



Programmazione su un calcolatore anni '50 (la Macchina Ridotta)

Esperienze di Programmazione 2013/14
6 maggio 2014





Due parole sul progetto





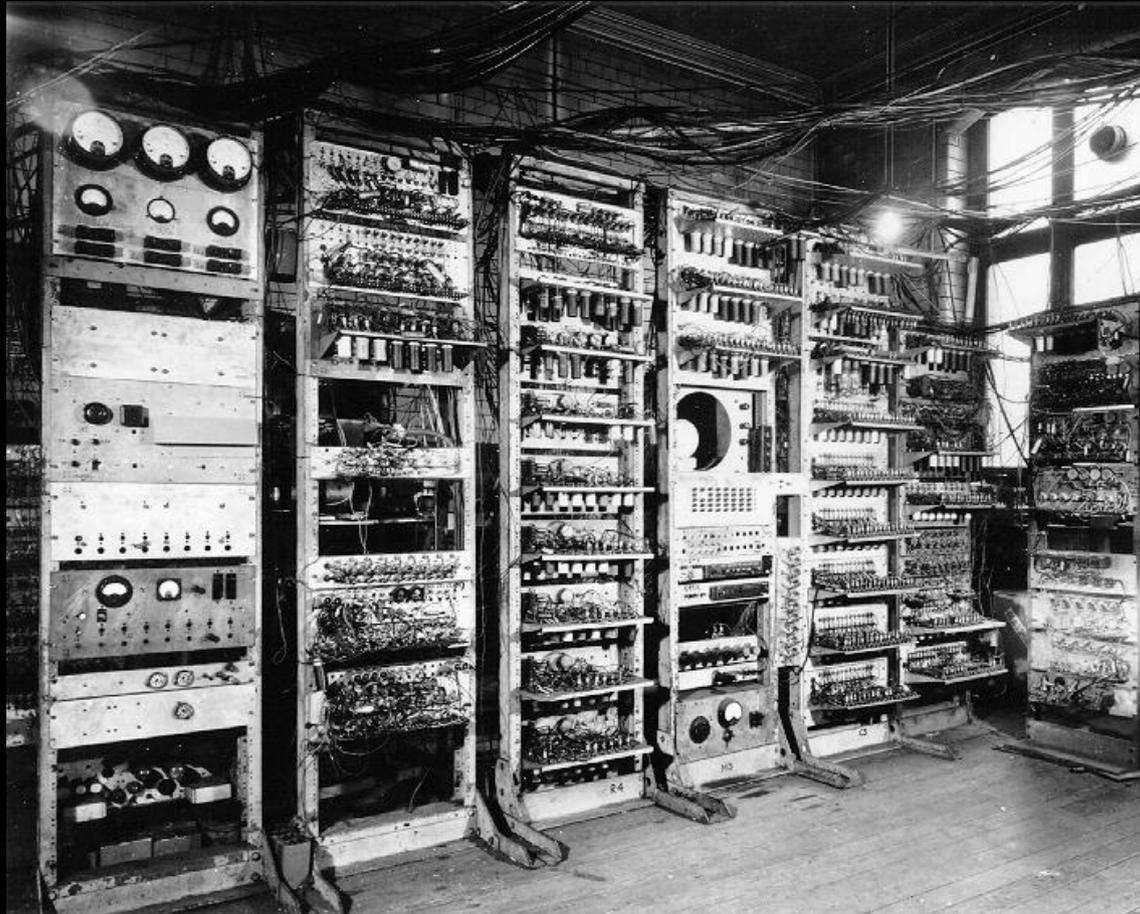
Hackerando la Macchina Ridotta

- Dal 2006
 - Dal 2008, come attività di diffusione
 - Dal 2010, al Museo degli Strumenti per il Calcolo
- Partecipanti
 - Dipartimento di Informatica
 - Dipartimento di Ingegneria Informatica
 - INFN, sezione di Pisa
- Attività
 - Ricerca, la MR, la CEP, gli Olivetti, le meccaniche...
 - Diffusione, allestimenti/attività al Museo, Pagina Q...
 - Dintorni, Caruso, cultura popolare, caratteri...

L'informatica degli anni '50

- 1936, i risultati di Church e Turing
- Vari precursori:
 - 1937-42 ABC Atanasoff-Berry
 - 1940 CNC ai Bell Labs
 - 1943 Zuse Z3
 - 1943-44 Colossus a Bletchley Park
 - 1944 Harvard/IBM Mk1/ASSC
 - 1945 ENIAC
- 1945-46 Rapporti di Von Neumann e di Turing
- 1948 SSEM (Baby) a Manchester

La Baby di Manchester





Primi strumenti di programmazione

- 1949, EDSAC, caricatore (in ROM)
- 1950, UNIVAC, A-0... A-2, Arith-Matic...
- 1951-55, IBM
 - 701, proto assembler
 - 650, Symbolic Optimal Assembly Program
- 1952, MIT Whirlwind, George
- 1957, IBM 704, FORTRAN
- 1959, CoDaSyL, COBOL

19/7/49
 - Kilburn Highest Factor Routine (amended) -

function	C	26	26 ²	27	line	012348	1345
-24 C	-b ₁	-	-	-	1	00011	010
< to 26		-b ₁			2	01011	110
-26 C	b ₁				3	01011	010
< to 27			-b ₁	b ₁	4	11011	110
-23 C	a	tau	-b ₂	b ₂	5	11101	010
subr. 27	a-b ₂				6	11011	001
stop					7	-	011
add 20 to d					8	00101	100
subr. 26	tau				9	01011	001
< to 25	tau	tau			10	10011	110
-25 C					11	10011	010
stop					12	-	011
stop	0	0	-b ₂	b ₂	13		111
-26 C	b ₂	tau	-b ₂	b ₂	14	01011	010
subr. 21	b ₂ !				15	10101	001
< to 27	b ₂ !			b ₂ !	16	11011	110
-27 C	-b ₂ !				17	11011	010
< to 26			-b ₂ !	b ₂ !	18	01011	110
22 to 26	tau	-b ₂ !	b ₂ !		19	01101	000

20	-3	10111 etc
21	1	10000
22	4	00100

or 10100

23	-a
24	b ₁

25	-	tau b ₂
26	-	-b ₂
27	-	b ₂

- Sottoprogrammi...
- Librerie di nastri, letteralmente
- Codice “auto” modificante
- Wheeler jump, sottoprogrammi auto modificanti
- Salto con indirezione, per il ritorno
- Celle parametriche
- Poi, finalmente lo stack



La Macchina Ridotta



- 1954, il sincrotrone mancato, la manovra Fermi
- 1955, il Centro Studi Calcolatrici Elettroniche
- 1956, entra, ufficialmente, Olivetti
- 1957, la Macchina Ridotta
- 1958, le prime applicazioni (servizi!)
- 1959, la crisi, fondi, transistor e separazioni
- 1961, la CEP, in ritardo, ma in gran pompa
- 1962, il CNR e la ripresa

□ Primi acquisti

- 1954, CRC 102A, ottobre, Milano
- 1955, Ferranti Mk1*, primavera, Roma

□ Prime costruzioni

- 1957, Macchina Ridotta, luglio
- 1958, Macchina Zero (Elea 9001V), primavera
- 1959, la macchina di Padova,
- 1959, ELEA 9002, novembre
- 1960, ELEA 9003, primavera
- 1960, ELEA 6001, novembre
- 1961, Calcolatrice Elettronica Pisana, primavera



La Macchina Ridotta



- Macchina parallela (con il significato del tempo)
- 1024 parole da 18 bit, nuclei di ferrite
- 4/8 μ sec di clock, \sim 70000 istruzioni al sec
- 32 istruzioni
 - Controllo microprogrammato
 - 1 microistruzione di fetch, 1 di execute
- I/O su registri, 2 in, 1 out
 - registri di ingresso, pilotante o pilotato
 - telescriventi e lettori/perforatori di nastro

- Nastri perforati, standard telegrafico
 - Legati alla codifica Baudot
 - Introdotti ai primi del 1900 (Murray)
 - Uniformati negli anni '30, CCITT ITA2
 - Periferiche disponibili che parlano a pacchetti di bit

- Pratici
 - Si leggono e si scrivono
 - Si tagliano e si incollano
 - Anche a mano

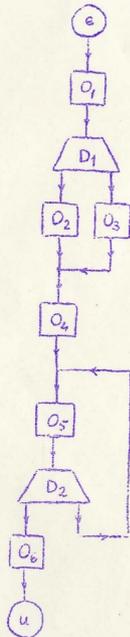
- Alternativi alle schede perforate (USA vs UK)

- In linguaggio macchina
 - Si penserà al Fortran solo con la CEP
 - Assemblato e rilocato a mano (anche minutato?)
- Supporto ai sottoprogrammi
 - Manca il Wheeler jump, nessuno è perfetto
 - Modifica del sottoprogramma da parte del chiamante
- Caricatore di programmi
 - Mnemonico e operando decimale (EDSAC)
 - Previsto nella MR56, sparisce dopo
 - Caricamento di immagini binarie

- Formato fisso per le istruzioni
 - 3 bit (17-15), codice di breakpoint
 - 5 bit (14-10), codice operazione
 - 10 bit (0-9), argomento, indirizzo di memoria
- Codici mnemonici
 - Operando, operazione, destinazione
 - L'altro operando è la cella indirizzata dall'argomento
- Esempi
 - $AM\ x, M[x] = A; B-B\ x, B = B - M[x]$
 - $Z\ x, N = x; Z-A\ x, \text{if } A < 0 \text{ then } N = x$

- 18 -

Sottoprogramma II: 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
 cost. : 842-846

- 19 -

Sottoprogramma II: tavola II

O_1	814	n+A	37					
D_1	815	Z+A	818					
O_2	816	n+A	842	}	830	A+A	38	
	817	Z	819		831	Wd	---	
O_3	818	n+A	843	}	832	AI	38	
	819	AI	839		833	BI	37	
O_4	820	n+B	37	}	834	n+A	32	
	821	nI	38		835	A+A	29	
	822	nI	32	}	836	AI	32	
	823	n+A	37		837	A-A	17	
	824	IpA	29	}	D_2 838	Z-A	823	
	825	A+A	844		O_6 { 839	---	---	
	826	AI	827	}	u 840	A+A	38	
	827	---	---		841	---	---	
	828	AI	829	}	cost.	842	n-A	36
	829	---	---			843	Z	840
						844	n+A	845
						845	Z	830
						846	n+A	36

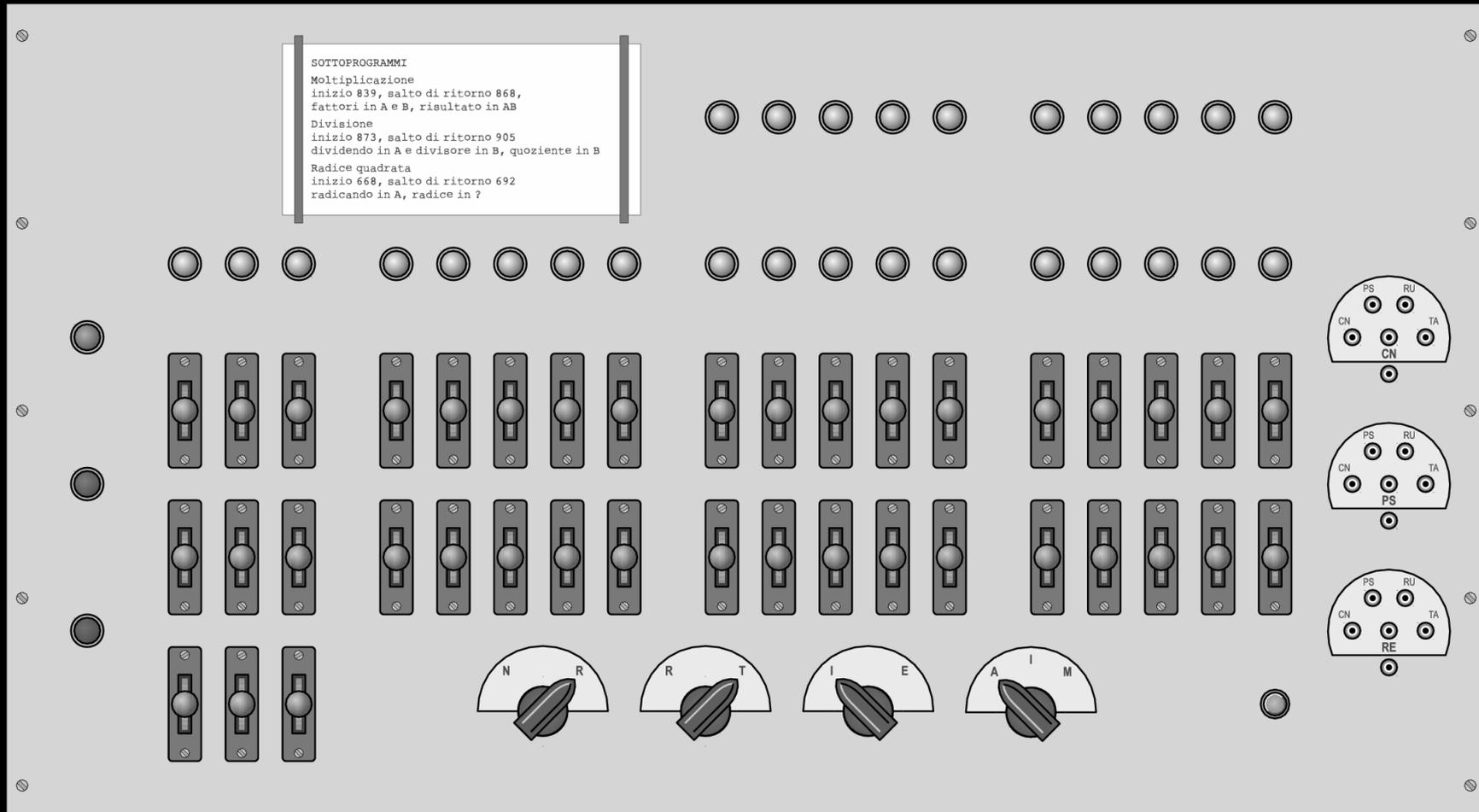


Un esempio

```
100 000 n+A 119 # strt 000 n+A seed // load seed in A
101 000 AM 117 # 000 AM x // copy seed in x to init seq.
102 000 n+A 117 # loop 000 n+A x // load x in A
103 000 Vs 000 # 000 Vs // multiply 2
104 000 Vs 000 # 000 Vs // multiply 4
105 000 Vs 000 # 000 Vs // multiply 8
106 000 Vs 000 # 000 Vs // multiply 16
107 000 Vs 000 # 000 Vs // multiply 32
108 000 Vs 000 # 000 Vs // multiply 64
109 000 Vs 000 # 000 Vs // multiply 128
110 000 A+A 117 # 000 A+A x // multiply 129
111 000 A+A 118 # 000 A+A c // add increment
112 100 AM 117 # 100 AM x // store x+1
113 000 n+B 117 # 000 n+B x // load x in B
114 000 LpB 120 # 000 LpB mask // light off some bits
115 001 BM 121 # 001 BM v // store to visualize
116 000 Z 102 # 000 Z loop // go to generate next number
117 000000 # x 000000 // generated pseudo random value
118 100003 # c 100003 // increment
119 012345 # seed 012345 // seed
120 014798 # mask 014798 // 000 01110 01110 01110
121 000000 # v 000000 // visualized value
```



Interfaccia utente



Il simulatore

□ Modellazione

- Per capire la macchina mentre la si riscopre
- Per tramandarla in un linguaggio moderno
- Per confrontarla con altre macchine

□ Simulazione

- Un modo di provare il modello
- Documentazione incompleta o incomprensibile
- Per completare/verificare il puzzle si va per ipotesi
- Per provare/ricostruire/verificare il software

□ Strumento dimostrativo/educativo (al Museo)



Modellazione, hardware

