

# 1 Lezione del 24-10-24

## 1.0.1 Rappresentazione dei numeri interi

Vogliamo rappresentare numeri interi su  $n$  cifre. Finora avevamo definito la legge di rappresentazione:

$$A = \sum_{i=0}^{n-1} a_i \cdot \beta^i$$

per i numeri naturali.

Nel sistema decimale, usiamo solitamente la rappresentazione **modulo e segno**, cioè usiamo un segno prefisso (+ o -), e poi indichiamo il modulo come un naturale.

In base 2, invece, decidiamo di sfruttare la seguente proprietà: preso un insieme di  $\beta^n$  numeri interi, posso sempre trovare una legge biunivoca che gli fa corrispondere un insieme di  $\beta^n$  numeri naturali.

Definisco quindi una legge  $L : \mathbb{Z} \rightarrow \mathbb{N}$ , e chiamo  $A$  un numero naturale (che indicheremo in Maiuscolo da qui in avanti) e  $a$  un numero intero (che indicheremo in minuscolo da qui in avanti). Si ha quindi che:

$$A = L(a) \Leftrightarrow a = L^{-1}(A)$$

e cioè:

$$a \leftrightarrow^L A \equiv (a_{n-1}a_{n-2}\dots a_1a_0)_\beta$$

cioè con la stessa sequenza di cifre indichiamo sia un naturale che il corrispondente intero. Notiamo che la  $a$  minuscola qui significa *cifra*, che è indifferentemente di  $A$  o di  $a$  (numeri naturali e interi).

Scegliendo leggi  $L$  valide possiamo ottenere dei significativi vantaggi implementativi: ad esempio potremmo definire una legge che permette di usare la stessa circuiteria per le operazioni aritmetiche sia sui naturali che sugli interi.

Il **dominio** di  $L$  dovrà essere contiguo, cioè un'intervallo, magari il più simmetrico possibile rispetto allo zero. Questo è possibile solo se  $\beta$  è dispari. Nel caso di  $\beta$  pari, come sarà nel nostro caso di interesse  $\beta = 2$ , dovremmo prendere un numero "in più" a destra o a sinistra. Nel sistema adottato (sarà il complemento a 2) prendiamo il numero a sinistra, cioè quello negativo, ergo avremo l'**intervallo di rappresentabilità**:

$$\left[ -\frac{\beta^n}{2}, \frac{\beta^n}{2} - 1 \right]$$

Notiamo che da qui in poi assumeremo di lavorare in  $\beta$  pari, in quanto in caso contrario dovremmo usare la seguente notazione:

$$\left\lfloor -\frac{\beta^n}{2} \right\rfloor, \quad \left\lceil -\frac{\beta^n}{2} \right\rceil, \quad \dots$$

che appesantirebbe la trattazione, cosa inutile in quanto abbiamo stabilito che il nostro interesse finale è trovare metodi che si applichino a  $\beta = 2$ .

Abbiamo infatti che, posto  $\beta = 2$ , l'intervallo riportato precedentemente rappresenta quello a cui siamo abituati per la rappresentabilità dei numeri interi in complemento a 2:

$$\left[ -2^{n-1}, 2^{n-1} - 1 \right]$$

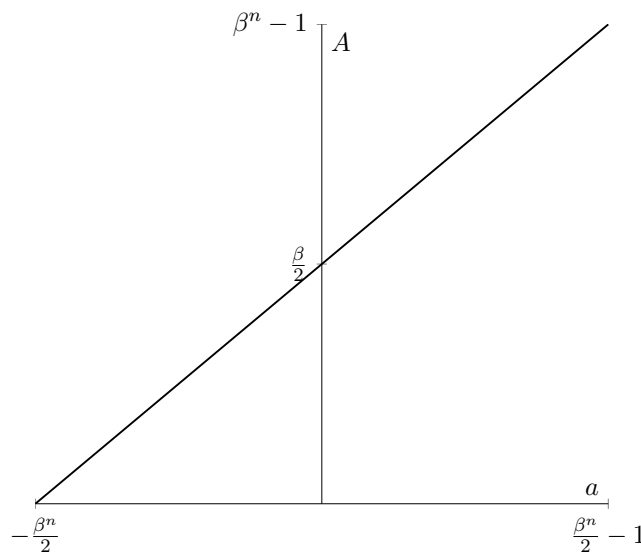
### 1.0.2 Rappresentazione in traslazione

Una possibile legge di rappresentazione è data da:

$$L(a) = A = a + \frac{\beta^n}{2}$$

chiamiamo  $\frac{\beta^n}{2}$  **fattore di polarizzazione**. Questa rappresentazione è utile, **monotona** ( $a < b \Leftrightarrow A < B$ ), e viene usata nei convertitori analogico/digitale e digitale/analogico, dove viene chiamata *binario bipolare*. Inoltre, ricordiamo che rappresenta l'esponente nei numeri reali in virgola mobile secondo lo standard IEEE 754.

Su un grafico dove le ordinate rappresentano  $A$  e le ascisse  $a$ , abbiamo la mappa:



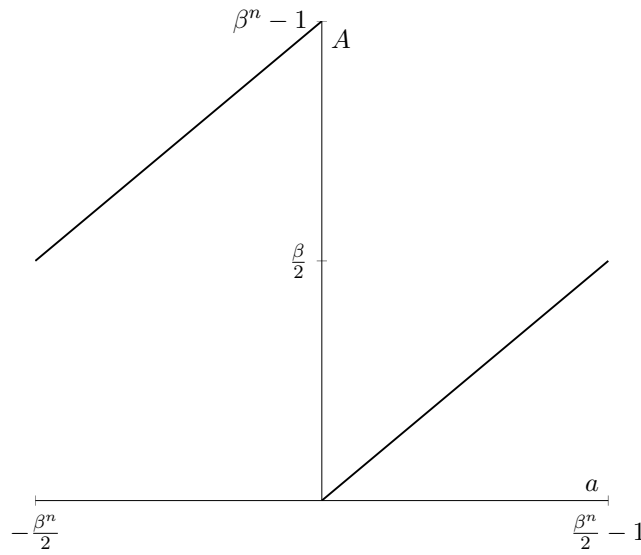
### 1.0.3 Complemento alla radice

Definiamo la legge, che abbiamo solitamente chiamato **complemento a 2**:

$$L(a) = A = \begin{cases} a, & 0 \leq a < \frac{\beta^n}{2} \\ \beta^n + a, & -\frac{\beta^n}{2} \leq a < 0 \end{cases}$$

Con questa legge perdiamo la **monotoneità**. Ciò nonostante, è la legge usata di norma dal processore.

Ricordiamo che graficamente si può rendere con il cosiddetto diagramma a farfalla:



#### 1.0.4 Modulo e segno

Possiamo sempre usare la legge di rappresentazione in modulo e segno, cioè  $(s, M) \leftrightarrow a$ :

$$s = \begin{cases} 0, & a \geq 0 \\ 1, & a < 0 \end{cases}$$

$$M = \text{abs}(a)$$

Notiamo che noi intendiamo, per modulo e segno, una rappresentazione dove  $n$  bit rappresentano il modulo, e l' $n$ -esimo bit rappresenta il segno, per un totale di  $n + 1$  bit. Non stiamo quindi mettendo in relazione intervalli di naturali con intervalli di interi, ma intervalli di naturali complementati da un bit in più di segno, con intervalli di interi. Abbiamo quindi che questo tipo di rappresentazione non ricade nella categoria definita prima. Ricordiamo comunque che viene applicata per rappresentare il segno dei numeri reali in virgola mobile secondo lo standard IEEE 754.

#### 1.0.5 Determinazione del segno

Posso determinare il segno di un numero intero  $a$  dalle cifre della sua rappresentazione  $A$ :

$$\begin{cases} a \geq 0 \Leftrightarrow 0 \leq A < \frac{\beta^n}{2} \\ a < 0 \Leftrightarrow \frac{\beta^n}{2} \leq A < \beta^n \end{cases}$$

Facciamo le solite considerazioni:

- Il **massimo numero rappresentabile** è  $\frac{\beta^n}{2} - 1$ , che in CR ha rappresentazione  $\left(\frac{\beta}{2} - 1, \beta, \dots, \beta\right)_\beta$ ;
- Il **minimo numero rappresentabile** è  $\frac{\beta^n}{2}$ , che in CR ha rappresentazione  $\left(\frac{\beta}{2}, 0, \dots, 0\right)_\beta$ ;
- Lo **0** coincide in  $A$  e  $a$ , ergo vale 0 e ha rappresentazione in CR  $(0, \dots, 0)_\beta$ ;
- Il **-1** è  $\beta^n - 1$ , che in CR ha rappresentazione  $(\beta, \dots, \beta)_\beta$ .

Quindi, per capire se la rappresentazione  $A$  è un numero naturale maggiore o minore di  $\frac{\beta^n}{2}$ , che equivale a capire se l'intero che rappresenta è maggiore o minore di zero, basta guardare la cifra più significativa:

$$\begin{cases} a_{n-1} < \frac{\beta}{2} \Leftrightarrow 0 \leq A < \frac{\beta^n}{2} \\ a_{n-1} \geq \frac{\beta}{2} \Leftrightarrow \frac{\beta^n}{2} \leq A < \beta^n \end{cases}$$

Questo ci permette di riscrivere  $L$  nella forma più elegante:

$$L(a) = A = \begin{cases} a, & a_{n-1} < \frac{\beta}{2} \\ \beta^n + a, & a_{n-1} \geq \frac{\beta}{2} \end{cases}$$

### 1.0.6 Legge inversa del CR

Possiamo ottenere per sostituzione la legge inversa della legge di rappresentazione CR:

$$L(a) = A = \begin{cases} a, & 0 \leq a < \frac{\beta^n}{2} \\ \beta^n + a, & -\frac{\beta^n}{2} \leq a < 0 \end{cases}, \quad L^{-1}(A) = a = \begin{cases} A, & 0 \leq A < \frac{\beta^n}{2} \\ A - \beta^n, & \frac{\beta^n}{2} \leq A < \beta^n \end{cases}$$

Che possiamo riscrivere nel modo più elegante:

$$L^{-1}(A) = a = \begin{cases} A, & a_{n-1} < \frac{\beta}{2} \\ -(\bar{A} + 1), & a_{n-1} \geq \frac{\beta}{2} \end{cases}$$

usando quanto detto sulla MSD e quanto conosceamo sui complementi (ancora, è sostanzialmente un complemento a 2).

In particolare, il cambio delle disequazioni viene fatto note le proprietà sul MSD. La trasformazione  $A - \beta^n = -(\bar{A} + 1)$  si ricava invece dalla proprietà fondamentale:

$$A + \bar{A} = \beta^n - 1$$

con semplici passaggi algebrici.

### 1.0.7 Forma alternativa del CR

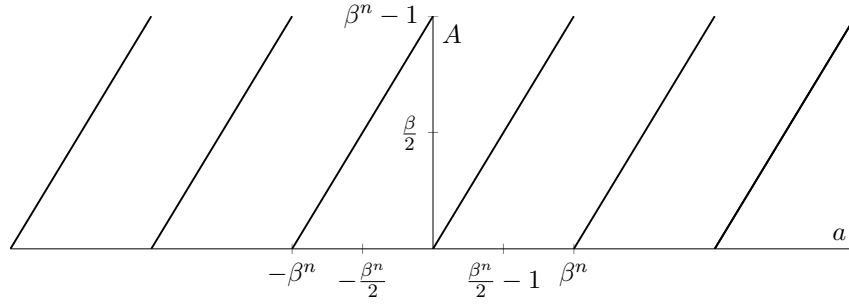
Possiamo usare la forma più concisa (ma anche più pericolosa) delle legge di rappresentazione CR:

$$L(a) = A = |a|_{\beta^n}, \quad \text{se } -\frac{\beta^n}{2} \leq a < \frac{\beta^n}{2} - 1$$

Abbiamo che:

1. Se  $a \geq 0$ , allora è anche  $< \beta^n$ , quindi  $A = a = |a|_{\beta^n}$ ;
2. Se  $a < 0$ , allora è compreso in  $[-\frac{\beta^n}{2}, 0[$ , quindi diviso  $\beta^n$  dà quoziente  $-1$ , cioè dal teorema della divisione con resto,  $a = -1 \cdot \beta^n + |a|_{\beta^n}$ . Avevamo dalla legge di rappresentazione in complemento a radice che, sotto questa ipotesi, volevamo esattamente  $\beta^n + a$ , ergo si ottiene ugualmente  $A = |a|_{\beta^n}$ .

Possiamo quindi interpretare il complemento a radice come una rappresentazione modulare:



Occorre fare attenzione in quanto questo è vero solo nel caso  $a$  sia **rappresentabile**, cioè se rispetta:  $-\frac{\beta^n}{2} \leq a < \frac{\beta^n}{2} - 1$ . In caso contrario, come è chiaro dal grafico, si potrebbero avere le stesse rappresentazioni per interi diversi fra di loro (cioè essenzialmente un overflow).

## 1.1 Operazioni su interi in CR

Vogliamo progettare circuiti che lavorano sulle rappresentazioni, come avevamo fatto per i naturali. Ricordiamo che la rappresentazione vale per un naturale  $A$  o un intero  $a$  a seconda di quanto deciso dal programmatore, e nient'altro.

### 1.1.1 Valore assoluto

Vogliamo trovare il valore assoluto di un numero intero  $B = \text{abs}(a)$ , con:

$$a \in \left[-\frac{\beta^n}{2}, \frac{\beta^n}{2} - 1\right] \Rightarrow B \in \left[0, \frac{\beta^n}{2}\right]$$

Si ha che  $B$  è un numero naturale rappresentabile su  $n$  cifre. Sappiamo che:

$$\text{abs}(a) = \begin{cases} a, & a \geq 0 \\ -a, & a < 0 \end{cases}$$

Posso quindi ottenere  $\text{abs}(a)$  complementando la rappresentazione nel range di valori che so essere negativo (cioè quando  $a_{n-1} \geq \frac{\beta}{2}$ ):

$$B = \text{abs}(a) = \begin{cases} A, & a_{n-1} < \frac{\beta}{2} \\ \bar{A} + 1, & a_{n-1} \geq \frac{\beta}{2} \end{cases}$$

Logicamente, questo sarà rappresentato da un multiplexer che discrimina fra  $A$  e  $\bar{A} - 1$ . La variabile di comando sarà data da  $b_{out}$  di un comparatore fra  $a_{n-1}$  e  $\frac{\beta}{2}$ .

In base 2 questo è notevolmente più semplice: avrò che basta prendere  $a_{n-1}$  come variabile di comando. Si potrà quindi complementare con uno XOR fra le cifre di  $A$  e  $a_{n-1}$ , e usare  $a_{n-1}$  anche come ingresso di un incrementatore, con le cifre di  $A$  all'altro ingresso.

## 1.2 Conversione da CR a MS

Vediamo come convertire un numero in CR nella rappresentazione modulo e segno. Prima di tutto notiamo una discrepanza nell'intervallo di rappresentabilità:

$$i_{MS} = [-\beta^n + 1, \beta^n - 1] \not\equiv i_{CR} = \left[-\frac{\beta^n}{2}, \frac{\beta^n}{2}\right]$$

Abbiamo però che  $i_{CR} \subset i_{MS}$ , tolto il bit di segno, quindi l'operazione è sempre fattibile, calcolando l'assoluto e stabilendo:

$$\text{sgn}(a) = \begin{cases} 0, & a_{n-1} < \frac{\beta}{2} \Leftrightarrow a_{n-1} = 0 \\ 1, & a_{n-1} \geq \frac{\beta}{2} \Leftrightarrow a_{n-1} = 1 \end{cases}$$

dove si è riportato il valore di  $a_{n-1}$  in base  $\beta = 2$ .

### 1.3 Calcolo dell'opposto

Vediamo come trovare l'opposto di un numero in CR, quindi dato  $A \leftrightarrow a$ ,  $B \leftrightarrow b$  tale che  $b = -a$ . Questo' operazione non è sempre possibile, a causa dell'asimmetria dell'intervallo di rappresentabilità in CR  $[-\frac{\beta^n}{2}, \frac{\beta^n}{2}]$ : avremo che il numero in  $-\frac{\beta^n}{2}$ , negativo, non ha opposto positivo rappresentabile.

Avremo quindi bisogno di un flag di overflow, diciamo  $ow$ . Le due uscite, l'opposto e  $ow$ , andranno quindi calcolate separatamente. Assumendo  $ow = 0$ , si ha:

$$\begin{aligned} B &= |-a|_{\beta^n} = |-1|_{\beta^n} \cdot |a|_{\beta^n}|_{\beta^n} = |(\beta^n - 1) \cdot A|_{\beta^n} \\ &= |\beta^n \cdot A - A|_{\beta^n} = |-A|_{\beta^n} = |-\beta^n + 1 + \bar{A}|_{\beta^n} = |1 + \bar{A}|_{\beta^n} \end{aligned}$$

cioè si ritrova sostanzialmente la legge di rappresentazione inversa  $L^{-1}$ .

Sappiamo di poter implementare questa legge con una negazione di tutte le cifre, seguita da un incremento di 1. Nel caso precedente, avevamo usato lo XOR in quanto volevamo che la negazione fosse condizionale (pilotata dal bit di segno). In questo caso vogliamo negare sempre, quindi basta una porta NOT.

L' $ow$  viene invece impostato sulla base di un AND fra le cifre più significative  $a_{n-1}$  del numero non negato e  $b_{n-1}$  del numero negato.

#### 1.3.1 Richiamo all'assembly

Si ricorda che in assembly avevamo l'istruzione **NEG**, che interpretava una sequenza di bit come un numero intero, e ne calcolava l'opposto se possibile, impostando il flag OF altrimenti.

### 1.4 Estensione di campo per gli interi

Avevamo detto che l'estensione di campo per gli interi richiedeva logica. Possiamo infatti ricavare, per via algebrica il valore dell'intero con l' $n$ -esima cifra aggiunta,  $A_{EST}$ :

$$A_{EST} = \begin{cases} a, & 0 \leq a < \frac{\beta^n}{2} \\ \beta^{n+1} + a, & -\frac{\beta^n}{2} \leq a < 0 \end{cases} = \begin{cases} A, & 0 \leq a < \frac{\beta^n}{2} \\ \beta^n \cdot (\beta - 1) + a, & -\frac{\beta^n}{2} \leq a < 0 \end{cases}$$

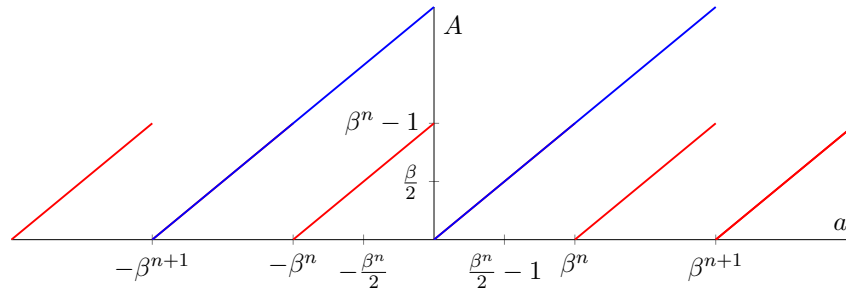
da cui troviamo:

$$= \begin{cases} 0 \cdot \beta^n + A, & 0 \leq a < \frac{\beta^n}{2} \\ \beta^n \cdot (\beta - 1) + a, & -\frac{\beta^n}{2} \leq a < 0 \end{cases}$$

Notiamo che i termini che moltiplicano  $\beta^n$  sono quelli della cifra che vogliamo aggiungere. Possiamo quindi definire la cifra aggiunta:

$$a_n = \begin{cases} 0, & a_{n-1} < \frac{\beta^n}{2} \\ \beta - 1, & a_{n-1} \geq \frac{\beta^n}{2} \end{cases}$$

Graficamente, possiamo pensare all'estensione di campo come una traslazione verso l'alto del lato sinistro dell'intervallo di ordinate, cioè quello che rappresenta gli interi negativi. Più propriamente, se avevamo dato la rappresentazione  $|a|_{\beta^n}$  per  $A$ , adesso dobbiamo prendere  $|a|_{\beta^{n+1}}$  che graficamente dà:



da cui si nota ancora meglio che il lato per interi positivi resta tale, mentre il lato negativo trasla in alto (in rosso si ha  $|a|_{\beta^n}$ , e in blu  $|a|_{\beta^{n+1}}$ ).

Dal punto di vista della base 2, questo tipo di estensore può essere realizzato semplicemente replicando la  $n - 1$ -esima cifra.