Master II livello 2006

NASO GIOVANNI

Progettazione della testabilita' (DFT) di memorie FLASH

INDICE

in partnership with

Micron®

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

Scopi di DFT sono :

1a. Aumentare la resa con l'uso di ridondanza

1b. Aumentare l'affidabilita' con l'uso di stress elettrici

1c. Aumentare la manufatturabilita' con l'uso di calibrazioni

1d. Aumentare la flessibilita' con l'uso della configurazione

1e. Diminuire il ramp-up con tecniche di debug

2) CARATTERISTICHE

Le caratteristiche che DFT deve avere per assolvere i suoi scopi sono:

2a. Osservabilita' di tensioni, frequenze, riferimenti, sensori, distribuzioni, evoluzione di algoritmi.

L'osservabilita' e' un concetto diverso da internal probing.

2b. Controllabilita' di tensioni, frequenze, polarizzazione di celle, algoritmi, durate, configurazioni.

2c. Ridotto test time attraverso tecniche di compressione, parallelizzazione, self test.

3) ORGANIZZAZIONE

L'organizzazione di DFT si basa su una struttura fondamentale costituita da :
3a. celle di memoria FLASH (fusibili) che si usano per configurare, trimmare, riparare il chip (Fig. 3.1-3.2)

L'ingresso in test mode non e' fornito al cliente ma e' abilitato solo in factory con
3b. opportune tecniche di entry (low voltage, high voltage).

Due diverse filosofie di gestione di DFT sono prese in considerazione :
3c. organizzazione modale (con l'uso di decoder di test modes) (Fig. 3.3-3.4)
3d. organizzazione a registri (Fig. 3.5-3.6-3.7-3.8)

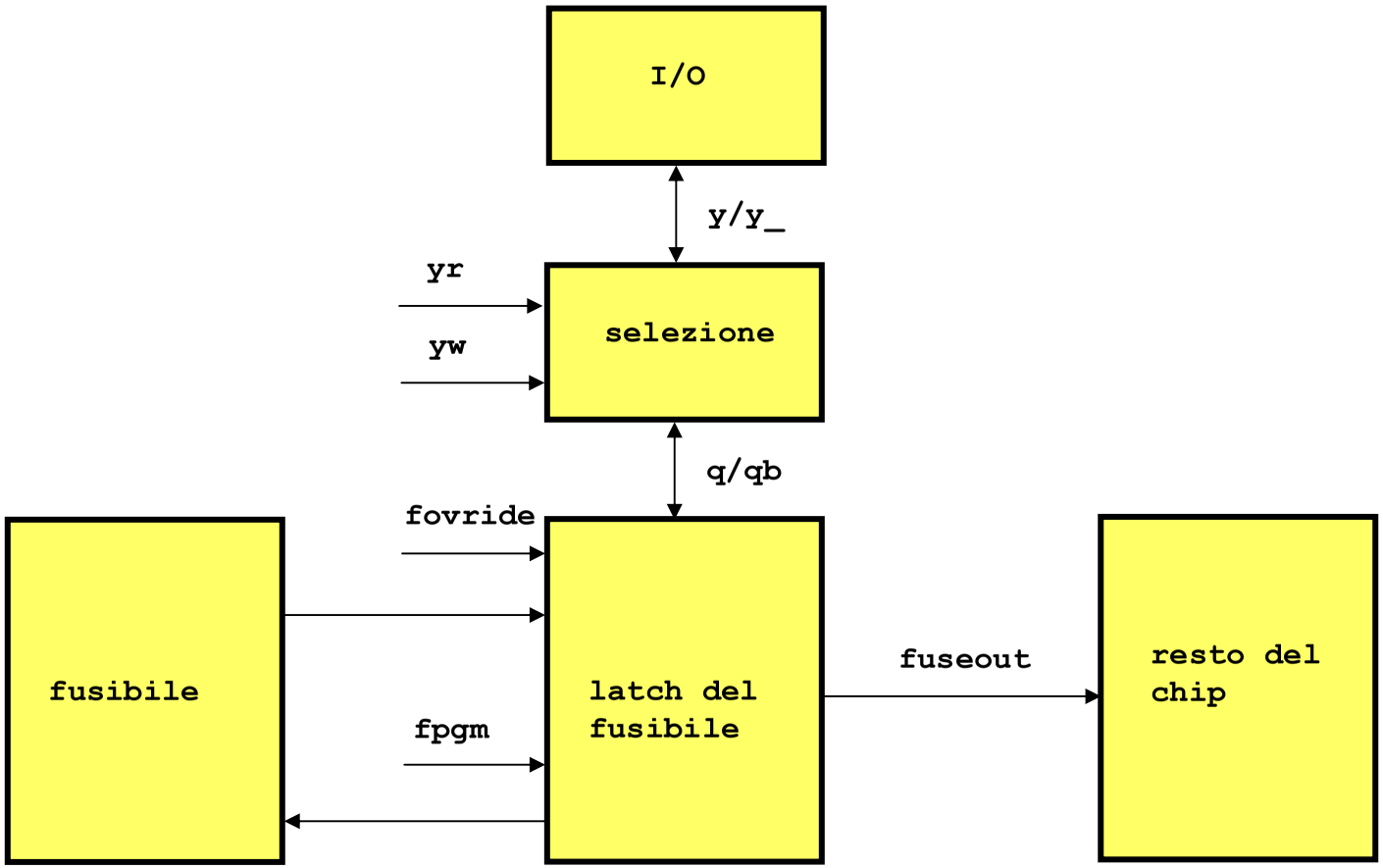


Fig. 3.1 - schema a blocchi del fusibile

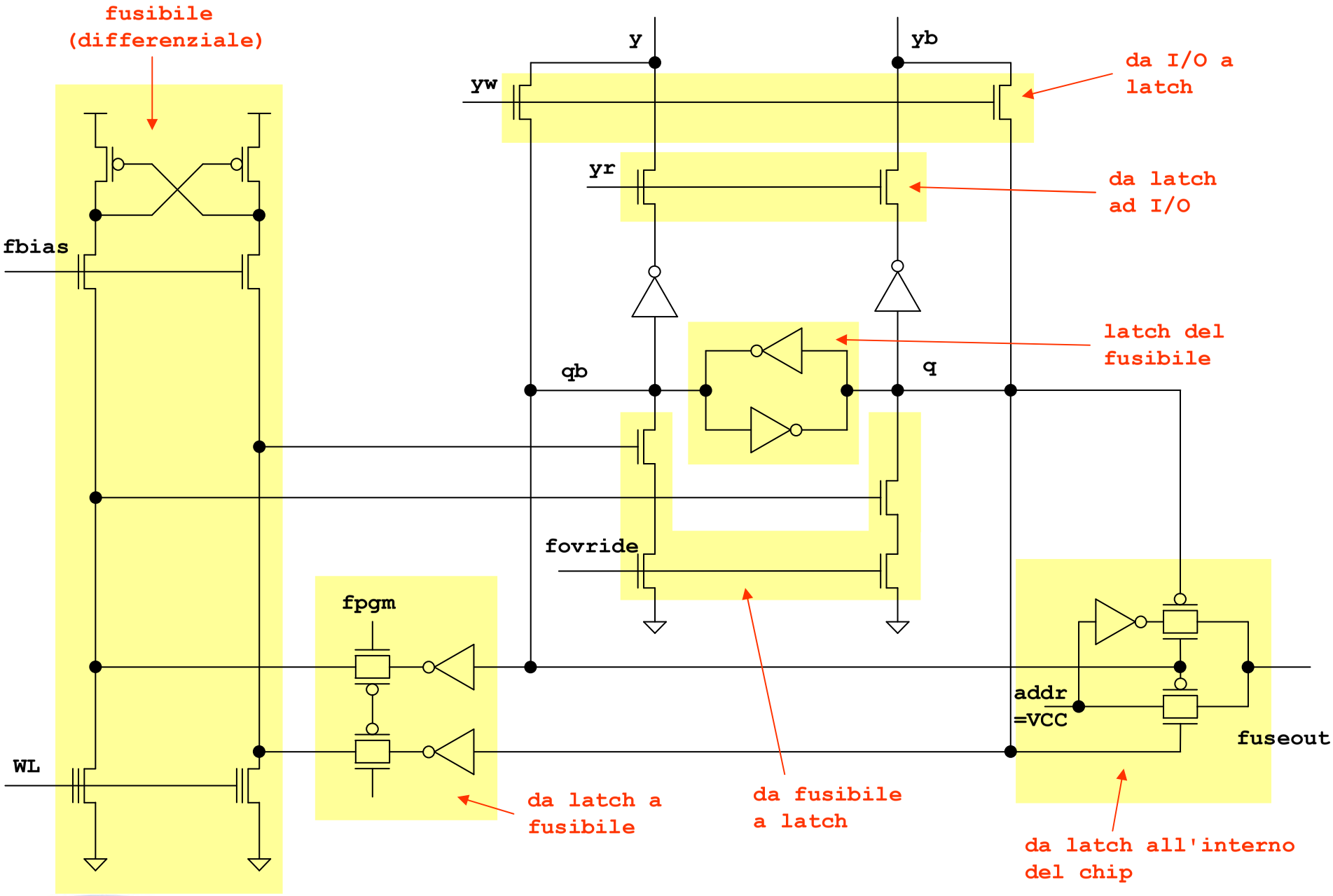


Fig. 3.2 - struttura dettagliata del fusibile

Fig. 3.3 - Organizzazione DFT modale : architettura

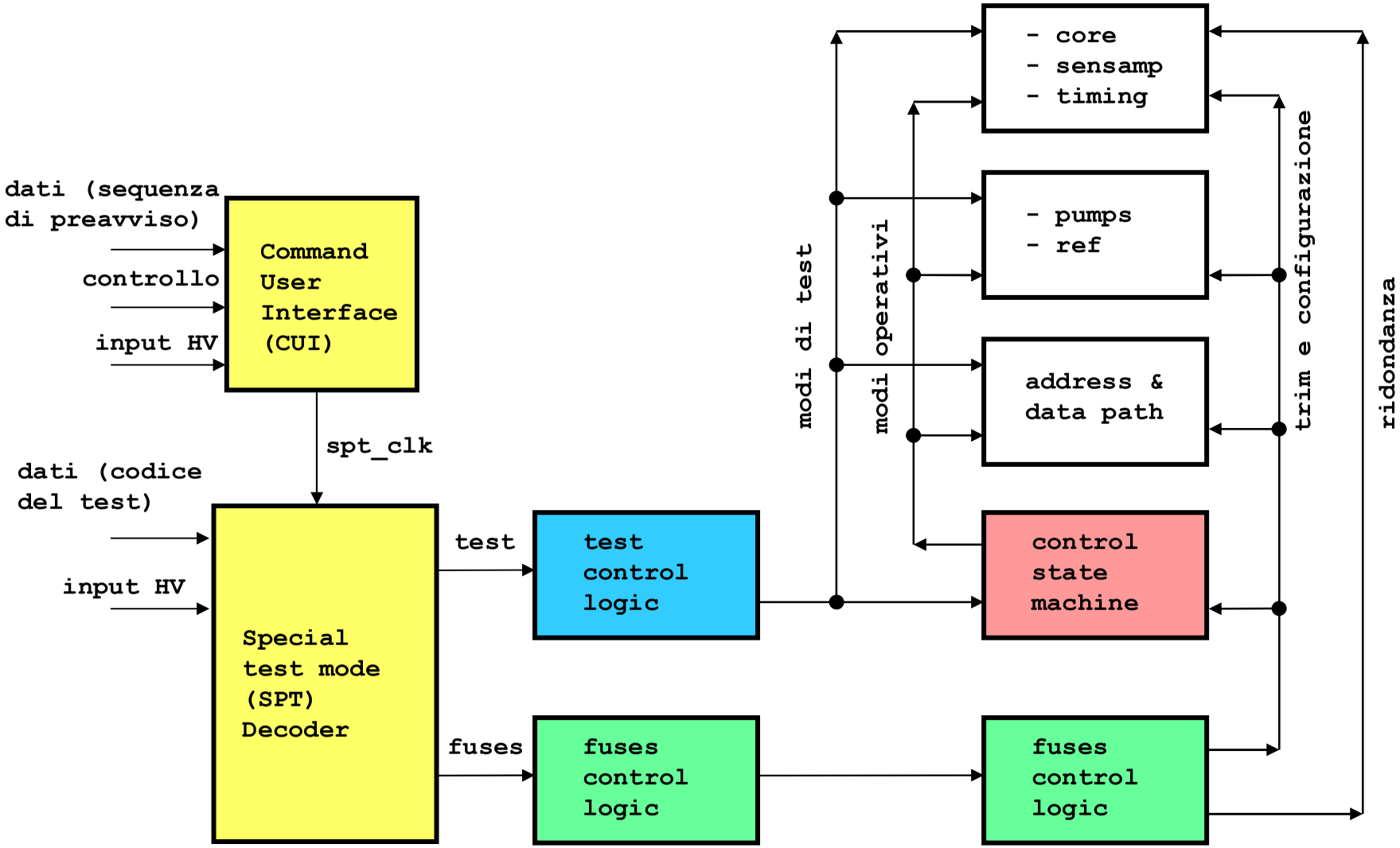
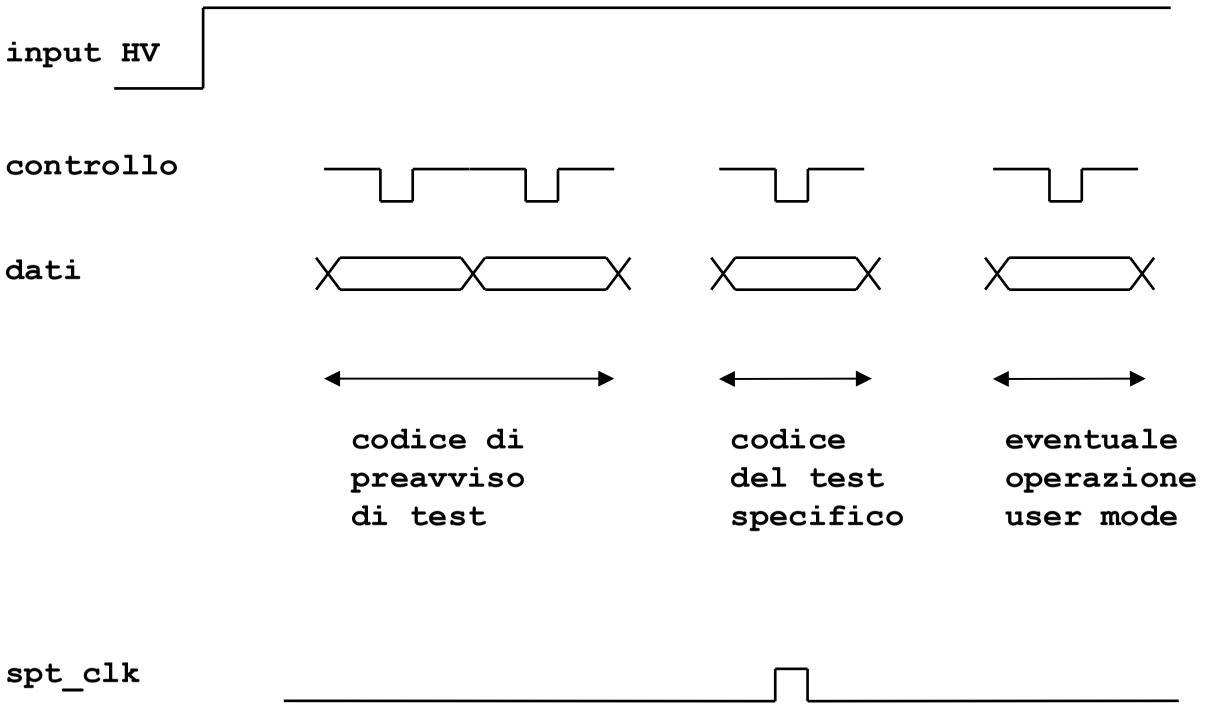


Fig. 3.4 - Organizzazione DFT modale :
forme d'onda per l'attivazione di test



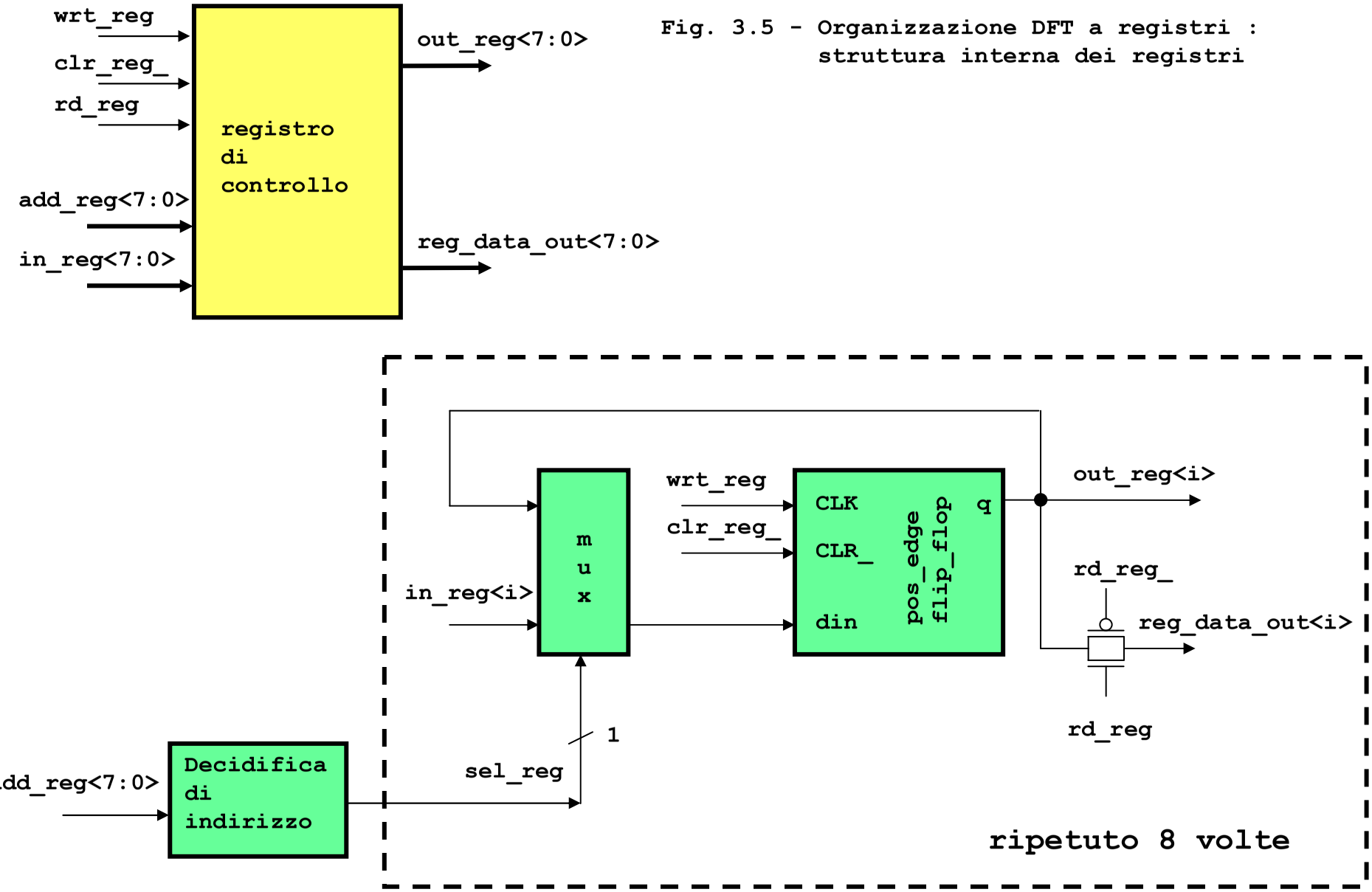


Fig. 3.5 - Organizzazione DFT a registri : struttura interna dei registri

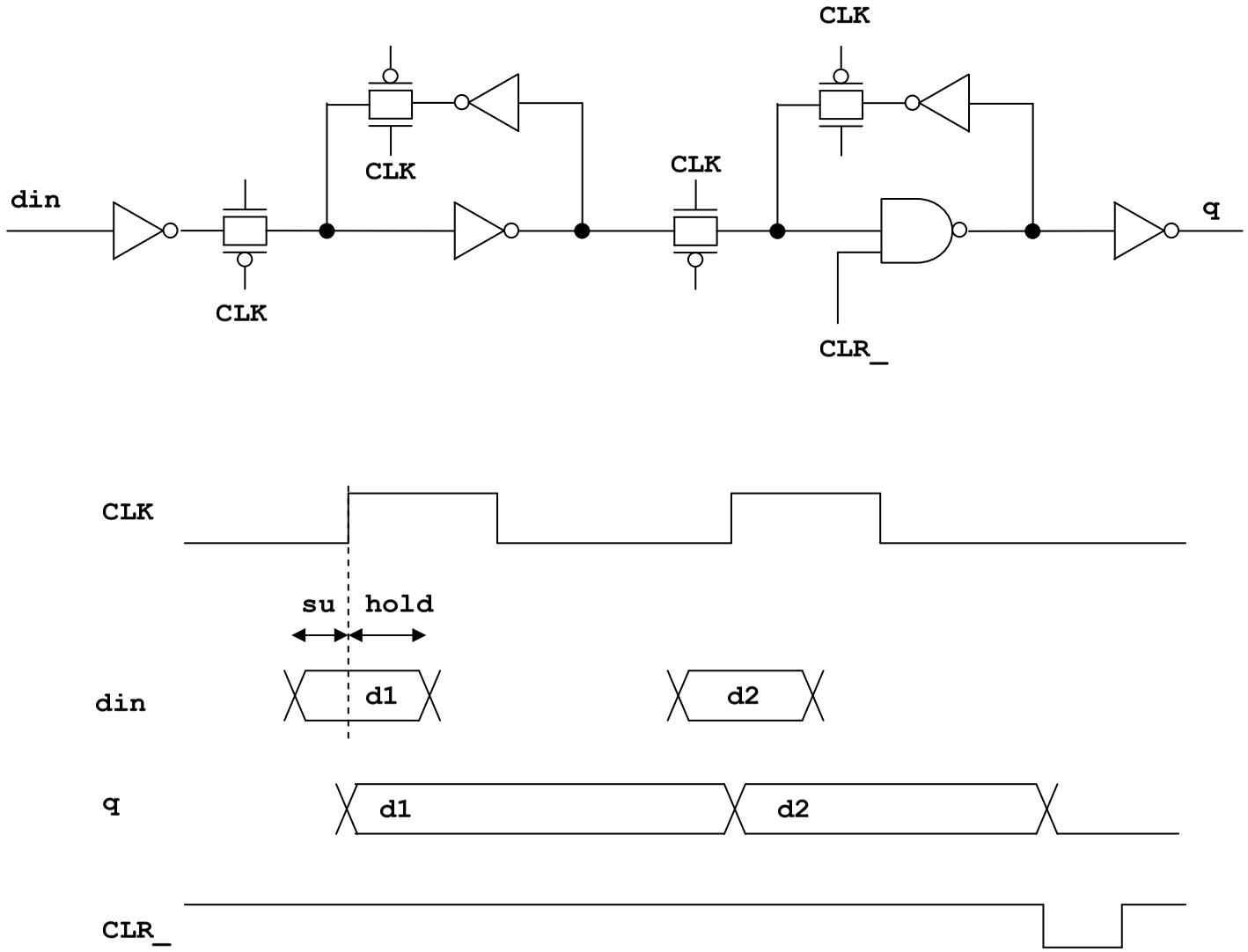


Fig. 3.6 - positive edge triggered flip-flop

Fig. 3.7 - Organizzazione DFT a registri : architettura

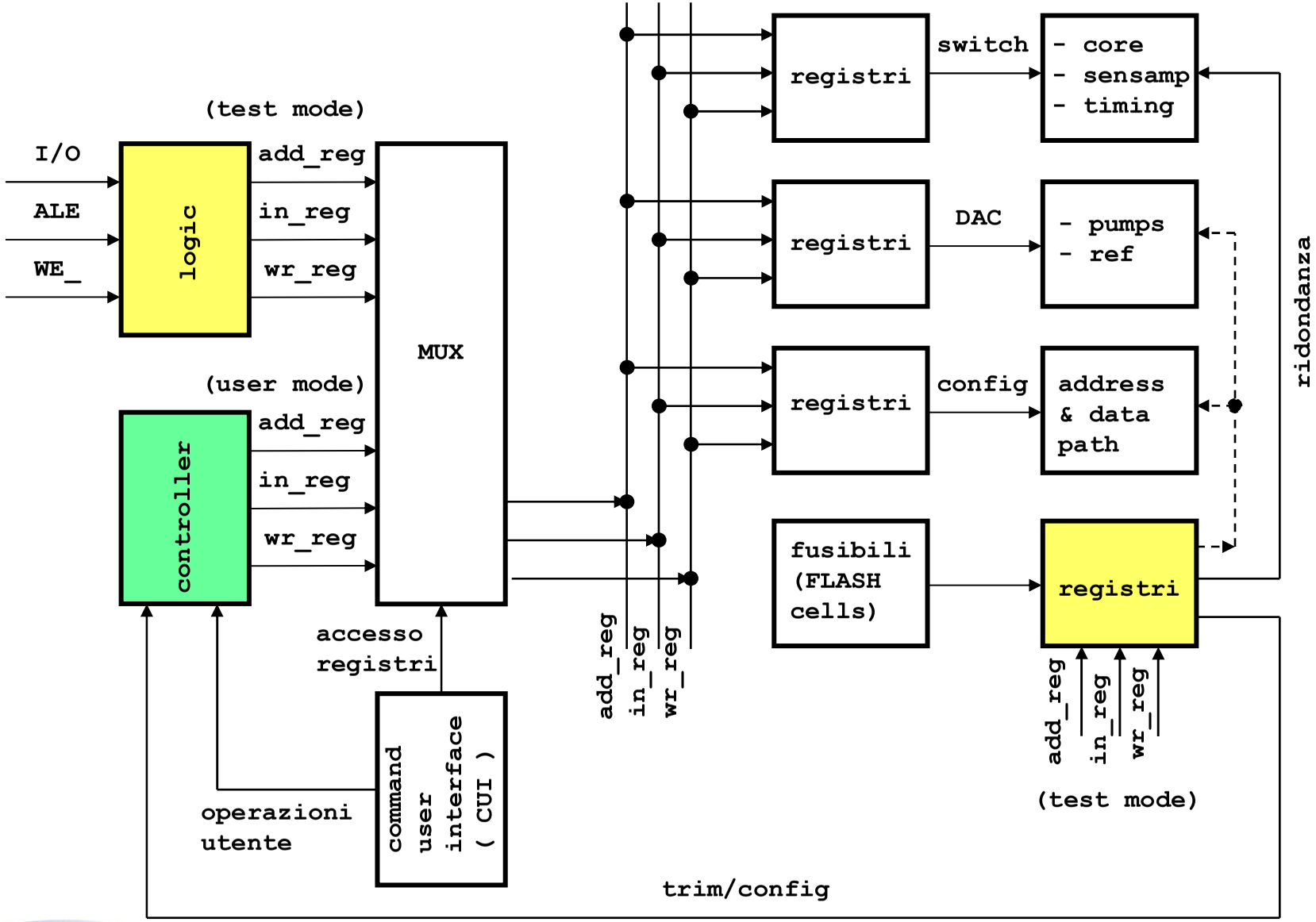
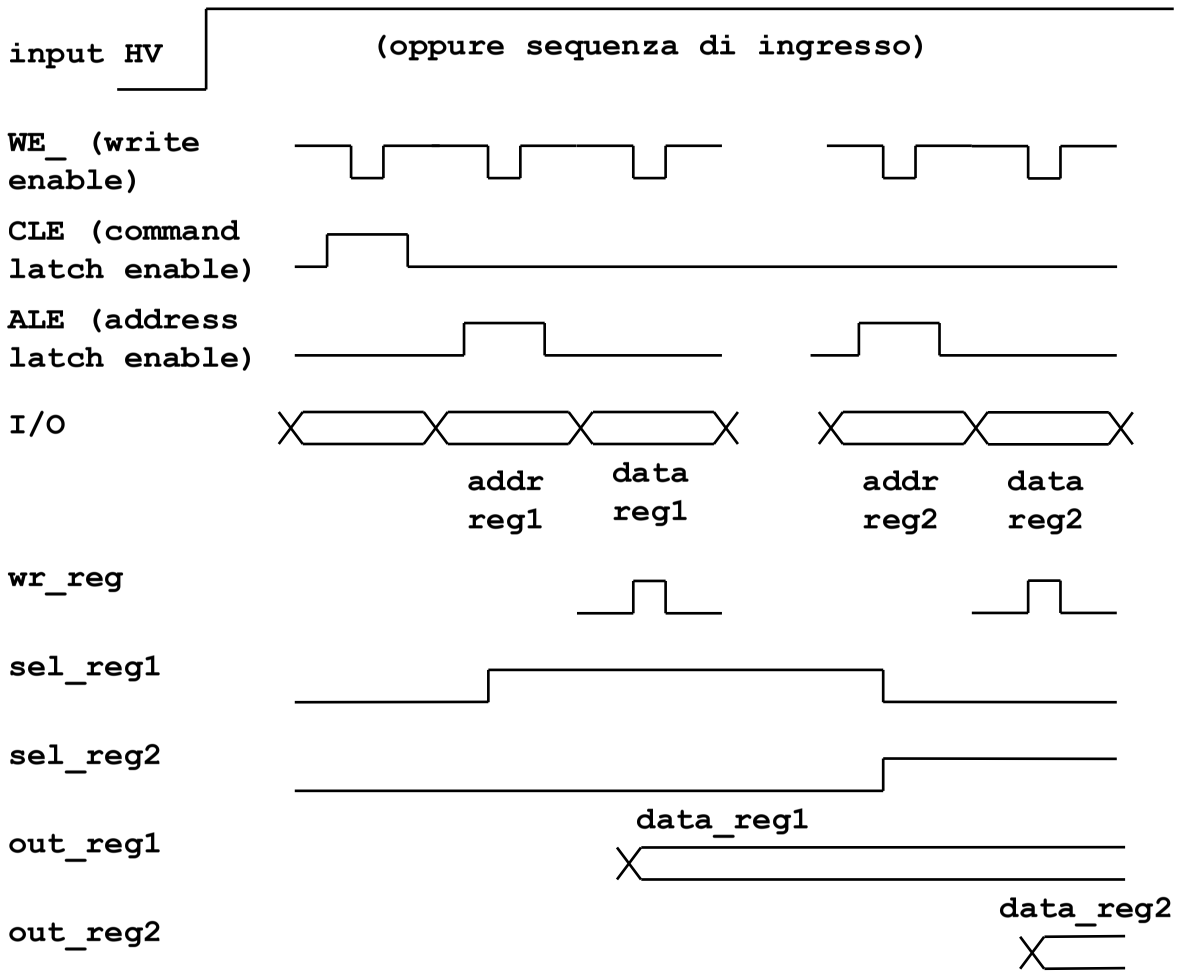


Fig. 3.8 - Organizzazione DFT a registri :
forme d'onda per la scrittura nei registri di controllo



4) RIDONDANZA

La ridondanza permette di riparare strutture di memoria guaste con strutture funzionanti.

La resa di produzione viene notevolmente migliorata con l'uso della ridondanza soprattutto in processi non maturi e ad elevata integrazione (Fig. 4.1).

I circuiti alla base della ridondanza sono :
3a. circuiti di match tra una configurazione di fusibili e una particolare configurazione di indirizzi (Fig. 4.2-4.3).

Si possono considerare tre tipi di ridondanza :
4b. riga (Fig. 4.4)
4c. colonna (Fig. 4.5-4.6)
4d. blocco (Fig. 4.7)

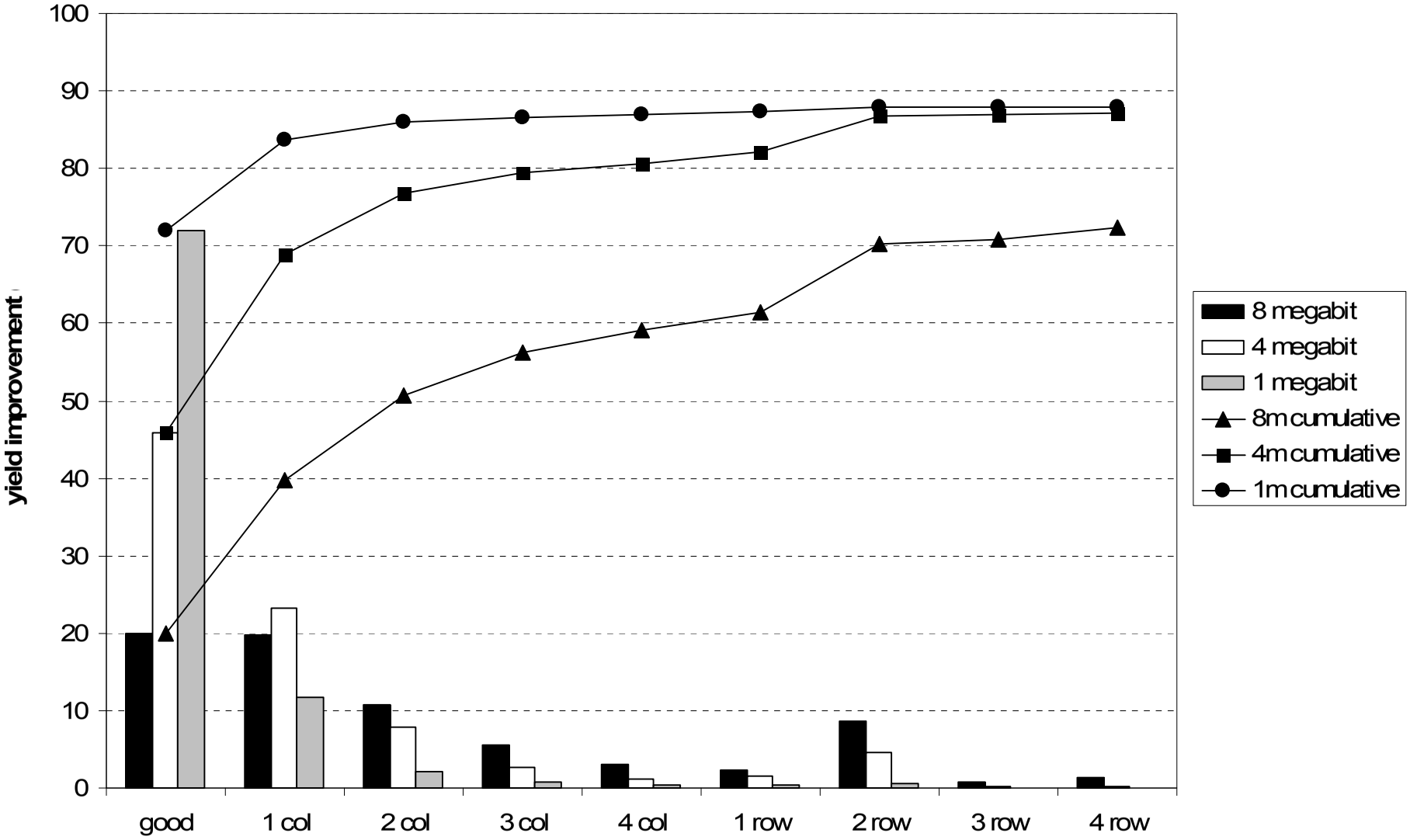
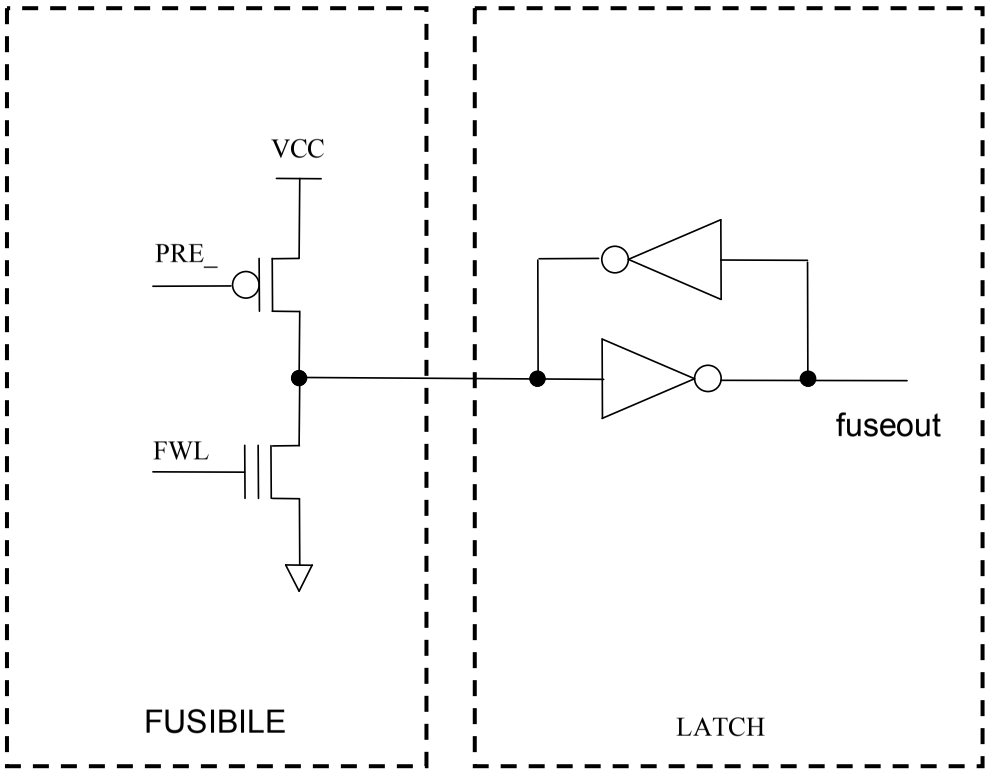


Fig. 4.1 - Effetto della ridondanza sulla resa di produzione



. **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.

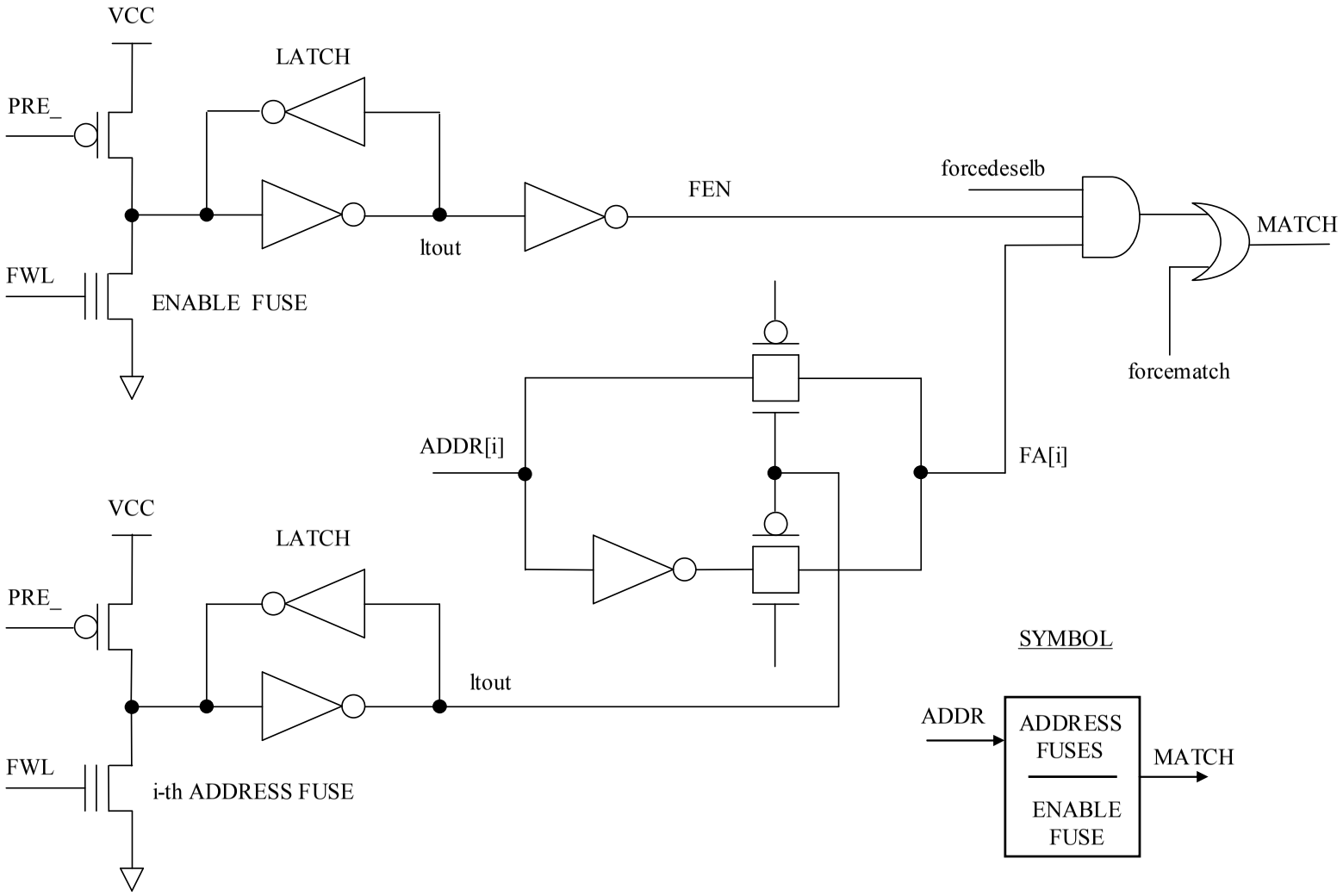


Fig. 4.3 - Schema a blocchi del circuito di match

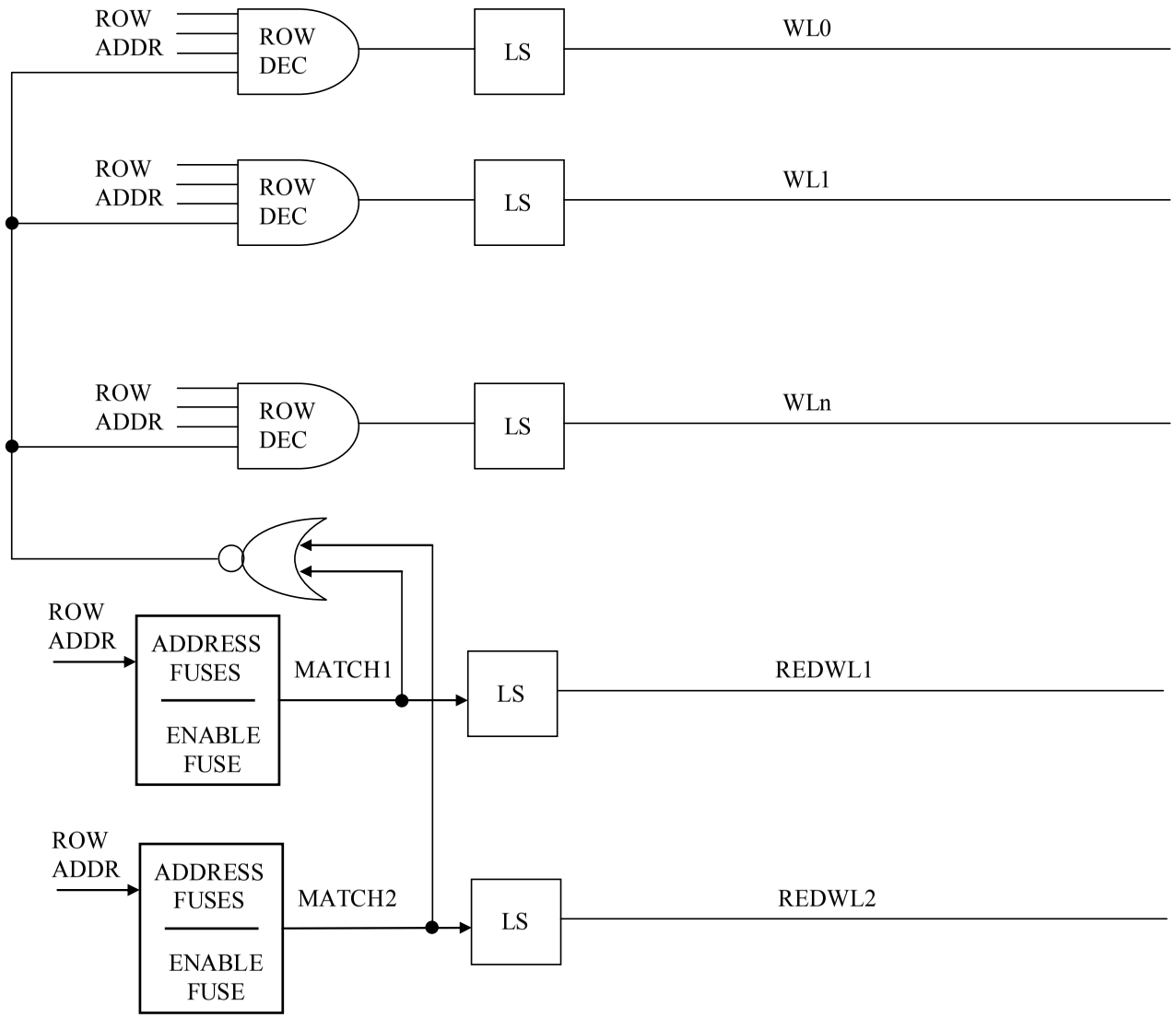


Fig. 4.4 - Schema a blocchi della ridondanza di riga

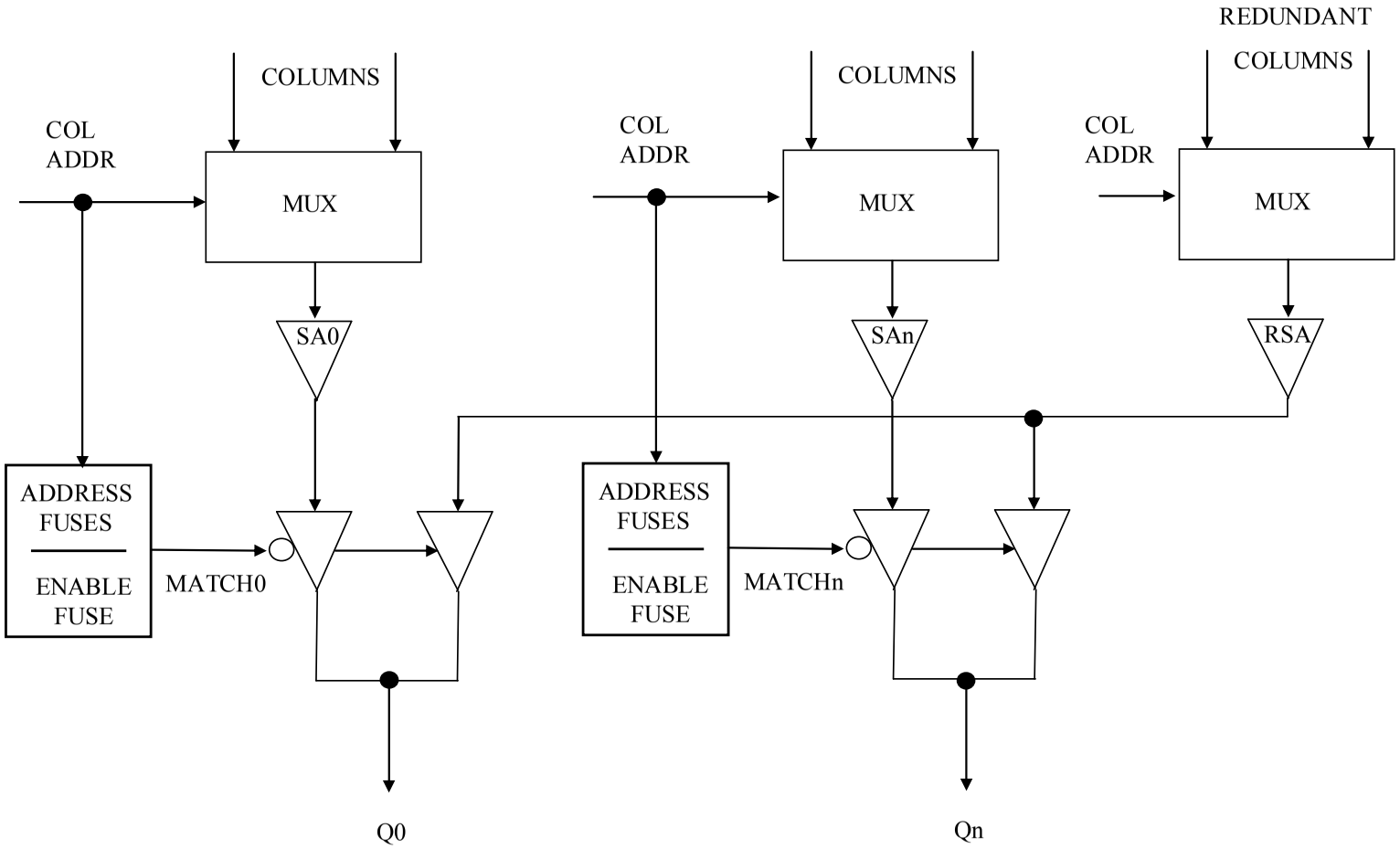


Fig. 4.5 - Schema a blocchi della ridondanza di colonna (circuiti di lettura)

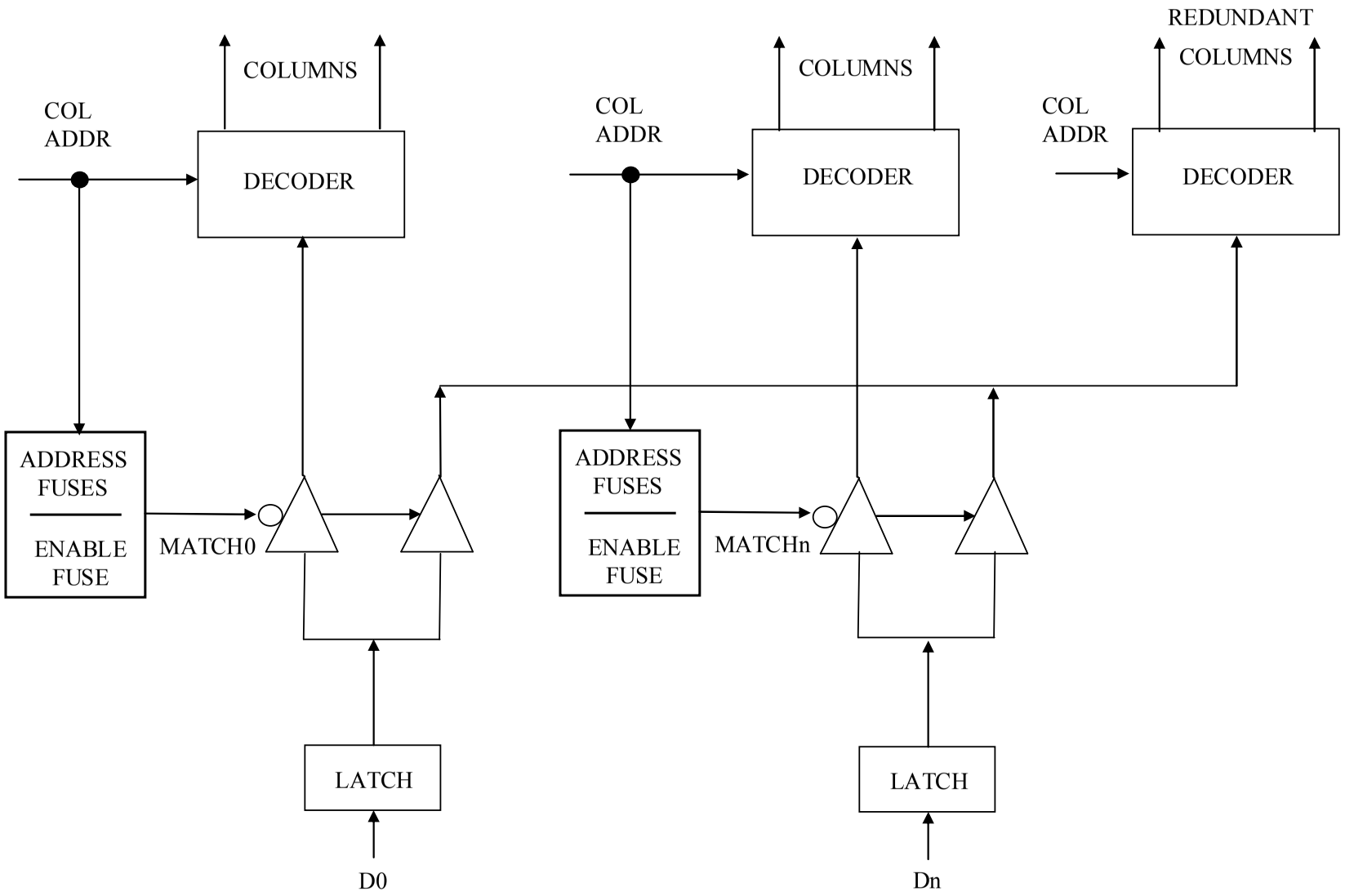


Fig. 4.6 - Schema a blocchi della ridondanza di colonna (circuiti di scritura)

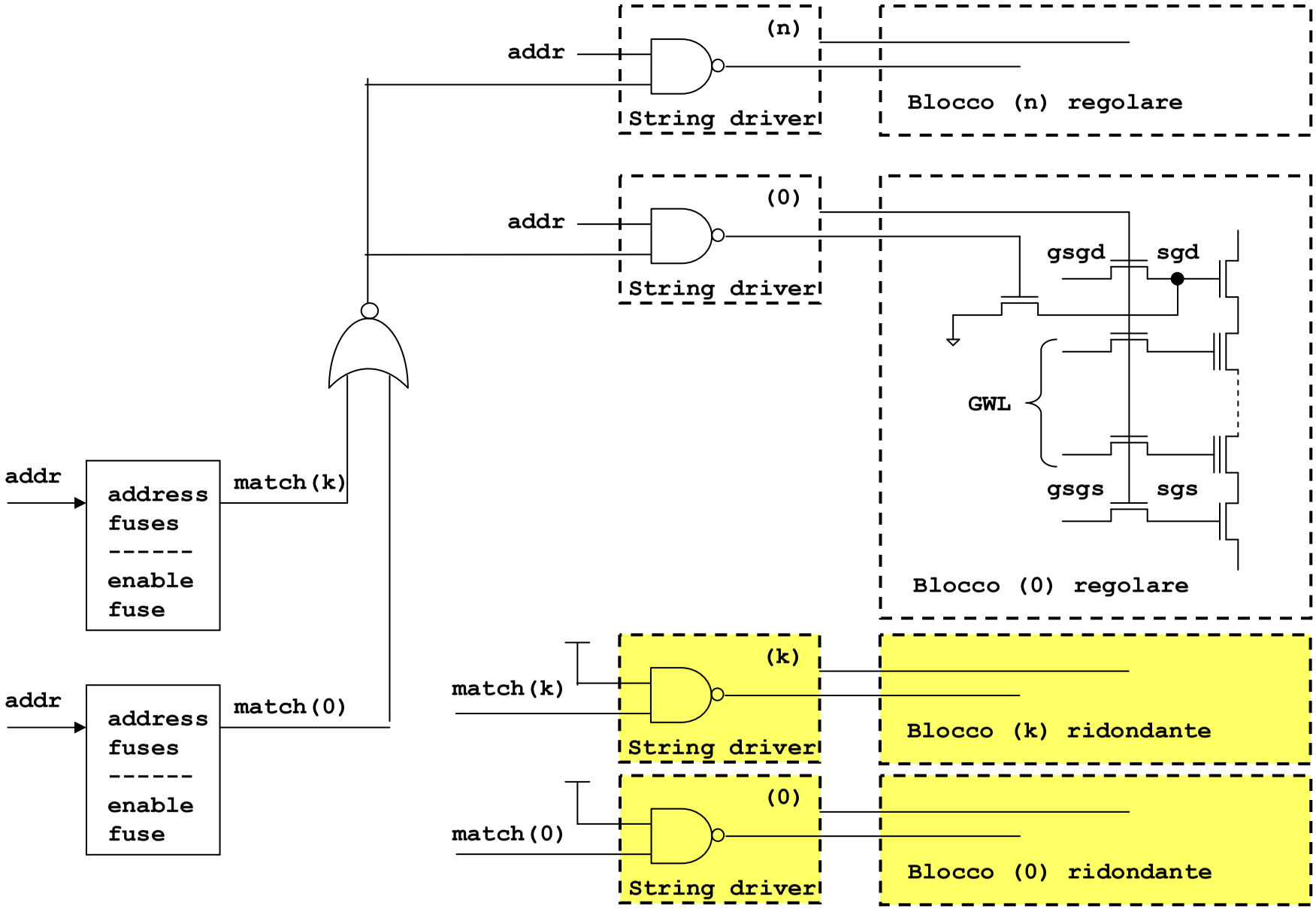


Fig. 4.7 - Schema di principo della ridondanza di blocco

5) TRIMS

Esempi di modi di test per effettuare operazioni di calibrazione o di ottimizzazione di strutture e operazioni :

5a. Calibrazione di un termometro (Fig.5.1-5.2-5.3-5.4)
Un termometro puo' essere usato in una memoria FLASH per effettuare compensazioni in temperatura di tensioni di programmazione o di lettura. Un termometro deve essere tarato per avere una funzione di trasferimento il piu' possibile indipendente dal processo.

5b. Aggiustamento della frequenza di un oscillatore (Fig. 5.5)
Un generatore di clock interno per la temporizzazione di un controllore deve poter essere tarato per avere una frequenza il piu' possibile indipendente dal processo.

5c. Trim della tensione di word line in un algoritmo di programmazione (Fig. 5.6a-5.6b-5.7)
La tensione degli impulsi di programmazione e di verifica deve poter essere aggiustata in funzione delle caratteristiche del processo.

5d. Trim della durata di impulsi (Fig. 5.8)
La durata degli impulsi di programmazione o cancellazione deve poter essere calibrata in funzione delle caratteristiche del processo per evitare operazioni inefficienti.

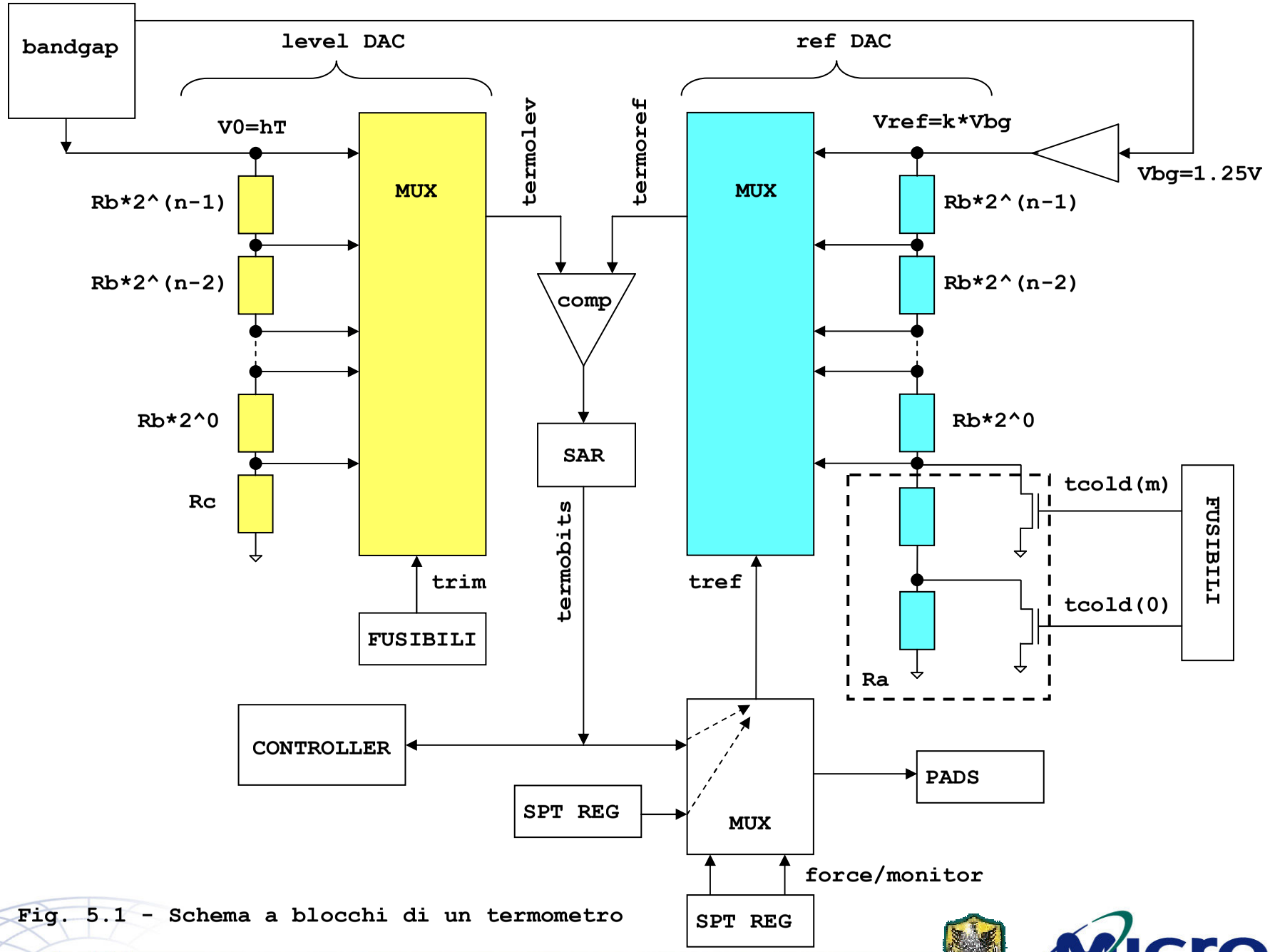


Fig. 5.1 - Schema a blocchi di un termometro

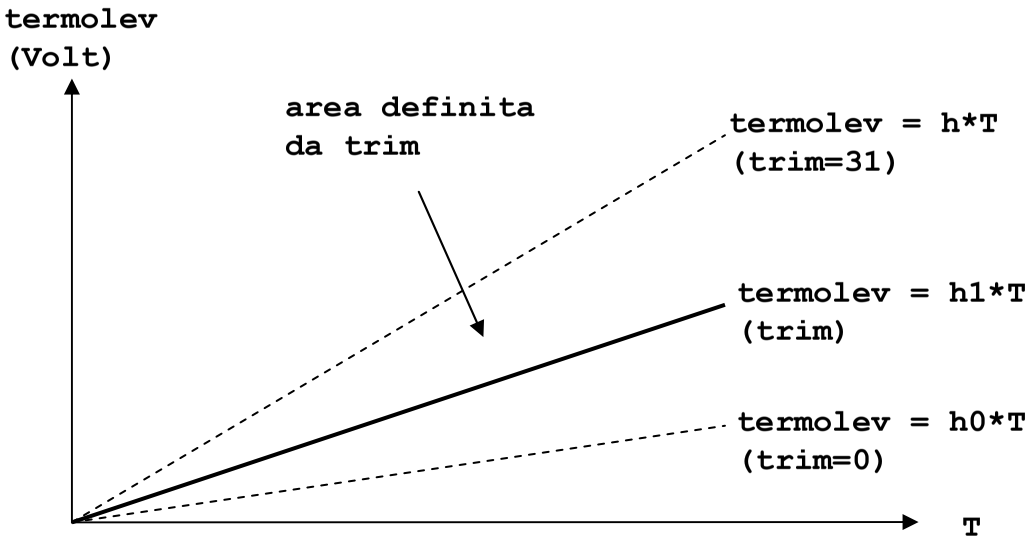


Fig. 5.2 -
Curva di funzionamento
del DAC di trasduzione
di livello (level DAC)
Esempio : n=5

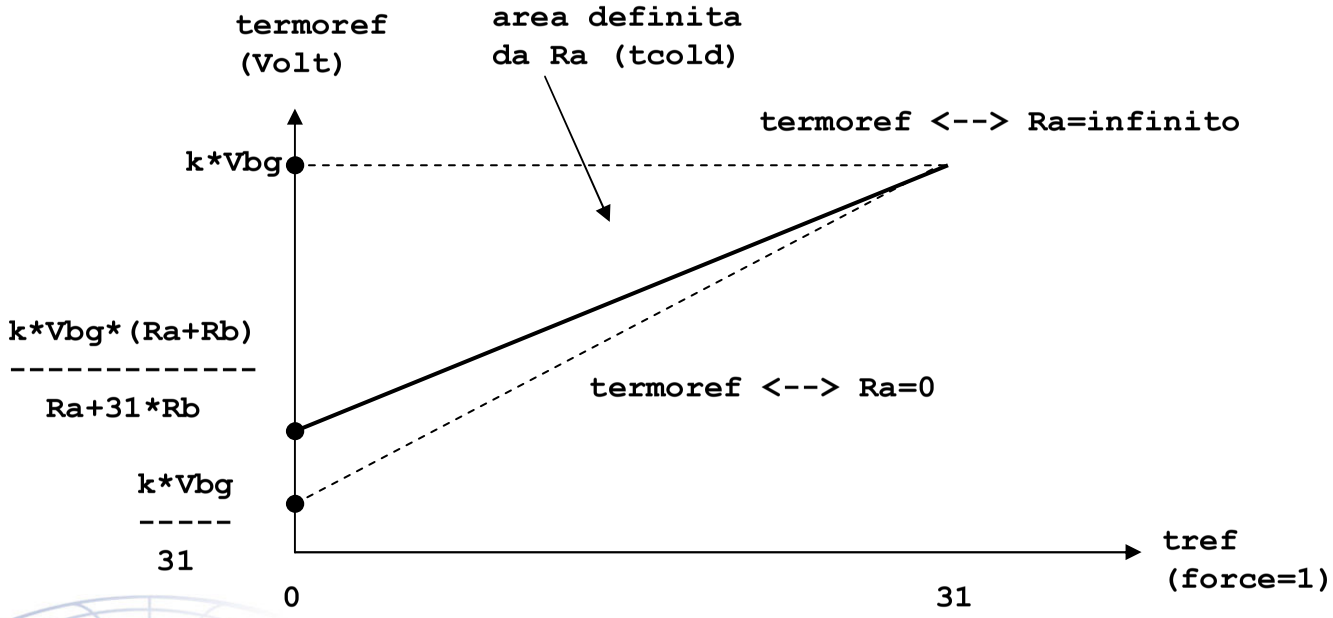


Fig. 5.3 -
Curva di funzionamento
del DAC di trasduzione
del riferimento (ref DAC)
Esempio : n=5

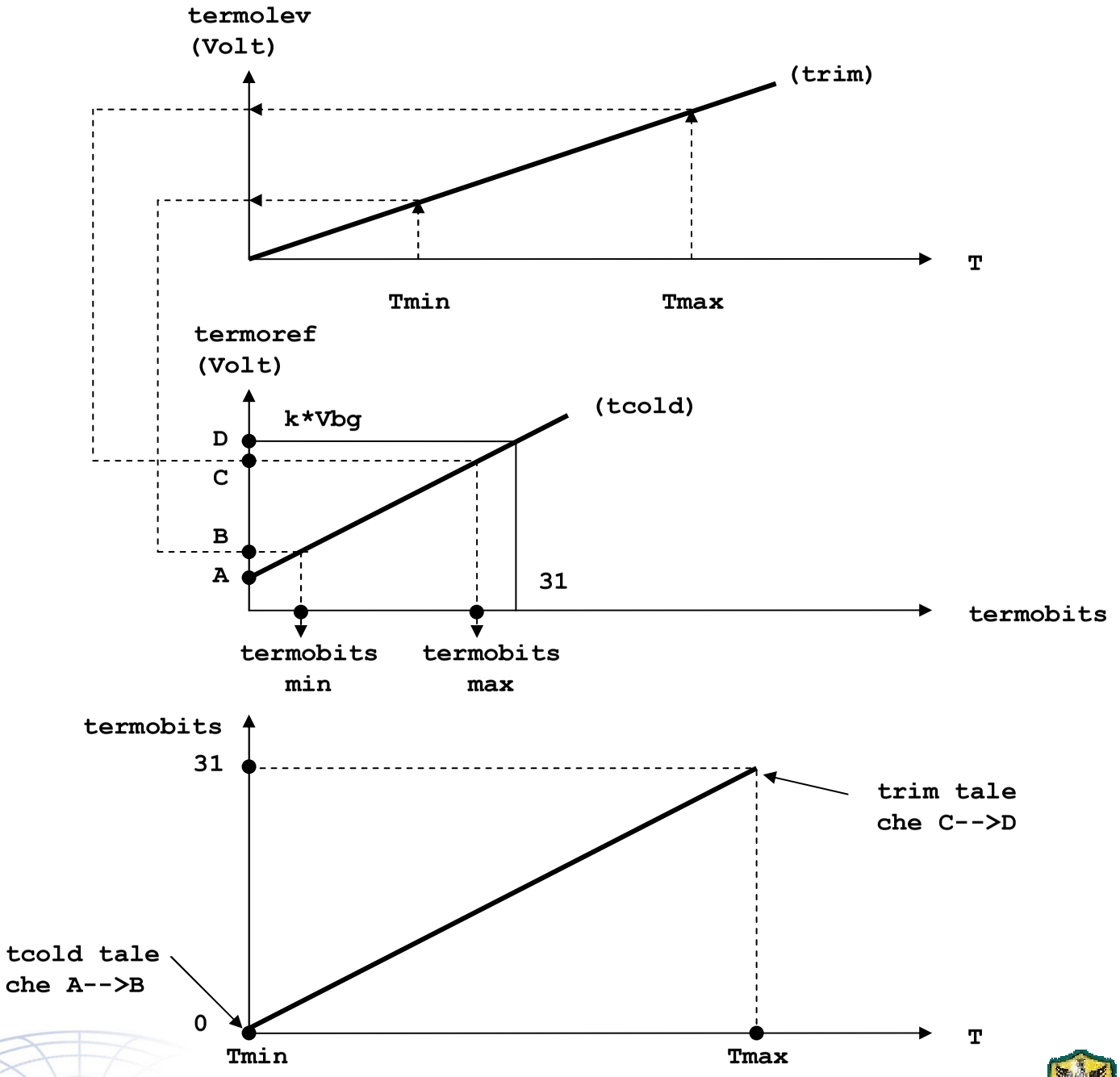


Fig. 5.4 - Calibrazione del termometro

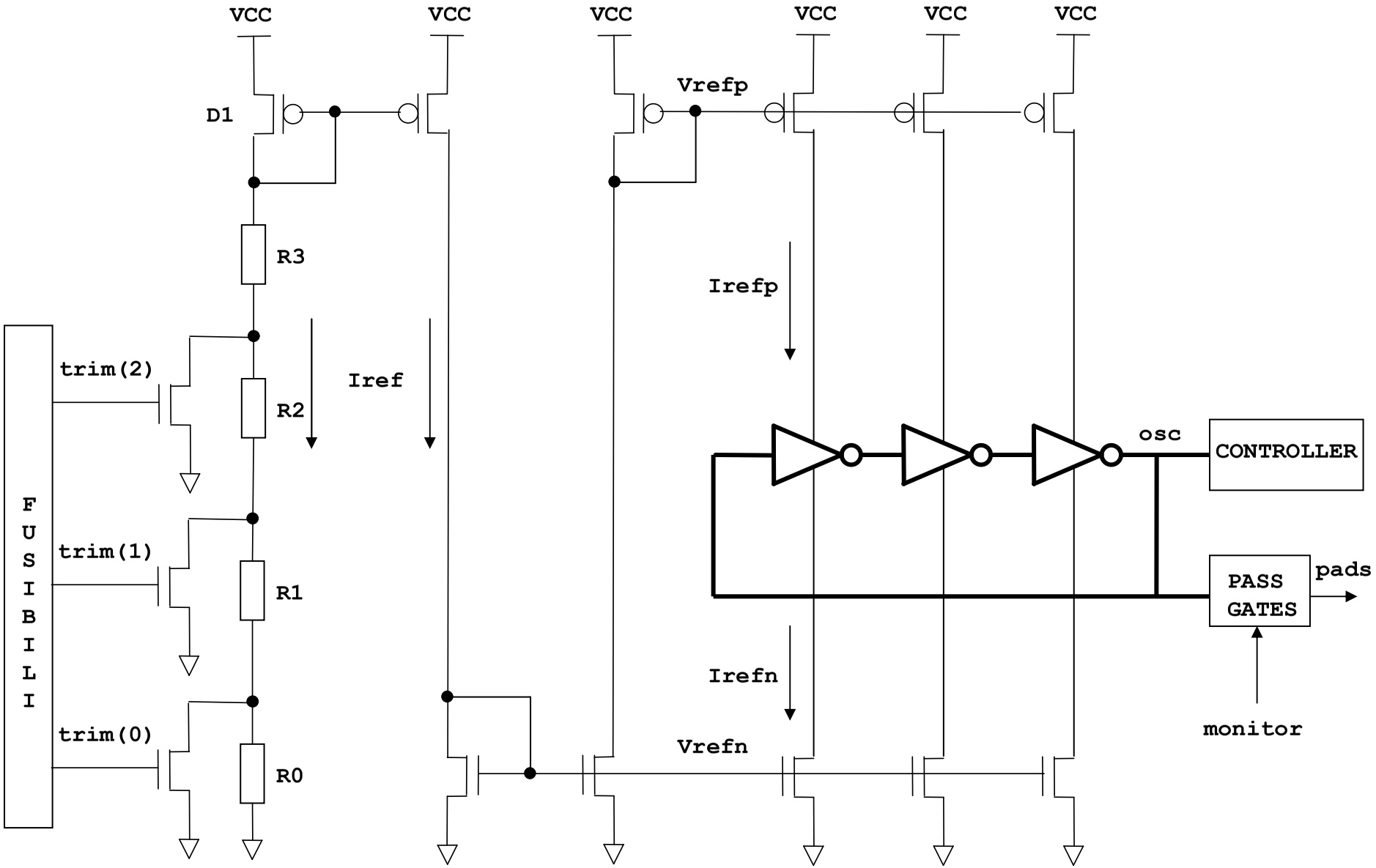


Fig. 5.5 - Variazione della frequenza di un oscillatore

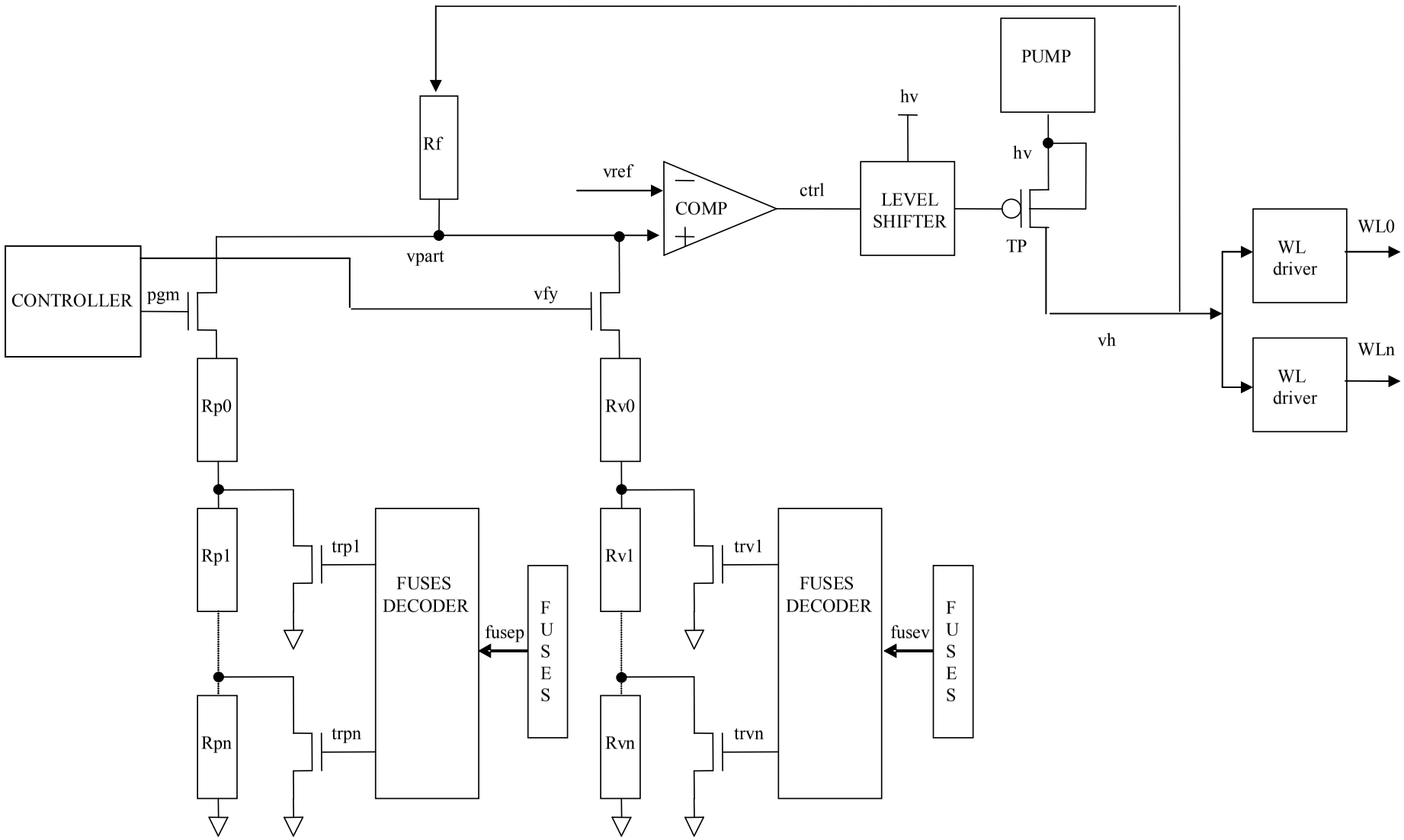


Fig. 5.6a - Trim della tensione di WL in un algoritmo di programmazione (organizzazione modale)

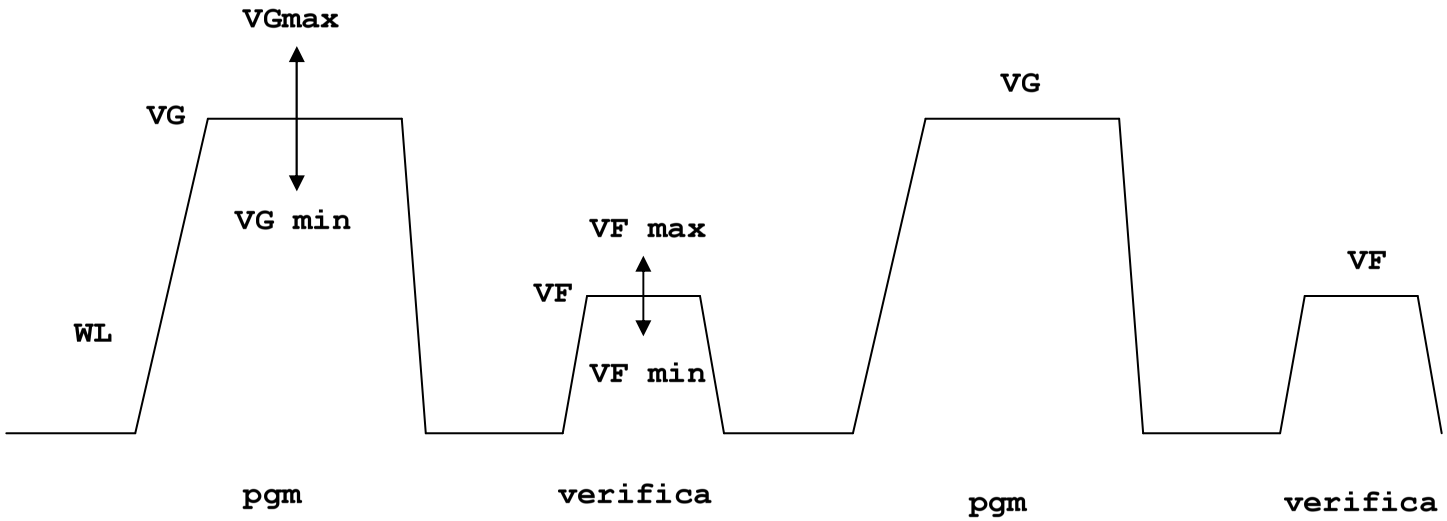


Fig. 5.6b - Forme d'onda della tensione di WL in un algoritmo di programmazione (organizzazione modale - FLASH singolo livello)

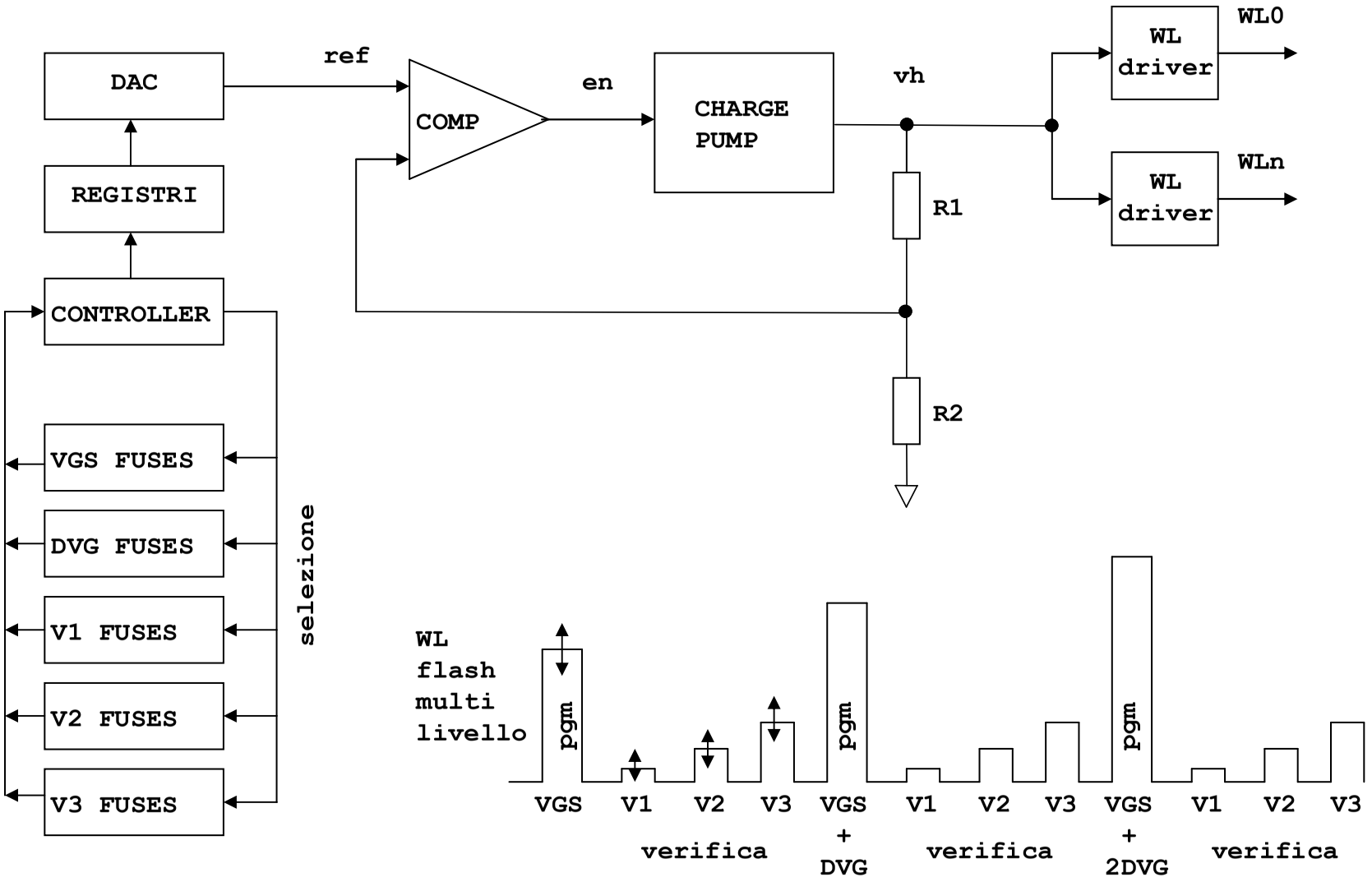


Fig. 5.7 - Trim della tensione di WL in un algoritmo di programmazione (organizzazione a registri)

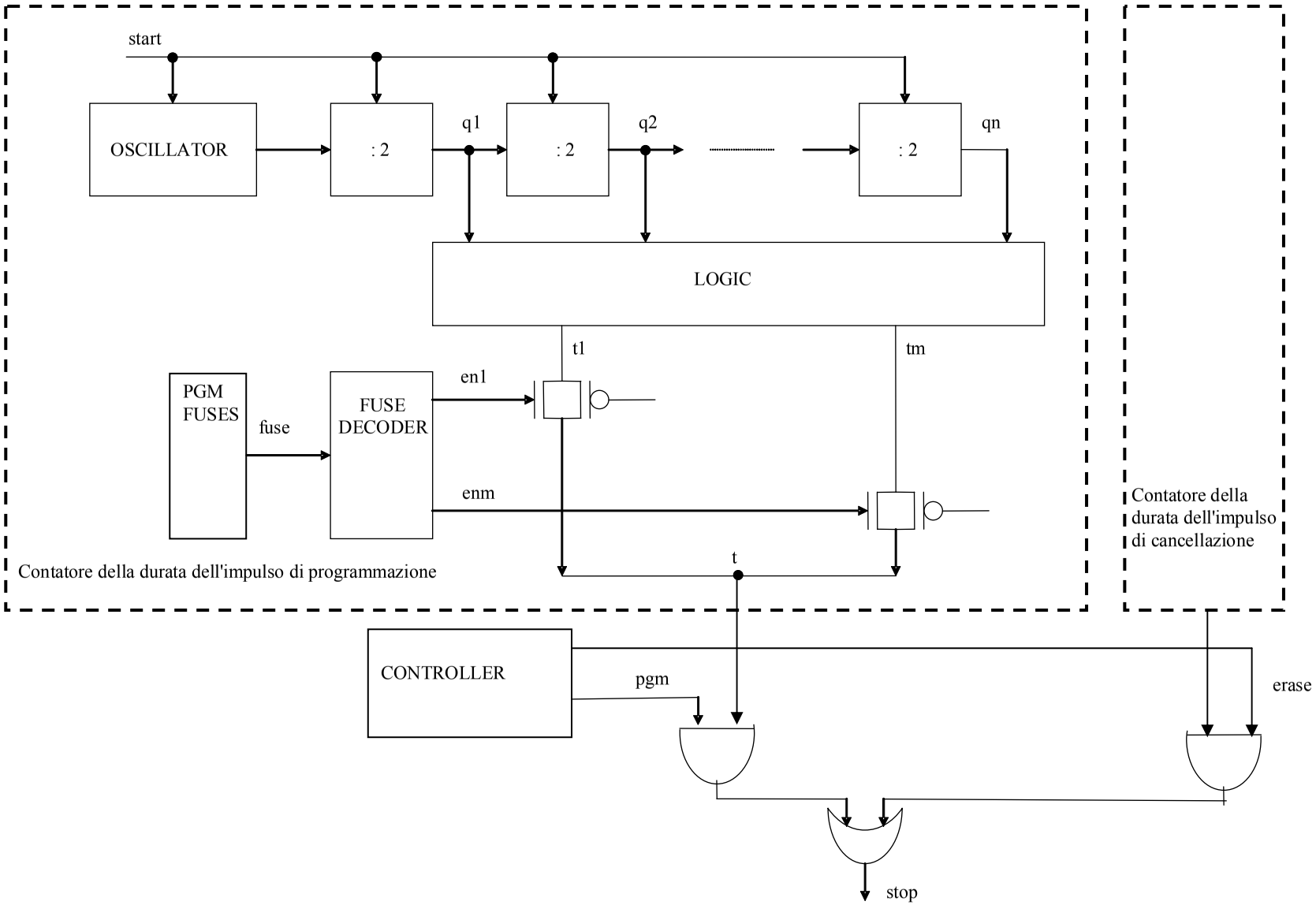


Fig. 5.8 - Trim della durata di impulsi di programmazione e di cancellazione (organizzazione modale)

6) ALGO SKIPS (Fig. 6.1)

Con l'uso di fusibili e' possibile condizionare l'evoluzione di un algoritmo allo scopo di :
. saltare dei passi
. seguire percorsi alternativi
. attivare multilivello o singolo livello
. fermare in modo diverso una operazione di erase (parte alta o bassa di una distribuzione)
. applicare tecniche di rallentamento (slow program)
. variare i time-out

7) MONITOR E FORZAMENTO DI TENSIONI (Fig. 7.1-7.2)

In test mode e' interessante misurare ai pads valori di tensione generati internamente per vedere se pompe e regolatori funzionano correttamente e per effettuarne il trimming. E' anche interessante forzare dai pads alcune tensioni interne per effettuare operazioni su piu' blocchi in contemporanea.

8) FORZAMENTO ESTERNO DELLE DURATE DI IMPULSI (Fig. 8.1)

Imporre a piacimento dall'esterno le durate dei singoli impulsi di program o erase puo' essere utile per effettuare stress di parti della memoria.

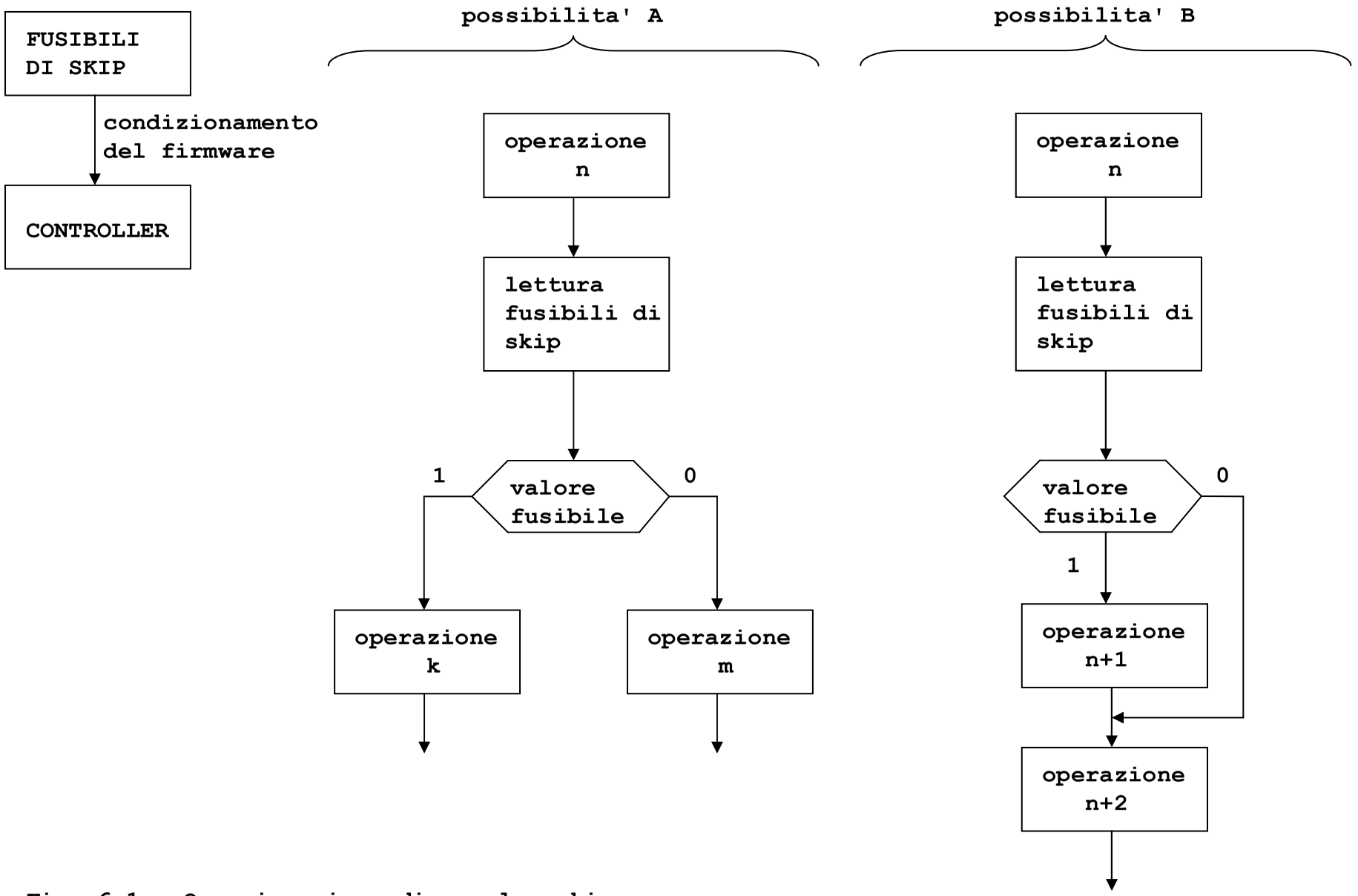


Fig. 6.1 - Organizzazione di un algo skip

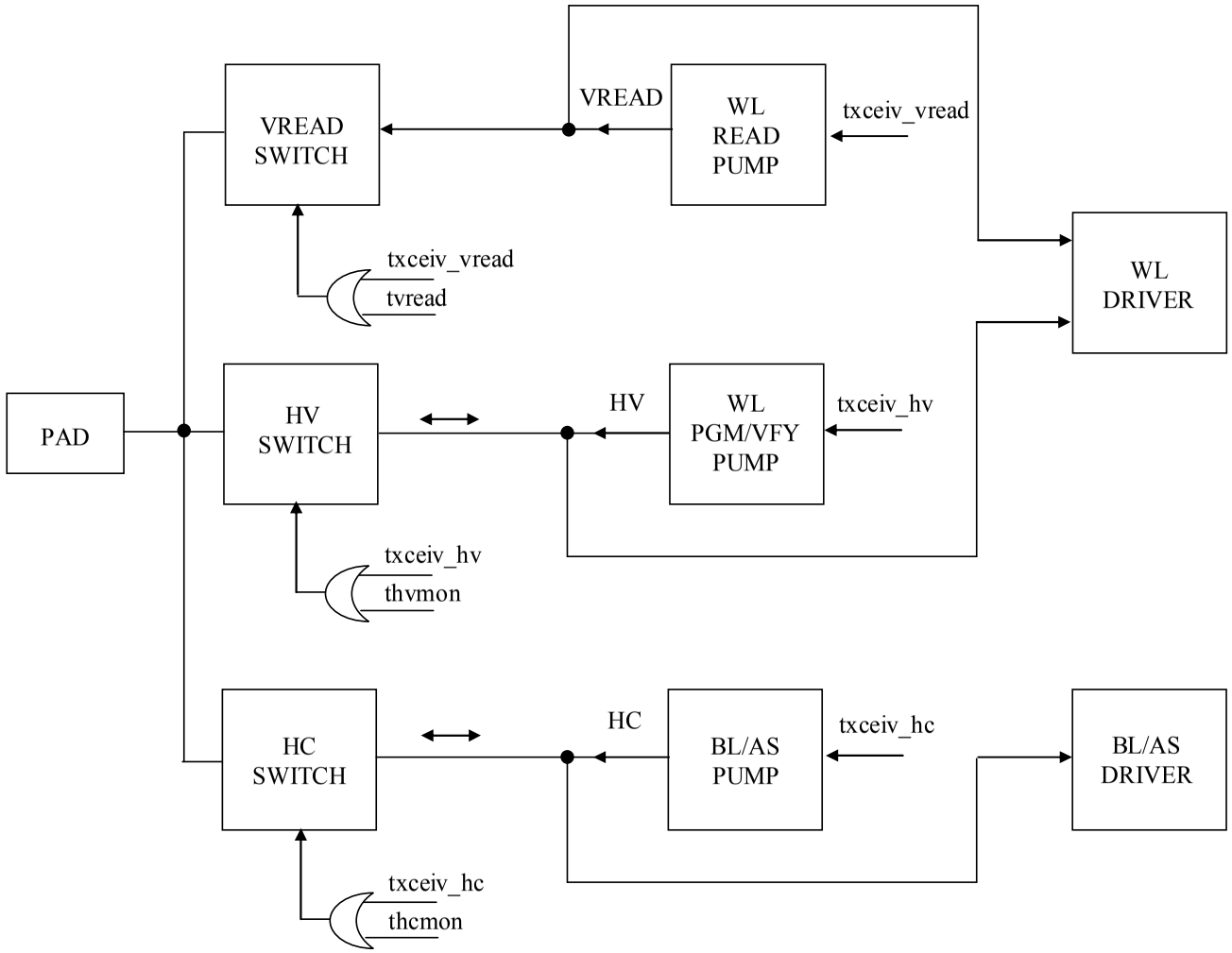


Fig. 7.1 - Schema a blocchi di un forzamento e monitor di un'alta tensione.

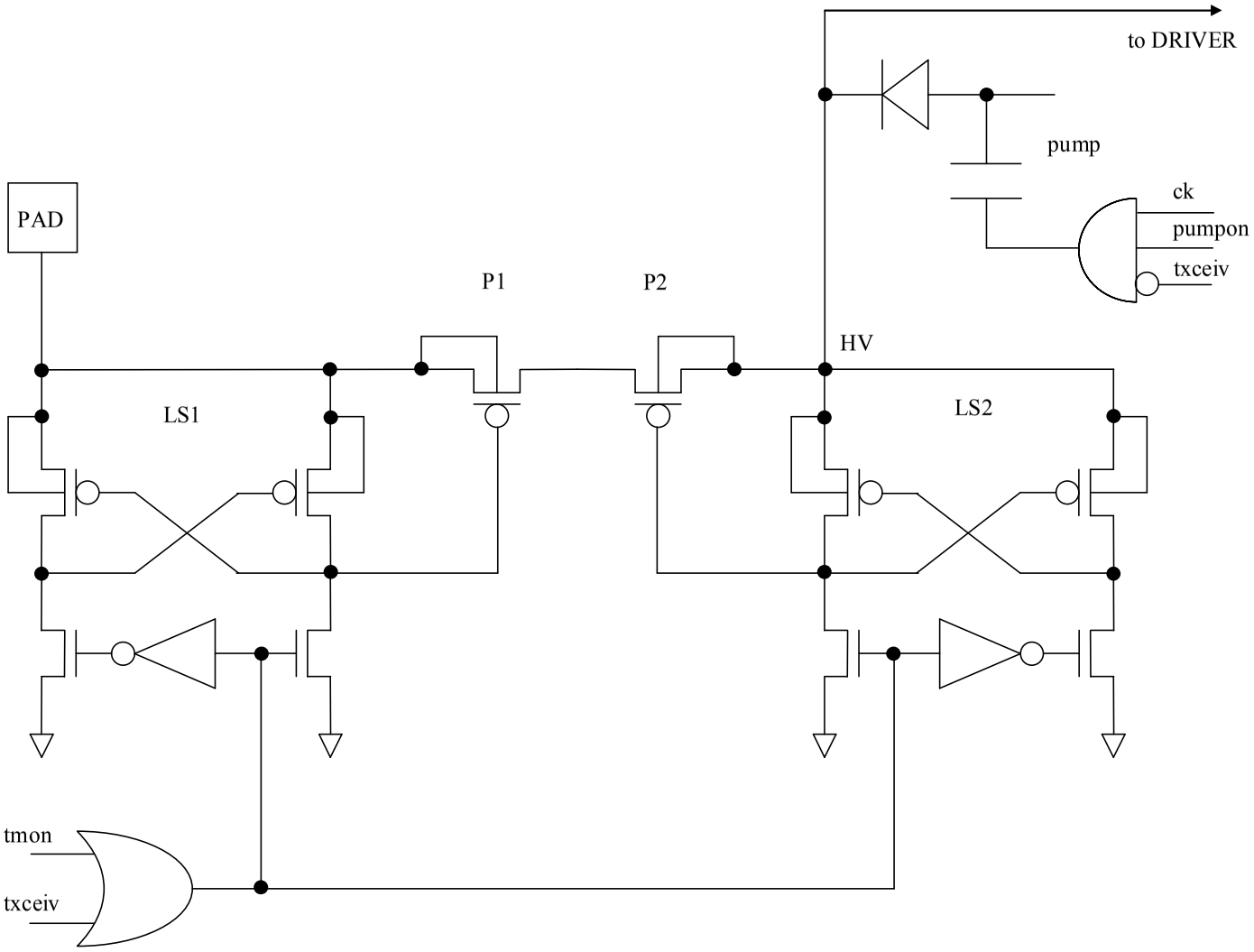


Fig. 7.2 - Circuito di switch per un forzamento e monitor di un'alta tensione.

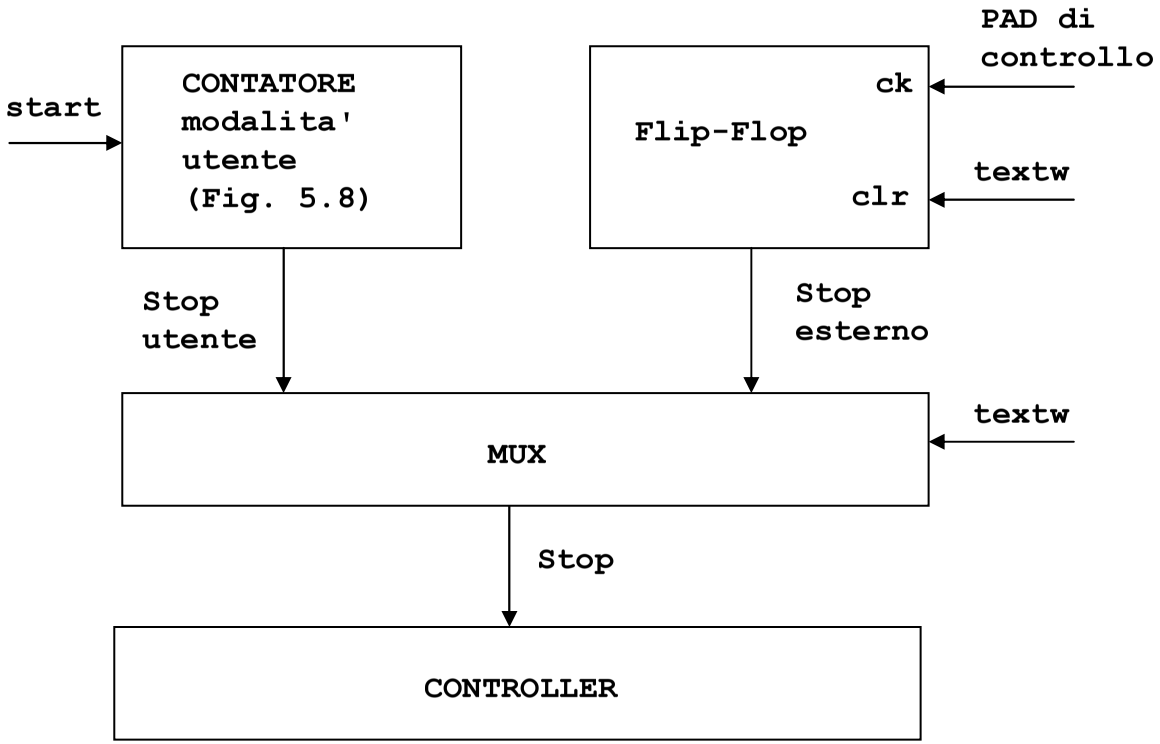
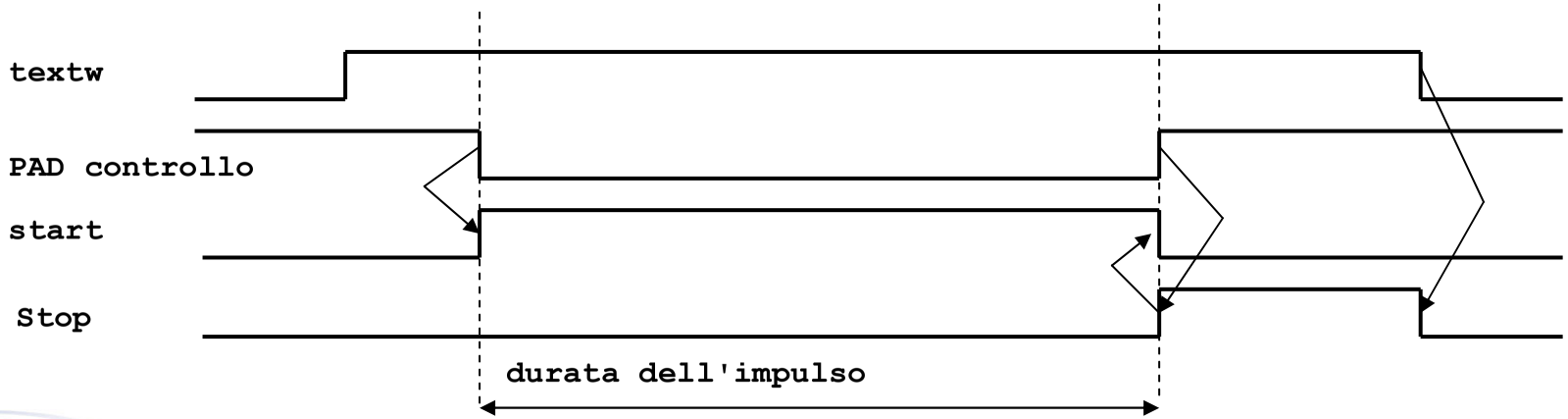


Fig. 8.1 - Forzamento esterno delle durate di impulsi in test modale



9) ACCESSO DIRETTO IN ARRAY (Fig. 9.1)

Il test mode di accesso diretto in array permette di misurare la tensione di soglia VT di una cella o stringa.

La bit line BL si polarizza dall'esterno ad un valore fisso, la word line WL si fa variare dall'esterno da 0V ad un valore alto. Il valore di WL corrispondente ad un valore di corrente di BL di 1uA e' la VT.

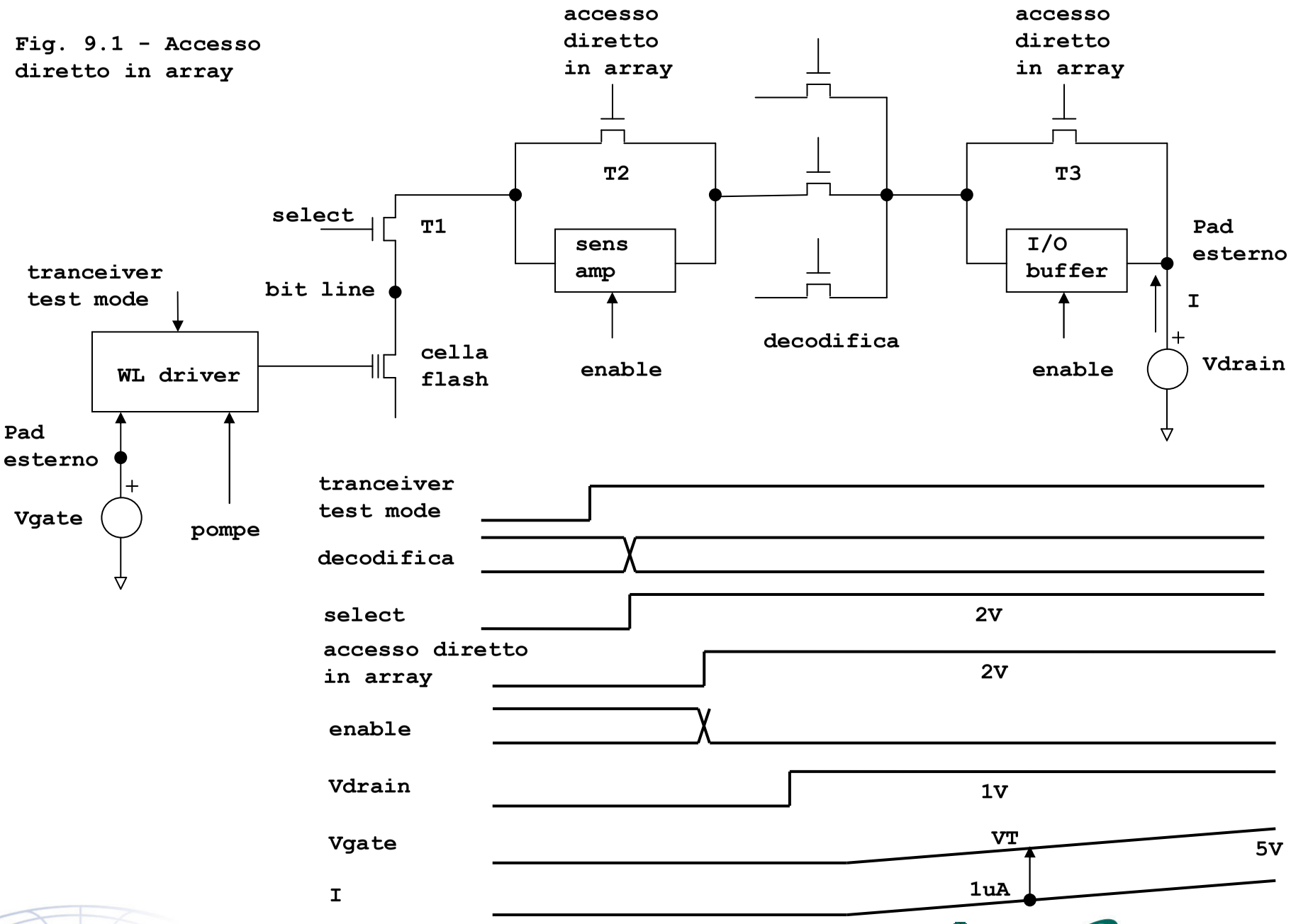
10) MODALITA' DI STRESS

L'array di memoria FLASH puo' essere sottoposto a varie modalita' di stress : gate, drain, source stress.

In queste modalita' un terminale viene sottoposto ad alta tensione mentre gli altri sono posti a gnd.

La distribuzione di VT dopo stress e' spostata verso il basso. L'entita' di questo spostamento e' indicativa della bonta' di ossidi e giunzioni.

Fig. 9.1 - Accesso diretto in array



11) TECNICHE DI COMPRESSIONE

Per ridurre il tempo di test di memorie si adottano tecniche di compressione. Con una sola operazione di lettura si acquisiscono informazioni relative al contenuto di tutte le parole di una pagina.

11a. Compressione di parola (Fig. 11.1-11.2-11.3-11.4-11.5)

Se tutti i bit di posizione i di una parola sono 1 (oppure 0) l'uscita Q[i] e' 1 (oppure 0), altrimenti e' 3-state.

11b. Verifica interna (IVR) (Fig. 11.6-11.7-11.8)

Se tutti i bit di una pagina letta corrispondono ad una pattern di riferimento si attiva un segnale interno di 'pass'.

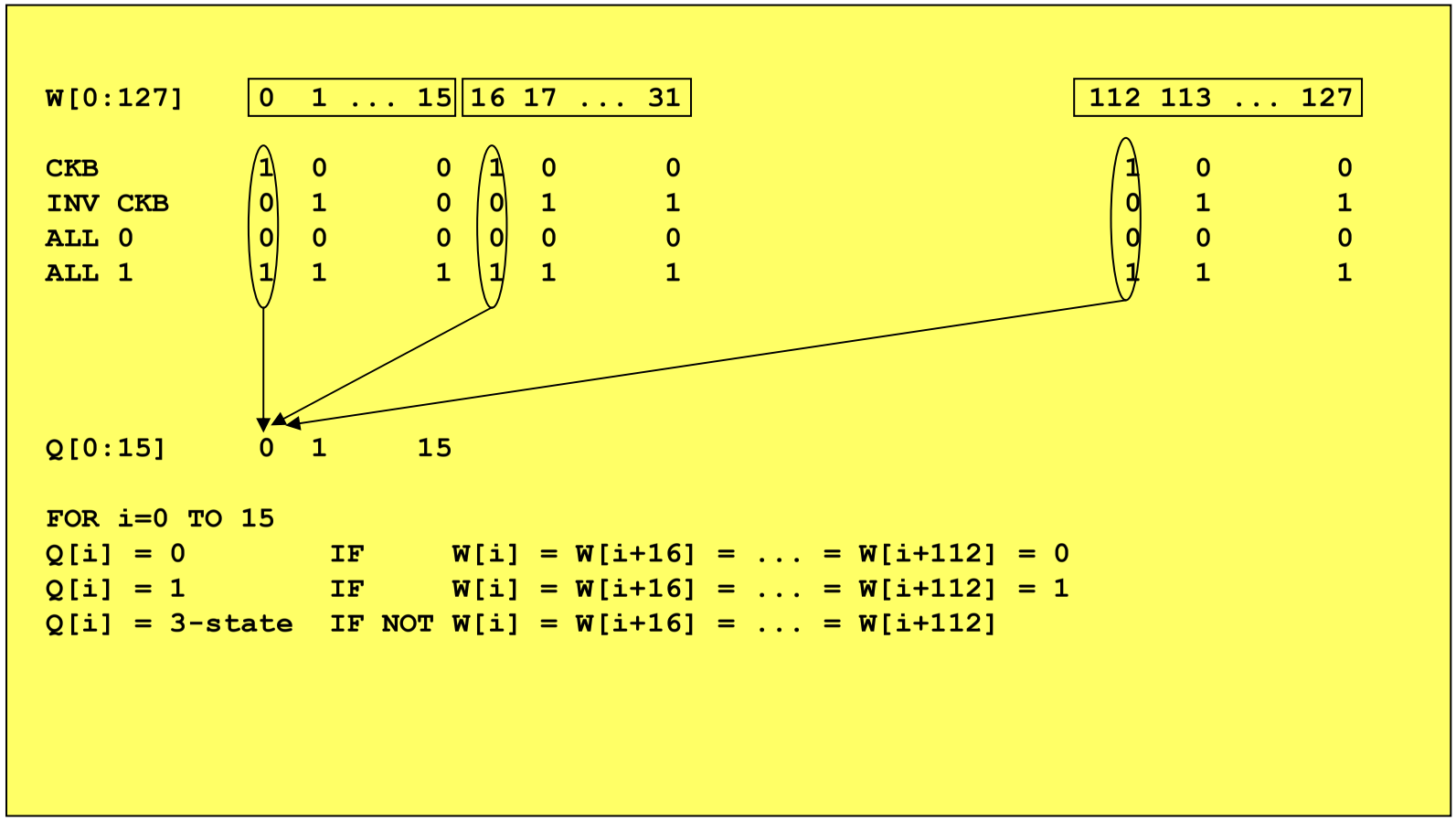
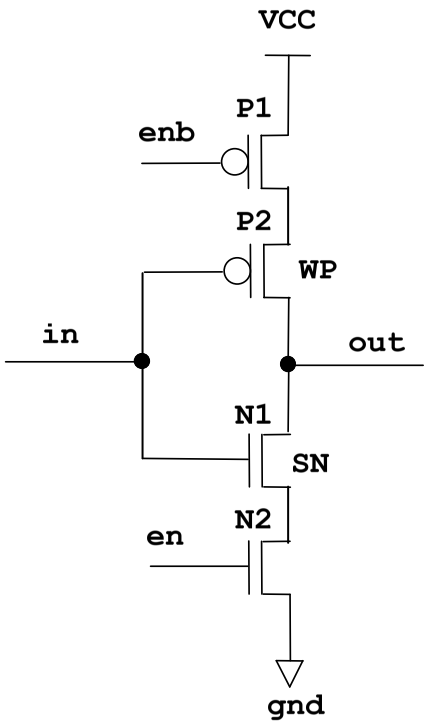
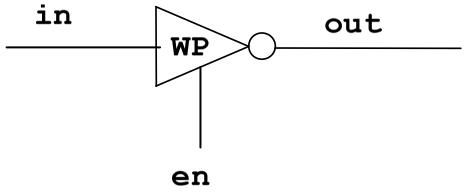


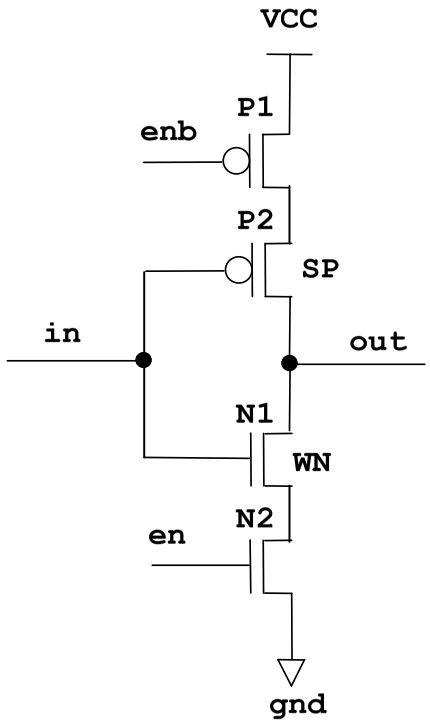
Fig. 11.1 - Tabella illustrativa della compressione di word



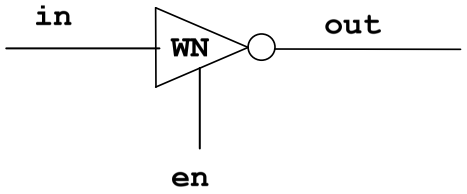
(a) - inverter weak P



(b) - simbolo di un inverter weak P

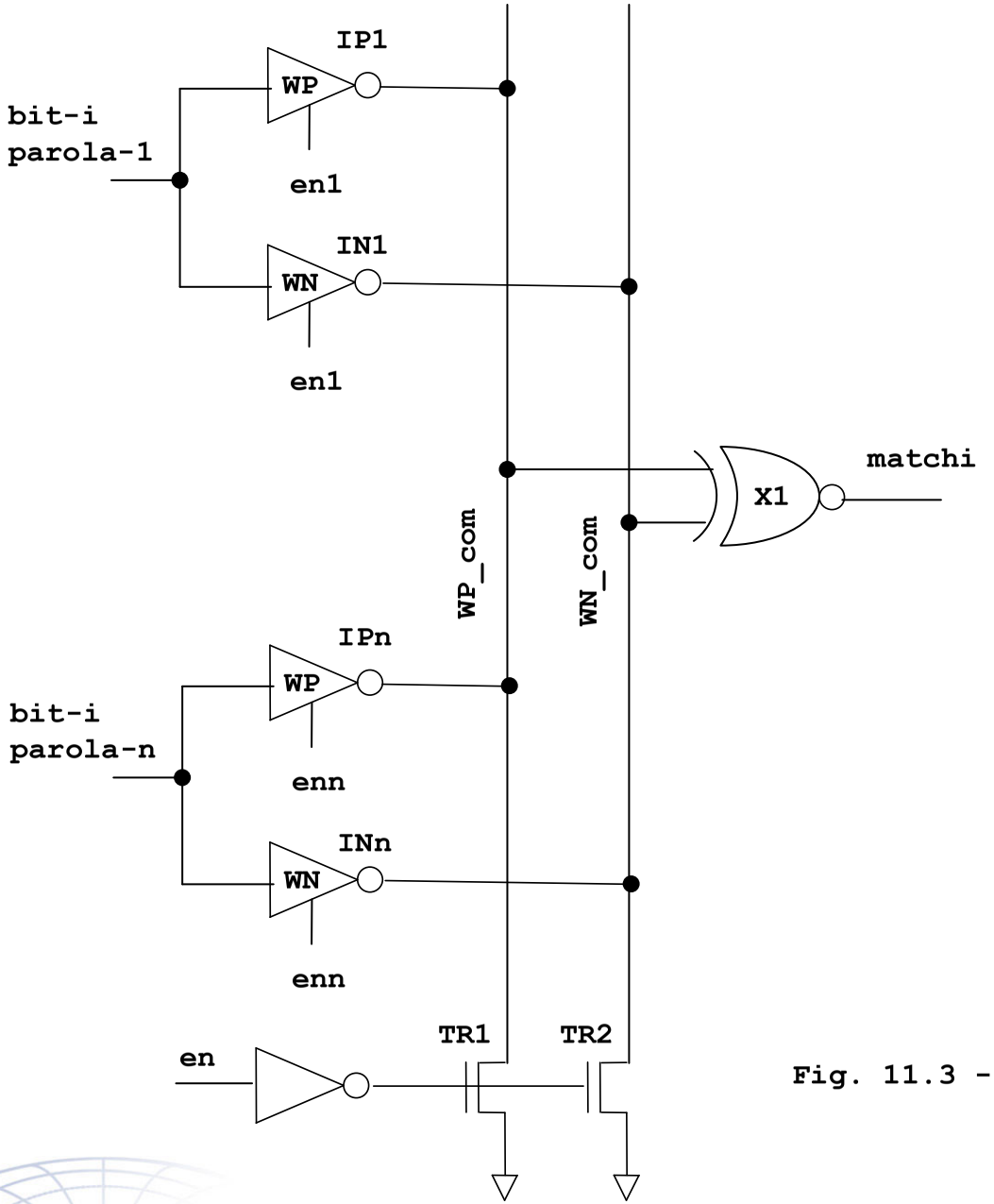


(c) - inverter weak N



(d) - simbolo di un inverter weak N

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



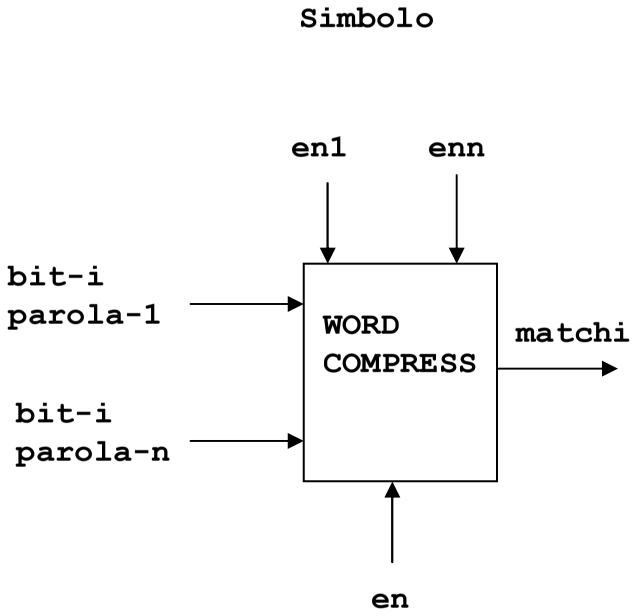
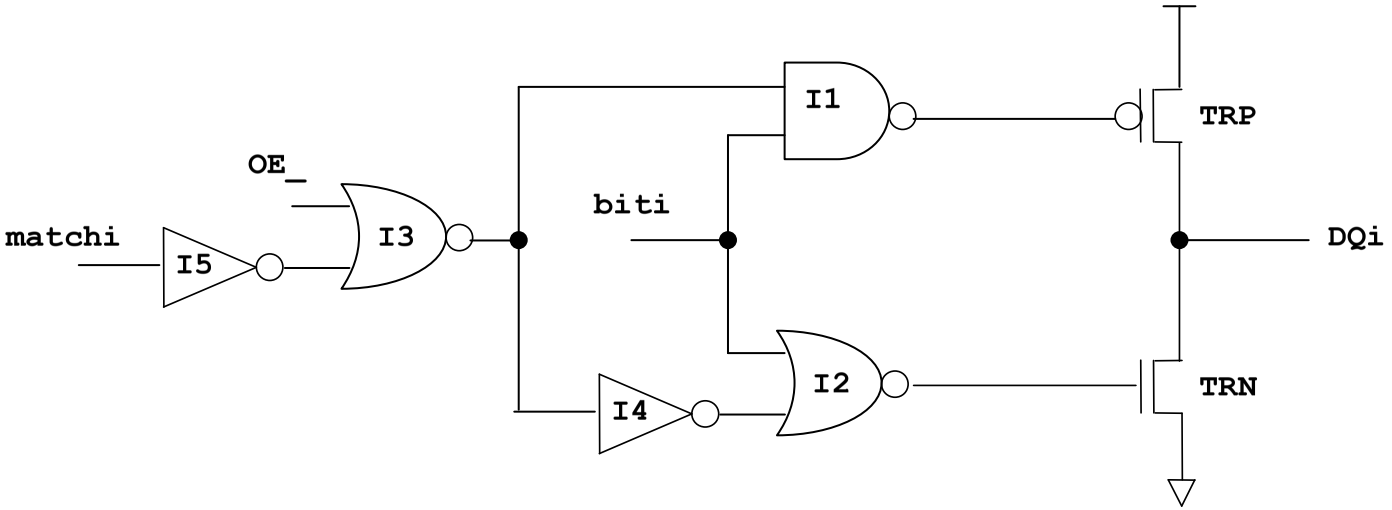


Fig. 11.3 - Compressione dell'i-mo bit in n parole.



Simbolo

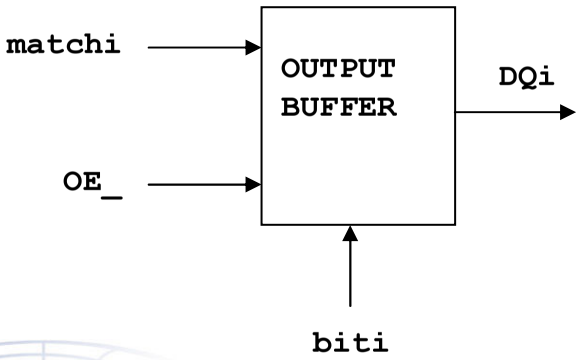


Fig. 11.4 - Output buffer con segnale di match per la compressione di parola.

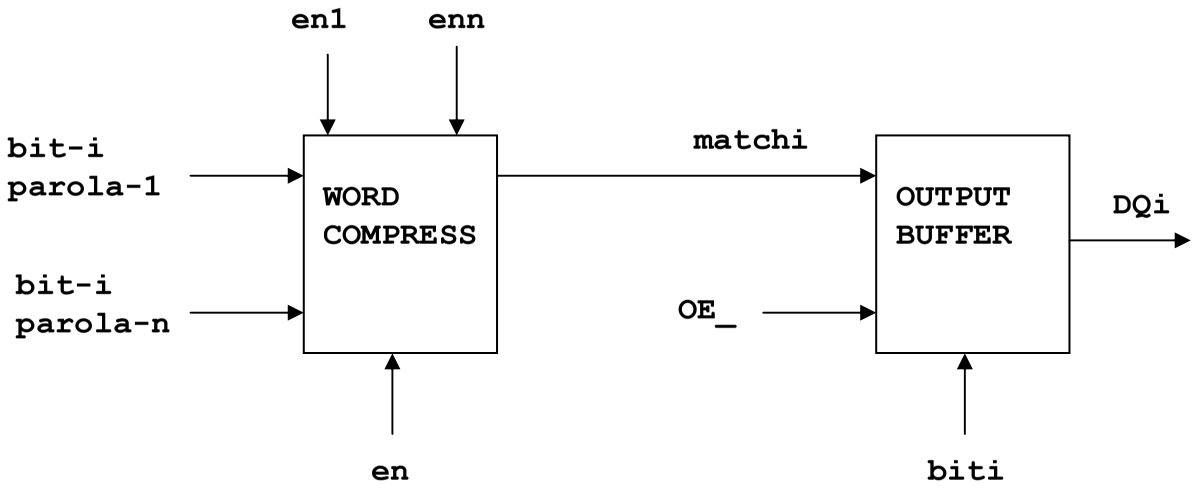


Fig. 11.5 - Organizzazione della compressione di n parole (i-mo bit)

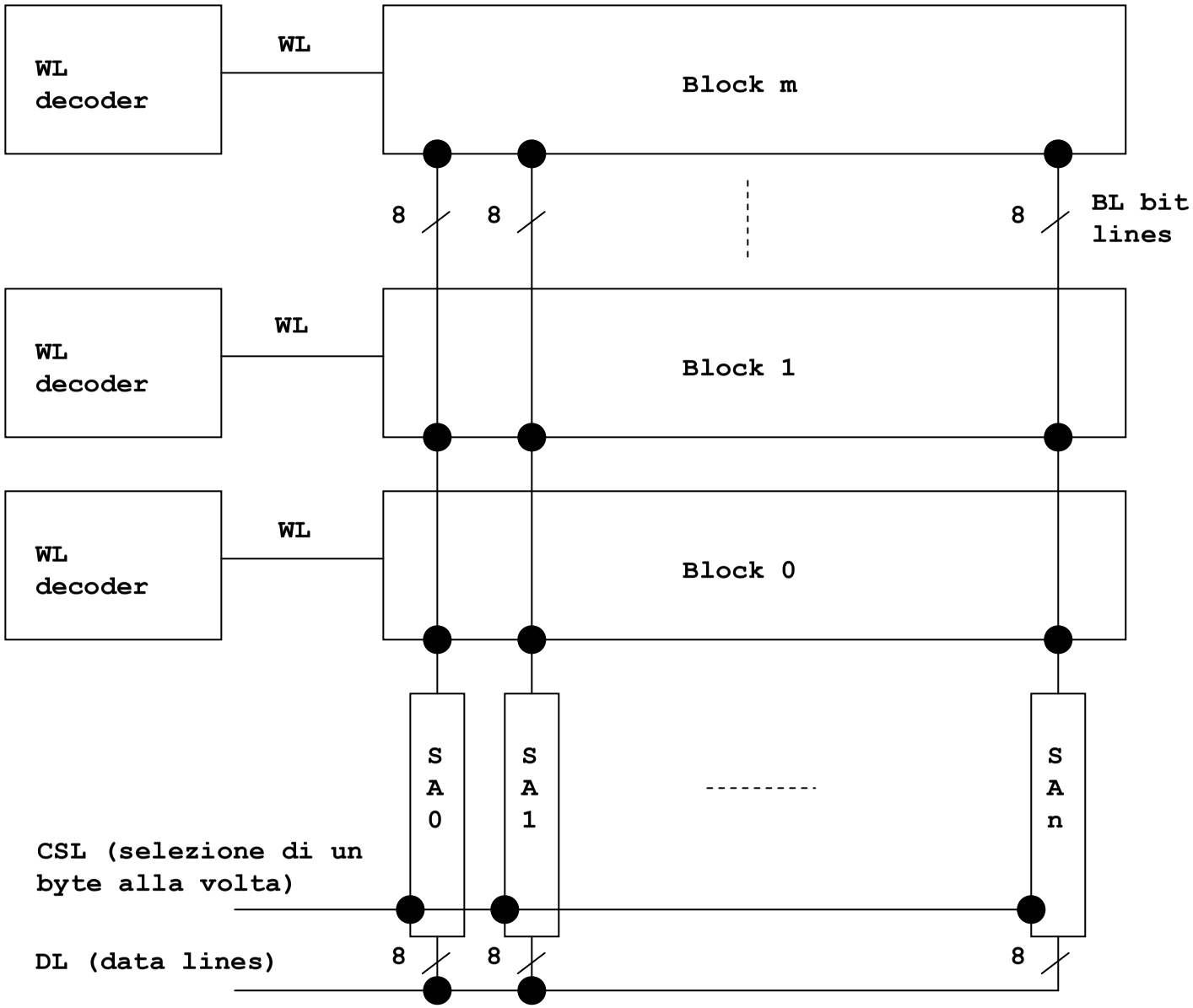


Fig. 11.6 - Organizzazione di blocco e pagina in una memoria NAND FLASH

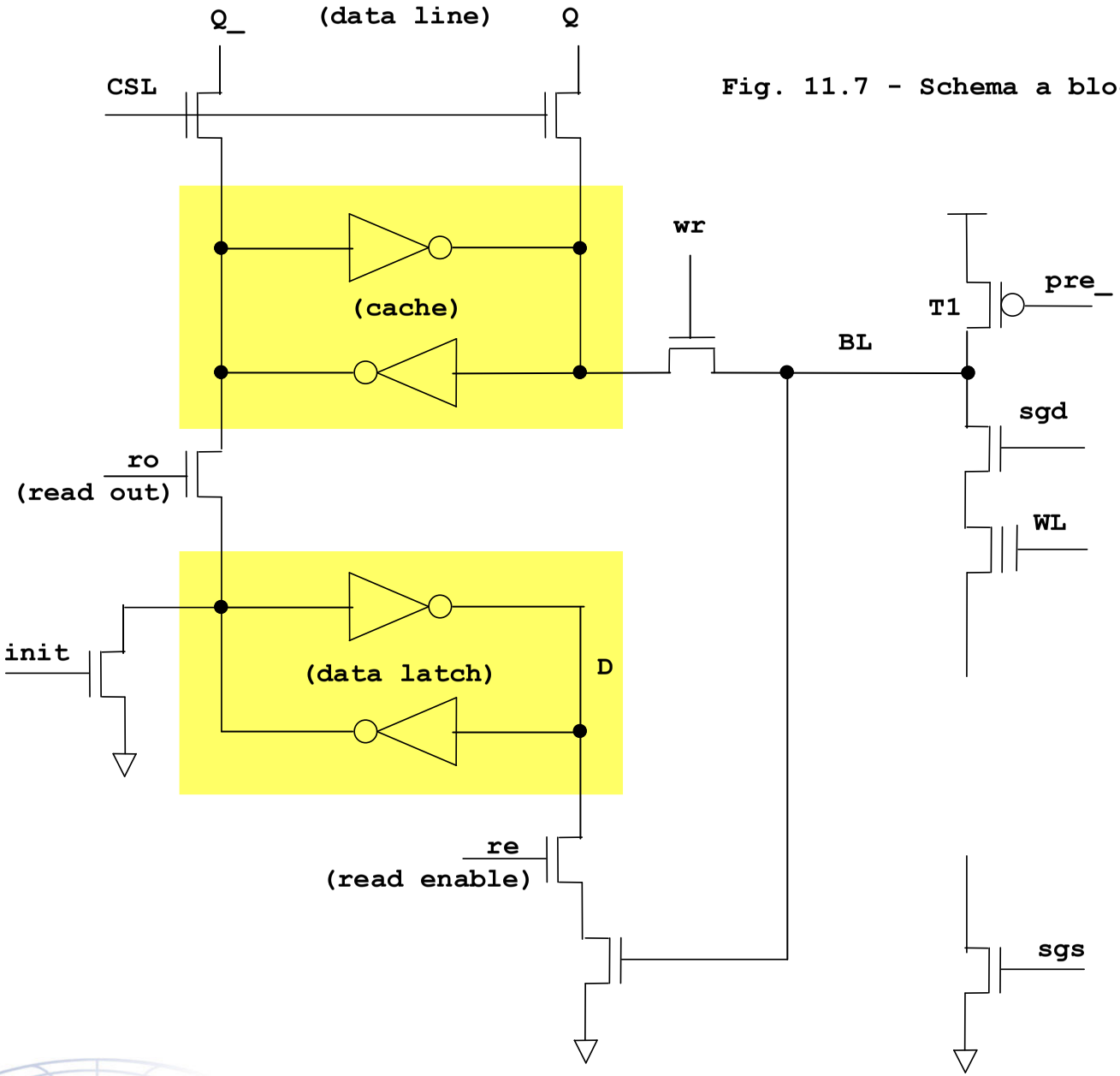
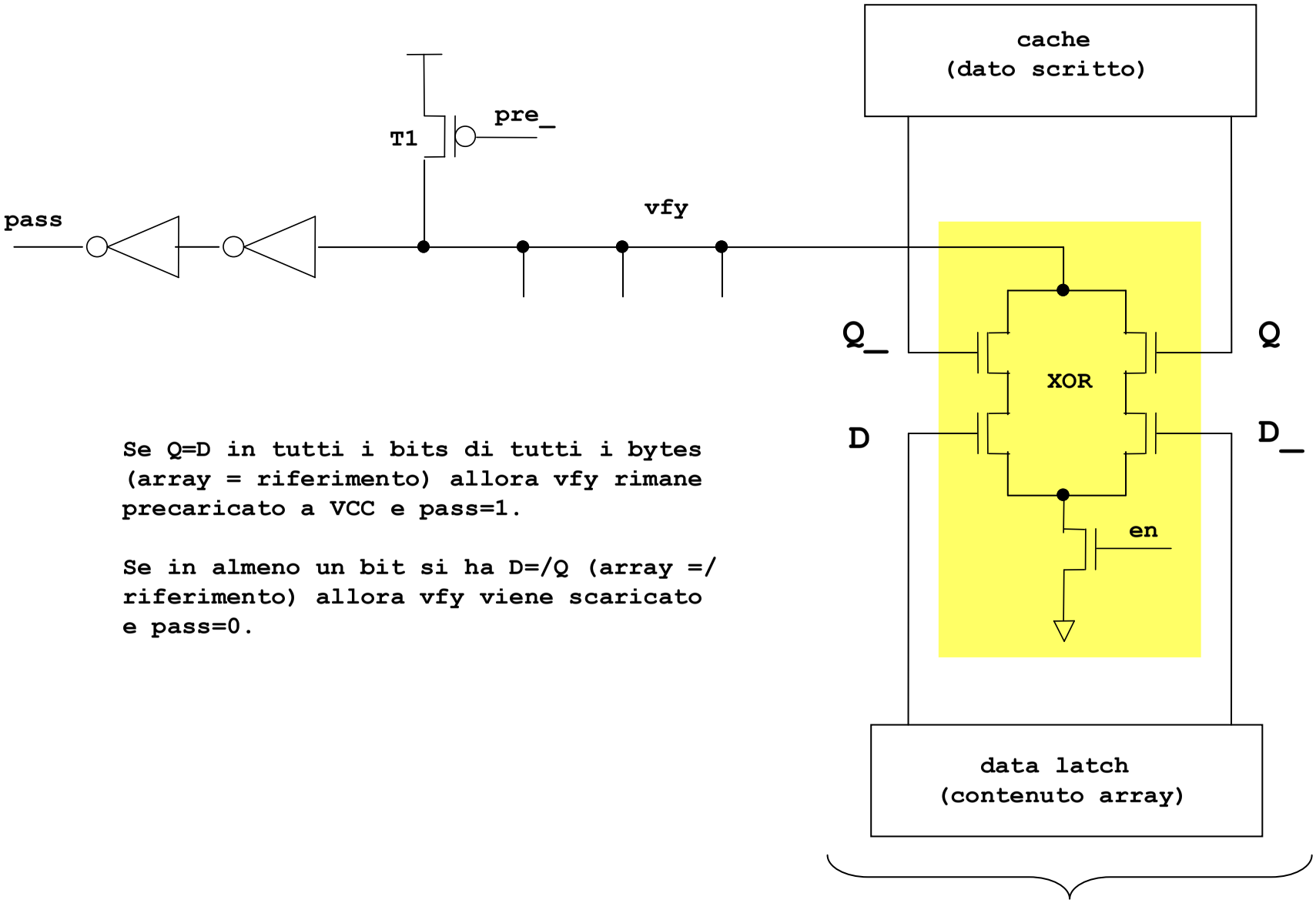


Fig. 11.7 - Schema a blocchi del sens amplifier.



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=/Q (array =/ riferimento) allora vfy viene scaricato e pass=0.

Fig. 11.8 - Schema a blocchi della struttura di IVR

12) TECNICHE DI PARALLELIZZAZIONE

Per ridurre il tempo di test si possono testare piu' memorie in contemporanea. Una volta richiesto il test a tutte le memorie, ognuna di esse esegue autonomamente il test sotto la supervisione di un controllore interno (SED = self error detect) memorizzando internamente il risultato in un latch associato ad ogni blocco (BBF = bad block flag) che puo' alla fine essere velocemenmte letto dalla macchina di test.

12a. SED con verifica interna (Fig. 12.1-12.2-12.3)

Permette di verificare velocemente e in parallelo se il contenuto di memorie corrisponde ad una pattern di riferimento

12b. SED con interruzione (Fig. 12.4)

Una operazione utente (read, erase, program) e' attivata su un certo blocco di piu' memorie in parallelo. Dopo un certo tempo si invia a tutte le memorie un comando specifico (interruzione). La memoria che ha completato con successo l'operazione ha BBF = 0 le altre hanno BBF = 1.

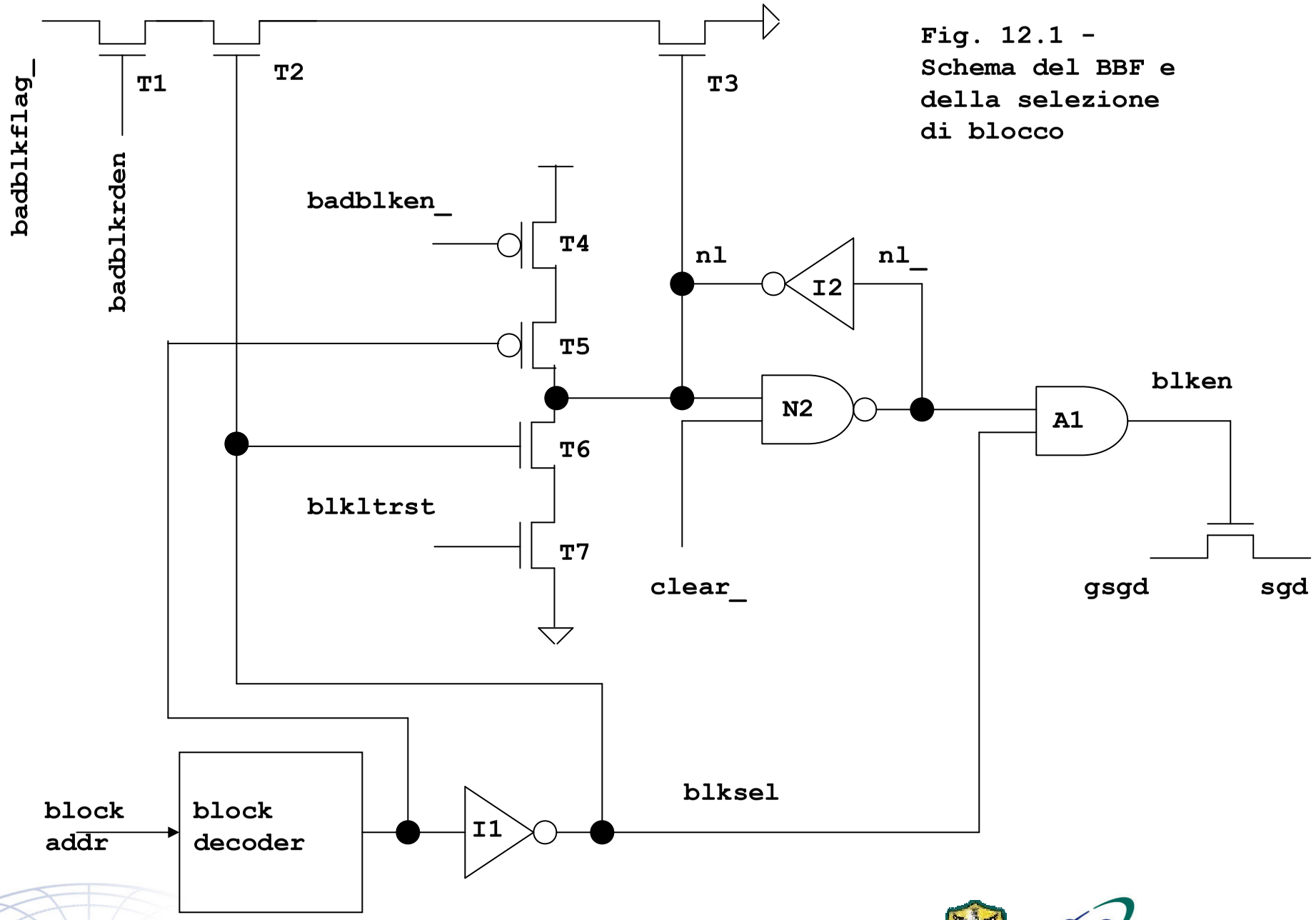
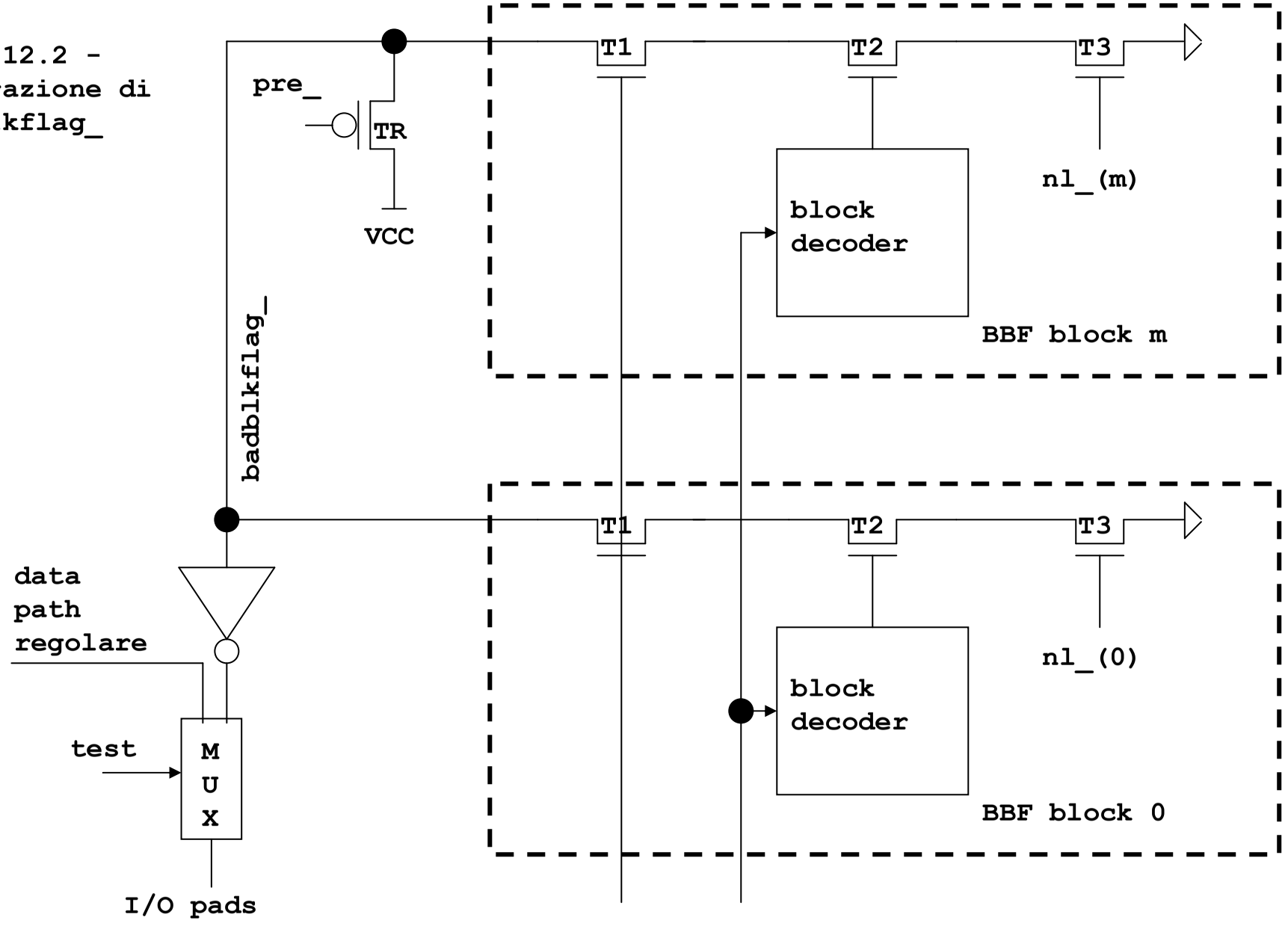


Fig. 12.1 - Schema del BBF e della selezione di blocco

Fig. 12.2 - generazione di badblkflag_



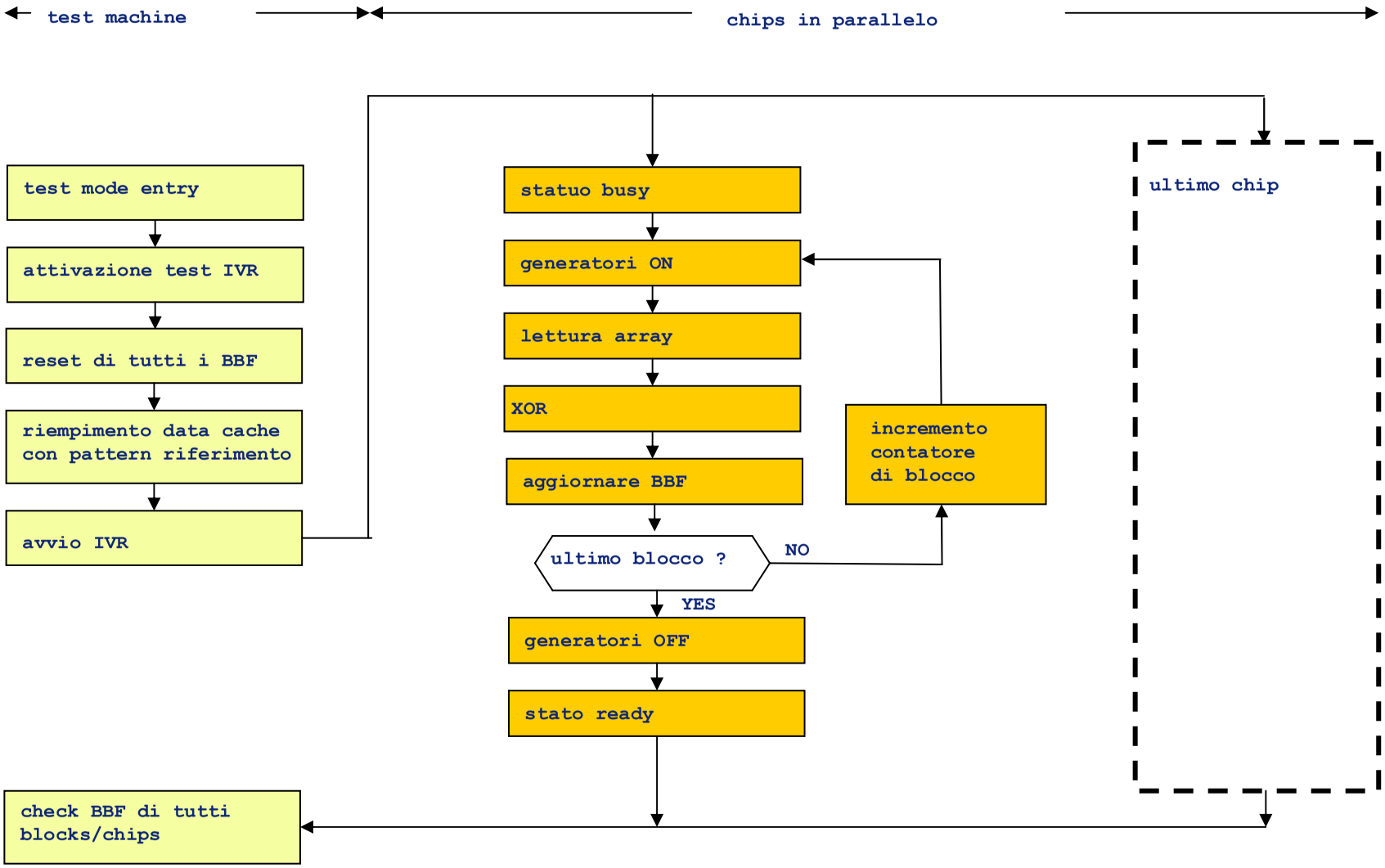


Fig. 12.3 - Flusso operativo di SED IVR

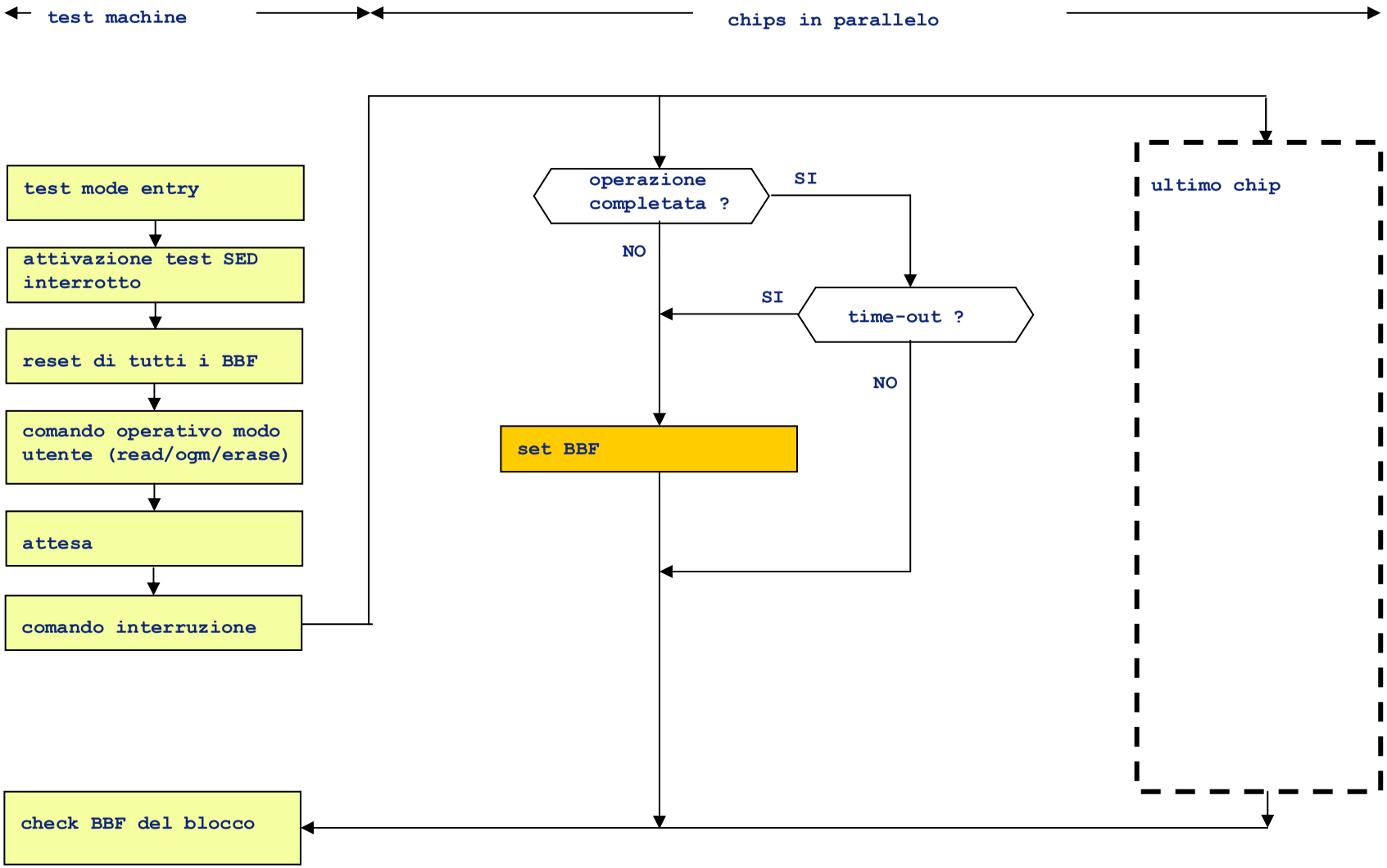


Fig. 12.4 - Flusso operativo di SED interrotto

13) MONITOR DI ALGORITMI (Fig. 13.1)

Un algoritmo e' composto di numerosi passi e diramazioni che vengono eseguiti in modo adattativo in base ai risultati ottenuti. Non e' quindi possibile prevedere a priori quale step dell' algoritmo e' in esecuzione ad un certo istante.

A scopo di debug e' molto utile disporre di un test mode per effettuare il monitor dell'esecuzione degli algoritmi.

Un esempio e' organizzato in questo modo :
. forzamento esterno del clock di sincronizzazione (tck)
. disponibilita' ai pads dell'indirizzo di ROM (tra)
. disponibilita' ai pads dei segnali di accesso ai registri (twsm)

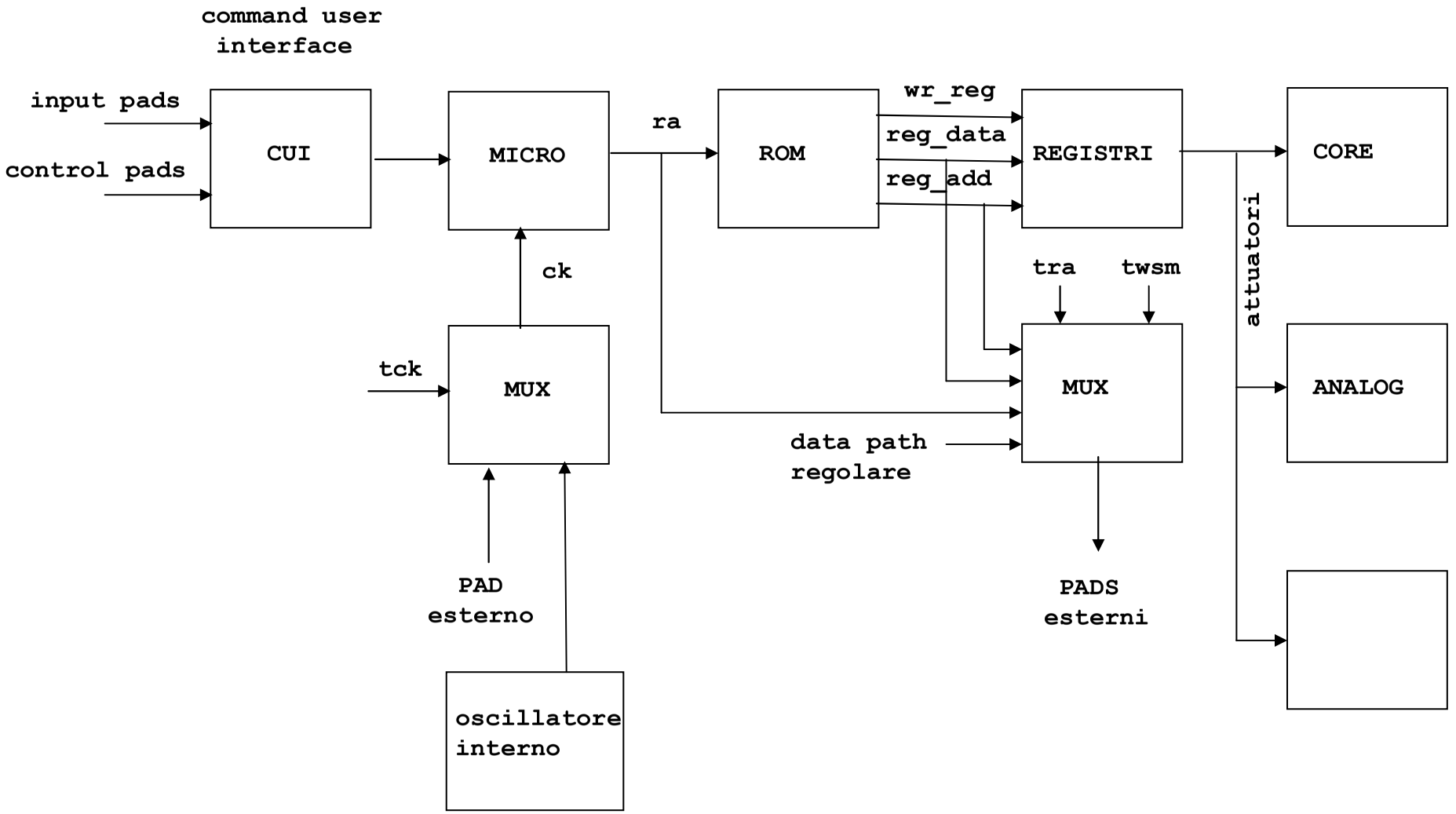


Fig. 13.1 - Shema a blocchi relativo al monitor di algoritmi e forzamento di clock