



Master II Livello 2006

NASO GIOVANNI

Progettazione della testabilita' (DFT) di memorie FLASH

INDICE

- 1) Scopi
 - 1a. resa
 - 1b. affidabilita'
 - 1c. manufatturabilita'
 - 1d. configurabilita'
 - 1e. Debug

- 2) Caratteristiche
 - 2a. osservabilita' (tensioni, frequenze, distribuzioni)
 - 2b. controllabilita' (tensioni, durate, frequenze, sequenze, architetture)
 - 2c. tempi di esecuzione (compressione, parallelismo, self test)

- 3) Organizzazione
 - 3a. fusibili
 - 3b. attivazione della modalita' di test (test mode entry)
 - 3c. organizzazione DFT modale
 - 3d. organizzazione DFT a registri

- 4) Ridondanza
 - 4a. il circuito di match
 - 4b. ridondanza di riga
 - 4c. ridondanza di colonna
 - 4d. ridondanza di blocco

Micron[®]

in partnership with



Università degli Studi dell'Aquila

- 5) Trims
 - 5a. calibrazione di un termometro
 - 5b. aggiustamento della frequenza di un oscillatore
 - 5c. trim della tensione di word line in un algoritmo di program
 - 5d. trim della durata di impulsi
- 6) Algo skip
- 7) Monitor e forzamento di tensioni
- 8) Forzamento delle durate di impulsi
- 9) Accesso diretto in array
- 10) Modalita' di stress
- 11) Tecniche di compressione
 - 11a. compressione di word
 - 11b. verifica interna (IVR)
- 12) Tecniche di parallelizzazione
 - 12a. SED con IVR
 - 12b. SED con interruzione
- 13) Monitor di algoritmi



1) SCOPI

Una autonoma branca di progettazione nel settore delle memorie FLASH, detta design for testability (DFT), si occupa di introdurre nel dispositivo di memoria le circuiterie atte a garantire :

- 1a. La riparazione di celle di memoria guaste sostituendole con strutture di memoria ridondanti. La ridondanza puo' essere di riga, colonna o blocco. L'obiettivo e' di aumentare la resa di produzione (yield).
- 1b. L'individuazione di debolezze di processo, essenzialmente relative alla integrita' degli ossidi, in modo tale da riparare o scartare memorie che nel tempo potrebbero presentare degradazioni tali da inficiarne il funzionamento. L'obiettivo e' di aumentare l'affidabilita' di funzionamento (reliability).
- 1c. La calibrazione di alcune parti di circuito quali per esempio riferimenti di tensione interna o sensori interni di temperatura o tensioni interne di programmazione in modo tale da avere memorie con prestazioni il piu' uniformi possibile. L'obiettivo e' di aumentare la manufatturabilita' dei dispositivi.
- 1d. La configurazione di alcuni circuiti, come ad esempio il parallelismo degli I/O o la capacita' di drive degli outputs o il funzionamento a singolo o multilivello, in modo da soddisfare specifiche necessita' del cliente. L'obiettivo e' di aumentare la flessibilita' del prodotto per meglio ottemperare alle specifiche esigenze dell'applicazione.
- 1e. L'individuazione di malfunzionamenti nelle circuiterie di gestione dell'array di memoria in modo tale da rilevare errori nella progettazione elettronica/logica dei circuiti. L'obiettivo e' di individuare con facilitata' e rapidita' gli errori di progettazione (bugs) e avere rapidi tempi nel raggiungimento della produzione di massa (ramp-up).



2) CARATTERISTICHE

Affinche' la design for testability di una memoria FLASH assolve con efficacia gli scopi Elencati deve essere progettata in modo da soddisfare al meglio le seguenti caratteristiche :

2a. Garantire l'osservabilita' del chip, ovvero :

- poter misurare dall'esterno le tensioni usate nelle operazioni di lettura, programmazione e cancellazione della memoria.
- poter misurare dall'esterno frequenze operative nella generazione di tensioni o nelle temporizzazione degli algoritmi.
- poter misurare dall'esterno tensioni ausiliarie di riferimento diverse dalla tensione di alimentazione.
- poter misurare dall'esterno parametri elettrici atti a calibrare sensori interni di temperatura.
- poter misurare dall'esterno valori delle distribuzioni delle tensioni di soglia VT a cui si sono portate le celle dell'array dopo una operazione di programmazione o di cancellazione o di disturbo o di stress.
- poter misurare dall'esterno l'evoluzione di un algoritmo di programmazione o di cancellazione. Infatti tali algoritmi sono adattativi e non si puo' predire a priori la loro evoluzione.

L'osservabilita' estensiva del funzionamento di una memoria a scopi di calibrazione, caratterizzazione o debug puo' essere effettuata con il metodo dell' internal probing ovvero con l'uso di sonde passive o attive che vengono poggiate sulle metallizzazioni interne del chip prima del deposito della passivazione oppure rimuovendo la passivazione. Il metodo dell'internal probing ha comunque delle limitazioni: a) richiede di effettuare un layout con la presenza di test points di metallo sufficientemente grandi da permettere il contatto di una sonda b) non permette di esaminare quei problemi che si manifestano solo in presenza di passivazione o delle plastiche di impacchettamento c) richiedono una azione manuale di manovra di una sonda d) non permettono l'uso contemporaneo di piu' di 2/3 sonde.



La design for testability di una memoria FLASH invece si occupa della osservabilita' dall'esterno, cioe' attraverso gli I/O pads del chip, permettendo quindi di osservare il chip attraverso una interfaccia utente direttamente tramite la macchina di test anche in presenza di chips passivati o gia' in package.

2b. Garantire la controllabilita' del chip, ovvero :

- poter forzare dall'interno o dall'esterno tensioni usate nelle operazioni di lettura, programmazione o cancellazione. I forzamenti dall'esterno sono particolarmente utili quando si vogliono effettuare operazioni che richiedono tensioni operative con range, forma o capacita' di drive diverse da quelle ottenibili da generatori interni, pur provvisti di una loro possibilita' di configurazione.
- poter forzare dall'interno o dall'esterno clocks usati nei generatori o nel sequenziamento di algoritmi ed influenzare la capacita' di corrente dei generatori o i margini operativi del controllore di algoritmi.
- Poter forzare dall'esterno particolari condizioni di polarizzazione delle celle di memoria in modo da effettuare delle operazioni di stress ed accelerare la fase di fallimenti iniziali (EFR = early failure rate). Per esempio tutte le word lines di un settore abilitate e connesse ad una stessa tensione con il substrato dello stesso settore a massa.
- Poter influenzare l'evoluzione di un algoritmo di program o erase prevedendo di evitare (skip) alcune operazioni. Cio' permette di effettuare delle sequenze di test usando parti di sequenze studiate per modalita' utente.
- Poter influenzare dall'interno o dall'esterno le durate di alcune operazioni eseguite durante un algoritmo. Tipico esempio e' la possibilita' di avere lunghi impulsi di programmazione o di cancellazione per effettuare stress elettrici degli ossidi.
- Poter configurare il chip per meglio adattarsi alle varie situazioni di test. Esempi tipici sono la configurazione degli I/O x8/x16 oppure la configurazione del chip in SLC/MLC.



2c. Garantire un test time breve attraverso :

- tecniche di compressione degli indirizzi, consistenti nel leggere/scrivere in tutte le parole di una pagina dell'array di memoria una stessa pattern di dati afferenti agli I/O esterni. Si comprimono gli indirizzi di parola e si realizza un elevato parallelismo all'interno del singolo chip.
- Tecniche di parallelismo su piu' chips, consistenti nel dotare ogni chip di hardware e software interni tali da effettuare verifiche veloci su pagine intere di array di memoria e memorizzare in latch interni i risultati della verifica. Queste operazioni si possono attivare su molti chips in parallelo risparmiando una notevole quantita' di tempo. Rileggendo poi singolarmente i latches di verifica si rileva se ci sono stati fallimenti o meno.
- Tecniche di auto test, consistenti nell'effettuare automaticamente operazioni di calibrazione o verifiche che possono essere attivate su molti chips in contemporanea.

3) ORGANIZZAZIONE

Una delle strutture fondamentali che interviene nella organizzazione di DFT di una memoria FLASH e' denominata fusibile.

3a. Fusibili

- Nelle odierne memorie Flash sono utilizzati elementi di memoria non volatile detti, per ragioni storiche, fusibili. Infatti nelle prime DRAM, dove furono originariamente introdotti, erano fisicamente realizzati con filo di metallo che veniva fatto 'fondere' con raggio laser. In linea di principio i fusibili non appartengono all'array e non necessariamente ne utilizzano la tecnologia base. Nei fusibili sono memorizzate essenzialmente tre tipologie di informazioni :
- Indirizzi degli elementi (righe, colonne, settori) non funzionali dell'array. Gli elementi non funzionali verranno poi sostituiti (e quindi riparati) da strutture funzionali ridondanti.
 - Informazioni atte alla individuazione del singolo chip: codici specifici di sicurezza o informazioni relative alla tracciabilita' di produzione.



- Informazioni atte a 'configurare' il chip in modo da ottimizzarne il funzionamento. Tali informazioni sono memorizzate in quelli che sono comunemente chiamati 'trim bits'.
- I trim bits a loro volta si dividono in:
- trim bits che effettuano la calibrazione di circuiti particolarmente delicati (bandgap, sensori di temperatura) in modo da riportare i valori elettrici ad essi associati entro i valori nominali di progetto correggendo le imprecisioni dovute alle variazioni di processo.
 - trim bits che configurano il modo di funzionamento per una specifica applicazione e/o impacchettamento. Per esempio, indicando il parallelismo dei dati, o abilitando la diversa funzionalità 'SLC/MLC (singolo livello o multilivello), o abilitando differenti insiemi di comando utente.
 - trim bits che definiscono vari parametri atti al corretto funzionamento di uno specifico algoritmo. Per esempio: la tensione iniziale da applicare alle WL durante una operazione di programmazione, il suo incremento nel susseguirsi dei vari impulsi, la durata degli impulsi.

Le informazioni immagazzinate nei fusibili per la gestione della ridondanza e del trim si suppone siano totalmente trasparenti all'utente e strettamente legati alla produzione ed ottimizzazione del chip. Per questo motivo la gestione dei fusibili e' una fase che viene controllata esclusivamente in modalita' di test.

Le strutture circuitali per la gestione dei fusibili sono riportate in Fig. 3.1 e Fig. 3.2 e sono composte da :

- Il fusibile, cioè le celle di memoria non volatile. In genere le celle non volatili possono essere organizzate in due possibili configurazioni: differenziale o single ended. La struttura differenziale e' composta da due rami in uno dei quali deve sempre essere presente una cella programmata e nell'altro una cella cancellata. Situazioni in cui entrambi i rami contengono celle cancellate o celle programmate sono anomale e fonti di seri malfunzionamenti.
- Il latch del fusibile, cioè una struttura di memorizzazione temporanea del valore dei fusibili.



- Circuiti per trasferire il contenuto del latch ai pads di I/O e viceversa.
- Circuiti per trasferire il contenuto del latch nei fusibili e viceversa.
- Circuiti per propagare il contenuto del latch all'interno del chip per l'esplicazione della funzione del fusibile.

La macchina di test gestisce le informazioni associate ai vari fusibili scrivendole nel latch e facendole propagare nel chip. Quando la riparazione e ottimizzazione del chip saranno state completate, il contenuto del latch verra' trasferito permanentemente nei fusibili. Cio' e' dovuto al fatto che la scrittura nei latch e' molto piu' veloce che non la scrittura nei fusibili.

Analoga situazione avviene nel caso di lettura dei fusibili nel funzionamento del chip in modalita' utente: al power up i fusibili vengono tutti trasferiti nei rispettivi latch e propagati nel chip.

Per diminuire il numero di connessioni, alcuni fusibili sono gestiti in modo dinamico, nel senso che i contenuti dei latch non si propagano tutti contemporaneamente nel chip ma vengono selezionati dal controller a seconda delle necessita'.



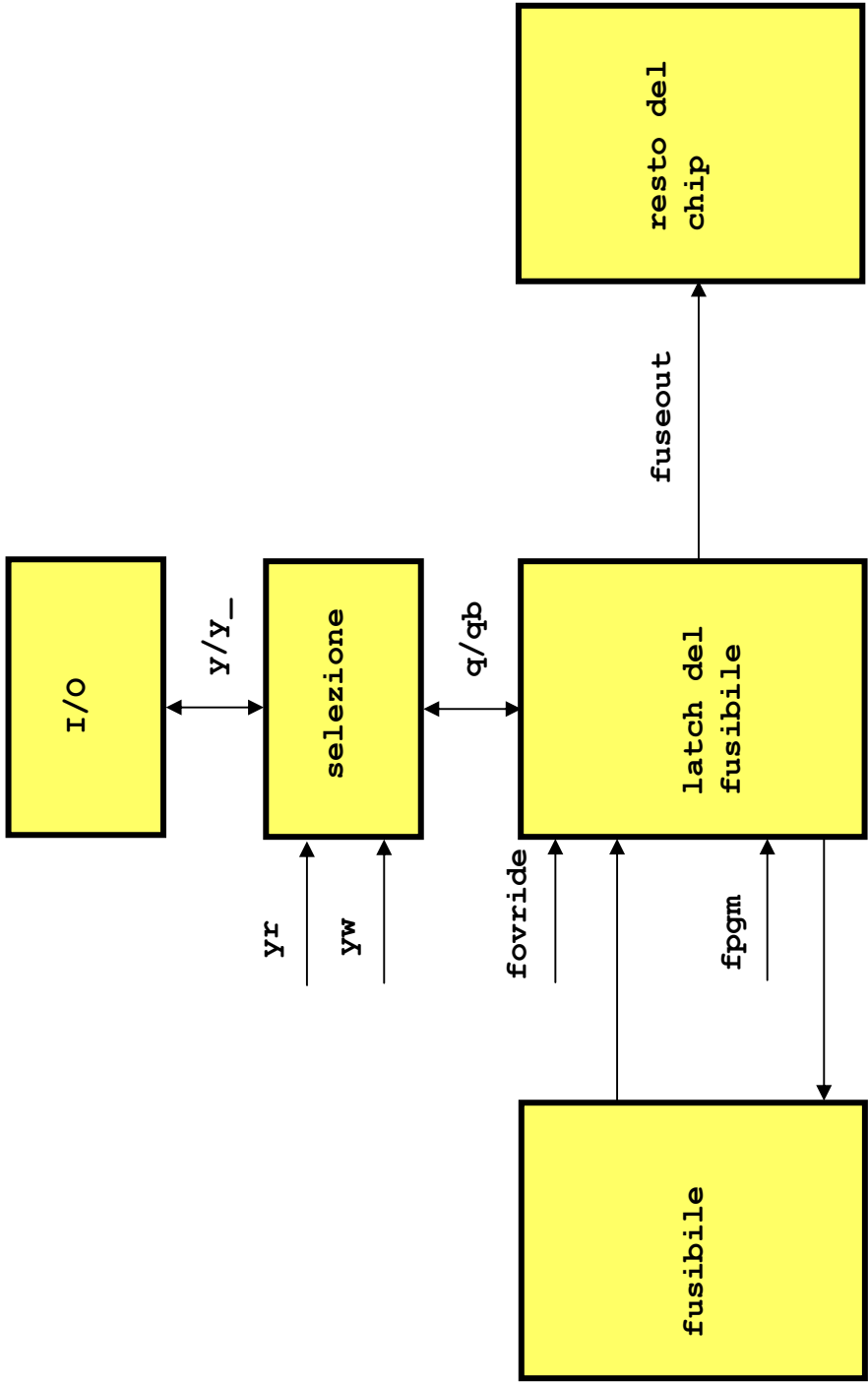


Fig. 3.1 - schema a blocchi del fusibile



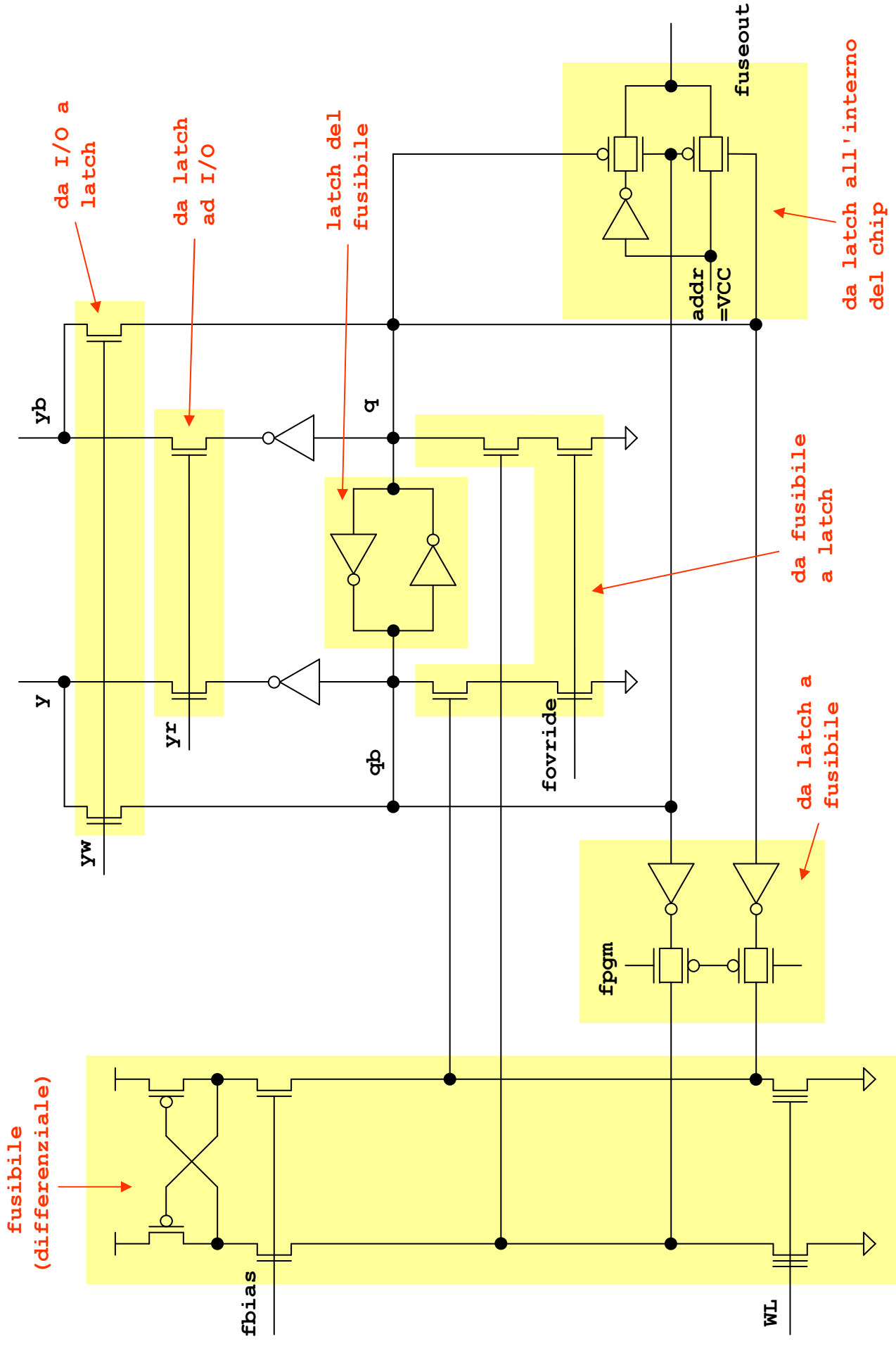


Fig. 3.2 - struttura dettagliata del fusibile



3b. Attivazione della modalita' di test (test mode entry)

La gestione della memoria in modalita' di test e' una operativita' che, per motivi di proprieta' intellettuale e per la delicatezza delle operazioni coinvolte, si vuole gestire esclusivamente in ambiente di produzione (in factory) e non si vuole fornire all'utente. Per evitare che l'utente attivi accidentalmente la modalita' di test, si adottano vari accorgimenti. Il piu' diffuso (HV mode) e' quello di avere un pin del chip che deve essere mantenuto ad una tensione di molto superiore alla tensione VCC di alimentazione per tutto il tempo che si intende mantenere attiva la modalita' di test. In alternativa, se si vuole risparmiare un driver per la generazione di HV nella macchina di test, si puo' attivare la modalita' di test con l'inserimento di una lunga e specifica sequenza di comandi (LV mode) nessuno dei quali e' contemplato nelle specifiche utente. Per maggiore sicurezza, dopo aver attivato la modalita' di test in HV mode o in LV mode, lo specifico test mode viene attivato dopo aver immesso delle opportune sequenze di avviso o di abilitazione.

L'organizzazione interna in modalita' utente di una memoria FLASH puo' esser di due tipi: modale o a registri. L'organizzazione modale prevede che un controllore interno generi segnali di modalita' utente, per esempio modalita' di programmazione, lettura, cancellazione, ... e tali segnali di modo si propagano ai circuiti periferici quali blocchi analogici, sens amplifiers, indirizzi e I/O path, i quali attiveranno diverse parti al loro interno corrispondenti alla specifica modalita' operativa. L'organizzazione della modalita' utente a registri invece prevede che il controllore interno, in base alle richieste operative dell'utente non generi segnali di modo ma attivi direttamente gli specifici segnali di attuazione (che vengono memorizzati in registri composti da flip-flop) delle operazioni elementari che ogni singolo circuito di periferia deve svolgere. L'organizzazione modale, a differenza di quella a registri, prevede una intelligenza piu' distribuita nelle periferiche che concentrata nel controllore. L'organizzazione modale puo' avere delle performances di velocita' maggiori rispetto all'organizzazione a registri che per contro ha una piu' elevata flessibilita'. La organizzazione circuitale della modalita' di test deve adeguarsi alla organizzazione della modalita' utente: si avra' quindi una organizzazione DFT modale e una a registri.



3c. Organizzazione DFT modale

L'organizzazione modale dei modi di test e' riportata in Fig. 3.3 e Fig. 3.4 e prevede che dopo l'attivazione della modalita' di test ci sia un circuito di interfaccia con l'esterno (CUI = Command User Interface) in grado di rilevare una sequenza fissa di comandi di preavviso. Il comando successivo alla sequenza di preavviso, viene interpretato come comando per l'attivazione di uno specifico test.

Un circuito di decodifica dei modi di test (spt decoder) in corrispondenza ad un segnale di temporizzazione spt_clk genera un particolare modo di test in corrispondenza ad uno specifico comando. Si possono attivare diversi modi di test che rimangono attivi fino a che non vengono disattivati.

Dopo aver attivato i voluti modi di test si possono attivare anche operativita' utente che vengono eseguite con i vincoli imposti dai modi di test attivi. Per esempio se si attiva un modo di test che impone l'abilitazione di tutte le word lines di un settore (awlon = all word lines ON) e poi si impone un comando utente di programmazione, l'operazione di programmazione verra' effettuata sul blocco selezionato con tutte le word lines abilitate.

I modi di test si dividono in due categorie: modi di test per la gestione dei fusibili (cancellazione, scrittura, lettura dei fusibili) e modi di test per la gestione delle altre parti del chip (address/data path, circuiti analogici delle pompe e dei riferimenti, array di celle di memoria).



Fig. 3.3 - Organizzazione DFT modale : architettura

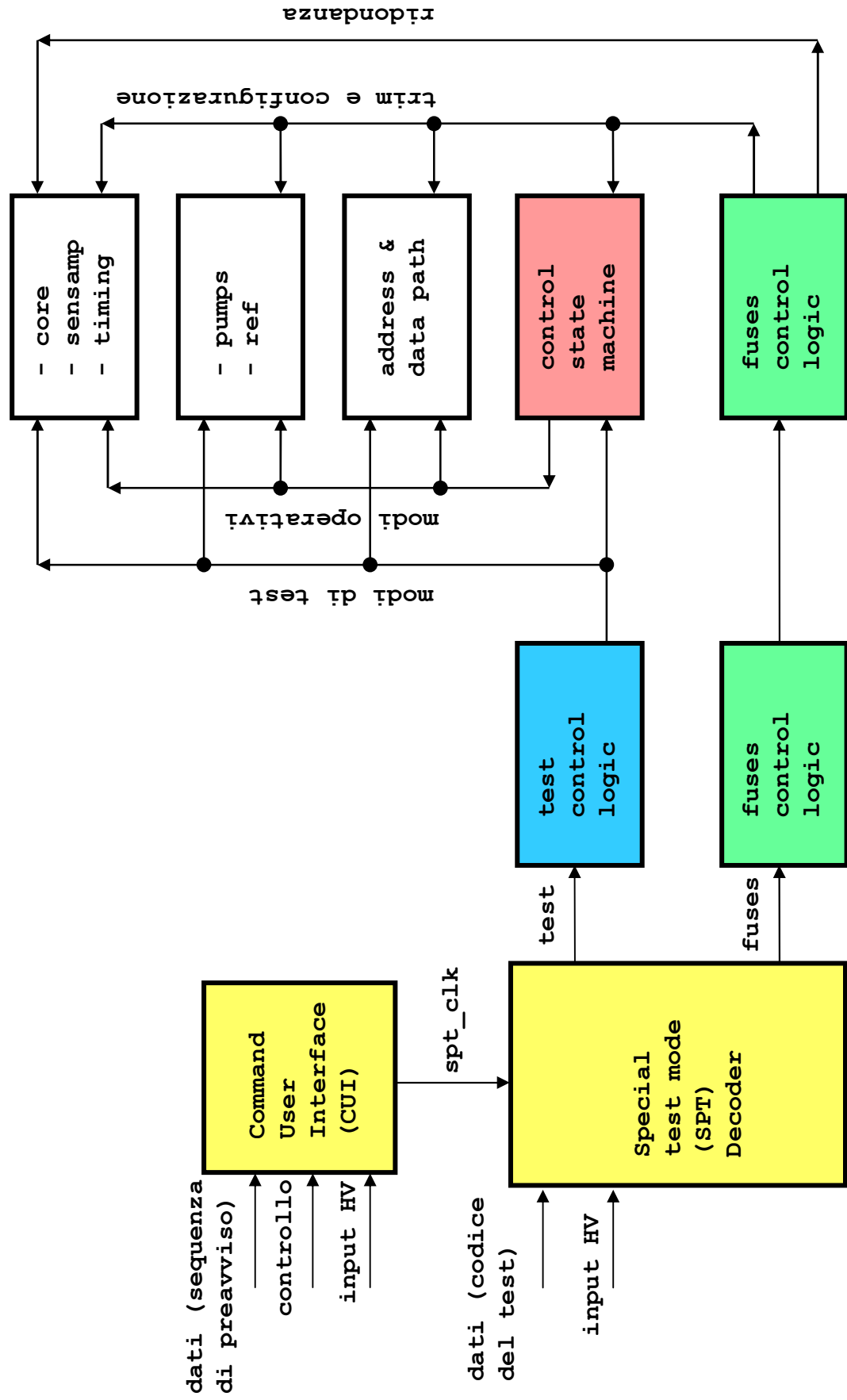
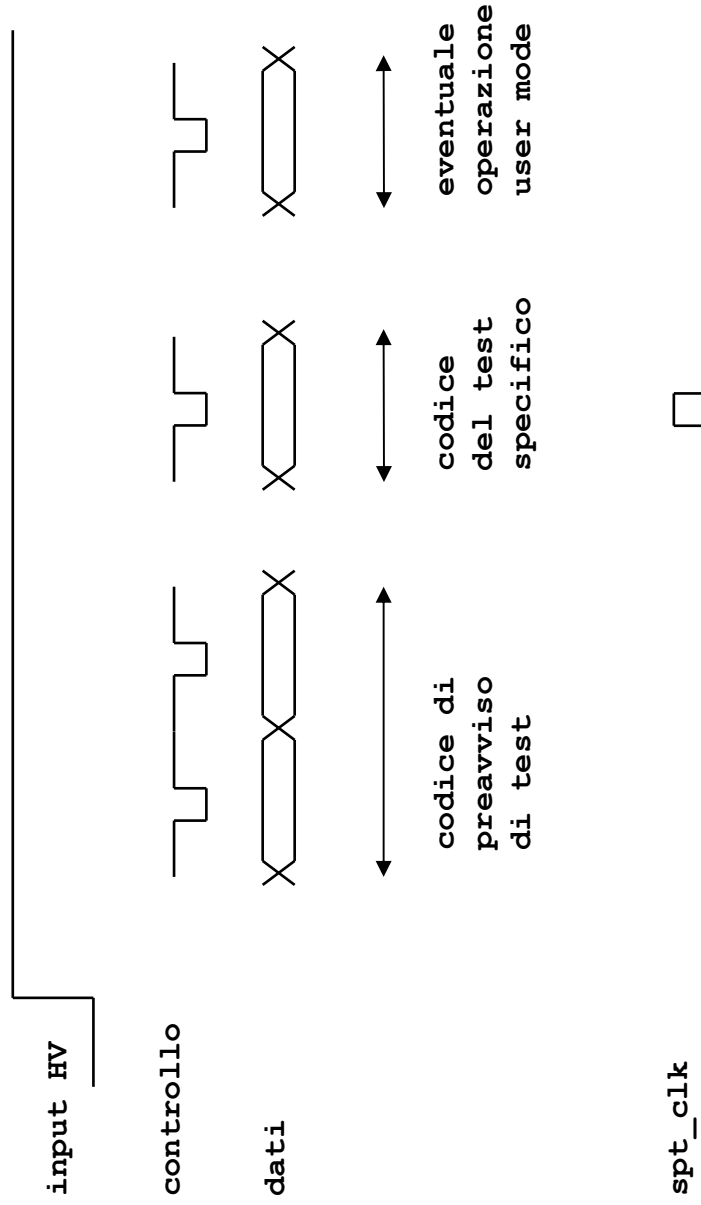


Fig. 3.4 - Organizzazione DFT modale :
forme d'onda per l'attivazione di test



3d. Organizzazione DFT a registri

La struttura base della organizzazione DFT a registri e' il 'registro di controllo' (control register) riportata in Fig. 3.5 e Fig. 3.6. Il registro di controllo e' organizzato, per ottimizzare i collegamenti, a gruppi di 8 strutture di flip-flop. Un bus di indirizzi $\text{add_reg} \langle 7:0 \rangle$ a 8 componenti e' comune a tutti i registri del chip, cosi' come un bus dati $\text{in_reg} \langle 7:0 \rangle$ a 8 componenti e' comune a tutti i registri del chip. Tre segnali di controllo wrt_reg (per la scrittura), clr_reg (per il reset generale) e rd_reg (per la lettura) sono comuni a tutti i registri del chip. Ogni registro di controllo e' dotato di una decodifica di indirizzi in grado di abilitare lo specifico registro ($\text{sel_reg} = 1$) quando l'indirizzo fornito coincide con l'indirizzo del registro. Se l'indirizzo fornito coincide con l'indirizzo del registro, i dati di ingresso vengono memorizzati nei flip-flop e resi disponibili all'uscita del registro in corrispondenza al fronte positivo del segnale wrt_reg .

L'architettura DFT a registri e le forme d'onda per scrivere nei registri di controllo sono riportati in Fig. 3.7 e in Fig. 3.8.

Dopo essere entrati in modalita' di test in HV mode o LV mode, l'organizzazione DFT a registri non prevede l'attivazione di uno o piu' modi specifici di test ma prevede essenzialmente tre comandi :

- 1) cancellazione di tutti i registri di controllo,
- 2) lettura di uno specifico registro di controllo,
- 3) scrittura di uno specifico registro di controllo.

Dopo aver dato il comando di scrittura nei registri di controllo e' possibile indirizzare e immettere dati in ogni registro di controllo ed attivare ognuno dei segnali di attuazione in uscita dai registri di controllo. I registri di controllo sono gestiti completamente da un controllore interno per le operazioni utente o da una macchina di test in modalita' di test. Le due operativita' sono mutualmente esclusive. Nel DFT a registri non esistono dei veri e propri modi di test ma uno specifico test viene realizzato con una sequenza di passi residenti nella macchina di test e che si attuano scrivendo opportuni contenuti entro registri di controllo. Nel DFT a registri c'e' una interazione minore, rispetto a DFT modale, tra fusibili e blocchi da controllare: infatti le informazioni dei fusibili vengono usate dal controllore per generare gli opportuni contenuti da scrivere entro i registri di controllo.



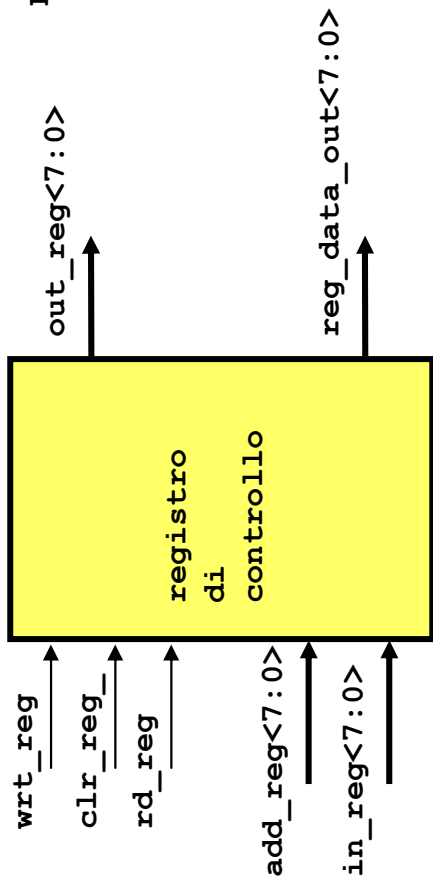
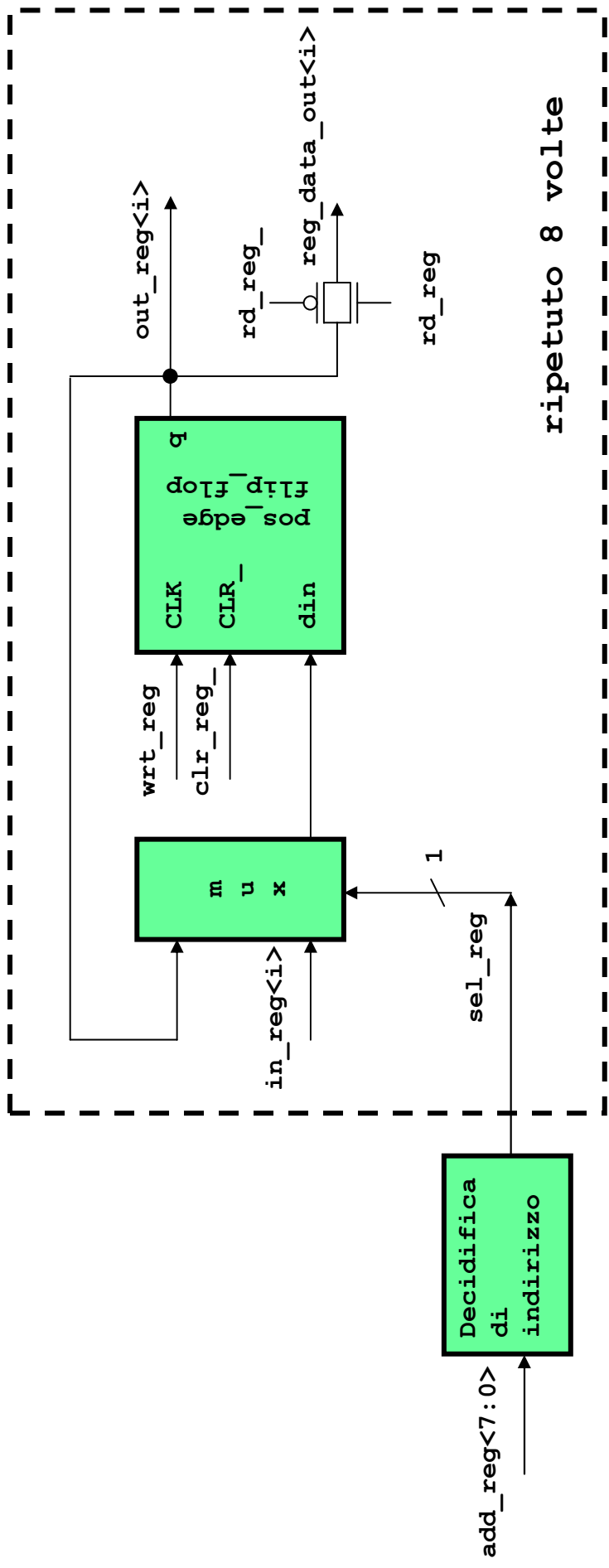


Fig. 3.5 - Organizzazione DFT a registri :
struttura interna dei registri



ripetuto 8 volte



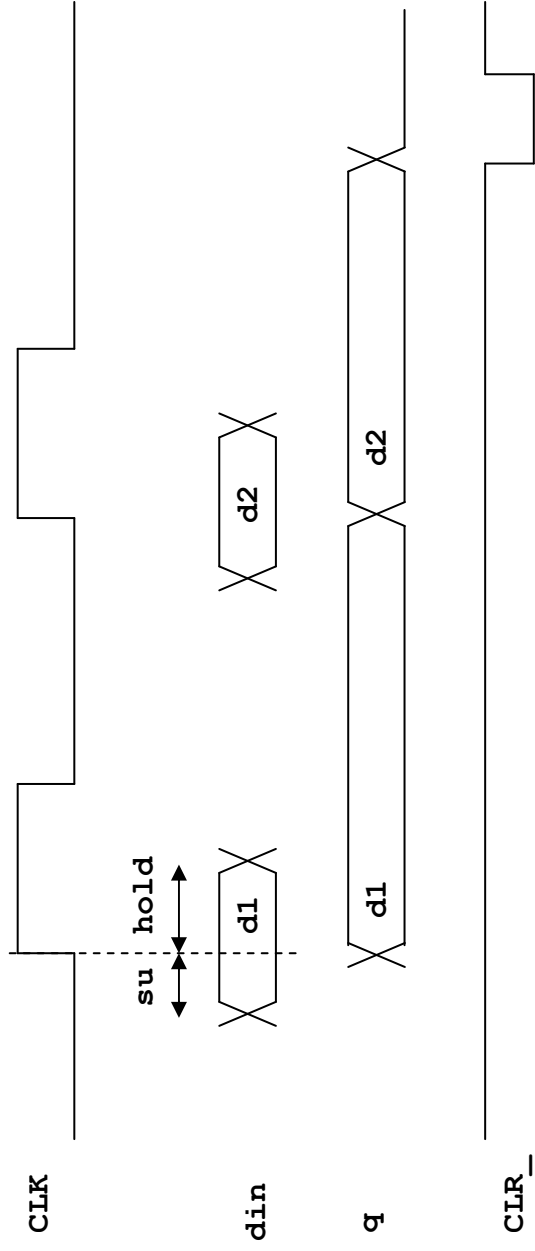
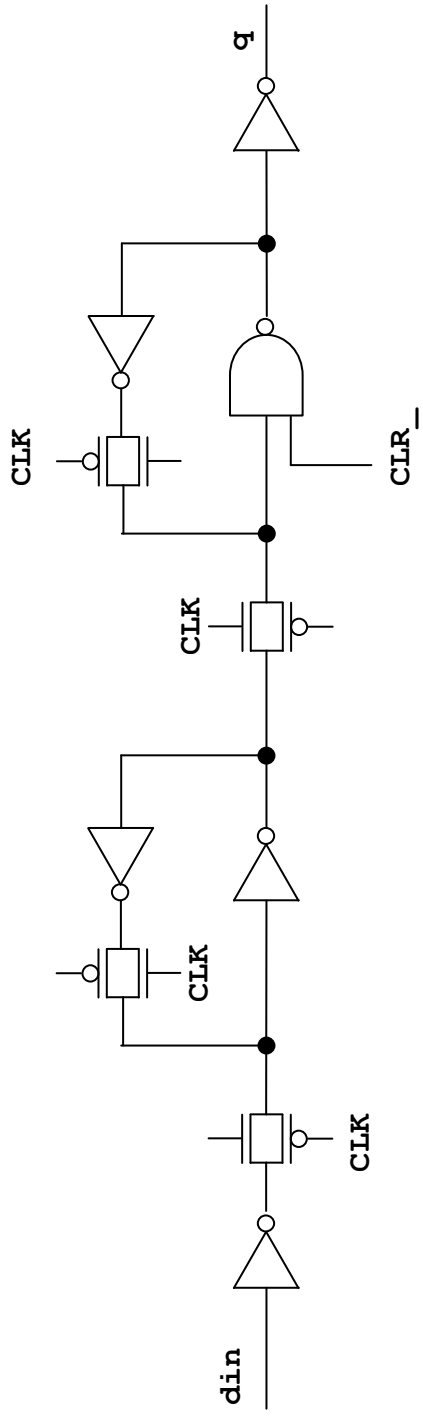


Fig. 3.6 - positive edge triggered flip-flop



Fig. 3.7 - Organizzazione DFT a registri : architettura

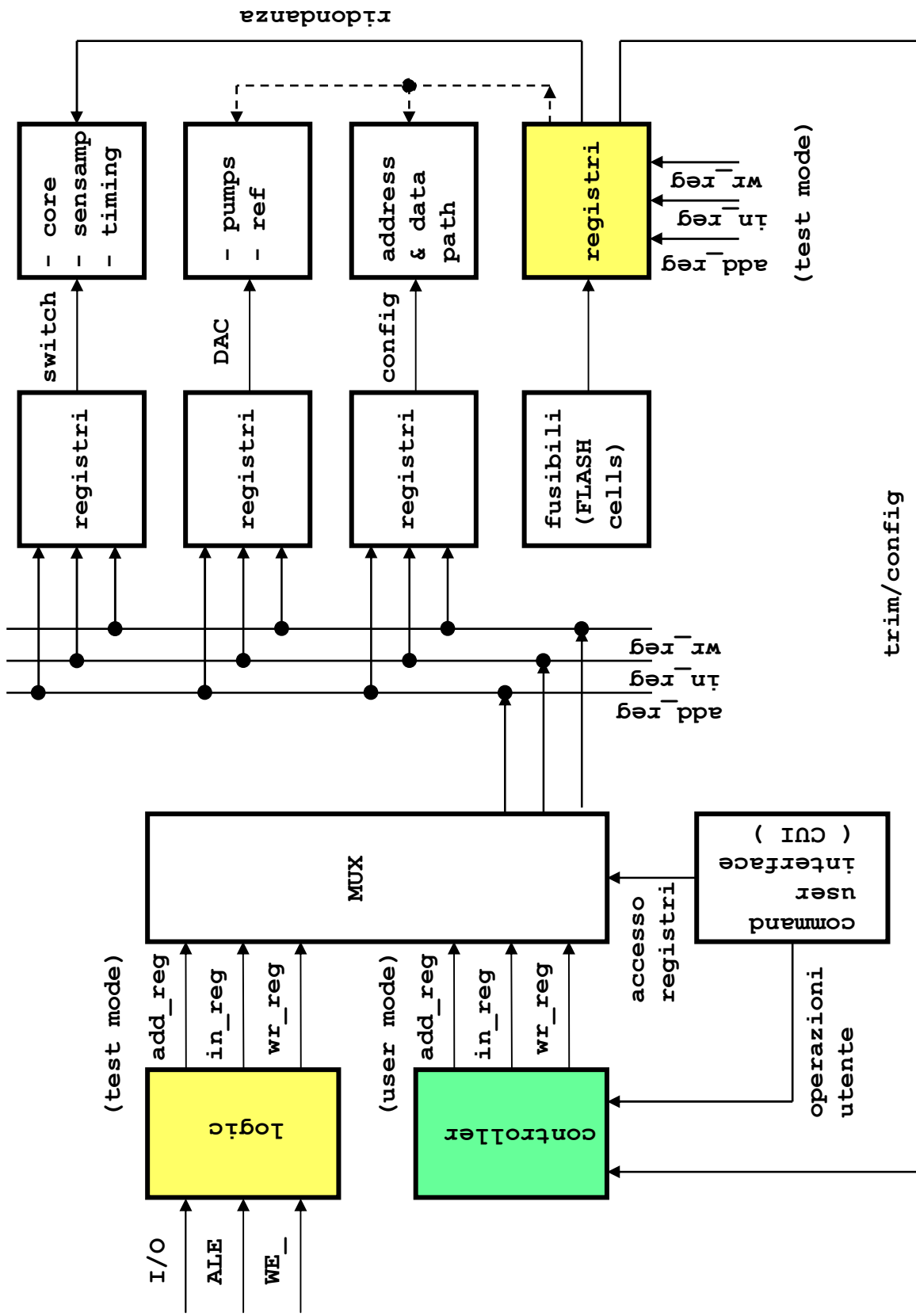
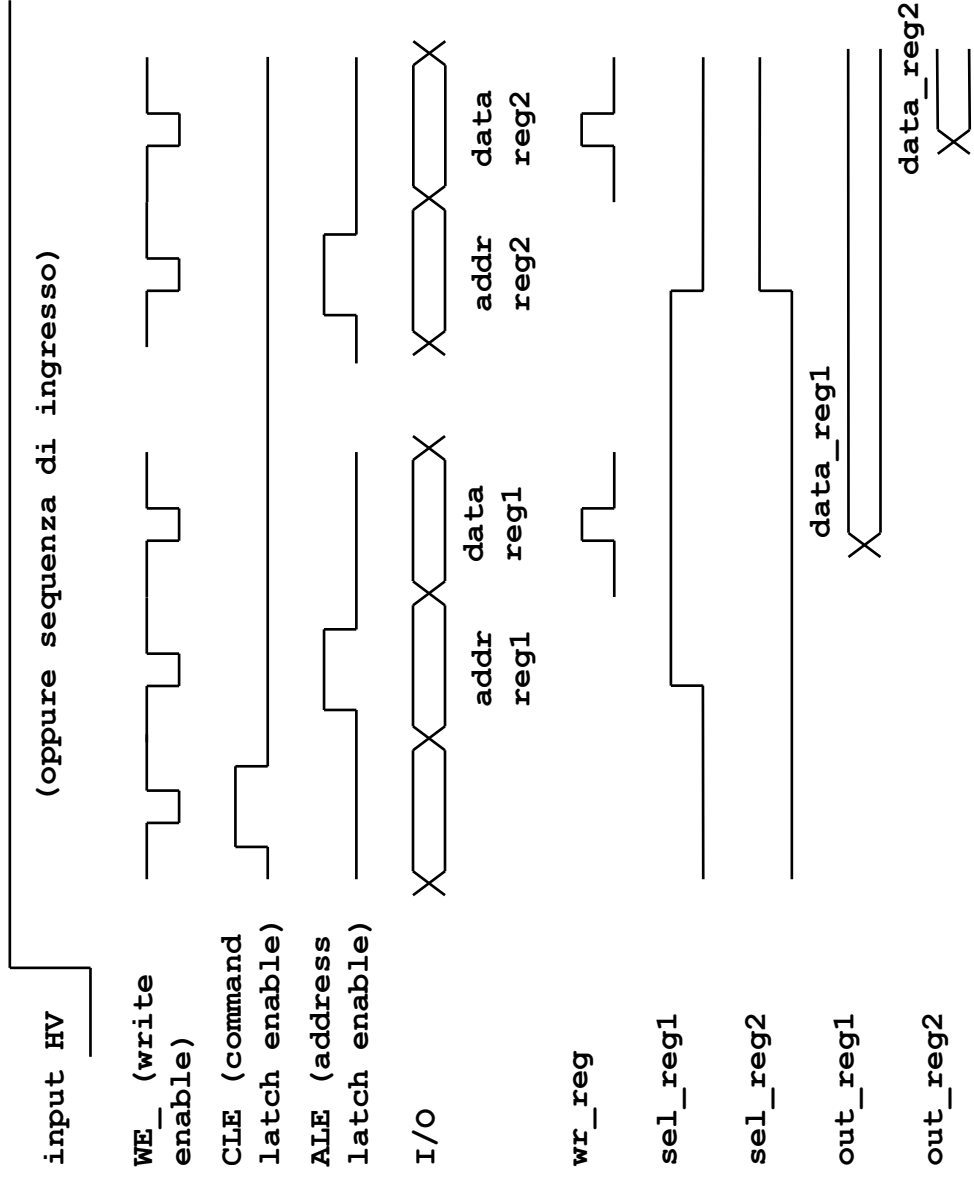


Fig. 3.8 - Organizzazione DFT a registri :
 forme d'onda per la scrittura nei registri di controllo



4) RIDONDANZA

L'uso della ridondanza nelle memorie FLASH puo' migliorare in modo vistoso la resa di produzione. Cio' e' particolarmente vero durante l'avvio di produzione di una nuova tecnologia (ramp-up) quando la resa puo' essere molto bassa a causa della fase di apprendimento insita in un nuovo flusso di processo. Il miglioramento della resa nell'avvio di produzione di un nuovo processo permette di ridurre i tempi di immissione nel mercato di nuovi prodotti e di abbassare i costi iniziali di sviluppo. Man mano che le memorie diventano piu' dense e i processi piu' complessi la resa di produzione si prevede sempre piu' bassa : l'uso della ridondanza puo' aiutare nel raggiungere rese di produzione molto piu' alte della resa naturale anche quando il processo ha raggiunto la sua fase di maturazione. Tre tipi di ridondanza possono essere presi in considerazione : ridondanza di riga, colonna e di blocco; ognuna delle quali e' adatta a riparare situazioni specifiche di fallimento.

La ridondanza di riga, che si usa in memorie FLASH NOR, prevede la presenza in un blocco di ulteriori righe (ridondanti) oltre a quelle di normale uso (regolari). Nel caso in cui una o piu' righe regolari presentano dei difetti, esse possono essere sostituite da righe ridondanti dello stesso blocco.

La ridondanza di colonna prevede la presenza in un blocco di ulteriori colonne (ridondanti) oltre a quelle di normale uso (regolari). Nel caso in cui una o piu' colonne regolari presentano dei difetti, esse possono essere sostituite da colonne ridondanti dello stesso blocco.

Nel caso di memorie FLASH NAND, l'uso di righe ridondanti non e' possibile perche' celle di memoria afferenti a diverse righe sono in serie. In questo caso difetti di riga vengono trattati sostituendo l'intero blocco con uno degli extra blocchi (ridondanti) presenti nel chip.

Per dare una idea di quanto puo' essere l'impatto della ridondanza, un esempio di resa di produzione in funzione della ridondanza di riga/colonna e' riportata in Fig. 4.1 per memorie FLASH NOR da 1Mb, 4Mb, 8Mb in cui si vede che l'uso della ridondanza puo' migliorare la resa dal 20% al 45% a seconda della densita'.

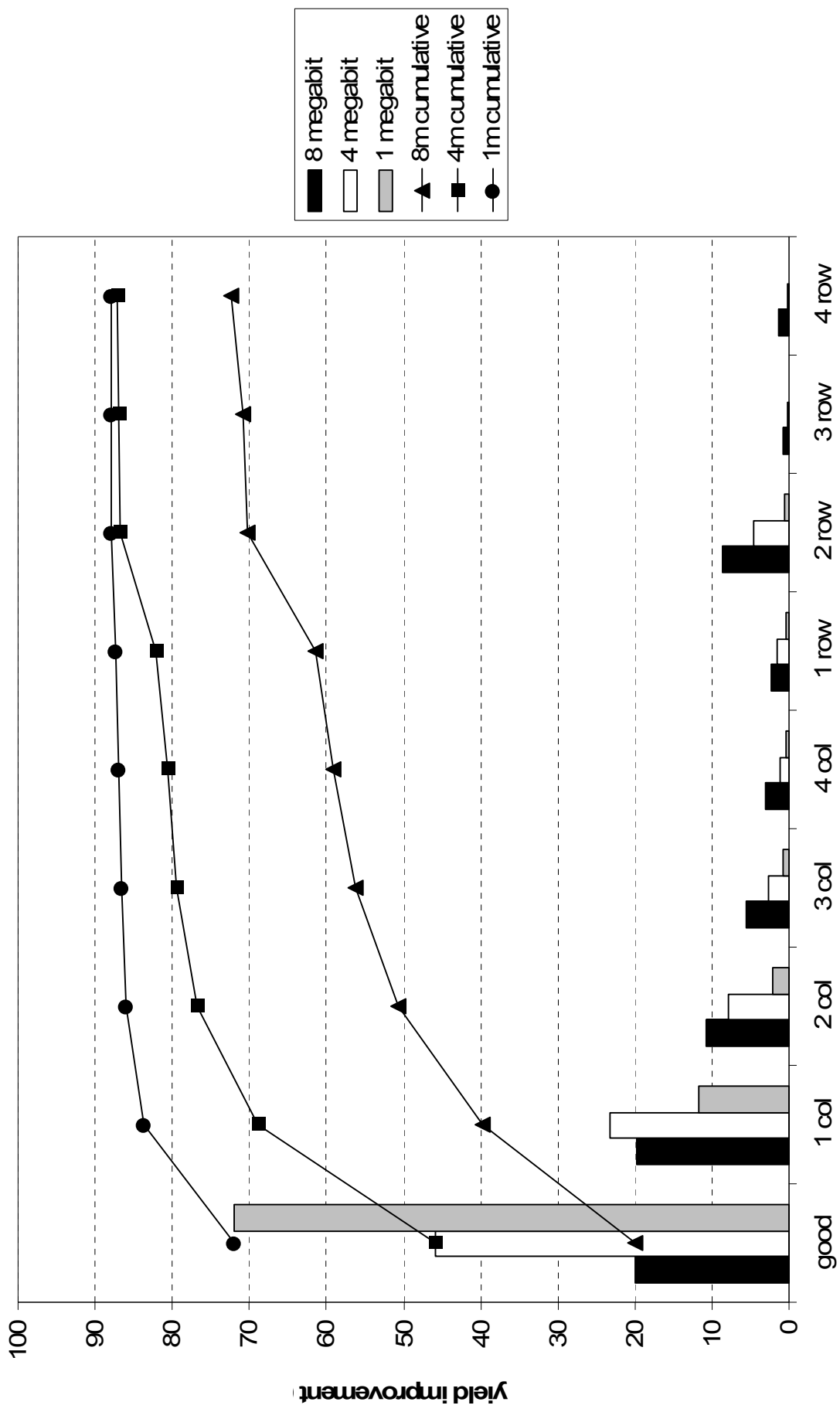


Fig. 4.1 - Effetto della ridondanza sulla resa di produzione



4a. Il circuito di match

Il circuito usato nella selezione di una struttura di ridondanza (riga/colonna/blocco) in corrispondenza di un particolare indirizzo (di riga/colonna/blocco) ha subito una evoluzione nel corso degli anni ma si può ricondurre ad una struttura base come quella riportata nello schema a blocchi di Fig. 4.2 e Fig. 4.3. Lo schema differenziale (fusibile/latch/pass gates) riportato in Fig. 3.2 può essere facilmente usato per costruire uno schema di match del tipo riportato in Fig. 4.3.

Il fusibile di abilitazione (ENABLE FUSE) e' un fusibile associato ad ogni specifica struttura di ridondanza e serve a 'marcare' quella struttura di ridondanza come struttura da usare in corrispondenza all' indirizzo scritto negli n fusibili di indirizzo (ADDRESS FUSES).

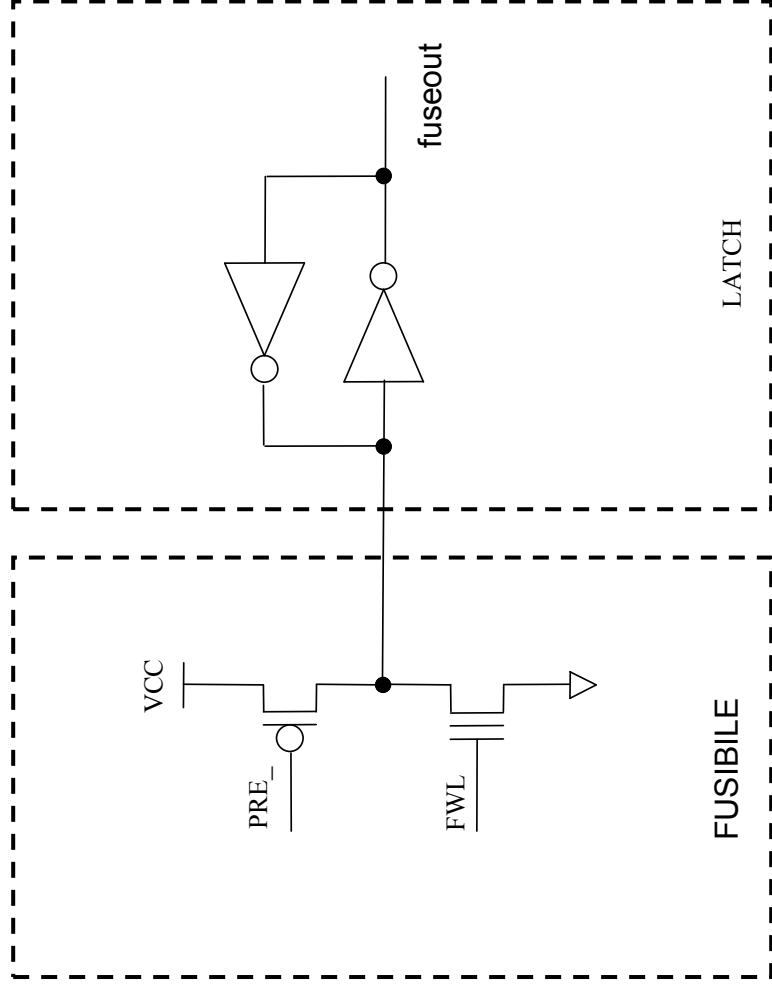
Se il generico i-mo fusibile di indirizzo e' cancellato si ha $FA[i]=ADDR[i]$, se e' programmato si ha $FA[i]=\neg ADDR[i]$. Perche' si abbia MATCH=1 e quindi la sostituzione della struttura regolare con la ridondante, si deve avere $FEN=FA[0]=\dots=FA[n]=1$ il che si ottiene programmando il fusibile di abilitazione, lasciando cancellati tutti i fusibili corrispondenti alle componenti 1 dell'indirizzo da riparare e programmando tutti i fusibili corrispondenti alle componenti 0 dell'indirizzo da riparare.

Il segnale forcedeselb e' un segnale di test che serve a forzare $MATCH=0$ anche se l'indirizzo fornito corrisponde ad una struttura da riparare (deselezione forzata della ridondanza).

Il segnale forcematch e' un segnale di test che serve a forzare $MATCH=1$ anche se l'indirizzo fornito non corrisponde ad una struttura da riparare (selezione forzata della ridondanza).

I segnali forcedeselb e forcematch servono ad accedere in test mode alle sole strutture regolari o alle sole strutture ridondanti indipendentemente dal contenuto scritto nei fusibili di ridondanza.





. PRE_ = 0

precarica la bit line del fusibile e reset del latch

. FWL > 0

lettura del fusibile e set del latch nel caso di cella cancellata

Fig. 4.2 - Schema semplificato della struttura fusibile+latch riportati in Fig. 3.1 e 3.2.



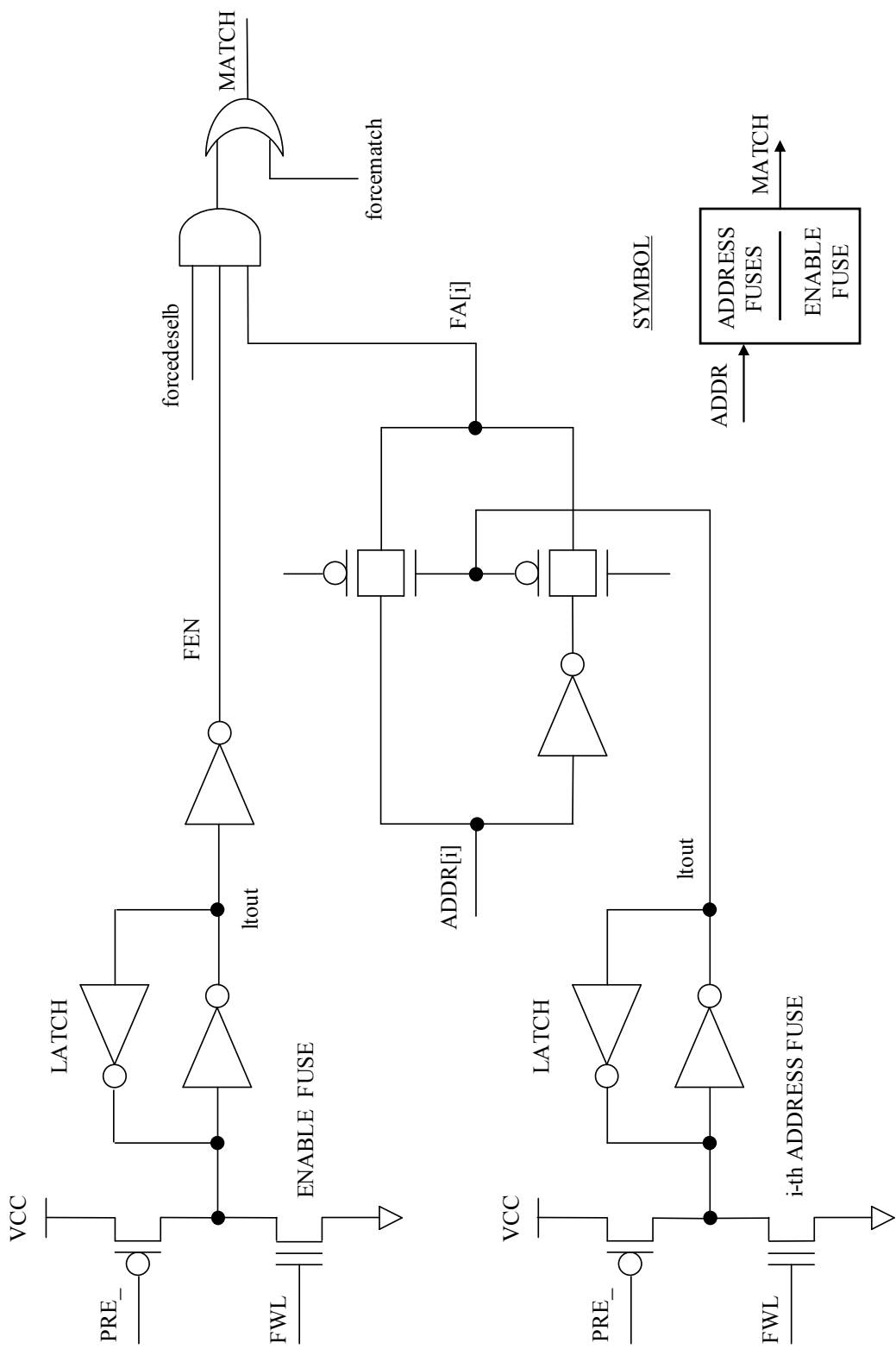


Fig. 4.3 - Schema a blocchi del circuito di match



11/30/2006



4b. Ridondanza di riga

La ridondanza di riga e' effettuata nelle memorie FLASH NOR aggiungendo una o piu' righe ad ogni settore. In Fig. 4.4 e' riportato uno schema a blocchi di tale architettura usando due word lines ridondanti (REDWL1 e REDWL2). Le word lines ridondanti sono pilotate da un level shifter (LS) che ha la stessa struttura di quelli usati nelle word lines regolari WL0,WL1,...,WLn. Ognuna delle righe ridondanti ha la sua struttura di fusibili per rilevare un particolare indirizzo di riga. Se REDWL2 e' scelta per riparare la WL0, il fusibile di enable associato a REDWL2 deve essere programmato e i fusibili di indirizzo associati a REDWL2 devono essere programmati per rilevare l'indirizzo di WL0. Quando viene fornito l'indirizzo di WL0, il segnale MATCH2 e' forzato alto a REDWL2 e' selezionata mentre tutte le word lines regolari sono deselezionate. REDWL2 sostituirà WL0 per operazioni di lettura, scrittura e verifica. Per quanto riguarda l'operazione di cancellazione, tutte le righe regolari WL0,WL1,...,WLn e ridondanti REDWL1,REDWL2 sono cancellate insieme trovandosi nello stesso settore.

4c. Ridondanza di colonna

La ridondanza di colonna e' effettuata nelle memorie FLASH aggiungendo una o piu' colonne in ogni settore. La colonna ridondante sostituirà la colonna fallente corrispondente alla linea DQ di IN/OUT che risulta fallente. In Fig. 4.5 e 4.6 sono riportati gli schemi a blocchi del percorso di lettura e di scrittura della ridondanza di colonna. Con riferimento al percorso di lettura di Fig. 4.5, ogni linea di uscita Q0,..., Qn e' associata a un gruppo di colonne selezionate da un indirizzo di colonna (COL ADDR). Un gruppo corrispondente di colonne ridondanti e' selezionato per ogni Q in corrispondenza ad uno stesso indirizzo di colonna. All'uscita del multiplexer di colonna (MUX), le colonne regolari selezionate sono connesse ai sensamplicifiers SA0, ..., SAn mentre le colonne ridondanti selezionate sono connesse al sens amp ridondante RSA. Tra ognuno dei sensamp SA0,...,SAn e le corrispondenti linee di uscita Q0,...Qn ci sono due buffers con controllo 3-state aventi un gruppo di fusibili associato. Se il fusibile di enable di un gruppo e' programmato, quando l'indirizzo di colonna e' consistente con quello programmato nei fusibili, il segnale di MATCH e' forzato a 1 e la colonna ridondante sarà connessa alla linea di uscita al posto di quella regolare.



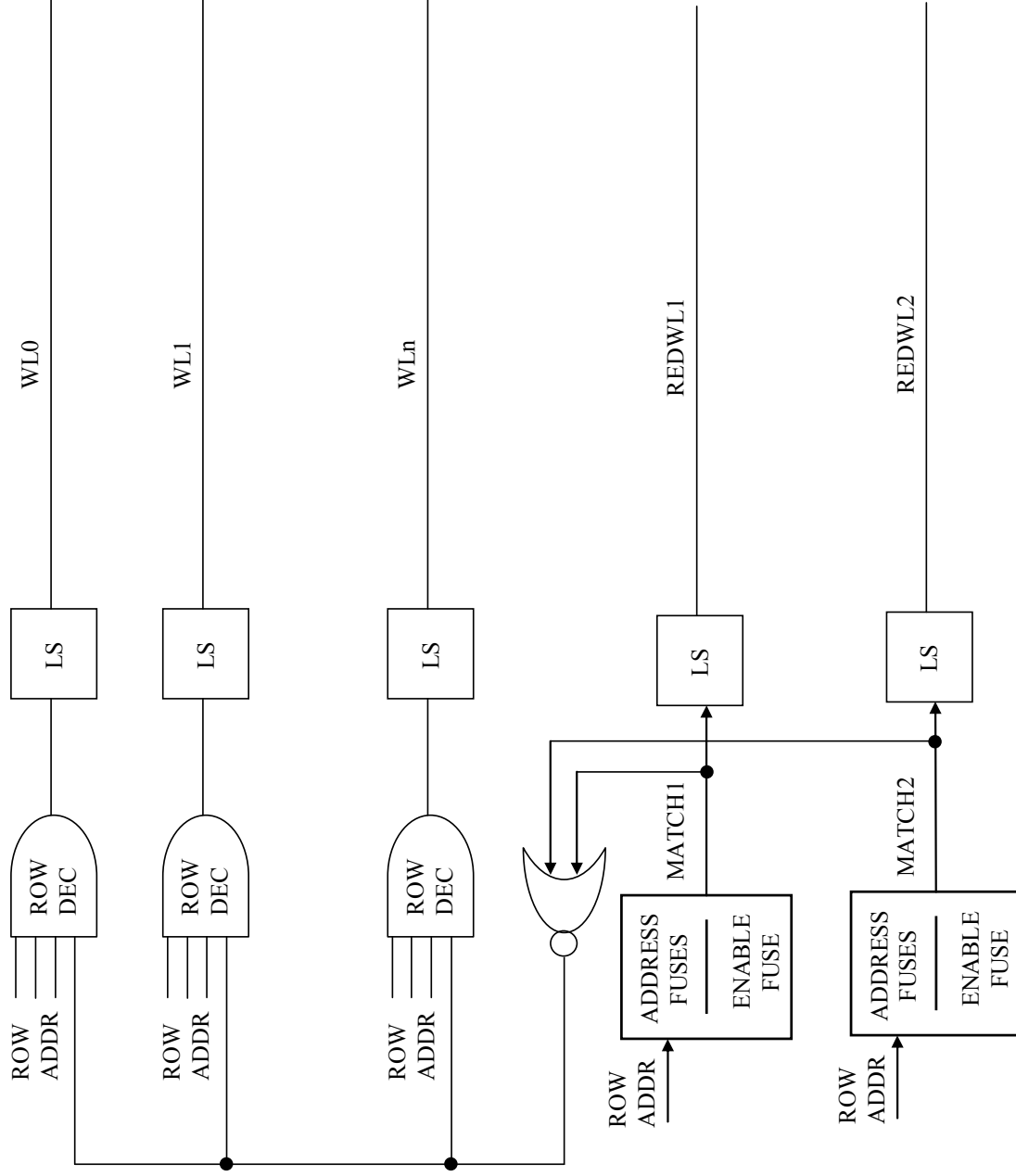


Fig. 4.4 - Schema a blocchi della ridondanza di riga



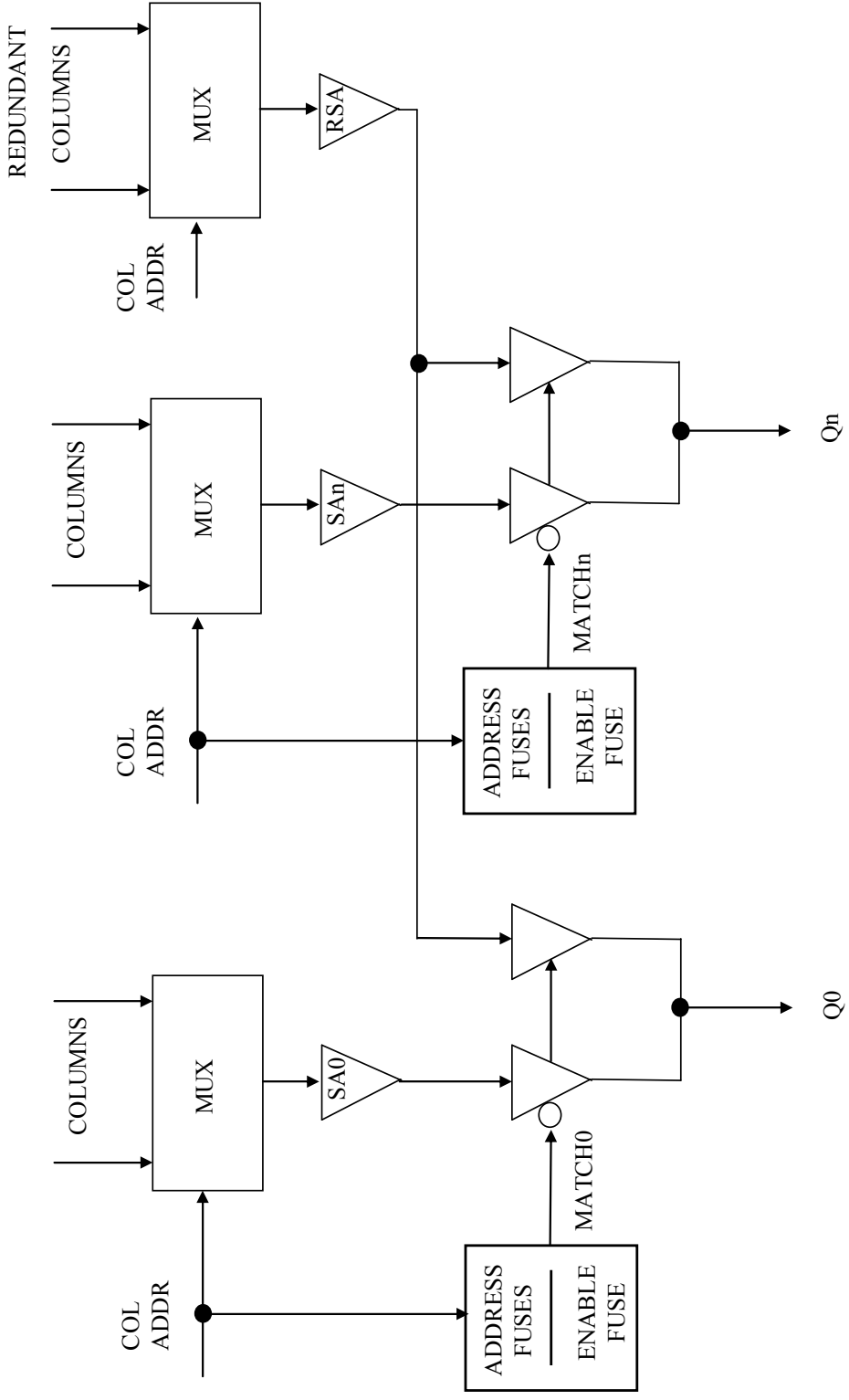


Fig. 4.5 - Schema a blocchi della ridondanza di colonna (circuiti di lettura)



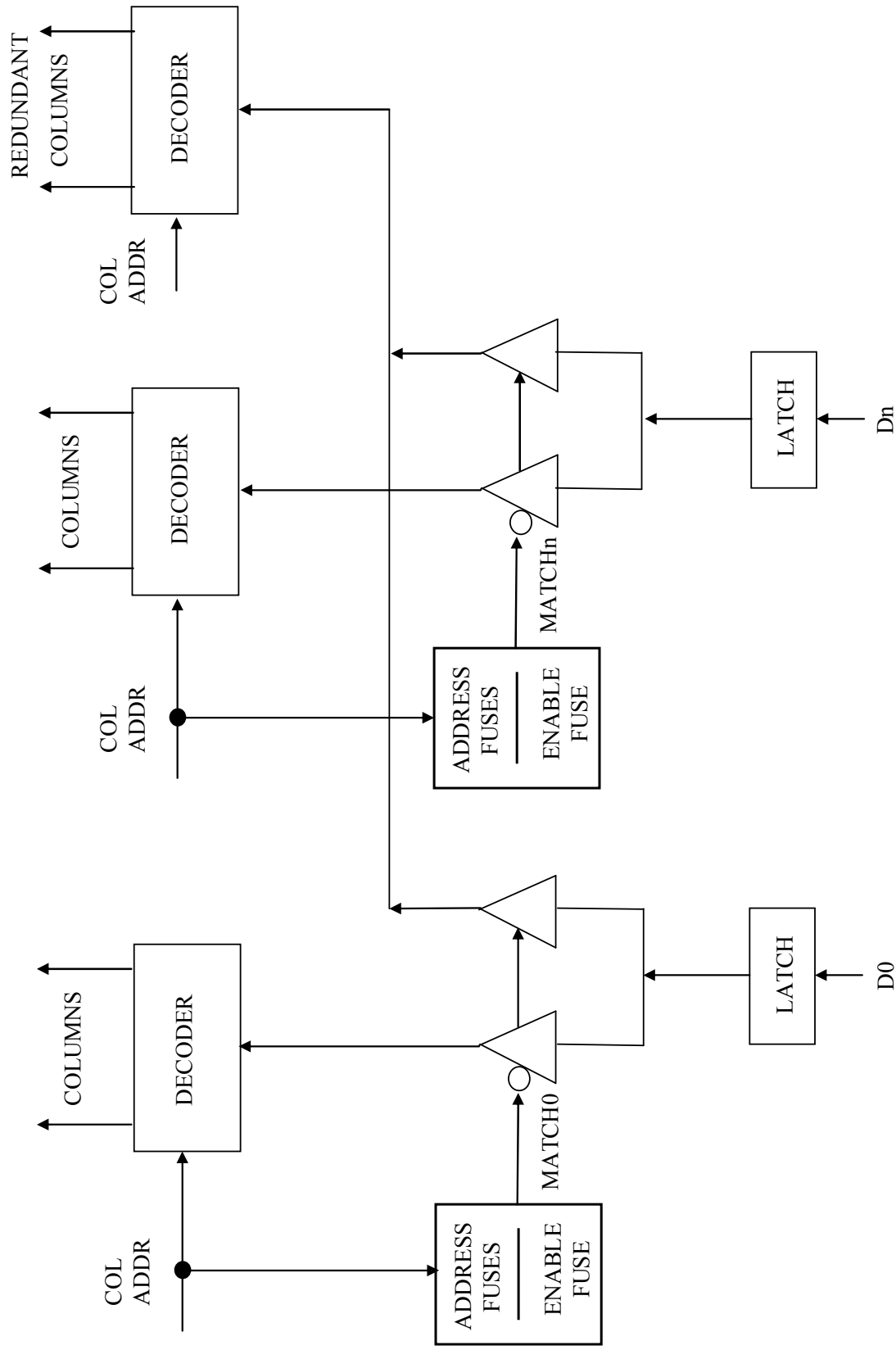


Fig. 4.6 - Schema a blocchi della ridondanza di colonna (circuiti di scrittura)



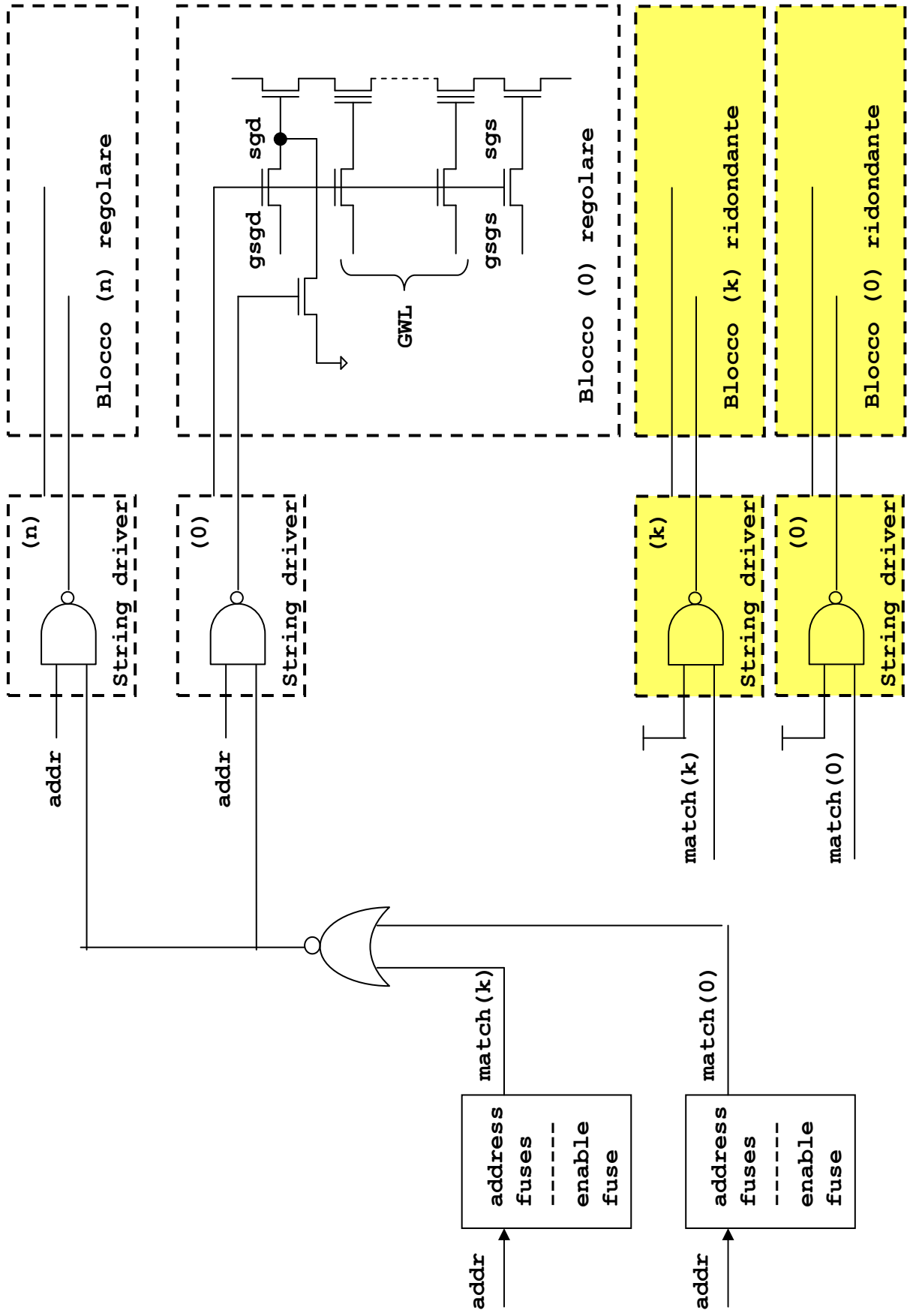


Fig. 4.7 - Schema di principio della ridondanza di blocco



Il percorso di scrittura nella ridondanza di colonna, riportato in Fig. 4.6, e' simile a quello di lettura. Il gruppo di fusibili e' lo stesso usato nel percorso di lettura. Ogni linea di ingresso D0,...Dn, dopo un latch usato per bloccare il valore del dato durante l'algoritmo di scrittura, puo' essere usata per programmare la colonna regolare o la colonna ridondante se il corrispondente gruppo di fusibili genera una condizione di match. La discussa struttura di ridondanza di colonna non ha penalita' di velocita' perche' la colonna regolare e ridondante sono lette in parallelo e sono scambiate dopo il sens amp. In questa situazione il segnale MATCH ha un valore stabile quando il bit di informazione ha completato la sua propagazione dalla cella di memoria al sens amp.

4d. Ridondanza di blocco

Un possibile schema di principio per la ridondanza di blocco e' riportato in Fig. 4.7 con riferimento ad una memoria FLASH di tipo NAND. La struttura e' molto simile a quella di ridondanza di riga riportata in Fig. 4.4. Nel caso di ridondanza di blocco tutte le stringhe di celle di un blocco regolare da sostituire vengono isolate al momento in cui un indirizzo di blocco coincide con il contenuto programmato in un determinato gruppo di fusibili. Il blocco ridondante a cui quei fusibili sono associati verra' abilitato.

5) TRIMS

Di seguito vengono riportati alcuni esempi di uso di modi di test in congiunzione con fusibili per effettuare operazioni di calibrazione od ottimizzazione di strutture o di operazioni.

5a. Calibrazione di un termometro

Una memoria FLASH, in particolare multilivello, puo' essere dotata di un sensore di temperatura (termometro) in grado di fornire al controller una certa configurazione di n bits (termobits) che in modo lineare rileva il valore della temperatura assoluta (T). Il controller prima di ogni algoritmo acquisisce questo valore di temperatura e puo' effettuare compensazioni di vario tipo : secondo una legge memorizzata nel firmware (lineare o non lineare), su varie grandezze (tensione di word line, durate di impulsi) e in varie modalita' operative (lettura o scrittura).



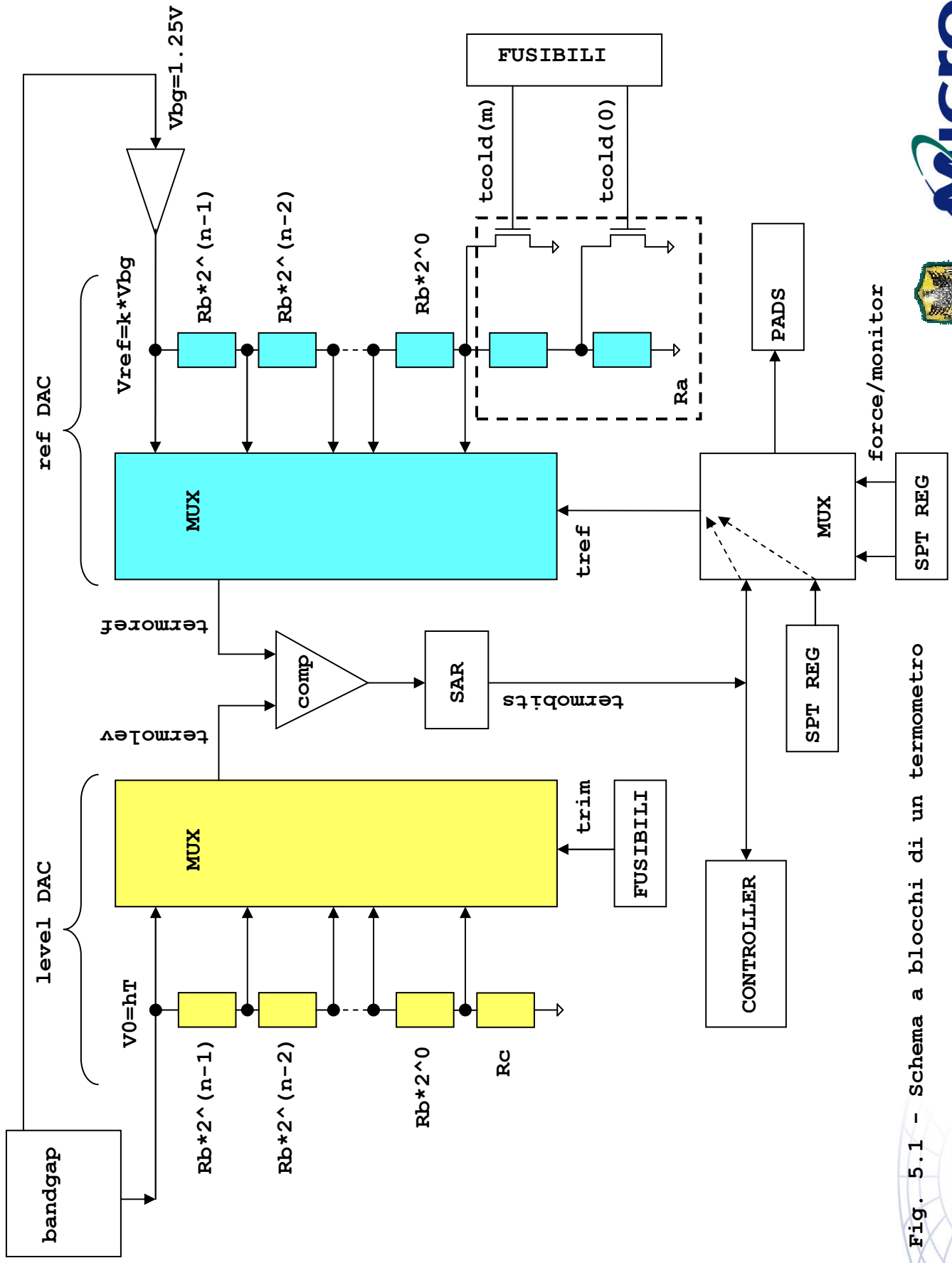


Fig. 5.1 - Schema a blocchi di un termometro



in questo modo si possono rendere le distribuzioni delle soglie VT poco dipendenti dalla temperatura. Uno schema a blocchi di un termometro e' riportato in Fig. 5.1. Da un punto di vista di test un termometro deve essere fornito di strutture di calibrazione in modo tale da avere una caratteristica di risposta il piu' possibile uniforme da chip a chip. Il funzionamento del termometro puo' essere riassunto nel seguente modo :

- . Un circuito di bandgap fornisce una tensione di riferimento di 1.25V stabile in temperatura e fornisce anche una tensione V0 proporzionale alla temperatura assoluta T secondo un coefficiente h.
- . Come riportato in Fig. 5.2, la pendenza della curva termolev(T) puo' essere variata in accordo ad una certa configurazione degli n bits di trim tra un valore massimo h ad un valore minimo. In corrispondenza ad un certo valore di trim si ha $termolev = h1 * T$.
- . Il valore di bandgap viene amplificato per un valore k ed un DAC di riferimento (ref DAC) permette di variare la pendenza di termoref in funzione di n bits di selezione (tref). Come riportato in Fig. 5.3 nel caso $n=5$, la legge che lega termoref a tref puo' essere una costante (nel caso $Ra \gg Rb$) o puo' essere sensibilmente dipendente da tref (nel caso $Ra=0$).
- . termoref e termolev vengono confrontati da un comparatore (COMP) : se $termoref = / termolev$ un registro ad approssimazioni successive (SAR) varia il valore di tref fino a che $termoref=termolev$. Il valore finale di tref (termobits) per cui $termoref=termolev$ rappresenta il valore binario associato a T secondo un coefficiente che puo' essere arbitrariamente fissato (calibrazione) in modo tale che alla massima temperatura misurabile corrisponda il massimo valore di termobits ($termobits=31$ nel caso $n=5$) e alla minima temperatura misurabile corrisponda $termobits=0$.

Con riferimento alla Fig. 5.4, la procedura di calibrazione si puo' organizzare secondo i seguenti passi :

- . $T=90C$ (T_{max})
- . si varia trim fino a che $termobits=31$ ($C \rightarrow D$)
- . $T=-40C$ (T_{min})
- . si varia tcold fino a che $termobits=0$ ($B \rightarrow A$)

Per applicare la procedura di calibrazione occorre attivare una modalita' di test (monitor) che permette di rilevare ai pads il valore di termobits ed occorre attivare una modalita' di test per variare i fusibili di trim e tcold. Per verificare che la catena ref DAC,COMP, SAR stia lavorando correttamente e' utile poi poter attivare una modalita' di test (force) in grado di forzare il valore di tref indipendentemente da termobits e verificare che termobits sia sempre uguale a tref.



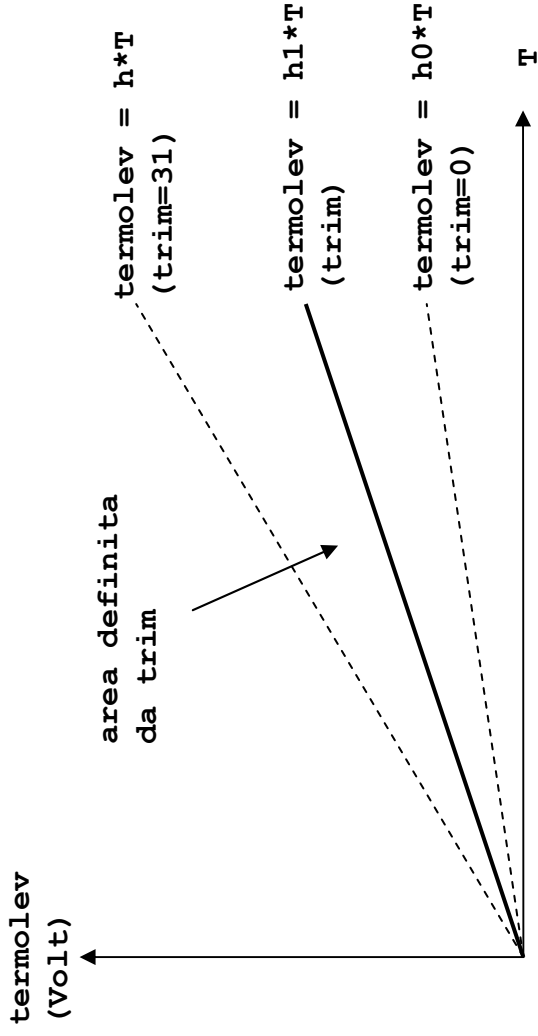


Fig. 5.2 -
Curva di funzionamento
del DAC di trasduzione
di livello (level DAC)
Esempio : n=5

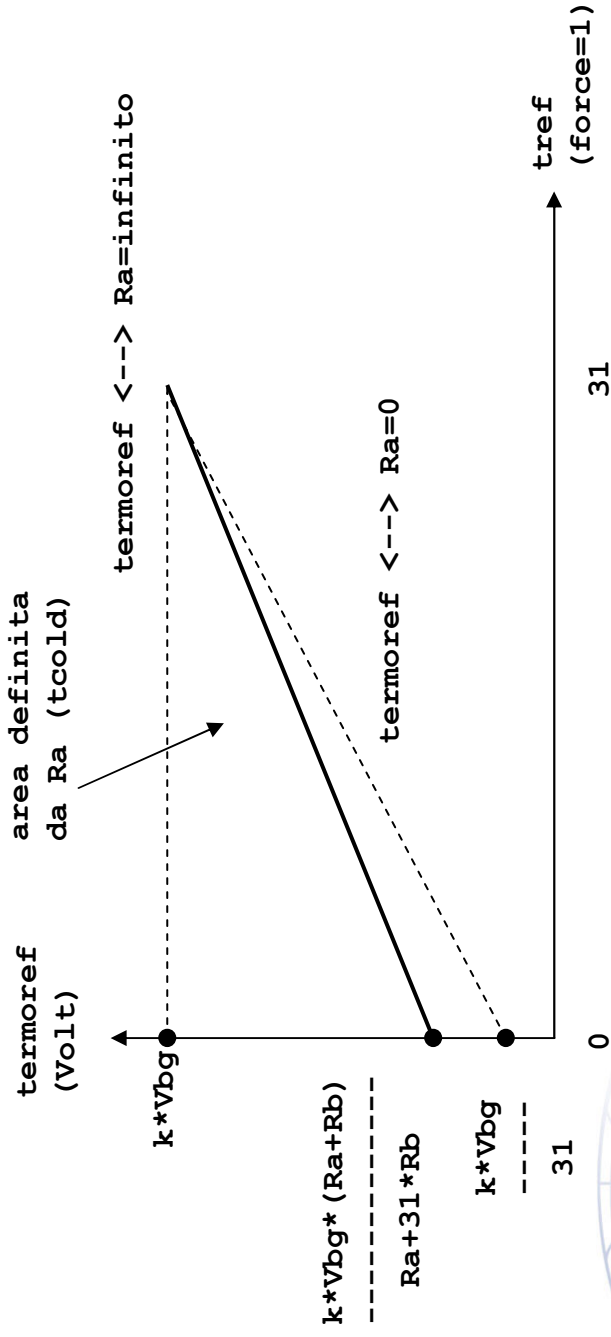
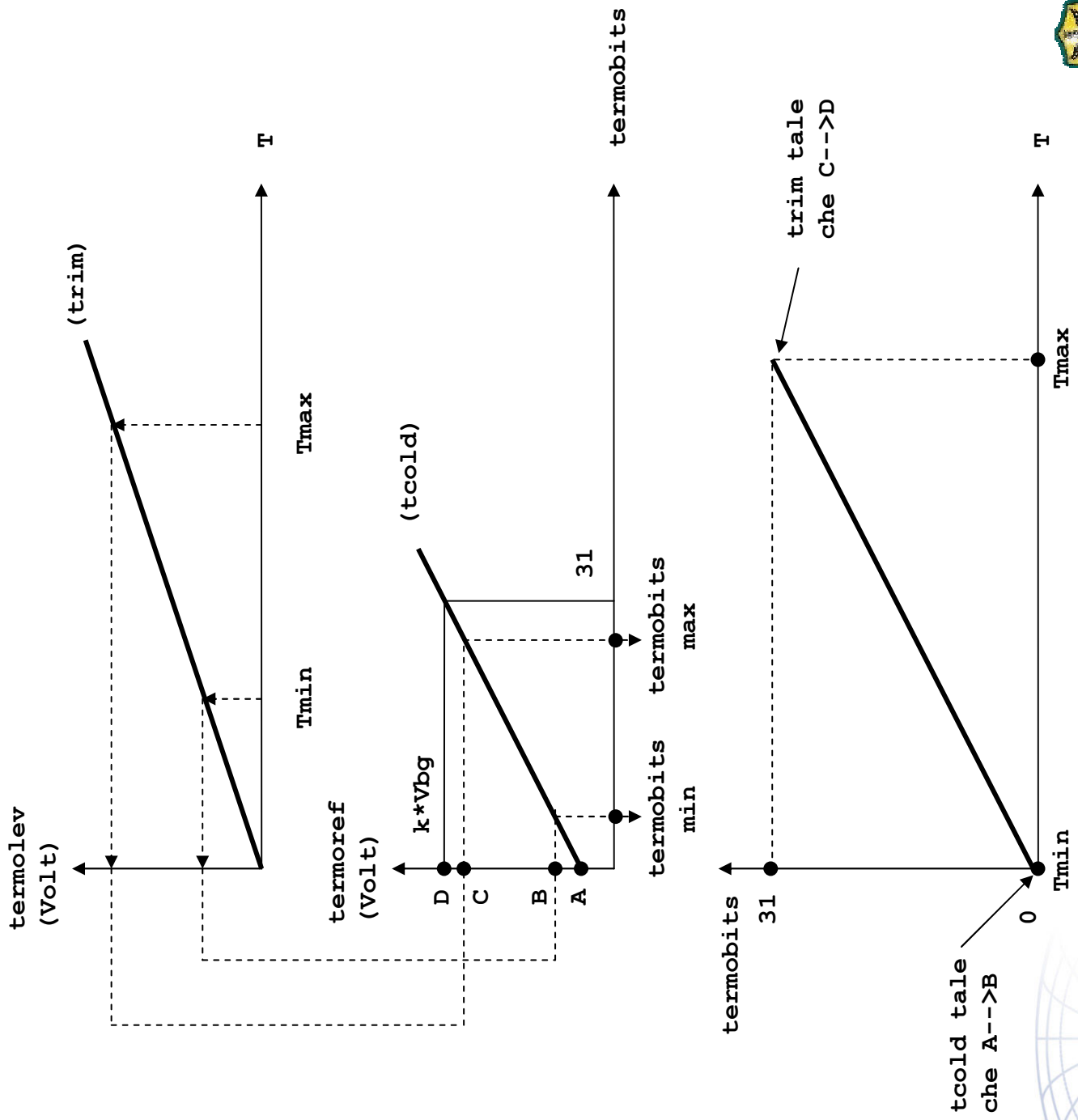


Fig. 5.3 -
Curva di funzionamento
del DAC di trasduzione
del riferimento (ref DAC)
Esempio : n=5



Fig. 5.4 -
Calibrazione
del termometro



5b. Aggiustamento della frequenza di un oscillatore

Nelle memorie FLASH sono presenti degli oscillatori aventi diversi scopi. Tipici esempi sono :

- . oscillatori per la creazione delle fasi di un charge pump
- . oscillatori per la scansione delle istruzioni di un controllore a cui e' demandata l'esecuzione degli algoritmi operativi.

Una caratteristica importante di questi oscillatori deve essere quella di poterne variare la frequenza in base ad un certo numero di fusibili (clk trimming).

Lo scopo di questa variabilita' puo' essere :

- . aumentare la capacita' di un charge pump di fornire corrente anche nel caso di un processo debole
- . diminuire i tempi di esecuzione di algoritmi operativi anche in presenza di processi deboli

Per illustrare con un certo dettaglio come il clk trimming puo' essere effettuato si puo' far riferimento ad uno schema a blocchi di oscillatore come riportato in Fig. 5.5. Uno specchio di corrente puo' generare una corrente di riferimento Iref piu' o meno elevata a seconda del valore della resistenza :

$$R = R3 + \overline{\text{trim}(2)} * [R2 + \overline{\text{trim}(1)} * (R1 + \overline{\text{trim}(0)} * R0)]$$

collegata al catodo del diodo D1. Tale corrente viene poi specchiata nelle correnti di alimentazione Irefp e Irefn che attraversano gli inverters costituenti l'oscillatore. Se le correnti di alimentazione sono basse, la frequenza e' bassa altrimenti la frequenza e' elevata. Per effettuare l'operazione di variazione della frequenza di un oscillatore, occorre attivare una modalita' operative di test (monitor) che permette di rilevare ai pads la forma d'onda di uscita dell'oscillatore ed occorre attivare una modalita' di test per variare i fusibili di trim.



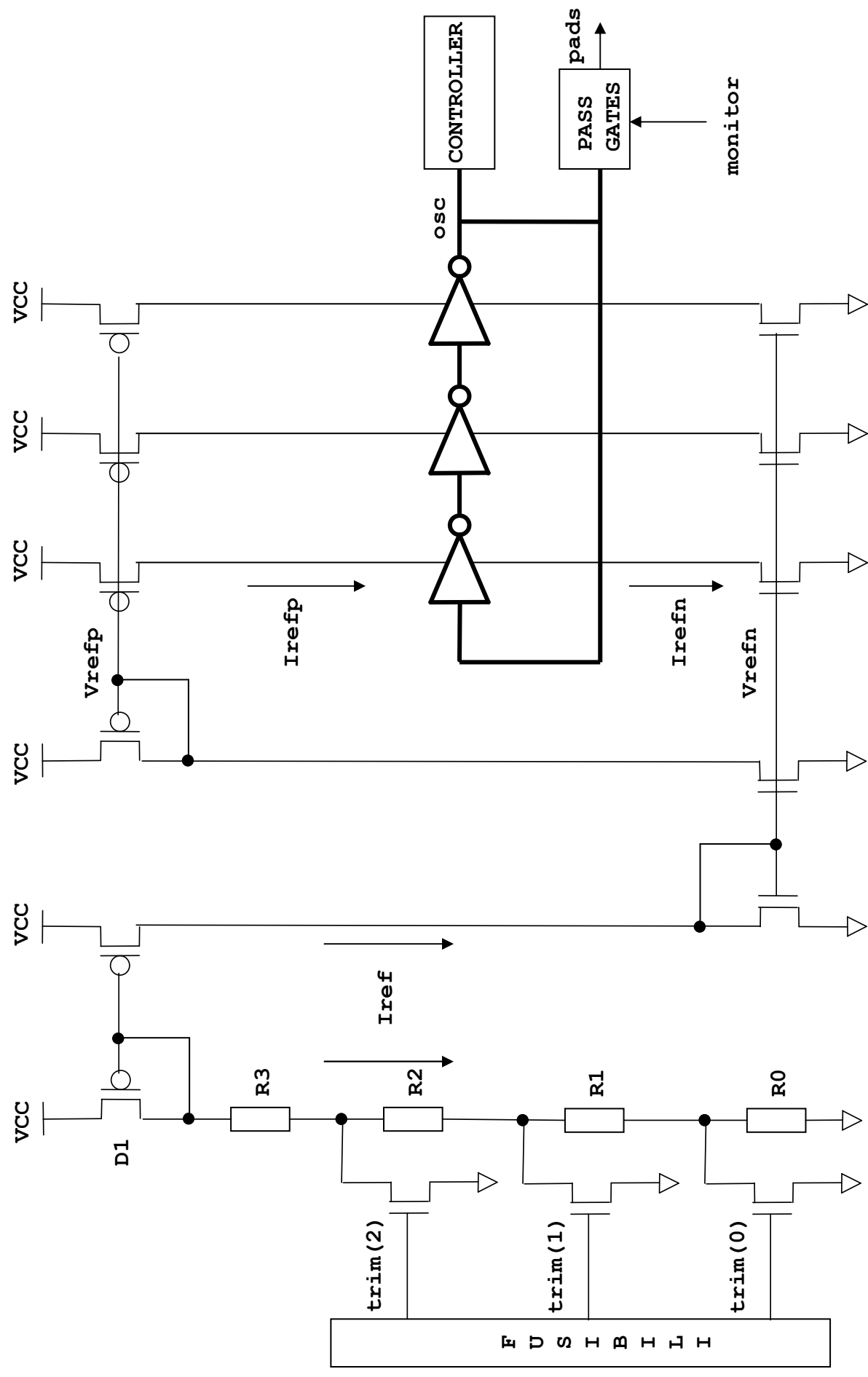


Fig. 5.5 - Variazione della frequenza di un oscillatore



5c. Trim della tensione di word line (WL) in un algoritmo di programmazione

La tensione di WL in un algoritmo di programmazione deve poter variare da un valore alto, per esempio 20V durante l'impulso di programmazione, ad un valore piu' basso, per esempio 5V durante la fase in cui si verifica dove si e' portata la distribuzione della tensioni di soglia dopo ogni impulso di programmazione.

Il valore ottimo di WL nella fase di programmazione e' frutto di compromesso: non deve essere troppo basso altrimenti l'operazione di programmazione e' troppo lenta e non deve essere troppo alto altrimenti si rischia di portare le soglie delle celle troppo al di sopra di cio' che si vuole, invadendo in tal modo delle aree riservate ad altre distribuzioni. Una volta trovato, tale valore ottimo VG di tensione di programmazione viene memorizzato in appositi fusibili.

Analogo discorso vale per il valore della tensione di verifica: il valore a cui si deve spingere la distribuzione delle tensioni di soglia delle celle programmate non puo' essere troppo basso altrimenti si ha poco margine tra celle cancellate e celle programmate e non puo' essere troppo alto altrimenti si erode margine verso il massimo limite di soglia permesso dalla tecnologia. Una volta trovato, tale valore ottimo VF della tensione di verifica viene memorizzato in appositi fusibili.

Diverse soluzioni circuitali possono essere adottate nel progettare un circuito che realizzi il trim delle WL in un algoritmo di programmazione. In Fig. 5.6a e' riportato un circuito compatibile con una organizzazione modale. Un comparatore (COMP) effettua un confronto tra una tensione di riferimento vref ed una tensione vpart generata da un partitore resistivo e proporzionale alla tensione vh il cui valore si vuole regolare. A seconda della modalita' di programmazione (pgm) o di verifica (vfy) si abilita un differente partitore resistivo ognuno associato a propri fusibili per la generazione dell'opportuno rapporto di partizione. A seconda del risultato del confronto tra vpart e vref, un transistor TP viene posto ON oppure OFF e l'alimentazione hv viene abilitata ad alimentare la WL selezionata o meno: il valore vh viene regolato intorno al valore VG oppure VF.



In Fig. 5.7 e' riportato un circuito compatibile con una organizzazione a registri. Una pompa di alta tensione fornisce l'alimentazione v_h alla WL selezionata. Tale pompa ha una uscita regolata tramite un comparatore (COMP) che effettua il confronto tra il valore v_h che si vuole ottenere ed un riferimento fornito da un controllore attraverso un DAC. Il controllore, a seconda della particolare fase dell'algoritmo (programmazione o verifica) preleva l'opportuno valore da un insieme di fusibili e lo applica al DAC.

L'organizzazione a registri e' molto flessibile e con l'uso di un hardware molto semplice nei circuiti analogici permette di realizzare algoritmi molto sofisticati come riportato nella forma d'onda di Fig. 5.7. Tale forma d'onda illustra il caso di una memoria multilivello con impulsi di programmazione a rampa (VGS, VGS+DVG, VGS+2*DVG, ...), per i quali si puo' effettuare il trim del valore iniziale e dell'incremento, e diversi valori di verifica (V1, V2, V3) per le diverse campane di distribuzione delle tensioni di soglia.



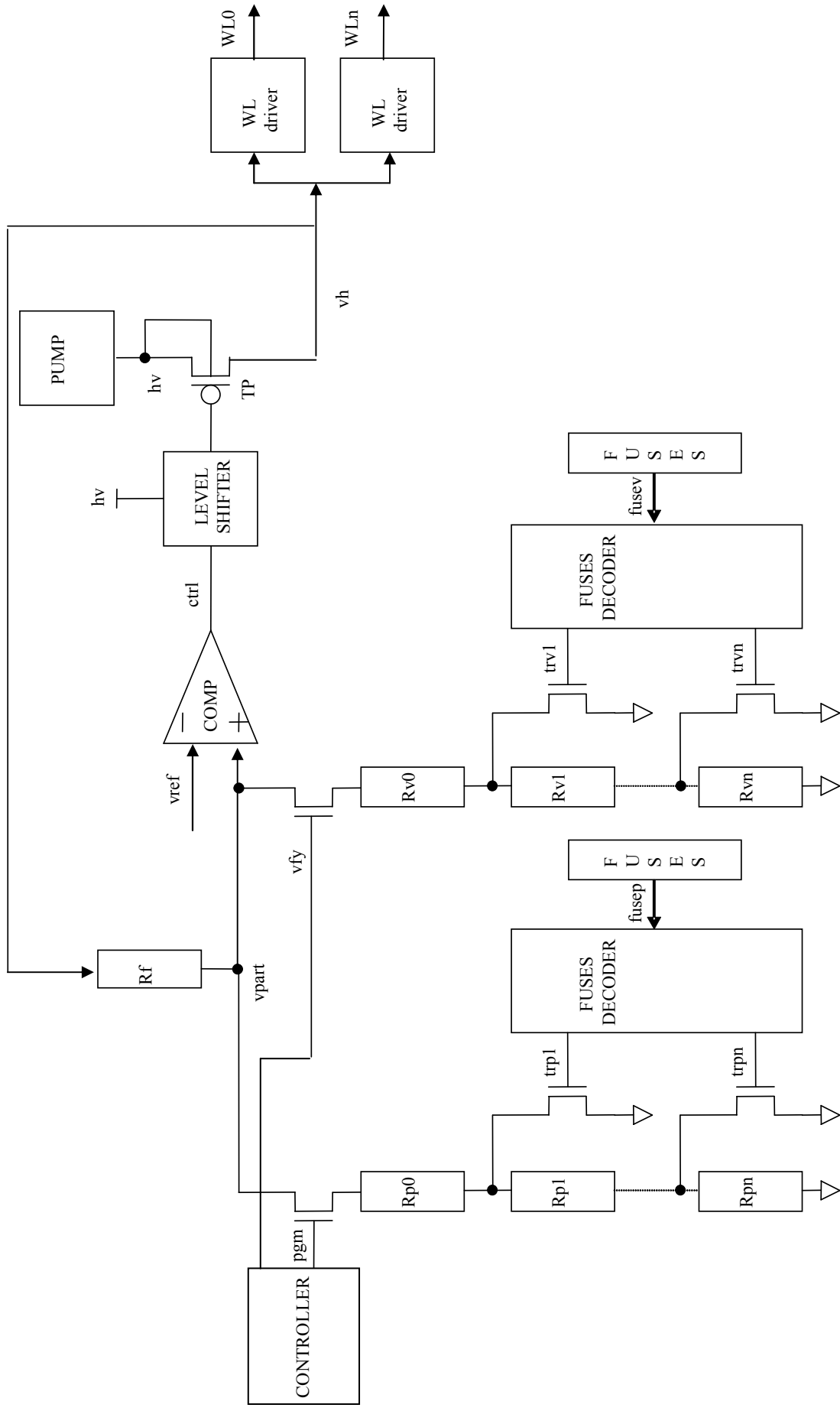


Fig. 5.6a - Trim della tensione di WL in un algoritmo di programmazione (organizzazione modale)



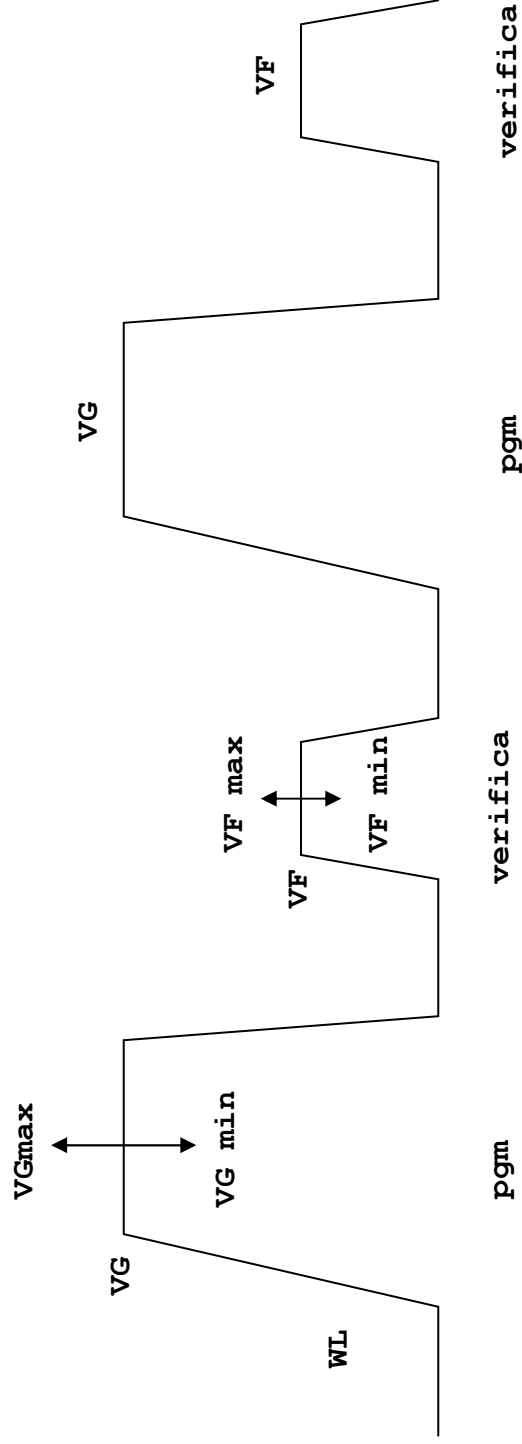


Fig. 5.6b - Forme d'onda della tensione di WL in un algoritmo di programmazione (organizzazione modale - FLASH singolo livello)



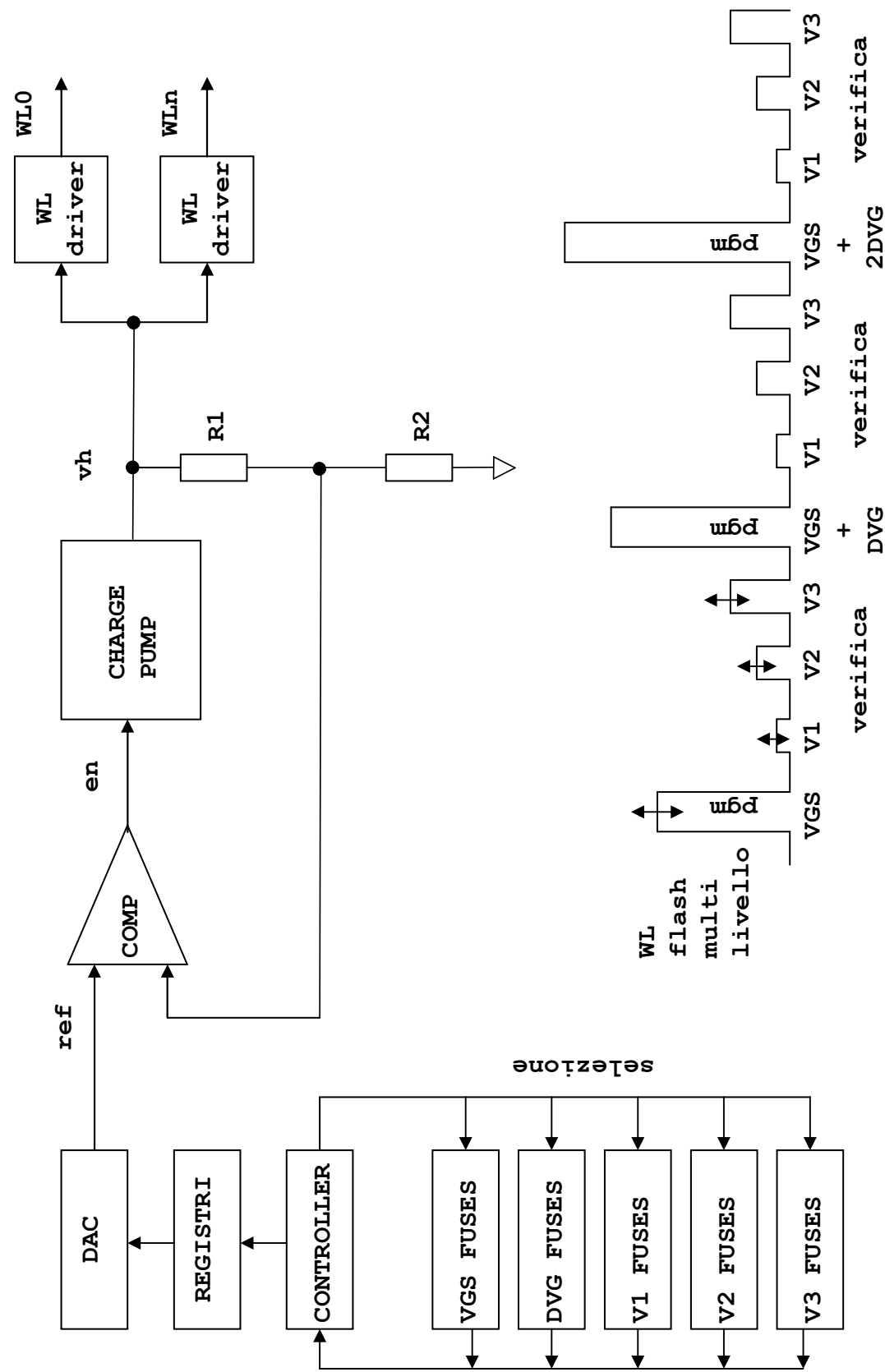


Fig. 5.7 - Trim della tensione di WL in un algoritmo di programmazione (organizzazione a registri)



5d. Trim della durata di impulsi

Le operazioni di programmazione e cancellazione di una memoria FLASH consistono nell'applicare opportune condizioni di polarizzazione alla cella EEPROM selezionata tali da provocarne la programmazione o la cancellazione. Queste operazioni sono eseguite da algoritmi durante i quali si misurano gli effetti dei singoli impulsi e si procede con l'operazione fino a che si raggiunge un risultato soddisfacente.

Ogni singolo impulso di programmazione o di cancellazione ha una durata fissa ottimizzata per raggiungere il risultato con la desiderata accuratezza e con il minimo numero di iterazioni dell'algoritmo. Valore tipico per la durata di un impulso di programmazione e' 5-15 microsecondi mentre per un impulso di erase e' di 1 millisecondo.

Variazioni di processo e affinamenti dell'algoritmo richiedono di avere questi valori tipici aggiustabili in un certo intervallo intorno al valore di default. Un modo di aggiustare la larghezza degli impulsi compatibilmente con una organizzazione modale e' basato su un generatore di impulsi controllato da fusibili come riportato in Fig. 5.8.

Il generatore di impulso e' composto da un contatore avente n stadi di flip-flop che dividono la frequenza a meta' e un oscillatore free-running che genera un clock applicato al primo stadio del contatore. Il numero di stadi del contatore deve essere abbastanza elevato per raggiungere il massimo valore della larghezza di impulso. Un circuito logico e' progettato per generare il desiderato numero di larghezze di impulso t_1, \dots, t_m a partire dalle uscite q_1, \dots, q_n dei flip-flop. Un decoder di fusibili genera un egual numero di segnali di abilitazione en_1, \dots, en_m che vengono applicati alle pass gates di un multiplexer per selezionare l'opportuno valore di larghezza di impulso. Un circuito logico combina la selezionata larghezza di impulso con l'associata condizione operativa e genera un unico segnale di stop. Il circuito per la generazione della larghezza di impulso riceve il segnale di start da una macchina a stadi interna all'inizio di una operazione di programmazione o di cancellazione e invia un segnale di stop alla macchina a stadi per indicare che l'operazione e' stata completata.



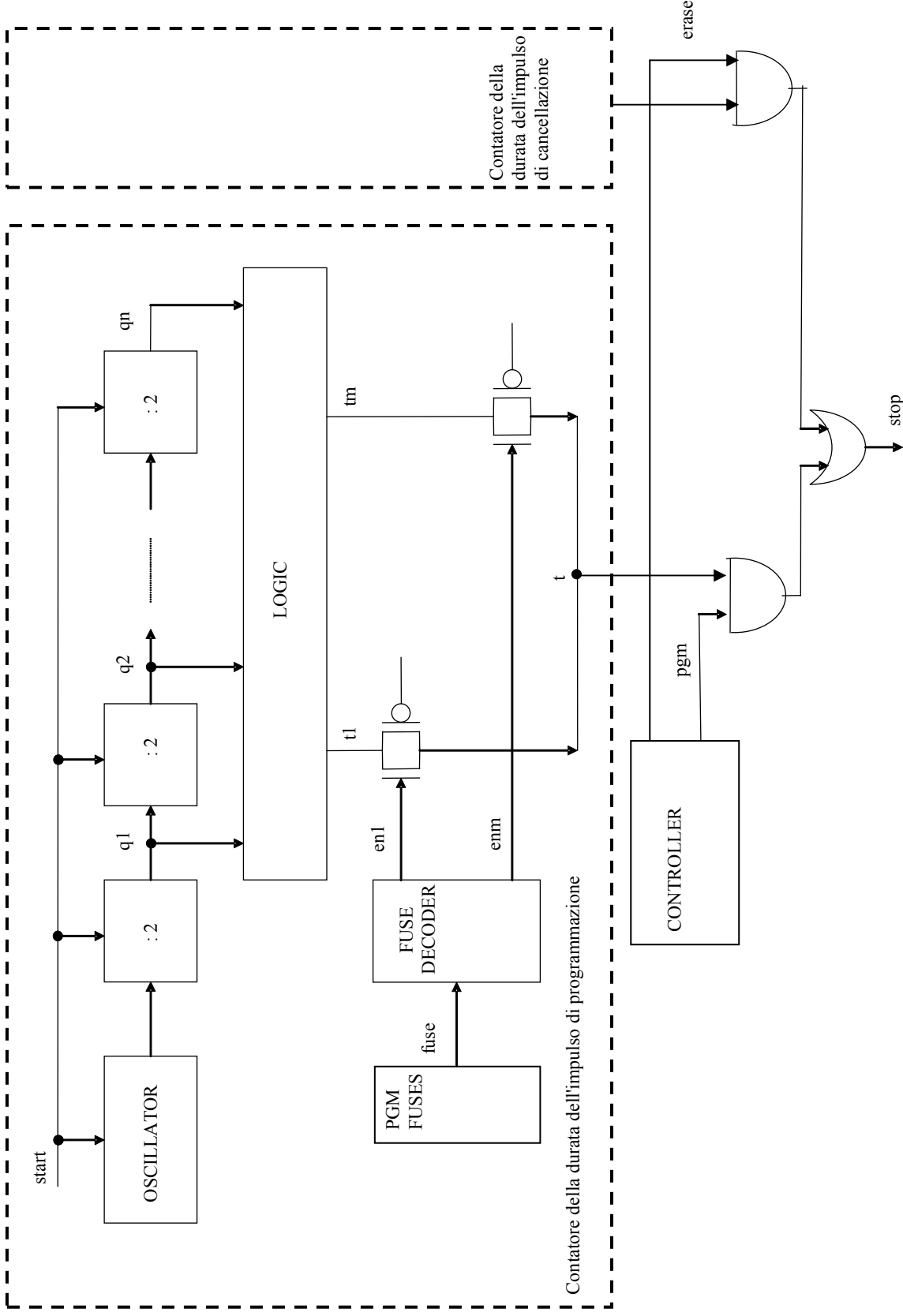


Fig. 5.8 - Trim della durata di impulsi di programmazione e di cancellazione (organizzazione modale)



Una organizzazione a registri non richiede hardware specifico per gestire la durata degli impulsi. Infatti, con riferimento alla Fig. 5.7, specifici fusibili associati alla durata dell'impulso di programmazione o di cancellazione sono disponibili per il controller. Il controller applica l'opportuno valore di ampiezza dell'impulso al DAC dopo di che si mette in attesa per un tempo dettato dal contenuto dei fusibili di durata dell'impulso. Alla fine di tale attesa il controller scrive 0 nel DAC dando luogo al completamento dell'impulso.

6) ALGO SKIPS

Le operazioni di cancellazione o scrittura di una memoria FLASH sono operazioni complesse costituite da numerose operazioni elementari organizzate in un flusso (algoritmo; algo). Durante l'esecuzione di questi algoritmi operativi in modalita' utente, alcune delle operazioni elementari possono essere soggette a varianti saltando dei passi (skip) o seguendo dei percorsi alternativi.

Tipici esempi di questo controllo del flusso operativo in modalita' utente sono :

- . Condizionamento del flusso per seguire una modalita' di programmazione a singolo livello (SIC) o multilivello (MLC).
- . Eliminazione della lunga fase di pre-program prima di un impulso di cancellazione.
- . Fermare una operazione di cancellazione quando la parte alta di una distribuzione cancellata ha raggiunto un certo valore (almeno una stringa di una memoria FLASH NAND ha raggiunto un valore minimo di cancellazione) oppure quando la parte bassa di una distribuzione cancellata ha raggiunto un certo valore (tutte le stringhe di una memoria FLASH NAND hanno raggiunto un valore minimo di cancellazione).
- . Adottare o meno tecniche di rallentamento (slow program) quando il valore di una distribuzione si e' avvicinato al valore di verifica desiderato.
- . Variare il numero di iterazioni di loop negli algoritmi in modo da condizionare il tempo massimo di permanenza di un algoritmo in una certa fase operativa (time-out).

Un possibile modo di organizzare un algo skip in modalita' utente e' riportato in Fig. 6.1.



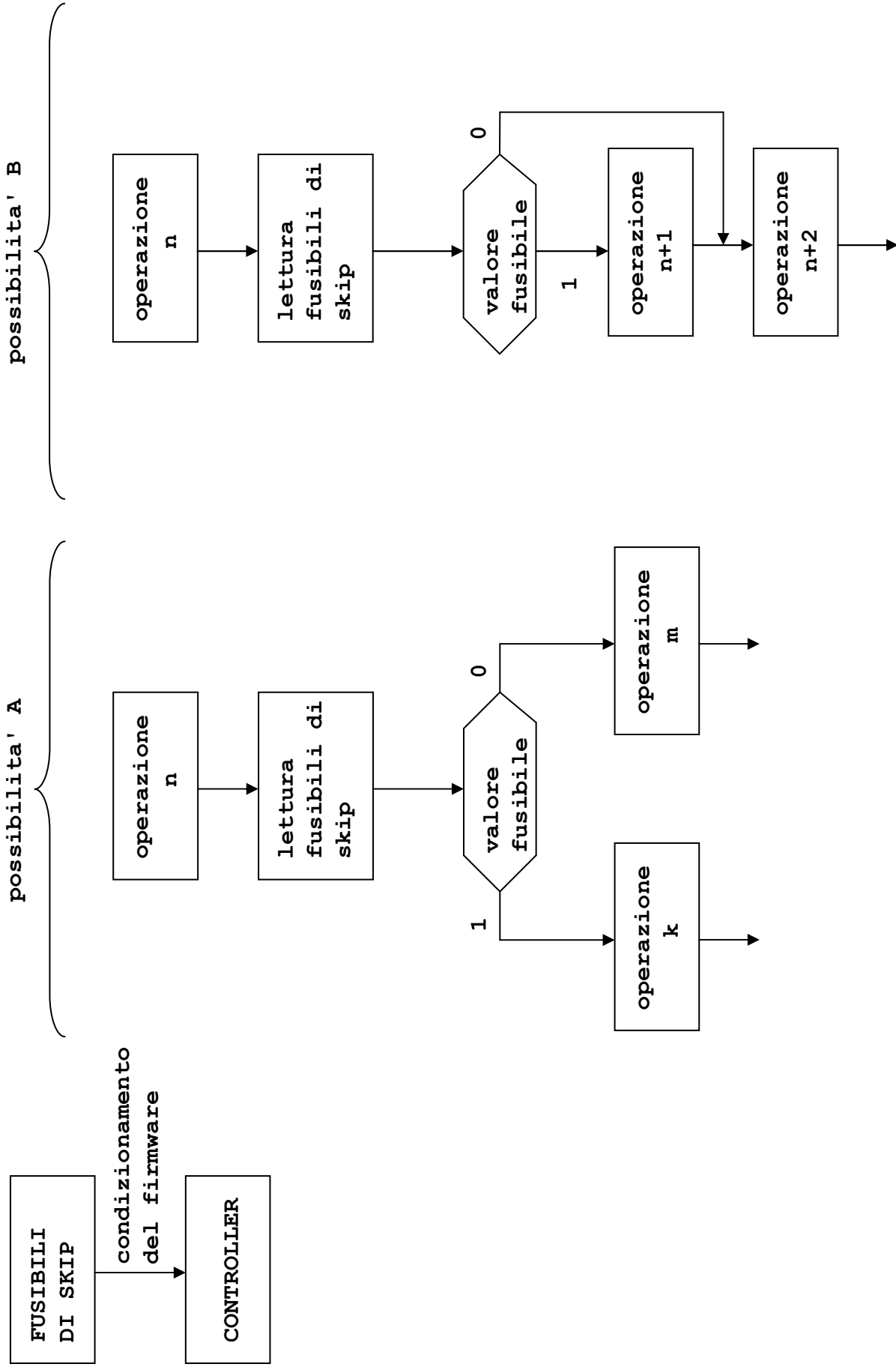


Fig. 6.1 - Organizzazione di un algo skip



Oltre che in modalita' utente, l'algo skip e' interessante anche in modalita' di test.

Per esempio in test modale si potrebbe effettuare lo skip di tutte le operazioni di un algoritmo di programmazione ad eccezione del solo impulso di programmazione per effettuare uno stress dell'array. L'algo skip non ha senso in modalita' di test in un chip organizzato a registri. In questo caso infatti ogni singolo passo del test e' effettuato dalla macchina di test che scrive direttamente il valore di ogni segnale di attuazione nei registri. L'algoritmo non ha bisogno di condizionamenti perche' di volta in volta si scrive nella macchina di test la specifica versione del flusso di test.

7) MONITOR E FORZAMENTO DI TENSIONI

A causa di variazioni del processo le tensioni generate da charge pump interni usate in operazioni di lettura, programmazione o cancellazione possono variare in modo tale da alterare le performances finali del chip sia provocando errori, sia producendo operativita' inefficienti a causa di un aumento del numero di iterazioni di un algoritmo. Per questa ragione sono stati progettati dei circuiti per effettuare il monitor della tensione VREAD applicata alla gate delle celle in lettura, o il monitor della tensione HV applicata alla gate delle celle durante operazioni di programmazione o di verifica, o il monitor della tensione HC applicata al drain delle celle in operazioni di programmazione o al source durante operazioni di erase. Questa operazione di monitor e' ottenuta attivando un modo di test per connettere ognuna delle tensioni menzionate ad un pad esterno.

Un'altra utile modalita' e' simile al monitor ma viene realizzata nella direzione opposta. E' possibile infatti disabilitare una pompa interna e forzare la tensione alla sua linea di uscita applicando una tensione esterna ad un pad del chip. Questa operazione di forzamento puo' essere ottenuta attivando un modo di test per connettere ad un pad esterno una qualsiasi delle tensioni da forzare. Il forzamento delle tensioni dall'esterno puo' essere utile per fornire corrente alle bit lines per rendere piu' veloci le operazioni di programmazione o di cancellazione o per avere la capacita' di attivare operazioni di programmazione o cancellazione in parallelo su piu' parole o piu' blocchi durante test di produzione. Puo' essere anche usato per applicare sulla gate delle celle una forma d'onda a rampa per misurare la tensione di soglia reale di una cella.



Una possibile architettura per realizzare il monitor delle tre tensioni VREAD, HV, HC e la funzione di forzamento e' riportata in Fig. 7.1. Queste tre operazioni di monitor sono attivate da tre differenti modi di test : il monitor di VREAD e' operativo quando il segnale di test tvread e' attivo, il monitor di HV e' associato al segnale di test thvmon, e il monitor di HC e' associato al segnale di test thcmon. Questi segnali di test che sono attivati in modo mutuamente esclusivo possono forzare gli switches ad essere ON e connettere le pompe ad un pad esterno.

In Fig. 7.1 e' anche riportata la possibilita' di forzare le linee VREAD, HV, HC attivando i segnali di test txceiv_vread, txceiv_hv, txceiv_hc. Questi segnali di test attivati in modo esclusivo possono forzare gli switches ad essere ON disattivando contemporaneamente la pompa corrispondente. Le linee VREAD, HV, HC possono essere forzate connettendo un generatore al pad esterno. I drivers di word lines o di bit lines possono quindi ricevere la loro alimentazione da una sorgente interna o da una sorgente esterna.

Un modo per realizzare la funzione di switch e' riportato in Fig. 7.2. Quando non e' attivo ne' un modo di test di monitor ne' un modo di test di forzamento, i generici segnali di modo di test tmon e txceiv sono entrambi bassi. Il level shifter LS1 forza il pass transistor P1 ad essere OFF connettendo la sua gate al pad e il level shifter LS2 forza il pass transistor P2 ad essere OFF connettendo la sua gate ad HV : la pompa e' attiva ed il driver puo' ricevere la alimentazione HV direttamente dalla pompa. Quando tmon e' attivo, il level shifter LS1 forza il pass transistor P1 ad essere ON applicando 0V alla sua gate e il level shifter LS2 forza il pass transistor P2 ad essere ON applicando 0V alla sua gate : la pompa e' abilitata e la tensione HV puo' essere applicata al driver e puo' essere anche letta dall'esterno. Quando txceiv e' attivo, P1 e P2 sono entrambi ON mentre la pompa e' disabilitata. In questa condizione l'ingresso HV al driver puo' essere forzato dall'esterno senza influenzare la pompa che ha il diodo in stato di OFF.



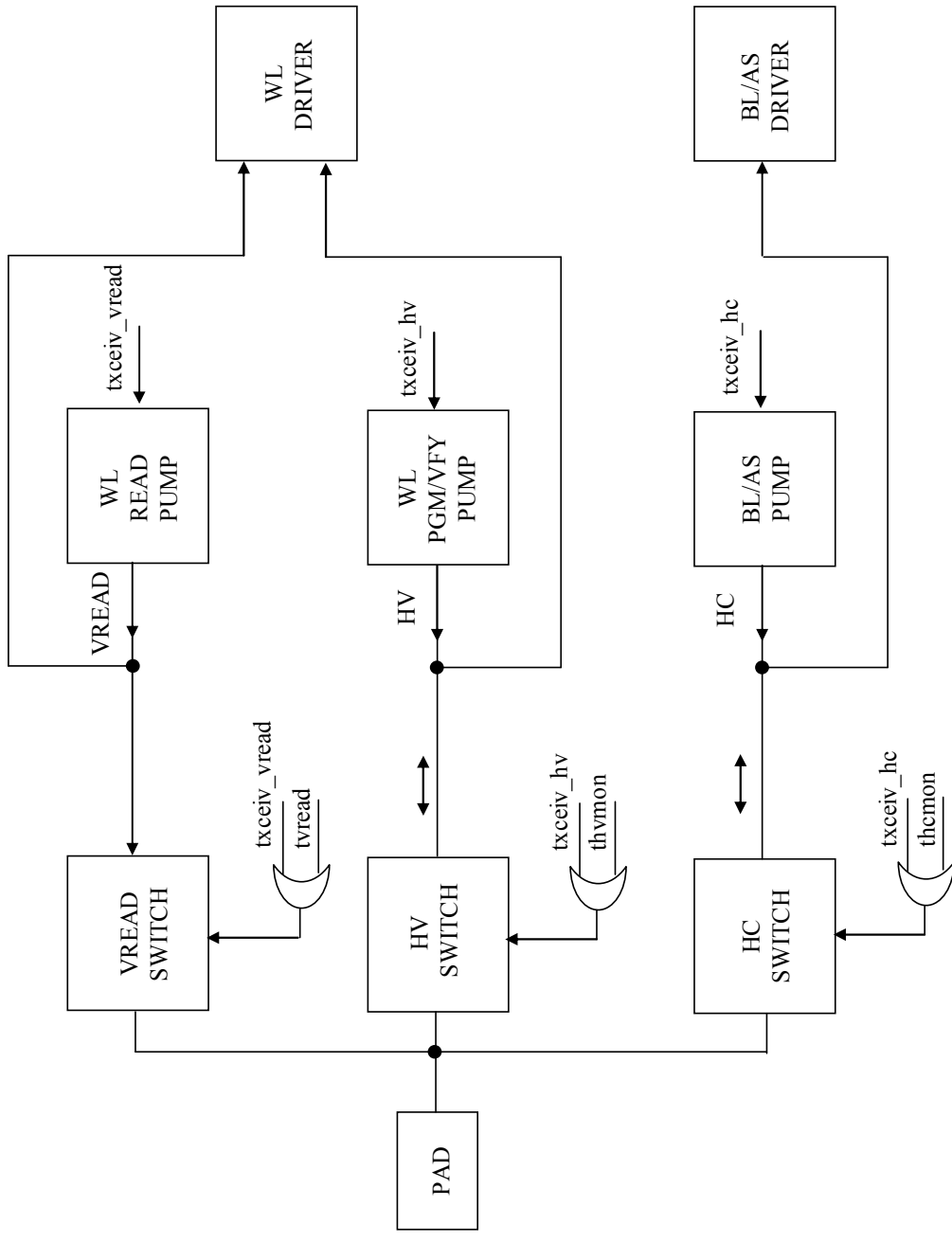


Fig. 7.1 - Schema a blocchi di un forzamento e monitor di un'alta tensione.



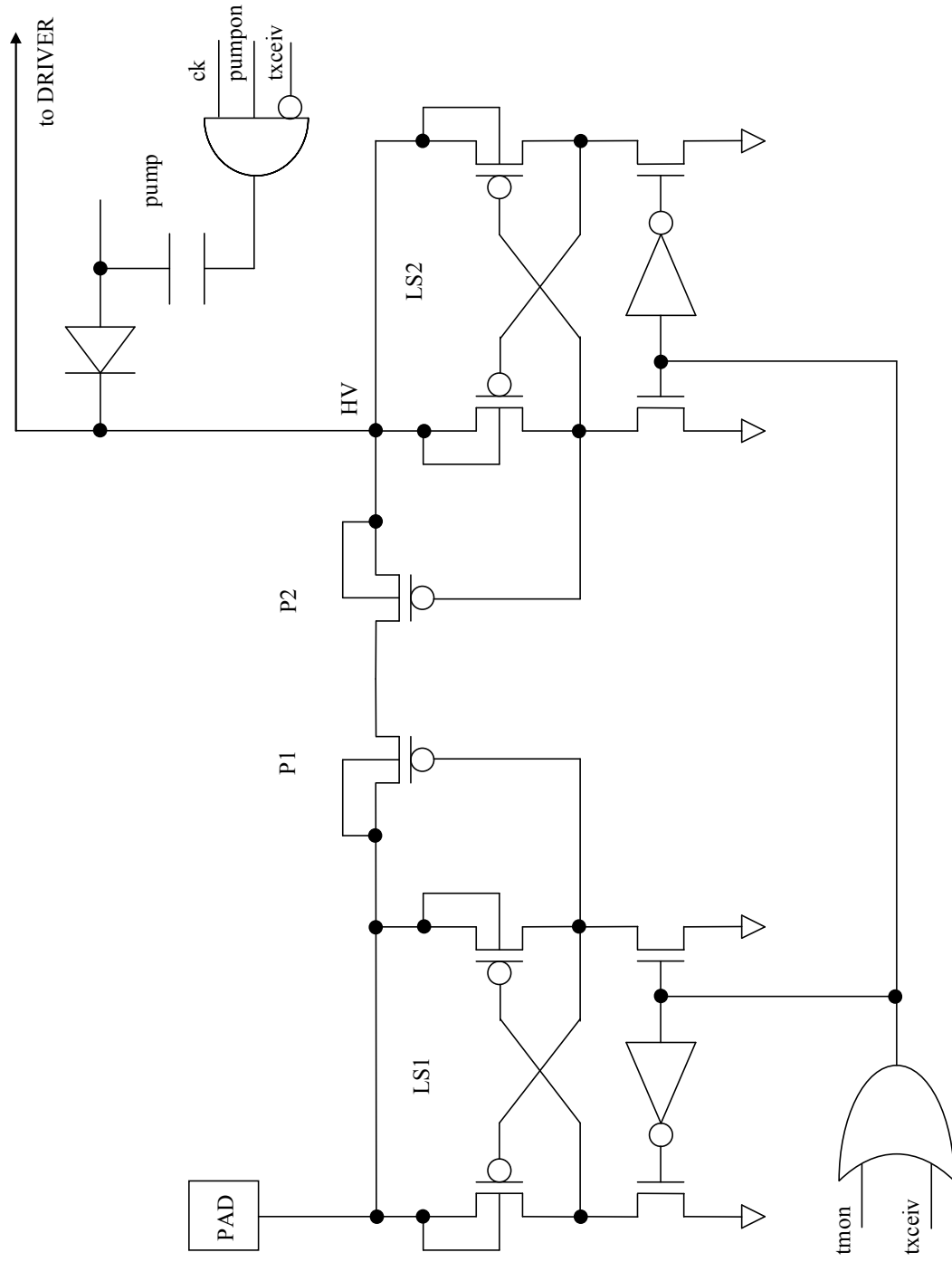


Fig. 7.2 - Circuito di switch per un forzamento e monitor di un'alta tensione.



8) FORZAMENTO DELLE DURATE DI IMPULSI

Una possibilita' molto interessante in modalita' operativa di test e' quella di poter far durare a piacimento delle singole operazioni elementari quali ad esempio impulsi di cancellazione o di programmazione. Questa possibilita' e' molto utile durante la fase di stress di una memoria FLASH in cui si vuole sottoporre il circuito ad un impulso di cancellazione o programmazione per un tempo molto piu' lungo (ordini di grandezza) della modalita' operativa utente. Un esempio di come poter organizzare il forzamento esterno della durata di impulsi in un test modale e' riportato in Fig. 8.1.

Il modo di test per il forzamento esterno della durata di impulsi (textw) viene attivato da una opportuna sequenza. Una modalita' operativa di cancellazione o di programmazione viene attivata da un comando con gli tutti gli skip attivati ad eccezione dell'impulso attivo. Il fronte negativo di un segnale di controllo da' inizio all'impulso (start) ed il fronte positivo pone termine all'impulso (stop). Un MUX permette di isolare il contatore di impulsi della modalita' utente e di rendere attiva la logica di controllo per le durate esterne.

In un ambiente di test organizzato a registri, la circuiteria di forzamento esterno delle durate non ha senso perche' ogni singolo passo del test e' effettuato dalla macchina di test. I segnali di start e stop non esistono e la durata dell'impulso e' stabilita con una attesa tra quando la macchina di test scrive nei registri il valore dei segnali per l'attivazione dei valori analogici associati all'impulso a quando la macchina di test scrive nei registri i valori dei segnali per disattivare i valori analogici associati all'impulso.



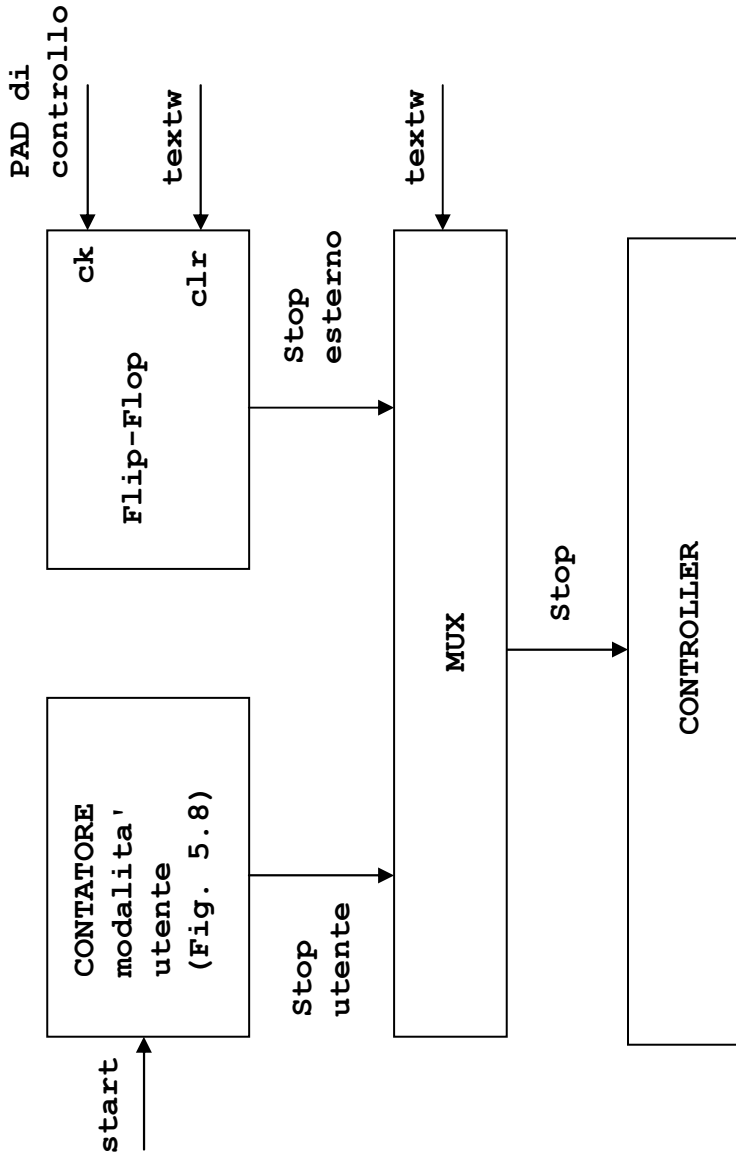
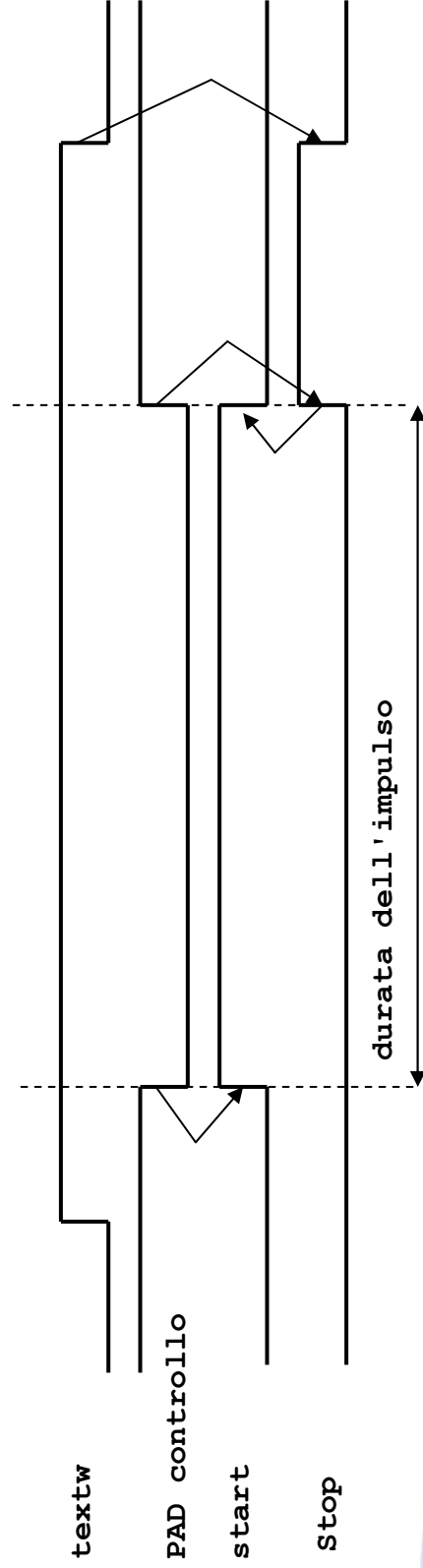


Fig. 8.1 - Forzamento esterno delle durate di impulsi in test modale



9) ACCESSO DIRETTO IN ARRAY

L'accesso diretto in array e' uno dei punti fondamentali nel test di memorie FLASH in quanto esso permette di misurare la tensione di soglia VT di una cella e di ottenere la distribuzione di VT di gruppi di celle. L'accesso diretto in array (BLA = Bit Line Access) e' un modo di test i cui punti essenziali, per quanto riguarda l'hardware e le forme d'onda, sono riportati in Fig. 9.1.

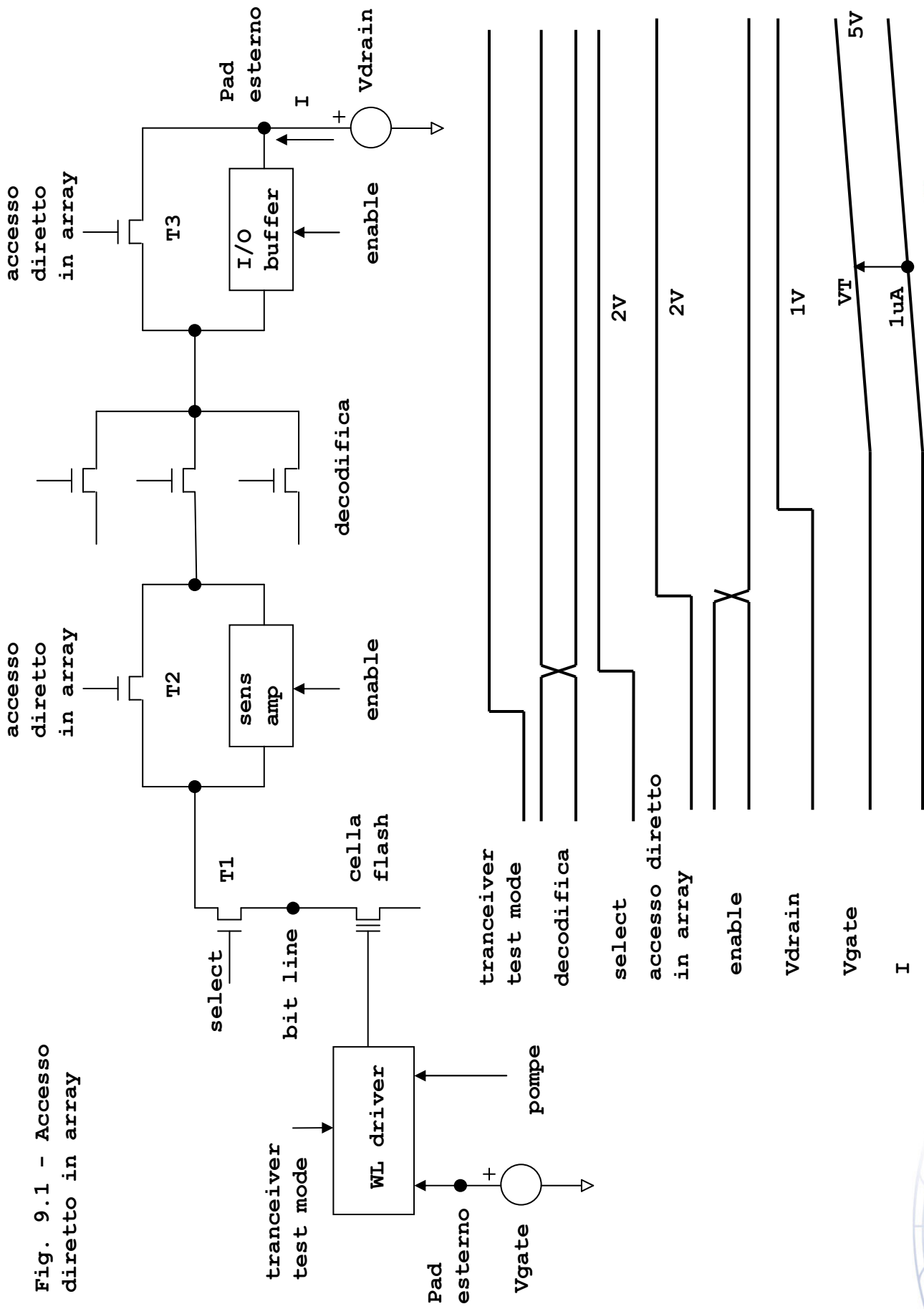
Il percorso regolare per la scrittura e lettura di una cella di memoria FLASH e' costituito dalla cella FLASH, da un eventuale transistor di selezione T1, dal sens amplifier, da circuiti di decodifica, dal buffer di I/O e dal pad I/O di ingresso/uscita. Alcuni di questi circuiti, tipicamente il sens amplifier e il buffer di I/O, possono essere isolati da un segnale di abilitazione che li forzano in alta impedenza.

Quando il modo di test di accesso diretto in memoria e' attivato, il percorso regolare viene isolato (enable = 0) ed un percorso alternativo (T2,T3) viene attivato per connettere direttamente la bit line della cella selezionata al pad di I/O. Applicando una tensione esterna Vdrain al pad di I/O, tipicamente 1V, ed attivando un modo di test (tranceiver test mode) che connette un pad esterno al gate della cella FLASH, e' possibile misurare la sua tensione di soglia VT.

Applicando una rampa di tensione da 0V ad un valore superiore al valore della soglia programmata, si produrrà un passaggio di corrente I crescente man mano che la tensione di gate supera il valore di soglia VT. Per convenzione si definisce il valore di soglia VT come quel valore di tensione di gate corrispondente al passaggio di una determinata corrente I, per esempio 1 uA, quando il drain ha un determinato valore, per esempio 1V.



Fig. 9.1 - Accesso diretto in array



10) MODALITA' DI STRESS

La progettazione della testabilita' di una memoria FLASH deve prevedere una serie di test atti ad effettuare lo stress delle celle di memoria allo scopo di rilevare celle deboli le cui prestazioni non sono accettabili o che hanno una significativa probabilita' di divenire inaccettabili durante la vita del dispositivo.

La distribuzione delle tensioni di soglia VT e' misurata prima e dopo lo stress. La distribuzione delle tensioni VT dopo lo stress e' tipicamente spostata verso valori piu' bassi: per ogni insieme di valori di tensione e durata di stress si puo' fissare un criterio per definire un abbassamento accettabile della distribuzione di VT consistente con cio' che gli ossidi di quel processo possono tipicamente fornire. Se la diminuzione di VT e' anomala, si sospetta un difetto nella struttura delle celle FLASH ed una indagine si rende necessaria.

Ci sono tre tipi di test per effettuare lo stress di celle FLASH : gate stress, source stress, drain stress. Il gate stress e' uno stress dell'ossido di canale realizzato con drain e source connessi a ground e la control gate connessa ad una alta tensione (positiva o negativa). Il source stress e' uno stress che serve a rilevare debolezze nella giunzione di source dove nelle NOR FLASH viene realizzata l'operazione di cancellazione e viene realizzato con la control gate connessa a ground, source ad alta tensione e drain floating. Il drain stress e' uno stress che serve a rilevare debolezze nella giunzione di drain dove nelle NOR FLASH avviene l'operazione di programmazione e viene realizzato con control gate connessa a ground, drain ad alta tensione e source floating.

I circuiti per la realizzazione di drain (source) stress sono realizzati con tecniche tradizionali per forzare tutte le word lines in tutti i settori a ground, per forzare tutti i drain (source) di tutti i settori ad alta tensione e per forzare tutti i source (drain) di tutti i settori in condizioni di floating. Il forzamento delle alte tensioni si puo' realizzare sia con l'uso di pompe interne che con l'uso di modi di test per il forzamento esterno.



Quando fenomeni di 'short channel' non sono coinvolti nelle celle FLASH, le correnti band-to-band durante lo stress e' indipendente dalla lunghezza di canale ed ha lo stesso valore sia se il source (drain) stress e' realizzato con drain (source) floating o connesso a ground. Nel caso di celle FLASH con canale corto si osserva una significativa degradazione dopo lo stress. L'abbassamento di VT e' piu' evidente e si puo' osservare una degradazione di gm per lunghi periodi di stress. Inoltre lo stress e' piu' efficace quando gli stress delle giunzioni laterali sono realizzati con l'altro terminale non flottante ma connesso a ground. In base a queste osservazioni, i modi di stress di celle FLASH a canale corto devono includere piu' modi di test rispetto ai protocolli tradizionali, includendo gli stress delle giunzioni laterali con il secondo terminale floating.

Per lunghezze di canale maggiori, la velocita' di scarica laterale delle correnti di Fowler-Nordheim e' indipendente dalla lunghezza di canale ma per canali molto corti si osserva una velocita' di scarica molto elevata. Questo non solo causa degradazioni nel dispositivo ma anche un ampliamento delle distribuzioni cancellate a causa della variazione delle lunghezze di canale nell'array.

11) TECNICHE DI COMPRESIONE

Prima di spedire chips di memoria ad un cliente, il fabbricante deve testare i chip di memoria per assicurare un alto livello di affidabilita' nell'utilizzo della memoria. Uno dei test realizzati consiste in una operazione di verifica del contenuto scritto. In generale la verifica del contenuto scritto e' usata per verificare la capacita' della memoria di ricevere, mantenere e fornire i dati in modo affidabile. Un tale test tipicamente coinvolge una prima operazione di scrittura di un certo contenuto nell'array di memoria (per esempio tutti i 0 o tutti 1 o una configurazione mista di 1 e 0). L'array di memoria e' successivamente acceduto per leggere i dati memorizzati. I dati sono spesso internamente organizzati in pagine contenenti n parole (words), essendo ogni parola composta da un numero di bits uguale al numero di componenti di I/O. I dati possono essere letti una parola alla volta e confrontati bit a bit con la configurazione di dati originariamente immessi nella memoria. Se ognuno dei bit di ogni parola in lettura e' uguale al corrispondente bit della configurazione scritta, il dispositivo passa il test.



Una differenza in uno qualsiasi dei bit di una qualsiasi parola in lettura rispetto al corrispondente bit scritto e' indicativa di un malfunzionamento. Questo test , consistente nel leggere una parola alla volta, sebbene efficace nel verificare la funzionalita' del circuito puo' essere migliorato, dal punto di vista del consumo di tempo, adottando delle tecniche di compressione.

Due tecniche di compressione verranno prese in considerazione nel seguito :

- a) compressione di parola (word compression)
- b) verifica interna (IVR = Internal Verify Read) .

11a. Compressione di parola

Una tecnica di compressione di parola permette di rilevare un fallimento in un bit in un gruppo di n parole nelle quali e' stata scritta una stessa configurazione di dati. Questa rilevazione e' effettuata con una operazione di lettura di una singola parola. La generica uscita $Q[i]$ sara' 1 se tutte le parole hanno un 1 nella stessa posizione i, oppure sara' 0 se tutte le parole hanno uno 0 nella stessa posizione i, oppure sara' 3-state se nessuna delle due precedenti condizioni e' realizzata. La situazione e' sintetizzata nella tabella riportata in Fig. 11.1.

Se si rileva un fallimento, la posizione del bit fallente entro la parola e' nota ma non c'e' la possibilita' di sapere in quale particolare parola il fallimento e' presente. La riparazione di colonna puo' essere effettuata in due modi : si riparano tutte le colonne corrispondenti alla posizione fallente in tutte le parole, oppure si esce dal modo di test di compressione e si scandiscono singolarmente tutte le parole della pagina fallente riparando poi la colonna nella sola parola fallente.

Un possibile modo di realizzare la compressione di parola e' quello riportato nelle figure da 11.2 a 11.5.



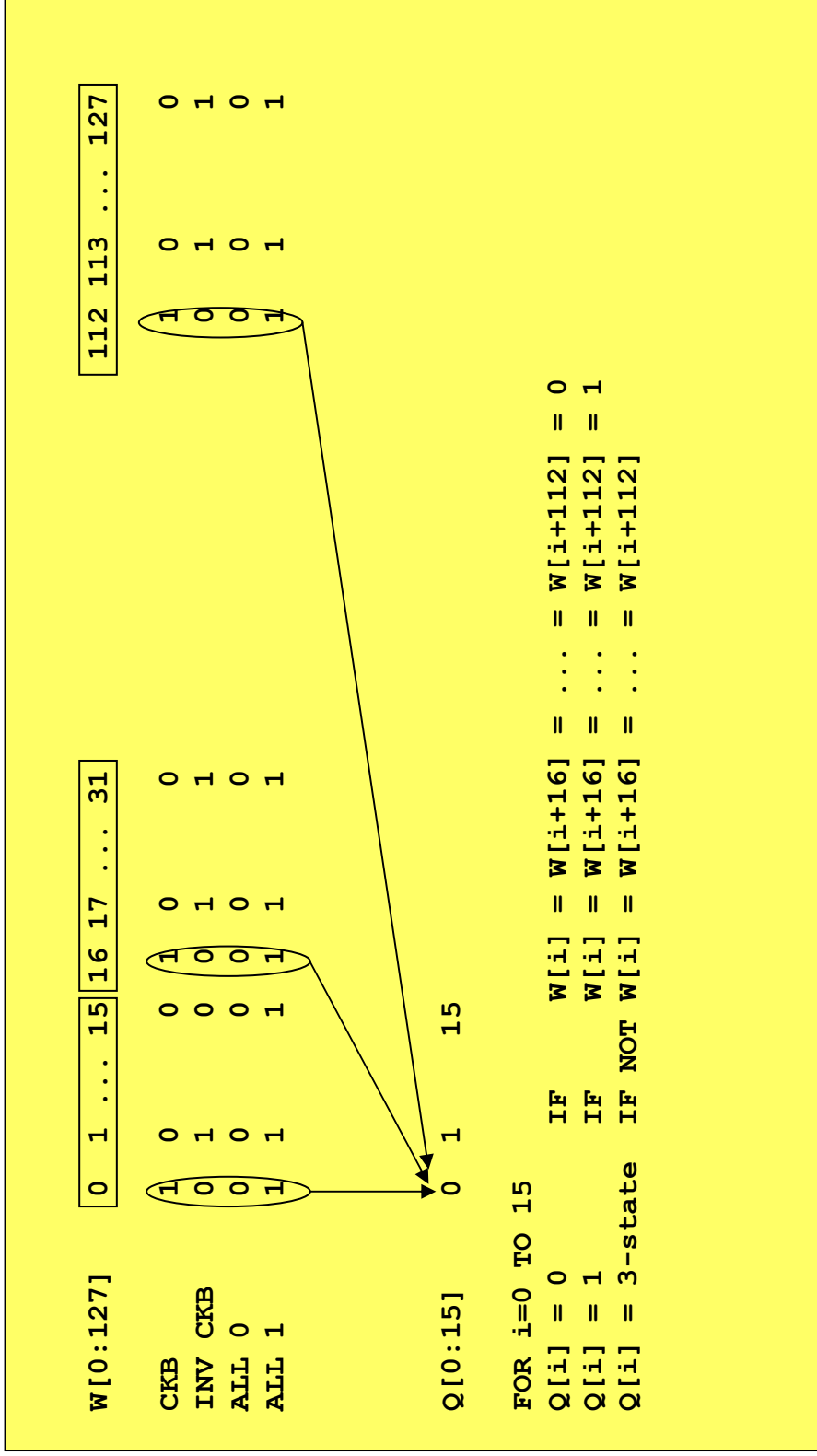


Fig. 11.1 - Tabella illustrativa della compressione di word

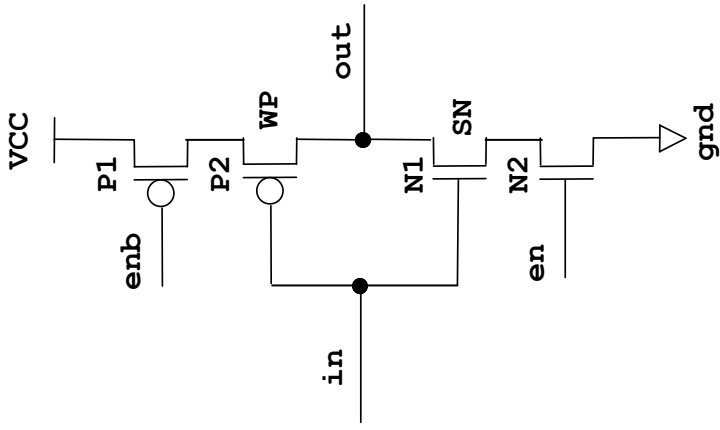


Tale tecnica di compressione di parola e' basata su due blocchi fondamentali chiamati inverter weak P (P debole) e inverter weak N (N debole), come riportato in Fig. 11.2. Gli inverter weak P e weak N sono composti da due transistors Pch, P1 e P2, e da due transistors Nch, N1 e N2. I transistors P1 e N2 sono transistors di abilitazione: quando le loro gates hanno valori $enb=1$ e $en=0$, l'inverter e' disabilitato, cioe' isolato dalle linee di alimentazione VCC e gnd, e l'uscita out non e' pilotata. Quando le loro gates hanno valori $enb=0$ ed $en=1$, l'inverter e' abilitato, cioe' connesso alle linee di alimentazione VCC e gnd, e l'uscita out e' pilotata. Nell'inverter weak P, il transistor P2 ha il W/L molto piu' piccolo del W/L del transistor N1 e per questa ragione il transistor P2 e' chiamato P debole (WP) mentre il transistor N1 e' chiamato N forte (SN); il transistor WP deve pilotare una corrente piu' piccola di almeno n volte di quella che il transistor SN e' in grado di pilotare. Quando l'inverter weak P e' abilitato e $in=1$, l'uscita e' fortemente connessa a gnd. Quando l'inverter weak P e' abilitato e $in=0$, l'uscita e' debolmente connessa a VCC. Nell'inverter weak N, il transistor N1 ha il W/L molto piu' piccolo del W/L del transistor P2 e per questa ragione il transistor P2 e' chiamato P forte (SP) mentre il transistor N1 e' chiamato N debole (WN). Il transistor WN deve pilotare una corrente piu' piccola di almeno n volte di quella che il transistor SP e' in grado di pilotare. Quando un inverter weak N e' abilitato e $in=1$, l'uscita e' debolmente connessa a gnd. Quando un inverter weak N e' abilitato e $in=0$, l'uscita e' fortemente connessa a VCC.

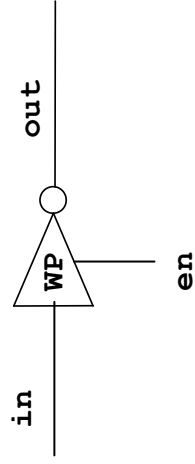
Nella circuiteria di test per la lettura compressa di una memoria, gli inverters weak N e weak P sono connessi come riportato in Fig. 11.3 per rilevare se bits corrispondenti hanno lo stesso valore in parole differenti.

L'i-mo bit di ogni parola e' connesso ad una coppia di inverters weak P e weak N abilitati dal segnale di abilitazione di quella parola. Tutti gli inverters weak P hanno la loro uscita connessa ad una linea comune WP_com e tutti gli inverters weak N hanno la loro uscita connessa ad una linea comune WN_com. Un segnale di abilitazione comune evita condizioni di floating sulle linee WP_com e WN_com. Quando il modo di test di lettura compressa non e' attivo, tutti i segnali di abilitazione della parole $en1$, $en2$, ..., enn sono 0 e tutti gli inverters weak P e weak N sono disabilitati; il segnale en di abilitazione generale e' 0 e le linee WP_com e WN_com sono forzate a ground dai transistors TR1 e TR2.

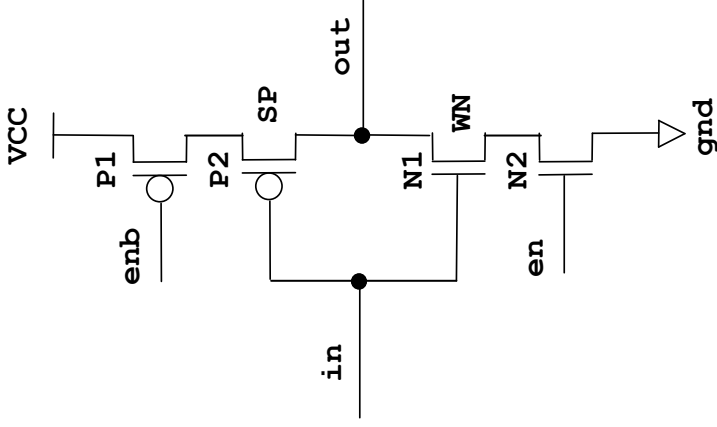




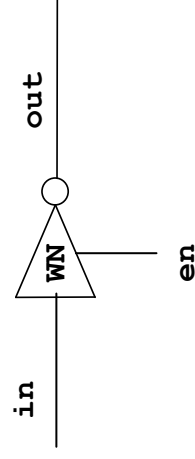
(a) - inverter weak P



(b) - simbolo di un inverter weak P



(c) - inverter weak N



(d) - simbolo di un inverter weak N

Fig. 11.2 - Inverters weak P e weak N usati nella compressione di parola.



Quando il modo di test di lettura compressa e' attivo, il segnale en e' 1 e i transistors TR1/TR2 non forzano le linee WP_com e WN_com a ground; almeno uno degli inverters weak P e weak N e' abilitato e le linee WP_com/WN_com sono pilotate.

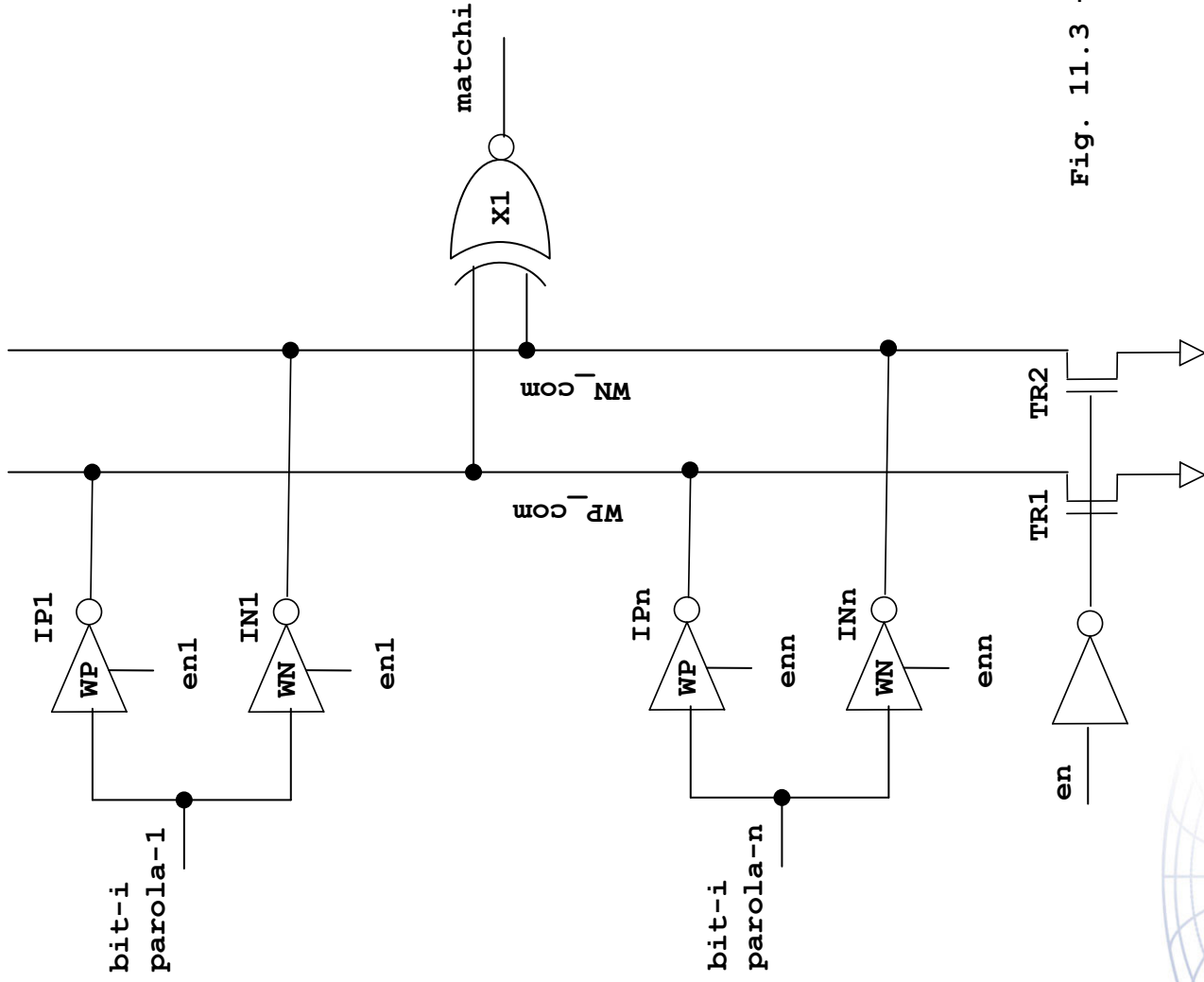
Si consideri ora la situazione in cui tutte le parole sono coinvolte nella compressione : en=1, en1=en2=...=enn=1 e l'i-mo bit di tutte le parole sia 0. In questo caso tutti gli inverters weak P pilotano la linea WP_com a 1 e tutti gli inverters weak N pilotano la linea WN_com a 1. Un circuito di confronto, composto da una XNOR (X1), ha i suoi due ingressi entrambi allo stesso valore e la sua uscita 'match' e' quindi 1, indicando che tutta le parole abilitate hanno lo stesso valore in posizione i-ma. Lo stesso valore match=1 si ottiene se il bit i-mo in tutte le parole abilitate ha valore 1 perche' anche in questo caso entrambe le linee WP_com e WN_com sono pilotate allo stesso valore.

Si supponga invece che, a causa di un difetto nell'i-mo bit di una parola, tutti i bit in posizione i-ma in tutte le parole hanno valore 1 ad eccezione del bit i-mo di una parola che invece ha valore 0. In questo caso la linea WP_com e' pilotata a zero da n-1 transistors strong N e pilotata a 1 da un singolo transistor weak P; la linea WN_com e' pilotata a 0 da n-1 transistors weak N ed e' pilotata a 1 da un singolo transistor strong P. Gli inputs del circuito di confronto sono differenti e quindi match=0, indicando che un fallimento e' presente in almeno una parola. Un simile ragionamento si applica nel caso in cui l'i-mo bit in tutte le parole ha valore 0 eccetto l'i-mo bit di (almeno) una parola che ha valore 1: anche in questo caso si ha match=0.

L'i-mo segnale 'matchi' e' usato come ingresso di un output buffer come riportato in Fig. 11.4. L'output buffer e' composto da due transistors TRP e TRN per pilotare l'i-ma componente di uscita DQi in funzione dell'ingresso biti. Tutti i DQ non sono pilotati se il segnale esterno comune OE_ e' 1. Il singolo DQi e' lasciato non pilotato nel modo di test di compressione di parola se matchi=0. Se matchi=1 allora DQi=biti.

L'organizzazione generale della compressione di parola e' riportata in Fig. 11.5. Ci sono tante strutture di questo tipo quanti bits ci sono in una parola.





Simbolo

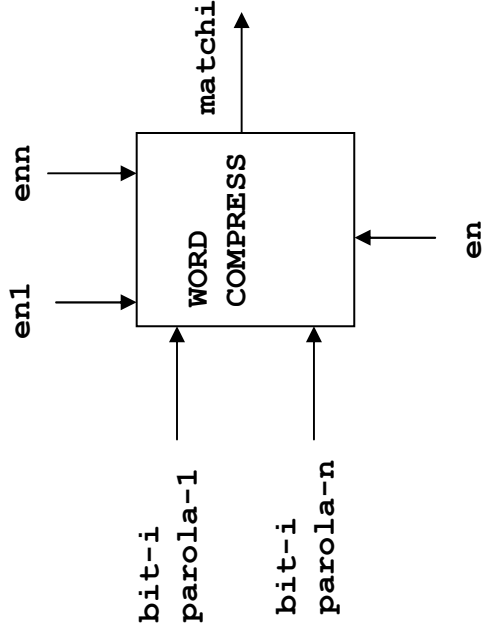
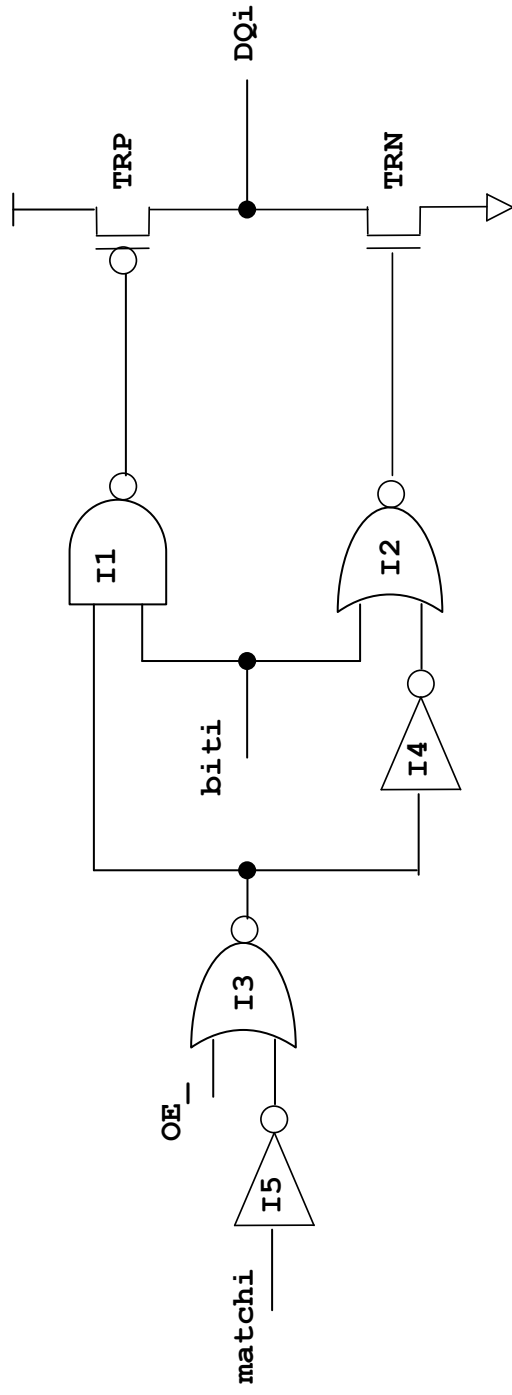


Fig. 11.3 - Compressione dell'i-mo bit in n parole.





Simbolo

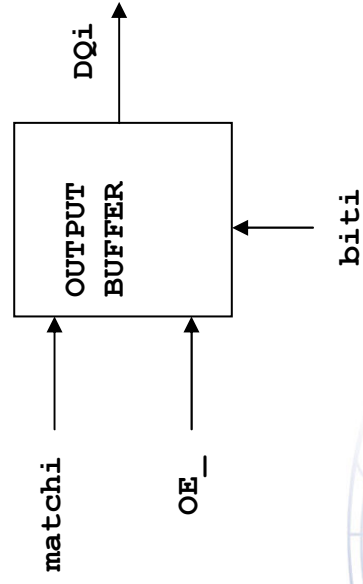


Fig. 11.4 - Output buffer con segnale di match per la compressione di parola.



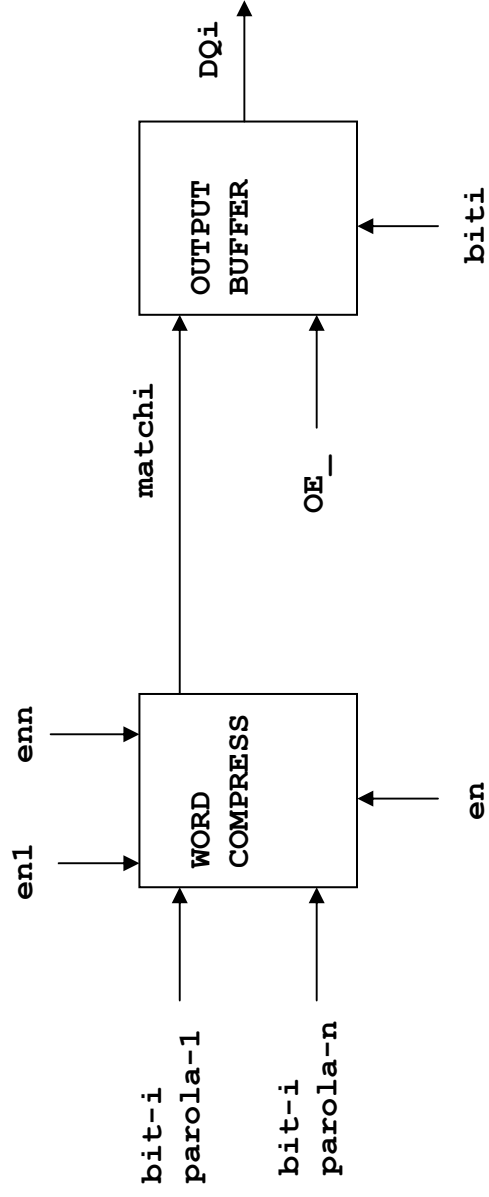


Fig. 11.5 - Organizzazione della compressione di n parole (i-mo bit)



11b. Verifica interna (IVR)

La tecnica di verifica interna si applica ad una memoria organizzata per pagine, come riportato in Fig. 11.6, in cui in una singola operazione si trasferisce il contenuto di tutti gli n bytes relativi ad una singola word line WL ad una corrispondente struttura di sens amplifiers SA0, SA1, ..., SAN ognuno dei quali composto di 8 strutture per la lettura degli 8 bits di un byte.

In Fig. 11.6 e' riportata una organizzazione dei sens amp per la lettura/scrittura in modalita' utente in cui un bus CSL (column select lines) di selezione dei sens amps e' in grado di selezionare uno specifico sens amp e di inviare i suoi 8 bits su una linea comune DL di 8 bits.

Il circuito che costituisce il sens amp del singolo bit puo' essere schematizzato come riportato in Fig. 11.7 per le operazioni di lettura/scrittura in modalita' utente. Prima di ogni operazione di lettura il nodo D del data latch viene inizializzato a 1 (init=1) e la bit line BL viene precaricata a 1 tramite il transistor T1.

In fase di lettura si ha $ro=0$, $re=0$, $wr=0$, pre_1 , $sgd=sgs=1$ e WL al valore intermedio tra una distribuzione cancellata e una programmata. Se la cella selezionata ha una VT bassa (cancellata), lungo la stringa scorre corrente e quindi dopo un certo tempo $BL=0$. Se invece la cella selezionata ha un valore VT alto (programmata), lungo la stringa non scorre corrente e quindi BL rimane al valore di precarica. Dopo un certo tempo si forza il segnale $re=1$ e quindi il nodo D rimane al valore 1 o si porta al valore 0 a seconda se la cella e' cancellata o programmata. Forzando poi il segnale $ro=1$ il dato D viene trasferito sulla data line Q selezionata dal segnale CSL.

In conclusione : cella cancellata --> $D=1$ --> $Q=1$. Cella programmata --> $D=0$ --> $Q=0$.



La struttura costituita dagli inverters di cache e' parte del percorso di scrittura ed originariamente contiene il dato che si vuole scrivere nella cella.

Se si vuole scrivere uno 0, si deve avere pre_1 e $Q=0$, $wr=1$. La BL si trova a zero ed attivando sgd,sgs,WL si effettua una programmazione. Se si vuole mantenere un 1 nella cella (cancellata), si deve avere pre_1 e $Q=1,wr=1$. La BL si trova a VCC e attivando sgd,sgs,WL non si effettua una programmazione ma una inibizione.

In conclusione : $Q=0$ --> cella programmata; $Q=1$ --> cella cancellata.

Il sens amplifier di Fig. 11.7 puo' esser completato da una struttura XOR per effettuare il confronto contemporaneo dei dati di cache e dei data latch di tutti i sense amplifiers, come riportato in Fig. 11.

La linea comune vfy viene precaricata a VCC tramite il transistor T1 prima di ogni operazione IVR. Quando l'operazione IVR viene abilitata, si ha pre_1 e $en=1$ e di conseguenza se $Q=D$ (contenuto array = riferimento) per tutti i bits di tutti i bytes, si ottiene $vfy=1$ e $pass=1$ altrimenti si ha $pass=0$.



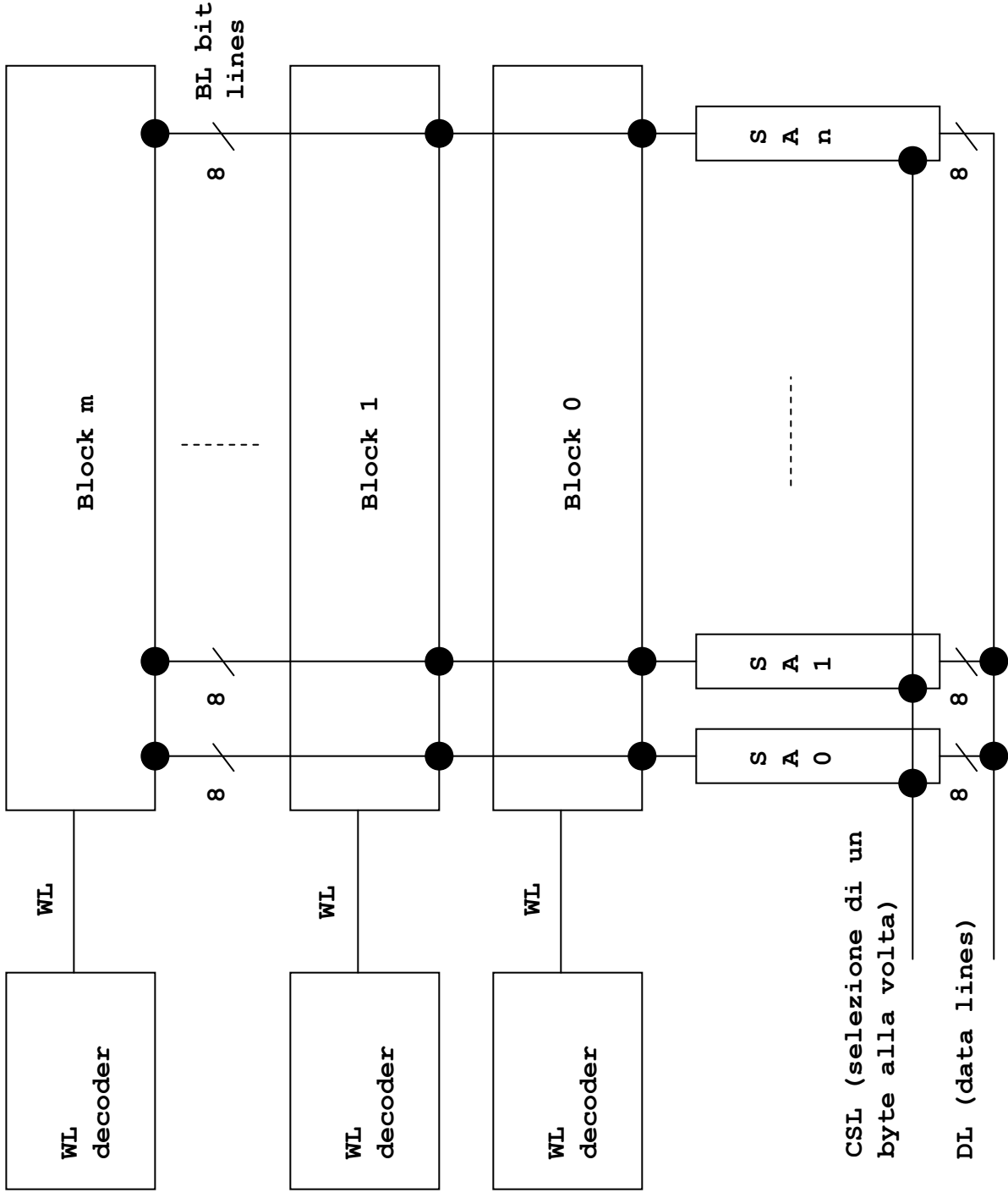


Fig. 11.6 - Organizzazione di blocco e pagina in una memoria NAND FLASH



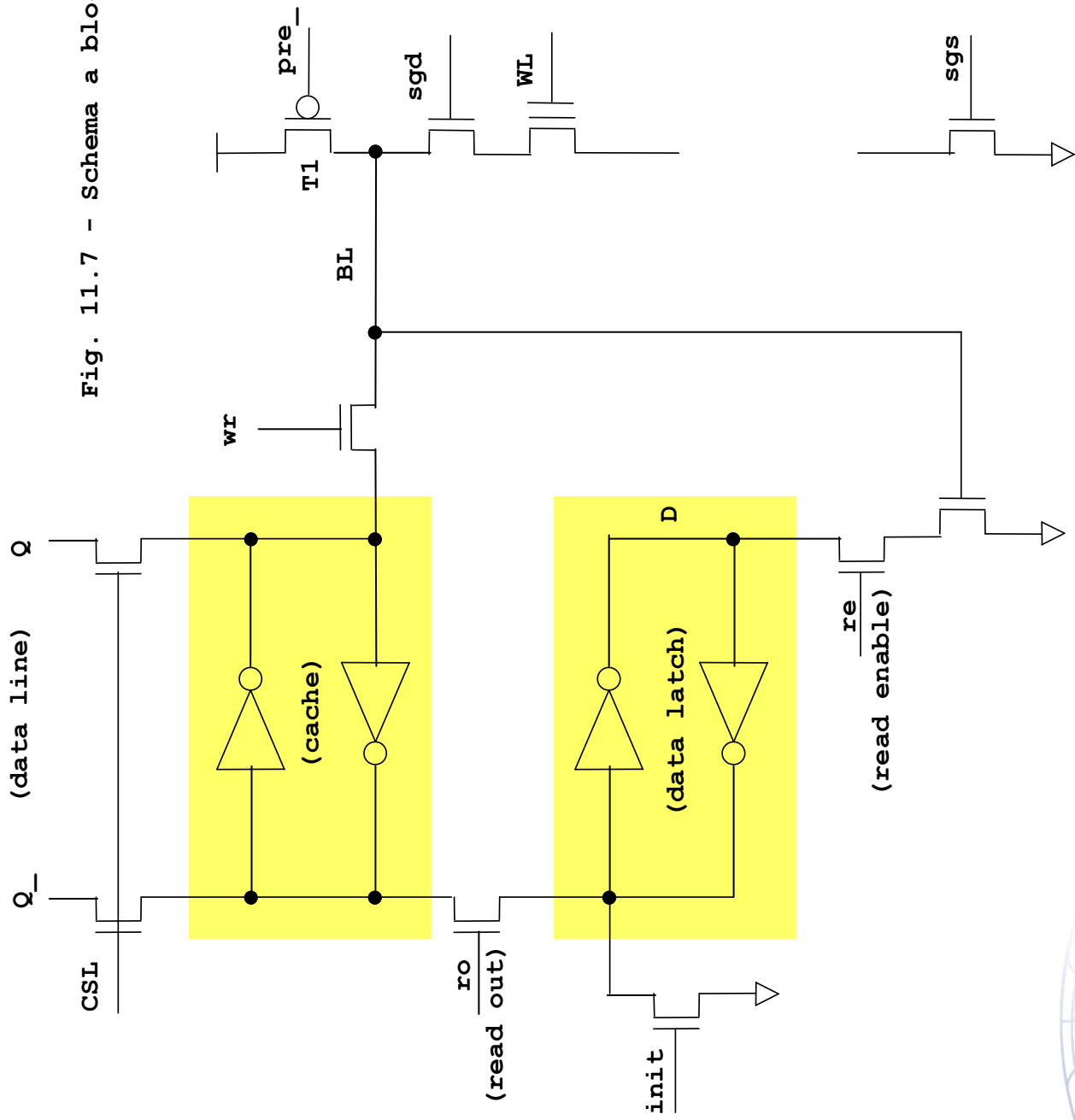
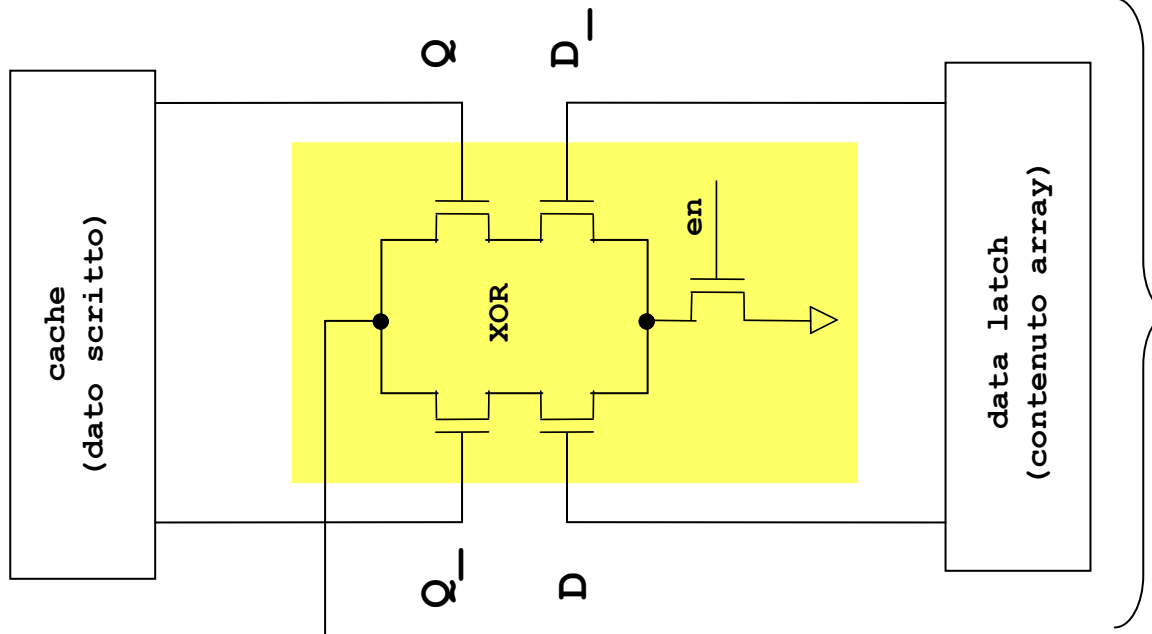
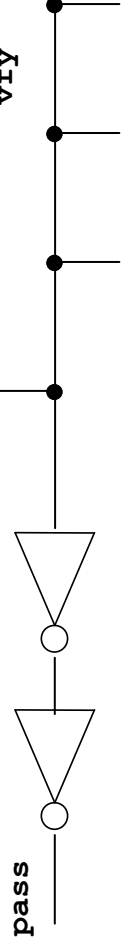


Fig. 11.7 - Schema a blocchi del sens amplifier.





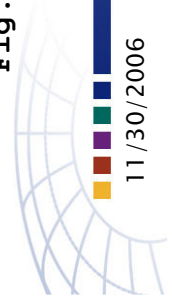
sensamp bit i parola j



Se $Q=D$ in tutti i bits di tutti i bytes (array = riferimento) allora vfy rimane precaricato a VCC e $pass=1$.

Se in almeno un bit si ha $D \neq Q$ (array \neq riferimento) allora vfy viene scaricato e $pass=0$.

Fig. 11.8 - Schema a blocchi della struttura di IVR



La linea comune vfy viene precaricata a VCC tramite il transistor T1 prima di ogni operazione IVR. Quando l'operazione IVR viene abilitata, si ha pre_1 e $en=1$ e di conseguenza se $Q=D$ per tutti i bits di tutti i bytes, si ottiene $vfy=1$ e $pass=1$ altrimenti si ha $pass=0$.

12) TECNICHE DI PARALLELIZZAZIONE

Con lo scopo di ridurre i tempi di esecuzione delle operazioni di test si sono progettate flussi di test su piu' memorie in contemporanea. Una volta richiesto il test su piu' memorie in contemporanea, il singolo test viene eseguito autonomamente da ogni memoria (SED = self error detect) senza la supervisione dettagliata da parte della macchina di test. Il risultato viene memorizzato internamente alla memoria per poter essere poi rapidamente letto dalla macchina di test ad operazione completata.

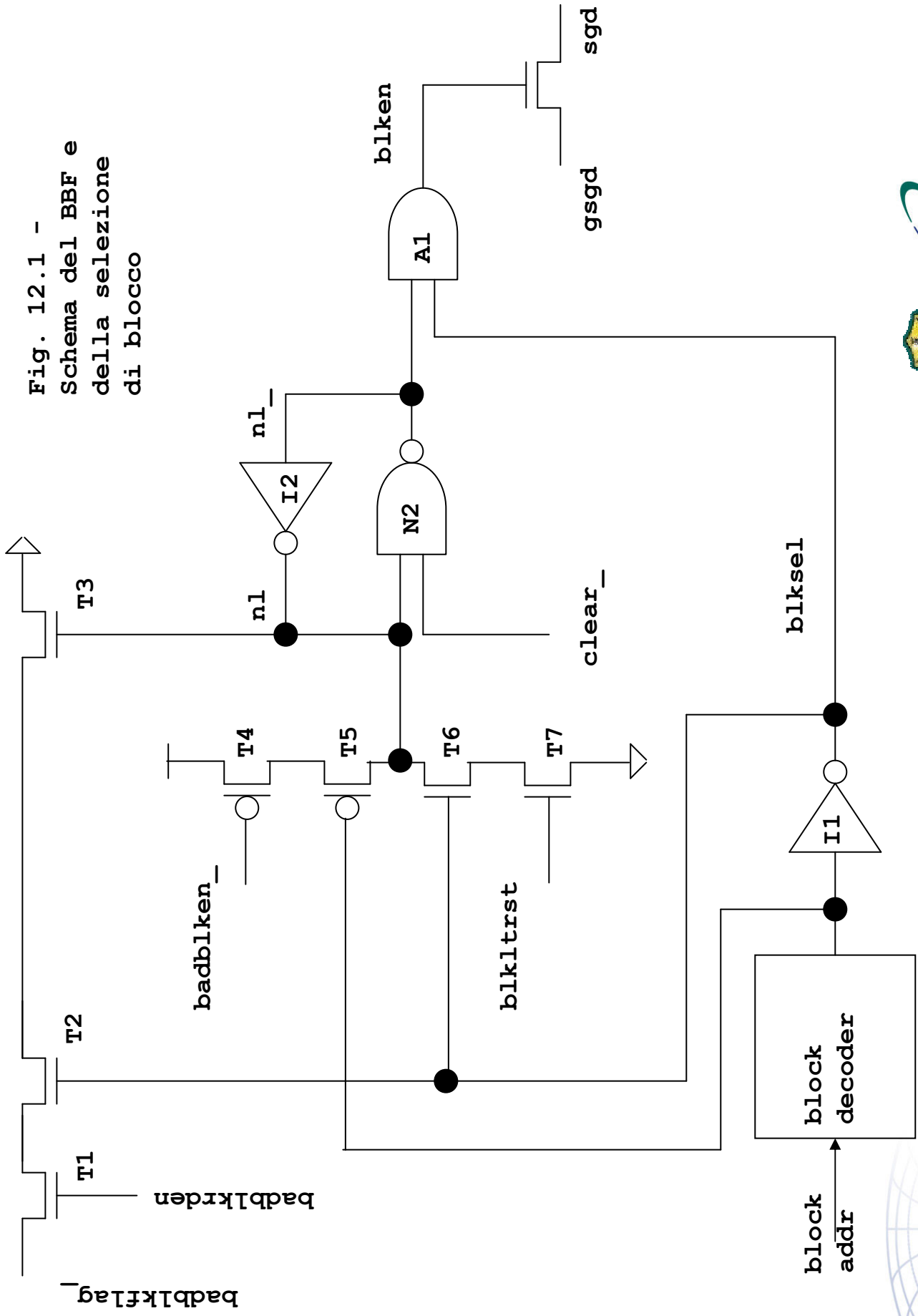
Di seguito verranno presi in considerazione due esempi di SED : SED con verifica interna (IVR) e SED con interruzione. Alla base di entrambe queste due tecniche di parallelizzazione ci sono alcuni concetti che le rendono possibili :

- ogni blocco della memoria e' munito di un latch (BBF = bad block flag) che viene posto a 1 dal controllore interno alla memoria qualora venga rilevato un errore sul blocco (SED IVR oppure SED interrotto).
- il firmware del controllore e' munito di una routine che seleziona i vari blocchi della memoria, esegue su ognuno di essi il test SED IVR, e forza 1 il BBF nel caso di fallimento
- la memoria e' munita di un circuito (R/B_ = ready/busy) che forza un segnale esterno a zero se il test e' in corso e lo forza a 1 solo quando il test e' terminato con la scansione interna di tutti i blocchi (SED IVR).
- la memoria deve essere munita di circuiterie apposite affinche' la macchina di test possa leggere il contenuto di ognuno dei BBF su un pad esterno (SED IVR oppure interrotto).

Di seguito vengono forniti dettagli relativi alla organizzazione, scrittura, lettura dei BBF.

Il bad block flag e' un latch associato ad ogni blocco della memoria ed e' usato per marcare un blocco come guasto ed inibire la selezione del blocco per una qualsiasi operazione di lettura, scrittura, cancellazione. Lo schematico di un BBF e' riportato in Fig. 12.1.

Fig. 12.1.1 -
 Schema del BBF e
 della selezione
 di blocco



Il BBF e' composto dalla struttura di latch I2,N2. Se il latch e' stato forzato a 1 (set), il segnale nl_ e' 0 e il blocco e' marcato come guasto; blken non puo' essere 1 in alcuna configurazione di indirizzi e il segnale sgd di selezione di stringa non puo' mai essere asserito. Se invece il latch e' stato forzato a 0 (reset), il segnale nl_ e' 1 e il blocco non e' marcato come guasto; blken puo' essere posto a 1 a seconda dello specifico indirizzo di blocco. Sul latch si possono effettuare le seguenti operazioni :

|<----- inputs ----->|

badblken_	blktrst	blkaddr	clear_	operazione	nl_
1	1	x	0	clear	1
1	1	addr	1	reset	1
0	0	addr	1	set	0

Lo stato del latch puo' essere letto attraverso il segnale badblkflag_ in queste condizioni : badblkrden = 1, badblken_ = 1, blktrst = 0, blkaddr = indirizzo di blocco, clear_ = 1, in questo caso la linea badblkflag_ e' forzata a gnd se nl_ = 0 altrimenti non e' pilotata e rimane al valore di precarica.

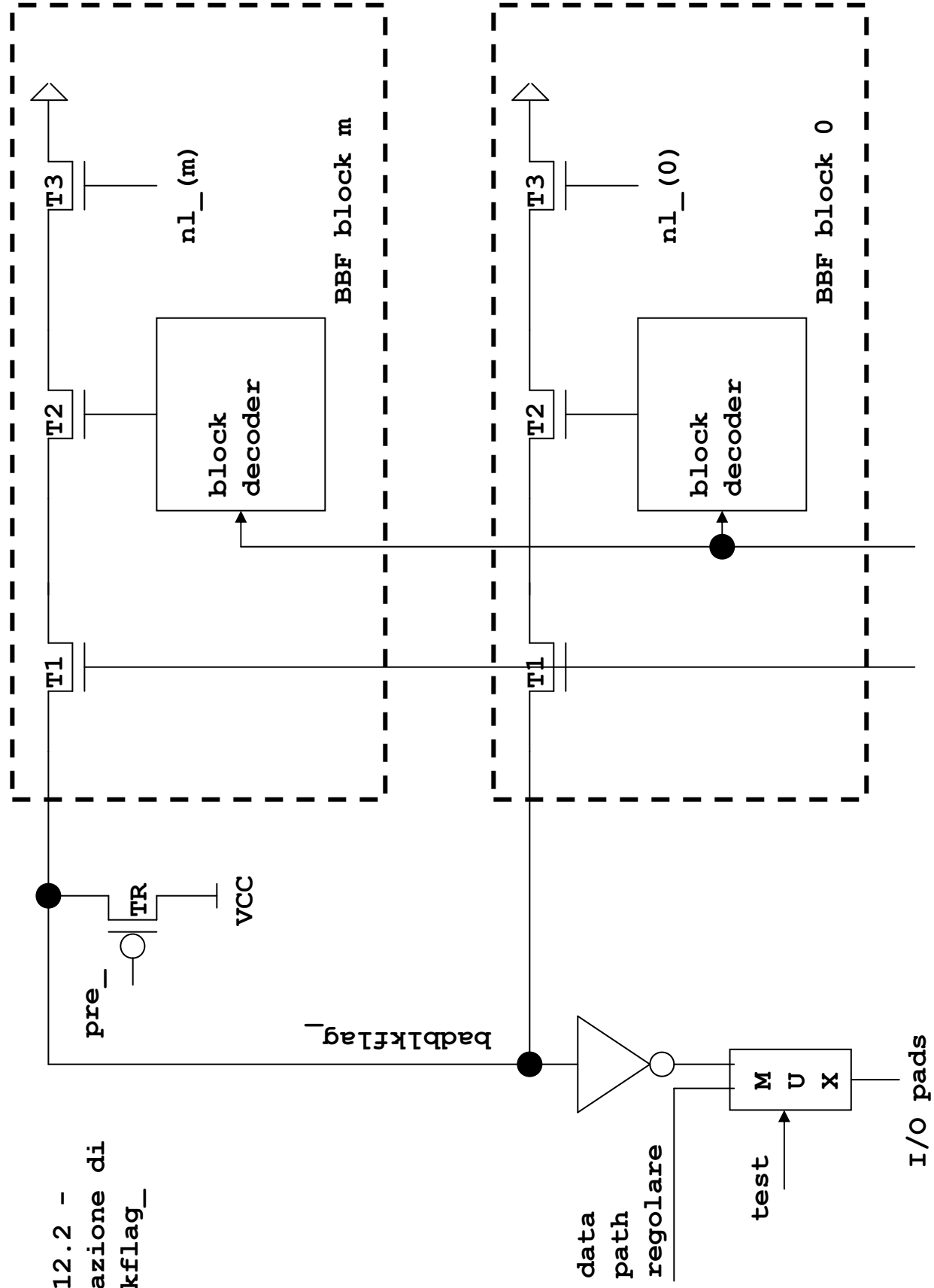
La rilevazione del BBF e' organizzata come riportato in Fig. 12.2.

I segnali badblkflag_, provenienti dal BBF di ogni blocco, sono connessi tutti insieme per generare una linea unica badblkflag_. La condizione di partenza, prima di una operazione di lettura di BBF, e' badblkflag_ = 1 ottenuta dalla precarica attraverso il transistor TR (pre_ = 0). Quando si vuole leggere lo stato del BBF di uno specifico blocco, si puo' procedere nel seguente modo :

- pre_ = 1 per disabilitare la precarica della linea badblkflag_ (il transistor TR e' OFF)
- badblkrden = 1 per connettere tutti i BBF alla linea badblkflag_ (tutti i transistors T1 sono ON)

Solamente il blocco indirizzato avra' il suo transistor T2 On, il transistor T2 degli altri blocchi saranno OFF. Se il blocco indirizzato e' marcato come guasto, il suo segnale nl_ e' 1 ed il corrispondente transistor TR3 e' ON : la linea badblkflag_ e' scaricata a gnd attraverso la catena di transistors T1,T2,T3 del blocco indirizzato.

Fig. 12.2 -
generazione di
badblkflag_



Se il blocco selezionato non e' marcato come guasto, il segnale nl_ e' 0 ed il transistor T3 e' OFF : il segnale badblkflag_ non e' scaricato a gnd ed il suo valore logico rimane 1. C'e' la possibilita' di selezionare i blocchi in gruppi oppure tutti insieme; in questo modo si puo' rilevare con una singola operazione se c'e' almeno un blocco marcato come guasto in un gruppo di blocchi o nell' intera memoria. Il segnale badblkflag_ si puo' connettere ad un pad esterno in modalita' di test per fornire informazioni relative al fatto che un blocco e' marcato come guasto oppure no.

12a. SED con verifica interna

Il flusso completo di una operazione SED IVR e' riportato in Fig. 12.3 ed ha lo scopo di confrontare il contenuto di una o piu' pagine della memoria con una stessa pattern di riferimento. Una prima parte del flusso e' demandata ad una macchina di test per : entrare in modalita' di test, abilitare il modo di test IVR, inizializzare tutti i BBF e riempire la data cache con una pattern di riferimento in tutti i chips in parallelo. La macchina di test puo' effettuare il monitor del latch di stato di ogni memoria individualmente e se e' ready, la macchina di test puo' effettuare una verifica sui BBF di quella memoria per rilevare se c'e' stata una condizione di blocco guasto.

12b. SED con interruzione

La tecnica di SED interrotto e' una tecnica di test parallelo che serve a verificare se chip di memoria sono in grado di effettuare una operazione in modalita' utente (lettura/cancellazione/programmazione) in un certo tempo. L'operazione in modalita' utente viene attivata su un certo blocco di piu' memorie in contemporanea e dopo un certo tempo viene applicato a tutti i chip di memoria un comando di check del completamento (fase di interruzione). Ogni chip di memoria ha una routine di test di SED interrotto tale che se l'operazione richiesta e' ancora in corso al momento del comando di interruzione oppure se l'operazione e' terminata con insuccesso per time-out allora BBF viene portato a 1 (fallimento) altrimenti viene lasciato a 0 (successo). L'intero flusso di un SED interrotto puo' essere riassunto come in Fig. 12.4.



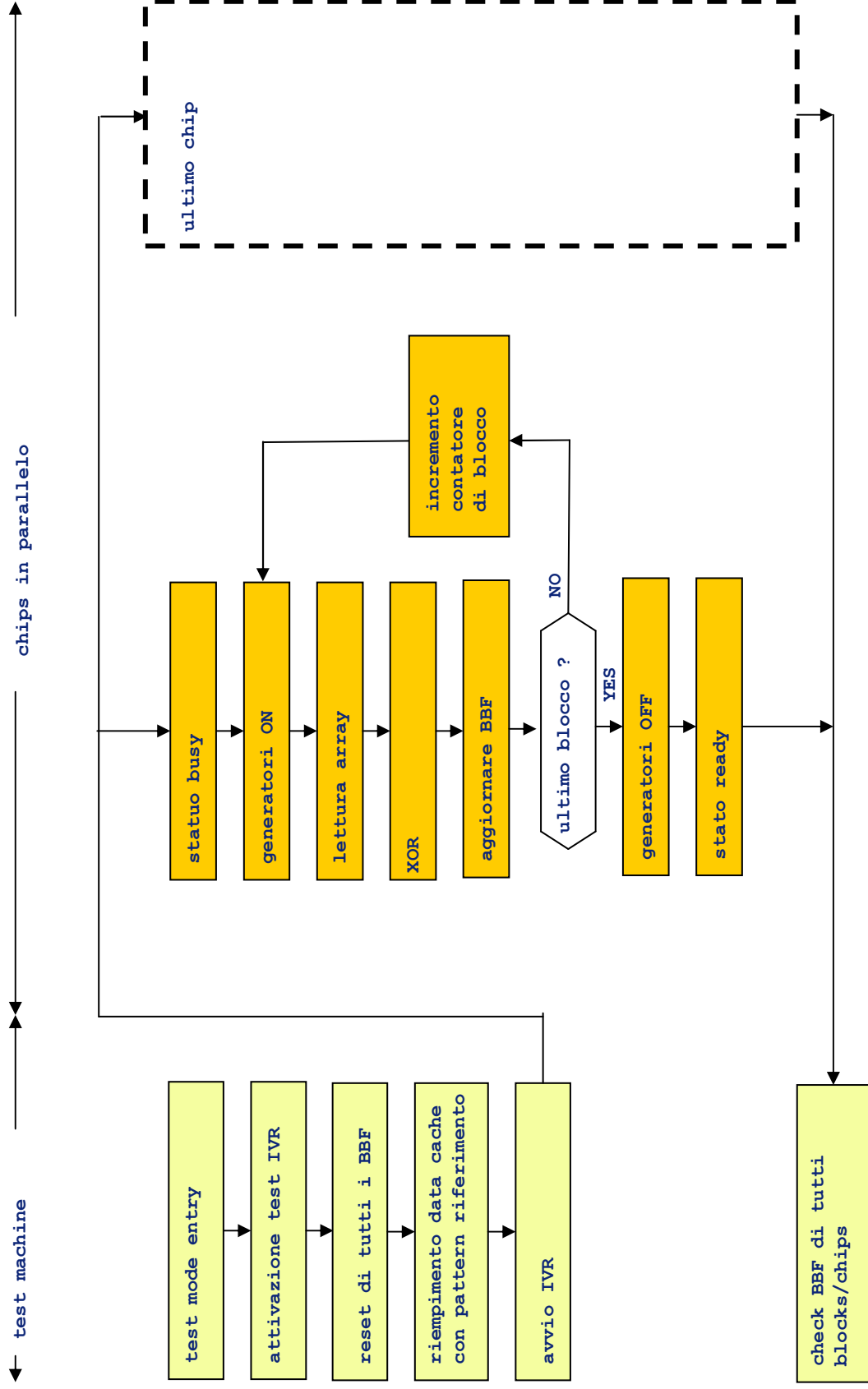


Fig. 12.3 - Flusso operativo di SED IVR



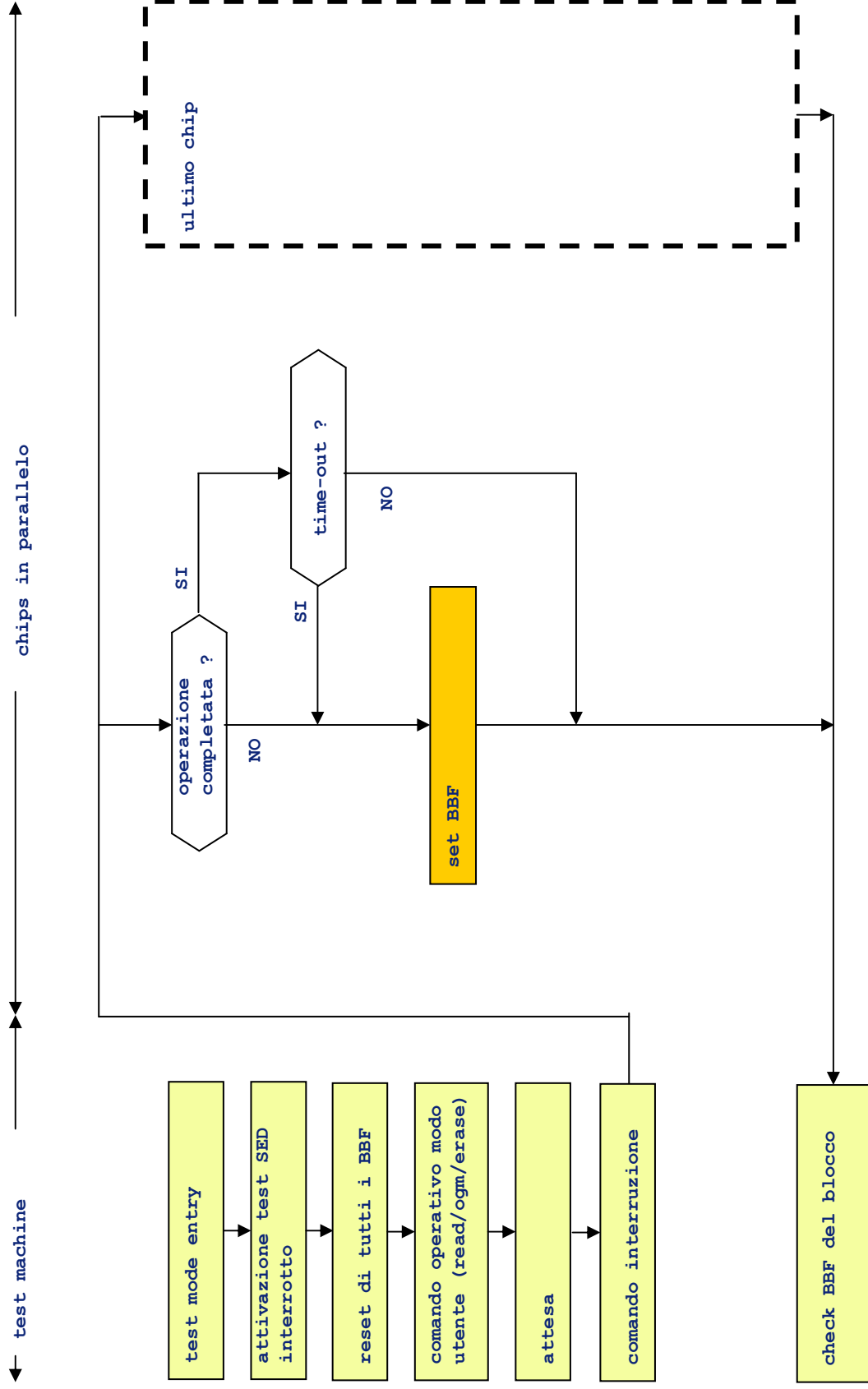


Fig. 12.4 - Flusso operativo di SED interrotto



13) MONITOR DI ALGORITMI

Le operazioni di cancellazione e scrittura in memorie FLASH sono realizzate usando algoritmi complessi composti di molti passi differenti e che richiedono azioni di controllo su temporizzazioni, conteggi, e valori analogici. Una operazione di scrittura e' tipicamente un loop di impulsi di programmazione e di verifica fino a che il valore voluto non e' memorizzato con il giusto margine. La cancellazione di un blocco e' tipicamente una sequenza di passi : pre-programmazione di tutte le parole di una pagina, verifica della pre-programmazione, impulso di cancellazione. verifica della cancellazione.

Lo schema a blocchi di un controllore di algoritmi basato su firmware scritto in ROM e' riportato in Fig. 13.1. Un micro-controllore e' progettato per gestire l'esecuzione dei differenti passi negli algoritmi di programmazione e cancellazione. Una volta che una operazione di cancellazione/scrittura e' stata richiesta, e l'algoritmo e' avviato, non e' possibile a priori dire quale particolare passo il micro-controllore sta eseguendo ad un certo istante. Infatti gli algoritmi di cancellazione e scrittura sono adattativi ed un passo e' attivato solo quando le condizioni associate al passo precedente sono soddisfatte. Un modo per effettuare il monitor molto elementare dell'attivita' di un controller di algoritmi e' il concetto di registro di stato che e' stato ampiamente usato nelle memorie FLASH ed e' disponibile all'utente finale per una corretta interfaccia tra la memoria FLASH ed il resto del board dove la memoria e' inserita.

Oltre alla tecnica del registro di stato, l'attivita' durante operazioni di cancellazione o scrittura possono essere accuratamente monitorate a scopo di debug usando una specifica modalita' di test. Un primo modo di test chiamato 'forzamento esterno del clock' (tck) permette di sostituire il clock interno del micro-controller con un clock forzato da un pad esterno : in questo modo si puo' imporre uno specifico periodo di clock e si puo' forzare una sincronizzazione. Un secondo modo di test chiamato 'monitor dell'indirizzo di ROM' (tra) permette di seguire l'evoluzione dell'algoritmo tramite la lettura, sui pads di uscita, dell'indirizzo delle istruzioni di ROM. Un terzo modo di test chiamato 'monitor della write state machine' (twsm) permette di effettuare il monitor, sui pads esterni, dei segnali reg_add e reg_data rappresentanti la selezione e il contenuto dei registri che generano i segnali di attuazione per il core, i blocchi analogici e tutti gli altri circuiti della memoria.

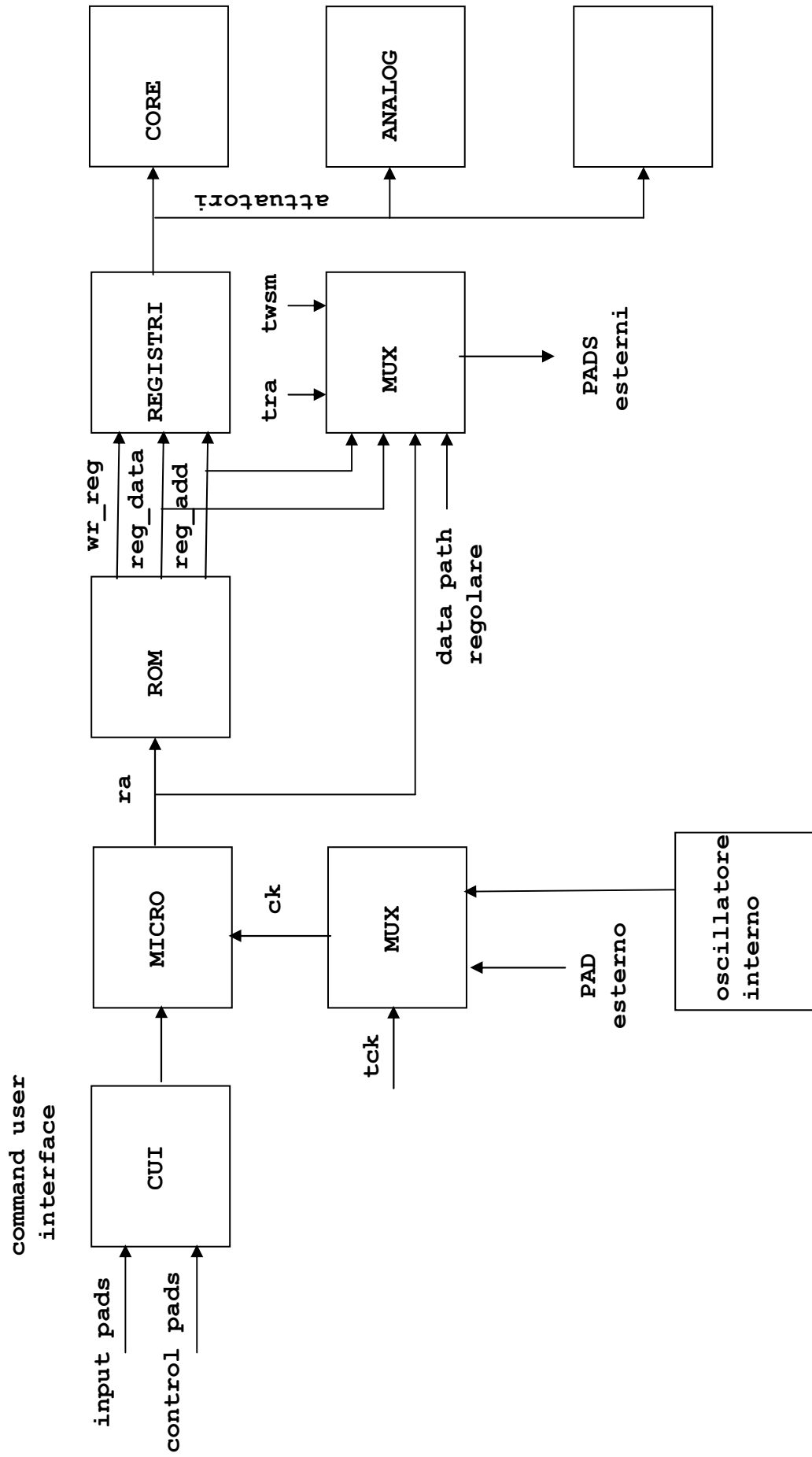


Fig. 13.1 - Schema a blocchi relativo al monitor di algoritmi e forzamento di clock

