

1

~~14 MARZO 1956~~

C.S.C.E.P.  
Sezione Logico Matematica

31 luglio 1955

NOTA TECNICA N° 1

Caratteristiche fondamentali del progetto di massima logico-strutturale della calcolatrice Elettronica.  
(A. Caracciolo, E. Fabri)

Presentiamo in modo schematico le caratteristiche fondamentali logico-matematiche e strutturali del progetto di Calcolatrice elettronica allo studio.

Considerazioni generali intorno alle scelte fatte con un esame critico delle stesse verranno esposte in una Nota successiva, attualmente in via di complemento.

Nella presente Nota ci limitiamo a considerare gli organi interni della macchina (memoria principale e ausiliaria; organo aritmetico, controllo) escludendo gli organi di comunicazione con l'esterno (entrata, uscita, quadro di comando manuale).

Ciò è giustificato dal fatto che per una macchina di tipo universale la scelta di tali organi e delle loro prestazioni va studiata in base alle caratteristiche proprie della macchina, per cui non sarebbe possibile, in questa fase, darne altro che indicazioni sommarie, di scarso interesse effettivo.

Questa nota è divisa in due parti: nella 1a. il progetto viene presentato dal punto di vista matematico; nella 2a. viene discussa nelle grandi linee la struttura e il funzionamento dei singoli organi.

1° Parte

Dal punto di vista matematico, una calcolatrice è completamente caratterizzata dalla lista delle istruzioni e, naturalmente, da tutte quelle informazioni supplementari che sono necessarie per la loro comprensione. La lista presentata è da ritenersi ancora incompleta e provvisoria; la lista completa dovrebbe contenere 32 istruzioni distinte, vale a dire 304 istruzioni oltre quelle presentate; essa sarà completata in seguito sia includendo le necessarie istruzioni riguardanti gli organi esterni, sia esaminando accuratamente fra le tante possibilità aggiunte quelle più importanti dal punto di vista di un'efficiente programmazione. Anche le istruzioni ora presentate potranno subire qualche lieve modifica. Tuttavia, nel loro insieme esse caratterizzano già adeguatamente la macchina in progetto, che presenta, soprattutto per quanto riguarda il problema della modifica degli indirizzi, soluzioni in parte nuove e originali, che verranno descritte quando parleremo del controllo e della rappresentazione delle istruzioni.

Senza entrare in dettagli sui problemi, alquanto complessi dal punto di vista matematico, connessi con la rappresentazione dei dati numerici, accenniamo le caratteristiche fondamentali.

L'unità base per la rappresentazione dei dati numerici sono gruppi fissi di 40 cifre binarie (parole), corrispondenti a 12 - 13 cifre decimali. La virgola è pensata posta immediatamente dopo la prima cifra, la quale, però pur avendo valore posizionale  $2^0$ , sarà usata per indicare il segno. Più precisamente, i numeri che sono direttamente rappresentabili sono i multipli di  $2^{-39}$  compresi fra -1 e 1, estremo superiore escluso. I numeri  $\geq 0$  hanno come cifra del segno 0. Quelli negativi sono rappresentati mediante il complemento a 2; ed hanno, quindi, come prima cifra 1. Per i dettagli si rimanda, per il momento, alla letteratura sull'argomento. Con tale convenzione sul campo numerico, si evita il pericolo che il prodotto esca dal campo numerico sopraindicato (sebbene la sua rappresentazione completa richieda una precisione di  $2^{-78}$  anziché  $2^{-39}$ ). Resta la possibilità che la somma algebrica sia in modulo maggiore di 1 (traboccamento). Di ciò va tenuto conto in sede di programmazione: tale compito è agevolato da una particolare istruzione di salto condizionato. Con la convenzione adottata per la rappresentazione dei negativi l'operazione di somma algebrica (che include, previa complementazione del sottraendo, anche la sottrazione) viene ricondotta in ogni caso a una semplice addizione modulo 2, ciò che permette una notevole semplificazione della struttura dell'organo aritmetico. La complementazione di un numero si ottiene facilmente invertendo le singole cifre (negazione logica) e sommando un'unità nel posto meno significativo ( $2^{-39}$ ).

Per quanto riguarda la lista delle istruzioni, converrà presentarle contemporaneamente allo schema logico della macchina, accompagnando la descrizione sommaria dei singoli organi con l'indicazione delle istruzioni che ne caratterizzano l'uso e le funzioni. Una discussione sulla rappresentazione delle istruzioni verrà fatta parlando del controllo.

Nelle linee generali lo schema segue quello, ormai classico, di tutte le calcolatrici. Tale schema, (in cui si prescinde, come s'è detto, dagli organi di entrata e uscita e dal quadro di comando manuale) è indicato nella figura 1:

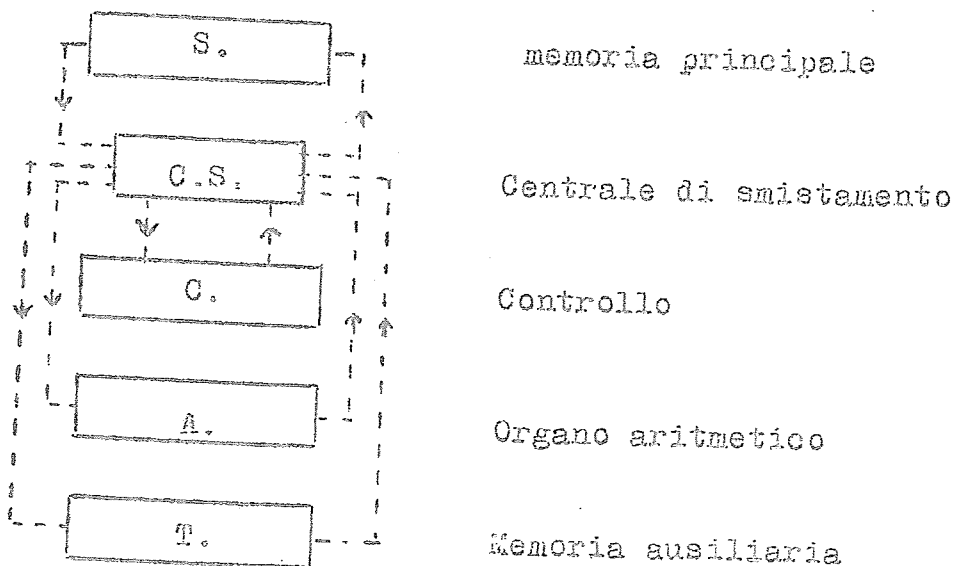


Fig. 1

Esso si compone:

MEMORIA PRINCIPALE (a tempo d'accesso rapido) di 1024 celle di 40 bit disposte a matrice quadrata di 32 righe per 32 colonne.

L'indirizzo di una cella é quindi dato da 10 bit (due caratteri di telescrivente). Per indicare l'indirizzo di una cella generica useremo le due lettere XY, mentre con (XY) indicheremo l'informazione (parola) contenuta nella cella d'indirizzo XY.

La memoria principale rapida ha accesso a e da tutti gli altri organi della macchina. Non ci sono istruzioni che riguardano esclusivamente la memoria principale.

MEMORIA AUSILIARIA - Ha grande capacità e precisamente di 8192 parole. E' suddivisa in 4 fasce di 8 blocchi di 32 sezioni.

Ad essa ci si riferisce sempre per sezioni che sono in totale 1024. L'indirizzo di una sezione é ancora specificato da 10 bit (2 bit per la selezione di fascia, 3 bit per la selezione di blocco e 5 per la selezione di sezione).

L'accesso é possibile dalla memoria principale e avviene per trasferimento, in ambo i sensi, di sezione intere ovvero sia di gruppi di 8 parole.

Dovendosi indicare, per tali trasferimenti, due indirizzi sono necessarie in ogni caso due istruzioni.

- H XY Preparare il tamburo (a leggere o scrivere indifferentemente) alla sezione di indirizzo XY (preselezione).
- H1 XY Leggere successivamente le parole contenute nella sezione preselezionata e trasferirle nelle celle da XY a XY+7 della memoria principale.
- Ha XY Scrivere successivamente nella sezione preselezionata le 8 parole contenute nelle celle da XY a XY+7 della memoria principale.

### UNITA' ARITMETICA

E' composta essenzialmente da due registri per la conservazione degli operandi e dei risultati, ambedue accessibili dalla memoria principale in ambo i sensi e dotati di traslazioni a sinistra e a destra (prodotti per potenze positive e negative di 2). Contiene inoltre elementi opportuni di calcolo e di controllo interno che verranno descritti in seguito. Indicheremo i due registri con Acc. (accumulatore) e Mr. (moltiplicatore). I nomi provengono dal fatto che in Acc. si formano fra lo altro, i risultati delle somme e Mr. contiene, all'inizio dell'istruzione di moltiplicazione, il moltiplicatore. Con (Acc) e (Mr) indicheremo poi il contenuto di tali registri.

Il funzionamento dell'unità aritmetica, dal punto di vista della programmazione, risulta dalla lista delle istruzioni che la riguardano. Si noti, però, che per la natura discreta e finita della rappresentazione dei numeri le operazioni che qui di seguito sono indicate come Sommare, Sottrarre, Moltiplicare etc. non corrispondono sempre esattamente alle operazioni aritmetiche di egual nome. Ciò ha molta importanza

per l'esatta comprensione del funzionamento di queste macchine dal punto di vista matematico. Questo argomento, tuttavia, é complesso e delicato e sarà ripreso in modo esauriente e sistematico in una prossima Nota. Per la definizione delle operazioni sottocindicate ci si può riferire per il momento al noto rapporto di A.W. Burks, H.H. Goldstine, J. von Neumann (Princeton, 1947).

La lista delle istruzioni aritmetiche presentata é da ritenersi ancora in parte provvisoria: essa é, infatti, logicamente e matematicamente completa, cioè sufficiente per la programmazione di qualsiasi tipo di calcoli; é possibile tuttavia che ulteriori studi sulla programmazione rivelino la convenienza e l'opportunità di aggiunte o modifiche di qualche entità:

- A XY Sommare il contenuto (XY) della cella XY all'Acc. In simboli:  
 $(Acc) + (XY) \rightarrow Acc.$
  - Ac XY Sommare cancellando;  $(XY) \rightarrow Acc.$
  - B XY Sottrarre;  $(Acc) - (XY) \rightarrow Acc.$
  - Bc XY Sottrarre cancellando:  $-(XY) \rightarrow Acc.$
  - L XY Prodotto logico (cifra a cifra);  $(Acc) \& (XY) \rightarrow Acc.$
  - Tr XY Trasferire (XY) nel moltiplicatore:  $(XY) \rightarrow Mr.$
  - M XY Moltiplicare (XY) per (Mr). Le prime 39 cifre del prodotto col segno si formano in Acc, le rimanenti (col giusto segno) in Mr;  $(Mr) \cdot (XY) \rightarrow \{Acc, Mr\}$
  - Vd XY Traslazione corta (solo accumulatore) di  $n=XY$  posti a destra  
 $(Acc) \cdot 2^n \rightarrow Acc.$
  - Vs XY Traslazione corta a sinistra:  
 $(Acc) \cdot 2^{-n} \rightarrow Acc.$
  - Wd XY Traslazione lunga (accumulatore e moltiplicatore insieme) a destra  
 $[(Acc) + 2^{-39} (Mr)] \cdot 2^n \rightarrow \{Acc, Mr\}$
  - Ws XY Traslazione lunga a sinistra:  
 $[(Acc) + 2^{-39} (Mr)] \cdot 2^{-n} \rightarrow \{Acc, Mr\}$
- Una definizione più precisa delle ultime istruzioni che mostra gli spostamenti delle diverse cifre é data nella descrizione strutturale dell'organo aritmetico.

A queste istruzioni bisogna aggiungere quelle di trasferimento dei risultati ottenuti in Acc e in Mr alla memoria.

- Ta XY: Trasferire (Acc) in XY:  $(Acc) \rightarrow XY.$
- Tm XY: Trasferire (Mr) in XY:  $(Mr) \rightarrow XY.$

**CONTROLLO:** E' formato dal complesso dei dispositivi che servono ai due scopi fondamentali della ricezione, modificazione e interpretazione delle istruzioni e del coordinamento di tutte le parti della macchina fra loro. La struttura dettagliata verrà esposta in seguito: Qui ci limitiamo a descrivere le funzioni. Occorre premettere alcune considerazioni generali sulle rappresentazioni delle istruzioni. Un'istruzione generica deve contenere essenzialmente due dati: innanzitutto un'indicazione dell'operazione da eseguire (ad esempio A, Ac, M, Tr etc...) e poi un numero che rappresenta l'indirizzo di una

e via di seguito. Tale numero, in base ai dati precedentemente descritti, è un numero intero compreso fra 0 e 1023 e quindi rappresentabile con un gruppo di 10 cifre binarie. È possibile assegnare anche ai simboli d'operazione (cioè A, Ac etc..) con cui vengono indicate le singole istruzioni un gruppo di cifre binarie. Con 5 cifre binarie (corrispondenti abitualmente a un carattere di telescrivente) si ottengono 32 combinazioni diverse con cui rappresentare 32 simboli d'operazione, e quindi istruzioni, diverse. Come vedremo, altre cinque cifre binarie verranno usati per introdurre eventuali modifiche nell'istruzione prima che essa venga eseguita. In totale, quindi, un'istruzione richiede 20 cifre binarie cioè mezza parola. In questo modo è possibile conservare le istruzioni nella memoria, due per cella.

Per comprendere il funzionamento della macchina cominciamo col pensare di avere scritto in modo qualsiasi, in certe celle della memoria principale, un certo programma di istruzioni che la macchina deve eseguire l'una dopo l'altra. Come si è detto, le istruzioni occupano ciascuna mezza parola. Distingueremo così fra la prima e la seconda metà di una cella. Il controllo provvede a chiamare nell'opportuna successione le singole istruzioni, a interpretarle e a diramare tutti gli opportuni comandi e segnali ai diversi organi della macchina per l'esecuzione dell'istruzione stessa. Possiamo distinguere due fasi: la prima, che chiameremo ciclo d'istruzione, corrisponde a tutte le operazioni che il controllo compie per chiamare, ed eventualmente modificare, le istruzioni ad una ad una; la seconda, che chiameremo ciclo esecutivo, consiste invece nell'interpretazione ed effettiva esecuzione dell'istruzione stessa. Prescindendo per il momento dalla questione della modifica delle istruzioni, ciò avviene nel modo seguente: nel controllo c'è un registro, detto il numeratore  $N$  che contiene l'indirizzo di una mezza cella della memoria. In questo registro è contenuto all'inizio del ciclo d'istruzione l'indirizzo dell'istruzione che la macchina deve eseguire. Opportuni segnali fanno sì che tale mezza parola venga trasferita dalla memoria a un altro registro del controllo che chiameremo il registro delle istruzioni R.I.. Prima di giungere a tale registro essa può essere modificata in base al contenuto dei cosiddetti registri di controllo come specificheremo fra breve. Giunta al registro delle istruzioni essa viene decodificata ed eseguita. Al termine dell'esecuzione, salvo casi speciali, il numeratore viene modificato in maniera che il suo contenuto indichi l'indirizzo della mezza parola che contiene l'istruzione successiva (proprio nel senso letterale del termine). Comincia allora un nuovo ciclo di istruzione e via di seguito (fino che non giunga un ordine di arresto). Le eccezioni a queste sequenze (che chiameremo tratti lineari del programma) sono date da tutte le circostanze in cui si voglia saltare da una certa successione di istruzioni ad un'altra. Questo viene ottenuto mediante uno speciale gruppo di istruzioni dette genericamente salti.

Di questi vi sono vari tipi: c'è un salto incondizionato il cui effetto è quello di scrivere direttamente nel numeratore l'indirizzo di cui deve essere estratta la prossima istruzione; ci sono diversi tipi di salti il cui effetto è di cambiare come sopra il contenuto del numeratore oppure passare semplicemente all'istruzione successiva subordinatamente al verificarsi di certe condizioni. Tre di questi salti condizionati riguardano condizioni aritmetiche e sono:

S° XY Se in un opportuno indicatore della parte aritmetica c'è l'indicazione di un traboccamento avvenuto, saltare all'istruzione contenuta nella prima metà della cella di indirizzo XY, in caso contrario proseguire normalmente con l'istruzione successiva.

S +XY Come sopra, solo che il salto è subordinato all'essere il contenuto dell'Acc.  $\geq 0$

S -XY Come sopra, solo che il salto è subordinato all'essere il contenuto dell'Acc.  $< 0$

Queste istruzioni permettono ovviamente di decidere sull'ulteriore corso del programma in base ad eventualità previste ma non conosciute al momento della programmazione.

Di tipo analogo sono gli ordini di salto incondizionato e di arresto già ricordati.

Z XY Saltare incondizionatamente alla prima metà della cella XY.

F XY Preparare in N l'indirizzo XY, ma arrestarsi subito dopo.

Vi sono altre due istruzioni dello stesso tipo dei salti condizionati solo che sono subordinati all'essere  $\geq 0$  oppure  $< 0$  il contenuto di un certo registro di controllo. Non differiscono sostanzialmente dalle istruzioni S+ e S-; prima di precisarle però è necessario descrivere l'uso dei registri di controllo cui si è più volte accennato. L'opportunità di introdurre dei registri di controllo R.C. con le funzioni che specificheremo, proviene dalle seguenti due circostanze. Nella maggior parte dei programmi si incontrano dei "cicli", vale a dire delle successioni di operazioni che vanno ripetute più volte (il numero di volte può essere o no fissato a priori). Accade sovente che il ciclo debba essere ripetuto su operandi contenuti in celle diverse della memoria. In tal caso occorre modificare di volta in volta la parte dell'istruzione che contiene l'indirizzo dell'operando. Ciò può farsi inviando l'istruzione prima di eseguirla nell'Acc della parte aritmetica e sommando ad essa una opportuna quantità in modo da modificare come voluto le cifre riguardanti l'indirizzo. L'istruzione così modificata viene rinviata nella memoria e di lì passata al controllo per l'esecuzione. Quanto è detto dà solo un'idea generica di questo procedimento che si presenta nella realtà abbastanza complesso. L'inconveniente maggiore è che, per l'esecuzione di tali trasformazioni, bisogna interrompere il ciclo operativo e vuotare l'unità aritmetica. La cosa risulta maggiormente complicata quando si tenga conto del fatto che molti cicli fanno parte di sottoprogrammi, i quali possono essere preparati in forma standard. È chiaro che in tal forma le istruzioni contenute non possono far riferimento alle celle in cui sono effettivamente contenuti gli operandi, variando queste da caso a caso. È questo il cosiddetto problema dell'invarianza dei sottoprogrammi.

L'idea di introdurre nel controllo stesso un secondo accumulatore e un certo numero di registri di memoria per eseguire direttamente la modifica di istruzioni senza passare per la parte aritmetica è stata introdotta e usata con successo nelle macchine costruite dalla ditta Ferranti. La soluzione da essi adottata presenta tuttavia ancora diverse difficoltà per quanto riguarda il problema dell'invariazione dei sottoprogrammi. I vantaggi che si possono ottenere nella programmazione e nell'efficienza delle macchine calcolatrici trovando delle buone soluzioni per questi problemi.

anche se a prezzo di qualche ulteriore complessità; ci hanno indotto a studiare attentamente l'argomento. Né è risultato lo schema di controllo che presentiamo nella II parte; e di cui ora indicheremo il funzionamento generale.

Il controllo è stato dotato di un complesso di 16 registri di 20 bit ciascuno. Per specificare uno di tali registri occorrono quindi 4 bit. Ordinariamente un'istruzione prima di giungere in R.I. arriva in un altro registro, detto il registro di entrata, R.E. Ricordiamo che un'istruzione è composta di 20 cifre binarie di cui le prime cinque sono state riservate (gruppo di controllo) appunto in connessione con l'uso dei registri di controllo. Della funzione della prima di tali cifre diremo dopo; le altre 4 specificano appunto l'indirizzo di uno dei registri di controllo. Quando l'istruzione giunge in R.E. nel caso ordinario (prima cifra 1) viene immediatamente selezionato il registro di controllo indicato dai rimanenti 4 bit. L'istruzione contenuta in tale registro e il contenuto del registro selezionato giungono ad un addizionatore (addizionatore di controllo A.C.) sommate e il risultato inviato in R.I. Solo a questo punto inizia il ciclo di esecuzione, secondo la terminologia precedentemente descritta. Questo avviene sempre automaticamente. Qualora non si volesse modificare la istruzione si farà riferimento a un R.C. che contiene lo zero. In questo modo non si introduce una modifica permanente delle istruzioni. Per modificare un indirizzo in modo diverso ad ogni ciclo è sufficiente alterare il contenuto del R.C. relativo; e ciò che non impegna né la parte aritmetica né trasferimenti con la memoria.

Nell'introduzione di sottoprogrammi, come già si è accennato, è in generale necessaria una doppia modifica delle istruzioni. Una di queste deve essere permanente e serve per sistemare il valore dei parametri posizionali che adattano il sottoprogramma stesso alla posizione da esso effettivamente occupata nella memoria in quel dato problema e alla posizione dei dati che tale sottoprogramma deve elaborare; la seconda invece è della natura ripetitiva precedentemente descritta. Si ottiene tutto ciò nel modo seguente, facendo uso della prima cifra del gruppo di controllo. Abbiamo descritto cosa avviene quando essa è eguale a 1. Tutte le istruzioni di un sottoprogramma che richiedono la doppia modifica vengono inizialmente trascritte nella memoria con la prima di tali cifre =0. Quando l'istruzione giunge in R.E., essa viene modificata come d'ordinario; ma invece di passare al registro delle istruzioni per la sua esecuzione viene rispedita sia al R.E. sia alla cella della memoria da cui era stata estratta.

Si ottiene così la modifica permanente dell'istruzione nella memoria principale, e al tempo stesso l'istruzione già così modificata si presenta al R.E. per una seconda modifica. In generale si fa in modo che ora la prima cifra sia =1 onde questa seconda modifica non è permanente e rientra nello schema di funzionamento precedentemente descritto. I dettagli di tutte queste cose sono alquanto complessi: del resto si tratta né più né meno che dell'intera tecnica della programmazione. Per terminare questa parte generale sul controllo resta da specificare il gruppo delle istruzioni che si trasferiscono ai R.C. (preparazione, modifica, etc....).

Vi sono due istruzioni che riguardano il caricamento e lo scaricamento dei R.C. in blocco; le altre quattro riguardano invece un solo registro, di cui bisogna quindi specificare l'indirizzo. Sarebbe utile adoperare le cifre corrispondenti al primo carattere: ma, in tal modo, queste istruzioni non sarebbero più modificabili, e questo sarebbe un grave inconveniente. Poiché non vi sono altri bit a disposizione in una mezza parola è stato deciso che tali istruzioni sono rappresentate da una intera parola: la prima metà viene trattata nel modo ordinario; quando però essa giunge in R.I. viene automaticamente chiamata la seconda parte che si ferma in R.E. e che opera la selezione del particolare R.C. in questione. Di questa seconda parte vengono usati quindi in sostanza solo 5 bit. Questo appare uno spreco: sono allo studio alcuni metodi che permettono di utilizzare efficientemente le informazioni supplementari che tali cifre possono portare evitando tale inconveniente.

Seguono le istruzioni speciali per i registri di controllo:

- Z+ XY, i/// Se il contenuto (i) dell'i.mo R.C. è  $\geq 0$  saltare all'istruzione (XY), in caso contrario proseguire alla prossima.
- Z- XY, i/// Id. solo che il salto avviene per (i)  $< 0$ .
- C XY, i/// Sommare al contenuto del R.C. i il numero XY.
- Ch XY, i/// Trasferire in i la seconda metà di (XY).
- P XY Trasferire nelle celle da XY a XY+7 il contenuto dei 16 registri di controllo.
- Q XY Trasferire nei 16 registri di controllo il contenuto delle celle da XY a XY+7.

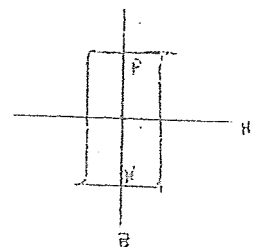
Le prime due di queste istruzioni sono del tutto analoghe ai salti condizionati sulla parte aritmetica. La loro funzione è importantissima nella programmazione di cicli ripetuti un numero prefissato di volte.

L'istruzione C serve per la modifica dei registri di controllo, mentre le rimanenti tre regolano gli scambi con la memoria.

## II° Parte

Passiamo alla descrizione dello schema strutturale, ovvero di un diagramma a blocchi dei singoli organi, in cui sono indicate le unità fondamentali che compongono i diversi organi con la specificazione delle funzioni che devono svolgere. Non includeremo, per brevità, lo schema relativo alla memoria ausiliaria a tamburo magnetico, dato che si tratta di un organo ormai standard e le cui funzioni sono limitate a scambi di informazioni solo con la memoria principale: memoria principale, organo aritmetico e controllo.

La memoria principale è a nuclei magnetici con ciclo d'isteresi rettangolare. La conservazione delle cifre binarie avviene sfruttando i due stati stabili di magnetismo residuo, per esempio assegnando il valore 1 a un nucleo nello stato P e il valore 0 a un nucleo nello stato N. Una memoria di questo tipo è permanente nel senso che la conservazione delle informazioni non dipende





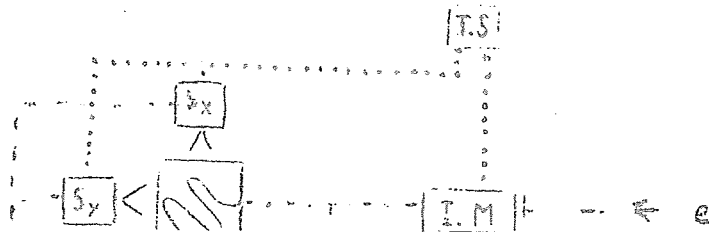
disposti su 40 piani paralleli, di  $32 \times 32 = 1024$  nuclei ciascuno. La selezione di una cella avviene impulsando tutti i nuclei di stesse coordinate X, Y sui 40 piani. L'uscita e l'entrata avvengono, nel modo che vedremo, su di un'unica linea a 40 canali, ciascuno dei quali attraversa tutti i nuclei di uno stesso piano. Il sistema di selezione, la cui realizzazione fisica presenta alcune difficoltà per le forti correnti che deve erogare, è formato essenzialmente da due matrici indipendenti,  $S_x$  ed  $S_y$ , una per la coordinata X, l'altra per la Y. Ciascuna matrice ha una linea di entrata in continua a 5 canali e 32 uscite impulsive, più un'entrata di comando impulsiva.

Al giungere del comando impulsivo, su una delle 32 uscite, che dipende dall'informazione presente sulle linee d'entrata, esce un impulso di corrente della forma a) e di intensità 0,5 A circa pari alla metà di quella occorrente per far subire transizioni ai nuclei.

Tale impulso non ha praticamente alcun effetto sui nuclei della riga e colonna selezionate, tranne su quelli (uno per ogni piano) posti all'intersezione di tale riga o colonna, in cui ne giungono due. I nuclei selezionati eseguono un intero ciclo se si trovano nello stato 0, mezzo ciclo se nello stato 1; in ogni caso al termine di questa prima fase tutti i nuclei sono stati portati nello stato 0, il che equivale a dire che la corrispondente cella della memoria è stata cancellata. Durante la transizione, ogni nucleo induce nel canale di uscita del suo piano un impulso della forma b), se il nucleo selezionato era nello stato 0, della forma c), se era nello stato 1. In questo modo è possibile effettuare la lettura della parola conservata nella cella selezionata della memoria.

La scrittura si esegue partendo dalle celle cancellate come detto sopra, e inviando di nuovo un impulso all'entrata delle matrici di selezione. Questo provoca ancora la formazione di un impulso della forma a), che farebbe eseguire un intero ciclo a tutti i nuclei selezionati. Se però, simultaneamente all'impulso descritto, se ne invia uno della forma d) sul canale di uscita di un piano, il nucleo selezionato esegue solo metà del ciclo (la seconda metà risultando inibita) e si trova quindi nello stato 1 al termine dell'operazione. Poiché i canali di uscita sono uno per piano, è possibile scrivere una parola nella cella selezionata inviando l'impulso di inibizione dove si vuole scrivere 1, non inviandolo dove si vuole 0.

- a) \_\_\_\_\_
- b) \_\_\_\_\_
- c) \_\_\_\_\_
- d) \_\_\_\_\_

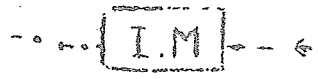


Nella fig. 2 è indicata schematicamente la struttura dell'intera memoria. Si notano: le matrici di selezione  $S_x$  ed  $S_y$ ; la memoria propriamente detta; il registro di memoria R.M. e altri organi di cui illustreremo ora il funzionamento.

T' un temporizzatore: l'arrivo dell'impulso in forma un impulso in  $T1^{\wedge}$ , e dopo un tempo opportuno impulsi su entrambe le linee,  $T1^{\wedge}$  e  $T2^{\wedge}$



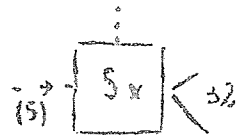
Il comando in entrata provoca l'emissione di un impulso inibitore della forma d)



R' un registro con scrittura automatica: scrive 0 se riceve un impulso del tipo b), 1 se del tipo c); l'uscita è continua.



Matrice di selezione; al giungere di un comando di funzionamento in  $S_x$  trasmette su una delle 32 linee in uscita, selezionata dalla  $S_y$  entrate, un impulso della forma a). Ogni linea d'uscita è a 40 canali.



Il funzionamento del sistema è il seguente: quando TS riceve (dal controllo) il comando di funzionamento, trasmette dapprima un impulso in  $T1^{\wedge}$  che giunge in  $S_x^{\wedge}$  e  $S_y^{\wedge}$  provocando la lettura della cella selezionata, il cui contenuto passa in R.M. e ivi viene convertita in livelli continui. Questo provoca la cancellazione della cella letta. Nella seconda fase TS trasmette un altro impulso su entrambe le uscite  $T1^{\wedge}$  e  $T2^{\wedge}$  provocando, mediante gli impulsi di inibizione provenienti da TS accoppiati con quelli di selezione da  $S_x^{\wedge}$  e  $S_y^{\wedge}$ , la scrittura nella stessa cella dell'informazione presente sulla linea d'entrata della memoria.

In scrittura tale informazione proverrà da altri organi della macchina. In lettura si fa in modo che l'uscita di R.M. sia connessa all'entrata di I.M. in modo da ottenere automaticamente la reinscrizione della parola letta. Di ciò sarà detto in dettaglio parlando del controllo.

### Organo aritmetico

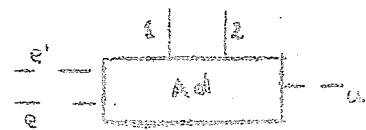
Abbiamo già descritto nelle grandi linee la struttura dell'unità aritmetica dal punto di vista matematico e sono state già indicate le operazioni che essa deve sapere eseguire dietro opportuni comandi del controllo. Riportiamo ora uno schema a blocchi dell'unità stessa, con l'indicazione precisa delle funzioni che i singoli componenti devono svolgere. Per il momento si tratta di indicare i collegamenti fra i singoli blocchi e il modo in cui essi devono operare per eseguire le istruzioni che riguardano l'unità aritmetica, e che sono precisate dalla lista descritta nella I parte.

Come risulta dalla figura 3, l'unità aritmetica consta essenzialmente

vono essenzialmente per la conservazione di una parola; sono dotati di un comando impulsivo di entrata e di uscita continua. Possono inoltre eseguire traslazioni a destra e a sinistra. I due gruppi di trasformazione posseggono invece due linee di entrata a 40 canali e una sola linea di uscita, sempre a 40 canali, tutte in continua. Essi non sono dotati di memoria, ma eseguono opportune combinazioni delle informazioni che ricevono in entrata e le presentano direttamente all'uscita. L'uscita dipende quindi dai livelli continui presenti all'entrata oltre che da certi segnali che dipendono dal tipo di combinazione richiesto in un certo momento. Tali segnali, che determinano sostanza l'operazione che la macchina deve eseguire, giungono in generale dal controllo. Nel caso della moltiplicazione che richiede tutta una serie di operazioni più elementari giungeranno da un elemento di controllo interno all'unità aritmetica stessa. Tale sistema, che non figura nel diagramma indicato, verrà descritto in seguito. Per ora ci limitiamo a descrivere il funzionamento dell'unità aritmetica nell'esecuzione delle istruzioni più semplici (somme, sottrazioni, prodotto logico, trasferimenti alla memoria principale). A tale scopo bisogna precisare le funzioni dei due elementi di trasformazione, cioè precisare quale combinazione delle due entrate si presenta in uscita a seconda della combinazione di livelli (0 o 1 presenti sulle entrate e di segnale. Ciò è indicato nella tabella seguente:

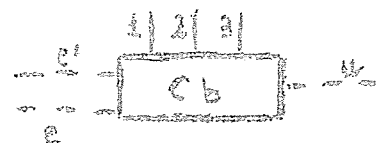
Addizionale

Ade	Ade'	Ad1	Ad2	Adu
A	B	0	0	A
A	B	0	1	$A+2^{-39}$
A	B	1	0	$A+B$
A	B	1	1	$A+B+2^{-39}$



Combinatore

Cbe	Cbe'	Cb1	Cb2	Cb3	Cbu
A	B	0	1	0	A
A	B	0	0	1	A'
A	B	1	0	0	$A \& B$
A	B	0	0	0	0



Come si è detto, tali elementi di trasformazione presentano direttamente all'uscita la combinazione determinata dai livelli di segnale. La presentazione è praticamente istantanea nel senso che, dopo un breve transitorio, essa segue direttamente le variazioni nelle entrate.

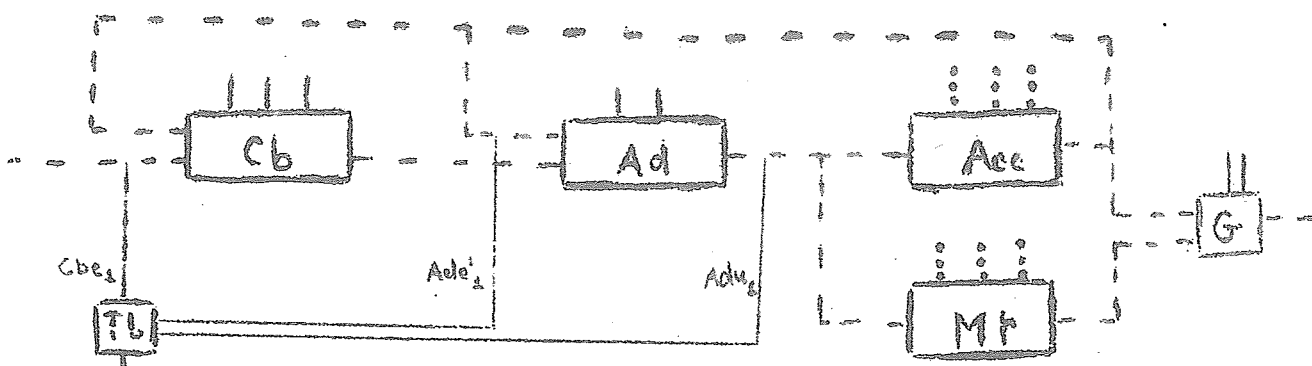


fig. 3

Possiamo ora descrivere ciò che occorre per l'esecuzione delle singole istruzioni.

XY Sommare al contenuto di Acc il contenuto della cella XY delle memorie. - Ricevendo tale istruzione il controllo fa in modo che la parola contenuta nella cella indicata si presenti sotto forma di livelli di tensione all'entrata Cbe del Combinatore.

Intanto il controllo ha inviato anche la combinazione di segnale (0 1 0) per il Combinatore e la combinazione (1 0) per l'addizionatore. All'uscita di Cb, che è anche una delle entrate di Ad, compare quindi (XY). All'altra entrata Ad c'è (Acc). Né segue che all'uscita di Ad compare la somma richiesta (Acc)+(XY). Dopo un tempo opportuno perché le tensioni si siano correttamente stabilizzate, il controllo invia un comando impulsivo ad Acc  $\uparrow$ ; il registro Acc è fatto in maniera tale che al giungere di tale comando il contenuto precedente viene cancellato e sostituito dalla parola in entrata, precisamente (Acc)+(XY). Bisogna osservare a questo punto che se per scrivere qualcosa nel registro Acc fosse veramente necessario cancellarlo prima, questo porterebbe automaticamente una modifica dell'informazione stessa che deve essere scritta, che dipende a sua volta dal contenuto di Acc. Questa difficoltà logica è superabile facendo uso, per tale registro di un peculiare elemento di memoria a linea di ritardo, ideato dall'I.B.M. e descritto nella letteratura. Per il momento basterà pensare che le cose vanno come se la cancellazione seguisse la scrittura, evitando quindi la difficoltà logica accennata.

XY Cancellare dall'Accumulatore e sommarvi (XY). Tale operazione segue esattamente lo schema precedente, solo che la combinazione (1 0) per l'addizionatore viene sostituita dalla (0 0). Si vede allora che il contenuto della cella XY presente all'ingresso Cbe passa inalterato attraverso Cb e Ad e giunge all'ingresso di Acc, ove, al comando in Acc  $\uparrow$ , sostituisce semplicemente il contenuto precedente.

B XY Per la sottrazione  $a=b - c$  ricordiamo che  $-c$  é rappresentato come  $c' + 2^{-39}$  onde si ha:  $a = b + c' + 2^{-39}$  ove  $c'$  é la negazione (logica) di  $c$ . Ne segue per B XY (Sottrarre al contenuto di Acc il contenuto della cella XY della memoria), che tale operazione si ottiene disponendo la combinazione (0 0 1) per il combinatore, da cui esce quindi  $(XY)'$  e la combinazione (1 1) per l'addizionatore, che forma così il numero  $z = (Acc) - (XY) = (Acc) + (XY)' + 2^{-39}$ . Per la scrittura di tale numero in Acc vale quanto detto precedentemente.

Bc XY Cancellare l'Accumulatore e sottrarvi  $(XY)$ -Questa operazione equivale a scrivere nell'Accumulatore il negativo  $-(XY)$  di  $(XY)$ . Si ottiene con le combinazioni (0 0 1) per il combinatore e (0 1) per l'addizionatore.

Tutte queste operazioni possono dare luogo a traboccamenti (che sono i casi in cui il risultato della somma algebrica esce dal campo fondamentale  $-1 < c < 1$ ). Per il controllo del traboccamento si dispone nell'unità aritmetica di un elemento Tr che confronta in modo opportuno la prima cifra in entrata su Cbe con la prima cifra in entrata su Ade e la compara con la prima cifra in uscita da Adu. Da tale elemento esce un segnale che viene inviato al controllo. Tale elemento deve formare un 1 (indicazione di avvenuto traboccamento) se

Addizione			Sottrazione		
Cbe	Ade'	Adu	Cbe	Ade'	Adu
0	0	1	0	1	1
1	1	0	1	0	0

L XY Formare il prodotto logico cifra a cifra di  $(Acc)$  con  $(XY)$  e porlo in Acc. - Si ottiene semplicemente formando le combinazioni (1 0 0) per il combinatore e (0 0) per l'addizionatore, e inviando il comando di scrittura in Acc  $1c^*$

Poiché la moltiplicazione fa uso delle traslazioni, descriviamo dapprima tale gruppo di istruzioni. Per ragioni connesse con la moltiplicazione che risulteranno chiare in seguito, il registro Mr ha una lunghezza di 42 bit, invece degli ordinari 40. Per accedere a tale registro si fa uso dell'istruzione:

Tr XY Trasferire in (nei 40 bit intermedi di) Mr il contenuto della cella XY della memoria, cancellando il contenuto precedente. Le combinazioni per Cb e Ad sono identiche all'istruzione Ac XY (vedi sopra), tranne che alla fine il controllo invia un comando di lettura in Mr 1 (anziché Acc 1).

In tal modo i 20 bit della parola in entrata vanno a sostituire i 40 bit intermedi di Mr mentre nell'ultimo e nel primo posto viene scritto senza altro uno 0. Lo 0 finale serve per la moltiplicazione. L'uscita di Mr avviene invece sui primi 40 canali.

Il primo di questi, che é messo a 0 in entrata, viene, invece, accor-

tivo di ciò é connesso con il fatto che il risultato di tali operazioni non é direttamente rappresentabile con un numero semplice (di 40 bit) ma con una coppia (p,q) di cui la seconda parte é contenuta in Mr, e che deve avere come prima cifra quella di Acc.

Indicando con  $(a_0, a_1, \dots, a_{39} / m_0, m_1, \dots, m_{40}, m_{41})$  il contenuto originario dei registri Acc e Mr, le istruzioni di traslazione risultano così definite:

Vd XY traslazione corta di  $n=XY$  posti a destra;

$$(a_0, a_0, \dots, a_0, a_1, \dots, a_{39-n})$$

Vs XY traslazione corta di  $n=XY$  posti a sinistra:

$$(a_n, \dots, a_{39}, 0, \dots, 0)$$

Wd XY; traslazione lunga di  $n=XY$  posti a destra:

$$(a_0, a_0, \dots, a_0, a_1, \dots, a_{39-n} / a_0, a_{39-n+1}, \dots, a_{39}, a_1, \dots, m_{39-n})$$

Ws XY; traslazione lunga a sinistra:

$$(a_n, \dots, a_{39}, m_1, m_n, a_n, m_{n+1}, m_{41}, 0, \dots, 0)$$

Facendo uso dei registri a linee di ritardo, cui si é già accennato, l'esecuzione di tali istruzioni é semplicissima.

Per le prime due istruzioni basterà inviare un impulso di durata proporzionale a  $n$ , rispettivamente, in Acc2 e Acc3, che indicheremo con Acc 2 n e Acc 3 n.

Per le traslazioni lunghe si apre dapprima, nella direzione opportuna, una porta Pr che permette il passaggio dell'ultima cifra di Acc alla seconda di Mr e viceversa. Seguono, per le traslazioni a destra, gli impulsi Acc2 n e Mr 2 n, e per le traslazioni a sinistra gli impulsi Acc 3 n e Mr 3 n.

Per quanto riguarda la moltiplicazione ricordiamo che il risultato di tale operazione dà luogo a due numeri semplici  $p, q$ . Per conservare tali numeri occorrerebbe un registro doppio o due semplici.

Seguendo un'idea di Von Neumann, é possibile combinare le cose in modo che il prodotto del numero contenuto in Mr con quello presente sulla linea di ingresso dell'unità aritmetica, proveniente da una cella XY della memoria, si presenti, alla fine del ciclo operativo, con la prima metà  $p$  in Acc e con la seconda metà  $q$  in Mr. Questo é, infatti, quanto la macchina fa quando esegue la istruzione:

M XY E' ovvio che la moltiplicazione si può eseguire mediante ripetute operazioni di somma e traslazione (lunga) a destra. Alcune complicazioni provengono dalla decisione presa di rappresentare i numeri negativi mediante complementi a2 (anziché, ad esempio, in modulo a segno). Questa difficoltà, tuttavia, si può superare in vari modi. Noi seguiremo il procedimento di Booth & Booth, che presenta il vantaggio di non richiedere correzioni finali a seconda dei casi, ma fornisce direttamente il risultato corretto (ad eccezione, ovviamente del caso  $(-1) \cdot (-1)$  che dà luogo a traboccamento).

Segue la descrizione schematica del procedimento, senza giustificazione. Per la giustificazione matematica di esso, come pure per la descrizione degli altri metodi si rimanda alla letteratura sull'argomento.

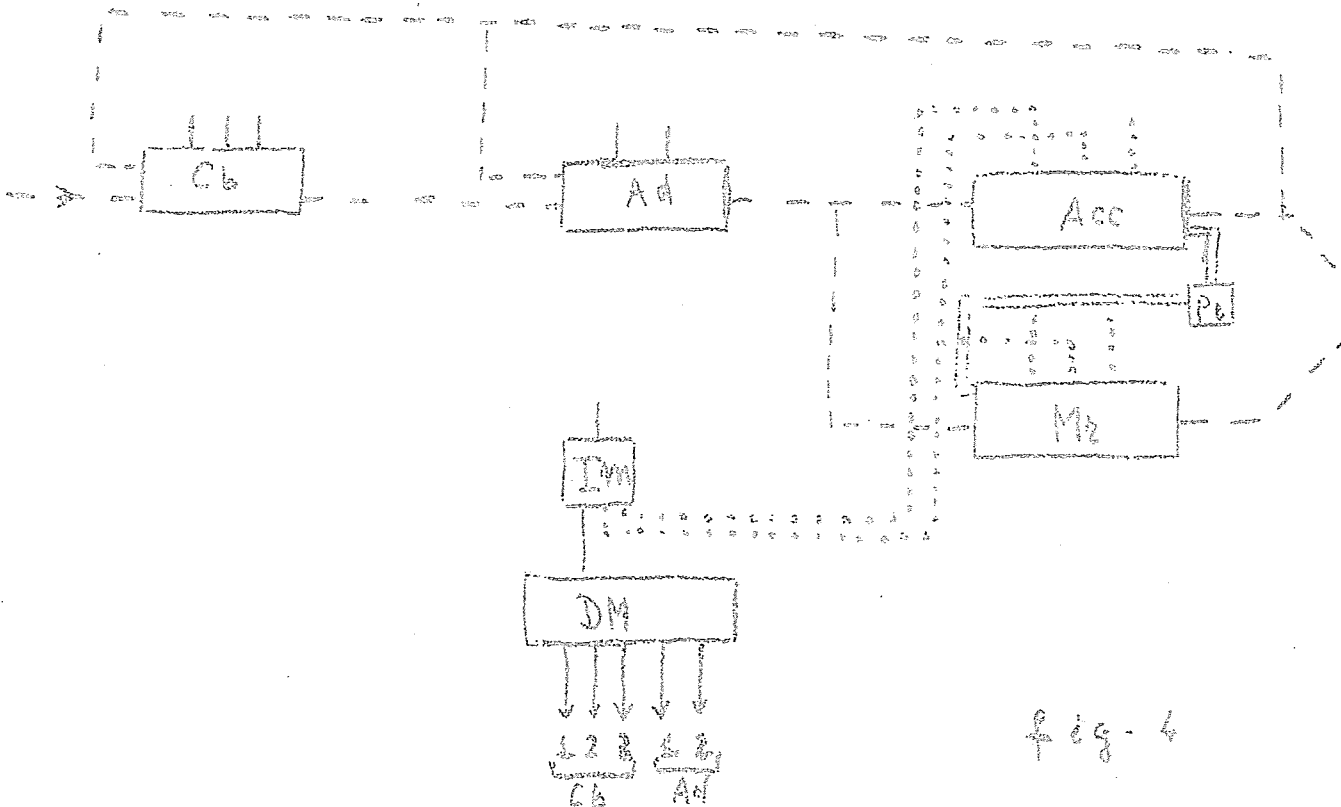
Indichiamo con  $a_0, a_1, \dots, a_{20}$  le cifre dell'Accumulatore Acc, con

$r_1, \dots, r_{39}$  le cifre del moltiplicando (XY).

Il procedimento si svolge ora mediante 40 passi successivi, che si possono così descrivere: si cancella, inizialmente Acc. Seguono le fasi:

- I) Si sentono le ultime cifre di Mr:  $m_{40}$  e  $m_{41}$ :
- se  $m_{40} = m_{41}$  si esegue la sostituzione  $(\underline{Acc}) \quad (\underline{Acc})$
- se  $m_{40} \quad m_{41} \quad " \quad " \quad " \quad "$   $(\underline{Acc}) - (\underline{XY}) \quad (\underline{Acc})$
- se  $m_{40} \quad m_{41} \quad " \quad " \quad " \quad "$   $(\underline{Acc}) = (\underline{XY}) \quad (\underline{Acc})$
- II) In ogni caso, segue una traslazione lunga a destra.

Queste due fasi si succedono nell'ordine 39 volte. Dopo  $k$  passi ( $k=0, 1, \dots, 39$ ) in Acc ci sono le prime 39 cifre del  $k$ .mo prodotto parziale con segno, traslato di un posto a destra: in  $m_0$  c'è e resta 0; in  $m_1 \dots m_k$  ci sono le cifre residue del  $k$ .mo prodotto parziale; in  $m_{k+1} \dots m_{41}$  ci sono le prime  $41-k$  cifre del moltiplicatore. Per  $K=40$  (ultimo passo) si ripete la fase I), ma non la seconda che viene sostituita dall'aggiungere al primo posto di Mr e all'ultimo di Acc la prima cifra di Acc cioè la sua cifra del segno.



Indichiamo ora schematicamente, completando il diagramma a blocchi dell'unità aritmetica, come si possa eseguire automaticamente l'intero ciclo.

Prestando dalla cancellazione iniziale dell'Accumulatore e delle correzioni che seguono la prima fase dell'ultimo passo, il ciclo si svolge nel modo seguente.

Quando il controllo riceve l'istruzione  $M \cdot XY$  fa giungere  $(XY)$  all'ingresso  $C_{pe}$ . Invia poi l'impulso ad un impulsatore  $I_m$  interno alla unità aritmetica il quale, a sua volta, la trasmette al decodificatore  $DK$  che collega le due entrate con l'uscita delle ultime due cifre di  $Mr$ . Questo a seconda dei casi trasmette nelle diverse uscite indicate nella figura 4 diversi segnali. Questi predispongono  $C_b$  e  $A_d$  a sommare, sottrarre o semplicemente copiare  $Acc$  a seconda dei casi. Da  $I_m$  escono poi a tempi ben determinati due treni di impulsi: il primo di 40 impulsi giunge ad  $Acc 1^{\circ}$  e dà luogo così all'esecuzione della fase I) precedentemente descritta, il secondo di 39 impulsi intercalati fra i precedenti giunge contemporaneamente ad  $Acc 2^{\circ}$  e  $Mr 2^{\circ}$  dando una traslazione lunga destra, cioè all'esecuzione della fase II) di cui sopra.

Restano da descrivere le due operazioni di trasferimento alla memoria  $TaXY$  e  $TrXY$ , dai due registri aritmetici. Dal punto di vista dell'unità aritmetica ciò è semplicissimo trattandosi unicamente di eseguire l'opportuno collegamento mediante il commutatore G.

Controllo: Il funzionamento logico del controllo è già stato illustrato nella I parte. Si tratta ora di precisare di quali elementi esso deve essere composto in modo che possa svolgere la funzione assegnatagli, che è quella di chiamare, modificare, interpretare e ordinare l'esecuzione delle singole istruzioni. Secondo quanto già accennato, l'esecuzione di un'istruzione richiede due fasi distinte. Il ciclo d'istruzione comprende il complesso di operazioni che inizia con la lettura della ~~della~~ cella della memoria che contiene l'istruzione da eseguire e il cui indirizzo è contenuto in apposito registro  $H$  (numeratore) e termina con la scrittura dell'istruzione nel registro delle istruzioni  $R.I.$ . Questo ciclo è percorso da tutte le istruzioni. Segue il ciclo d'esecuzione, in cui le cinque cifre binarie che specificano l'istruzione da eseguire vengono decodificate in un apposito elemento  $Df$  (decodificatore) il quale in base all'informazione così ricevuta ed altre eventuali informazioni che giungano da altri organi della macchina (per es. dispositivi di sincronizzazione, l'indicatore di traboccamento o del segno dell' $Acc$ ; etc. predispone prima i diversi organi per l'esecuzione dell'istruzione richiesta mediante opportuni segnali continui 1,0, e invia poi i necessari comandi impulsivi d'esecuzione. La struttura interna del Decodificatore è molto complessa e rappresenta la parte meno standard di ogni macchina. Essa comunque non può essere precisata che quando si siano ben stabilite anche dal punto di vista tecnico, oltre che logico, la struttura fisica di tutti gli altri elementi della macchina. In una tabella che segue, la funzione del decodificatore è precisata dal punto di vista logico limitatamente agli organi e alle istruzioni finora presi in considerazione.



Descriviamo dapprima la struttura generale del controllo e i suoi necessari collegamenti con la memoria, compresi i circuiti relativi alla riscrittura della memoria principale delle parole lette, qualunque ne sia la destinazione.

Come si vede dalla fig. 5 le linee di entrata e uscita della memoria S (a 40 canali) giungono in una centrale di smistamento C.S., a cui giungono pure le linee d'entrata e uscita dell'organo aritmetico, (e degli altri organi che in questa nota non sono stati presi in considerazione). C.S. è collegato infine al controllo mediante due linee a 20 canali per la trasmissione delle istruzioni, che, come s'è detto, occupano in genere solo mezza parola. A tale C.S. è anche affidato il compito di riconnettere opportunamente la linea d'uscita della memoria con quella d'entrata per permettere la riscrittura delle celle cancellate da una precedente lettura. C.S., come i commutatori KeK' che figurano nel diagramma, è composta essenzialmente di congiunzioni "e" ed "o". La struttura logica di C.S. verrà indicata fra breve.

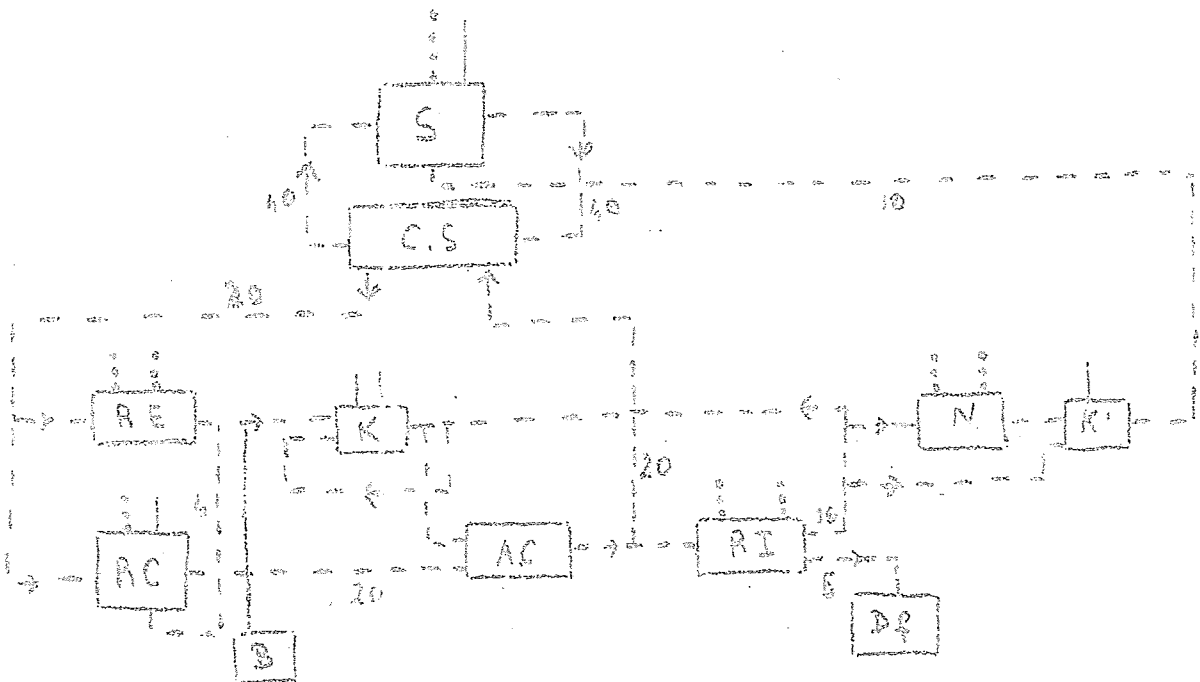


Fig. 5

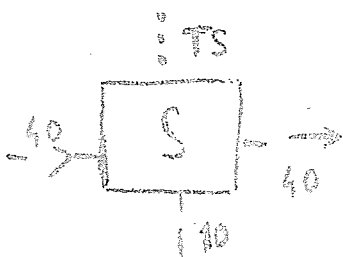
Il controllo é inoltre dotato di tre registri RE, RI, N con uscita in continua, scrittura su comando in RE° RI° N°. Essi sono anche dotati di una entrata impulsiva RE1° RI1° N1° che permette di sommare un'unità in posizione opportuna e precisamente nell'ultima per quanto riguarda N e RI e nella 5.a per RE) eseguendo i necessari riporti.

Fisicamente tali registri possono ottenersi con opportune modifiche del tipo dei registri usati nell'organo aritmetico.

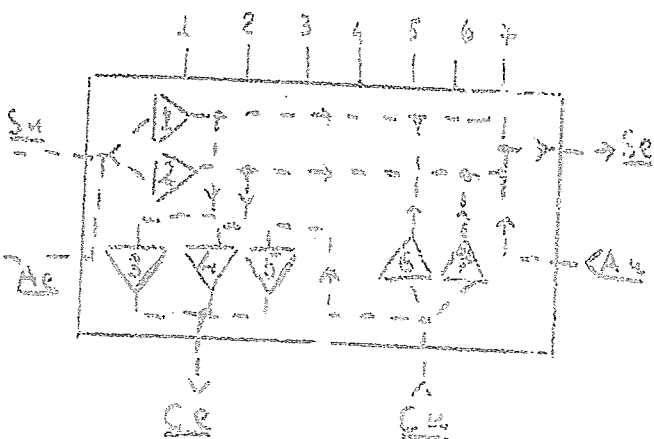
Questi registri, per le loro stesse funzioni, hanno lunghezza diversa RE dovendo contenere un'istruzione completa é di 20 bit, mentre RI é di soli 15 bit poiché nel ciclo esecutivo le prime 5 cifre binarie non giocano più alcun ruolo. N invece deve contenere l'indirizzo di un'istruzione da chiamare: esso deve avere quindi 11 bit: i primi dieci specificano una cella della memoria, l'ultimo se si tratta della prima o della seconda metà di tale cella.

Vi é anche una memoria speciale di 16 celle di 20 bit ciascuna e un addizionatore in continua di 20 bit analogo a quello aritmetico che servono per la modifica delle istruzioni. Completano il controllo gli elementi B e Df. Al primo di questi, B é affidato di decidere in base a una prima lettura dell'istruzione (più precisamente alla lettura delle prime cinque cifre binarie) sulle modalità del ciclo di istruzione (modifica permanente o temporanea delle istruzioni). Al secondo, Df, é affidata, come si é già accennato, la decodificazione dell'istruzione, e la sua esecuzione.

Segue la descrizione dei diversi elementi con la specificazione delle funzioni. Per C.S. e R.G. é indicata anche la struttura interna.



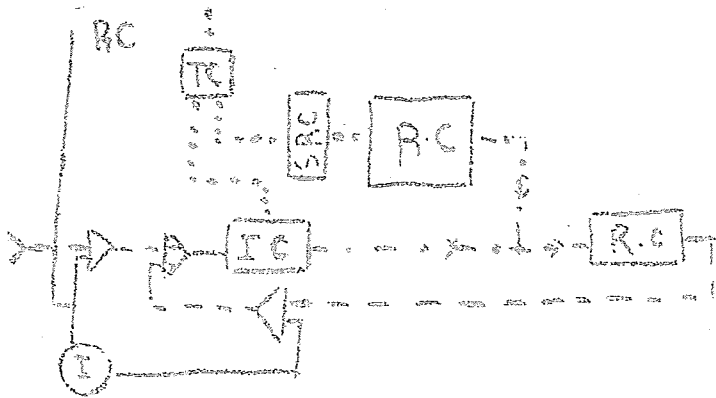
MEMORIA PRINCIPALE. Caratteristiche e struttura sono già state indicate. Per ciò che qui concerne basta ricordare che al giungere di un comando in TS° avviene prima la lettura della cella specificata dai 10 bit dell'indirizzo con uscita continua in Su e poi, automaticamente, la scrittura nella stessa cella dell'informazione presente su Se.



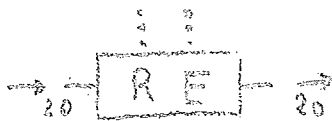
Centrale smistamento- A parte le necessarie congiunzioni "o" nei punti di confluenza ed eventuali ulteriori punti di diramazione verso altri organi, C.S. é costituito essenzialmente dalle congiunzioni "e" a 20 canali indicate in figura; comandate nell'ordine dalle entrate indicate superiormente. Le combinazioni che interessano sono:

1	2	3	4	5	6	7
1	1	/	/	/	0	0
0	0	/	/	/	0	0
1	1	1	0	0	0	0
0	1	0	0	1	1	0
1	0	0	0	1	0	1
0	1	/	/	/	1	0
1	0	/	/	/	0	1

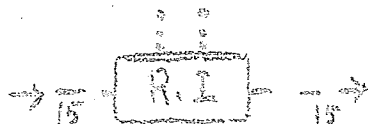
Lettura memoria e semplice riscrittura  
 Scrittura semplice della memoria  
 Lettura di un'istruzione (prima metà)  
 " " " " (seconda " )  
 Modifica permanente di un'istruzione (prima metà)  
 " " " " " " (seconda  
 Scrittura di un'istruzione (prima metà)  
 " " " " " " (seconda " )



Registri di controllo - formano una piccola memoria di 16 parole da 20 bit. Nello schema indicato a fianco essa è a nuclei magnetici del tipo della memoria principale, con riscrittura automatica. Più precisamente la lettura avviene ponendo R.C.1=0 e inviando comando in T.C.<sup>1</sup>. La scrittura avviene ponendo R.C.1=0 con lo stesso comando in T.C.



Registro d'entrata - 20 bit con uscita in continua. Comando di lettura RE<sup>1</sup> e comando di aumento di 1 in 5<sup>a</sup> posizione RE.1<sup>1</sup>



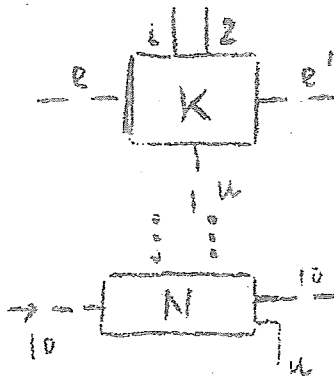
Registro istruzioni - 15 bit, uscita in continua. Comando lettura RI<sup>1</sup> e comando aumento di 1 in ultima posizione RI.1<sup>1</sup>.



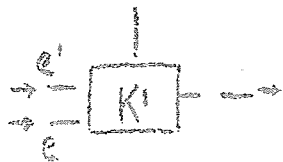
Addizionatore di controllo - A 20 canali con combinazione diretta dei livelli e uscita in continua.

Commutatore unidirezionale -  
(2+1) su linee a 20 canali 3 posizio-  
ni.

K 1	K 2	K u
1	0	e'
0	1	e
0	0	0



Numeratore - Registro di 11 bit cc  
uscita in continuo. Comando di let-  
tura  $N^{\wedge}$ . Comando per l'aumento di  
uno in ultima posizione  $N1^{\wedge}$ . Entra-  
ta sui primi 10 bit. Ultima cifra.



<u>Commutatore unidirezionale</u>		K' 1	K' u
2	1) su linee a 10 bit -	0	e
2	posizioni	1	e'

Passiamo alla descrizione del funzionamento, cominciando dal ciclo d'istruzione. Esso avviene in più fasi:

1.a fase; Si prepara a leggere mezza cella di memoria, (C.S. 1,1,u,a,0,0,0) all'indirizzo specificato da N ( $K'1=1$ ) u infatti indica l'ultima cifra di N, che decide se l'istruzione va letta nella prima o nella seconda metà della cella specificata dai rimanenti 10 bit di N.

Seguono i comandi  $TS^{\wedge}$ . (lettura memoria)  $RE^{\wedge}$ . L'istruzione selezionata viene così scritta in  $RE$ . La prima cifra giunge all'elemento B, mentre le quattro che seguono vanno al sistema di selezione dei RC.

Se in B è presente la cifra 0 inizia la II fase.

2.a fase: Si prepara la memoria a scrivere mezza parola allo indirizzo specificato da N. (C.S. u,u,0,0,1,u,u) e i R.C. a leggere ( $RC=0$ ). Si connette  $REe$  con  $AC$  e si sconnette  $R1u'$  ( $K1=1, K2=0$ ). Si invia comando di lettura in  $RC^{\wedge}$ . ai registri di controllo. Il contenuto di quello selezionato si somma in AC con il contenuto di RE. L'istruzione così modificata entra nella centrale di smistamento che la dirige sia all'entrata della memoria principale che a quella del registro RE.

Comandi in  $TS^{\wedge}$ . e  $RE^{\wedge}$ . effettuano la scrittura dell'istruzione così modificata nel registro d'entrata e nella memoria da do carattere permanente a tale modifica.

La situazione è come al termine della fase 1.

3.a fase: La memoria non interviene più (S indifferentemente). La preparazione di K e RC resta quella precedente. Segue comando di lettura in  $RC^*$ , e, avvenuta la somma in AC, comando di scrittura, in  $RI^*$ . L'istruzione completamente modificata giunge nel registro delle Istruzioni, terminando così il ciclo d'istruzione.

Un segnale di sincronismo inviato a  $Df^*$  dà allora inizio al ciclo di esecuzione.

Mentre il ciclo d'istruzione a parte il numero di ripetizioni della fase 2 é lo stesso per tutte le istruzioni, il ciclo esecutivo ovviamente si svolge in modo diverso a seconda delle istruzioni. Indicheremo con numeri d'ordine successivi le fasi che interessano le singole istruzioni, alcune delle quali ne richiedono una sola; altre due o tre e magari degli interi cicli (es. istruzioni P e Q)

Nella fig. 6 sono indicate le diverse fasi per le diverse istruzioni.

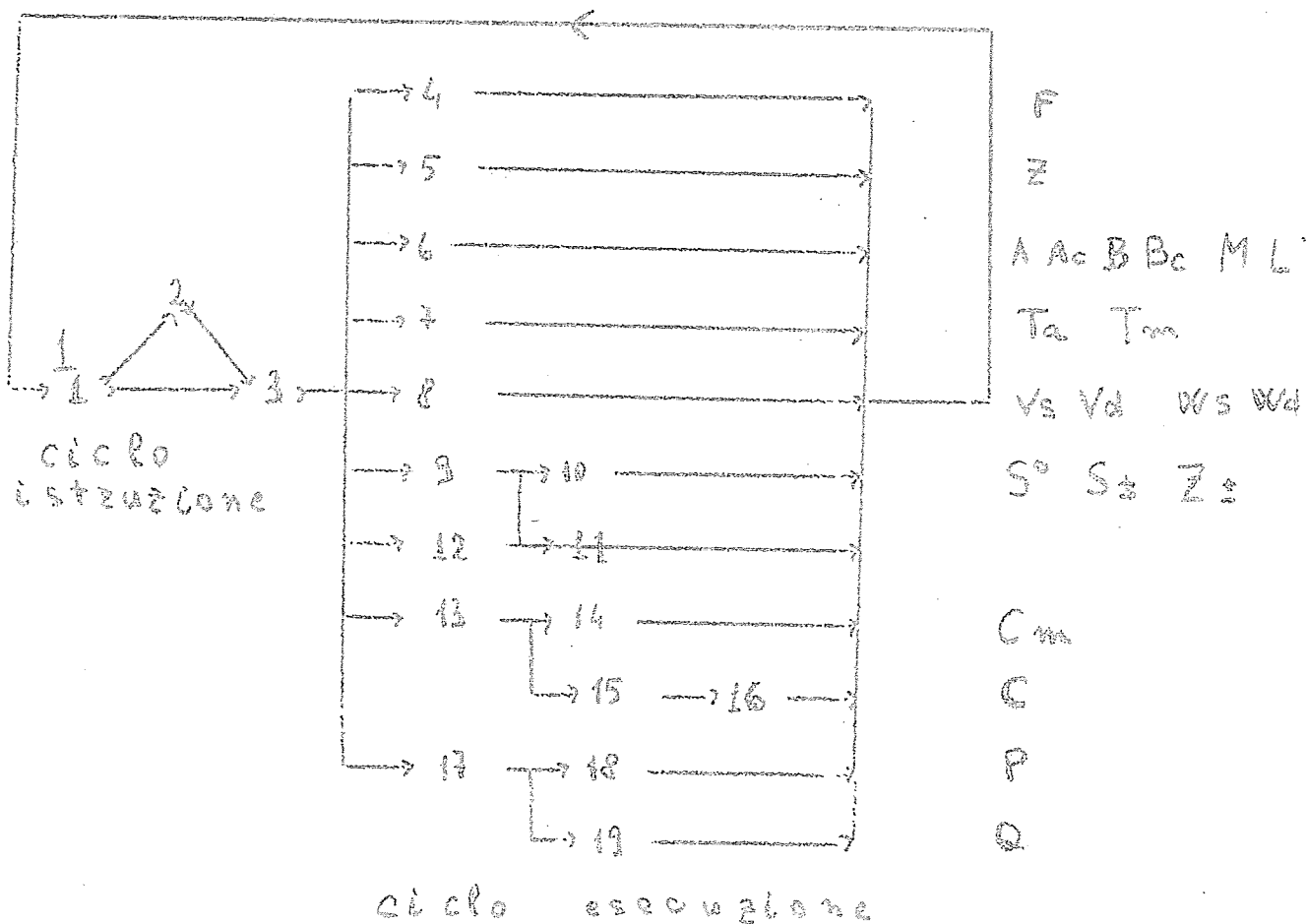
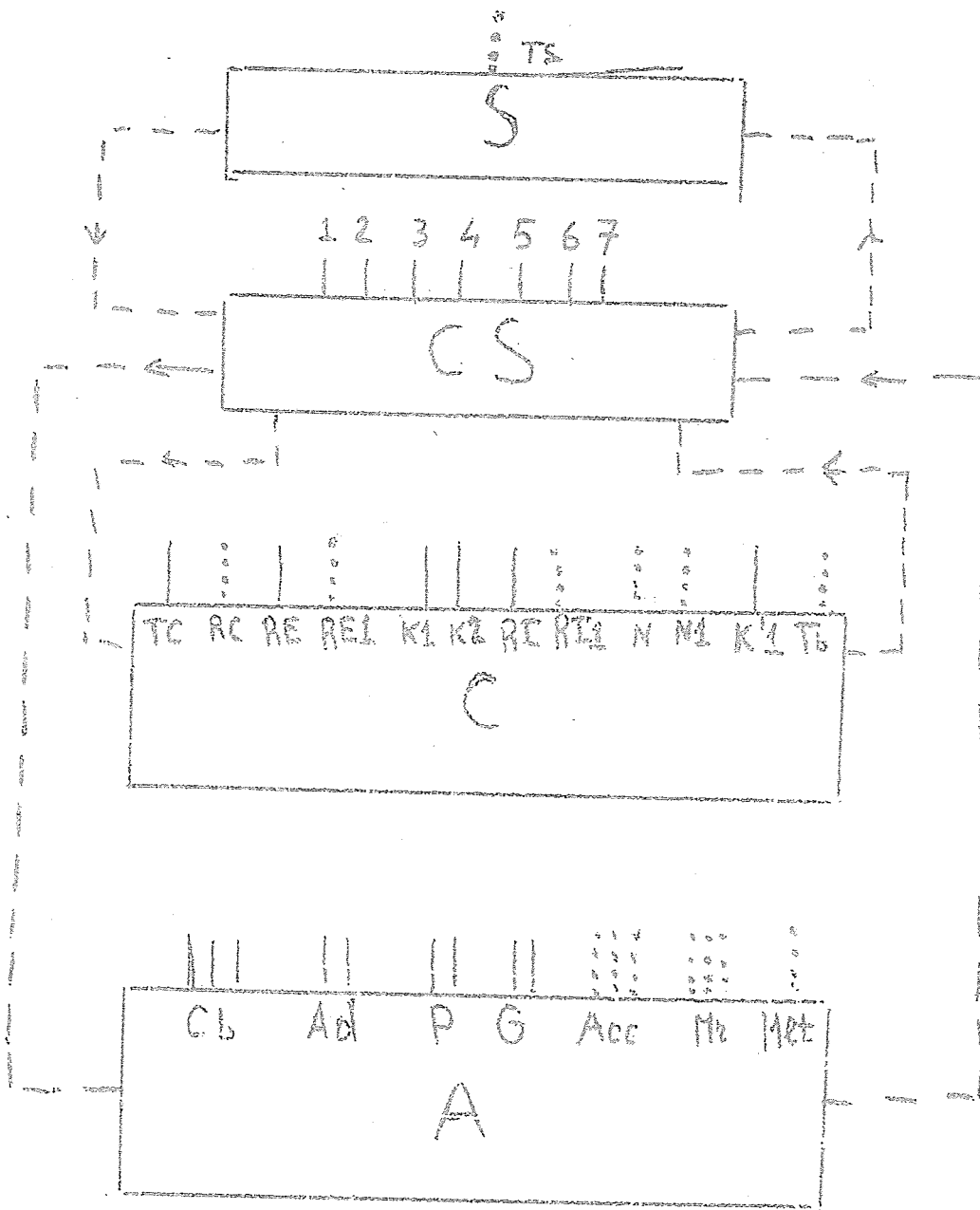


fig. 6

Per descrivere nel modo più chiaro lo svolgimento di tali fasi conviene riunire in un diagramma complessivo (Fig.7) lo schema della macchina segnando per ogni organo accanto alle linee di trasmissione delle informazioni anche le linee d'entrata e uscita dei diversi segnali e comandi. Per i riferimenti specifici si rimanda, ovviamente ai dettagli già forniti. Le linee di comando del controllo in realtà sono interne al controllo stesso, ma è parec conveniente indicarle perché ciò permette di completare la descrizione delle funzioni del Df il quale, istruzione per istruzione, manda gli opportuni segnali e comandi a tutti gli organi, compreso il controllo stesso.



Tutte le linee dei comandi e segnali sono da pensarsi uscenti dal controllo e più precisamente dal Decodificatore la cui funzione è appunto di predisporre i comandi e inviare i segnali per l'esecuzione delle varie fasi. Questi sono specificati nella tabella seguente. Nelle prime colonne di questa sono segnati i segnali di preparazione per le varie fasi, nell'ultime i comandi impulsivi che Df deve inviare (nell'ordine indicato, ma a tempi opportuni) per l'esecuzione. Ogni fase termina con il passaggio automatico a un'altra che è indicata per ogni riga da una freccia seguita dal numero d'ordine della fase che deve seguire. Per completezza sono state riportate anche le indicazioni relative alle fasi 1-2-3 già discusse. La descrizione delle fasi rimanenti, sarà agevolata appunto se si tiene presente ~~l'~~ l'intero quadro del ciclo operativo. Passiamo, ora, alla descrizione delle singole fasi.

4) - F- L'uscita degli ultimi 10 bit di RI è permanentemente connessa con l'entrata di N. Basta un comando di lettura  $N1^{\wedge}$  per effettuare il cambiamento di indirizzo. Non seguendo altri comandi, la macchina si ferma.

5) Z Come sopra ma con ritorno alla fase 1.

6) A, Ac, B, Bc, K, L, Tr. La descrizione delle combinazioni necessarie per l'esecuzione delle diverse operazioni aritmetiche è già stata fatta parlando dell'organo aritmetico. Esse sono riportate comunque integralmente nella tabella finale, ove compaiono anche i segnali per C.S. e K'. Questi sono eguali per tutte le istruzioni di questa classe, dovendo in sostanza semplicemente permettere la riscrittura dei dati letti nella memoria (CS 11///00) all'indirizzo indicato in R.I. ( $K^1=0$ ). Seguono i comandi in  $TS^{\wedge}$  (lettura memoria)  $AC1^{\wedge}$  (scrittura dell'accumulatore che viene così a conservare il risultato voluto), e il comando in  $N1^{\wedge}$  mediante cui il numeratore viene aumentato di 1 nell'ultima posizione, contenendo così l'indirizzo (semicella) dell'istruzione successiva. Prima di ciò nel corso di istruzioni che possono aver dato luogo a traboccamento un comando opportuno  $Tr^{\wedge}$  permette la conservazione di tale informazione per uso del controllo stesso (Istruzione  $S^{\circ}$ ). Fanno eccezione le istruzioni  $Trz$  in cui in luogo di  $AC1^{\wedge}$  segue  $Mr1^{\wedge}$  che serve per l'iscrizione del dato proveniente dalla memoria in  $Mr$  anziché in  $Acc$  e  $M$  (Moltiplicazione) in cui ai primi due comandi, che servono per la cancellazione di  $Acc$ , segue un comando  $Mlt^{\wedge}$  che mette in moto il meccanismo di controllo interno dell'organo aritmetico già descritto, per l'esecuzione del complesso ciclo di operazioni elementari richieste nella moltiplicazione stessa.

7) Ta, Tm-La combinazione (CS 00///00) è quella occorrente per la scrittura nella memoria all'indirizzo in RI ( $K^1=0$ )

Basta quindi predisporre il collegamento delle uscite di  $Acc$  (G/10) o di  $MR$  (G 01) ambedue normalmente chiuse (Coo), all'entrata della memoria.

Col comando in  $TS^{\wedge}$  avviene la scrittura. Con  $N1^{\wedge}$  avviene infine il passaggio all'istruzione successiva.

8) Vd, Vs, Wd, Ws. Tali istruzioni non impegnano la memoria. Per le traslazioni lunghe bisogna collegare i due registri (F 10) e (P 01).

	C.S.														TC		K		Cb.			Ad.				P.		G.						
	1	2	3	4	5	6	7	1	2	1	2	1	2	3	1	2	1	2	1	2	1	2	1	2	1	2								
1)	1	1	ù	u	0	0	0	/	/	/	/	1	/	/	/	/	/	/	/	/	/	/	/	/	/	/	TS^	RE^	B^	0				
2)	u	ù	0	0	1	u	ù	0	1	0	1	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	TS^	RE^	RC^	B^	0			
3)	/	/	/	/	/	/	/	0	1	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	RC^	RI^	DF^					
4) F	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	N1^							
5) Z	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	N1^							
6) A	1	1	/	/	/	0	0	/	/	/	0	1	0	0	1	0	/	/	0	0	TS^	Acc1^	Tb^				N1^							
Ac	1	1	/	/	/	0	0	/	/	/	0	1	0	0	0	0	/	/	0	0	TS^	Acc1^					N1^							
B	1	1	/	/	/	0	0	/	/	/	0	0	1	0	1	1	/	/	0	0	TS^	Acc1^	Tb^				N1^							
Bc	1	1	/	/	/	0	0	/	/	/	0	0	1	0	0	1	/	/	0	0	TS^	Acc1^					N1^							
L	1	1	/	/	/	0	0	/	/	/	0	0	0	1	0	0	/	/	0	0	TS^	Acc1^					N1^							
Ts	1	1	/	/	/	0	0	/	/	/	0	1	0	0	0	0	/	/	0	0	TS^	Mr1^					N1^							
M	1	1	/	/	/	0	0	/	/	/	0	0	0	0	0	0	/	/	0	0	TS^	Acc1^	Mlt				N1^							
7) Ta	0	0	/	/	/	0	0	/	/	/	0	/	/	/	/	/	/	/	1	0	TS^						N1^							
Tm	0	0	/	/	/	0	0	/	/	/	0	/	/	/	/	/	/	/	0	1	TS^						N1^							
8) Vd	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	0	0	Acc2	h				N1^							
Vs	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	0	0	Acc3	h				N1^							
Wd	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	1	0	0	0	Acc2	h	Mr2	h		N1^						
Ws	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	0	1	0	0	Acc3	h	Mr3	h		N1^						
9) S°	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	Legge	Tb	o	Sa	=	0						1		
10) "	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	1						N1^						1	
11) "	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/							N^							
12) Z	0	1	1	1	0	0	0	0	0	0	1	/	/	/	/	/	/	/	/	/	N1^	TS^	RE^	RC^	Ss=	0						1		
13) C, Cm	0	1	1	1	0	0	0	0	0	1	/	/	/	/	/	/	/	/	/	/	N1^	TS^	RE^	RC^	C							1		
14) Cm	0	0	1	1	0	0	0	1	/	/	1	/	/	/	/	/	/	/	/	/	TS^						N1^							
15) C	0	0	/	/	1	/	/	0	0	1	/	/	/	/	/	/	/	/	/	/	RC^													
16) "	0	0	/	/	1	/	/	1	0	1	/	/	/	/	/	/	/	/	/	/	RC^							N1^						
17) P, Q	0	0	/	/	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	RE^				P									
18) P	/	/	0	1	/	1	0	0	0	0	/	/	/	/	/	/	/	/	/	/	TC^	TS^	RE1^	½RI1^	N1^									
19) Q	1	0	0	1	0	0	0	1	/	/	1	/	/	/	/	/	/	/	/	/	TS^	TC^	RE1^	½RI1^	N1^									

N.B. Nelle prime righe della presente tabella é stato scritto u e ù per indicare l'ultimo bit del numeratore N e la sua negazione.



Bastano direttamente i segnali di traslazione ai registri aritmetici per un tempo  $\hat{n}$  proporzionale al numero di posti di cui si deve traslare il contenuto dei registri stessi. Con  $N1^{\wedge} \text{---} 1$  si passa all'istruzione successiva.

9/10/11)  $S^{\circ} S^{+}$  Salti condizionati sull'unità aritmetica. Il Df esamina, a seconda dei casi, l'indicatore di traboccamento o del segno di Acc. In base al loro contenuto (0 o 1) passa all'esecuzione della fase 10): procedere all'istruzione successiva ( $N1^{\wedge} \text{---} 1$ ), oppure della fase 11): saltare all'istruzione contenuta in N ( $N^{\wedge} \text{---} 1$ ).

12) Z - Salto sui R:C. E' una delle istruzioni doppie, che richiede un'intera parola. E' necessario infatti specificare il registro di controllo sul cui segno va effettuato il salto. Tale indirizzo  $\acute{e}$  contenuto nella seconda met\`a della parola, che bisognerà chiamare dalla memoria e inviare in RE. Ciò si ottiene con la combinazione indicata e con la seguente serie di comandi:  $RE^{\wedge}$  (selezione della seconda met\`a della parola)  $TS^{\wedge}$  (lettura memoria),  $RE^{\wedge}$  (scrittura nel registro di entrata),  $RC^{\wedge}$  (lettura del registro di controllo selezionato). In base al segno, come per la fase 9 precedentemente descritta, il controllo passa all'esecuzione della fase 10) o della fase 11).

13) C, Cm. Anche queste sono istruzioni doppie: la preparazione e la esecuzione di tale fase coincide con la 12) precedentemente descritta. Per l'istruzione Cm si passa a 14); e per l'istruzione C si passa a 15-16).

14) Cm (cont.) Si tratta di trasferire l'istruzione contenuta nella seconda met\`a della cella il cui indirizzo  $\acute{e}$  in RI nel registro di controllo il cui indirizzo  $\acute{e}$  in RE. La predisposizione  $\acute{e}$  quella della fase precedente, eccetto il segnale 1 per TC (disporre il registro di controllo a scrivere).

L'esecuzione avviene mediante comandi  $TS^{\wedge}$  (leggere la memoria),  $RC^{\wedge}$  (scrivere nel registro di controllo).  $N1^{\wedge} \text{---} 1$  conduce infine alla istruzione successiva.

15-16) C (cont.) Si tratta di sommare il numero XY contenuto negli ultimi 10 bit di RI al contenuto del registro di controllo preselezionato dalle preme 5 cifre di RE.

Nella fase 15) si esegue tale somma mediante lettura dei RC ( $RC^{\wedge}$ ) dopo opportuna predisposizione delle connessioni interne del controllo.

Nella fase 16) si esegue in sostanza la riscrittura nella stessa cella dell'informazione modificata.

17) P,Q. In preparazione di tali istruzione, basta sostanzialmente cancellare RE, di modo che si comincia col selezionare il registro di indirizzo (0000). La cancellazione si effettua isolando il registro RE e inviando un comando di lettura in  $RE^{\wedge}$ .

18) P (cont.) Si tratta di trasferire successivamente le 16 mezze parole contenute in RC, in 8 celle consecutive della memoria principale a partire dalla prima metà di quella il cui indirizzo è indicato in RI. Ciò si ottiene mediante un ciclo ripetuto 16 volte di lettura di RC (RC), scrittura di S (TS<sup>+</sup>), aumento degli indirizzi in RI (RI1<sup>+</sup>) e in RE (RE 1<sup>+</sup>, cioè una volta sì e una no). Il segno posto sopra le combinazioni CS 3,4,6, indica che tali combinazioni vanno invertite ad ogni ciclo permettendo così di scrivere alternativamente nella prima e nella seconda metà di S. Terminato il ciclo N1<sup>+</sup> 1 conduce alla istruzione successiva.

19) Q (cont.) È l'operazione inversa della precedente e ne differisce quindi sia per la preparazione (leggere S e scrivere in Rc) sia perché nel ciclo (pur esso ripetuto 16 volte) il comando di lettura della memoria TS<sup>+</sup> deve precedere quello di scrittura in RC, RC<sup>+</sup>.