

Manuale di utilizzo della TI-57

by **Claudio Larini**
(www.claudiolarini.altervista.org)

Indice dei contenuti

1. Configurazione e funzionamento
 1. Nomenclatura e funzioni
 2. Alimentazione
 3. Incominciamo i calcoli
2. Calcoli manuali
 1. Calcoli di base
 2. Funzioni speciali
 3. Calcoli di funzioni
3. Programmazione della TI-57
 1. Che cos'è un programma?
 2. Struttura generale di un programma
 3. Tipi di dati
 4. L'editing dei programmi della TI-57
 5. Comandi di programmazione
 1. Comandi di salto
 2. Sottoprogrammi
 3. Comandi di test
 4. Operatori logici AND e OR
 5. Comandi di anello
 6. Altri comandi

Appendici

- Appendice 1: Caratteristiche tecniche
- Appendice 2: Tabella delle istruzioni LMS
- Appendice 3: Domini di definizione delle funzioni
- Appendice 4: Programmazione sintetica

Nota

Il presente manuale NON è il manuale ufficiale della TI57, bensì un manuale INDIPENDENTE per l'utilizzo per tale calcolatrice.

Convenzioni e notazioni

Significato dei simboli usati :

- * [xxx] rappresenta il tasto "xxx". Per esempio il tasto "=" sarà denotato come [=].
- * Il simbolo → indica un risultato, una conseguenza dell'impostazione sul visualizzatore.
- * Il carattere x rappresenta il registro del visualizzatore della TI-57, cioè vale a dire l'ultimo risultato calcolato, o il numero impostato da tastiera.
- * / corrisponde a un numero di etichetta (label)
- * m corrisponde a un numero di memoria
- * pp corrisponde a un numero di passo

1 - Configurazione e funzionamento

1.1 - Nomenclatura e funzioni



La TI 57

Visualizzatore

Il visualizzatore permette di impostare 11 caratteri. In modalità di calcolo, il visualizzatore può impostare fino a 8 cifre, il punto decimale, un segno e 2 cifre di esponente.

Interruttore di alimentazione

L'accensione si ottiene spostando verso destra l'interruttore di alimentazione.

Tastiera

La tastiera è composta da 40 tasti organizzati su una matrice di 8 righe per 5 colonne. La colonna di sinistra è composta dai tasti di programmazione: qui sono raggruppate la maggior parte delle funzioni inerenti alla programmazione. Le 4 colonne rimanenti corrispondono a tasti che si trovano su una calcolatrice scientifica.

Sulla maggior parte dei tasti figura una serigrafia che agisce da funzione secondaria di quel tasto: sulla TI-57 la maggior parte dei tasti consente di accedere alla funzione principale che ha il nome scritto sul tasto stesso e ad una funzione secondaria che corrisponde al nome serigrafato.

Tasti di utilizzo speciale

- * [INV] tasto di inversione, permette di invertire il significato di certe funzioni.
- * [2nd] tasto che permette di accedere alle funzioni secondarie serigrafate sopra i tasti.
- * [=] calcola e richiama la formula in corso
- * [CE] sopprime l'ultimo operando di una formula (Esempio: 1 [+] 2 [CE] 3 [=] fornisce 4).
- * [CLR] sopprime la totalità della formula (esempio: 1 [+] 2 [CLR] 3 [=] fornisce 3 ; [CLR] sopprime 1 [+] 2).
- * [LRN] permette di entrare ed uscire dal modo programmazione.

Nota : I tasti [2nd] e [INV] sono detti "tasti morti" per il fatto che non svolgono nessuna azione di per se stessi: sono fatti per modificare l'azione degli altri tasti.

Esempio : il tasto [x^2]

- * premere 0 [.] 5 [x^2]: ottenete 0.25 (che è il quadrato di 0.5)
- * premere 0 [.] 5 [2nd] [x^2]: ottenete 0.0087265 (che è il seno di un angolo di 0.5°)
- * premere 0 [.] 5 [2nd] [INV] [x^2]: ottenete 30 (seno di un angolo di $30^\circ = 0.5$, cioè viene calcolata la funzione arcoseno)
- * premere 0 [.] 5 [INV] [x^2]: ottenete $0.25=0.5^2$ (non tutte le funzioni sono "invertibili").

Utilizzando la convenzione del manuale ufficiale, una combinazione del tipo tasto "2nd" + tasto " x^2 " sarà scritta come [2nd][sin]. [sin] rappresenta la funzione chiamata e [2nd] indica che è utilizzata la funzione secondaria del tasto [x^2].

Tasti numerici - Virgola decimale - Tasto di esponente

Per impostare un valore numerico, impostare le cifre che lo compongono secondo l'ordine.

Premere il tasto [.] per impostare la virgola decimale nella posizione desiderata.

Per impostare un numero in notazione esponenziale come, ad esempio, 1.23×10^{-6} premere 1 [.] 23 [EE] [+/-] 6.

Tasti di calcolo

[+][-][x][÷] tasti di operazioni aritmetiche.

Per effettuare una addizione, sottrazione, moltiplicazione o divisione, impostare il calcolo nell'ordine in cui compare nell'espressione.

Tasti di funzione

Si utilizzano per effettuare un calcolo di funzione. Differenti funzioni sono ottenibili in combinazione con i tasti [2nd] e [INV]. Confronta il paragrafo 2.3 per una descrizione delle funzioni disponibili.

Modi di funzionamento

La TI-57 dispone di 2 modi di funzionamento:

- * Modo calcolo
- * Modo programmazione

Il passaggio da una modalità all'altra si ottiene con il tasto "LRN".
La modalità di "default" (all'accensione) è il modo calcolo.

1.2 - Alimentazione

La TI-57 è alimentata con una batteria ricaricabile. Quando la batteria si scarica, è necessario collegare l'adattatore per ricaricare la batteria e continuare a utilizzare la calcolatrice. E' possibile utilizzare anche una comune batteria a 9 V.

1.3 - Incominciamo i calcoli

Ordine di priorità dei calcoli

La TI-57 utilizza la logica algebrica (SOA - Sistema Operativo Algebrico) per calcolare le differenti parti d'una formula secondo l'ordine sotto riportato:

1. Tutte le funzioni: x^2 , $1/x$, y^x , \sqrt{x} , $\ln x$, \log , \sin , \cos , \tan , \sin^{-1} , \cos^{-1} , \tan^{-1} , int , $|x|$
2. \times , \div
3. $+$, $-$

La TI-57 non attende la fine dell'impostazione della formule per cominciare i calcoli. Ogni volta che un operatore o una funzione viene impostata, la calcolatrice tenta di calcolare un risultato secondo le regole di priorità sopra esposte.

Questa particolarità permette alla TI-57 di non essere mai bloccata a causa della lunghezza di una formula. Qualunque sia il numero degli argomenti e degli operandi in una formula, quando viene premuto il tasto [=] finale, la TI-57 ha già calcolato la maggior parte della formula.

La logica algebrica funziona con un catasta che viene riempita con i dati man mano che questi vengono impostati: la lunghezza di questa catasta determina il numero massimo di operazioni che possono essere tenute in sospeso: nel caso della TI-57 questo numero è 4.

Numero di cifre impostabili e di calcolo

Il numero di cifre consentito per l'impostazione è di 8 per la mantissa e di 2 per l'esponente.

I risultati dei calcoli superiori a 99999999 o inferiori a 0.0000001 sono automaticamente convertiti nella forma mantissa + esponente.

Overflow ed errori

Se la capacità di calcolo della TI è superata o sono effettuati ingressi non corretti, il visualizzatore lampeggia, per indicare che si è prodotta una condizione di errore.

E' necessario allora premere i tasti [CE] o [CLR] per interrompere il lampeggiamento.

Le operazioni seguenti provocano errore:

- * un risultato supera $\pm 9.9999999 \times 10^{99}$
- * un tentativo di effettuare dei calcoli di funzione, oltrepassando il loro dominio di definizione (vedi tabella dei domini di definizione).
- * un tentativo di utilizzare più di 9 livelli di parentesi nidificate o di quattro operazioni in sospeso
- * un errore di impostazione (es : 5 + + 3).
- * una divisione per zero

2 - Calcoli manuali

2.1 - Calcoli di base

Operazioni aritmetiche

Le operazioni aritmetiche sono effettuate impostandole da tastiera nel medesimo ordine nel quale sono scritte le formule.

Per i valori negativi, premere [+/-] prima o dopo aver impostato il valore.

La moltiplicazione e la divisione hanno la priorità sull'addizione e la sottrazione.

Calcoli con parentesi

Le parentesi permettono di modificare la priorità degli operatori: le espressioni dentro le parentesi hanno la priorità sulle altre. Le parentesi chiuse situate all'estrema destra della formula possono essere omesse e sostituite con il tasto =.

La TI-57 accetta 9 livelli di parentesi nidificate.

Calcoli in memoria

Il contenuto delle memorie è conservato fino allo spegnimento.

I tasti che si riferiscono alle memorie sono:

- * [2nd][C.t] : cancella la memoria 7
- * [INV][2nd][C.t] : cancella tutte le memorie
- * x [STO] m : memorizza x nella la memoria m
- * [RCL] m : richiama la memoria m
- * x [EXC] m : scambia x con la memoria m
- * [x↔t] : scambia x con la memoria 7
- * x [SUM] m : addiziona x alla memoria m
- * x [INV][SUM] m : sottrae x alla memoria m
- * x [2nd][PRD] m : moltiplica la memoria m per x
- * x [INV][2nd][PRD] m : divide la memoria m per x

Nota: A causa della struttura interna della TI-57, il registro t (vedi comandi di test) coincide con il registro 7, mentre le posizioni 3 e 4 della catasta algebrica (che corrispondono rispettivamente ai dati relativi alla 3° e 4° operazione in sospeso) coincidono con i registri 5 e 6.

Specificazione del numero dei decimali

Per specificare il numero di decimali (sul visualizzatore, premere [2nd] [Fix] n, con $0 \leq n \leq 7$). Tutti i valori impostati saranno arrotondati alla n-ima decimale.

Per ritornare all'impostazione standard (numero di decimali liberi), premere [2nd] [Fix] 9 o [INV][Fix].

La specificazione del numero di decimali influisce solo sulla visualizzazione del numero: i decimali sono mascherati, ma non troncati.

2.2 - Funzioni speciali

Funzione di calcolo continuo

Anche se i calcoli sono finiti dopo aver premuto il tasto [=], il risultato ottenuto potrà essere riutilizzato per i calcoli successivi.

2.3 - Calcolo di funzioni

Unità di misura degli angoli

L'unità di misura degli angoli (gradi, radianti, gradi centesimali) è specificata tramite i tasti [2nd] [Deg], [2nd] [Rad] e [2nd] [Grad].

Una volta fissata una unità di misura angolare, questa resta attiva finché non se ne specifica un'altra.

L'ultima unità di misura angolare impostata è efficace fino allo spegnimento.

L'unità di misura angolare all'accensione è gradi.

L'unità angolare corrente può essere verificata grazie alla sequenza 1 [2nd] [INV] [SIN] che fornisce:

- * 90 se l'unità è in gradi
- * 1.5707963 ($\pi/2$) se l'unità è in radianti
- * 100 se l'unità è in gradi centesimali

Un angolo può essere convertito da una unità ad un'altra grazie alle funzioni trigonometriche ed a un cambio di unità angolare.

Esempio: per convertire 30 gradi in radianti si preme [2nd] [Deg] 30 [2nd] [sin] [2nd] [Rad] [2nd] [INV] [sin] e si ottiene 0.5235988 ($\pi/6$).

Notazione sessagesimale

La notazione sessagesimale corrisponde all'espressione di un angolo in gradi/minuti/secondi o di un'ora in ore/minuti/secondi.

La TI-57 non permette di calcolare direttamente in notazione sessagesimale, ma dispone di funzioni di conversione:

- * [2nd][D.MS] : conversione sessagesimale \leftrightarrow decimale
- * [INV][2nd][D.MS] : conversione decimale \leftrightarrow sessagesimale

Esempi:

- * Conversione di 2h40m29s in notazione decimale : 2.4029 [2nd][D.MS] uguale a 2.6747222 ore
- * Conversione di 1.5 ore in notazione sessagesimale : 1.5 [INV][2nd][D.MS] uguale a 1.3 cioè 1h30m00s

Nota: Le funzioni di conversione sessagesimale convertono sempre in gradi, a prescindere dall'unità di misura angolare corrente.

Funzioni trigonometriche e trigonometriche inverse

Va specificata l'unità di misura angolare prima di eseguire dei calcoli con funzioni trigonometriche e trigonometriche inverse.

- * [2nd][π]: imposta il valore di Pi greco ($\pi=3.1415927\dots$)
- * x [2nd][sin]: calcola il seno di x
- * x [INV][2nd][sin] : calcola un angolo il cui seno è x. Ricorda che $\sin(x)=\sin(-x)$
- * x [2nd][cos]: calcola il coseno di x
- * x [INV][2nd][cos]: calcola un angolo il cui coseno è x. Ricorda che $\cos(x)=\cos(-x)$
- * x [2nd][tan]: calcola la tangente di x
- * x [INV][2nd][tan]: calcola un angolo la cui tangente è x

Funzioni logaritmiche ed esponenziali

- * [lnx] : calcola il logaritmo naturale di x
- * [INV][lnx] : calcola l'esponenziale di x (e^x , con $e=2.7182818\dots$)
- * [2nd][log] : calcola il logaritmo decimale di x
- * [INV][2nd][log] : calcola l'antilogaritmo decimale di x (10^x)

Trasformazione di coordinate

- * a [x \leftrightarrow t] b : valori delle coordinate
- * [2nd][P \rightarrow R] : conversione di coordinate polari in coordinate rettangolari
- * [INV][2nd][P \rightarrow R] : conversione di coordinate rettangolari in coordinate polari

Esempio: Conversione delle coordinate $x=2$ $y=1$ in coordinate polari :

- * Premendo 2 [x \leftrightarrow t] 1 [2nd][P \rightarrow R] la TI-57 visualizza l'angolo : 26.565051°
- * Premendo [x \leftrightarrow t] si ottiene la distanza : 2.236068

Nota: gli angoli sono espressi in notazione decimale secondi l'unità di misura angolare corrente.

Funzioni statistiche

La TI-57 dispone delle funzioni statistiche di base: si può operare sia su un solo insieme di dati numerici che su due.

- * [INV][2nd][C.t] : comincia una nuova serie statistica (cancella le memorie)
- * y [x \leftrightarrow t] x [2nd][Σ +]: aggiunge x alla prima serie e y alla seconda serie
- * y [x \leftrightarrow t] x [INV][2nd][Σ +]: toglie il valore x alla prima serie e y alla seconda
- * [2nd][x_m] : media della prima serie
- * [INV][2nd][x_m] : media della seconda serie
- * [2nd][σ^2] : varianza della prima serie
- * [2nd][σ^2][\sqrt{x}] : deviazione standard della prima serie
- * [INV][2nd][σ^2] : varianza della seconda serie
- * [INV][2nd][σ^2][\sqrt{x}] : deviazione standard della seconda serie

Nota: le memorie sono utilizzate dalle funzioni statistiche come:

- * m_0 : N
- * m_1 : Σx

- * $m_2 : \sum x^2$
- * $m_3 : \sum y$
- * $m_4 : \sum y^2$
- * $m_5 : \sum xy$
- * $m_6 : \text{non usato}$
- * $m_7 : \text{usato nell'impostazione dei dati come registro t.}$

Altre funzioni

- * $x [\sqrt{x}]$: radice quadrata di x
- * $x [x^2]$: x al quadrato
- * $x [1/x]$: l'inverso di x (1/X)
- * $y [y^x] x [=]$: y elevato alla x
- * $y [INV][y^x] x [=]$: y elevato alla $1/x =$ radice x-sima di y
(esempio: $8 [INV][y^x] 3 =$ radice cubica di $8 = 2$)
- * $x [|x|]$: valore assoluto di x
- * $x [2nd][Int]$: valore intero di x
- * $x [INV][2nd][Int]$: valore decimale di x

Nota: Tra tutte le funzioni della TI-57, solo le funzioni $[y^x]$ e $[INV][y^x]$ attendono la pressione di [=] per visualizzare il risultato. Tutte le altre funzioni utilizzano il contenuto del registro del visualizzatore come argomento e vengono calcolate immediatamente.

3 - Programmazione della TI-57

3.1 - Che cos'è un programma?

Un programma è una serie di istruzioni destinate a effettuare un calcolo. Quando un programma è immesso in memoria, la macchina può eseguirlo tante volte quanto è necessario: molto utile per tutti i calcoli ripetitivi.

Esempio: Si vuole tracciare su carta una curva di equazione $y=x^2+2x+3$. Per fare ciò, bisogna riempire un tabella per 15 valori di x e y :

x:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
y:															
<i>Tabella dei valori della funzione $f:y=x^2+2x+3$</i>															

Con un calcolo manuale, si deve ripetere 15 volte la sequenza : x [x²] [+] 2 [x] x [+] 3 [=] Questo significa battere 145 tasti : è lungo, pesante e con alte possibilità di errore. E' necessario programmare la formula. Accendete la TI-57 e premete (senza fare attenzione al visualizzatore) :

[LRN] [STO] 0 [x²] [+] 2 [x] [RCL] 0 [+] 3 [=] [R/S] [RST] [LRN] [RST]

Il programma è pronto. Non resta nient'altro da fare che usarlo...

Premere : 0 [R/S] → la TI-57 esegue il programma e visualizza 3.

Premere : 1 [R/S] → la TI-57 visualizza 6

Premere : 2 [R/S] → la TI-57 visualizza 11

Continuate per ogni al valore di x. Quando la tabella viene riempita, avete premuto solo 35 tasti e i valori di y sono stati tutti calcolati senza rischio d'errore.

3.2 - Struttura generale d'un programma

Un programma è una serie di istruzioni.

Una istruzione è:

- * una parte di una formula di calcolo (cifre, operatori, funzioni... cfr. cap. 2);
- * un comando di programmazione.

Esempio: nel programma precedente :

Formule di calcolo:	[STO]	0	[x ²]	[+]	2	[x]	[RCL]	0	[+]	3	[=]		
Comandi di programmazione:												[R/S]	[RST]

Ogni istruzione occupa 1 passo di programma.

La TI-57 può memorizzare un programma di 50 passi, numerati da 00 a 49.

Il programma viene perso quando si spegne la calcolatrice.

3.3 - Tipi di dati

Un programma manipola dei dati: i numeri, nel nostro caso.

Questi dati possono essere:

- * delle costanti (es.: 1, -12.5, ...) il cui valore non cambia nel corso dell'esecuzione del programma,
- * delle variabili, il cui valore cambia nel corso dell'esecuzione del programma, e che sono memorizzate nelle 8 memorie numeriche disponibili.

3.4 - L'editing dei programmi

L'editing dei programmi è il modo di funzionamento della TI-57 che permette la impostazione e la modifica dei programmi.

- * [LRN] : Entra ed esce dal modo di editing
- * [SST] : Richiama il passo successivo
- * [BST] : Richiama il passo precedente
- * [GTO][2nd] pp : In modo di calcolo, posiziona il puntatore sul passo numero pp
- * [RST] : In modo di calcolo, posiziona il puntatore sul passo 00
- * [2nd][Ins] : Inserisce un nuovo passo al posto del passo successivo
- * [2nd][Del] : Cancella il passo corrente

3.5 - Comandi di programmazione

I programmi sono basati su formule di calcolo.

Tuttavia, può essere necessario in un programma parecchie volte e con argomenti differenti, o scegliere formule differenti a seconda della situazione.

Questo è il ruolo dei comandi di programmazione.

Le formule di calcolo si sviluppano in modo lineare e invariabile. I comandi di programmazione consentono salti, test, anelli e sottoprogrammi.

3.5.1 Comandi di salto

Normalmente, il programma si svolge in maniera lineare e invariabile: la TI-57 parte al passo 00, esegue l'istruzione là situata, poi passa alla posizione successiva fino alla fine del programma.

Questo modo di funzionamento è mantenuto da una memoria interna (un registro) della calcolatrice chiamato IP (*Instruction Pointer*). IP indica in permanenza il passo attualmente in corso d'esecuzione. All'accensione, la TI-57 inizializza IP a 0 e lo incrementa di 1 dopo l'esecuzione di ogni istruzione.

I comandi di salto permettono di modificare IP a piacimento, e dunque l'ordine di esecuzione delle istruzioni:

- * [GTO] I : segue la sequenza del programma a partire dall'etichetta I
- * [2nd][Lbl] I : definisce l'etichetta I
- * [RST] : segue la sequenza del programma a partire dal passo 00.

I salti permettono anche di memorizzare più programmi differenti nei 50 passi della TI-57. Ciascuno di essi può essere richiamato separatamente tramite un comando [GTO].

I comandi di salto sono relativamente dispendiosi in termini di passi di programma perché obbligano a definire delle etichette: si possono definire 10 etichette numeriche (da 0 a 9) tramite l'istruzione [Lbl].

Il solo comando di salto che non necessita di etichetta è [RST].

3.5.2 Sottoprogrammi

Certe formule o sequenze d'istruzioni possono essere utilizzate in parecchie parti di un programma. E' possibile ripeterle ogni volta che si vuole in un programma, ma questa soluzione è dispendiosa in termini di passi de programma.

E' più conveniente creare un sottoprogramma che contenga queste istruzioni e che possa essere richiamato dal programma principale: un sottoprogramma comincia con una istruzione [2nd][Lbl] e termina con [INV][SBR].

Si possono richiamare 2 livelli nidificati di sottoprogrammi.

- * [SBR] I : chiamata al sottoprogramma situato all'etichetta I
- * [2nd][Lbl] I : inizio del sottoprogramma (etichetta)
- * [INV][SBR] : fine del sottoprogramma, ritorno automatico al programma chiamante.

3.5.3 Comandi di test

Può essere necessario scegliere tra diverse soluzioni per far proseguire l'esecuzione del programma.

I comandi di test verificano se una condizione è vera o falsa:

- se la condizione è vera, la TI-57 esegue l'Istruzione situata al passo immediatamente successivo al comando di test;
- se la condizione è falsa, la TI-57 salta il passo successivo e continua l'esecuzione del programma.

I comandi di test confrontano tutti i valori con il contenuto della memoria 7 (detta registro "t" come test).

- * [2nd][x=t] : testa se x è uguale a t
- * [INV][2nd][x=t] : testa se x è differente da t
- * [2nd][x>t] : testa se x è maggiore o uguale a t
- * [INV][2nd][x>t] : testa se x è strettamente inferiore a t

Esempio. Si vuole calcolare la funzione $f : y = \sqrt{x}$ se $x \geq 0$ e $y = 0$ se $x < 0$.

Programma per calcolare f(x)

Passo	Codice	Istruzione	Note
00	19	[2nd][C.t]	Inizializza t a 0
01	-76	[INV][2nd][x>t]	Test se x<t
02	51 0	[GTO] 0	Se si, continua l'esecuzione dall'etichetta 0
03	24	[√x]	Se no, calcola la radice quadrata
04	85	[=]	Richiama il risultato
05	81	[R/S]	Attendi un nuovo valore di x
06	71	[RST]	Ricomincia al passo 0
07	86 0	[2nd][Lb1] 0	Etichetta 0
08	00	0	Calcola 0
09	85	[=]	Richiama il risultato
10	81	[R/S]	Attende un nuovo valore di x
11	71	[RST]	Ricomincia al passo 0

E un altro programma più corto :

Passo	Codice	Istruzione	Note
00	19	[2nd][C.t]	Inizializza t a 0
01	-76	[INV][2nd][x>t]	Test se x<t
02	00	0	Se si (x<t), rimpiazza x con 0
03	24	[√x]	Calcola la radice quadrata
04	81	[R/S]	Attende un nuovo valore per x
05	71	[RST]	Ricomincia dal passo 0

In questo secondo programma si è ricorso ad una astuzia per evitare i comandi [GTO] e [2nd][Lb1]: quando x è inferiore a 0, si rimpiazza x con 0 e si continue il programma, sapendo che $\sqrt{0}=0$... Si guadagna ancora un passo sopprimendo il comando [=] inutile: la funzione \sqrt{x} richiama tutti i suoi risultati sul visualizzatore.

3.5.4 Come simulare gli operatori logici AND e OR

Il linguaggio della TI-57 non dispone di tali operatori, ma è possibile simulare tali operatori al prezzo di qualche passo di programma e di un piccolo sforzo di riflessione. In effetti, una condizione complessa quale (x>0) e (x≤100) e (x è pari) può essere sempre ricondotta a n condizioni elementari (nel nostro caso: "x>0", "x≤100" e "x è pari") combinata per mezzo di operatori logici.

Operatore OR

Consideriamo la condizione "x≤t" ("x è inferiore o uguale a t"): il linguaggio della TI-57 non dispone di alcun comando per testare questa condizione. Ma possiamo testare separatamente se x=t o se x<t. Se anche una di queste condizioni elementari è VERA, allora la condizione x≤t è VERA pure.

Perciò, è sufficiente mettere le condizioni elementari "in parallelo" nel programma, come si possono connettere lampadine elettriche in parallelo: se una delle lampadine si guasta le altre continuano a brillare. E' sufficiente avere almeno una lampadina in buone condizioni per avere luce.

Esempio di programma :

Passo	Codice	Istruzione	Note
00	66	[2nd][x=t]	Test se x=t
01	51 0	[GTO] 0	Se si, vai all'etichetta 0
02	-76	[INV][2nd][x>t]	Se no (x≠t) testa se x<t
03	51 0	[GTO] 0	Se sì (x<t), allora vai all'etichetta 0
04	Se no (x>t) continua il programma

Un'altra sintassi permette di risparmiare 1 passo :

Passo	Codice	Istruzione	Notes
00	-66	[INV][2nd][x=t]	Testa se $x \neq t$
01	-76	[INV][2nd][x>t]	Se sì ($x \neq t$), testa se $x < t$
02	51 0	[GTO] 0	Se $x = t$ o $x < t$, allora vai all'etichetta 0
03	Se no ($x > t$) continua il programma

La prima sintassi è più leggibile, e sarà facilmente modificabile se si devono usare test di 3 o 4 condizioni elementari. La seconda sintassi è più corta (1 passo di meno) ma meno leggibile e funziona solo per 2 condizioni elementari.

Operatore AND

Consideriamo la condizione " $x > t$ " ("x è strettamente superiore a t"): il linguaggio della TI-57 non dispone di nuovo di nessun comando per testare questa condizione. Ma possiamo testare se $x \neq t$ e se $x \geq t$. Se queste 2 condizioni elementari sono VERE entrambe, allora la condizione $x > t$ è pure VERA.

Questa volta le condizioni elementari vanno messe in serie, come se collegassimo delle lampadine in serie in un circuito elettrico: se una lampadina si guasta, tutta la catena si spegne.

Esempio di programma :

Passo	Codice	Istruzione	Note
00	-66	[INV][2nd][x=t]	Test se $x \neq t$
01	51 0	[GTO] 0	Se sì, vai all'etichetta 0
02	51 2	[GTO] 2	Se no ($x = t$) vai all'etichetta 2
03	86 0	[2nd][Lbl] 0	Etichetta 0
04	-76	[INV][2nd][x>t]	Uno ha $x \neq t$. Testa se l'altro è anche $x \geq t$
05	51 1	[GTO] 1	Se sì ($x > t$), vai all'etichetta 1
06	86 2	[2nd][Lbl] 2	Se no ($x \leq t$) continua il programma
07

Questa sintassi è abbastanza dispendiosa in termini di passi di programma perché obbliga a definire a definire una etichetta per ogni condizione elementare da testare. E' molto meglio ricondurre il tutto a condizioni meno gravose. Per esempio, osservando che se $x > t$, allora $t < x$, si può scrivere allora il programma seguente (molto meno dispendioso ma che modifica il registro t) :

Esempio di programma :

Passo	Codice	Istruzione	Note
00	22	[x↔t]	Scambia i registri x e t
01	-76	[INV][2nd][x>t]	Test se $x > t$
02	51 0	[GTO] 0	Se sì ($x > t$), salta all'etichetta 0
03	Se no, continua il programma

3.5.5 Comandi di ciclo^(*)

I comandi di test e i comandi di salto permettono già di creare dei cicli di programmazione. Un ciclo (*loop*) è l'esecuzione di una parte di codice un certo numero di volte finché viene raggiunta una condizione di arresto.

^(*) Nei manuali in lingua italiana della *Texas Instruments* si usa il termine "anello".

Si può, per esempio, eseguire un ciclo per calcolare la funzione fattoriale. La funzione fattoriale di n (nota come $n!$) è la funzione che a un intero n associa il valore $1 \times 2 \times 3 \times 4 \times \dots \times (n-1) \times n$.

Esempio di programma per il calcolo di $n!$:

Passo	Codice	Istruzione	Note
00	32 7	[STO] 7	Memorizza n
01	01	1	1
02	32 0	[STO] 0	Memorizza 1 in m_0
03	32 1	[STO] 1	Memorizza 1 in m_1
04	86 0	[2nd][Lb1] 0	Etichetta 0
05	33 0	[RCL] 0	Richiama m_0
06	39 1	[2nd][Prd] 1	Moltiplica m_1 per m_0
07	01	1	Aggiunge 1...
08	34 0	[SUM] 0	... a m_0
09	33 0	[RCL] 0	Richiama m_0
10	-66	[INV][2nd][x=t]	Test: se m_0 è diverso da n
11	51 0	[GTO] 0	Se si, ritorna all'etichetta 0
12	33 1	[RCL] 1	Se no, richiama m_1
13	81	[R/S]	Attende un nuovo valore di x
14	71	[RST]	Ricomincia dal passo 00

Istruzioni per l'operatore: impostare il valore di n e premere [R/S]. Dopo qualche secondo viene visualizzato $n!$.

In questo programma, si usa la memoria 0 come contatore, che va a prendere successivamente tutti i valori tra 1 e il registro t (memoria 7). Il risultato del calcolo è messo nella memoria 1. I passi da 00 a 03 inizializzano i dati. Il ciclo stesso è posto tra i passi 04 e il passo 08. Il test della condizione di uscita occupa i passi da 09 a 11 e i passi da 12 a 14 visualizzano il risultato e permettono la scelta di un nuovo valore di n : 15 passi in totale.

Il programma funziona ma si può fare di meglio se si utilizzano i comandi di anello specifici della TI-57:

- * [2nd][Dsz] : "Decrementa e Salta se non Zero" - decrementa^(*) la memoria 0, test e si $m_0 \neq 0$, se si, esegui il passo successivo. Se no, salta il passo successivo.
- * [INV][2nd][Dsz] : "Decrementa e Salta se Zero" - decrementa^(*) la memoria 0, testa se $m_0 = 0$, se si, esegui il passo successivo. Se no, salta il passo successivo.

Questi comandi permettono di risparmiare qualche passo come si vede dal programma di calcolo di $n!$ usando [Dsz] :

Passo	Codice	Istruzione	Note
00	33 0	[RCL] 0	Richiama m_0
01	55	[x]	Moltiplica per...
02	-56	[INV][2nd][Dsz]	Decrementa m_0 , testa se 0 è stato raggiunto...
03	81	[R/S]	Se $m_0 = 0$, visualizza $n!$ e attendi un nuovo valore di n
04	71	[RST]	Se no, o se un nuovo valore di n è stato scelto, ritorna al passo 00

Istruzioni per l'operatore: premere [CLR], il valore di n poi [STO] 0 poi [R/S]. Dopo qualche secondo, $n!$ viene visualizzato.

^(*) Decrementa: Sulla TI-57, decrementare significa "avvicinare m_0 al valore 0". Così, se $m_0 > 0$, la TI-57 toglie 1 da m_0 . Se $m_0 < 0$, aggiunge 1 a m_0 .

Questo programma beneficia di parecchi miglioramenti in rapporto alla versione precedente:

- * Si usa una sola memoria, che viene inizializzata dall'utilizzatore (guadagno: 4 passi).
- * Questo permette di collocare il corpo dell'anello al passo 00, evitando l'uso di una etichetta (guadagno 1 passo).
- * Viene calcolato il prodotto $1 \times 2 \times \dots \times n$ non più in una memoria ma "direttamente" al visualizzatore, perciò non c'è bisogno di una istruzione [RCL] per richiamare il risultato: c'è la [x] finale che lo richiama (guadagno: 1 passo).
- * [Dsz] decrementa e testa il contenuto della memoria 0, e non altera il registro del visualizzatore. I comandi 1, [SUM] 0, [RCL] 0 sono dunque inutili (guadagnati 3 passi).
- * L'istruzione [RST] del passo 04 è utilizzata per 2 casi differenti: è eseguita quando $m_0 \neq 0$ per continuare il calcolo del fattoriale di n (cfr. istruzione [GTO] 0 del passo 11 del programma precedente).
- * E' anche eseguita quando $m_0 = 0$ per lanciare il calcolo del fattoriale di un nuovo valore di n (idem passo 14 del programma precedente). Questo ci permette ancora di guadagnare un passo).

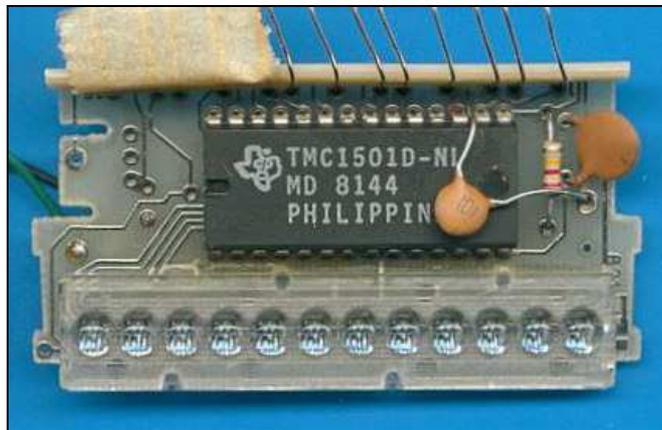
Risultato: 5 passi al posto di 15. I comandi di ciclo della TI-57 sono MOLTO interessanti...

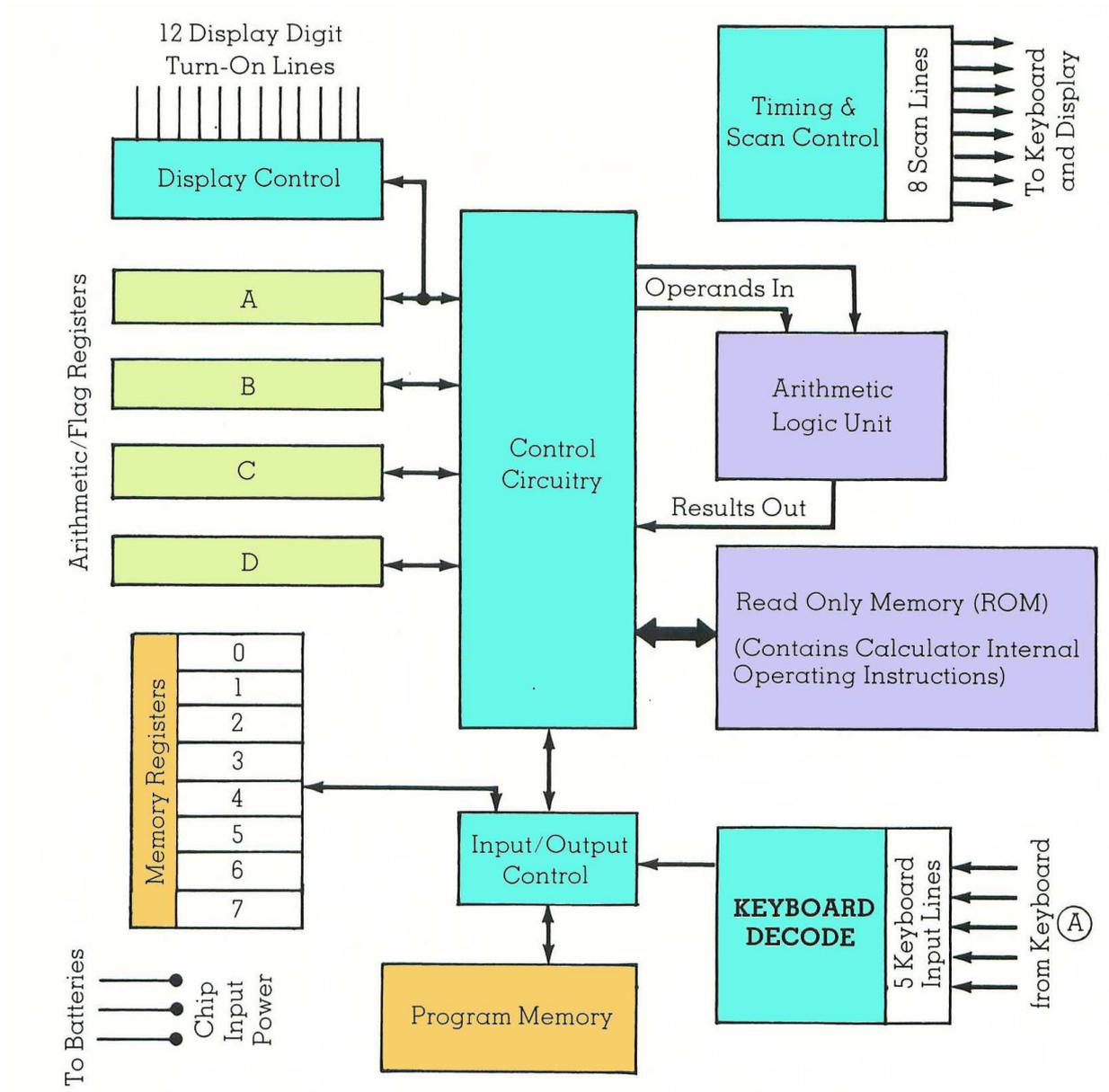
3.5.6 Altri comandi

- * [R/S] : Arresta l'esecuzione del programma e ritorna la modo calcolo. Permette di visualizzare un risultato e/o l'impostazione di un valore. Una nuova pressione sul tasto [R/S] rilancia il programma che è stato arrestato.
- * [Pause] : Interrompe qualche secondo lo svolgimento del programma. Utile per richiamare un risultato.
- * [Nop] : "No Operation" = comando autoesplicativo....

Appendice 1 : Caratteristiche tecniche

Processore:	Texas TMC 1501 (4 bit)
RAM:	4 registri interni A, B, C e D. 16 registri da 16 cifre (X0..X7 e Y0..Y7): sono accessibili da tastiera 50 passi di programma compatti e 8 registri (memorie numeriche). In tutto 64 * 20 =1280 bit
ROM:	81 istruzioni LMS (Linguaggio Macchina Specializzato): in tutto 2Kw (word da 13 bit.)
Visualizzatore:	LED rossi, 8+2 cifre
Porte di Espansione:	Nessuna
Peso:	185 grammi
Dimensioni:	14.8 x 8.1 x 3.6 cm
Alimentazione:	batterie ricaricabili con alimentatore o batteria da 9V





La struttura interna della TI-57 (dal manuale ufficiale)

Appendice 3: Domini di definizione delle funzioni

I domini di definizione delle funzioni preprogrammate della TI-57 sono i seguenti:

arc sin x, arc cos x	con $-1 \leq x \leq 1$
e^x	con $-227.05502 \leq x \leq 230.25850$
10^x	con $-99 < x < 100$
ln x	con $1 \times 10^{-99} \leq x < 1 \times 10^{100}$
log x	con $1 \times 10^{-99} \leq x < 1 \times 10^{100}$
INV P \rightarrow R	con $R \leq 1 \times 10^{50}$

Appendice 4: Programmazione sintetica

Come si vede nell'Appendice 2, non tutti i codici hanno una istruzione corrispondente: la TI-57 ha una specie di "grimaldello" che rende possibile sintetizzare codici normalmente inaccessibili tramite la tastiera e taluni di questi nuovi codici equivalgono a nuove funzioni.

La sequenza **[Exc] [Lbl 0]** impostata come da programma costituisce il nostro "Sintetizzatore di Codici" (lo chiameremo S.C.). Da notare che questa sequenza immessa in modo calcolo produce, giustamente, un errore.

Impostiamo, per esempio, dal passo 00 l'istruzione Exc senza parametro (premendo [SST] dopo [2nd] [Exc]): la TI-57 lo accetta così com'è (mostrando **00 38**). Impostiamo poi come parametro al passo 01 [Lbl 0]. Passiamo poi di nuovo in modo calcolo e premiamo [RST], poi [SST]. Non sarà dato nessun segnale di errore, anzi premendo [LRN], otterremo **02 00 0**. Dopo parecchie prove è stato possibile ottenere una logica di funzionamento dell' S.C.

Ad esempio, vogliamo sintetizzare i nostri codici al passo 02. Dopo aver caricato in memoria la sequenza generatrice:

```
00 38  Exc
01 86 0  Lbl  0
```

passiamo in modo diretto, andiamo al passo 00 con [RST] e premiamo:

- 1) [CLR], [Exc], [Lbl], [0] (avanziamo con [SST]), [LRN], poi una cifra da 0 a 7: otteniamo i codici da 00 a 07 al passo 02. In breve, gli usuali codici dei tasti numerici.
- 2) [CLR], introdurre una (o più di una) cifra, [Exc], [Lbl], [0], [LRN], poi una cifra da 0 a 7: otterrete i codici da 08 a 15 al passo 02.
- 3) [CLR], [Exc] [Lbl] [0], [EE], [LRN], poi una cifra da 0 a 7: otterrete i codici da 13 a 83 al passo 02.
- 4) [CLR], cifra, [Exc] [Lbl] [0], [EE], [LRN], poi una cifra da 0 a 7: otterrete i codici da -13 a -83 al passo 02.
- 5) [CLR], [Exc], [Lbl], [0], [.] o [+/-], [LRN], poi una cifra da 0 a 7: otterrete i codici da 11 a 81 al passo 02.
- 6) [CLR], una cifra, [Exc] Lbl] [0] [.] o [+/-], [LRN], poi una cifra da 0 a 7: otterrete i codici da -11 a -81 al passo 02.
- 7) [CLR], [Exc], [Lbl], [0], cifra, [EE], [LRN], poi una cifra da 0 a 7: otterrete i codici da 12 a 82 al passo 02.

- 8) [CLR], una cifra, [Exc], [Lbl], [0], cifra, [EE], [LRN], poi una cifra da 0 a 7: otterrete i codici da -12 a -82 al passo 02.

Cosa fanno questi nuovi codici?

Primo: usando la seconda sequenza, otteniamo i codici da 10 a 15 che non hanno niente a che fare con i vari Inx, CE, CLR ecc... Eseguendoli, otteniamo i caratteri A, b, C, d, E, F! Questi sono visualizzabili senza problemi, ma qualsiasi tentativo di convalidare tali ingressi li fanno trasformare in "numeri normali". Infatti, abbiamo ottenuto i codici da 0A a 0F (trasformati in numeri da LRN) che sono la logica continuazione, in esadecimale, delle cifre da 0 a 9. In ogni caso possono essere usati per ottenere brevi messaggi (preceduti da [CLR]) ed usati in unione con [Fix] permettono di ottenere fissaggi dei decimali particolari.

Secondo: usando la quinta sequenza, otteniamo i nuovi codici 11, 21, 31 e 41. I codici 21 e 31 eseguiti in modo programma funzionano come [LRN] e [SST]. Il codice 41 (BST) non fa niente ma d'altro canto il codice 11 non ha niente a che vedere con la programmazione del tasto [2nd]! Noi otteniamo qualcosa di molto utile e che ricorda la funzione **HIR** (Hierarchy Internal Register) della TI 59. Programmiamo HIR al passo 02 poi passiamo in modo diretto, introduciamo un numero e premiamo [=]. Eseguiamo la sequenza [INV] [C.t] Il numero sembra irrimediabilmente perso. Andiamo al passo 02 ed eseguiamo HIR premendo poi [=]. Il numero non è andato perso perché non viene memorizzato in un registro, dove va allora a cercarlo l'istruzione HIR? In un registro interno non interessato da [INV] [C.t] (*per la precisione, HIR legge il registro C del microprocessore TMC1501*). Il programma per il calcolo dei decimali del numero "e" usa questa funzione. Si capisce allora l'importanza di questa funzione se il vostro programma deve cancellare le memorie durante la sua esecuzione...

Terzo: utilizzando la settima sequenza, otteniamo il nuovo codice 12, che non ha niente a che fare con INV! E' stato chiamato **DEE** (delta EE), poiché dato un numero con N cifre decimali visualizzate, ne prende il valore assoluto e lo divide per 10^{9-N} . Poiché la funzione DEE lavora come EE, attenzione agli arrotondamenti!
Ad esempio: π "12" da come risultato 0.0314159, infatti $N=7$ da cui la divisione per 10^{9-7} (10^2) mentre 9 INV log "12" da come risultato 1, infatti 9 INV log è 1×10^9 da cui $N=0$ da cui il risultato 1.

Sintetizzando invece i codici -11 (INV HIR) e -12 (INV DEE) nulla cambia niente rispetto ai codici precedenti.

Se infine proviamo a sintetizziamo i codici da:

- 52 a 82, otteniamo le istruzioni Lbl 7, 4, 1 e 0
- 53 a 73, otteniamo le istruzioni Lbl 8, 5 e 2
- 52 a -82, otteniamo le istruzioni Gto 7, 4, 1 e 0
- 53 a -73, otteniamo le istruzioni Gto 8, 5 e 2

Per estensione i codici da 54 a 74 corrispondono a Lbl 9, 6, 3 e i codici da -54 a -74 a Gto 9, 6, 3. La tabella dei codici è quindi ulteriormente arricchita...

I codici 16, 17 si comportano come i corrispondenti 11 e 12, mentre 37 e 47 sono dei Nop.

Mancano da sintetizzare i codici i codici 57, 67, 77, 87, 58, 68, 78, 59, 69, 79 e i loro inversi (che corrispondono alla posizione dei tasti numerici).

Applicazioni pratiche

La memoria costante. C'è una prima applicazione pratica che gli S.C. possono proporre: la TI- 57C, cioè con memoria costante!

Il metodo è semplice: programmate un S.C. da qualche parte e fatelo eseguire. Poi premete [INV] [STO] [3] [+/-] [+/-]: il display si spegne ma la TI-57 è sempre sotto tensione. Si recupera il visualizzatore con [INV] [2nd] [Fix]! Si può verificare che in questo stato la TI-57 consuma meno della TI-58C! Si consiglia di porre la sequenza che segue al termine della memoria:

Lbl 1, Exc, Lbl 0, INV STO (SST), 3, +/-, +/-, R/S, INV Fix.

SBR 1 equivale a spegnere la TI 57C, poi R/S la “riaccende”.

Si può anche verificare che dopo aver impostato la sequenza INV STO, 3 la TI-57 è in modalità Fix -1 (!). Se invece di usare INV STO, 3 si usa da INV STO, 0 a INV STO, 9 si ottiene la TI-57 in modalità da Fix 5 a Fix -4.

Nota: per minimizzare lo spazio occupato dalla routine si può mettere nei passi 48 e 49 solo la sequenza **Exc (SST) Lbl 0**. Per "spegnere" la TI-57 basta fare **GTO 2nd 48 R/S INV STO 3 +/- +/-** e per riaccenderla **INV FIX CLR**.

Il Fix esteso. Quanto detto sopra è un caso particolare: infatti oltre alle normali operazioni di Fix da 0 a 9 esistono altre ben 6 operazioni di Fix. Le istruzioni sono: *Fix A*, *Fix b*, *Fix C*, *Fix d*, *Fix E*, *Fix F*. L'istruzione *Fix F* è ben nota perché serve a fare un Fix -1 ed anche per spegnere il display, o meglio a far comparire un segno “-“. Le altre si comportano, partendo da *Fix A*, come: Fix -6, Fix -5, Fix -4, Fix -3, Fix -2, Fix -1.

La prima, Fix -6 (*Fix A*) si comporta come nel seguente modo: se nel display c'è un numero minore di 99999, viene visualizzato il numero come se ci fosse un *Fix 7* o un *Fix 8*. Se c'è un numero compreso tra 100.000 e 499.999 il display si spegne completamente. Se c'è un numero maggiore di 499.999, c'è l'arrotondamento al milione immediatamente superiore o inferiore del numero. La cosa più interessante è che se vogliamo, possiamo memorizzare il numero 100.000 in una qualsiasi memoria e quando vorremo spegnere il display basterà richiamarla (a patto però che non sia stato modificato il Fix).

Tutti gli altri, fino al *Fix d*, si comportano come quello descritto sopra, solo che il numero per spegnere il display deve essere compreso fra 10.000 e 49.999 per il *Fix b*, fra 1.000 e 4.999 per il *Fix C*, fra 100 e 499 per il *Fix d*. Naturalmente se il numero è inferiore al limite viene lasciato così com'è, mentre se è maggiore è arrotondato a seconda del fissaggio in uso. Un discorso a parte merita il *Fix E*: come condizioni di arrotondamento è come gli altri, nel senso che arrotonda al centinaio superiore o inferiore il numero nel display, e lo lascia intatto se minore di 10. La cosa diversa consiste nello spegnimento del display: mentre negli altri *Fix* si spegneva completamente, ora se il numero è compreso tra 10 e 49 si avranno a destra del display (al posto delle cifre destinate

all'esponente) i caratteri "0-". Le utilizzazioni di queste nuove istruzioni, oltre al già troppo volte citato spegnimento del display, sono molto varie: possiamo usarle per arrotondamenti mediante la sequenza [EE] [INV] [EE] e poi moltiplicarle per il fattore di arrotondamento, oppure spegnere il display quando un risultato superi o eguagli una certa quantità.