



UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

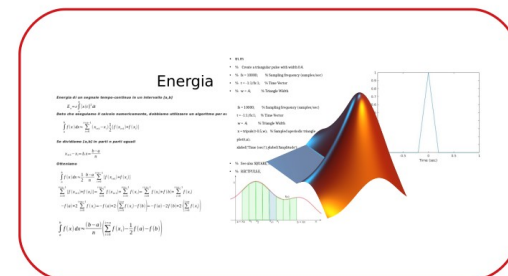
Programmazione Informatica

Ingegneria Civile e Ambientale
Ingegneria Navale

Lezione Teoria 1. Introduzione – Modalità d'Esame

Antonio Fiumara
antonio.fiumara@dia.units.it

A. A. 2025-26





Struttura del Corso

Lezioni ed esercitazioni

Il corso si articola in **13 lezioni** di **3 ore** e **6 esercitazioni** di **1 ora** (mercoledì pomeriggio).

Le 13 lezioni si dividono in due gruppi

6 lezioni di teoria (tendenzialmente martedì)

7 lezioni di programmazione in Matlab (tendenzialmente mercoledì mattina)

A fine corso (mercoledì 3 dicembre) sarà predisposto un esonero per la parte di programmazione (vedi modalità d'esame)

Iscrizione su piattaforma Moodle

<https://moodle2.units.it/course/view.php?id=15724>



Struttura del Corso

Gli argomenti trattati nel corso saranno:

- Lezione Teoria 1 – Introduzione, presentazione del corso, codifica dell'informazione
- Lezione Teoria 2 – Rappresentazione dell'informazione in un computer
- Lezione Teoria 3 – Algebra booleana, introduzione e applicazioni nella programmazione
- Lezione Teoria 4 – Architettura dei calcolatori. HW, SW, Sistemi Operativi
- Lezione Teoria 5 – Algoritmi e strutture dati
- Lezione Teoria 6 – Implementazione di algoritmi notevoli (ordinamento , ricerca, ecc)
- Lezione Programmazione - Matlab 1 – Introduzione, installazione, comandi di base, matrici
- Lezione Programmazione - Matlab 2 – Input/Output (I/O), strutture di controllo e funzioni
- Lezione Programmazione - Matlab 3 – Sistemi Lineari e Algebra lineare
- Lezione Programmazione - Matlab 4 – Analisi matematica
- Lezione Programmazione - Matlab 5 – Funzioni nel dominio del tempo e della frequenza
- Lezione Programmazione - Matlab 6 – Analisi ed elaborazione dei segnali
- Lezione Programmazione - Matlab 7 – Ripasso e approfondimenti
- Esonero esame Programmazione Matlab (3/12/25)



Struttura del Corso

Esercitazioni di gruppo

Durante il corso saranno assegnate esercitazioni di gruppo (max 4 persone per gruppo) da svolgere a casa

Le esercitazioni di gruppo sono facoltative ma saranno valutate (bonus per la votazione dell'esame finale)

Frequenza

La frequenza delle lezioni non è obbligatoria ma è fortemente raccomandata.

Supporto in remoto

Martedì dalle 9:00 alle 10:00 sarò a disposizione da remoto (video chiamata) per eventuali dubbi e fornire chiarimenti sugli argomenti trattati a lezione. La richiesta deve essere inoltrata alla mia email antonio.fiumara@dia.units.it entro il venerdì della settimana precedente. Eventuali altre date possono essere concordate sulla base delle eventuali necessità che potranno emergere

Testi

- Dispense ed esercitazioni fornite dal docente e reperibili alla pagina moodle del corso (<https://moodle2.units.it/course/view.php?id=15724>).
- Testo consigliato per approfondimenti: MATLAB, Edition 6 - A Practical Introduction to Programming and Problem Solving, Dorothy C. Attaway, Ph.D., Publication Date: 06 Sep 2022



Modalità d'Esame

Gli esami si svolgeranno in presenza mediante l'utilizzo della piattaforma Moodle.

Ogni studente dovrà essere dotato del proprio computer o tablet personale per svolgere l'esame. L'aula è dotata di connessione internet wireless e prese di corrente. Per gli studenti che non sono in possesso di computer/tablet personale saranno messe in atto le opportune soluzioni.

La prova d'esame ha una durata complessiva di 2 h e consiste in:

- 1) Prova scritta da svolgere su piattaforma Moodle articolata su 17 domande a risposta multipla (per un totale 30 punti)
 - n. 3 domande da 3 punti - è richiesto lo svolgimento di calcoli
 - n. 7 domande da 2 punti - potrebbe essere richiesto lo svolgimento di calcoli
 - n. 7 domande da 1 punti - domande di teoria
- 2) Realizzazione di un programma in Matlab (solo per chi non ha superato l'esonero)
- 3) Prova orale facoltativa



Modalità d'Esame

Valutazione

Per superare l'esame è necessario

- aver raggiunto la votazione minima di 18/30 nella parte teorica (domande)
- aver ottenuto la votazione minima di 18/30 nella parte di programmazione (o esonero)

La valutazione finale dell'esame sarà data dalla media delle votazioni delle due parti e l'eventuale orale più i bonus maturati con le esercitazioni di gruppo



Modalità d'Esame

Svolgimento

Utilizzo di Moodle per lo svolgimento dell'esame:

Eseguire l'accesso a Moodle con le proprie credenziali istituzionali (units), entrate nella sezione "Esami" e selezionate il quiz

Domande [data esame]

- Una volta risposto a tutte le domande, finalizzate il quiz ("Termina il tentativo" in basso e poi "Invia tutto e termina");
- Salvate gli script nel vostro pc, così nel caso di eventuali problemi di connessione internet al momento della conclusione dello svolgimento del quiz sarà sufficiente reinserirli in Moodle;

Elaborato in Matlab [data esame]

- Caricare su Moodle i file .m e ogni altro file che ritenete utile per la corretta valutazione
- Verificate con il professore se la consegna è andata a buon fine
- Gli script Matlab devono essere funzionanti in maniera corretta

Consultazione appunti e materiali durante lo svolgimento dell'esame

E' possibile consultare liberamente il materiale e gli esercizi del corso ed è necessario utilizzare Matlab;

Accettazione o rifiuto del voto

Riceverete una mail quando gli esami saranno stati corretti: prendete visione del voto su esse3 per l'accettazione o l'eventuale rifiuto entro i termini previsti.

Potete visionare gli eventuali commenti ai singoli esercizi su Moodle, andando sul medesimo quiz.

Gli studenti che non hanno raggiunto la sufficienza saranno considerati "Ritirati"



UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Introduzione

Informatica



Informatica

Informatica (o Scienza dell'informazione): insieme delle discipline riguardanti la rappresentazione e l'elaborazione automatica delle informazioni:

informazione: qualsiasi forma espressiva che possa essere compresa ed elaborata dall'uomo;

automatico: le informazioni a disposizione vengono elaborate senza l'intervento diretto dell'uomo, al fine di ottenere un certo risultato (rappresentato a sua volta da informazioni).

Elaborazione automatica delle informazioni tramite calcolatori automatici elettronici digitali o computer (in inglese l'informatica è Computer Science):

digitale: opera su cifre e simboli di un alfabeto;

il computer deve essere istruito dall'uomo su come eseguire l'elaborazione automatica mediante l'assegnazione di istruzioni e l'organizzazione delle informazioni;

l'assegnazione di una ben determinata sequenza di istruzioni all'elaboratore avviene attraverso la scrittura di un programma in un determinato linguaggio di programmazione.



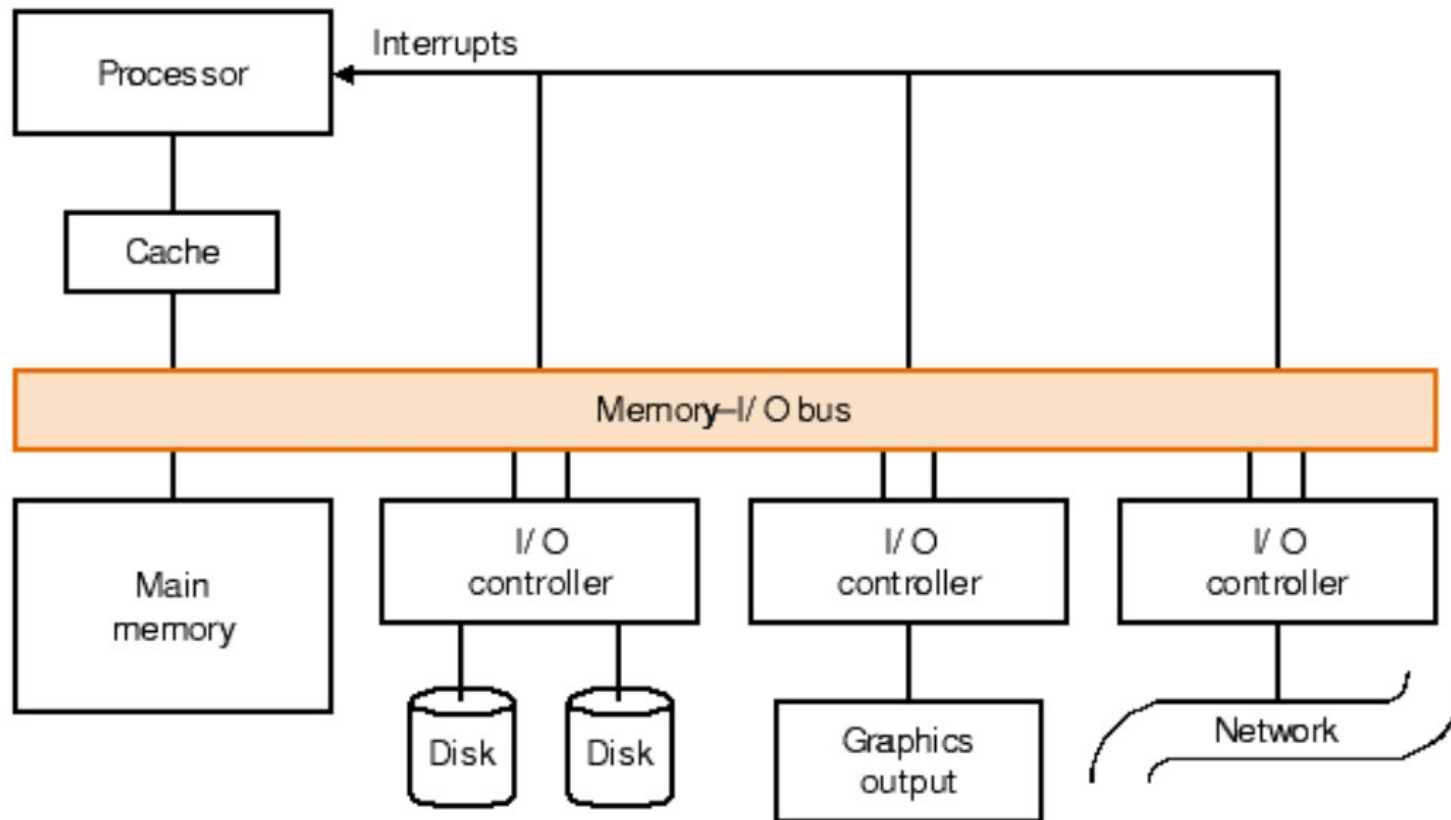
UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Introduzione

Architettura di un calcolatore elettronico



Architettura di un calcolatore





Componenti principali di un calcolatore elettronico

1. Il processore esegue istruzioni
2. La memoria principale (RAM/ROM) contiene i dati e la codifica del programma
3. Dispositivi/periferiche di input/output (I/O). Esempi:
 - tastiera (I)
 - mouse (I)
 - schermo (O)
 - microfono (I)
 - touch screen (I/O)
 - stampante (O)
 - altoparlante (O)
 - memoria di massa/disco (I/O)
 - interfaccia di rete (I/O)
4. Interconnessione fra le varie componenti (Memory bus – I/O bus)



UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Introduzione

Rappresentazione dei dati

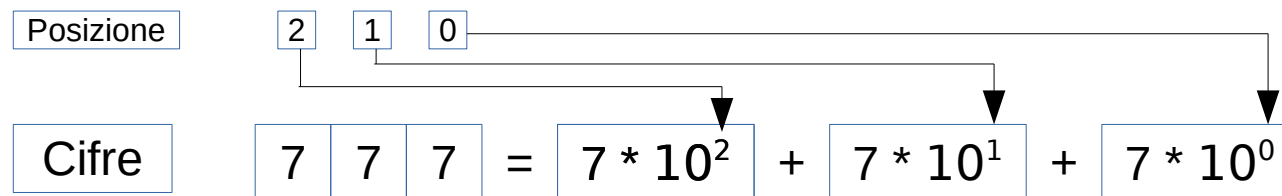
Rappresentazione dei dati - Numerazione posizionale

In un calcolatore elettronico, per la rappresentazione dei dati e la relativa elaborazione si utilizza il sistema di numerazione binario.

I numeri interi che utilizziamo quotidianamente sono basati sul sistema di numerazione posizionale a base 10. Cosa significa?

Il numero è rappresentato da cifre che possono assumere valori da 0 a 9 (cioè da 0 al valore della base - 1)

Le cifre sono rappresentate affiancate e il valore numerico che ogni cifra rappresenta dipende dalla posizione. Ad esempio, se scriviamo 777 intendiamo, procedendo da sinistra verso destra, 7 centinaia (10^2) + 7 decine (10^1) + 7 unità (10^0) = $700 + 70 + 7$. La posizione è l'esponente della base, 10 in questo esempio

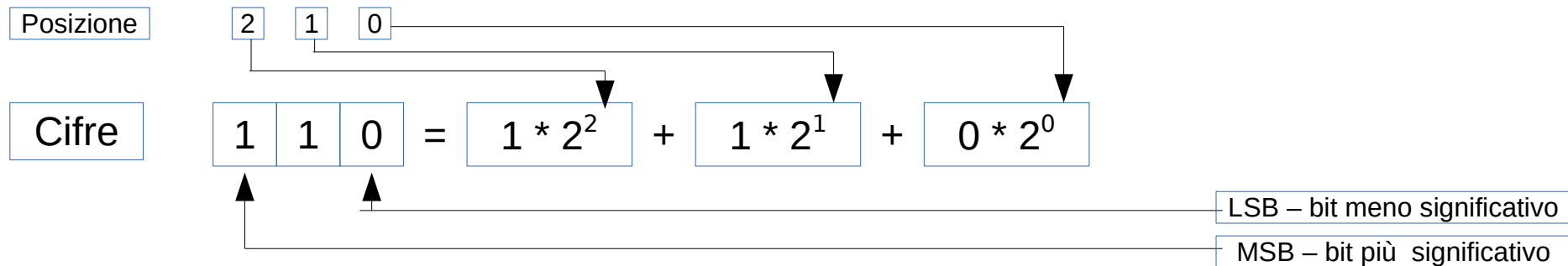




Rappresentazione dei dati – Numerazione binaria

- In un calcolatore elettronico, per la rappresentazione dei dati e la relativa elaborazione si utilizza il sistema di numerazione binario.
- Analogamente al caso della numerazione decimale, un numero binario, cioè un numero posizionale a base=2, è composto da cifre che possono assumere i valori 0 e 1

Esempio $(110)_2 = [1 (2^2) + 1 (2^1) + 0 (2^0)]_{10} = 1 * 4 + 1 * 2 + 0 * 1 = (6)_{10}$



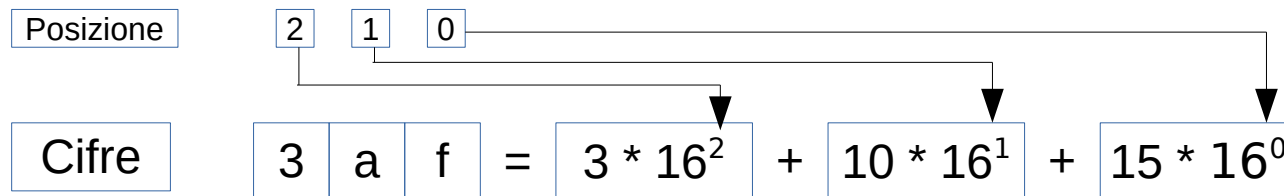
- la cifra binaria è denominata **bit** (dall'inglese binary digit)
- Un insieme di 8 bit è denominato **byte**
- Un numero composto da un byte può assumere al massimo 256 valori
- Un numero composto da N bit, può assumere al massimo 2^N valori
- La notazione binaria si presta agevolmente all'utilizzo dell'algebra booleana



Rappresentazione dei dati – Numerazione esadecimale

- Un numero esadecimale, cioè un numero posizionale a base=16, è composto da cifre che possono assumere i valori 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f

Esempio $(3af)_{16} = [3(16^2) + 10(16^1) + 15(16^0)]_{10} = 3*256 + 10*16 + 15*1 = (943)_{10}$



- Ogni cifra esadecimale può essere contenuta in un gruppo di 4 bit
- Un byte può contenere due cifre esadecimali
- La conversione da binario a esadecimale e viceversa è immediata,
- basta convertire ogni cifra con un gruppo di 4 bit (mediante la tabella accanto)
- $(3af)_{16} = 0011\ 1010\ 1111$
- $1100\ 0001\ 0111\ 0110 = (c176)_{16}$

Hex	Dec	Bin
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111



Conversione da binario a decimale

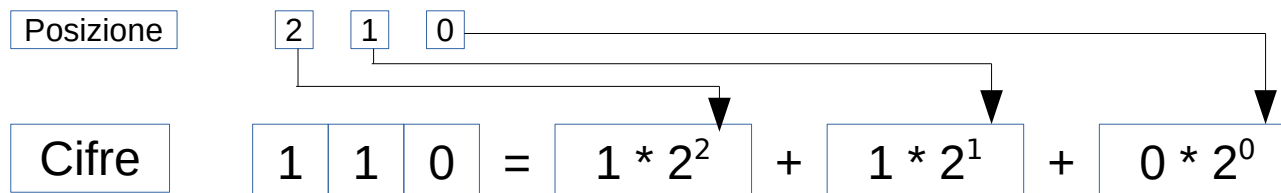
- Per convertire un numero al corrispondente decimale si procede mediante calcolo diretto

Esempio $(110)_2 = [1(2^2) + 1(2^1) + 0(2^0)]_{10} = 1*4 + 1*2 + 0*1 = (6)_{10}$

Esempio $(110100)_2 = [1(2^5) + 1(2^4) + 0(2^3) + 1(2^2) + 0(2^1) + 0(2^0)]_{10} =$
 $= 1*32 + 1*16 + 0*8 + 1*4 + 0*2 + 0*1 = (52)_{10}$

Esempio $(0)_2 = [0(2^0)]_{10} = (0)_{10}$

Esempio $(111111)_2 = [1(2^5) + 1(2^4) + 1(2^3) + 1(2^2) + 1(2^1) + 1(2^0)]_{10} =$
 $= 1*32 + 1*16 + 1*8 + 1*4 + 1*2 + 1*1 = (63)_{10} = 2^6 - 1$





Conversione da decimale a base diversa da 10

- Conversione da base 10 a base $b \neq 10$: conviene utilizzare lo schema iterativo dei resti delle divisioni intere per b , eseguendole in base 10:

$$(d_k \dots d_0)_{10} = \boxed{q_{10}} \cdot b_{10} + \underline{r_{10}}$$

- da base 10 a base 2:

$$(13)_{10} = \boxed{6} \cdot 2 + \underline{1} \Rightarrow a_0 = 1$$

$$(6)_{10} = \boxed{3} \cdot 2 + \underline{0} \Rightarrow a_1 = 0$$

$$(3)_{10} = \boxed{1} \cdot 2 + \underline{1} \Rightarrow a_2 = 1$$

$$(1)_{10} = \boxed{0} \cdot 2 + \underline{1} \Rightarrow a_3 = 1$$

$$(13)_{10} = (\underbrace{a_3 \ a_2 \ a_1 \ a_0})_2 = (1101)_2$$

cifre a_i dalla più significativa (a_3) alla meno significativa (a_0)



Conversione da decimale a base diversa da 10

- Conversione da base 10 a base $b \neq 10$:

- da base 10 a base 8:

$$(87)_{10} = \boxed{10} \cdot 8 + \underline{7} \Rightarrow a_0 = 7$$

$$(10)_{10} = \boxed{1} \cdot 8 + \underline{2} \Rightarrow a_1 = 2$$

$$(1)_{10} = \boxed{0} \cdot 8 + \underline{1} \Rightarrow a_2 = 1$$

$$(87)_{10} = (a_2 a_1 a_0)_8 = (127)_8$$

- da base 10 a base 16:

$$(2607)_{10} = \boxed{162} \cdot 16 + \underline{15} \Rightarrow a_0 = (15)_{10} = F_{16}$$

$$(162)_{10} = \boxed{10} \cdot 16 + \underline{2} \Rightarrow a_1 = 2$$

$$(10)_{10} = \boxed{0} \cdot 16 + \underline{10} \Rightarrow a_2 = (10)_{10} = A_{16}$$

$$(2607)_{10} = (a_2 a_1 a_0)_{16} = (A2F)_{16}$$



Conversione da esadecimale a binario e viceversa

La conversione da binario a esadecimale e viceversa è immediata,

- basta convertire ogni cifra con un gruppo di 4 bit (mediante la tabella accanto)

- $(3af)_{16} = 0011\ 1010\ 1111$

3 a f

- $1100\ 0001\ 0111\ 0110 = (c176)_{16}$

c 1 7 6

Hex	Dec	Bin
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111



Rappresentazione dell'informazione - Codifica

Utilizzando la numerazione binaria, è possibile codificare dati, testi, immagini, suoni e video, istruzioni, ecc.

L'esempio più semplice ed immediato è costituito dalla codifica dei numeri interi positivi.

Per rappresentare dei testi, in metodo più immediato è quello di associare un numero binario ad ogni carattere.

Per le immagini esistono molti tipi di codifica. Bitmap, JPEG, TIFF, PNG, GIF, ecc.

Analogamente per le codifiche video. MPEG-2, MPEG-4, ecc.



Rappresentazione di numeri interi

Codifica in binario puro

Supponiamo di voler utilizzare un byte, (il nostro sistema numerico ha un «parallelismo» di 8 bit), per rappresentare il nostro numero e di utilizzare una codifica in binario puro. In questo caso, tutto quello che dobbiamo fare è trasformare il nostro numero intero in un numero binario di 8 bit

La prima domanda che dobbiamo porci è quale sia il valore massimo che posso rappresentare. Dato che con 8 bit sono possibili 256 valori, con questa codifica posso rappresentare numeri interi da 0 a 255. Con 16 bit si possono rappresentare numeri interi positivi da 0 a 65'535. In generale, su N bit, con la codifica in binario puro posso rappresentare numeri interi da 0 a $2^N - 1$.

0 → 0000 0000

1 → 0000 0001

2 → 0000 0010

3 → 0000 0011

...

254 → 1111 1110

255 → 1111 1111



Operazioni su numeri interi

Codifica in binario puro

Consideriamo le 4 operazioni algebriche elementari, $A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \dots\}$

Somma $A + B$

Sottrazione $A - B$

Moltiplicazione $A * B$

Divisione A / B

Spostamento a destra

Spostamento a sinistra

Vediamo come risolverle e quali insidie possono nascondersi dietro ad un semplice calcolo



Operazioni su numeri interi

Codifica in binario puro

$A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \dots\}$, parallelismo 8 bit

Somma: $S = A + B$, si procede esattamente come con le somme tra numeri in base 10,

$$A = 1111\ 0110 + (246)_{10}$$

$$B = 0000\ 0101 = (5)_{10}$$

$$S = 1111\ 1011 \quad (251)_{10}$$

N.B. Il valore massimo che può assumere $A+B$ è 255. Cosa accade se $A + B$ supera il valore massimo possibile? Il risultato sarà errato (**overflow**). Un sistema ben progettato deve rilevare l'overflow e comunicare all'utente il corrispondente codice di errore in modo che l'errore possa essere gestito

$$A = 1111\ 0110 + (246)_{10}$$

$$B = 0000\ 1101 = (13)_{10}$$

$$S = 0000\ 0011 \quad (3)_{10} \quad !!$$



Operazioni su numeri interi

Codifica in binario puro

$A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \dots\}$, parallelismo 8 bit

Differenza: $D = A - B$, si procede esattamente come con le somme tra numeri in base 10,

$A = 1111\ 0110 - (246)_{10}$

$B = 0000\ 0101 = (5)_{10}$

$D = 1111\ 0001 \quad (241)_{10}$

N.B. Se $A < B$ l'operazione non è possibile. Un sistema ben progettato deve rilevare l'errore e comunicare all'utente il corrispondente codice di errore in modo che l'errore possa essere gestito.



Operazioni su numeri interi

Codifica in binario puro

Prodotto: $P = A * B$ $A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \dots\}$

Un algoritmo per il calcolo del prodotto potrebbe essere il seguente:

Si esegue la somma di A con se stesso per un numero di volte pari a B

Esempio: $6 * 3 = 6 + 6 + 6 = 18$

0000 0110 + $(6)_{10}$

0000 0110 + $(6)_{10}$

0000 0110 = $(6)_{10}$

0001 0010 $(18)_{10}$

Un altro metodo per eseguire il prodotto è la cosiddetta “operazione in colonna”, analogamente al caso della moltiplicazione in base 10

0000 0110 * $(6)_{10}$

0000 0011 = $(3)_{10}$

0000 0110 + $(6)_{10}$

0000 1100 = $(12)_{10}$

0001 0010 $(18)_{10}$

N.B. Anche nel caso della moltiplicazione può verificarsi l'errore di **overflow**.



Operazioni su numeri interi

Codifica in binario puro

$$B \neq 0 \quad A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \dots\}$$

$$A/B = Q * B + R$$

Q è il quoziente

R è il resto

Possiamo procedere nel seguente modo

- 1) Pongo $Q = 0$ e $R = 0$
- 2) se $A < B$, pongo $R = A - B$ e termino il calcolo
- 3) altrimenti {incremento(Q), $A = A - B$, vado al punto 2)}

N.B. Se $B = 0$ l'operazione non è possibile. Un sistema ben progettato deve rilevare l'errore e comunicare all'utente il corrispondente codice di errore in modo che l'errore possa essere gestito.

Operazioni su numeri interi

Codifica in binario puro

Possiamo continuare ad utilizzare la codifica binaria pura riservando un bit al segno.

Ad esempio, possiamo utilizzare il bit in posizione 7, quello più a sinistra, per rappresentare il segno e i rimanenti 7 bit (posizione da 6 a 0) per codificare il valore assoluto del numero.

Codifichiamo il segno in questo modo:

Se il numero è positivo, il bit 7 vale 0

Se il numero è negativo, il bit 7 vale 1

Ad esempio il numero su 8 bit 0000 0011 è uguale a +3

Il numero 1000 0011 è uguale a -3.

Semplice ma c'è un problema. Il numero 0000 0000 rappresenta 0, e il numero 1000 0000 cosa rappresenta? Anch'esso rappresenta lo zero? Non abbiamo più una codifica univoca.

Possiamo escludere 1000 0000 dalla codifica. Può funzionare ma non è la soluzione più efficace.

Soluzione: codifica in “complemento a 2”



Codifica in complemento a 2

- 1) Rappresento i numeri positivi in binario puro + un bit (MSB) per il segno
- 2) Rappresento i numeri negativi mediante il complemento a 2 del corrispondente positivo

Esempio

In una codifica ad 8 bit (7+segno)

il numero 9 è rappresentato da 0000 1001

il numero -9 è rappresentato dal complemento a 2 di (0000 1001)



Codifica in complemento a 2

Come si calcola complemento a 2 di un numero di n bit?

Invertire tutti i bit del numero di partenza (operatore not),
sommare 1 al risultato,

Scartare eventuale riporto oltre l'n-esimo bit

Nel caso dell'esempio (numero 9) il complemento a 2 di 0000 1001

Not (0000 1001) $\rightarrow$ 1111 0110

1111 0110 +

0000 0001 =

1111 0111

Quindi -9 $\rightarrow$ 1111 0111



Codifica in complemento a 2

1) Il complemento a due del complemento a 2 di n è uguale a n

Complemento a 2 di (1111 0111) $\rightarrow$ (0000 1000)+
(0000 0001) =
0000 1001

2) Il valore minimo (negativo) rappresentabile su n bit è $2^{(n-1)}$

Esempio: -128 se $n=8$, -32'768 se $n=16$

3) Il valore massimo (positivo) rappresentabile su n bit è $2^{(n-1)}-1$

Esempio: +127 se $n=8$, +32'767 se $n=16$

4) E' possibile eseguire la sottrazione tra due numeri attraverso una somma senza la necessità di esaminare il segno dei numeri

$A - B = A + (-B) = A + \text{complemento a 2 di } B$



Energia

Energia di un segnale tempo-continuo in un intervallo $[a,b]$

$$E_x = \int_a^b |x(t)|^2 dt$$

Dato che eseguiamo il calcolo numericamente, dobbiamo utilizzare un algoritmo per e:

$$\int_a^b f(x) dx \approx \sum_{i=0}^{i=n-1} (x_{i+1} - x_i) \frac{1}{2} [f(x_{i+1}) + f(x_i)]$$

Se dividiamo $[a,b]$ in parti n parti uguali

$$x_{i+1} - x_i = \Delta x = \frac{b-a}{n}$$

Otteniamo

$$\begin{aligned} \int_a^b f(x) dx &\approx \frac{1}{2} \frac{b-a}{n} \sum_{i=0}^{i=n-1} [f(x_{i+1}) + f(x_i)] \\ \sum_{i=0}^{i=n-1} [f(x_{i+1}) + f(x_i)] &= \sum_{i=0}^{i=n-1} f(x_{i+1}) + \sum_{i=0}^{i=n-1} f(x_i) = \sum_{i=0}^{i=n-1} f(x_i) + f(b) + \sum_{i=0}^{i=n-1} f(x_i) \\ &= -f(a) + 2 \sum_{i=0}^{i=n-1} f(x_i) - f(b) + 2 \left(\sum_{i=0}^{i=n-1} f(x_i) - f(b) \right) = -f(a) - 2f(b) + 2 \left(\sum_{i=0}^{i=n-1} f(x_i) \right) \\ \int_a^b f(x) dx &\approx \frac{(b-a)}{n} \left(\sum_{i=0}^{i=n-1} f(x_i) - \frac{1}{2} f(a) - \frac{1}{2} f(b) \right) \end{aligned}$$

- `W1.m`
- % Create a triangular pulse with width 0.4.
- % `fs = 10000;` % Sampling frequency (samples/sec)
- % `t = -1:1/fs:1;` % Time Vector
- % `w = .4;` % Triangle Width

```
fs = 10000; % Sampling frequency (samples/sec)
t = -1:1/fs:1; % Time Vector
w = .4; % Triangle Width
x = tripuls(t-0.5,w); % Sampled aperiodic triangle
plot(t,x);
xlabel('Time (sec)'); ylabel('Amplitude');
```

- % See also SQUARE,...
- % RECTPULSE,

