

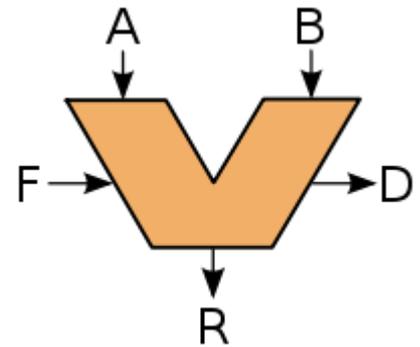
Unità aritmetica e logica

Da Wikipedia, l'enciclopedia libera.

In informatica l'**unità aritmetica e logica** o **unità aritmetico-logica** (abbreviazione utilizzata comunemente: **ALU**, dall'inglese "arithmetic and logic unit" o "arithmetic-logic unit") è una tipologia particolare di processore digitale che si contraddistingue per essere preposta all'esecuzione di operazioni aritmetiche o logiche.

L'ALU è una componente fondamentale della CPU, della GPU e del DSP, ed è presente sia nell'architettura di von Neumann (l'architettura hardware su cui sono basati la maggior parte dei computer moderni) che nella meno diffusa architettura Harvard. Attualmente l'ALU è tipicamente implementata come parte di un microprocessore monolitico. Le moderne CPU e GPU sono dotate di ALU molto sofisticate e un singolo microprocessore può contenere anche più di un'ALU.

Il matematico John von Neumann propose il concetto di ALU nel 1945 mentre sviluppava le basi del computer EDVAC.



Classico simbolo di un'ALU: A e B sono gli operandi in ingresso, R è il risultato, F sono i segnali provenienti dall'unità di controllo e D sono i segnali di stato dell'unità

Indice

Prime ricerche

Sistema numerico

Schema circuitale d'esempio

Operazioni basilari

Operazioni complesse

Input e Output

L'ALU e la FPU

Note

Bibliografia

Altri progetti

Collegamenti esterni

Prime ricerche

Nel 1946 von Neumann lavorò con i suoi colleghi allo sviluppo di un computer per il Princeton Institute of Advanced Studies (IAS). L'IAS machine fu un sistema che fu utilizzato come esempio dalla maggior parte dei computer successivi. Durante lo sviluppo von Neumann delineò l'architettura generica della macchina, l'architettura comprendeva il concetto di ALU.

Von Neumann ideò il concetto di ALU in modo da consentire al computer di eseguire le operazioni aritmetiche elementari come somma, sottrazione, divisione e moltiplicazione. Von Neumann riteneva ragionevole sviluppare un'unità specializzata allo scopo.^[1]

Sistema numerico

L'ALU deve processare i numeri utilizzando lo stesso sistema numerico utilizzato dai rimanenti componenti del computer. Nei moderni computer questa è praticamente sempre la notazione binaria in complemento a due. I primi computer invece utilizzavano una varietà molto ampia di sistemi di numerazione come la notazione decimale, la notazione con segno e il complemento a uno.

A seconda del sistema di rappresentazione utilizzato lo schema dell'ALU può variare significativamente.

Schema circuitale d'esempio

Molte delle operazioni eseguite dal computer vengono svolte dall'ALU. L'ALU recupera i dati dai registri del processore, processa i dati nell'accumulatore e provvede a salvare il risultato nel registro di uscita. Altre strutture provvedono a trasferire i dati dai registri alla memoria principale.^[2]

Un'unità di controllo gestisce l'ALU tramite segnali di controllo che attivano le unità contenute dell'ALU.

Operazioni basilari

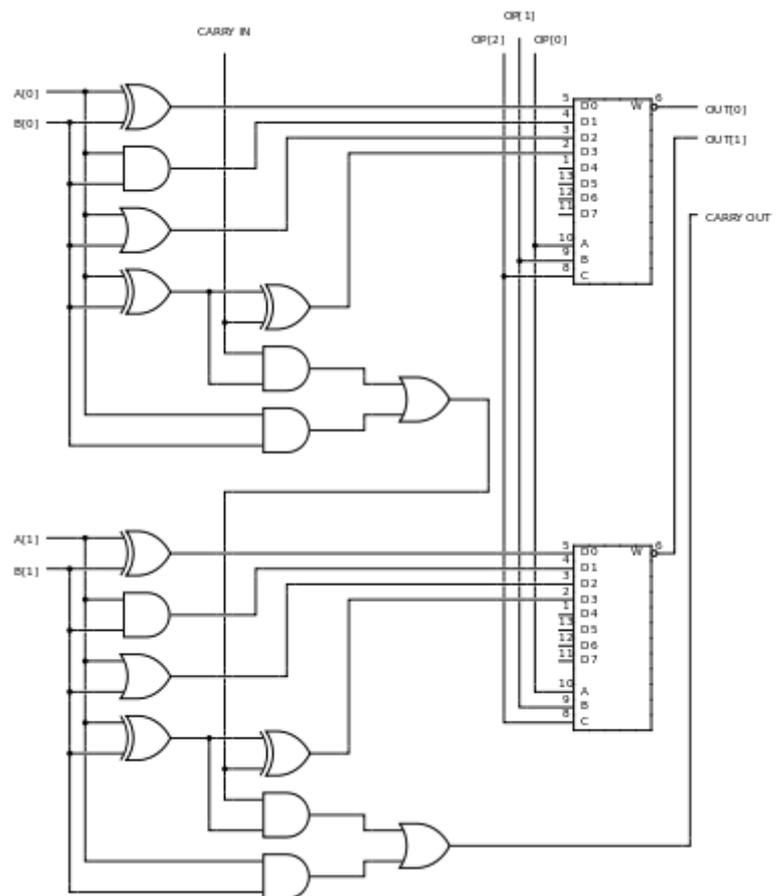
La maggior parte delle ALU sono in grado di eseguire:

- Operazioni aritmetiche su numeri interi (addizione, sottrazione ma spesso anche moltiplicazione e divisione, in alcuni casi svolte tramite l'utilizzo di sottrazioni o somme multiple).
- Operazioni logiche (AND, OR, XOR).
- Operazioni di scorrimento binarie tramite registri a scorrimento.

Operazioni complesse

Un ingegnere elettronico può progettare un'ALU in grado di eseguire tutte le operazioni logiche e matematiche immaginabili ma questo comporterebbe degli enormi problemi dato che l'unità sarebbe molto costosa e lenta. Quindi gli ingegneri devono effettuare delle scelte cercando di realizzare delle unità potenti ma nello stesso tempo non troppo complesse. Per esempio per realizzare l'operazione di radice quadrata gli ingegneri si trovano a poter percorrere le seguenti strade:

1. Progettare un'ALU estremamente complessa in grado di eseguire l'operazione in un singolo passaggio (un singolo ciclo di clock).
2. Progettare un'ALU molto complessa in grado di calcolare l'operazione in una serie di passi. Ma se si organizza l'ALU con una struttura a pipeline si può permettere all'ALU di accettare nuovi dati mentre i dati precedenti sono ancora in elaborazione mascherando i vari cicli di clock necessari per eseguire l'operazione.
3. Progettare un'ALU complessa che esegue l'operazione in più passaggi ma che non utilizza una struttura a pipeline. Questo rende l'operazione più lenta ma semplifica il progetto rispetto alla soluzione a pipeline.



Una semplice ALU a 2 bit In grado di eseguire le operazioni logiche AND, OR, XOR e l'operazione di addizione.

4. Progettare un'ALU semplice che demanda le operazioni di radice quadrata a un'unità specializzata che lavora in parallelo con l'ALU. L'unità specializzata (coprocessore) può essere venduta a parte e acquistata solo dagli utenti che realmente necessitano di eseguire realmente molte operazioni di radice quadrata.
5. Non fornire l'operazione di radice quadrata e demandare il problema al programmatore che dovrà sviluppare delle proprie procedure che utilizzando le operazioni fornite dell'ALU eseguiranno la radice quadrata.
6. Emulare la presenza del coprocessore, quando un programma cerca di eseguire l'operazione il processore verifica la presenza del coprocessore e nel caso sia presente lo utilizza, tuttavia se questo non è presente il processore attiva un interrupt di errore, il sistema operativo riceve l'interrupt ed esegue l'operazione tramite un algoritmo software. Questa soluzione è estremamente lenta.

Le opzioni elencate vanno dalla più veloce (e costosa) alla meno veloce (ed economica). Quindi anche le ALU semplici sono in grado di eseguire operazioni complesse ma richiedono molti cicli di clock per svolgere le operazioni mentre le ALU complesse eseguono le operazioni in un solo ciclo di clock.

I processori per computer implementano l'opzione 1 per le operazioni semplici, l'opzione 2 per la maggior parte delle operazioni e l'opzione 3 per le operazioni estremamente complesse e rare. Questo è permesso dalla presenza di ALU molto complesse, invece processori semplici per esempio per applicazioni embedded spesso implementano la soluzione 3 anche per le operazioni di media complessità mentre non gestiscono nativamente le operazioni complesse demandandole al programmatore (opzione 5).

Input e Output

Gli input (ingressi) dell'ALU sono i dati da processare (gli operandi) e il codice che attiva l'unità di controllo che gestisce l'ALU. L'output sono i risultati delle operazioni.

Le ALU sono progettate per generare anche alcune uscite che segnalano particolari risultati ottenuti. Queste uscite, detti flag, vengono usualmente raccolti in un registro di stato e sono utilizzati per indicare la presenza di resto, la divisione per zero, l'overflow, il risultato pari a zero etc.^[2] Questi flag vengono utilizzati per cambiare il flusso di esecuzione del programma tramite i salti condizionati. Quindi il tipo di risultato ottenuto può servire a controllare delle condizioni: ad esempio, sottraendo due numeri, ottengo il risultato zero se essi sono uguali. Il flag zero si attiva, ed una seguente istruzione di salto può effettuare il salto se il flag zero è stato attivato, oppure proseguire senza salto se il flag zero non è attivato. Gli if..then...else, i cicli e gli altri costrutti della programmazione sono basati sulla combinazione del meccanismo dei flag e dei salti condizionati.

L'ALU e la FPU

La FPU è un'unità preposta all'esecuzione di operazioni aritmetiche su numeri in virgola mobile. Dato che la rappresentazione in virgola mobile è molto più complessa della rappresentazione a complemento a due utilizzata tipicamente dalle ALU. Difatti le FPU tendono ad essere molto più complesse delle ALU, includono circuiti molto complessi e spesso anche più ALU.

Usualmente gli ingegneri tendono a definire ALU le unità in grado di eseguire operazioni sui numeri interi mentre le unità che eseguono operazioni su numeri in virgola mobile, numeri complessi, ecc. ricevono nomi meno generici.

Note

1. [^] Stallings, Pag. 19.
2. Stallings, Pag. 290, 291.

Bibliografia

- Enoch Hwang, *Digital Logic and Microprocessor Design with VHDL*, Thomson, 2006, ISBN 0-534-46593-5.
- William Stallings, *Computer Organization & Architecture: Designing for Performance 7th ed*, Pearson Prentice Hall, 2006, ISBN 0-13-185644-8.

Altri progetti

-  [Wikimedia Commons \(' attr\(href\) '\)](#) contiene immagini o altri file su **[Unità aritmetica e logica \(' attr\(href\) '\)](#)**

Collegamenti esterni

- (EN) *A Simulator of Complex ALU in MATLAB*, su *mathworks.com*.
 - [Harvard Machine 74 \(' attr\(href\) '\) Processore TTL didattico basato sull'ALU 74181](#)
-

Estratto da "https://it.wikipedia.org/w/index.php?title=Unità_aritmetica_e_logica&oldid=106154782"

Questa pagina è stata modificata per l'ultima volta il 29 giu 2019 alle 18:44.

Il testo è disponibile secondo la [licenza Creative Commons Attribuzione-Condividi allo stesso modo](#); possono applicarsi condizioni ulteriori. Vedi le [condizioni d'uso](#) per i dettagli.