

**CENTRO STUDI CALCOLATRICI ELETTRONICHE**

del Consiglio Nazionale delle Ricerche

presso l'Università degli Studi di Pisa

**A. Andronico, A. Baldini, P. L. Casalini, M. Chèccucci, M. Martelli**

**IL SISTEMA DI PROGRAMMAZIONE SIMBOLICA  
CEP CON L'USO DEI NASTRI MAGNETICI**

Nota interna N.° II - 66

Pisa, Febbraio 1968

**CENTRO STUDI CALCOLATRICI ELETTRONICHE**

del Consiglio Nazionale delle Ricerche

presso l'Università degli Studi di Pisa

**A. Andronico, A. Baldini, P. L. Casalini, M. Checcucci, M. Martelli**

**IL SISTEMA DI PROGRAMMAZIONE SIMBOLICA  
CEP CON L'USO DEI NASTRI MAGNETICI**

Nota interna N.° II - 66

Pisa, Febbraio 1968

---

LITO-OFFSET FELICI

— P I S A —

## I N D I C E

Prefazione	Pag.	1
1. Generalita'	"	3
2. Modalita' di scrittura dei programmi	"	6
3. Descrizione operativa del TSBC-NS	"	10
4. Struttura dei programmi simbolici e binari su NC e su NM	"	14
5. Struttura del dump	"	18
6. L'assemblamento dei programmi	"	23
7. Struttura di un assemblamento su NC e su NM	"	35
8. Stampa dello schema della distribuzione in memoria relativa a un assemblamento	"	42
9. Indicazioni di errore ed indicazioni operative segnalate dal TSBC-NS e dall'ASS-NS	"	45
APPENDICE - Modalita' ad uso dell'operatore		
A - Modalita' per il TSBC-NS	"	59
B - Modalita' per l'ASS-NS	"	64
BIBLIOGRAFIA	"	73

## § 1. Generalità

Il nuovo sistema di programmazione simbolica (NSPS) per l'uso dei nastri magnetici sulla CEP comprende:

- a) un programma traduttore (TSBC-NS);
- b) un programma assemblatore (ASS-NS);
- c) un programma per la formazione della Biblioteca (FORBIB) per la cui descrizione si rimanda a [1].

1.1. Il traduttore TSBC-NS accetta programmi scritti direttamente in linguaggio simbolico base CEP, programmi provenienti dalla traduzione di programmi scritti originariamente in linguaggio FORTRAN CEP e programmi provenienti dalla traduzione di programmi scritti originariamente in linguaggio Fortran Ridotto CEP (FORRID). Naturalmente il programmatore può scrivere direttamente dei programmi sia nel linguaggio simbolico oggetto del linguaggio sorgente Fortran CEP, sia nel linguaggio simbolico oggetto del linguaggio sorgente FORRID. La distinzione del tipo di linguaggio simbolico base usato (simbolico base, simbolico oggetto del linguaggio sorgente Fortran, simbolico oggetto del linguaggio sorgente FORRID).

viene fatta attraverso la sigla del programma. Le sigle ammesse sono:

- XSC per le pseudo-istruzioni scritte in linguaggio simbolico base CEP, da associare ai gruppi di tipo S.
- XSFC per le pseudo-istruzioni scritte in linguaggio simbolico base CEP da associare ai gruppi di tipo F.
- XDSC per le pseudo-istruzioni definite localmente scritte in linguaggio simbolico base CEP.
- XDFC per le pseudo-istruzioni definite localmente scritte in linguaggio simbolico base CEP da associarsi a programmi scritti originariamente in linguaggio sorgente Fortran CEP.
- SSC per i sottoprogrammi scritti in linguaggio simbolico base CEP.
- SSFC per i sottoprogrammi scritti in linguaggio simbolico oggetto del linguaggio sorgente Fortran CEP, o provenienti dalla traduzione di sottoprogrammi scritti originariamente in linguaggio Fortran CEP.
- SSRC per i sottoprogrammi scritti in linguaggio simbolico base CEP o provenienti dalla traduzione di sottoprogrammi scritti originariamente in linguaggio FORRID.
- GSC per i programmi guida scritti in linguaggio simbolico base CEP.

GSFC per i programmi guida scritti in linguaggio simbolico oggetto del linguaggio sorgente Fortran CEP, o provenienti dalla traduzione di programmi guida scritti originariamente in linguaggio Fortran CEP.

GSRC per i programmi guida scritti in linguaggio simbolico base CEP o provenienti dalla traduzione di programmi guida scritti originariamente in linguaggio FORRID.

1.2. L'assemblatore ASS-NS puo' assemblare insiemi di sottoprogrammi (dati in forma binaria rilocabile) in numero  $\leq 150$  compreso il programma guida e insiemi di pseudo-istruzioni, sia appartenenti ai gruppi gia' esistenti, sia definite localmente (DPX), che siano dello stesso tipo, cioe' tali che il programma guida, i sottoprogrammi, le pseudo-istruzioni dei gruppi e quelle definite localmente abbiano le sigle compatibili. La compatibilita' e' stabilita o dal programma guida, nel caso di assemblamento libero (senza schema) o dallo schema di assemblamento, nel caso di assemblamento programmato (§6). Le compatibilita' ammesse sono:

1) Se il programma guida ha la sigla GSFC tutti i sottoprogrammi che fanno parte dell'assemblamento devono avere la sigla SSFC e tutte le pseudo-istruzioni definite localmente associate all'assemblamento devono avere la sigla XDFC. A un assemblamento di questo tipo, che diciamo di tipo F, vengono associate, se occorre, le pseudo-istruzioni dei gruppi per il Fortran CEP (o pseudo-istruzioni dei gruppi di tipo F).

2) Se il programma guida ha la sigla GSC i sottoprogrammi che fanno parte dell'assemblamento possono avere le sigle SSC oppure SSRC e tutte le pseudo-istruzioni definite localmente associate all'assemblamento devono avere la sigla XDSC.

3) Se il programma guida ha la sigla GSRC per i sottoprogrammi e le pseudo-istruzioni definite localmente valgono le stesse regole del caso 2).

Negli assemblamenti specificati in 2) e 3), che diciamo rispettivamente di tipo S e di tipo R, vengono associate, se occorre, le pseudo-istruzioni dei gruppi del linguaggio simbolico base CEP (o pseudo-istruzioni dei gruppi di tipo S).

1.3. Nei paragrafi che seguono illustreremo le modalita' di scrittura dei programmi e quelle relative all'assemblamento. Verranno infine date le specifiche d'uso del traduttore e dell'assemblatore e la lista degli errori segnalati dai due programmi con la loro interpretazione.

## § 2. Modalita' di scrittura dei programmi

Le modalita' di scrittura dei programmi rimangono quelle gia' usate per il linguaggio simbolico base CEP [11] nel senso che un programma e' composto da:

a) la testata;

- b) il corpo del programma;
- c) la dichiarazione di fine.

Tuttavia, poiche' il linguaggio simbolico nelle sue varie forme (linguaggio simbolico oggetto del linguaggio sorgente Fortran CEP ecc.) e' stato ampliato rispetto al vecchio linguaggio simbolico base, precisiamo le modifiche apportate [2].

2.1. Una prima modifica riguarda la stesura della testata in cui e' stato introdotto il **codice di programmatore**. Tale codice, che e' formato da una lettera seguita da quattro cifre consecutive, deve essere inserito nella prima riga di testata fra il nome del programmatore e il protocollo. La lettera specifica l'ente di appartenenza del programmatore (per i programmatori del Centro e' stata scelta la lettera C). Le quattro cifre successive sono suddivise in gruppi di due cifre ciascuno. Il gruppo delle prime due cifre specifica il lavoro al quale il programma appartiene e il loro valore e' assegnato a discrezione del programmatore, purché tale valore rimanga inalterato per tutti i programmi componenti un dato lavoro. Il gruppo delle ultime due cifre specifica, associato alla lettera, di quale programmatore si tratta. Il valore di questo ultimo gruppo di cifre, una volta stabilito per ciascun programmatore, non può essere cambiato.

- b) il corpo del programma;
- c) la dichiarazione di fine.

Tuttavia, poiche' il linguaggio simbolico nelle sue varie forme (linguaggio simbolico oggetto del linguaggio sorgente Fortran CEP ecc.) e' stato ampliato rispetto al vecchio linguaggio simbolico base, precisiamo le modifiche apportate [2].

2.1. Una prima modifica riguarda la stesura della testata in cui e' stato introdotto il **codice di programmatore**. Tale codice, che e' formato da una lettera seguita da quattro cifre consecutive, deve essere inserito nella prima riga di testata fra il nome del programmatore e il protocollo. La lettera specifica l'ente di appartenenza del programmatore (per i programmatori del Centro e' stata scelta la lettera C). Le quattro cifre successive sono suddivise in gruppi di due cifre ciascuno. Il gruppo delle prime due cifre specifica il lavoro al quale il programma appartiene e il loro valore e' assegnato a discrezione del programmatore, purché tale valore rimanga inalterato per tutti i programmi componenti un dato lavoro. Il gruppo delle ultime due cifre specifica, associato alla lettera, di quale programmatore si tratta. Il valore di questo ultimo gruppo di cifre, una volta stabilito per ciascun programmatore, non può essere cambiato.

2.2. Un'altra modifica importante riguarda la possibilità di inserire in un programma (oltre le pseudo-istruzioni dei gruppi di tipo S o di tipo F a seconda della sigla cfr. 1.1.) delle pseudo-istruzioni definite localmente. Tale possibilità fa ricorso a un nuovo dichiaratore \*DPX.

Ricordiamo che un blocco di pseudo-istruzioni è costituito da un programma unico con diversi punti di entrata relativa in numero non superiore a otto. La strutturazione a blocchi delle pseudo-istruzioni consente di avere più pseudo-istruzioni condensate in un solo programma, ciascuna delle quali può utilizzare tratti di programma in comune.

Il dichiaratore \*DPX ha due forme di scrittura a seconda che è usato nella scrittura di un blocco di pseudo-istruzioni definite localmente oppure no.

Ricordiamo [2] che la forma generale del dichiaratore \*DPX è:

\*DPX# <lista di pseudo-istruzioni locali>.

Nel caso in cui tale dichiaratore sia usato per compilare blocchi di pseudo-istruzioni, gli elementi della lista devono avere le seguenti forme:

1) <nome di pseudo-istruzione> : <nome di blocco di pseudo-istruzioni> (<entrata relativa>)

per specificare le pseudo-istruzioni appartenenti al blocco che si sta compilando;

2) <nome di pseudo-istruzione> : <nome di blocco di pseudo-istruzione>

per specificare le pseudo-istruzioni, appartenenti a blocchi, usate dal blocco che si sta compilando;

3) <nome di pseudo-istruzione>

per specificare le pseudo-istruzioni singole usate dal blocco che si sta compilando.

Le scritture di cui ai casi 2) e 3) valgono anche per le DPX singole, per i sottoprogrammi e per i programmi guida che usino a loro volta delle pseudo-istruzioni definite localmente.

Si voglia ad esempio scrivere il blocco di pseudo-istruzioni locale di nome B che contenga le pseudo-istruzioni B1 e B2, le quali abbiano le entrate relative rispettivamente a 0 e a 3, e inoltre, tale blocco, usi la pseudo-istruzione singola A e la pseudo-istruzione D1 appartenente al blocco D. Si riporta qui di seguito la scrittura corretta del dichiaratore che comparirà nel blocco di pseudo-istruzioni di nome B:

\*DPX # B1: B(0), B2: B(3), A, D1: D

Nota: Per ragioni di memoria il corpo di un programma non può superare le seguenti lunghezze:

665 celle di memoria per le pseudo-istruzioni e DPX sia singole che in blocchi;

6300 celle di memoria per i sottoprogrammi e i programmi guida.

### § 3. Descrizione operativa del TSBC - NS

Il TSBC-NS accetta i programmi di cui in 1.1.e. li traduce in forma binaria rilocabile o invariante<sup>(1)</sup>. Sia i programmi in forma simbolica che quelli in forma binaria possono essere posti:

- 1) su Nastri di Carta (NC) (a 5 e a 7 piste); <sup>(2)</sup>;
- 2) su Nastri Magnetici (NM).

Il TSBC-NS, note che siano le seguenti informazioni:

- 1) se il programma nella forma simbolica si trova su NC o su NM;
  - 2) se il programma tradotto (cioe' in forma binaria) si richiede su NC e/o su NM;
  - 3) se si desidera ottenere il dump del programma tradotto su NC e/o su NM;
- effettua la traduzione attraverso le fasi che descriviamo qui appresso.

---

(1) La forma binaria rilocabile o invariante e' legata alla scrittura del programma, cioe' al modo in cui sono scritte originariamente le istruzioni che lo compongono.

(2) E' usato: NC a 5 piste per l'uscita del dump e NC a 7 piste sia per l'entrata dei programmi simbolici, sia per l'uscita dei programmi binari.

- a) I<sup>^</sup> lettura - il TSBC-NS legge il programma, analizza la sigla e la trasforma secondo le corrispondenze

sigla originale	sigla trasformata
XSC	XBSC
XSFC	XBSC
XDSC	XBSD
XDFC	XBFD
SSC	SBSC
SSFC	SBFC
SSRC	SBSC
GSC	GBSC
GSFC	GBFC
GSRC	GBSC

costruisce le tabelle relative alle variabili dimensionate, alle etichette interne al programma, agli eventuali parametri e a tutte quelle variabili ricavate dalle dichiarazioni che risultino utili all'assegnazione degli indirizzi delle variabili stesse (tenendo conto, nel caso di programmi provenienti dalla traduzione di programmi scritti in Fortran CEP, delle eventuali relazioni di COMMON ed EQUIVALENCE esistenti). In questa fase si costruisce anche la tabella dei sottoprogrammi chiamati dal programma che si sta traducendo e a ogni istruzione o pseudo-istruzione viene associata una coppia di numeri (costituente la "traccia"), di cui il primo appartiene all'insieme  $\{0,1\}$  e il secondo all'insieme  $\{0,1,2,3,4\}$ . Le coppie costruite in questo mo-

do hanno il seguente significato:

- (0,0) = istruzione con indirizzo assoluto o invariante  
(tale coppia e' anche associata alle costanti)  
[2]
- (0,1) = istruzione con indirizzo riferito a una etichetta
- (0,2) = istruzione con indirizzo riferito a una variabile dimensionata in un dichiaratore \*DZV
- (0,3) = istruzione con indirizzo riferito a una variabile dimensionata in un dichiaratore \*DZT
- (0,4) = istruzione con indirizzo riferito a una variabile dimensionata in un dichiaratore \*DZC.

Le coppie (1,0) , (1,1) , (1,2) , (1,3) , (1,4) , si riferiscono alle pseudo-istruzioni di gruppo o definite localmente. (La coppia (1,0) non e' mai associata a una costante).

In questa fase, inoltre, viene effettuata una prima analisi di correttezza formale del programma fornendo eventualmente segnalazioni di errore (cfr. § 9.).

b) II<sup>^</sup> lettura - Sulla base delle informazioni memorizzate dalla I<sup>^</sup> lettura, il TSBC-NS rilegge il programma e passa alla traduzione vera e propria e alle eventuali uscite richieste. In ogni caso, la richiesta del dump, provoca anche l'uscita delle tabelle costruite dal TSBC-NS (vengono fatte uscire solo quelle contenenti almeno un elemento).

Il verificarsi di segnalazioni di errore in  $I^{\wedge}$  o in  $II^{\wedge}$  lettura provoca l'interruzione delle eventuali uscite richieste nel punto in cui avviene l'errore. In tali casi, le porzioni di uscite su NM vengono eliminate essendo le segnalazioni date sufficienti a individuare gli errori avvenuti. In ogni caso il TBSC-NS prosegue l'esame del programma che si sta traducendo fino alla fine, segnalando eventualmente ulteriori errori. Di tutto il lavoro di traduzione il TSBC-NS fornisce, a uso dell'operatore e del programmatore, una traccia, sulla stampante in linea (ST), così composta:

- 1) indicazione della richiesta fatta al TSBC-NS mediante il quadro di comando (QZ) nella forma: QZ..... seguito dalla configurazione in octal ivi specificata;
- 2) la testata del programma che si sta traducendo, su due righe;
- 3) eventuali segnalazioni di errore.

Nota: Durante la traduzione di un programma, il TSBC-NS, ai nomi simbolici delle istruzioni, associa i corrispondenti codici [9] (con il bit  $s_1 = 1$  solo per le istruzioni controllabili, anche se nel nome non figura il simbolo " $\wedge$ "); ai nomi simbolici delle pseudo-istruzioni dei gruppi (sia di tipo S che F) vengono fatti corrispondere i codici stabiliti dai gruppi stessi, mentre per le DPX viene definito un codice fittizio (valori da 0 a 63), corrispondente al numero d'ordine con il quale si incontra il nome nel dichiaratore \*DPX.

## § 4. Struttura dei programmi simbolici e binari su NC e su NM

### 4.1. Struttura dei programmi scritti in linguaggio simbolico su NC

Testata	Corpo del programma	*FIN # Y v
---------	---------------------	------------

Da notare che questa struttura, se il programma da tradurre e' scritto in linguaggio simbolico base, e' ripetuta per due volte; negli altri casi (linguaggio simbolico oggetto del linguaggio sorgente Fortran CEP e linguaggio simbolico oggetto del linguaggio sorgente FORRID) detta struttura e' invece prodotta una sola volta.

### 4.2. Struttura dei programmi scritti in linguaggio simbolico su NM

1) Un record di lunghezza fissa di 25 parole contenente la testata del programma con la seguente struttura:

I^parola = <sigla> .

II^parola = <nome>

III^parola = . <codice di programmatore>

dalla IV^parola in poi, vengono scritte tutte e due le righe della testata a 6 caratteri per parola; se questi non riempiono tutte le 25 parole disponibili, le rimanenti vengono azzerate.

2) Uno o piu' record di lunghezza variabile (comunque di un numero di parole non superiori a 500) contenenti un numero intero di righe simboliche, scritte a 6 caratteri per parola, costituenti il corpo del programma.

I caratteri letti vengono sempre sistemati a partire dall'estrema sinistra della parola e ogni riga simbolica inizia sempre da una nuova parola (questo vale anche per le due righe della testata).

In questa struttura, una parola contenente "\*FIN # Y", chiude l'ultimo record del corpo del programma. L'ultimo programma e' seguito da un record di fine di file e inoltre ogni programma e' scritto su NM una sola volta.

Nota: Per il caricamento dei programmi simbolici su NM, si veda la descrizione del nastro di servizio TRASF 2 [3].

#### 4.3. Struttura dei programmi binari su NC

Testata	lunghezza spec. varie	specific. varie	controllo longitud.	lunghezza traccia	traccia
---------	-----------------------	-----------------	---------------------	-------------------	---------

controllo longitud.	lunghezza programma	lista nomi sottoprogr. e corpo del programma	controllo longitud.
---------------------	---------------------	--	---------------------

Il controllo longitudinale e' costituito di una parola contenente l'intero ottenuto sommando i caratteri, costituenti le varie parole, presi da sinistra verso destra, che compongono il tratto di NC compreso fra la "lunghezza" e il "controllo longitudinale" stesso.

Nota: Due programmi binari consecutivi sono separati da un tratto di nastro contenente 300 caratteri "ω" corrispondente a  $\approx 75$  cm. di nastro bianco.

#### 4.4. Struttura dei programmi binari su NM

1) Un record contenente la testata del programma con struttura analoga a quella dei programmi scritti in linguaggio simbolico su NM con la sola variante della sigla (cfr. § 3).

2) Un record di lunghezza variabile così composto:

lunghezza spec. varie	specific. varie	lunghezza traccia	traccia
--------------------------	--------------------	----------------------	---------

3) Uno o più record di lunghezza variabile (comunque di un numero di parole non superiore a 700) così composti: la prima parola del primo record, divisa in due parti, contiene nella prima metà il numero di record che compongono il programma, nella seconda metà la lunghezza totale del programma; le successive, la lista dei nomi dei sottoprogrammi usati e il corpo del programma.

Nota: Per le due strutture di cui in 4.3. e 4.4. vale quanto segue:

1) Per la testata la lunghezza è fissa di 25 parole con la variante per NC che ogni carattere diverso da "ω" è preceduto dal suo stato ("φ" o "λ")

2) Per le specificazioni varie, la cui lunghezza è variabile a causa della presenza o meno delle DPX si possono avere le seguenti due strutture:

## per i sottoprogrammi

i	numero celle parametriche $H_0$	numero sottoprogrammi	lunghezza programma	lunghezza DZT	lunghezza DZV
$s_0$	$s_1$	$s_{11}$ $s_{12}$	$s_{20}$ $s_{21}$ $s_{35}$	$s_0$ $s_{17}$ $s_{18}$	$s_{35}$
I parola				II parola	

lunghezza DZC	0 se simbolico base o FORRID; Indirizzo Minimo COMMON se Fortran	Maschera gruppi pseudo-istruzioni	Tabella DPX
$s_0$ $s_{17}$ $s_{18}$ $s_{35}$			
III parola		dalla IV alla X parola	2i parole

## per le pseudo-istruzioni e le DPX

i	numero celle parametriche $H_0$	0	lunghezza programma	lunghezza DZT	lunghezza DZV	lunghezza DZC
$s_0$	$s_1$	$s_{11}$ $s_{12}$ $s_{23}$	$s_{24}$ $s_{35}$	$s_0$ $s_{11}$	$s_{12}$ $s_{23}$	$s_{24}$ $s_{35}$
I parola				II parola		

Maschera gruppi pseudo-istruzioni	Tabella DPX
dalla III alla IX parola	2i parole

La struttura della tabella delle DPX, la cui lunghezza e' 2i ( $0 \leq i \leq 64$ ), e' la seguente: ogni parola di indice  $1 \leq j \leq i$  contiene un nome di DPX posto a partire dalla estrema sinistra e la corrispondente parola di indice  $i+j$ , contiene le seguenti indicazioni:

- se il nome corrispondente si riferisce a una DPX singola, -1 su tutta la parola;
- se il nome corrispondente si riferisce a una DPX che fa parte di un blocco, la parola viene suddivisa nel seguente modo:

- 1 oppure entrata relativa	Nome del blocco
$s_0$	$s_{11}$ $s_{12}$ $s_{35}$

Da notare che l'entrata relativa ( $s_0 \div s_{11}$ ) comparirà soltanto quando il nastro binario è stato ottenuto dalla traduzione di un blocco di DPX (vedi 2.2.).

## § 5. Struttura del dump

### 5.1. Dump su NC a 5 piste

L'uscita su perforatore di nastro a 5 piste (PF5) della traduzione, viene effettuata in modo tale che la stampa su telescrivente (TL) venga impaginata in un formato di 30 righe.

La distinzione di una pagina da un'altra avviene tramite un tratteggio continuo per tutta la larghezza del foglio. Ogni programma esce in modo tale che la sua stampa avvenga a inizio pagina nel modo seguente:

- a) due righe di testata seguite da una interlinea;
- b) i dichiaratori, se esistono, scritti su tutta la riga con eventuale continuazione sulla riga successiva in colonne e i segni di continuazione ("#");

c) il corpo del programma così articolato su tre colonne:

- 1) su cinque cifre octal viene indicata la numerazione progressiva dell'istruzione (indirizzo relativo partente da 0),
- 2) su dodici cifre octal viene indicata la traduzione binaria della frase simbolica che si trova sulla colonna successiva,
- 3) la frase simbolica originaria, con la variante che, se per la traduzione di una frase sono occorse un numero di parole maggiore di uno, allora, a seconda del tipo di frase simbolica, questa o è scritta in corrispondenza alla parola di indirizzo più basso oppure accanto ad ogni parola è riportata la parte di frase corrispondente.

## 5.2. Dump su NM

L'uscita su NM del dump di un programma costituisce un file e ciascun record del file contiene una riga di stampa da far uscire sulla stampatrice IBM/1403 [4]. La stampa dell'intero programma è impaginata in un formato di 58 righe. Ogni programma comincia da inizio pagina e le prime righe portano la testata del programma seguita da 2 interlinee. Sulle righe di stampa che seguono la testata sono poste:

- a) un dichiaratore o parte di questo se una riga non è sufficiente a contenerlo per intero;

b) due gruppi di informazioni del tipo specificato in c) del numero precedente; in questo caso la numerazione progressiva e' fatta per righe.

### 5.3. Uscita delle tabelle

Il TSBC-NS fornisce le seguenti tabelle:

Dicitura	Significato	Struttura di un elemento	Formato per riga n° elementi su	
			NC	NM
TCPGHO	Tabella Celle Parametriche Gruppo HO	Numero d'ordine e nome	5	11
TCPGHI	Tabella Celle Parametriche Gruppo HI	Numero d'ordine e nome	5	11
TNMSTP	Tabella Nomi dei Sottoprogrammi	Numero d'ordine e nome	5	5
MASCHX	Maschera delle pseudo-istruzioni dei gruppi	Dodici cifre	1	7
TNMDPX	Tabella dei Nomi delle DPX	n° d'ordine, nome, entrata relativa, nome del blocco	3	5
TNMDNL	Tabella dei Nomi delle variabili additive (DNL)	n° d'ordine, nome, valore	3	5
TNMDZV	Tabella dei Nomi delle variabili dimensionate nel DZV	n° d'ordine, nome, origine di memoria relativa all'indirizzo zero	3	5
TNMDZT	Tabella dei Nomi delle variabili dimensionate nel DZT	idem	3	5

Dicitura	Significato	Struttura di un elemento	Formato per riga n° elementi su	
			NC	NM
TNMDZC	Tabella dei Nomi delle variabili dimensionate nel DZC	idem	3	5
TNMDRS	Tabella dei Nomi delle etichette o riferimenti simbolici	idem	3	5
TNREQV	Tabella dei Nomi delle relazioni di Equivalence, Common e parametri trasmessi con relazioni di Equivalence o Common.	n° d'ordine, nome, valore della costante di Equivalence, vuoto oppure VP (Variabile di Programma), vuoto opp. n° di ordine dell'eventuale parametro che compare in quel gruppo di Equivalence, indirizzo assoluto riferito al Common opp. indirizzo relativo riferito al parametro.  N.B.: Gli elementi di tale tabella sono raggruppati in blocchi di uno o più elementi secondo le re-	1	1

Dicitura	Significato	Struttura di un elemento	Formato per riga n° elementi su	
			NC	NM
		lazioni Common ed Equivalenze. Sulla stampacia scun blocco e' separato da una interlinea. Alla fine della stampa degli elementi di un blocco viene aggiunta in calce la memoria totale occupata da quel blocco.		

Nota: Poiche' non tutti i caratteri di TL sono presenti sulla stampatrice IBM/1403, sono state convenute delle corrispondenze che vengono effettuate automaticamente dal controllo della stampatrice [12].

Diamo qui di seguito la relativa tabella:

Caratteri di TL	Caratteri della stampatrice IBM/1403
^	' (apice)
:	,
<	(
>	)
?	\$
#	\$