

*Il modulo RPN della biblioteca Solid State Software è stato sviluppato per consentire di utilizzare sulle macchine con Sistema Operativo Algebrico (SOA) i programmi originariamente sviluppati per la notazione polacca inversa (RPN).*

*Più precisamente il modulo in questione consente di ottenere una compatibilità praticamente completa tra Texas Instruments, TI 58/59 e Hewlett Packard HP 67/97: basta inserire nella TI 58/59 i codici del programma per HP 67 e il modulo esegue la traduzione in linguaggio SOA: a questo punto si carica il programma nella calcolatrice Texas Instruments ed il gioco è fatto.*

### **Il problema**

Sia il linguaggio SOA che l'RPN sono linguaggi di tipo assemblativo (assembler mnemonici), in cui cioè il programmatore interviene direttamente sulle modalità di funzionamento della macchina (contenuto dei registri di memoria, del contatore di programma, indirizzi ecc.).

Per tale motivo essi risentono pesantemente dell'architettura della macchina per cui sono stati sviluppati. Ciò complica indubbiamente la vita quando si voglia tentare di "tradurre" un programma da un linguaggio di questo tipo ad un altro.

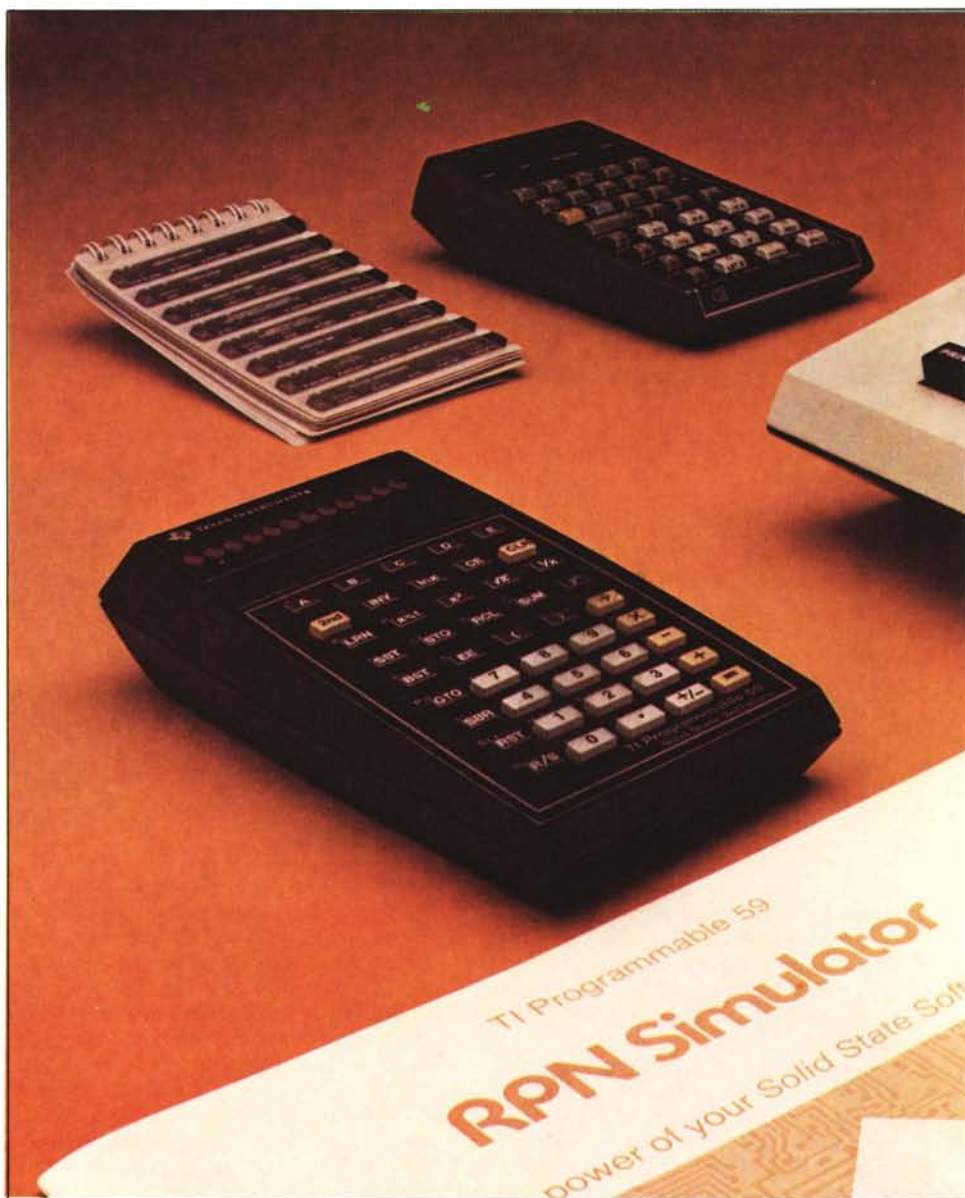
Quando poi si voglia fare ciò con un algoritmo valido in generale non ci si può limitare ad ottenere dal programma tradotto INPUT e OUTPUT uguali all'originale, ma bisogna anche per forza di cose replicare passo passo il funzionamento della macchina di partenza.

Nel nostro caso dobbiamo simulare il funzionamento della catasta operativa e delle istruzioni associate, oltre alle particolarità strettamente "grammaticali" del linguaggio RPN; ecco quindi spiegato il nome di "simulatore" dato al modulo SSS in esame.

### **Limitazioni**

Per forza di cose, l'uso del simulatore RPN è soggetto ad alcune limitazioni di carattere teorico che ne riducono la flessibilità e la speditezza operativa.

La limitazione forse più pesante è data dal fatto che non è possibile tradurre le istruzioni di salto indiretto (GTO (i) e GSB (i)) in quanto le TI 58/59 non consentono l'uso di label numeriche; il difetto purtroppo non è eliminabile, e, come specificato nel manuale, prima di iniziare la traduzione di un programma è opportuno effettuare una "esplorazione" preliminare delle istruzioni per vedere se per caso ve ne siano di tipo "proibito", nel qual caso bisogna



# TEXAS INSTRUMENTS modulo S.S.S. RPN simulator

**di Filippo Merelli**



rinunciare alla traduzione stessa (almeno per mezzo del modulo RPN) o apportare le necessarie modifiche.

L'intraducibilità dell'istruzione MERGE (che consente di unire tra loro più programmi) non provoca grosso fastidio dato il suo scarso impiego.

Particolare attenzione va posta nell'effettuare gli INPUT del programma tradotto: bisogna tenere presente che il modulo RPN non dispone di ENTER automatico (come i primi calcolatori in notazione polacca) e che questa funzione viene replicata da tastiera con la subroutine SBR; può quindi succedere che a causa di un ENTER non effettuato in ingresso il programma tradotto fornisca dei risultati sballati pur funzionando perfettamente.

Altre limitazioni sorgono nel caso di programmi in cui una stessa label venga usata più di una volta, ma non è certo un caso frequente; comunque è un fatto da tenere presente.

**Costruttore:**  
Texas Instruments Incorporated  
P.O. Box 1443, M/S 6404  
Houston  
Texas 77001  
U.S.A.

**Distributore per l'Italia:**  
Texas Instruments Semiconduttori Italia  
V.le delle Scienze  
02015 Cittaducale (Rieti)  
Cas. Post. 1

Prezzo: L. 29.000

Naturalmente, dato l'esteso impiego di sottoprogrammi per simulare il funzionamento della catasta, i programmi tradotti possono essere eseguiti solo con il modulo RPN installato sulla macchina.

Il simulatore RPN fa uso di 29 registri di memoria (26 per ottenere la stessa capacità della HP 67 ed il resto per usi interni) per cui la ripartizione da usare (sulla TI 59) è la 3 OP 17, che lascia 720 passi liberi per il programma tradotto.

Può quindi succedere che, nonostante le nostre buone intenzioni, un programma al limite dei 224 passi della HP 67 e con molte istruzioni complesse non possa essere poi girato per mancanza di spazio.

Il pericolo è ancora maggiore usando il modello TI 58 che dispone di metà memoria della "sorella maggiore" TI 59, ed in cui rimangono a disposizione solo 240 passi, certamente insufficienti per tradurre un programma RPN appena complesso.

### Descrizione

La dotazione dell'RPN simulator non comprende, contrariamente al caso degli altri moduli, il portasciede nel quale trovano posto anche il modulino stesso ed il Quick Reference Manual: è strano anche perché, in linea di massima, si tratta di un modulino che non viene utilizzato troppo spesso e sarebbe utile avere un posto in cui riporlo. Il portasciede deve essere acquistato a parte, come accessorio.

Punto di partenza per l'uso del modulo RPN è la disponibilità del programma da tradurre in FORMA CODIFICATA NUMERICA, la stessa che si ha sul display della HP 67 (la codifica per l'HP 97 è differente).

Molto opportunamente il manuale (purtroppo in lingua inglese) fornisce una tavola di comparazione tra istruzioni RPN in forma mnemonica e codificata HP 67; l'ultima colonna di tale tabella riporta inoltre le istruzioni tradotte. Le istruzioni stesse sono elencate "per soggetto" il che se da un lato è molto logico dall'altro non sempre è di agevole consultazione.

Sono possibili due modi di funzionamento: con il primo, dopo l'introduzione di ogni istruzione codificata va premuto il tasto A e la macchina fornisce in uscita i seguenti dati:

- conferma dell'istruzione RPN immessa
- numero del passo del programma originario
- numero del passo (o dei passi) del programma TI
- codice istruzioni TI
- istruzioni TI in forma mnemonica

### Listing per HP-67

1	-	32	25	11	-	g	LBL	a
2	-		33	11	-	STO	A	
3	-		35	22	-	h	RTN	
4	-	32	25	12	-	g	LBL	b
5	-			43	-	EEX		
6	-			02	-	2		
7	-			81	-	:		
8	-		33	12	-	STO	B	
9	-			43	-	EEX		
10	-			02	-	2		
11	-			71	-	X		
12	-		35	22	-	h	RTN	
13	-	32	25	13	-	g	LBL	c
14	-			33	13	-	STO	C
15	-			35	22	-	h	RTN
16	-	32	25	14	-	g	LBL	d
17	-			33	14	-	STO	D
18	-			35	22	-	h	RTN
19	-	31	25	11	-	f	LBL	A
20	-			34	14	-	RCL	D
21	-			34	12	-	RCL	B
22	-			01	-	1		
23	-			61	-	+		
24	-			34	13	-	RCL	C
25	-			35	63	-	h	y <sup>x</sup>
26	-			81	-	:		
27	-	22	31	11	-	GTO	f	a
28	-	31	25	12	-	f	LBL	B
29	-			34	14	-	RCL	D
30	-			34	11	-	RCL	A
31	-			81	-	:		
32	-			34	13	-	RCL	C
33	-			35	62	-	h	1/x
34	-			35	63	-	h	y <sup>x</sup>
35	-			01	-	1		
36	-			51	-	-		
37	-			43	-	EEX		
38	-			02	-	2		
39	-			71	-	X		
40	-	22	31	12	-	GTO	f	b
41	-	31	25	13	-	f	LBL	C
42	-			34	14	-	RCL	D
43	-			34	11	-	RCL	A
44	-			81	-	:		
45	-			31	53	f	LOG	
46	-			34	12	RCL	B	
47	-			01	1			
48	-			61	+			
49	-			31	53	f	LOG	
50	-			81	:			
51	-	22	31	13	GTO	f	c	
52	-	31	25	14	f	LBL	D	
53	-			34	11	RCL	A	
54	-			34	12	RCL	B	
55	-			01	1			
56	-			61	+			
57	-			34	13	RCL	C	
58	-			35	63	h	y <sup>x</sup>	
59	-			71	X			
60	-	22	31	14	GTO	f	d	

Foto 1 - Il modulino Solid State Software si inserisce nell'apposito alloggiamento sul fondo della calcolatrice.

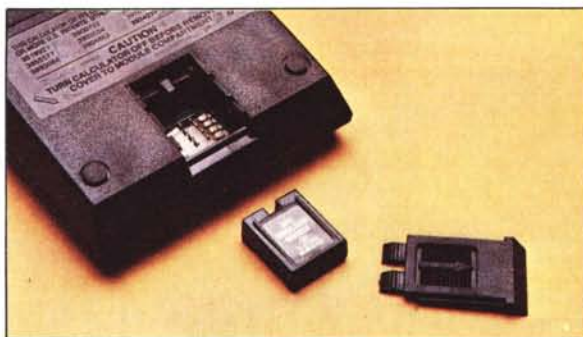


Foto 2 - L'interno del modulino è costituito da una "semplice" ROM.

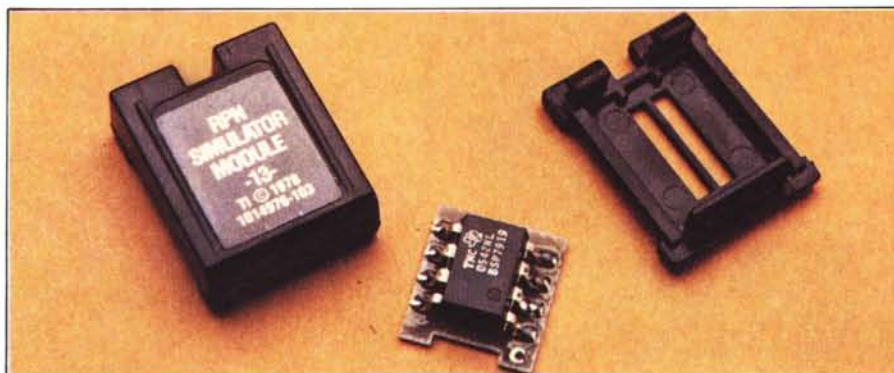


Foto 3 - Con l'RPN simulator è quasi indispensabile la stampante PC-100C; l'output è costituito dall'eventuale "check input" e dalla traduzione del programma da RPN a SOA.

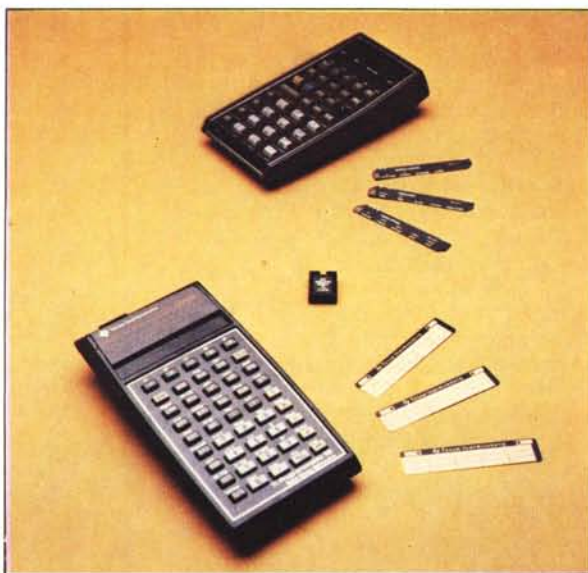


Foto 4 - Al centro della foto, il modulo RPN simulator, in primo piano la TI-59 e sullo sfondo l'HP-67, ciascuna con le sue schede.

questa ultima colonna della striscia di uscita del PC 100 rappresenta la sospirata traduzione del programma originario.

Per poter girare il risultato delle nostre fatiche è ovviamente necessario immettere tali istruzioni nella TI 58/59 una volta ultimata la traduzione (ed eventualmente registrarle su scheda per un uso futuro).

Il secondo modo di funzionamento (che dà origine alle stesse uscite del primo) permette di inserire i codici HP da tradurre di seguito, separati dal tasto R/S (fino ad un massimo di 81 istruzioni per il primo blocco e leggermente meno per i successivi) effettuando la traduzione una volta esaurito lo spazio a disposizione.

Con il tasto C è possibile avere il listato dei codici immessi per effettuare un controllo prima della traduzione. Siccome i codici HP sono memorizzati in registri è possibile modificare abbastanza agevolmente eventuali errori riscontrati (il discorso è diverso nel caso sia stata "saltata" qualche istruzione).

Questo secondo metodo ci sembra preferibile; tra l'altro l'utente può ingannare il tempo in qualche maniera mentre la macchina traduce le istruzioni accumulate (la traduzione di un blocco di 80 istruzioni richiede circa 30 minuti).

Con il tasto D è inoltre possibile ottenere le istruzioni (opzionali) necessarie per far sì che il programma tradotto dopo eseguito l'ultimo passo salti automaticamente al primo (come succede sulla HP 67); vengono inoltre elencate le subroutine necessarie per simulare l'ENTER, il ROLL UP e ROLL DOWN.

Il manuale riporta anche un esempio di traduzione di programma RPN, (un generatore di numeri casuali); nonostante il funzionamento sia corretto, i dati forniti in uscita dalla TI 58/59 sono diversi da quelli originali, perché le calcolatrici Texas hanno un numero di cifre significative interne diverso da quelle delle HP 67/97. Il difetto è dovuto alle particolari modalità di funzionamento dell'esempio scelto ma, nella grande maggioranza dei casi, i programmi tradotti danno (come è logico) gli stessi risultati di quelli originali.

## Funzionamento

Per verificare il funzionamento abbiamo realizzato un breve programma per i calcoli di interesse composto. Il programma consente di legare le quattro grandezze capitale, tasso di interesse, numero dei periodi, montante (capitale + interessi): note tre di esse, si può calcolare la quarta. Per la memorizzazione sono usate, nella 67, le label "a", "b", "c" e "d"; per il calcolo le corrispondenti "A", "B", "C" e "D". Il programma, composto di 60 passi, è stato tradotto dall'RPN simulator in poco più di venti minuti; l'output è costituito da 159 passi. Come era da prevedere, non solo il numero di linee è aumentato ma anche il tempo di esecuzione a causa, soprattutto, del notevole numero di subroutine. In ogni caso è stata ottenuta una perfetta "compatibilità", nel senso che l'uso del program-

CHECK INPUT	312511.	37 223112.	58
	3414.	38 312513.	59
	3412.	39 3414.	60
322511.	19	40 3411.	61
3311.	20	41 81.	62
3522.	21	42 3153.	63
312512.	22	43 3412.	64
43.	23	44 1.	65
2.	24	45 61.	66
81.	25	46 3153.	67
3312.	26	47 81.	68
43.	27	48 223113.	69
2.	28	49 312514.	70
71.	29	50 3411.	71
3522.	30	51 3412.	72
322513.	31	52 1.	73
3313.	32	53 61.	74
3522.	33	54 3413.	75
322514.	34	55 3563.	76
3314.	35	56 71.	77
3522.	36	57 223114.	78

	3563.	037	091	36	PGM	312513.					
			092	51	51						
034	081	36	PGM	093	11	R 041	102	76	LBL		
	082	13	13	094	01	1	CO	103	13	C	
	083	11	A	095	52	EE					
	1.			2.						3414.	
035	084	36	PGM	038	096	02	2	042	104	36	PGM
	085	11	51						105	47	47
	086	11	R						106	11	H
	087	01	1								
											3411.
				039	097	36	PGM				
					098	12	12	043	107	36	PGM
					099	13	C		108	44	44
036	088	36	PGM						109	11	H
	089	12	12								
	090	12	B								
				040	100	61	GTD				
					101	17	B <sup>1</sup>				
	43.										

## Programma tradotto

```

000 76 LBL 042 11 A 084 36 PGM 126 42 STD
001 16 A1 043 36 PGM 085 51 51 127 29 29
002 42 STD 044 45 45 086 11 A 128 28 LOG
003 20 20 045 11 A 087 01 1 129 36 PGM
004 92 RTN 046 36 PGM 088 36 PGM 130 12 12
005 76 LBL 047 51 51 089 12 12 131 14 D
006 17 B1 048 11 A 090 12 B 132 61 GTD
007 36 PGM 049 01 1 091 36 PGM 133 18 C1
008 51 51 050 36 PGM 092 51 51 134 76 LBL
009 11 A 051 12 12 093 11 A 135 14 D
010 01 1 052 11 A 094 01 1 136 36 PGM
011 52 EE 053 36 PGM 095 52 EE 137 44 44
012 02 2 054 46 46 096 02 2 138 11 A
013 36 PGM 055 11 A 097 36 PGM 139 36 PGM
014 12 12 056 36 PGM 098 12 12 140 45 45
015 14 D 057 13 13 099 13 C 141 11 A
016 42 STD 058 11 A 100 61 GTD 142 36 PGM
017 21 21 059 36 PGM 101 17 B1 143 51 51
018 36 PGM 060 12 12 102 76 LBL 144 11 A
019 51 51 061 14 D 103 13 C 145 01 1
020 11 A 062 61 GTD 104 36 PGM 146 36 PGM
021 01 1 063 16 A1 105 47 47 147 12 12
022 52 EE 064 76 LBL 106 11 A 148 11 A
023 02 2 065 12 B 107 36 PGM 149 36 PGM
024 36 PGM 066 36 PGM 108 44 44 150 46 46
025 12 12 067 47 47 109 11 A 151 11 A
026 13 C 068 11 A 110 36 PGM 152 36 PGM
027 92 RTN 069 36 PGM 111 12 12 153 13 13
028 76 LBL 070 44 44 112 14 D 154 11 A
029 18 C1 071 11 A 113 42 STD 155 36 PGM
030 42 STD 072 36 PGM 114 29 29 156 12 12
031 22 22 073 12 12 115 28 LOG 157 13 C
032 92 RTN 074 14 D 116 36 PGM 158 61 GTD
033 76 LBL 075 36 PGM 117 45 45 159 19 D1
034 19 D1 076 46 46 118 11 A 160 00 0
035 42 STD 077 11 A 119 36 PGM 161 00 0
036 23 23 078 42 STD 120 51 51 162 00 0
037 92 RTN 079 29 29 121 11 A
038 76 LBL 080 35 1/X 122 01 1
039 11 A 081 36 PGM 123 36 PGM
040 36 PGM 082 13 13 124 12 12
041 47 47 083 11 A 125 11 A

```

ma avviene esattamente nella medesima maniera con le due macchine (e, ovviamente, si ottengono gli stessi risultati).

**Conclusioni**

È difficile, sotto certi aspetti, esprimere un giudizio sul modulo RPN simulator. Dal punto di vista del funzionamento, senza dubbio, non c'è alcun problema: la traduzione viene eseguita correttamente ed il programma tradotto può essere inserito senza problemi in una calcolatrice Texas Instruments. Le limitazioni sono: l'aumento del numero dei passi, l'intraducibilità di un paio di istruzioni, l'aumento di durata dell'esecuzione. Non si poteva pretendere, d'altra parte, che il numero delle istruzioni non aumentasse, non perché il sistema SOA sia inferiore all'RPN ma perché non può non essere sconveniente la traduzione "pedissequa" da un linguaggio ad un altro con caratteristiche così diverse, in particolare riguardo alla gestione dell'aritmetica. L'intraducibilità di alcune istruzioni, abbiamo detto, è un problema che si può eliminare abbastanza facilmente, mentre l'aumento della durata di esecuzione deriva per forza di cose dal maggior numero sia di linee di programma, sia di subroutine che devono essere eseguite.

Il pregio dell'RPN simulator, però, è soprattutto nel fatto che consente di tradurre, senza grossi sforzi, programmi complessi nei quali, se non documentati, sarebbe laborioso intervenire. Può succedere, certo, che il programma tradotto sia troppo lungo e che riesca a saturare la capacità di memoria della TI; è conveniente, a nostro avviso, cercare di non utilizzare tutto il programma tradotto, ma solo la parte concettualmente più complessa, riscrivendo direttamente in SOA le parti più semplici che a volte, nella traduzione, possono occupare molto spazio: la traduzione di una divisione per 100, ad esempio ha comportato un aumento di numerosi passi nel programma presentato in queste pagine.

In sostanza, il problema fondamentale è di scegliere il modo più conveniente per utilizzare un oggetto le cui prestazioni sono, obiettivamente, notevoli ma che, come tutte le cose, non può ... fare miracoli. 