

## STORIA DEGLI STRUMENTI DI CALCOLO

### Strumenti di calcolo e di acquisizione dati nel Politecnico di Milano

#### anno 1954 - CRC 102A

- costo: 130.000\$ (attuali qualche milione di euro), costruita in 20 esemplari; acquistata con finanziamento del piano Marshall (fondi USA per la ricostruzione postbellica)
- memoria: 1.024 word da 42 bit (attuale 6kBy)
- stack memoria di calcolo: 8 word
- velocità di calcolo 70 istruzioni al secondo
- codifica numeri: 36 bit
- memoria di massa: nastro magnetico da 117 kword
- alimentazione: potenza elettrica installata 20 kW

#### anno 1964 – Calcolatore IBM 7040 - Politecnico

- 32 kword di memoria; byte da 6 bit; ogni numero intero era di 3Byte=18 bit (valori interi memorizzabili compresi tra -32.767 e +32767); word = numero a virgola mobile di 6Byte=36 bit (numeri reali con 8 cifre significative ed esponente di dieci compreso tra -38 e +38)
- data la dimensione del byte, erano memorizzabili e stampabili  $2^6 = 64$  caratteri e quindi solo le maiuscole, i numeri, alcuni simboli
- unità di introduzione dati e programmi: schede perforate
- unità di memorizzazione sistema operativo, dati e programmi: 6 unità a nastro magnetico a sette tracce:
  - il primo nastro magnetico conteneva il sistema operativo del calcolatore;
  - il secondo nastro magnetico conteneva il compilatore Fortran;
  - il terzo nastro magnetico conteneva la libreria delle routines matematiche e di sistema;
  - il quarto nastro magnetico era utilizzato per memorizzare le fasi intermedie delle compilazioni;
  - il quinto nastro magnetico era utilizzato per memorizzare il programma eseguibile;
  - il sesto nastro magnetico era a disposizione e poteva essere utilizzato per fornire dati al programma o per memorizzare in forma numerica i risultati delle operazioni di calcolo
- unità di output: stampante; perforatore di schede per conservare i programmi compilati; nastro magnetico per conservare i risultati o eventualmente trasferire ad un plotter risultati da disegnare
- velocità di calcolo: su numeri interi circa 120k operazioni/s,

### **anno 1968 - Calcolatore Univac 1108 - Politecnico**

- 128 kword di memoria; byte da 6 bit; ogni numero intero era di 3Byte=18 bit; word = numero a virgola mobile di 6Byte=36 bit
- unità di introduzione dati e programmi: schede perforate o nastro magnetico
- unità di memorizzazione sistema operativo e, temporaneamente, dati e programmi: tamburo magnetico (fastrand)
- unità di output: stampante seriale; perforatore di schede; nastro magnetico
- velocità di calcolo: su numeri interi circa 1M operazioni/s,

### **anno 1972 - Calcolatore HP 2100A - Meccanica**

- 16 kword di memoria, incrementati successivamente a 24kword con un investimento di circa 5 milioni di lire pari a più di 40k€ odierni; byte da 8bit; word da 2 Byte =16bit; numero a virgola mobile da 32bit
- unità di introduzione dati e programmi: nastro perforato (teletype) preparato offline su telescrivente esterna
- unità di memorizzazione dati e programmi; n. 2 dischi rimovibili da 5Mword (10MBy); un disco utilizzato per il sistema operativo, il secondo, personale di ogni utente, per i programmi ed i dati
- unità di conversione analogico-digitale: 24 canali, 12 bit, 100kHz frequenza massima teorica, 10kHz frequenza di campionamento su 4 canali per tempo indefinito
- velocità di calcolo: su numeri interi circa 1M operazioni/s,
- NB era il calcolatore di governo delle missioni spaziali Apollo, quelle che hanno portato l'uomo sulla Luna

### **Strumenti attuali: anno 2014**

**Legge di Moore:** la velocità dei calcolatori raddoppia ogni 18 mesi; oggi esistono calcolatori con velocità di calcolo misurate in T operazioni/s ( $10^{12}$  operazioni al secondo)

#### **- Telefonino tipico odierno**

- velocità di calcolo: 1,5G operazioni/s (=1.500 volte HP2100A)
- 1 GBy di memoria (= 300.000 volte HP2100A)
- 32GBy memoria di massa (=3.200 volte HP2100A)

## Curiosità

- 1 - Il sistema operativo dei primi calcolatori elettronici era pensato per occupare il minor numero possibile di memorie per cui, per esempio, i numeri interi da 0 a 100 erano memorizzati in maniera fissa nella memoria; tutte le volte che serviva un numero intero minore di 100, questo non veniva ridefinito ma il programma si indirizzava sulla memoria appropriata. Con errori di programmazione si poteva cambiare il contenuto di quelle caselle di memoria; l'effetto era che la assegnazione  $A=1$  poteva dare come risultato non il valore 1 ma il valore che l'utente, per errore, aveva caricato sulla corrispondente casella di memoria.
- 2 - I valori negativi venivano memorizzati come complemento ad 1 dei bit che li definivano; come risultato erano ottenuti e memorizzati, a seconda della successione delle operazioni di calcolo, sia il valore  $+0$  che il valore  $-0$ ; se si voleva controllare se il risultato di una operazione era nullo bisognava confrontare col valore 0 il valore assoluto del risultato ottenuto.
- 3 - tutto il calcolatore era dedicato allo svolgimento di un singolo programma di calcolo e le successioni delle operazioni erano rigorosamente seriali; data la lentezza di calcolo e la impossibilità di capire cosa stesse facendo il calcolatore, quando non succedeva nulla per qualche minuto, non veniva cioè stampato nessun risultato intermedio, veniva il dubbio che, per un errore di programmazione, il calcolatore ripettesse indefinitamente sempre la stessa serie di operazioni e quindi normalmente veniva dato l'ordine di abortire il programma. Nel caso di un calcolo molto complesso di integrazione che utilizzava un algoritmo iterativo, per controllare lo stato di avanzamento del programma decidemmo di far stampare ad ogni passo di integrazione la frase "Viva la torre di Pisa"; quando vedemmo che la frase veniva stampata ogni cinque minuti dovemmo decidere di limitare la precisione richiesta nella convergenza del risultato.

## FFT Fast Fourier Transform

La limitata memoria disponibile nei calcolatori elettronici dell'epoca, nonché la bassa velocità di calcolo, ha indotto nel 1967 due matematici dell'IBM, Cooley e Tukey, ad ideare un algoritmo che permettesse di calcolare, malgrado le limitazioni di cui sopra, la trasformata di Fourier di funzioni definite mediante un numero non limitatissimo di punti.

Dopo il campionamento di un segnale analogico di misura con frequenza  $f_c=1/\Delta t$  si ottengono N valori della funzione intervallati del tempo  $\Delta t$  per un tempo totale  $T=N \Delta t$ ; su questi N valori può essere eseguito il calcolo della serie di Fourier, che, rispetto al calcolo riferito ad una funzione continua periodica con periodo  $T_s$ :

$$x_t = \bar{x}_{t0} + \sum_{i=1}^{\infty} (A_i \cos(i\omega_1 t) + B_i \text{sen}(i\omega_1 t)) \quad (1-4)$$

$$\bar{x}_{t0} = \frac{1}{T_s} \cdot \int_{-T_s/2}^{T_s/2} x_t \cdot dt \quad = \text{valor medio della funzione } x_t \quad (1-5)$$

$$A_i = \frac{2}{T_s} \cdot \int_{-T_s/2}^{T_s/2} x_t \cdot \cos(i\omega_1 t) \cdot dt \quad (1-6)$$

$$B_i = \frac{2}{T_s} \cdot \int_{-T_s/2}^{T_s/2} x_t \cdot \text{sen}(i\omega_1 t) \cdot dt \quad (1-7)$$

si trasforma come segue:

il tempo assume solo i valori discreti:  $t_j = (j \cdot \Delta t)$

la pulsazione della armonica fondamentale<sup>1</sup>  $\omega_1$  vale:  $\omega_1 = 2\pi/T = 2\pi/N\Delta t$

il prodotto  $t_j \cdot \omega_1$  vale:  $t_j \cdot \omega_1 = j \cdot 2\pi/N$

e quindi l'espressione della trasformata di Fourier per una funzione reale campionata (Discrete Fourier Transform) diviene:

$$x(j \cdot \Delta t) = x(t_j) = x_j = \bar{x}_{t0} + \sum_{i=1}^N (A_i \cos(i \cdot j \cdot 2\pi/N) + B_i \text{sen}(i \cdot j \cdot 2\pi/N)) \quad (1-4')$$

$$\bar{x}_{t0} = \frac{1}{N} \cdot \sum_{j=1}^N x_j \quad = \text{valor medio dei valori } x_j \quad (1-5')$$

$$A_i = \frac{2}{N} \cdot \sum_{j=1}^N x_j \cdot \cos(i \cdot j \cdot 2\pi/N) \quad (1-6')$$

$$B_i = \frac{2}{N} \cdot \sum_{j=1}^N x_j \cdot \text{sen}(i \cdot j \cdot 2\pi/N) \quad (1-7')$$

---

<sup>1</sup> Si ricordi che mentre la armonica fondamentale di una funzione è definita dal periodo della funzione stessa, la armonica fondamentale di una funzione campionata è definita dal tempo totale di campionamento facendo insorgere la presenza di armoniche "fittizie" aventi ampiezza nulla se vengono campionati punti per un tempo multiplo intero del periodo della funzione o problemi di leakage se il tempo totale di campionamento non è multiplo intero del periodo della funzione.

Cooley e Tukey constatarono che se  $N$  è una potenza di 2,  $N=2^n$ , possono essere risparmiati molti calcoli in quanto gli argomenti di seno e coseno, che teoricamente avrebbero numerosità  $N^2$ , sono in effetti solo gli  $N$  calcolati per gli argomenti:

$$1 \cdot 2\pi / N ; 2 \cdot 2\pi / N ; 3 \cdot 2\pi / N ; \dots ; N \cdot 2\pi / N$$

e quindi bastava calcolare i soli  $\sin(1 \cdot 2\pi/N)$  e  $\cos(1 \cdot 2\pi/N)$  e ottenere successivamente tutti gli altri valori necessari di seni e coseni utilizzando le formule trigonometriche:

$$\cos\left((k+1) \cdot \frac{2\pi}{N}\right) = \cos(k \cdot 2\pi/N) \cdot \cos(1 \cdot 2\pi/N) - \sin(k \cdot 2\pi/N) \cdot \sin(1 \cdot 2\pi/N)$$

$$\sin\left((k+1) \cdot \frac{2\pi}{N}\right) = \sin(k \cdot 2\pi/N) \cdot \cos(1 \cdot 2\pi/N) + \cos(k \cdot 2\pi/N) \cdot \sin(1 \cdot 2\pi/N)$$

occorrevano cioè solo 4 moltiplicazioni e due somme invece che i più lunghi calcoli delle funzioni seno e coseno.

Idearono inoltre una successione facilmente programmabile delle operazioni di calcolo che permetteva di

- utilizzare la coppia di valori di seno e coseno appena calcolati per tutte le operazioni nelle quali servivano senza quindi la necessità di memorizzarli o di doverli ricalcolare,
- memorizzare i risultati intermedi al posto dei dati di partenza senza dover utilizzare ulteriori posizioni di memoria.

Il risultato finale era quindi di:

- ridurre al minimo lo spazio di memoria necessario => possibilità di calcolare la trasformata di Fourier di funzioni campionate con un numero elevato di punti malgrado la limitatezza delle memorie disponibili
- ridurre al minimo le operazioni necessarie => il numero di operazioni invece di essere proporzionale ad  $N^2$  era proporzionale ad  $N$  => velocizzazione del processo anche con  $N$  elevati

grazie a queste due caratteristiche chiamarono l'algoritmo sviluppato Fast Fourier Transform.

## Curiosità storiche: la moltiplicazione quando non si conosceva lo zero

### Gli antichi Egizi hanno inventato il calcolo binario?

Come noto, dalle elementari, basta la conoscenza delle tabelline dall'1 al 10, per essere in grado di effettuare la moltiplicazione di due numeri qualsiasi. Questo risultato è ottenuto grazie a due invenzioni arrivate da noi in epoche relativamente recenti, ovvero circa nel 1.200:

- l'invenzione dello zero, numero che separa i numeri positivi dai numeri negativi (si ricordi, ad esempio che l'anno precedente la nascita di Gesù Cristo è indicato col numero -1, l'anno della nascita è indicato col numero +1: per calcolare gli anni trascorsi a cavallo fra due date, una prima di Cristo l'altra dopo Cristo, bisogna sommare un uno alla differenza fra i due valori)
- l'introduzione della numerazione posizionale decimale (per gli antichi Romani, I=1, X=10, C=100, M=1.000,  $\bar{X}$ =10.000,  $\bar{C}$ = 100.000, ...)

Come era possibile effettuare le moltiplicazioni con numerazione non posizionale e senza lo zero?

Gli antichi Egizi scrivevano su due colonne i due numeri da moltiplicare, dividevano per due (senza resto) il numero della prima colonna, moltiplicavano per due il numero della seconda colonna; se il risultato sulla prima colonna non era pari ad uno iteravano l'operazione sino a che, dopo un numero sufficiente di divisioni, il valore della prima colonna diventava pari ad uno;

Il risultato della moltiplicazione era ottenuto sommando i soli valori, della seconda colonna, in corrispondenza di valori dispari dei numeri della prima colonna, come qui sotto indicati in giallo .

Si veda nell'esempio seguente cosa succede nel caso di moltiplicazione 177x77:

2**num	num	dispari	a	b	b solo dispari	a solo dispari
1	0	1	177	77	77	1
2	1	0	88	154	0	0
4	2	0	44	308	0	0
8	3	0	22	616	0	0
16	4	1	11	1.232	1.232	16
32	5	1	5	2.464	2.464	32
64	6	0	2	4.928	0	0
128	7	1	1	9.856	9.856	128

La somma dei numeri nella colonna "b, solo dispari" è 13.629 pari al prodotto 77\*177,

la somma dei numeri della colonna "a, solo dispari" è pari a 177:

13.629 è ottenuto cioè come:  $77*2^0+77*2^4+77*2^5+77*2^7=77*(1+16+32+128) = 77*177$