

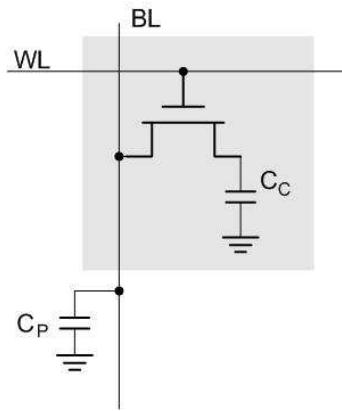
# Appunti V settimana

a cura di Andrea Di Giovanni

## Ram dinamica

A differenza delle ram statiche una cella di ram dinamica richiede solo un transistor e una capacità invece di 6 transistor.

Nelle DRAM il livello logico viene memorizzato su una capacità collegata ad un transistor che svolge la semplice funzione di interruttore. In questa maniera le dimensioni di una cella dinamica sono circa  $\frac{1}{4}$  della cella statica.



La costruzione delle celle di memoria dinamica segue un processo diverso dal processo analizzato in precedenza per la tecnologia CMOS, in quanto è necessario aumentare la capacità  $C_c$  senza aumentare l'area della cella. La capacità è quindi realizzata tramite una struttura verticale, scavando nel silicio una buca le cui pareti sono rivestite di uno strato di dielettrico e che poi viene riempita di polisilicio o metallo.

## Scrittura

L'operazione di scrittura dal punto di vista elettrico è molto semplice in quanto si attiva la word line (WL) relativa alla cella d'interesse e poi si pilota la bit line (BL) al livello logico desiderato, che sarà memorizzato nella capacità.

## Letture

L'operazione di lettura, a differenza della scrittura, è un'operazione critica in quanto "distrugge" l'informazione memorizzata nella capacità e pertanto necessita che il dato venga riscritto dopo che è stato letto (REFRESH).

Si procede attivando la WL e si "legge" il livello di tensione sulla BL. Bisogna fare attenzione perché la variazione di tensione sarà minima in quanto  $C_c$  "vede" la capacità parassita  $C_p$  della BL che è maggiore di un paio di ordini di grandezza.

Considerando  $C_c$  e  $C_p$ , al termine dei transistori il livello di tensione finale sulla BL si ottiene eguagliando la carica iniziale e la carica finale del sistema, ovvero

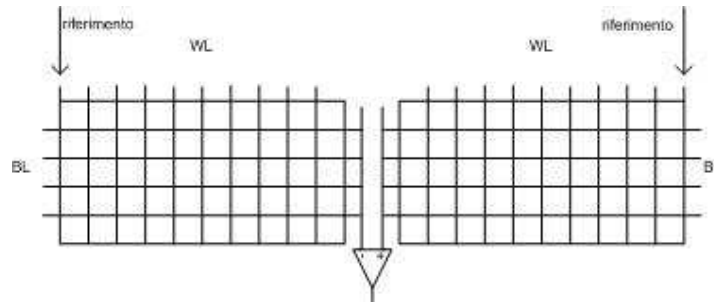
$$Q = C_c \cdot V_i = (C_c + C_p) \cdot V_f \Rightarrow V_f = \frac{C_c \cdot V_i}{(C_c + C_p)}$$

dove  $V_i = GND$  se scarica,  $V_i = V_{dd} - V_{TN}$  se carica.

Il guadagno  $\frac{C_c}{(C_c + C_p)}$  è ovviamente molto basso, considerando che  $C_c \gg C_p$ . Quindi è

necessario l'utilizzo di un sense amplifier differenziale. Il problema è realizzare una tensione di riferimento molto precisa con cui paragonare la tensione della cella.

Si usa quindi una riga di celle in cui viene "memorizzata" una tensione di riferimento intermedia tra  $V_{dd}$  e  $GND$ . Si divide la matrice di memoria in due parti, in ognuna delle quali c'è una riga di riferimento (*dummy cells*) che non viene utilizzata per la memorizzazione dei dati. In questa maniera ogni volta che viene selezionata una riga da leggere si utilizzerà la riga di riferimento del blocco opposto.



Prima di attivare la WL della cella da leggere verranno precaricate a  $V_{DD}/2$  sia la BL della cella da leggere che la BL della riga di riferimento speculare.

Successivamente vengono attivate le WL della riga da leggere e la WL della riga di riferimento, il che implica che attraverso le capacità parassite tra gate – source e gate – drain dei transistor delle due celle viene iniettata carica sulla capacità da leggere e di riferimento. Approssimativamente l'incremento di tensione su entrambe le BL sarà uguale, quindi questa piccola variazione non ha effetti negativi sull'amplificatore differenziale di uscita, ma semplicemente si compensa. A questo punto la tensione sulla BL di riferimento si mantiene costante mentre la tensione sulla BL da leggere varierà in aumento o in diminuzione (per effetto della distribuzione di cariche tra la  $C_c$  e  $C_p$ ) tanto quanto basta per far discriminare il valore al sense amplifier.

L'operazione di lettura ovviamente modifica la carica della  $C_c$  e quindi bisogna memorizzare il livello letto per poi riscrivere tale valore sulla cella appena letta. Questa operazione, detta refresh, è anche necessaria periodicamente per annullare gli effetti di scarica di  $C_c$  dovuti alle correnti parassite.

Nonostante tutti gli accorgimenti del caso, errori in lettura possono essere comuni per via di degradazioni della cella o a causa di eventi esterni, quindi i dati sono spesso protetti usando meccanismi come parità o Codici a Ridondanza Ciclica. Inoltre in fase di collaudo si può scoprire che alcune celle non funzionano. Per evitare di dover scartare l'intero chip, si aggiungono alcune righe ridondanti e dopo avere effettuato il collaudo si procede alla programmazione dei decoder e dei multiplexer per usare una riga ridondante invece di ogni riga che contiene una cella guasta.

## ***Analisi dei ritardi nella logica combinatoria***

### **Modello di ritardo di una porta logica**

Nella logica combinatoria ogni porta logica introduce un ritardo di propagazione e quindi il ritardo complessivo di una rete combinatoria si ottiene come somma dei ritardi di propagazione sul cammino più lungo. Il ritardo di propagazione di una porta logica è indice di quanto velocemente risponde (e quindi di quanto velocemente varia l'uscita) ad un cambiamento della configurazione dell'ingresso. E' misurato a partire dall'istante in cui l'ingresso raggiunge il 50% della sua escursione logica fino all'istante in cui il segnale d'uscita raggiunge il 50% della sua escursione.

In generale una porta logica può rispondere con tempi di propagazione differenti a seconda del tipo di commutazione dell'uscita, ragione per cui è necessario definire un ritardo di propagazione basso – alto  $t_{pLH}$  e un ritardo di propagazione alto – basso  $t_{pHL}$ . Il ritardo di propagazione globale è dunque definito per convenzione come

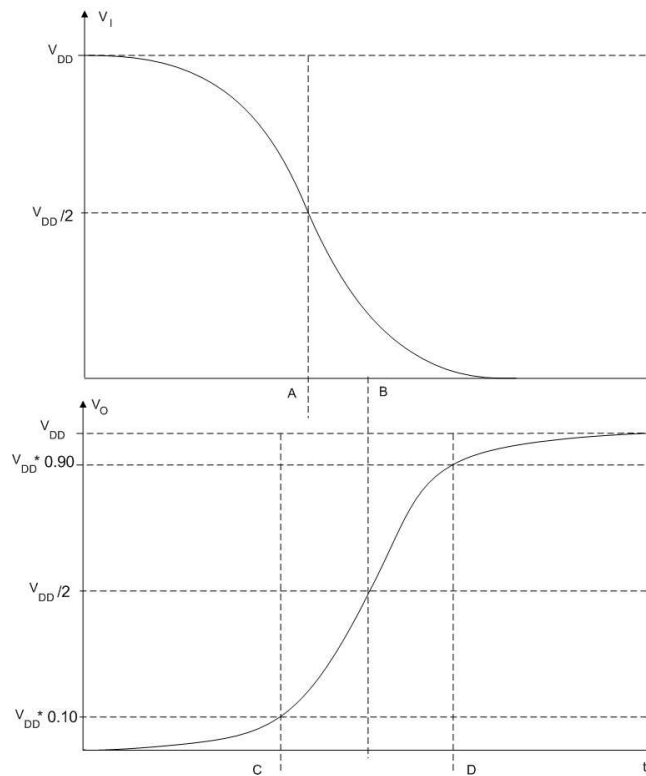
$$t_p = \frac{(t_{pLH} + t_{pHL})}{2}$$

Le porte CMOS statiche vengono generalmente progettate in modo che nel caso peggiore  $t_{pLH} = t_{pHL}$ .

Il tempo di transizione è classificato anche in questo caso in  $t_{tLH}$  (tempo di salita) e  $t_{tHL}$  (tempo di discesa) e si misura dall'istante in cui il segnale di uscita è al 10% della sua escursione sino al momento in cui è al 90% della sua escursione.

I tempi di transizione influenzano i tempi di propagazione: quanto maggiore è  $t_t$ , tanto maggiore sarà  $t_p$ .

I grafici illustrano queste definizioni per la transizione basso alto. Il  $t_{pLH}$  è pari a B-A, mentre il  $t_{tLH}$  è pari a D-C.

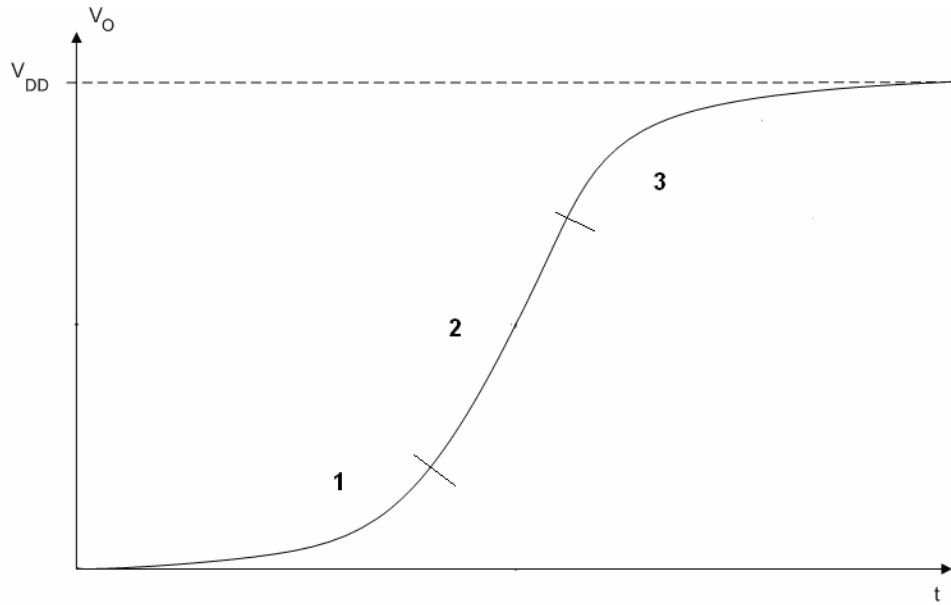


Oltre alla velocità di commutazione, il tempo di propagazione è in funzione de:

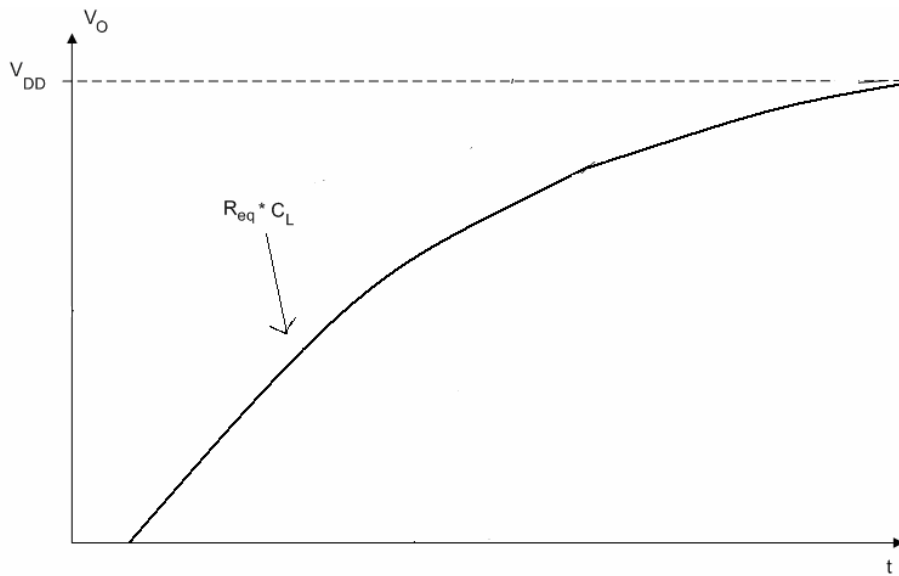
- l'abilità di pilotaggio in corrente della porta
- le capacità parassite interne
- le capacità di carico
- le capacità e resistenze parassite delle interconnessioni

L'abilità di pilotaggio in corrente della porta si quantifica attraverso una  $R_{eq}$  che modella in modo semplificato l'andamento reale del segnale di uscita. Il modello  $R_{eq} \cdot C_L$  è scelto in maniera tale da far coincidere  $t_{LH}$  e  $t_{HL}$  con il modello reale.

Quindi nel caso analizzato precedentemente di transizione dal basso all'alto l'andamento reale sarà

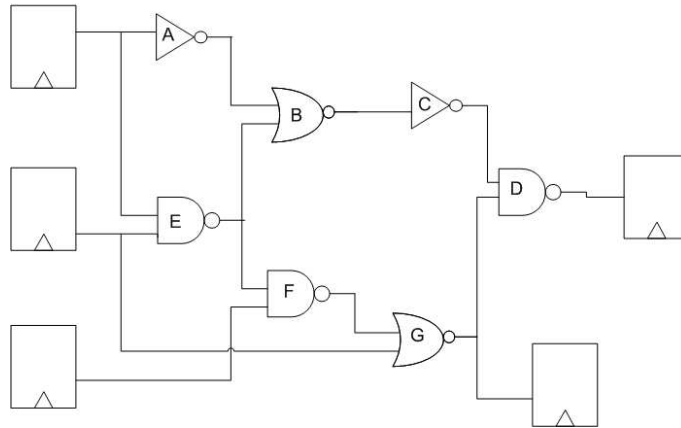


Nella parte 1 della curva il pull-up e' saturo e ha  $V_{gs}$  decrescente, nella parte 2 il pull-up e' saturo con  $V_{gs}$  costante e uguale a  $-V_{DD}$ , nella parte 3 il pull-up passa in regime lineare. Mentre l'andamento secondo il modello semplificato sar  un esponenziale con costante di tempo  $R_{eq} C_L$ :



## Analisi del ritardo di un circuito

Supponiamo di avere il seguente circuito a clock singolo



In un circuito che contiene componenti sequenziali e combinatori bisogna rispettare il seguente limite inferiore per la durata del ciclo di clock

$$T > t_{p_{CK \rightarrow Q}} + t_{p_{LC}} + t_{su}$$

dove

$$t_{p_{LC}} = \max_{\text{CAMMINI}} \left( \sum_{\text{CAMMINO}} t_{p_i} \right)$$

dove con  $t_{p_i}$  si indica il tempo di propagazione del cammino i-esimo.

Vediamo quanto vale il tempo di clock minimo  $T_{\min}$  affinché il circuito possa funzionare correttamente, considerando per esempio i seguenti ritardi per gli elementi del circuito:

INVERTER	0.1
NAND2	0.2
NOR2	0.3

$$t_{p_{CK \rightarrow Q}} = 0.1, t_{su} = 0.2$$

In base alle equazioni viste sopra, si tratta di trovare il cammino più lungo nel circuito.

Nell'esempio trattato si può facilmente analizzare tutti i cammini e scoprire che  $t_{p_{LC}} = 0.9$  in quanto gli ingressi al NOR2 B si stabilizzano a  $t = 0.1$  e  $t = 0.2$  e la sua uscita si stabilizza dunque a  $t = 0.5$ , mentre l'inverter C stabilizzerà l'ingresso del NAND2 D a  $t = 0.6$ . Il secondo ingresso del NAND2 D si stabilizzerà a  $t = 0.2 + 0.2 + 0.2 + 0.3$  (NAND2 E, NAND2 F, NOR2 G) dunque bisognerà aspettare quest'ultimo segnale prima considerare stabilizzato l'ingresso del flip flop. Il NAND2 D impiega 0.2 unità per stabilizzare l'uscita quindi  $t_{p_{LC}} = 0.7 + 0.2 = 0.9$ .

In circuiti molto complessi, per ovvie ragioni, non è possibile effettuare una tale analisi, per cui è necessario utilizzare un algoritmo di programmazione dinamica basato sull'ordinamento topologico, che ordina i nodi di un grafo diretto in maniera tale che tutti i predecessori di un determinato nodo siano incontrati prima di quel nodo.

Per ogni nodo in ordine topologico l'istante temporale in cui si stabilizza il segnale, anche detto tempo di arrivo  $t_a$ , può essere calcolato sapendo il tempo di arrivo degli ingressi e il tempo di propagazione della porta stessa. Quindi per ogni porta il tempo di arrivo  $t_a$  si può determinare come

$$t_a = \max_{\text{INGRESSI}}(t_{a_i}) + t_p$$

Questo algoritmo è molto rapido da eseguire anche per circuiti molto grandi, in quanto deve analizzare ogni porta una volta sola, grazie all'ordinamento topologico.

In realtà purtroppo il modello usato, basato su un ritardo di propagazione fisso, è molto approssimato. Quindi per migliorare la precisione si simula il funzionamento di un adeguato numero di percorsi critici o quasi critici, in modo da valutare più esattamente a quale frequenza il circuito può funzionare.

Inoltre, siccome non è possibile prevedere con certezza le caratteristiche esatte dei dispositivi sul silicio, si progettano i circuiti con un certo margine di sicurezza sul tempo di clock riducendo la frequenza operativa rispetto al valore simulato.

### **Cammini critici, quasi critici e slack**

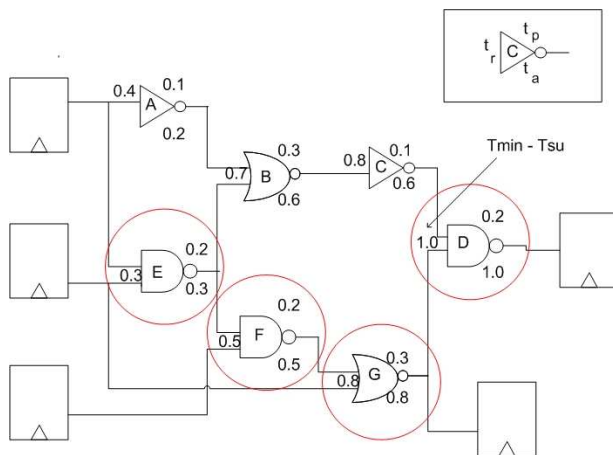
Un modo per valutare quali cammini siano "quasi critici", e quindi possano determinare il ciclo di clock se sono più lenti del previsto, si basa sul  $t_r$ , che è definito come l'istante temporale a cui è necessario che l'uscita di una porta logica sia stabile per non aumentare il tempo di clock.

Supponendo che l'uscita della porta  $(i-1)$ -esima sia connessa con l'ingresso della porta  $i$ -esima, il  $t_r$  della porta  $(i-1)$ -esima si ottiene considerando il  $t_r$  della porta  $i$ -esima meno il  $t_p$  della porta  $i$ -esima stessa (o il  $t_{su}$  nel caso di un FF), quindi  $t_{r_{i-1}} = t_{r_i} - t_{p_i}$ .

Nel caso in cui una porta abbia un  $t_a = t_r$ , quella porta è parte di un cammino critico. Il time slack  $t_s = t_r - t_a$  misura quanto la porta è distante da diventare critica.

Questa analisi trova i cammini critici in un tempo approssimativamente lineare in funzione del numero di porte logiche.

Considerando nuovamente l'esempio precedente si avrà:



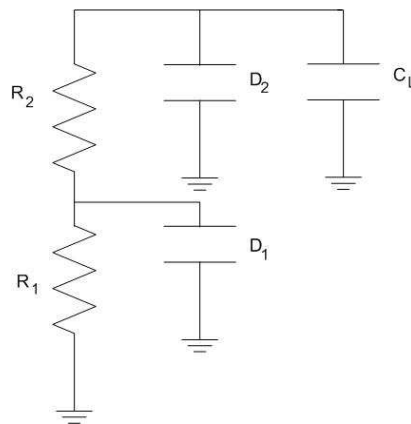
dove le porte logiche sul percorso critico sono evidenziate da un cerchio rosso.

## Ottimizzazione dei ritardi con il metodo del logical effort

Una volta trovati i percorsi critici bisogna passare all'ottimizzazione che può avvenire mediante modifica della funzione logica e/o tramite modifica delle dimensioni dei transistor. In questo corso ci occupiamo solo del secondo aspetto, mentre il primo è trattato nei corsi di progettazione logica.

Attraverso la modifica delle dimensioni dei transistor si cerca di ridurre la  $R_{eq}$  dei transistor ovvero di aumentare la loro "abilità" di pilotaggio in corrente senza aumentare troppo la capacità di carico delle porte precedenti. La tecnologia CMOS permette di effettuare delle valutazioni di dimensionamento in maniera semplice usando il modello di Elmore.

Supponendo di avere un NAND-2 e che la larghezza degli NMOS sia  $W$ , la dimensione dei PMOS che permette di avere tempi di salita e discesa uguali nel caso peggiore sarà  $\beta W$ , dove il rapporto tra le mobilità di portatori di carica è  $\beta = \frac{\mu_-}{\mu_+}$  (pari a circa 2 nelle tecnologie attuali). Per il computo del  $t_p$  consideriamo la transizione in discesa dell'uscita (2 NMOS in serie):



Usando il modello di Elmore si avrà che il tempo di propagazione è

$$t_p = R_1 \cdot (D_1 + D_2 + C_L) + R_2 \cdot (D_2 + C_L) = R_1 \cdot (D_1 + D_2) + R_2 \cdot D_2 + (R_1 + R_2) \cdot C_L$$

in cui si identifica una componente intrinseca, dovuta alle capacità parassite interne (primi due addendi), e una componente utile (ultimo addendo) per far cambiare di livello la capacità di carico.

La componente intrinseca del ritardo (che rappresenta uno spreco, in quanto la porta esegue solo la carica/scarica delle capacità parassite) in prima approssimazione è indipendente dalla larghezza  $W$  del transistor, in quanto resistenze e capacità sono rispettivamente inversamente e direttamente proporzionali alla larghezza dei transistor.

La componente utile è invece influenzata dalla larghezza del transistor e avendo una  $W$  maggiore il ritardo della porta in questione diminuisce, mentre quello della porta precedente aumenta. Il problema quindi è fare in modo da aumentare la larghezza dei transistor per cui l'effetto sulla porta precedente è minore e quello sulla porta in questione è maggiore. Si può dimostrare

analiticamente che un semplice calcolo permette di ottimizzare in modo esatto il ritardo, distribuendo il massimo rapporto tra capacità di ingresso e di carico alle porte che sono meglio in grado di sostenerlo.

Indicando con  $p$  la componente intrinseca e con  $G$  il guadagno capacitivo  $C_L/C_{in}$  si può riscrivere la relazione precedente come

$$t_p = p + (R_1 + R_2) \cdot C_{in} \cdot \frac{C_L}{C_{in}} = p + L \cdot G$$

dove con  $L$  si indica il logical effort della porta logica in questione.

Riassumendo, il ritardo di propagazione di una porta logica è composto da:

- $p$ : ritardo intrinseco (in prima approssimazione dipende solo dal tipo di porta logica e non da  $W$ )
- $L$ : logical effort (di nuovo dipende unicamente dal tipo di porta logica presa in considerazione)
- $G$ : guadagno di capacità, dipendente unicamente dal rapporto tra capacità di ingresso e capacità di carico della porta (quindi dipende dalla  $W$  dei transistor utilizzati, ma non dalla funzione logica).

## Logical Effort

Il logical effort è un'indice dell'abilità di pilotaggio in corrente di una porta logica. Questo significa che porte con un logical effort piuttosto grande (ovvero che eseguono un "pesante lavoro" logico) sono meno idonee a pilotare grandi carichi rispetto a porte logiche con un logical effort piccolo.

Il computo del logical effort di una porta può essere eseguito considerando direttamente le caratteristiche elettriche della porta, ovvero

$$L = R_T \cdot C_{in}$$

dove con  $R_T$  si indica la resistenza equivalente totale della porta logica durante la transizione dell'uscita analizzata e con  $C_{in}$  si indica la capacità equivalente della porta logica vista dall'ingresso.

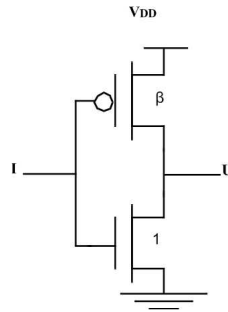
Il calcolo del logical effort di una porta può essere in parte svolto a priori, creando tabelle in cui sono indicati valori normalizzati per ogni porta rispetto all'inverter (che, per una data tecnologia, è la porta logica con logical effort minimo, e dunque ha un valore di logical effort normalizzato unitario) e moltiplicando questo valore per il logical effort non normalizzato (quindi assoluto) dell'inverter, in formule:

$$L = L_{norm} \cdot L_{inv}$$

## Computo del logical effort normalizzato

### *Inverter di dimensioni minime*

Consideriamo l'inverter di dimensioni minime (larghezza NMOS=1, larghezza PMOS= $\beta$ , in maniera tale che sia il più simmetrico possibile)

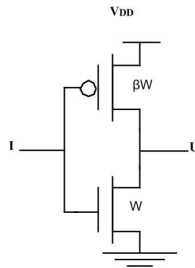


il logical effort non normalizzato sarà:

$$L_{inv_{min}} = L_0 = R_{eq_{NMOS}} \cdot (1 + \beta) \cdot C_{eq_{NMOS}} = R_0 \cdot (1 + \beta) \cdot C_0$$

### ***Inverter di dimensioni generiche***

Consideriamo l'inverter di generica larghezza W (NMOS W, PMOS  $\beta W$ ).



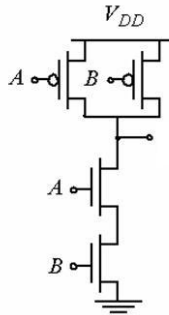
Ricordando che la resistenza equivalente di un transistor NMOS di dimensioni W è  $1/W$  volte la resistenza equivalente di un transistor NMOS a dimensioni minime e che la capacità equivalente invece è W volte, si ha che il logical effort di un inverter di dimensioni generiche sarà:

$$L = \frac{R_0}{W} \cdot (W + \beta W) C_0 = R_0 \cdot (1 + \beta) \cdot C_0$$

Come si era già visto, il logical effort è una caratteristica di una porta logica indipendente dalla larghezza dei transistor.

### ***NAND-2***

Si procede al computo del logical effort normalizzato del NAND a 2 ingressi rispetto ad un inverter con larghezza W.



Si procede dimensionando i transistor della rete di PD e di PU per ottenere la stessa  $R_{eq}$

Indicando con  $R_U$  la resistenza di un PMOS della porta logica considerata e con  $R_{U_{inv}}$  la resistenza equivalente del PU dell'inverter considerato, per il NAND2 consideriamo il caso peggiore ovvero in cui c'è solo un transistor acceso. Uguagliando la resistenza equivalente della rete di PU dell'inverter con la resistenza equivalente della rete di PU della porta nel caso peggiore si avrà:

$$R_U = R_{U_{inv}}$$

quindi la larghezza dei transistor della rete di PU deve essere  $\beta W$ .

Indicando con  $R_D$  la resistenza di un NMOS della porta logica considerata e con  $R_{D_{inv}}$  la resistenza equivalente della PDN dell'inverter considerato, anche per la rete di PD del NAND2 consideriamo il caso peggiore, ovvero in cui ci sono 2 transistor in serie accesi, dunque eguagliando si ottiene:

$$2R_D = R_{D_{inv}}$$

quindi la larghezza dei transistor della rete di PD del NAND2 deve essere pari a  $2W$ .

Per quanto riguarda la capacità di ingresso si avrà che

$$C_{in} = 2WC_0 + \beta WC_0 = (2 + \beta)WC_0$$

Considerando che le due reti sono state dimensionate in maniera tale che le resistenze equivalenti nel caso peggiore siano uguali, il logical effort sarà

$$L = (2 + \beta)R_0C_0$$

Il logical effort normalizzato si ottiene come

$$L_{norm} = \frac{L}{L_0} = \frac{(2 + \beta)}{(1 + \beta)}$$

### **NAND-3**

Calcoliamo il logical effort normalizzato del NAND a 3 ingressi rispetto ad un inverter con larghezza  $W$ .

Per la rete di PU come caso peggiore si considera il caso in cui c'è un solo transistor acceso, quindi:

$$R_U = R_{U_{inv}}$$

quindi la larghezza dei transistor della rete di PU deve essere pari a  $\beta W$ .

Per la rete di PD il caso peggiore sarà il caso in cui ci sono 3 transistor in serie accesi:

$$3R_D = R_{D_{inv}}$$

quindi la larghezza dei transistor della rete di PD deve essere pari a  $3W$ .

Per quanto riguarda la capacità di ingresso si avrà che

$$C_{in} = (3 + \beta)WC_0$$

Il logical effort della porta sarà

$$L = (3 + \beta)R_0C_0$$

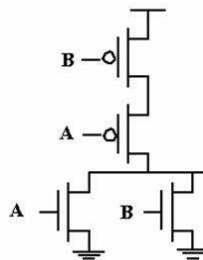
mentre il valore normalizzato sarà

$$L_{norm} = \frac{(3 + \beta)}{(1 + \beta)}$$

In generale, per una porta NAND a N ingressi  $L_{norm} = \frac{(1 + N + \beta)}{(1 + \beta)}$

### **NOR-2**

Seguendo l'approccio utilizzato negli esempi precedenti si passa al computo del logical effort normalizzato del NOR a 2 ingressi rispetto ad un inverter con larghezza W.



Per la rete di PU si avrà

$$2R_U = R_{U_{inv}}$$

quindi la larghezza dei transistor della rete di PU è pari a  $2\beta W$ .

Per la rete di PD invece

$$R_D = R_{D_{inv}}$$

di conseguenza la larghezza dei transistor della rete di PD è pari a W.

Per quanto riguarda la capacità di ingresso si avrà

$$C_{in} = (2\beta + 1)WC_0$$

e quindi il logical effort della porta sarà

$$L = (1 + 2\beta)R_0C_0$$

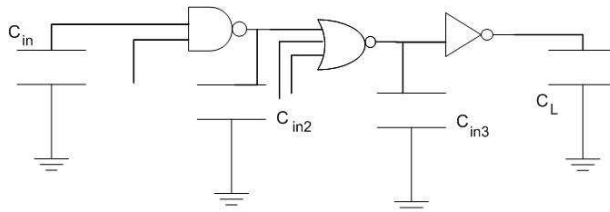
Il logical effort normalizzato sarà pari a

$$L_{norm} = \frac{(1 + 2\beta)}{(1 + \beta)}$$

In generale, per una porta NOR a N ingressi  $L_{norm} = \frac{(1 + N\beta)}{(1 + \beta)}$

### Dimensionamento dei transistor usando logical effort

Dato un circuito logico, per esempio:



e assegnate le capacità d'ingresso e di carico, si tratta di dimensionare i transistor di ogni porta per minimizzare la frequenza di clock.

Ricordando che per ogni porta logica il ritardo di propagazione è pari a  $t_p = p + L \cdot G$ , il problema, per le N porte che costituiscono il cammino critico di un circuito combinatorio, si riconduce alla minimizzazione della seguente funzione  $t_{tot} = \sum_{i=1}^N (p_i + L_i G_i)$  rispetto a  $G_i$  (dunque rispetto a  $W_i$ )

rispettando il vincolo  $\prod_{i=1}^N G_i = \frac{C_L}{C_{in}}$ . Questo si traduce in un problema che a livello di funzione di costo è lineare mentre a livello di vincolo è non lineare.

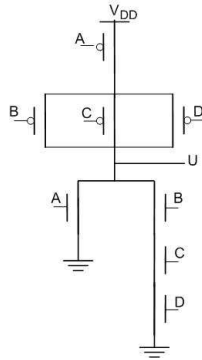
Una soluzione è attribuire un maggiore guadagno capacitivo intermedio (quindi un maggiore carico) alle porte che hanno un  $L_i$  minore (quindi maggiore abilità di pilotaggio in corrente, quindi minore resistenza, quindi maggiore velocità per caricare una capacità di carico). Si può dimostrare che il ritardo minimo si ottiene uguagliando tutti i prodotti  $L_i G_i$  in modo che

$$L_i G_i = \sqrt[N]{\prod_{i=1}^N L_i G}$$

## ESERCIZIO 1

Realizzare  $f = \overline{A + B \cdot C \cdot D}$  e dimensionare i transistor in maniera tale da avere la stessa  $R_{eq}$  di un inverter a dimensione minima.

La funzione richiesta è implementata dal seguente circuito



Per il pull-up, il caso peggiore è quando il transistor A e solo uno dei tre transistor B, C e D sono accesi in serie, dunque

$$2R_U = R_{U_{inv}}$$

il che implica che i transistor di PU devono essere larghi  $2\beta$ .

Per quanto riguarda il pull-down, si distinguono due casi:

1. i tre transistor B, C e D sono accesi in serie, dunque

$$3R_D = R_{D_{inv}}$$

il che comporta che i transistor B, C e D devono essere larghi 3 unità,

2. il solo transistor A e' acceso, e quindi esso deve avere larghezza unitaria.

## ESERCIZIO 2

Considerando il circuito logico della pagina precedente, si calcoli il ritardo di propagazione per ognuna delle tre porte logiche supponendo che i valori di  $p$  siano 0.03, 0.05, 0.03 rispettivamente per il NAND2, NOR3, INVERTER, che il logical effort non normalizzato dell'inverter sia pari a 0.01,  $C_{in} = 0.2$ ,  $C_L = 0.7$  e  $\beta = 2$ .

Ricordando che  $L = L_{norm} \cdot L_{inv}$ , possiamo utilizzare le formule precedentemente analizzate per il calcolo del logical effort non normalizzato di ogni porta. In particolare, siccome per il NAND2 si ha

che  $L_{norm} = \frac{(2 + \beta)}{(1 + \beta)} = \frac{4}{3}$ , L sarà pari a 0.013.

Per il NOR3 si ha che  $L_{norm} = \frac{(1+3\beta)}{(1+\beta)} = \frac{7}{3}$ , dunque  $L = 0.023$ .

Imponendo che  $L_i G_i = \sqrt[N]{\prod_{i=1}^N L_i G} = \sqrt[3]{0.01 \cdot 0.013 \cdot 0.023 \cdot \frac{0.7}{0.2}} = 0.022$  il guadagno capacitivo per il NAND2 sarà pari a 1.7, per il NOR3 sarà pari a 1 mentre per l'inverter sarà pari a 2.2.

Avendo i valori di p, L e G di ogni porta, si può passare al calcolo dei ritardi di propagazione per ognuna di esse, quindi per il NAND2 si avrà un ritardo pari a 0.05, per il NOR3 pari a 0.07 mentre per l'inverter il ritardo sarà 0.05.

### **Consumo di potenza**

La dissipazione di potenza nei circuiti integrati segue, in prima approssimazione, l'andamento della legge di Moore: se si considera una porta logica che deve pilotare altre porte logiche (il che si può idealizzare come un carica/scarica di una capacità) l'energia necessaria sarà proporzionale alla carica della capacità di carico, ovvero

$$E = Q \cdot V_{DD} = C_L \cdot V_{DD}^2$$

Per ottenere la potenza dobbiamo considerare l'energia divisa per il tempo in cui avviene la carica/scarica. Ipotizzando che avvenga in un ciclo di clock, la potenza si può definire come

$$P = C_L \cdot V_{DD}^2 \cdot f_{CK}$$

Nel caso in cui si considera una porta logica la cui uscita non cambia valore ad ogni ciclo di clock si deve considerare la probabilità p di commutazione dell'uscita in un ciclo di clock, per cui la potenza si ottiene come

$$P = C_L \cdot V_{DD}^2 \cdot f_{CK} \cdot p$$

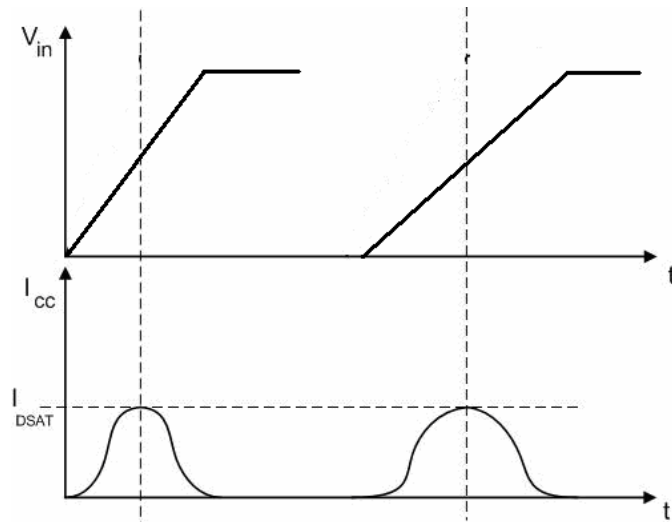
Nel caso in cui si vuole tenere conto degli effetti sulla potenza delle alee bisogna inserire nella formula precedente anche il numero medio di cambiamenti di valore in un ciclo di clock.

La potenza appena presentata è la potenza dinamica utile, cioè; quella usata per la carica/scarica della capacità di carico.

La potenza dissipata totale è composta dai seguenti elementi:

- potenza dinamica utile alla carica/scarica
- potenza dinamica non utile dovuta alle correnti di corto circuito
- potenza statica dovuta alle correnti di perdita.

Per quanto riguarda la potenza dinamica non utile si consideri il caso dell'inverter durante le fasi di transizione, mettendo a confronto due segnali d'ingresso con diverse pendenze (tempi di transizione differenti).



La corrente di corto circuito è dovuta al fatto che per un breve periodo durante la commutazione della porta logica esiste un cammino diretto tra alimentazione e massa. Nell'inverter ciò avviene quando entrambi i transistor sono in saturazione e formano un cammino conduttivo. Maggiore è il periodo della transizione maggiore sarà il tempo in cui è presente la corrente di corto circuito e quindi maggiore sarà la potenza dissipata inutilmente (maggiore area sottesa delle curve di corrente).

Per limitare gli effetti delle correnti di corto circuito si agisce riducendo il tempo di transizione quanto più possibile. Per le porte non critiche, cioè con un ampio slack, si può anche ridurre  $I_{Dsat}$ , che aumenta il ritardo.

## Potenza statica

La dissipazione dovuta alla corrente statica è funzione della tensione di soglia dei transistor utilizzati, poiché quanto più bassa è la soglia tanto più facile è superarla. Inoltre il consumo statico dipende dall'area dei transistor: nel caso di porte non critiche si può nuovamente agire su questo parametro rendendo i transistor più piccoli possibili.

Un altro fattore che influenza fortemente il consumo statico è la temperatura di funzionamento, che va tenuta il più possibile bassa. Questa soluzione ha un costo non indifferente, dovuto a package, ventilatore, etc.

## Minimizzazione della dissipazione di potenza

Ritornando alla potenza dinamica utile, si suppone che la  $f_{CK}$  non possa essere ridotta senza modificare anche l'architettura, in modo da mantenere le specifiche di prestazione globali.

I fattori da minimizzare sono principalmente  $C_L$  e  $V_{DD}$ . Nel caso di  $C_L$  la minimizzazione procede attraverso:

- diminuzione della lunghezza delle interconnessioni
- riduzione delle dimensioni delle porte non critiche
- utilizzo della logica dinamica.

Quest'ultimo accorgimento però, oltre a essere difficile da progettare, da un lato riduce le alee e la capacità d'ingresso, ma può aumentare il p a causa della fase di precarica.

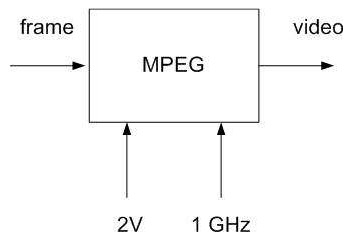
Per quanto riguarda invece la tensione di alimentazione, questa ha un effetto quadratico sulla potenza dinamica utile. Però una diminuzione della tensione ha come effetto una diminuzione (*approssimativamente* lineare) della  $f_{CK}$ , e quindi potrebbe portare a una violazione delle specifiche di prestazioni.

Inoltre riducendo  $V_{DD}$  si deve anche ridurre la tensione di soglia per mantenere la corrente di saturazione a livelli ragionevoli, ma questo implica un aumento del consumo statico.

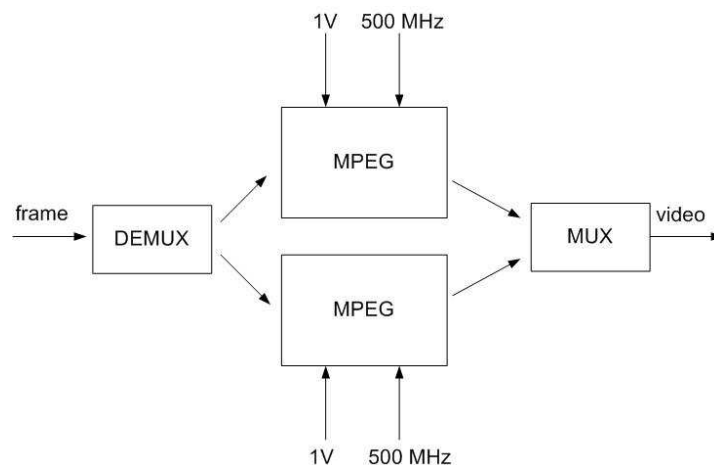
Per risolvere questi problemi si può agire a livello dell'architettura di sistema, differenziando le linee di alimentazione in base alle frequenze dei componenti del sistema, e tenendo presente che componenti a bassa frequenza possono essere alimentati con un basso livello di tensione e viceversa.

Per le componenti critiche, la cui frequenza di funzionamento non può essere facilmente ridotta, si possono mettere in parallelo più unità che funzionano a frequenza minore, come mostra il seguente esempio.

Supponiamo di dover progettare un decoder video usando una unità MPEG che funziona a 1GHz e che richiede una tensione di alimentazione di 2V:



Lo stesso decoder può essere progettato mettendo in parallelo due unità MPEG che funzionano a 500MHz ciascuna. Ognuna delle due unità si occuperà dell'elaborazione di una metà dei frame, e le immagini prodotte verranno rimesse insieme attraverso un multiplexer:



Considerando che la frequenza di clock dipende in modo approssimativamente lineare dall'alimentazione (se è sufficientemente lontana dalla soglia e si trascurano gli effetti del second'ordine), si può portare l'alimentazione ad 1V.

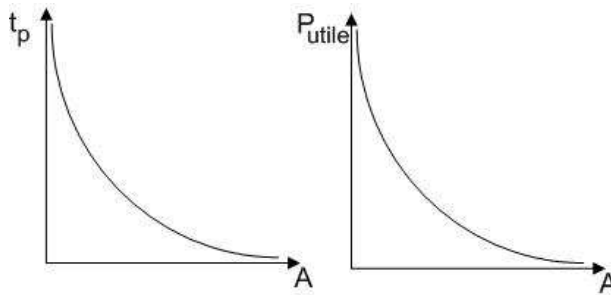
La soluzione riduce il consumo di potenza dinamica utile di un fattore 4, in quanto:

1. la dipendenza dalla tensione di alimentazione è quadratica, quindi un dimezzamento della tensione corrisponde a dividere per 4,
2. la dipendenza dalla frequenza di clock è lineare quindi il dimezzamento corrisponde a dividere per 2,
3. servono 2 unita' in parallelo, moltiplicando per 2 il risultato.

Per quanto riguarda la minimizzazione della probabilita' di transizione  $p$ , questa va gestita in fase di progetto logico di un circuito attraverso tecniche di clock gating e data gating.

Attraverso il clock gating viene inserito un enable sulle transizioni di clock in maniera tale da poter disabilitare all'occorrenza parti del circuito non utilizzate, mentre attraverso il data gating gli ingressi dei blocchi funzionali non utilizzati vengono disabilitati fissandoli a un valore costante.

Tutte le soluzioni analizzate hanno dei costi in termini di area del circuito, come evidenziato dai seguenti grafici area/ritardo e area/potenza



Questi grafici mostrano come in generale il ritardo di propagazione e la potenza utile diminuiscano con il crescere dell'area, poiche' (intuitivamente) quanto maggiore è la dimensione dei transistor tanto minore sarà il ritardo di propagazione, e quanto maggiore è il parallelismo tanto minore sarà la potenza utile dissipata. E' da notare come la dipendenza del ritardo dall'area puo' essere giustificata anche con considerazioni di tipo logico, in quanto realizzazioni con meno livelli logici, e quindi piu' veloci, in generale richiedono piu' area.