

08. Aritmetica del Calcolatore

Unità Aritmetica e Logica (ALU)

L'ALU esegue le operazioni aritmetiche e logiche del calcolatore. Riceve input dalla Control Unit e dai registri, produce output verso i registri e genera flag di stato. Gestisce sia numeri interi che reali.

Rappresentazione degli Interi

Numeri positivi

I numeri positivi si rappresentano direttamente in binario senza necessità di segno (es. $41 = 00101001$).

Rappresentazione in modulo e segno

Il bit più a sinistra indica il segno: 0 per positivo, 1 per negativo. Il resto rappresenta il valore assoluto.

Problemi:

- Le operazioni aritmetiche devono considerare separatamente modulo e segno
- Esistono due rappresentazioni per lo zero (+0 e -0)

Complemento a due

Il bit più significativo ha peso -2^{n-1} . Con n bit si rappresentano numeri da -2^{n-1} a $+2^{n-1} - 1$.

Formula generale per una sequenza $a_{n-1} \dots a_0$:

$$\text{numero} = -2^{n-1} \times a_{n-1} + \sum_{i=0}^{n-2} 2^i \times a_i$$

Interpretazione:

- Numeri positivi: bit più significativo a 0, lettura diretta
- Numeri negativi: bit più significativo a 1; il -1 è rappresentato da n uni, poi si procede all'indietro

Conversione da k a $-k$:

- Metodo 1: copiare i bit da destra fino al primo 1 incluso, poi complementare i rimanenti
- Metodo 2: complementare tutti i bit e sommare 1

Vantaggi:

- Unica rappresentazione dello zero
- Operazioni aritmetiche semplici
- Negazione facile

Intervalli rappresentabili:

- Su 8 bit: da -128 a $+127$
- Su 16 bit: da -32768 a $+32767$

Estensione della lunghezza

Per estendere da n a m bit ($m > n$):

- Numeri positivi: aggiungere zeri a sinistra
- Numeri negativi: replicare il bit di segno nelle nuove posizioni

Casi speciali

- L'opposto di 0 resta 0 (overflow ignorato)

- Il numero -2^{n-1} non ha opposto rappresentabile con lo stesso numero di bit

Operazioni Aritmetiche sugli Interi

Somma e sottrazione

La somma si esegue come normale somma binaria. La sottrazione si realizza sommando il complemento a due del sottraendo: $7 - 5 = 7 + (-5)$.

Overflow: si verifica quando due numeri dello stesso segno producono un risultato di segno opposto. Si riconosce confrontando il bit più significativo del risultato con i segni degli operandi.

Moltiplicazione

Si calcolano prodotti parziali per ogni cifra del moltiplicatore e si sommano. Il prodotto di due numeri a n bit può richiedere $2n$ bit.

Per numeri in complemento a due, la moltiplicazione diretta non funziona se almeno un operando è negativo. Soluzioni:

- Convertire in positivo, moltiplicare, correggere il segno
- Usare l'algoritmo di Booth

Divisione

Più complessa della moltiplicazione, si basa su traslazioni, somme e sottrazioni ripetute.

Rappresentazione dei Numeri Reali

Virgola mobile

I numeri reali si rappresentano come:

$$\pm 1.\text{mantissa} \times 2^{\text{esponente}}$$

L'esponente è polarizzato: si memorizza e_p e il vero esponente è:

$$e = e_p - (2^{k-1} - 1)$$

dove k è il numero di bit dell'esponente.

Normalizzazione

Il bit più significativo della mantissa è sempre 1 e non viene memorizzato (bit implicito). La mantissa memorizzata rappresenta la parte frazionaria dopo l'1 implicito.

Standard IEEE 754

Formati

Formato	Bit totali	Segno	Esponente	Mantissa	Bias
Singola	32	1	8	23	127
Doppia	64	1	11	52	1023

Valori speciali

- **Esponente 1-254, qualsiasi mantissa:** numeri normalizzati $\pm 2^{e-127} \times 1.f$
- **Esponente 0, mantissa 0:** zero (positivo o negativo)

- **Esponente 255, mantissa 0:** infinito (positivo o negativo)
- **Esponente 0, mantissa $\neq 0$:** numeri denormalizzati $\pm 2^{-126} \times 0.f$
- **Esponente 255, mantissa $\neq 0$:** NaN (Not a Number)

Numeri rappresentabili (32 bit)

- Positivi: da 2^{-127} a $2^{128} \times (2 - 2^{-23})$
- Negativi: simmetrici

Numeri non rappresentabili:

- Overflow: troppo grandi in valore assoluto
- Underflow: troppo piccoli in valore assoluto
- Zero esatto (approssimato)

I numeri rappresentati non sono equidistanti: la densità è maggiore vicino allo zero.

Aritmetica in Virgola Mobile

Somma e sottrazione

1. Controllo dello zero
2. Allineamento delle mantisse (rendere uguali gli esponenti shiftando la mantissa del numero con esponente minore)
3. Somma/sottrazione delle mantisse
4. Normalizzazione del risultato

Moltiplicazione

1. Controllo dello zero
2. Somma degli esponenti e sottrazione del bias
3. Moltiplicazione delle mantisse
4. Normalizzazione e arrotondamento

Divisione

Simile alla moltiplicazione ma con sottrazione degli esponenti, aggiunta del bias e divisione delle mantisse.

Eccezioni

- Overflow dell'esponente (numero troppo grande)
- Underflow dell'esponente (numero troppo piccolo)
- Overflow/underflow della mantissa

Precisione

Bit di guardia: bit aggiuntivi nei registri ALU che preservano precisione durante gli shift.

Arrotondamento (metodi):

- Al più vicino (default)
- Verso $+\infty$
- Verso $-\infty$
- Verso zero (troncamento)