinanze è quasi impossibile ascoltare una radio ad onde medie.

Con questo abbiamo terminato: nella prossima puntata analizzeremo in dettaglio i nuovi modi di indirizzamento e parleremo di nuove istruzioni.