

ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Bari

INFN/AE-77/12
15 Settembre 1977

M. DeBlasi, D. Marino, E. Meroni, P. G. Rancoita e S. Sala:
ANALISI DELLE CONDIZIONI OTTIMALI DI IMPIEGO DI
MEMORIE PER LA SELEZIONE PROGRAMMABILE DI
"TRIGGERS" IN PRESENZA DI NUMERO ELEVATO DI
SEGNALI.

INFN/AE-77/12
15 Settembre 1977

M. De Blasi, D. Marino, E. Meroni^(x), P. G. Rancoita^(x) e S. Sala^(x):
ANALISI DELLE CONDIZIONI OTTIMALI DI IMPIEGO DI MEMORIE
PER LA SELEZIONE PROGRAMMABILE DI "TRIGGERS" IN PRESEN
ZA DI UN NUMERO ELEVATO DI SEGNALI.

1. - INTRODUZIONE.

E' ormai abbastanza frequente negli esperimenti di Fisica dover combinare un numero sempre più elevato di segnali provenienti da ri velatori per generare i necessari segnali "trigger" da inviare agli strumenti di misura. L'impiego di memorie ad accesso casuale (RAM) è stato recentemente proposto (A. Fucci et al. report CERN 19 Jan. 1977) per un insieme di ragioni pratiche, tra esse collegate; in particolare: a) semplificare le operazioni manuali di interconnessione dei vari circuiti di coincidenza ed anticoincidenza (coi quali normalmente si realiz zano le volute funzioni di decisione); b) minimizzare le probabilità di errore; c) avere la possibilità di controllare rapidamente - con l'aiuto di un calcolatore - che siano verificate le assegnate relazioni di carat tere logico tra i segnali provenienti dai rivelatori.

L'uso delle RAM non è nuovo nei problemi di riconoscimento di forme, di "information retrieval" e simili, cui concettualmente si ri conduce il tipo di impiego anzidetto; ma è solo con i più recenti dispo sitivi RAM realizzati con tecnologia ECL che è possibile ottenere tempi di risposta dell'ordine dei 20+30 ns, normalmente sufficienti per produrre segnali di "trigger" utilizzabili.

Tuttavia, poichè la necessità di realizzare a programma determi nate funzioni di decisione e di controllarle con calcolatori si pone real mente solo quando il numero di segnali da combinare è elevato, ci sem bra opportuno analizzare quali sono le condizioni di impiego che ottimiz zano la utilizzazione delle memorie, sia in termini di ingombro che di

(x) - Istituto di Fisica dell'Università di Milano, e INFN Sezione di Milano.

tempi di risposta ed economicità. Risultato di queste analisi sono alcune proposte di schemi che soddisfano le condizioni dette.

2. - GENERAZIONE DI "TRIGGERS" MEDIANTE RAM: DISCUSSIONE.

Supponiamo per semplicità che i segnali provenienti dai rivelatori siano solo tre: A, B, C. L'estensione al caso di 10 linee di ingresso è diretta e facilmente realizzabile con le attuali memorie ECL 1024x1. Il problema dell'estensione in generale sarà considerato nel prossimo paragrafo. Le possibili configurazioni dei segnali all'ingresso del generatore di "trigger" sono 2^3 e ciascuna di esse può essere impiegata per indirizzare una delle 8 locazioni di una ipotetica RAM 8x1. Se ad es. (Fig. 1a) all'indirizzo (010) è stato scritto precedentemente un 1 logico,

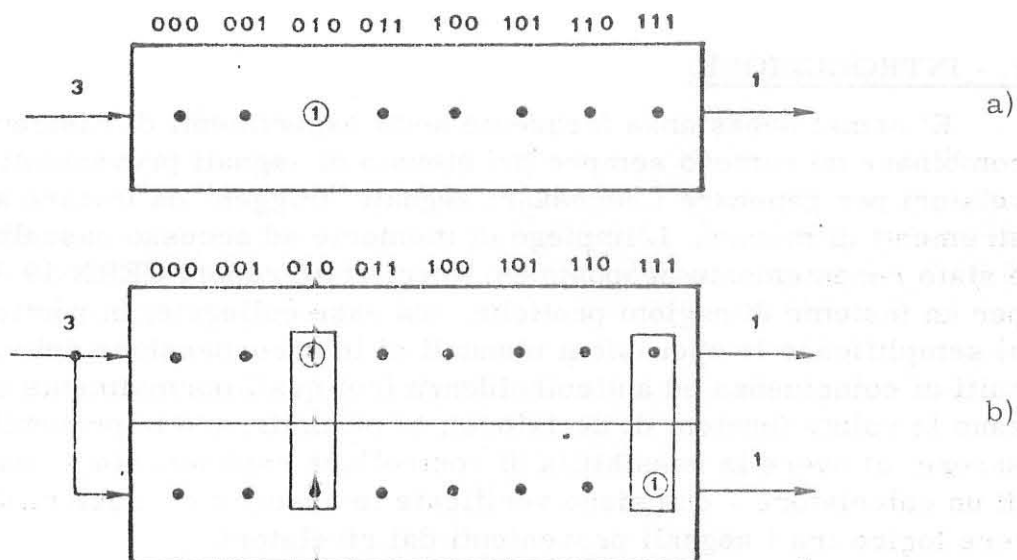


FIG. 1

ogni volta che all'ingresso della memoria (cioè sulle linee di indirizzo) si presenterà la "configurazione di segnali $C_2=(010) \equiv \bar{A}B\bar{C}$, alla uscita ci sarà un segnale 1 che potrà essere utilizzato come trigger, se effettivamente è C_2 la configurazione d'ingresso che rappresenta la relazione logica voluta tra i tre segnali utilizzati e purchè tutte le restanti locazioni interne contengano uno zero logico. Se si è interessati a due delle otto configurazioni possibili di ingresso, ad es. $C_2=(010)$ e $C_7=(111)$, basterà aver scritto 1 in corrispondenza di entrambi gli indirizzi (010) e (111). Questa soluzione produrrà sull'unica linea di uscita un trigger ogni volta che si presenta all'ingresso la configurazione C_2 o la configurazione C_7 , cioè ogni volta che sia soddisfatta tra le configurazioni di ingresso la relazione logica $C_{2,7}=C_2+C_7$.

Questo è il modo più semplice di identificare relazioni logiche (tra i segnali di ingresso) nella forma canonica "somma di prodotti" o duali.

Ma in generale si è interessati a non perdere la corrispondenza tra "trigger" e la particolare configurazione C_i dei segnali di ingresso che lo ha determinato: in questo caso si possono usare moduli di memoria con la lunghezza di parola maggiore (cioè più moduli con lunghezza di parola di 1 bit indirizzati in parallelo con gli stessi segnali). Si potranno così ottenere su linee distinte i "triggers" voluti. Se ad es. si considerano due moduli paralleli di formato 8x1, Fig. 1b, scrivendo agli indirizzi C_2 e C_7 rispettivamente le parole di due bits (10) e (01) si riesce a separare sulle due linee di uscita i "triggers" dovuti a C_2 da quelli dovuti a C_7 .

Adottando disposizioni di questo tipo, il numero di moduli impiegati è uguale al numero di differenti linee di trigger richieste.

Una migliore utilizzazione della memoria si ha introducendo (come ad es. proposto nel citato lavoro) un decodificatore. In questo modo il numero di moduli è ridotto al \lg_2 del numero di trigger diversi. Però, come si vede dalla breve analisi che segue, se non si opera correttamente, anche in questo caso si perde in generale la univocità nella identificazione del trigger e, comunque, le linee effettivamente utilizzabili non sono 2^N ma $2^N - 1$. Dalla Fig. 2a si vede ad

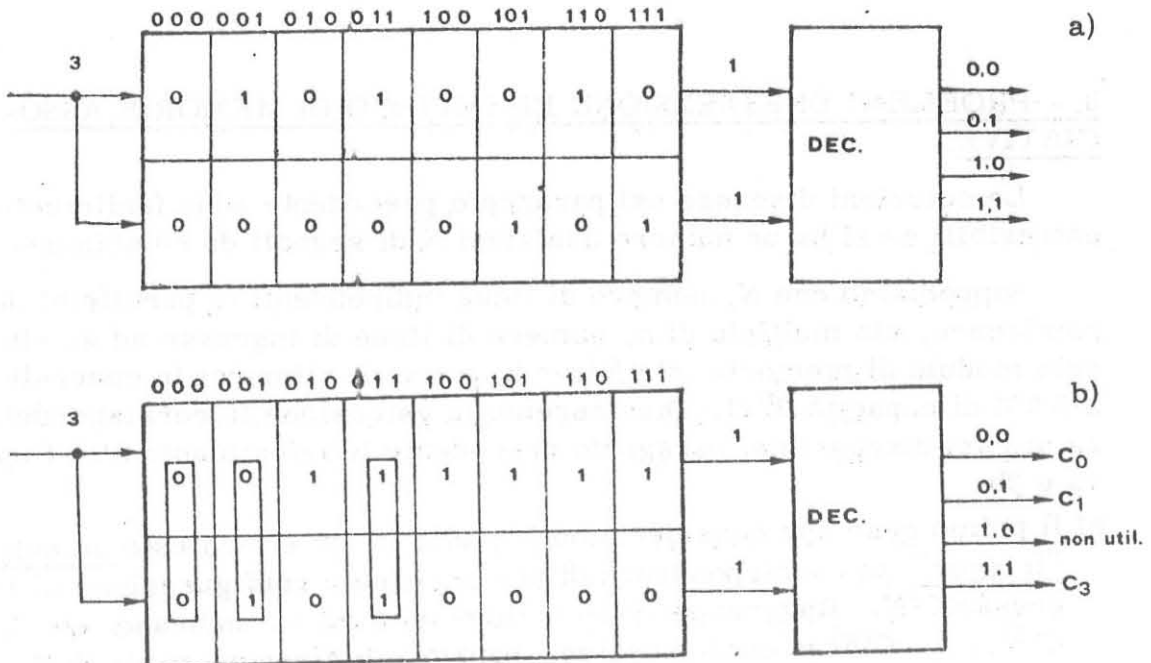


FIG. 2

es. che la linea di uscita (0, 0) è attiva in corrispondenza delle tre configurazioni di ingresso $C_0=(000)$, $C_2=(010)$ e $C_4=(100)$; la linea (0, 1), in corrispondenza delle configurazioni $C_5=(101)$ e $C_7=(111)$; la linea (1, 1) non viene mai attivata e la linea (1, 0) è attivata in corrispondenza delle configurazioni $C_1=(001)$, $C_3=(011)$ e $C_6=(110)$. Perchè la univocità voluta sia preservata, occorre rispettare le seguenti condizioni:

a) in corrispondenza degli indirizzi prescelti vanno scritte parole sempre differenti (se si vuole, anche a caso), ed invece in corrispondenza di tutte le altre configurazioni non di interesse va scritta sempre una identica parola differente dalle precedenti; b) se i moduli di memoria indirizzati in parallelo sono N, si potranno utilizzare non più di $2^N - 1$ linee di triggers emergenti dal decodificatore, in quanto una della 2^N possibili parole differenti (e quindi la corrispondente uscita del decodificatore) va riservata a riconoscere tutte le configurazioni di "non interesse". In riferimento al caso illustrato in Fig. 2a, se ad es. sono di interesse le configurazioni C_0, C_1, C_3 , per utilizzare correttamente questo schema si dovrà scrivere (vedi Fig. 2b) in ciascuno degli indirizzi corrispondenti tre parole differenti, ad es. in $C_0=(000)$ scrivere (0,0), in $C_1=(001)$ scrivere (0, 1), in $C_3=(011)$ scrivere (1, 1) ed in tutte le altre la restante parola (1, 0).

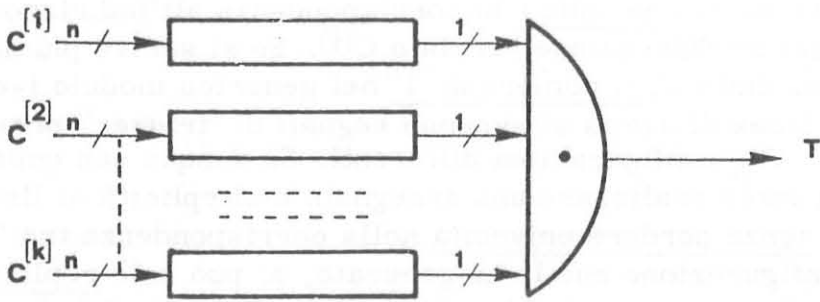
Un limite pratico all'aumento di linee di trigger con questa tecnica è che la complessità del decodificatore cresce esponenzialmente col numero dei segnali da decodificare.

3. - PROBLEMI DI ESTENSIONE ED IMPIEGO DI MEMORIE ASSOCIATIVE.

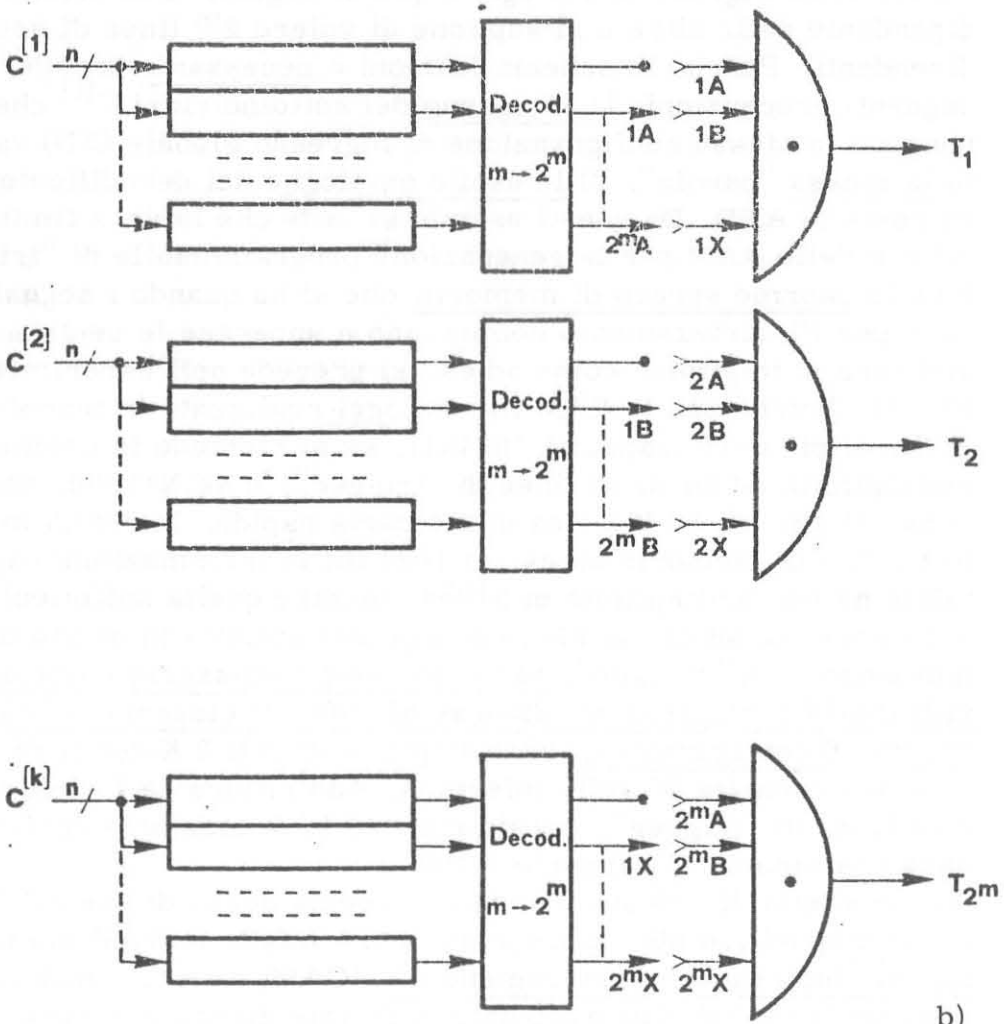
Le soluzioni discusse nel paragrafo precedente sono facilmente estensibili se si ha un numero qualsiasi N di segnali da combinare.

Supponiamo che N, numero di linee indipendenti (e parallele) da combinare, sia multiplo di n, numero di linee di ingresso ad un singolo modulo di memoria: sia $N/n = k$; potremo riferirci in generale a RAM di capacità $2^n \times 1$. Discuteremo la estensione, in corrispondenza ai casi discussi nel paragrafo precedente, in riferimento alle Figg. 1a e 2b.

a) Il primo caso che consideriamo è quello in cui è richiesto un solo "trigger" in corrispondenza di una assegnata configurazione di ingresso $C(N)$. Raggruppando le N linee ad n ad n e indicando con $C^{(1)}, C^{(2)}, \dots, C^{(k)}$ le configurazioni parziali di ciascuna n-pla di fili all'ingresso di ogni modulo RAM (in modo tale che la concatenazione della k configurazioni $C^{(i)}$ rappresenti globalmente $C(N)$) il trigger voluto è ovviamente ottenibile come prodotto logico dei "trigger" corrispondenti alle configurazioni parziali $C^{(1)}, C^{(2)}, C^{(k)}$, come è mostrato in Fig. 3a. In ogni modulo di memoria



a)



b)

FIG. 3

deve essere scritto un solo 1 in corrispondenza all'indirizzo associato ad ogni configurazione parziale $C^{(i)}$. Se si scrive più di un "1" per modulo, detto U_i il numero di "1" nel generico modulo i -esimo sull'unica linea di uscita si avranno segnali di "trigger" provenienti da $U_1 \cdot U_2 \cdot \dots \cdot U_k$ configurazioni differenti. Se dunque con questo schema si vuole realizzare una assegnata molteplicità di linee di "trigger" senza perdere univocità nella corrispondenza tra "trigger" e configurazione che lo ha generato, si può solo replicare lo schema dato costruendo blocchi identici di RAM, del tipo in Fig. 3a, programmando ovviamente ogni blocco di RAM in modo da riconoscere una sola delle configurazioni volute.

- b) Il modo generalizzato per ottenere una certa molteplicità di linee di "trigger" ognuna associata univocamente ad una sola configurazione di ingresso, come estensione dello schema in Fig. 2b, è mostrato nella Fig. 3b, in cui ogni n -pla di segnali di indirizzo è indipendente dalle altre e si suppone di volere 2^m linee di uscita indipendenti. Perchè lo schema funzioni è necessario prendere le seguenti precauzioni: 1) ad ognuno dei sottoindirizzi $C^{(i)}$ che compongono la stessa configurazione di ingresso globale $C(N)$ va scritta la stessa "parola"; 2) le uscite omologhe dei decodificatori vanno poste in AND. Da questi esempi si vede che la vera limitazione all'uso delle RAM per la generazione programmabile di "trigger" è nello enorme spreco di memoria che si ha quando i segnali da usare per l'indirizzamento cominciano a superare le centinaia o addirittura le migliaia, come ad es. si prevede nell'esperimento FRAM. Poichè per le RAM fino ad oggi realizzate in tecnologia ECL, al più $n=10$ (capacità 1024×1), se si richiede in uscita una molteplicità ad es. di 16 linee di "trigger", e se $N=1000$, sono necessari almeno 50 K-bytes di memoria rapida. In realtà in un solo modulo di memoria ad es. di 1024 bit la informazione rappresentabile ha una molteplicità di 2^{1024} , mentre quella sufficiente è molto minore: se ad es. si richiede una molteplicità in uscita di 16 linee indipendenti di "trigger", tutto ciò che è necessario e basta per individuare 16 configurazioni diverse di 1000 bit ciascuna è "ricordare" queste 16 configurazioni, cioè disporre di soli 2 K-bytes di memoria, cioè una capacità 25 volte inferiore. Addirittura se è richiesta una sola linea di "trigger", questo risparmio di memoria arriva, nel caso esaminato, al rapporto 1:100.

Nei problemi di "riconoscimento", specie quelli di queste dimensioni, il dispositivo più appropriato non è infatti la RAM ma una memoria "indirizzabile per contenuto" (CAM, "content addressable memory") cioè di tipo associativo. Di tale dispositivo esistono in commercio (dal 1970 circa) diversi modelli, ad es. il tipo 93402 Fairch. (formato 4×4), il tipo 8220 Sign. (formato 4×2), l'INTEL 3104 (4×4) etc... Con soli 8 moduli del tipo 93402 Fairch. ad es. si possono assemblare (con piccole varianti a schemi che gli stessi costruttori allegano alle specifiche del componente) dei blocchi di memoria 8×16 che su 16 linee distinte di uscita danno un segnale di

"trigger" ogni volta che sulle 8 linee di indirizzo si presenta una di 16 differenti configurazioni di 8 bit tra le 256 possibili. Su una sola scheda CAMAC dunque può essere alloggiata la logica di identificazione di una assegnata configurazione anche se le linee indipendenti di ingresso, provenienti dai rivelatori sono $8 \div 100$. I tempi di "decisione" sono compresi tra 20 ed i 30 ns.

4. - OTTIMIZZAZIONE DEI TEMPI DI RISPOSTA.

A questo punto è chiaro che l'ingombro maggiore è dato ormai non tanto dai componenti attivi quanto dalle dimensioni stesse degli attacchi (ad es. tipo LEMO) delle linee che portano i segnali dai rivelatori. Poichè in genere su una singola unità CAMAC non possono essere alloggiati più di $12 \div 16$ connettori di ingresso (a meno di non usare linee bifilari ed elettronica ausiliaria con ingressi differenziali) non è necessaria la compattezza ottenibile con i componenti anzidetti ma è possibile l'uso di memorie associative costruite "ad hoc" con componenti a bassa scala di integrazione (ad es. registri ad 8 bit del tipo SN74100 (Texas) ed elementi di identificazione 4×2 "Ex-NOR", abbastanza comuni). Un grande vantaggio di questa soluzione è, come vedremo, quello di minimizzare il tempo di decisione, riducendolo a valori del nanosecondo, per quanto grande sia il tempo di accesso ai registri. Sono possibili vari schemi di principio, di cui in particolare sembrano interessanti i due di Fig. 4a e 4b, del tutto equivalenti in quanto a tempi di decisione ed a numero di componenti usati (16 registri a 8 bit e 32 (4×2) - "Ex. NOR". In entrambi gli schemi la capacità totale della memoria impegnata ha il valore concettualmente minimo, cioè quello che basta a "ricordare" le configurazioni da riconoscere. Nel primo schema (Fig. 4a) ci sono 128 linee indipendenti (parallele) di ingresso ed un unico "trigger" di uscita: se si richiede una molteplicità di linee di "trigger" maggiore di 1, lo schema va semplicemente replicato tante volte quante sono le configurazioni di "trigger" da indentificare sul totale di 2^{128} . Lo spazio necessario ad ospitare questa elettronica è verosimilmente una scheda CAMAC standard (al più due). Nel secondo schema (Fig. 4b) si hanno in ingresso solo 8 linee indipendenti e 16 linee di "trigger" all'uscita, per 16 volute configurazioni tra le 256 possibili. L'estensione ad un numero qualsiasi di linee si può fare con le stesse modalità e con gli stessi accorgimenti discussi per il caso di Fig. 3b, ove i vari segnali da combinare in AND sono gli omologhi T_1, T_2, \dots, T_{16} ottenuti da ogni blocco di 8 linee di ingresso. La particolarità essenziale di questi schemi è che, una volta scritte nei vari registri le

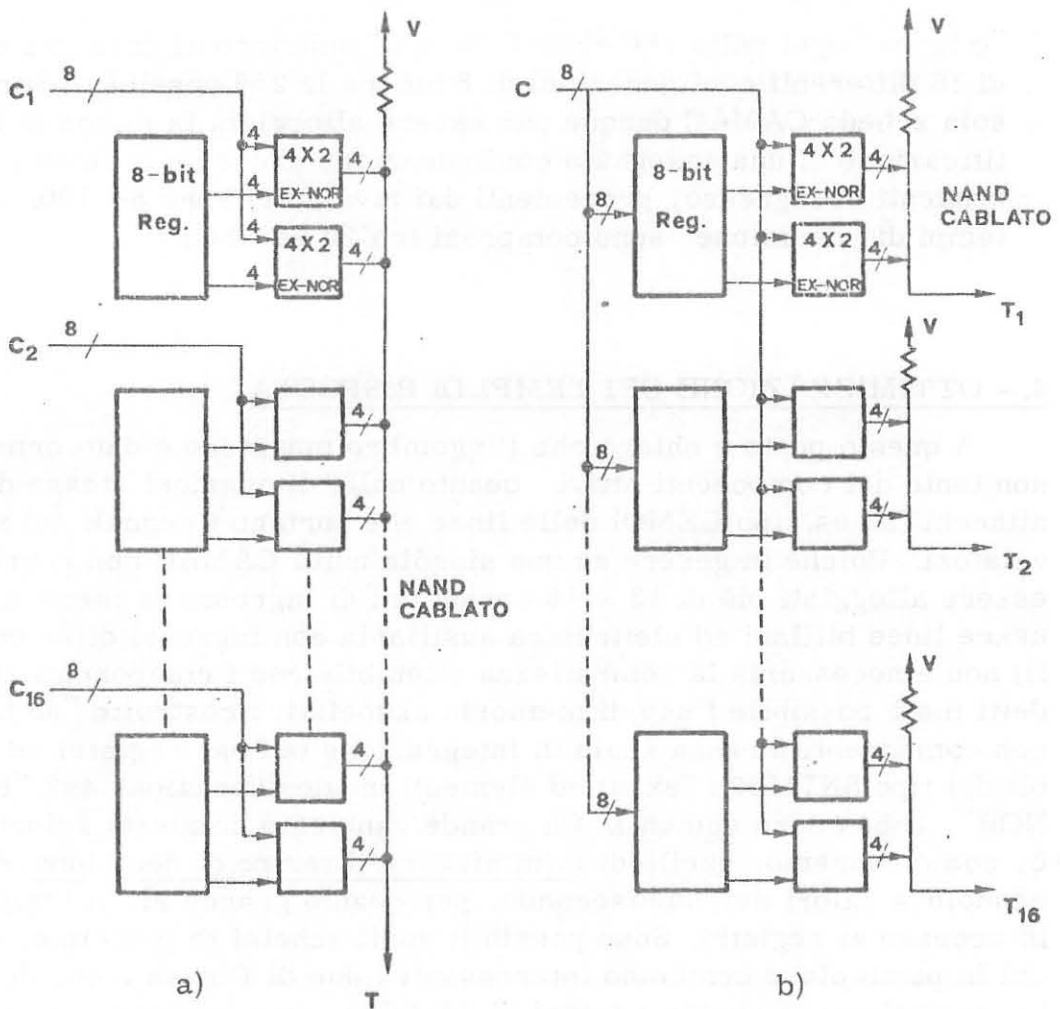


FIG. 4

configurazioni da ricordare, il tempo di decisione dipende esclusivamente dal ritardo introdotto dagli elementi di identificazione (4x2 "Ex. NOR") che può variare dai 0.7 ns. (ad es. per il tipo 100107 Fairch.) ai 3 ns. (ad es. per il tipo 10113 Fairch.). Da ciò si vede che i ritardi significativi sono in pratica solo quelli introdotti dalle linee esterne di connessione. Essendo inoltre minimizzato il n. di loca_zioni da "scrivere" e "leggere" (per controlli), i tempi globali di scrittura e controllo dei registri sono trascurabili in confronto a quelli necessari con memorie RAM.

5. -CONCLUSIONI.

E' stata mostrata la convenienza e la realizzabilità di "trigger" programmabili impieganti memorie associative costruite con componenti a media o piccola scala di integrazione, quando il numero di segnali provenienti dai rivelatori è grande. Con gli schemi indicati, sono minimizzati:

- la capacità totale di memoria necessaria;
- il tempo di decisione (circa 1 ns.);
- il tempo di programmazione e controllo;
- il costo (tenuto conto anche che le memorie possono essere lente).

L'ingombro resta limitato dalle sole dimensioni degli attacchi dei cavi che portano i segnali da combinare; l'ingombro della sola elettronica è tale da poter pensare di utilizzare come conduttori di ingresso cavi bifilari differenziali (con elettronica ausiliaria di ricezione) e connettori ad alta densità di contatti (dello standard CAMAC).