Corso di Calcolatori Elettronici I
A.A. 2012-2013

Esercizi

Rappresentazione di numeri interi

ing. Alessandro Cilardo

Accademia Aeronautica di Pozzuoli
Corso Pegaso V “GArn Elettronici”

Interi senza segno

- Qual è l'intervallo di rappresentazione di numeri interi senza segno su **6 bit**?
- Rappresentare, se possibile, il numero **91** su 6 bit.
- Rappresentare, se possibile, il numero **61** su 6 bit.
- Si vuole assegnare un codice identificativo numerico a ciascuno dei circa 60 milioni di Italiani. Qual è il numero minimo di bit da usare per codificarlo?
- **27** e **14** sono ciascuno rappresentabili su 5 bit. Si può rappresentare la loro somma sullo stesso numero di bit?
- Guardando la rappresentazione del numero **123** in binario, scrivere la rappresentazione di **81**=123-32-2-8 senza fare alcuna operazione di calcolo.

Interi senza segno: somma

- Applicando l'algoritmo di somma tra interi in binario, scrivere la rappresentazione di **121**+**47**.
- Indicare qual è il numero minimo di bit della rappresentazione con il quale la precedente somma non genera overflow.
- Da un punto di vista matematico, che significato ha ignorare il bit di riporto eventualmente generato oltre l'ultima cifra rappresentabile?
  - Ad esempio, su 4 bit: 1101 + 1001 = (*1*) 0110

Interi senza segno: shift

- Ottenere la rappresentazione in binario di **48**=12*2^2, partendo dalla rappresentazione di 12=$(1100)_2$.
- Senza fare alcuna operazione di calcolo, scrivere la rappresentazione di **11**+**13***2^4 partendo dalle rappresentazioni di 11 e 13.
- Lo shift a destra può essere interpretato come un'operazione di divisione per 2 (approssimata). Qual è il massimo errore che si può commettere facendo lo shift a destra di due bit ed ignorando i bit che traboccano? Ad esempio:
  - 101010 → 001010
  - 1101000 → 0011010

Interi senza segno: shift

- Calcolare **13** * **33** in binario senza segno facendo uso soltanto di operazioni di addizione e shift.

- Suggerimento:
  osservare che $(33)_{10} = (100001)_2 = 2^5+1$

Interi senza segno

- E’ possibile generalizzare quanto fatto nell’esercizio precedente per moltiplicare due numeri binari qualsiasi?
- Esempio: Calcolare **12** * **13** in binario senza segno, facendo uso soltanto di shift e ripetute operazioni di addizione.
  - suggerimento: per ogni 1 nella rappresentazione del moltiplicatore (**13**) scrivere una versione opportunamente shiftata del moltiplicando (**12**), e poi sommare ripetutamente.

Frazionari in virgola fissa

- E’ una normale rappresentazione posizionale pesata, con l’unica differenza che le potenze di 2 (i *pesi*) non partono da $\mathbf{2^0}$ ma da da una potenza negativa.
- Esempio su **9** bit, di cui **3** frazionari:

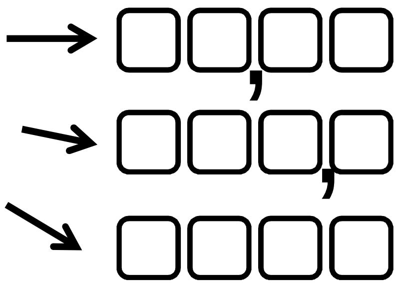
$$1\ 0\ 1\ 1\ 1\ 0\ ,1\ 0\ 1 \quad = \ 32 + 8 + 4 + 2 + \mathbf{1/2} + \mathbf{1/8}$$

$$= \mathbf{46,625}$$

2^5 2^4 2^3 2^2 2^1 2^0 $\mathbf{2^{-1}}$ $\mathbf{2^{-2}}$ $\mathbf{2^{-3}}$

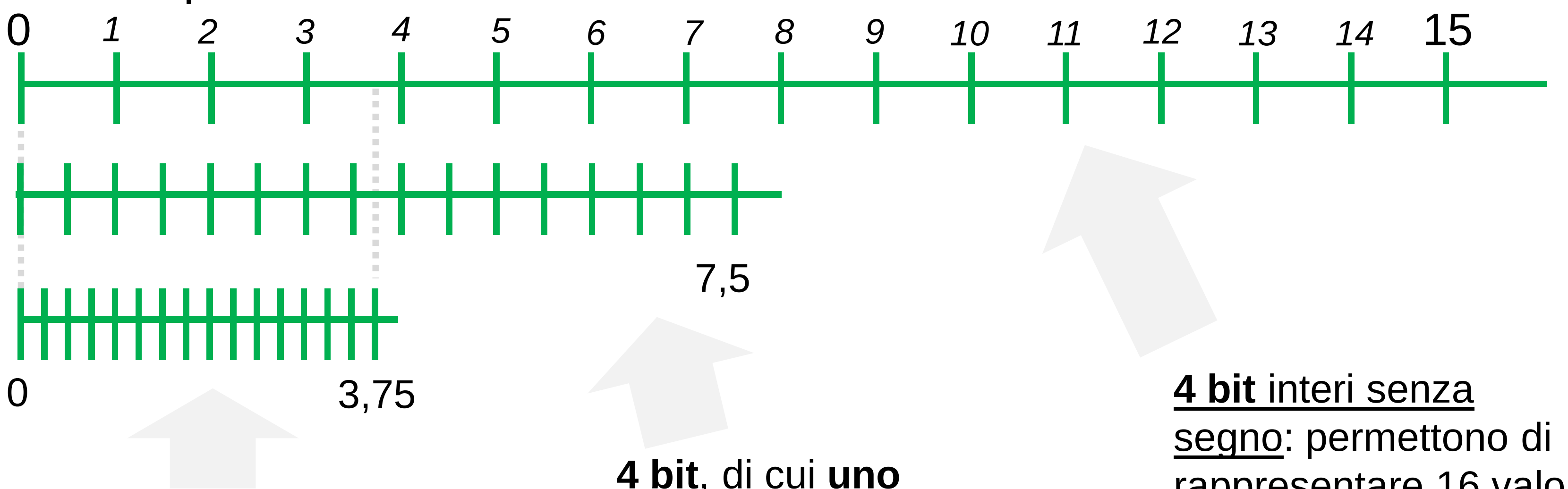
Frazionari in virgola fissa

- Qual è il più piccolo numero *diverso da zero* rappresentabile su 9 bit di cui 3 frazionari?
- Qual è il più grande numero rappresentabile su 9 bit di cui 3 frazionari?
  - suggerimento: è il più grande numero intero rappresentabile su 9 bit, diviso per $2^3=8$
- Quali sono il più grande numero ed il più piccolo numero diverso da zero rappresentabili su 4 bit, cui:
  - due frazionari
  - uno frazionario
  - zero frazionari



Frazionari in virgola fissa

- I numeri in virgola fissa permettono di bilanciare variamente l'ampiezza dell'intervallo di rappresentazione e la risoluzione con cui sono rappresentati i frazionari: maggiore è l'una, minore è l'altra. Esempio:



<u>4 bit</u><u>, di cui</u> **<u>2</u>** <u>frazionari</u>: permettono di rappresentare 16 valori tra 0 e **3,75** (inclusi) con passo pari a **0,25**

<u>4 bit</u><u>, di cui</u> **<u>uno</u>** <u>frazionario</u>: permettono di rappresentare 16 valori tra 0 e **7,5** (inclusi) con passo pari a **0,5**

<u>4 bit</u> <u>interi senza segno</u>: permettono di rappresentare 16 valori tra 0 e **15** (inclusi) con passo pari ad **1**

Frazionari in virgola fissa

- Usando il metodo delle moltiplicazioni successive, scrivere in binario su 8 bit, di cui 4 frazionari la rappresenzione di **0,6875**
- E' possibile rappresentare **21,125** su 8 bit, di cui 4 frazionari?
- Rappresentare **12,625** su 7 bit, di cui 3 frazionari
- I valori precedenti sono tutti rappresentabili in forma esatta. Questo è sempre possibile? Rappresentare su 8 bit di cui 4 frazionari il valore **8,7**
- Qual è il massimo valore dell'errore commesso nella precedente rappresentazione?
- Cosa accade provando a rappresentare il valore **8,7** su un qualsiasi numero di cifre decimali?

Modulo e segno

- Qual è l'intervallo di rappresentazione in modulo e segno su 6 bit?
- Rappresentare in modulo e segno su 8 bit, se possibile, i seguenti valori:

  -141, **+101**, **-3**, **+71**, **-1**
- Qual è il numero minimo di bit per rappresentare la somma di **-71** e **38** in modulo e segno?
- Qual è il numero minimo di bit per rappresentare la somma di **-71** e **-38** in modulo e segno?

Complementi a 2: definizione

- Ricordate che la rappresentazione in complementi a 2 è interpretabile come normale rappresentazione posizionale pesata, in cui però il peso più significativo (quello più a sinistra) è negativo:
- Esempio su 6 bit:

1 0 1 1 0 1 = **-32** + 8 + 4 + 1 = -19

$\underline{\mathbf{-2^5}}$ 2^4 2^3 2^2 2^1 2^0

Complementi a 2: cambio del segno

- Applicando il procedimento per il cambio di segno, rappresentare in complementi a 2 su 4 bit il valore $(\mathbf{-3})_{10}$
- Rappresentare in complementi a 2 su 6 bit il valore $(\mathbf{-30})_{10}$
- Derivare dalla rappresentazione di $(-13)_{10} = (\mathbf{110011})_2$ in complementi a 2 su 6 bit, la rappresentazione di $(+13)_{10}$
- Cosa accade provando a rappresentare il valore **-0** (ovvero, provando a cambiare il segno al valore 0)? Il risultato che si ottiene è corretto?

Complementi a 2

- Qual è l'intervallo di rappresentazione in complementi a 2 su 7 bit?
- Rappresentare, se possibile, il valore **38** su 6 bit in complementi a 2
- Rappresentare, se possibile, il valore -**16** su 5 bit in complementi a 2
- Rappresentare, se possibile, il valore **38** su 7 bit in complementi a 2
- Rappresentare, se possibile, il valore **-18** su 5 bit in complementi a 2
- Rappresentare il valore **-13** in complementi alla base su 5, 6 e su 7 bit. Confrontare le tre rappresentazioni.

Complementi a 2

- Sono dati i valori in complementi di $(-5)_{10}=(\mathbf{1011})_2$ su 4 bit e $(-12)_{10}=(\mathbf{110100})_2$ su 6 bit. Come è possibile effettuare la somma algebrica tra questi due numeri (che sarà su 6 bit), considerando che le rappresentazioni iniziali hanno un numero di bit diverso?
- Se $(10101)_2$ è la rappresentazione di $(-11)_{10}$ su 5 bit, come si rappresenta $(-11)_{10}$ su 8 bit?

Complementi a 2

- Effettuando la somma di **-13** e **-3** in complementi su 5 bit si ha overflow? Com'è possibile riconoscere questa situazione?
- Cosa accade sommando invece **-13** e **-4**?
- Mostrare come viene rilevato l'overflow sommando **13** e **41** su 6 bit in complementi a 2. In cosa differisce questa somma rispetto alla rappresentazione senza segno su 6 bit? Si avrebbe in quel caso overflow?
- La somma di **-16** e **+15** dà overflow in complementi a 2 su 5 bit?
- Calcolare il valore della sottrazione **7-1** in complementi a 2 su 5 bit. Suggerimento: scriverla come somma: 7+(-1).

Complementi diminuiti

- Qual è l'intervallo di rappresentazione in complementi diminuiti su 6 bit?
- Scrivere la rappresentazione in complementi diminuiti, ove possibile, per i seguenti valori:
- **+12** su 8 bit
- **-8** su 4 bit
- **-3** su 6 bit
- **-0** su 4 bit
- **+9** su 5 bit
- **+21** su 5 bit

Complementi diminuiti

- Per ciascuna delle seguenti somme, mostrare come è effettuata l'addizione e, dove opportuno, come viene rilevato l'overflow
  - suggerimento: ricordare i casi in cui è necessario correggere la somma aggiungendo 1
- **5-2** su 4 bit
- **-4-4** su 4 bit
- **12-8** su 6 bit
- **11+8** su 5 bit
- **-13-12** su 6 bit

Eccesso-k

- Rappresentare in eccesso-16 il valore -3
- Quanti bit sono richiesti e qual è l'intervallo di rappresentazione nel caso eccesso-128?
- E' possibile rappresentare in eccesso-16 il valore **-16**?
- E' possibile rappresentare in eccesso-16 il valore **+16**?
- Quale operazione di correzione occorrerà fare sommando i valori **-12** e **5** rappresentati in eccesso-16?
- Rispondere alla domanda precedente per le somme:
- **2** + **1** eccesso-4
- **-3** + (**-2**) eccesso-8
- **3** + **11** eccesso-16

Rappresentazioni: confronto

- Abbiamo introdotto le seguenti rappresentazioni per numeri interi con segno:
  - modulo e segno
  - complementi a 2
  - complementi diminuiti
  - eccesso-k
- Per ciascuna di queste quattro rappresentazioni, spiegare quali sono le operazioni che risultano più facilmente / più difficilmente realizzabili.