

# Layer Full Custom

17 marzo 2010

## Sommario

Discuteremo in queste pagine le caratteristiche del Layer full custom da utilizzare nel prototipo *Amchip04*. Saranno presentate le schematiche ed le temporizzazioni necessarie ad un corretto funzionamento.

Il layer è principamnte composto da tre elementi: le celle CAM, di tipo Nand o Nor, e l'amplificatore del segnale di match in uscita.

L'utilizzo di due diversi tipi di CAM è necessario al fine di ottenere notevoli risparmi energetici<sup>1</sup>.

## 1 Celle CAM

Nella cella di tipo Nand, quando avviene un match tra il dato presentato sulle SearchLine ( $SL_{in}$ )<sup>2</sup> coincide con quello memorizzato (D), il valore presente in  $ML_{in}$  viene trasmesso su  $ML_{out}$ .

Se ad esempio il dato in  $D$  è "1" l' n-mos (cerchiato in giallo in Fig.1) è acceso ed il gate del p-mos di valutazione (cerchiato in rosso) viene ad essere pilotato dal  $SL_{N_{in}}$ . Qualora quindi il valore sulla SL sia anch'esso alto (match) il p-mos si accende propagando il valore di  $ML_{in}$ .

Per quanto riguarda le celle Nor (Fig.2) quando si ha un mismatch la linea  $ML$  viene portata a massa. Il segnale  $PRE$  è infatti connesso direttamente a terra. In caso contrario rimane disconnessa e libera di mantenere il valore impostogli esternamente.

---

<sup>1</sup>K.Pagiamtzis e A.Sheikholeslami "Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey" 2006. IEEE journal of solid-state circuits, V.41 n.3.

<sup>2</sup>Le linee  $SL_{in}$  sono i bus in ingresso al pattern bank. Nel seguito saranno anche chiamate *Bit\_line*.

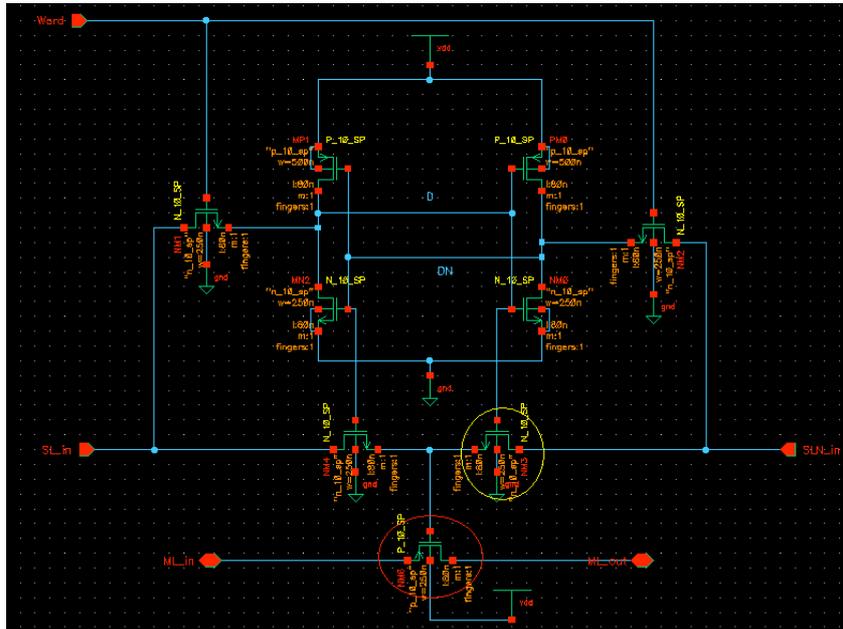


Figura 1: Schematico della cella tipo Nand

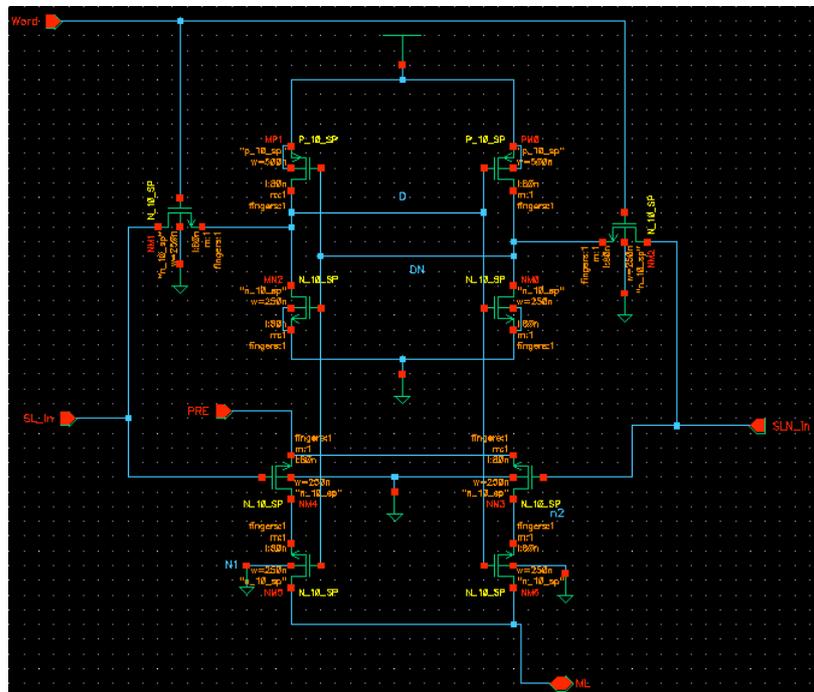


Figura 2: Schematico della cella tipo Nor

## 2 Il Layer

I due tipi di celle CAM sono utilizzate assieme per costituire il layer, presentato sinteticamente in figura 3.

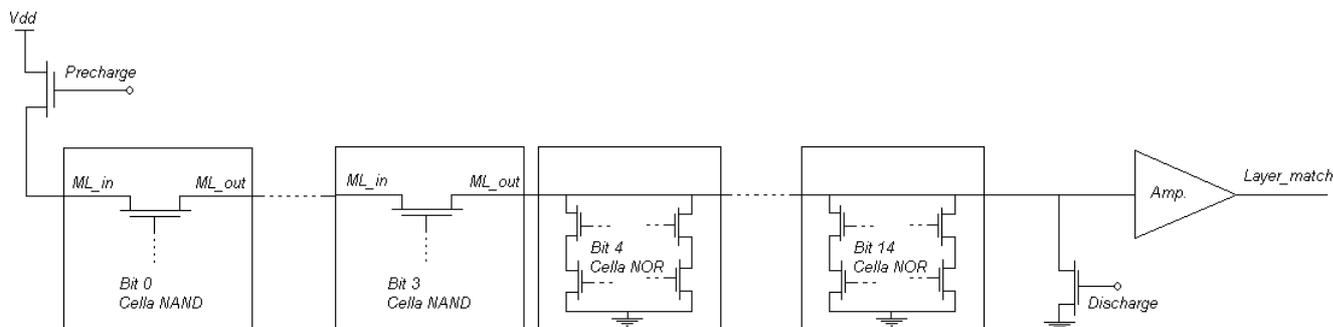


Figura 3: Rappresentazione sintetica del layer

La fase di valutazione del dato può essere schematizzata in tre operazioni:

- Valutazione del cambiamento del dato sui bus in ingresso
- Precharge della *ML*
- Valutazione e registrazione nel registro di ingresso della majority

Nella prima fase la logica a monte del pattern bank (in particolare nel Filter) controlla ad ogni ciclo di clock se i dati in ingresso al chip sono variati, e solo in caso affermativo attiva i processi successivi.

Questa operazione, che non coinvolge il layer, fornisce un primo contributo al risparmio energetico.

Nella seconda fase si attiva il transistor pilotato dal segnale *precharge* che inizia a caricare la match line. Il caricamento della sezione a valle delle celle Nand avviene solo se su queste è avvenuto un match.

Il caricamento completo della linea avviene solo per una frazione ridotta dei layer, quelli per cui tutte le Nand hanno un match. Scegliendo di porre tali celle sui bit meno significativi si aumenta la randomicità dei dati. Con questa ipotesi, poichè si sono utilizzate quattro celle, il numero di ML da caricare è ridotto di un fattore 16.

In questa fase è di fondamentale importanza che le *bit\_line* che pilotano le celle Nor siano scariche, assicurandosi che la ML a valle delle Nand sia disconnessa da terra. In caso contrario la *match\_line* potrebbe non caricarsi in quanto una delle Nor potrebbe metterla a terra. Per ovviare a questo è stata introdotta una modifica allo schema generale del pattern

bank, che prevede l'uso di 3-state per disconnettere le bit-line e di una serie di transistor per scaricarle (Fig.4).

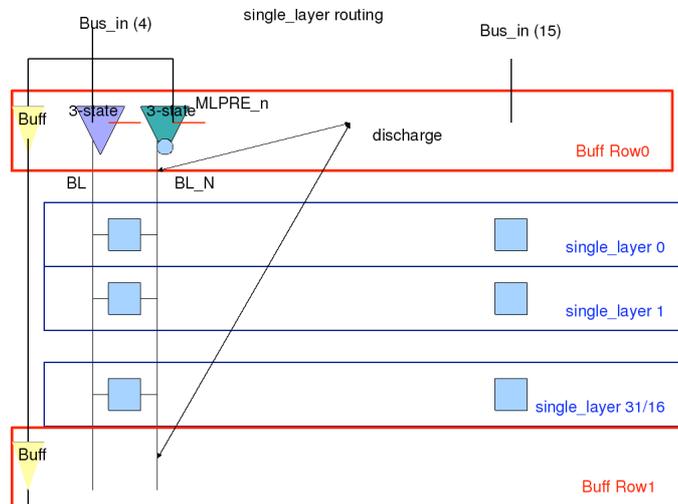


Figura 4: Schema del routing per i bus 4..14.

Nell'ultima fase il transistor di precharge viene aperto ed il dato dei bit 4..14 presentato nuovamente alle celle Nor. In caso di match di tutti i bit la ML rimane alta, registrando un hit, altrimenti viene scaricata dalle nor su cui non è avvenuto il match.

L'aggiunta del transistor di discharge presente in figura 3 è necessaria poichè una volta avvenuto il match, la ML rimane carica e al ciclo successivo non avrebbe importanza che le celle Nand abbiano o meno un match. Considerando che una volta registrato l'hit non ha più importanza avere il corretto funzionamento del layer fino all'*init\_event* successivo, si è scelto, soprattutto per ragioni di timing, di effettuare il discharge solo all'inizio del successivo evento.

### 3 Temporizzazione e consumi

Sfruttando entrambi i fronti del clock e seguendo la temporizzazione in Fig.5 (relativa alle celle Nor) siamo in grado di eseguire un confronto ad ogni ciclo di clock, cosa ben più complessa se vi fosse state un'ulteriore fase di discharge (come quella in figura). Nello schema sono riportati tutti i tempi critici su cui è necessario porre particolare attenzione per un corretto funzionamento del layer.

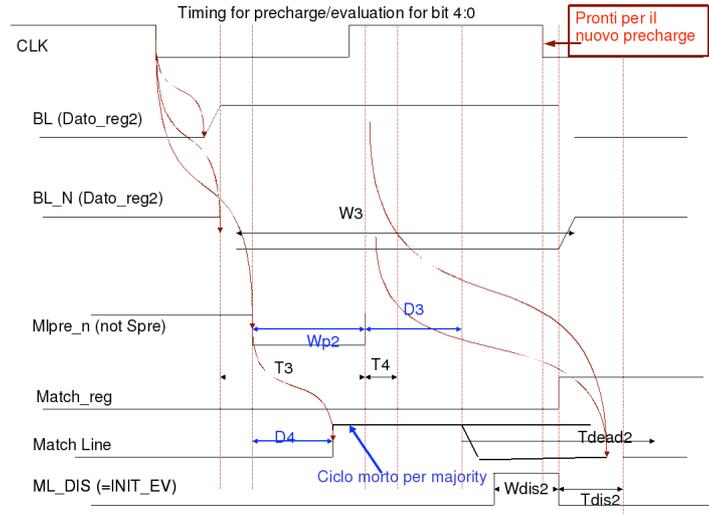


Figura 5: Timing per la valutazione. La fase di controllo sul dato in ingresso è svolta nel semiperiodo precedente a quello rappresentato.

Parametro	Valore(ns)	Parametro	Valore(ns)
T1	0.1	W1	1.7
T2	0.1	W2	0.25
T3	0.1	W3	1.5
T4	0	D1	0.1
Wp1	1.5	D2	0.2
$T_{dead}$	>1000	D3	0.2
$T_{dis}$	0.08		

Tabella 1: Timing ottenuti con simulazione delle schematico.

I consumi medi relativi al precharge sono invece riportati in Tab.2. Si assume piena randomicità sui quattro bit meno significativi.

Caricamento	Probabilità	Consumo singolo Layer( $\mu A$ )	Consumo pattren bank ( $mA$ )
completo	1/16	5.10	153
3 Nand	1/16	0.905	27.4
2 Nand	1/8	0.642	38.5
1 Nand	1/4	0.391	46.9
nessuno	1/2	0.143	34.3
<b>Totale</b>			<b>300</b>

Tabella 2: Consumi del precharge da simulazione schematico con parassiti estratti