

# Hackerando la Macchina Ridotta

Una moltiplicazione che non **voleva** funzionare

Diego Ceccarelli

27 novembre 2008



# Table of contents

- 1 La macchina ridotta
  - Progetto
  - La parola
  - Memoria
  - SO
  - Codifica
  - SO(2)
- 2 La Routine di Moltiplicazione
  - Intro
  - L'algoritmo
  - Preparazione
  - Moltiplicazione
  - L'errore

## 1 La macchina ridotta

- Progetto
- La parola
- Memoria
- SO
- Codifica
- SO(2)

## 2 La Routine di Moltiplicazione

- Intro
- L'algoritmo
- Preparazione
- Moltiplicazione
- L'errore



# Il progetto dettagliato

Il 31 luglio 1956 viene presentato il progetto dettagliato della macchina ridotta, le cui caratteristiche principali sono:

- lunghezza di parola di 18 bit;
- memoria a nuclei magnetici di 1024 (1K) parole;
- aritmetica in virgola fissa;
- 32 istruzioni di macchina;
- 70.000 addizioni o 500 moltiplicazioni al secondo;
- lettore fotoelettrico di nastro come dispositivo di ingresso e telescrivente come dispositivo di uscita.



# Il progetto dettagliato: la parola

a <i>3 bit</i>	b <i>5 bit</i>	c <i>10 bit</i>
----------------	----------------	-----------------

- a) i primi 3 bit non erano significativi;
- b) i 5 bit seguenti sono la rappresentazione binaria di uno degli interi (da 0 a 31) associati alle varie istruzioni.
- c) gli ultimi 10 bit sono la rappresentazione binaria di uno degli interi (da 0 a 1023) associati alle diverse celle della memoria, danno, nella maggior parte delle istruzioni, l'indirizzo della cella interessata nell'esecuzione dell'istruzione.



# La memoria: i nuclei di ferrite

- memoria *non volatile*;
- utilizza piccoli anelli magnetici di *ferrite*;
- i dati sono memorizzati tramite la variazione della polarità del campo magnetico degli anelli;
- tempo di lettura di 6 microsecondi
- mantengono lo stato indefinitivamente anche *senza alimentazione*
  - nella teoria
  - in pratica **NO!**
  - → dispositivo di *bootstrap*



# Il sistema operativo

**Tabella di decodifica:** Tabella per la codifica dei simboli *esterni* in simboli *interni* e viceversa.

Per poter essere utilizzata come strumento di calcolo la macchina viene estesa con alcune routine:

- una routine  $M$ , per effettuare la moltiplicazione tra due numeri;
- una routine  $D$ , per effettuare la divisione;
- una routine  $S$ , per leggere numeri in complemento a due e stampare su nastro il segno e cinque cifre decimali;
- una routine  $L$ , per leggere caratteri dal nastro.



# Il codice di Baudot

- Sistema di codifica per un set di caratteri utilizzato nelle telescriventi prima del sistema ASCII;
- Inventata nel 1874 per utilizzare una tastiera a 5 tasti;
- Introduce il carattere di scambio che permette di raddoppiare il numero di simboli rappresentabili;
- Nella MR è utilizzato il codice di Murray nella versione ITA2 (della Western Union Telegraph Company).
- Nell'ITA2 il bit più significativo è a *destra*

00	01	02	03	04	05	06	07
NUL	E 3	LF	A -	SP	S ' I 8	U 7	
08	09	0A	0B	0C	0D	0E	0F
CR	D ENQ	R 4	J BEL	N ,	F !	C :	K <
10	11	12	13	14	15	16	17
T 5	Z +	L >	W 2	H £	Y 6	P 0	Q 1
18	19	1A	1B	1C	1D	1E	1F
O 9	B ?	G &	FIGS	M .	X /	U ;	LTRS
Letters				Figures		Control Chars.	





# Il codice di Baudot

- La codifica era utilizzata per scrivere i programmi sui nastri di input della macchina e sui nastri di uscita;
- All'interno della macchina fu adottata una codifica che rispettasse per i numeri la rappresentazione binaria;
- La conversione in entrata e in uscita era quindi a carico delle routine.



# Organizzazione della memoria

La memoria era organizzata in questo modo:

- nelle celle da 0 a 31 era memorizzata la tabella di conversione tra il codice della telescrivente e la rappresentazione interna;
- le celle dalle 32 alla 36 erano riservate per gli *indici di ripetizione*;
- le celle dalla 37 alla 813 erano memoria utilizzabile per i programmi;
- dalla 813 fino alla 1024 erano allocate le routine.



## 1 La macchina ridotta

- Progetto
- La parola
- Memoria
- SO
- Codifica
- SO(2)

## 2 La Routine di Moltiplicazione

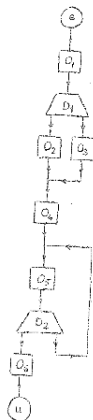
- Intro
- L'algoritmo
- Preparazione
- Moltiplicazione
- L'errore



# La Moltiplicazione che non **voleva** funzionare

Non c'erano i linguaggi ad alto livello e neanche *metalinguaggi*.

Sottoprogramma M: tavola I



$O_1$  : 814  
 $D_1$  : 815  
 $O_2$  : 816-817  
 $O_3$  : 818  
 $O_4$  : 819-822  
 $O_5$  : 823-837  
 $D_2$  : 838  
 $O_6$  : 839-840  
 $u$  : 841  
 cont. : 842-846



- 33 Celle; 28 istruzioni; 5 costanti; 3 celle di lavoro e una di conteggio; Entrata in  $\Delta 814$  istruzione di uscita in  $\Delta 841$ .
- Esegue il prodotto di  $\gamma\Delta 36$  e  $\gamma\Delta 37$
- risultato in doppia precisione in  $A$  e  $B$ .

Alcuni comandi che incontreremo:

- $n+A$   $addr$ : copia il valore all'indirizzo  $addr$  nel registro  $A$ ;
- $AM$   $addr$ : salva il contenuto del registro  $A$  all'indirizzo  $addr$ ;
- $LpA$   $addr$ : esegue l'and bit a bit tra il registro  $A$  e la parola in  $addr$  e salva il risultato in  $A$ ;
- $A+A$   $addr$ : somma il valore all'indirizzo  $addr$  al valore nel registro  $A$ ;
- $nM$   $addr$ : azzera la parola in  $addr$ ;



## Comandi di flusso

- $Z \text{ addr}$  setta il contatore delle istruzioni  $N$  a  $\text{addr}$ ;
- $Z+A \text{ addr}$  setta il contatore delle istruzioni  $N$  a  $\text{addr}$  se il valore in  $A$  è  $\geq 0$
- $Z-A \text{ addr}$  setta il contatore delle istruzioni  $N$  a  $\text{addr}$  se il valore in  $A$  è  $< 0$

Lo shift lungo: il comando  $\text{wD}$  esegue lo shift lungo a destra tra il registro  $A$  e il  $B$ :

**for**  $i=1:16$  **do**

|  $A_{i+1} \leftarrow A_i$ ;

|  $B_{i+1} \leftarrow B_i$ ;

**end**

$A_1 \leftarrow 0$ ;

$B_1 \leftarrow A_{17}$ ;



# Algoritmo di moltiplicazione

$B \leftarrow y;$

**repeat**

$A \leftarrow \gamma\Delta 37;$

If( $A_0 = 1$ )  $A \leftarrow x;$

Else  $A \leftarrow 0;$

$A \leftarrow A + \gamma\Delta 38;$

shift lungo di  $A$  e  $B;$

$\Delta 38 \leftarrow A;$

$\Delta 37 \leftarrow B;$

$A \leftarrow \gamma\Delta 32;$

$A \leftarrow A + 1;$

$\Delta 32 \leftarrow A;$

$A \leftarrow A - 17;$

**until**  $A < 0;$

If  $y < 0$   $A \leftarrow -x;$

$A \leftarrow A + \gamma\Delta 38;$



# Codice: il segno

blocco O1	814 n+A 37	$A \leftarrow y$
blocco D1	815 Z+A 818	if ( $A \geq 0$ ) then <i>go to 818</i>
blocco O2	816 n+A 842 817 Z 819	$A \leftarrow N-a$ 36 <i>go to 819</i>
blocco O3	818 n+A 843	$A \leftarrow Z$ 840

Il blocco successivo quindi viene eseguito il comando  $N-a$  36 con nel registro  $A$  se  $y < 0$ , altrimenti con il comando  $Z$  840.





# Codice: Preparazione

blocco 04	819 AM 839	$B \leftarrow y$ $\Delta 38, \Delta 32 \leftarrow 0$
	820 n+B 37	
	821 nM 38	
	822 nM 32	

Salva il contenuto di  $A$  nella cella vuota 839 e carica il fattore  $y$  in  $B$ . Azzera il contenuto della cella 38 e 32.



# Codice: Moltiplicazione

blocco 05	823	n+A	37	$A \leftarrow \gamma \Delta(37)$ if ( $A_{17} == 1$ ) then $A \leftarrow x$ $A \leftarrow A + \gamma \Delta 38$
	824	LpA	29	
	825	A+A	844	
	826	AM	827	
	827	-	-	
	828	AM	829	
	829	-	-	
	830	A+A	38	
	...			
	844	n+A	845	
845	Z	830		
846	n+A	36		

Codice *automodificante*: si carica un'istruzione nel registro e ci si somma 1 per far variare l'indirizzo a cui l'istruzione è riferita.



# Codice: Moltiplicazione(2)

blocco 05	831 Wd	Shift lungo su $A$ e $B$
	832 AM 38	$\Delta 38 \leftarrow A$
	833 BM 37	$\Delta 37 \leftarrow B$
	834 n+A 32	$A \leftarrow \gamma \Delta 32$
	835 A+A 29	$A \leftarrow A + 1$
	836 AM 32	$\Delta 32 \leftarrow A$
	837 A-A 17	$A \leftarrow A - 17$
	838 Z-A 823	if ( $A < 0$ ) then <i>go to 823</i>

- esegue lo shift lungo;
- salva il risultato in memoria;
- controlla la condizione di iterazione;



# Codice: Trattamento del segno

blocco 06	$Z \ 840 / n-A \ 36$ $A+A \ 38$ <i>istruzione di uscita</i>	$\text{if } (y < 0)$ $\text{then } A \leftarrow -(x)$ $\text{else skip}$ $A \leftarrow A + \gamma\Delta 38$
-----------	---	--

- All'ingresso del blocco blocco,  $A$  contiene 0;
- L'indirizzo 38 contiene la parte più significativa di  $x \cdot |y|$ ;
- Se  $y$  è positivo in  $A$  va  $\gamma\Delta 38$  con il segno di  $x$ ;
- Se  $y$  è negativo e  $x$  è positivo  $A$  viene posto uguale a:

$$A \leftarrow (x \cdot |y|) - x \Rightarrow A \leftarrow x(|y| - 1) \Rightarrow A \leftarrow x(-|y|)$$



# Codice: L'errore

```

823 n+A 37
824 LpA 29
825 A+A 844
826 AM 827
827 - -
828 AM 829
829 - -
830 A+A 38
831 Wd -
...
842 n+A 36
843 Z 840
844 n+A 845
845 Z 830
846 n+A 36

```

- Se il bit meno significativo di  $A$  è 0, viene caricato in  $A$  l'istruzione in posizione 845, ovvero  $Z$  830, che poi viene salvata in 829 ed eseguita:
- ... ma il registro  $A$  contiene ancora l'istruzione  $Z$  830
- Proseguendo il registro viene sommato alle somme parziali in 38 e successivamente shiftato.
- **Soluzione:** il registro  $A$  deve essere posto a 0, prima di proseguire, l'istruzione di skip in 845 viene sostituita con  $n+A$  13
- **Funziona!**