Arithmetic Logic Unit [ALU]

Mon, 14 Mar

L'**ALU** è la *parte del processore* che svolge le operazioni **artimetiche e logiche**.

È un insieme di circuiti *combinatori* che svolgono:

- Le operazioni **artimetiche**

  Somma e sottrazione.
- Le operazioni **logiche**

  AND e OR.

Quando si parla di *ALU* a *32 bit*, in realtà ci si riferisce a *32 ALU* ad *1 bit* in *cascata*.

Gestione delle operazioni

Somma artimetica

Esistono due tipi di *ALU* che *gestiscono* l'operazione di **somma**:

- **Half-adder**

  Riceve in entrata solo i *bit da sommare*, perciò è *incapace* di gestire il *carry*, e restituisce il *risultato* della *somma*.
- **Full-adder**

  Ricecve in entrata i *bit* da sommare e l'*eventuale bit di carry* (**CarryIn**), restitusice il *risultato* della *somma* e l'*eventuale carry* (**CarryOut**).

Il **full-adder** è il componente alla base di ogni *elaboratore* moderno, in quanto grazie alla gestione del **carry** è possibile *collegarlo* con *altri full-adder*.

Bit Carry

L'**equazione logica** che definisce il *valore* del bit di *CarryOut* in un *full-adder ALU* è:

```
CarryOut = (b · CarryIn) + (a · CarryIn) + (a · b)
```

Bit Somma

L'**equazione logica** che definisce il *valore* del bit di *Somma* in un *full-adder* ALU è:

```
Sum = (a · ¬b · ¬carryin) + (¬a · b · ¬carryin) + (¬a · ¬b · carryin) + (a · b · carryin)
```

Sottrazione aritmetica

Un *ALU* gestisce l'*operazione aritmetica* della **sottrazione** effettuando la *conversione* in **CA2** del *sottraendo* e *sommandolo* al *minuendo*.

~~Operazione di NOR~~

NOT(a · b)

de morgan

Operazioni di confronto

set-on-less-then [SLT]

Il risultato sarà 1 se $a < b$, 0 altrimenti.

Per eseguire l'operazione di **SLT**, si *azzerano* tutti i bit da *1 a 32*, e si assegna al *bit 0* (**LSB**) il *valore* del *risultato*.

Si utilizza la **sottrazione** per calcolare il risultato di $a < b$: a è minore di b se $(a - b)$ è *negativo*, quindi se il *bit 31* (**MSB**) del risultato di $(a - b)$ equivale a 1.

branch-on-equal [BEQ]

Verifica se i due input sono uguali.

Per eseguire l'operazione di BEQ, si utilizza la **sottrazione**, se il risultato di $(a - b)$ è 0, $a = b$.

A livello *logico*, viene **invertito** l'OR tra tutti i *risultati* della *sottrazione*.

Controllo di overflow

Il rilevamento di un eventuale **overflow** viene individuato tramite la seguente *formula*:

¬a · b · result + a · ¬b · ¬result