

11. Struttura e Funzione del Processore

Compiti della CPU

La CPU svolge cinque compiti fondamentali: prelevare istruzioni dalla memoria, interpretarle, prelevare i dati necessari, elaborarli e scrivere (memorizzare) i risultati.

Struttura della CPU

La CPU è connessa al sistema tramite il **bus di sistema**, composto da tre componenti: Control Bus, Data Bus e Address Bus. Internamente, la CPU contiene l'**ALU** (Arithmetic and Logic Unit), composta da Status Flags, Shifter, Complementer e logica aritmetico-booleana. I **Registri** sono collegati tramite un bus interno alla CPU, e la **Control Unit** gestisce i percorsi di controllo tra le componenti.

Registri

I registri costituiscono lo "spazio di lavoro" della CPU per memorizzare i dati e rappresentano il livello più alto della gerarchia della memoria. Il numero e le funzioni variano in base all'architettura della CPU.

Categorie di registri

Registri utente: utilizzati dal programmatore (umano o compilatore) per memorizzare dati da elaborare. Il programmatore può essere un umano che programma in assembler o un compilatore che genera codice assembler da linguaggi ad alto livello.

Registri di controllo e di stato: usati dall'unità di controllo per monitorare le operazioni della CPU e dal Sistema Operativo per controllare l'esecuzione dei programmi.

Registri utente

I registri visibili all'utente si suddividono in: registri ad uso generale, registri per dati, registri per indirizzi e registri per codici di condizione.

Registri ad uso generale: possono essere veramente generici o dedicati a funzioni particolari. Possono contenere dati (es. accumulatore) o indirizzi (es. indirizzo base di un segmento di memoria). I registri generici aumentano la flessibilità ma richiedono istruzioni più lunghe e complesse, poiché servono bit aggiuntivi per specificare quale registro utilizzare (es. 4 bit per 16 registri, 5 bit per 32 registri). I registri specializzati producono istruzioni più piccole e veloci ma sono meno flessibili.

Numero ottimale di registri: tipicamente tra 8 e 32. Con meno di 8 registri aumentano gli accessi alla memoria principale, poiché i dati intermedi devono essere ricaricati dalla memoria quando i registri disponibili non sono sufficienti. Oltre 32 registri non si ottengono ulteriori riduzioni degli accessi in memoria e si occupa molto spazio nella CPU. Le architetture RISC tipicamente hanno più di 32 registri.

Dimensione dei registri: devono essere abbastanza grandi da contenere un indirizzo di memoria principale o una "full word". È spesso possibile combinare due registri per ottenerne uno di dimensione doppia (es. `double int`, `long int` in C).

Registri per codici di condizione: insiemi di bit individuali che indicano condizioni come "risultato zero". Possono essere letti implicitamente dal programma (es. "Jump if zero") ma tipicamente non possono essere impostati direttamente dal programma.

Segmenti di memoria

La memoria principale può essere organizzata logicamente come un insieme di segmenti o spazi di indirizzamento multipli, visibili al programmatore. Una locazione di memoria viene riferita specificando il segmento e la posizione al suo interno. Per supportare questa organizzazione, occorrono registri che memorizzino l'indirizzo base del segmento nella memoria fisica e la sua lunghezza (limite).

Registri di Controllo e di Stato

I principali registri di controllo sono: **Program Counter (PC)** che contiene l'indirizzo della prossima istruzione, **Instruction Register (IR)** che contiene l'istruzione corrente, **Memory Address Register (MAR)** che contiene l'indirizzo di memoria da accedere, e **Memory Buffer Register (MBR)** che contiene i dati letti o da scrivere in memoria.

Program Status Word (PSW)

È un insieme di bit che include i codici di condizione: segno dell'ultimo risultato, zero, riporto, uguale, overflow, abilitazione/disabilitazione interrupt e bit supervisor.

Modo Supervisore

Permette al sistema operativo di utilizzare le procedure del Kernel per agire su componenti critiche del sistema, inclusa l'esecuzione di istruzioni "privilegiate". È disponibile esclusivamente al sistema operativo, non all'utente programmatore.

Altri registri

Possono esistere registri che puntano a: Process Control Blocks, Interrupt Vectors e tabella delle pagine della memoria virtuale. La progettazione della CPU e dei sistemi operativi sono strettamente correlate.

Esempi di organizzazione dei registri

Diverse architetture presentano organizzazioni differenti: il **MC68000** ha 8 registri dati (D0-D7) e 8 registri indirizzi (A0-A7) più Program Status, Program Counter e Status Register. L'**8086** ha registri generali (AX, BX, CX, DX), registri Pointer & Index (SP, BP, SI, DI), registri Segment (CS, DS, SS, ES) e Program Status. L'**80386-Pentium II** estende i registri a 32 bit (EAX, EBX, ECX, EDX, ESP, EBP, ESI, EDI) con FLAGS Register e Instruction Pointer.

Ciclo Fetch/Execute

Il ciclo esecutivo delle istruzioni comprende le fasi: calcolo indirizzo istruzione, fetch dell'istruzione, decodifica dell'operazione, calcolo indirizzo operandi, fetch degli operandi, operazione sui dati, calcolo indirizzo risultato, store del risultato, verifica interrupt e eventuale gestione interrupt. Le fasi possono ripetersi per operandi multipli o dati vettoriali.

Indirettezza

Per recuperare gli operandi può essere necessario accedere alla memoria. La modalità di indirizzamento indiretto richiede più accessi: l'istruzione contiene un indirizzo che punta a una locazione contenente il puntatore all'operando effettivo. L'indirettezza costituisce un sottociclo del ciclo fetch/execute.

Flusso dei dati

Instruction Fetch

Il PC contiene l'indirizzo della prossima istruzione, che viene copiato in MAR ed emesso sul bus degli indirizzi. La Control Unit richiede una lettura dalla memoria principale, il risultato viene trasferito sul bus dati, copiato in MBR e poi in IR. Contemporaneamente il PC viene incrementato.

Data Fetch (Indirettezza)

L'IR viene esaminato; se il codice operativo richiede indirizzamento indiretto, gli N bit più a destra di MBR vengono trasferiti in MAR, l'unità di controllo richiede la lettura dalla memoria, e il risultato (indirizzo dell'operando) viene trasferito in MBR.

Execute

Può assumere molte forme a seconda dell'istruzione: lettura/scrittura della memoria, Input/Output, trasferimento di dati fra registri e operazioni della ALU.

Interrupt

Il contenuto del PC viene salvato per permettere il ripristino dell'esecuzione: il contenuto di PC è copiato in MBR, l'indirizzo di una locazione speciale (es. stack pointer) è caricato in MAR, il contenuto di MBR è scritto in memoria. Poi il PC viene caricato con l'indirizzo della routine di gestione dell'interruzione e si esegue il fetch della nuova istruzione.

Prefetch

Durante la fase di fetch si accede alla memoria principale, mentre durante l'esecuzione tipicamente no. È possibile prelevare l'istruzione successiva durante l'esecuzione della corrente (instruction prefetch).

Limitazioni del prefetch

Il prefetch non raddoppia le prestazioni per due ragioni: le istruzioni di salto (jump/branch) possono rendere vano il prefetch, poiché l'istruzione precaricata potrebbe non essere quella effettivamente eseguita; inoltre la fase di fetch è tipicamente più

breve della fase di esecuzione. Una soluzione è prelevare più istruzioni o aggiungere più fasi per migliorare le prestazioni (pipelining).