

Logica dinamica

La logica statica utilizza i transistor PMOS, che sono piu' grossi degli NMOS a parita' di corrente fornita: si cercano metodi alternativi per la riduzione del numero e delle dimensioni.

All'interno delle porte logiche statiche i transistor PMOS occupano circa β volte (circa 2-3 in tecnologie moderne) l'area occupata da un transistor NMOS.

Le 3 principali alternative alla logica statica sono:

- Logica dinamica
- DCVSL (Dual-Cascode Voltage Switch Logic)
- Logica a pass transistor (o logica ad interruttori)

Nella logica pseudo-n-mos, che adesso non e' piu' in uso se non in casi particolari a causa del forte consumo stataico di corrente e della forte sensibilita' alle variazioni di processo nei parametri dei transistor, non viene creata la rete di pull-up. Si inserisce solamente un transistor PMOS piccolo ($W=1$, $L>1$), che si comporta come la "resistenza di pull-up" delle vecchie porte open-drain.

Questo tipo di realizzazione ha alcuni problemi:

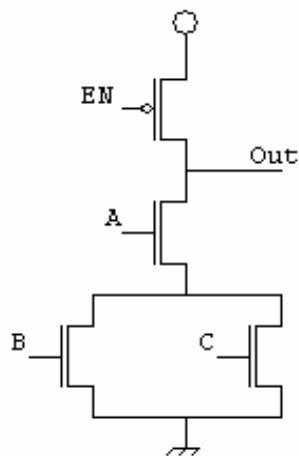
- consumo di potenza statica se l'uscita e' a livello basso;
- lento in salita, poiche' il pull-up deve essere piccolo per "perdere" rispetto ai pull-down;
- ridotto margine di rumore allo 0, poiche' il livello logico dipende dal rapporto tra le dimensioni di pull-up e pull-down.

Per ovviare al primo problema si possono adottare diverse soluzioni tra cui:

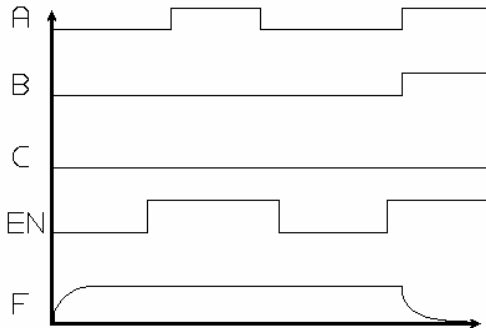
1° accendere il PMOS solo quando serve (DCVSL)

2° accendere il PMOS solo quando la rete di pull-down è spenta (logica dinamica).

Nel seguito considereremo solo la seconda, rappresentata (per $F = (A (B + C))'$) nella figura seguente, in cui EN vale 0 in fase di precarica (PMOS acceso) e 1 in fase di valutazione (PMOS spento).



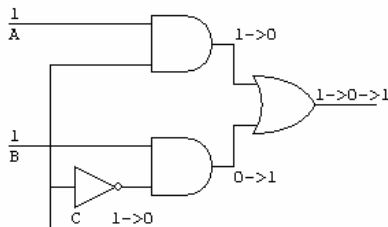
Se la rete di pull-down viene attivata in fase di valutazione, l'uscita viene portata a zero. Altrimenti, il valore logico uno rimane memorizzato (per un certo tempo) sulla capacita' parassita di uscita, in quanto il PMOS e' spento (EN vale 1) e i pull-down non conducono.



Questo tipo di logica deve soddisfare due vincoli principali per funzionare correttamente:

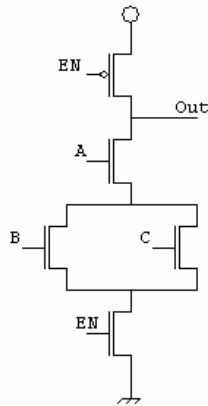
- tutti gli ingressi devono essere inattivi (a 0 con degli NMOS) in fase di precarica, per permettere al PMOS di precaricare la capacita' di uscita.
- gli ingressi non devono presentare alee (glitch) in fase di valutazione, perche' se gli NMOS vengono temporaneamente attivati, scaricano la capacita' di uscita. Nella logica dinamica il pull-up non e' in grado di riportare l'uscita a 1, perche' e' inattivo. Il risultato puo' essere un valore logico degradato o sbagliato.

Alee possono avvenire nel caso in cui gli ingressi siano pilotati da logica combinatoria statica (la logica dinamica e' inerentemente priva di alee). Per esempio, consideriamo un multiplexer con entrambi gli ingressi di dato ad 1, in cui l'ingresso di selezione passa da 1 a 0. L'uscita dovrebbe restare a 1, ma in realta' differenze nei ritardi tra le porte possono portare a un'alea in uscita.



Se si collegano in cascata due porte dinamiche di questo tipo, ci sono problemi per la precarica della seconda, in quanto ha gli ingressi attivi, portati al valore logico 1 dal primo pull-up. Il secondo pull-up non puo' quindi svolgere il suo lavoro.

Per evitare che la rete di pull-down sia attiva durante la fase di precarica una prima parziale soluzione e' l'inserimento di un transistor NMOS tra la rete e la massa. Questo transistor viene chiamato footer.



Questo transistor, in ogni caso risolve solamente una parte dei problemi: l'ingresso della porta successiva si troverebbe comunque ad 1 entrando nella fase di valutazione, e quindi immediatamente inizierebbe a scaricare l'uscita in ogni caso.

La soluzione completa si ha inserendo tra l'uscita della prima porta e l'ingresso della seconda un inverter, che porta l'uscita delle porte dinamiche a 0, e quindi inattive, durante la precarica.

Questo tipo di logica è chiamato **logica domino**.

Vantaggi e svantaggi rispetto alla logica statica

Vantaggi:

- piccola (2—3 volte più piccola in quanto è presente un solo PMOS)
- veloce (minore capacità di gate, solo quella dell'NMOS)

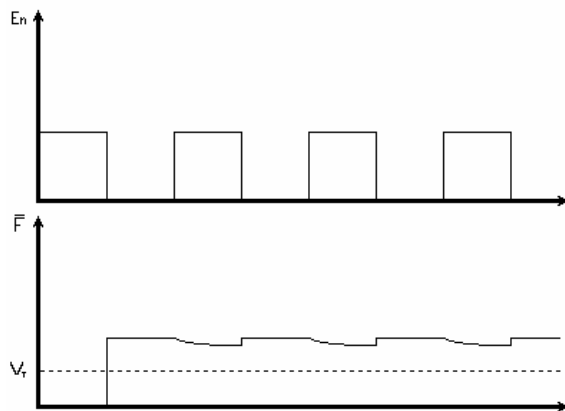
Incerta:

- potenza dissipata (dovuta principalmente, come vedremo, a carica e scarica delle capacità):
 - può aumentare frequenza di carica scarica delle capacità, se un segnale ha prevalentemente valore logico 1, e con la logica domino viene portato a 0 durante ogni ciclo di precarica.
 - riduce sempre le capacità.

Svantaggi:

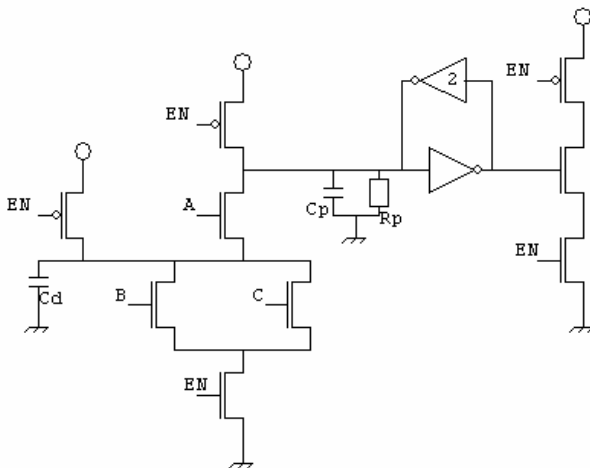
- difficoltà di progettazione (sensibile alle caratteristiche parassite, al dimensionamento dei transistor, al rumore, all'accoppiamento capacitivo)

1. Sensibilità al rumore: alee nel circuito possono essere generate o dalla logica statica (come visto prima) o da accoppiamenti capacitivi. Queste ultime si manifestano quando un segnale che subisce una transizione ha un forte accoppiamento capacitivo con un altro che dovrebbe essere stabile. La soluzione e' la riduzione delle capacità parassite tra le interconnessioni (cross-talk) direttamente nella progettazione del layout
2. Perdita di carica: la capacità non e' completamente isolata, in quanto i transistor nominalmente "spenti" in realta' conducono per le correnti inverse di giunzione (verso il substrato) e per le correnti di tunnel (verso il source).



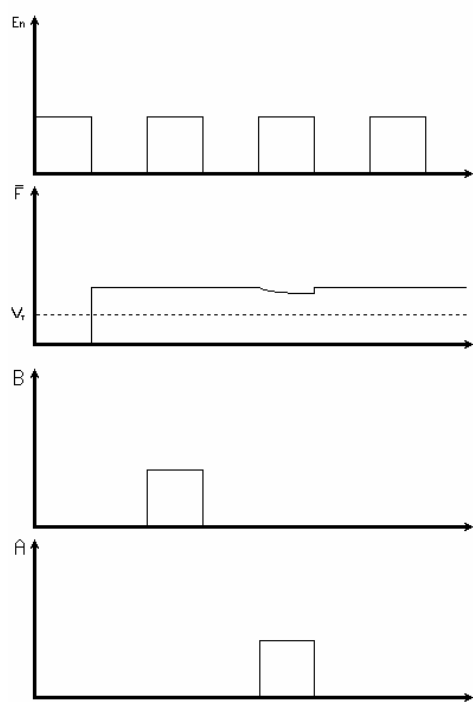
Le soluzioni possono essere:

- garantire una frequenza minima di funzionamento
- Soluzione 2: inverter di mantenimento (keeper o bleeder; n° 2 nella figura seguente), in reazione positiva per bilanciare le correnti di perdita, che deve essere piccolo per non contrastare i transistor di precarica e pull-down della porta precedente ($I_{dsat2} \ll I_{dsat1}$, quindi $W = W_{min}$, $L > L_{min}$).



3. Charge sharing: il problema si manifesta quando vengono scaricate le capacità interne della porta, per esempio nella figura soprastante Cd, cioè la capacità di drain e source tra i transistor pilotati da B e C e quello pilotato da A. Se nel ciclo di valutazione successivo solo A e' alto (e quindi l'uscita dovrebbe restare alta), la carica si ridistribuisce tra le capacità parassite, e quindi la tensione di uscita diventa $V_{dd} \cdot C_p / (C_p + C_d)$. Se Cp e' grande (per esempio, se ci sono molti transistor in parallelo come C e D), la tensione di uscita' puo' essere troppo bassa, riducendo il

margine di rumore (gia' problematico).

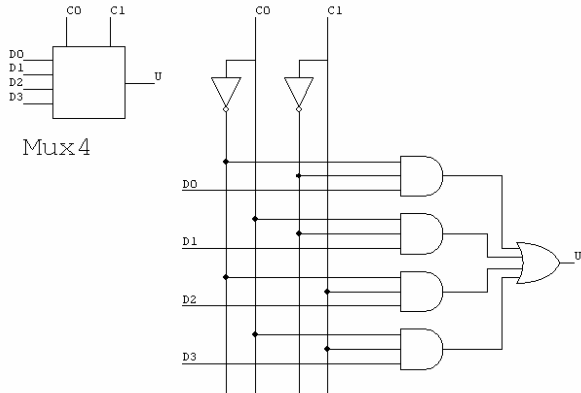


Soluzione: se si può ridurre le capacità parassite interne della rete pull-down; se non è possibile, si può usare un transistor come nella figura precedente per precaricare anche le capacità più significative tra quelle dei nodi interni.

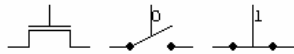
La logica domino viene utilizzata solamente per le parti critiche di un circuito, progettate e amano, poiché il supporto CAD è ancora quasi del tutto inesistente. Per esempio, nelle CPU viene usata per l'ALU o per il decoder del register file.

Logica ad interruttori (pass transistor)

Utilizzando la logica statica sono necessari per realizzare un mux a 4 vie almeno 36 transistor, che aumentano con legge quadratica all'aumentare del numero di ingressi

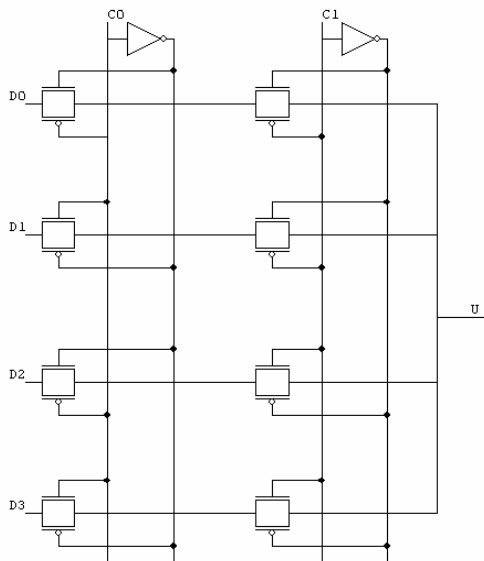


Nella realizzazione a pass transistor si usano i transistor direttamente come interruttori, collegati all'uscita di porte precedenti anziché a massa (come nella logica statica).



Anche nella realizzazione a pass-transistor, come nella logica statica e diversamente dalla logica dinamica, è sempre garantito il cammino tra Vdd o GND e l'uscita.

Il multiplexer può essere realizzato usando coppie di transistor in parallelo, un NMOS e un PMOS in modo da trasmettere ugualmente bene il valore logico 0 e 1, con ingressi di controllo complementari, in modo da accenderli insieme (se l'interruttore deve essere chiuso) o spegnerli insieme (se l'interruttore deve essere aperto). Richiede soltanto 20 transistor, rispetto ai 36 della logica statica.



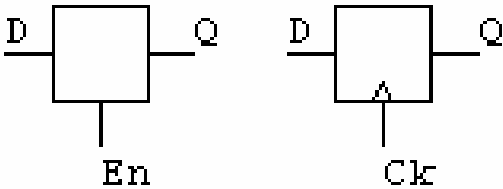
Problemi:

- Se si mettono solo transistor NMOS (piu' piccoli), questi non conducono bene l'uno logico. Per questo si inserisce anche un PMOS in parallelo, che conduce quando conduce l'NMOS.
- Un multiplexer ad n ingressi di controllo avrebbe n transistor in serie, che portano un aumento quadratico dei ritardi. Questo problema si risolve mettendo ogni 3-4 celle un inverter, che riporta i ritardi ad una funzione lineare.

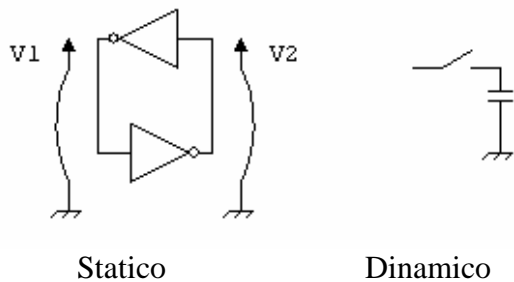
Logica sequenziale

I principali elementi sequenziali usati nel progetto CMOS sono:

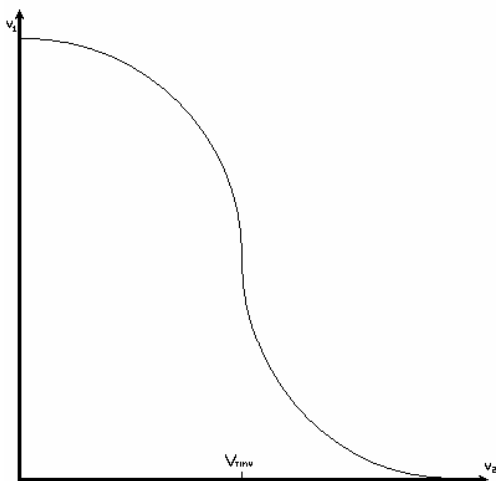
- il latch, attivo sul livello, trasparente quando il segnale En e' alto e in memoria quando questo e' basso.
- il flip-flop, attivo sul fronte, che copia su Q il dato presente su D quando arriva il fronte attivo (salita o discesa) so Ck .



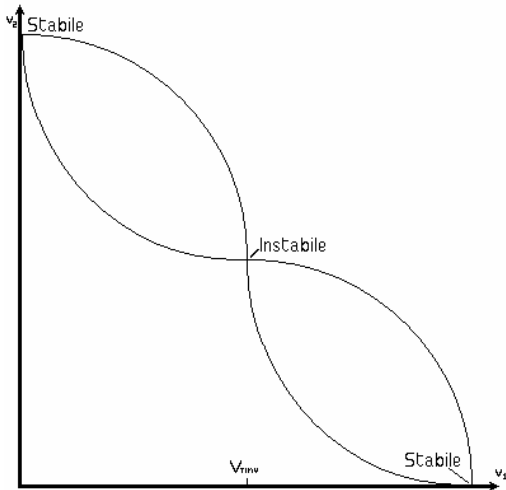
Il latch elementare CMOS puo' essere realizzato, come sempre, in forma statica con una coppia di inverter in reazione positiva, oppure in forma dinamica, con una capacita' controllata da un interruttore.



La caratteristica di funzionamento del latch statico si ottiene, come per ogni collegamento ciclico di bipoli, incrociando le caratteristiche V_{in}/V_{out}



Considerando che la V_{in} del secondo inverter e' la V_{out} del primo e vice-versa, la seconda caratteristica e' semplicemente simmetrica scambiando gli assi:



Si nota che esistono tre punti di funzionamento, di cui due sono stabili (variazioni dovute al rumore riportano al punto di funzionamento) e uno instabile (variazioni dovute al rumore portano ad uno dei due punti stabili). Si può dimostrare che, in base al teorema che dice che ogni funzione continua ha un massimo tra due minimi, ogni dispositivo di memorizzazione statica con un bit di capacità ha un punto di metastabilità

Il circuito completo del latch statico, includendo anche il multiplexer che permette di metterlo in modo trasparente o di memoria, è riportato nella figura seguente.

