

PROGETTAZIONE DI SISTEMI MICROELETTRICI DIGITALI

[Fotocopie di Appunti]

A CURA DI ALESSANDRO PAGHI

PROFESSORE: Luca Fanucci (<http://unimap.unipi.it/cercapersone/dettaglio.php?ri=5021>)

LINK AL CORSO ANNO 2016/2017: Materiale didattico fornito su DropBox.

FREQUENTAZIONE: Consigliata.

CIRCUITO INTEGRATO (IC)

È un dispositivo avente più componenti elettronici e le loro connessioni su un singolo substrato.

AVANZAMENTO TECNOLOGICO :

Manufacturing : Tecnologia commerciale.

Development : Tecnologia commerciale ma non ancora commercializzata.

Research : Tecnologie in fase di sviluppo.

LEGGE DI MOORE : MINIALIZATION

Il numero di componenti di un circuito integrato raddoppia ogni 18 mesi.

MORE THAN MOORE : DIVERSIFICATION

Diversificazione delle funzioni dei dispositivi elettronici.

SOC : SYSTEM ON A CHIP

Molte funzionalità integrate su singolo chip.

IP : SYSTEM IN A PACKAGE

Chip separati che svolgono diverse funzionalità o unità all'interno di un unico package.

LSI : VERY LARGE SCALE INTEGRATION

Sistema integrato contenente più di 10.000 transistor.

ITRS : INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS

Definire gli obiettivi da raggiungere, in termini di nodi tecnologici, per le aziende del settore IC.

LIVELLI DI ASTRAZIONE:

- 4) RTL : Registry Transfer Level;
- 3) GATE ;
- 2) TRANSISTOR ;
- 1) LAYOUT ;

APPROCCI ALLA PROGETTAZIONE DI UN CIRCUITO DIGITALE

1) FULL CUSTOM

Il progettista ha massima libertà nel progettare il design dell'IC.

Si ottiene un miglioramento di prestazioni in termini di velocità di esecuzione di un task e di riduzione di area, a patto di innalzare costi di progettazione e tempo di sviluppo.

Sono richieste competenze in tecnologie ASIC, circuiti elettronici ed architetture elettroniche.

2) SEMI-CUSTOM

Il designer definisce interconnessioni tra blocchi già progettati.

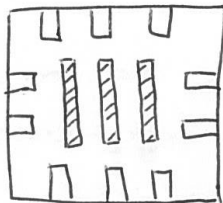
non fornisce un'ottimizzazione globale ma riduce costi e tempo di implementazione.

Sono richieste competenze più elevate nel settore dei circuiti elettronici.

2.1) GATE ARRAY TECHNOLOGY

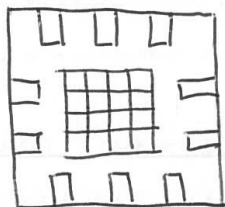
Il IC è costituito da una matrice di transistor predefinitamente collegati.

Il progettista si occupa di interconnettere i transistor tramite linee di metal.



2.2) SEA OF GATE TECHNOLOGY

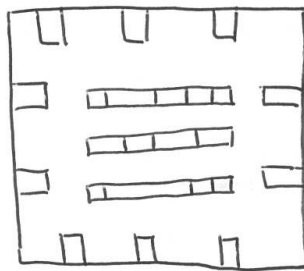
Non ci sono canali dedicati alle interconnessioni, a differenza del caso GATE ARRAY.



2.3) STANDARD CELL TECHNOLOGY

La tecnologia di base dell'utente di celle progettate da terzi. L'ordine di queste celle è fisso, mentre la layout varia con la complessità del sistema integrato.

Queste celle vengono organizzate in righe separate da canali di grondaia riservati dedicati alle interconnessioni.



Tali celle si suppongono ottimizzate in termini di area e di prestazioni.

C'è la possibilità di utilizzare macrocelle come RAM, ROM, ADC, DAC, ...

2.4) FPGA TECHNOLOGY (FIELD PROGRAMMABLE GATE ARRAY)

La tecnologia di base dell'utente di ^{programmabili} celle V disposte su array con interconnessioni programmabili.

Il risultato è un dispositivo non ottimizzato in termini di area e di performance.

$$NRE \sim \phi$$

RE sono molto maggiori rispetto a una tecnologia FULL CUSTOM

Quando si usa FPGA ?

- Budget insufficiente per coprire NRE ;
- Basso e medio volume di produzione ;
- Prototipazione ;

METODI DI PROGRAMMAZIONE FPGA

1) ANTIFUSE

Due masi vengono fisicamente collegati ponendosi in fusione del materiale.

→ Programmabile una sola volta.

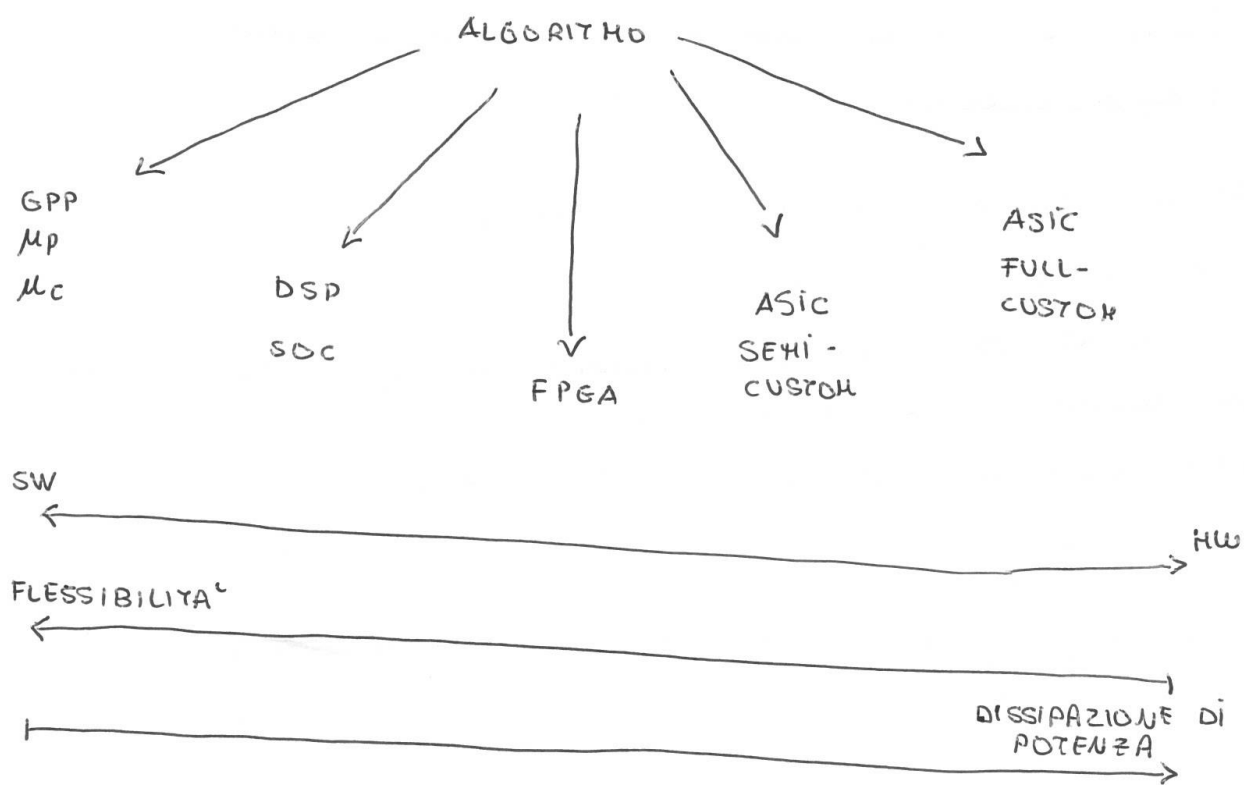
2) SRAM BASED

La programmazione di un'interconnessione avviene mediante l'utilizzo di una cella SRAM.

Lo stato della connessione dipende dal valore associato nella cella SRAM.

→ Riprogrammabile.

MAPPARE UN ALGORITHM CON UN DESIGN HW - SW



ASIC SOLUTION COSTS

$$\text{Total cost} = \text{Fixed cost} + m \cdot \text{Recurrent cost}$$

m : number of product units.

Fixed cost := NRE including:

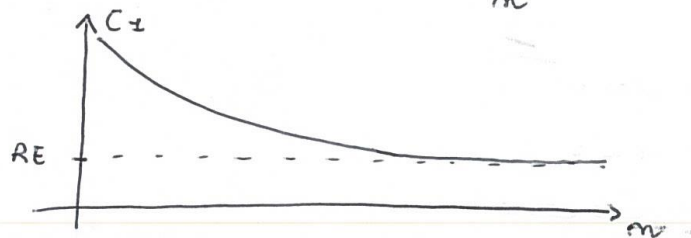
- Costs of Design;
- Costs of Production & Test fixed;

Recurrent cost: RE including:

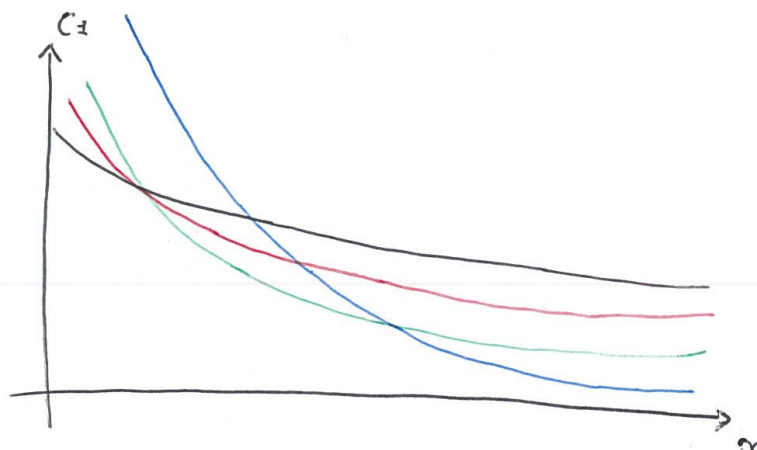
- Costs of production during the process;
- Costs of Test during the process;

$$\text{Costo Totale} : C_T = NRE + m \cdot RE$$

$$\text{Costo per unità} : C_1 = \frac{NRE}{m} + RE.$$



COMPARISON COST



- FPGA
- GATE ARRAY
- STANDARD CELL
- FULL CUSTOM

METRICHE DI DESIGN FONDAMENTALI

Operazioni di progettazione di un chip:

- 1) Creazione di un wafer;
- 2) Fabricazione di circuiti integrati sul wafer;
- 3) Test degli IC;
- 4) Packaging dei die;
- 5) Test del Package;

COSTO

URE:

- Costi fissi per la produzione di un design;
- Influenzati dalla complessità del progetto;
- Molto promemmati per bassi volumi di produzione;

RE:

- Processi su silicio, proporzionali all'area del wafer;
- Costi di Packaging;
- Test;

$$C_t = \frac{URE}{n} + RE$$

$$RE = (\text{Cost of die} + \text{Cost of die test} + \text{Cost of Packaging}) / \frac{\text{Area}}{\text{Process}}$$

$$\text{Cost of die} = \frac{\text{Cost of wafer}}{\text{dies per wafer} \cdot \text{Area die}}$$

$$\text{Area die} \propto a^{-3 \div 4}$$

a: misura complessità del processo.

⑧

AFFIDABILITÀ, ROBUSTEZZA

Rumore: Narazioni non volute di notafgato e
esistente in un modo logico.

Noise margin: Espone la capacità di un
esecuto di sopraggiare una ostigento
di rumore (rumore aggiuntivo,
cross talk, interferente, offset).

Un modo fessante il modo più ensicce al
rumore rispetto ad un modo pilotato a
cassa impedimento.

Noise Immunity: Espone la capacità di un
sistema di processare e trasmettere
informazione esistente in presenza di
rumore.

PERFORMANCE

Espresso in termini di velocità (ritardo
insesto da un esecuto) e consumo di
potenza.

LOGICA

COMBINATORIA

L'uscita è una funzione dei soli ingressi presenti in un determinato istante temporale.

SEQUENZIALE

L'uscita è funzione degli ingressi presenti e della loro storia.

SEQUENZIALE ASINCRONA

Le uscite variano in corrispondenza a variazioni degli ingressi.

- Design senza clock
- Alta velocità e bassi consumi.
- Race conditions: condizioni della velocità di variazione di certi segnali.
- Design difficile

SEQUENZIALE SINCRONA

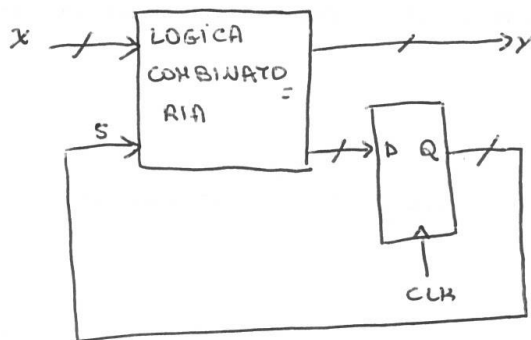
Le uscite variano solo in corrispondenza di un fronte di salita, o di discesa, di un segnale di riferimento, chiamato clock.

- Design con clock
- Facile Design e Ristrutturazione

alta complessità, alto consumo di potenza e velocità ridotta.

MACCHINA DI MEALY

Logica Asincrona Sequenziale



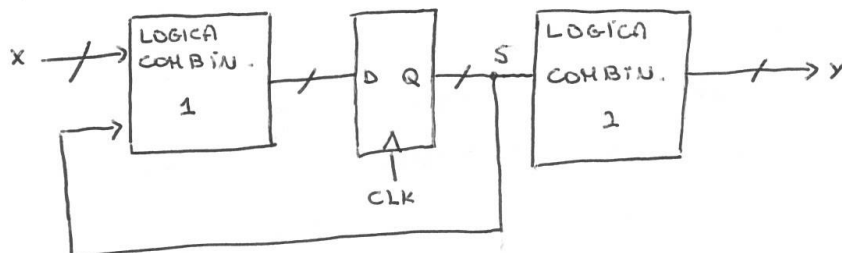
Le uscite sono determinate contemporaneamente dagli stati correnti e dagli input correnti

È una macchina trasparente: è presente un comune logico combinatorio tra INPUT e OUTPUT.

È un problema collegare più macchine di mealy in cascata.

MACCHINA DI MOORE

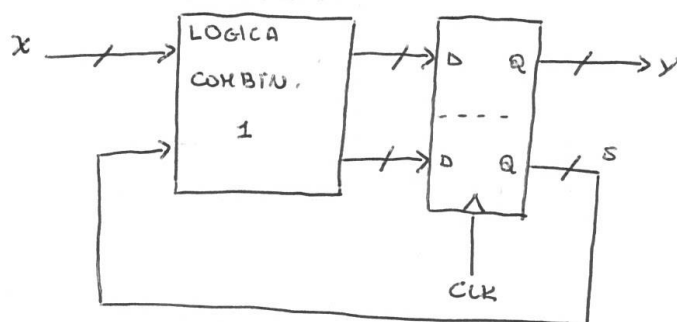
Logica Sincrona Sequenziale



Le uscite sono determinate solo dagli stati correnti.

MACCHINA DI HEALY MODIFICATA

Logica Simbolica Sequenziale



Le variabili sono
sintetizzate in
modo da risolvere
i problemi di
staticità.

Queste macchine possono essere costruite in:
LOGICA STATICA (HARDWARE)

Lo stato dei nodi viene preservato fin quando è
presente il segnale di alimentazione.

LOGICA DINAMICA (SOFTWARE)

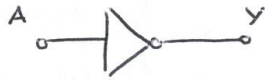
Gli stati sono memorizzati nelle capacità parassite
e sono quindi mantenibili per brevi intervalli
di tempo.

Sono richiesti refresh periodici.

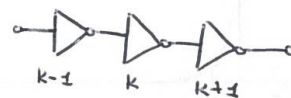
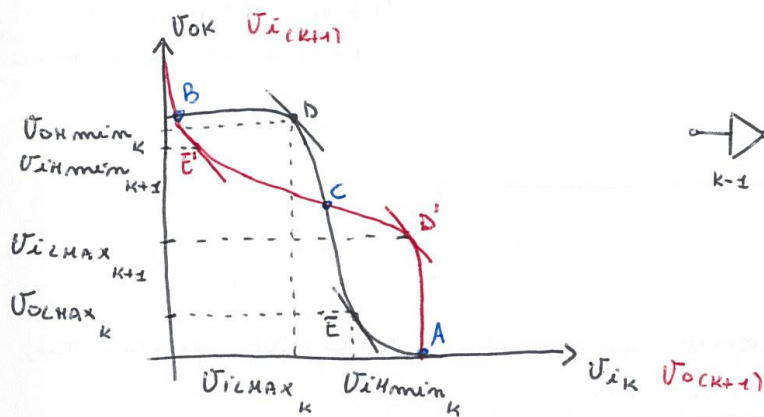
Altamente permette maggiore semplicità
realizzativa, maggiore velocità e minor consumo
di potenza.

ELETTRONICA DIGITALE

INVERTER



PARAMETRI STATICI



$$V_{OK} = V_{I(k+1)}$$

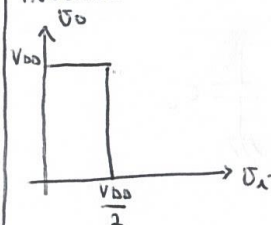
$$NM_H = V_{OHmin} - V_{IHmin}$$

$$NM_L = V_{ILmax} - V_{OLmax}$$

$$V_{soglia\ logica} = V_{SL} = \begin{cases} f(V_O) \\ V_O = V_I \end{cases}$$

$$P_{statica} = V_{DD} \left(\frac{I_{OL} + I_{OH}}{2} \right)$$

INVERTER IDEALE



$$NM_H = NM_L = \frac{V_{DD}}{2}$$

$$V_{SL} = \frac{V_{DD}}{2}$$

$$R_{in} = \infty$$

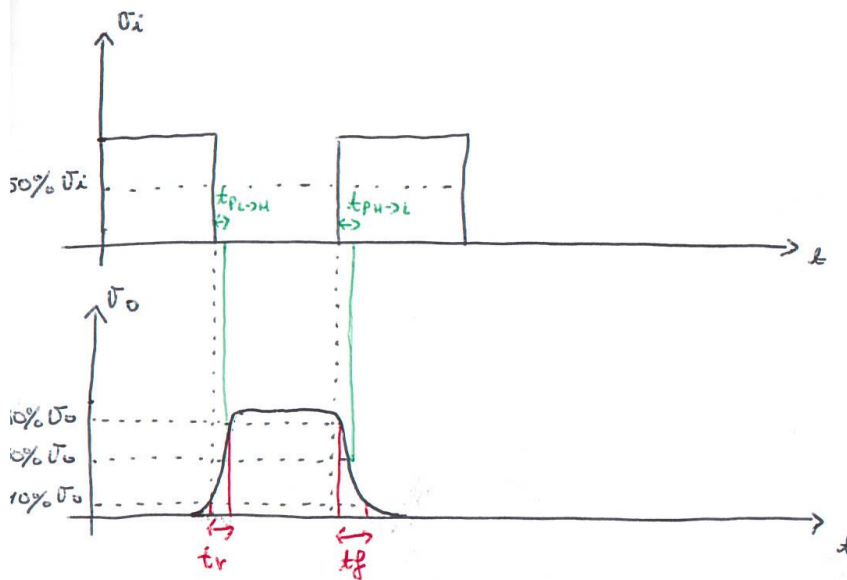
$$R_{out} = \phi$$

$$FAN\ OUT = \infty$$

FAN IN : Numero max di inverter collegabili in input

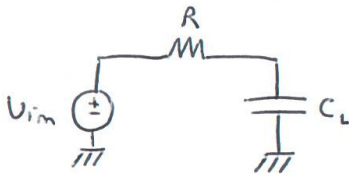
FAN OUT : Numero max di inverter collegabili in output.

PARAMETRI DINAMICI



t_r : tempo di salita
 t_f : tempo di discesa
 $t_{PL \rightarrow H}$: tempo di propagazione L $\rightarrow$ H
 $t_{PH \rightarrow L}$: tempo di propagazione H $\rightarrow$ L

Si schematizza l'inverter come un circuito equivalente del 1° ordine:



$$V_{out}(t) = (1 - e^{-t/\tau}) V_{im}(t)$$

con $\tau = R \cdot C$

$C_L = \sum C_i$: Sommatario delle capacità a valle più capacità di uscita inverter K

Se connesso un solo inverter $\rightarrow C_L = C_T$.

$$\rightarrow U_o = V_{FIN} - (V_{FIN} - V_{IN}) e^{-\frac{\Delta t}{\tau}}$$

$$t_r \sim 2,2 R_r \cdot C_L$$

$$t_f \sim 2,2 R_f \cdot C_L$$

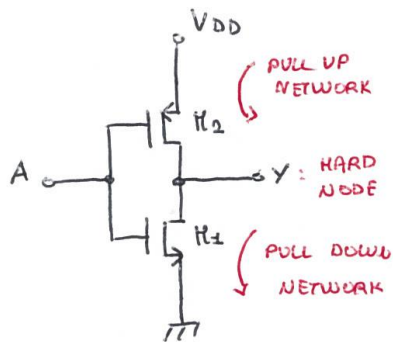
$$t_p = \frac{t_{PL \rightarrow H} + t_{PH \rightarrow L}}{2} \sim 0,69 R \cdot C_L \quad (\text{Tempo di propagazione})$$

$$f_{CLKMAX} = \frac{1}{t_r + t_f}$$

$$f_{DINAMICA1} = C_L \cdot V_{DD}^2 \cdot f_{H \rightarrow L}$$

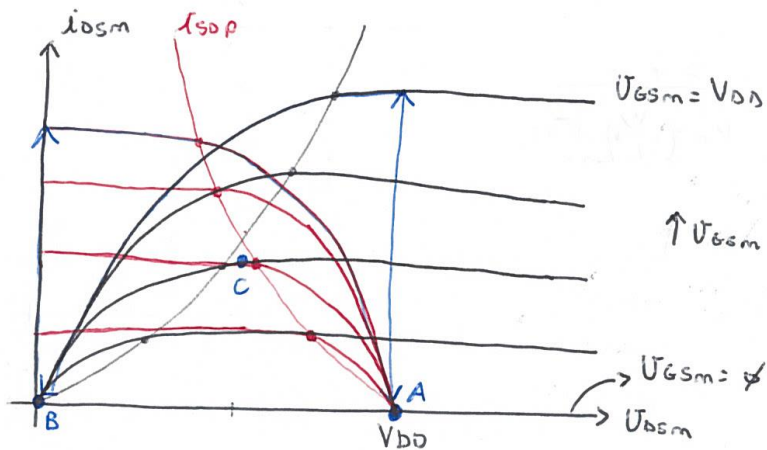
(11)

LOGICA CMOS



A	Y
0	1
1	0

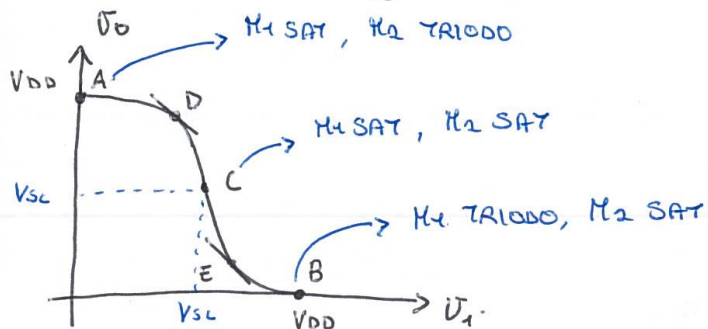
PARAMETRI STATICI



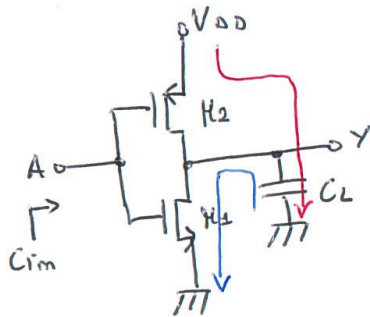
A: $V_i = 0$
 $V_o = V_{DD}$

B: $V_i = V_{DD}$
 $V_o = 0$

Statica: V_{DD} . $\frac{I_{OH} + I_{OL}}{2} = 0$



PARAMETRI DINAMICI



$$C_L = m \cdot C_T$$

metodo scorciatoia : ●

$$t_{p_{H \rightarrow L}} = \frac{K C_L}{\beta_m} \cdot \frac{1}{V_{DD} - V_{tm}}$$

metodo completo : ●

$$t_{p_{L \rightarrow H}} = \frac{K C_L}{\beta_p} \cdot \frac{1}{V_{DD} + V_{tp}}$$

$$C_{im} = C_{im_m} + C_{im_p}$$

$$\beta_m = \beta_m' \cdot \left(\frac{W}{L}\right)_m$$

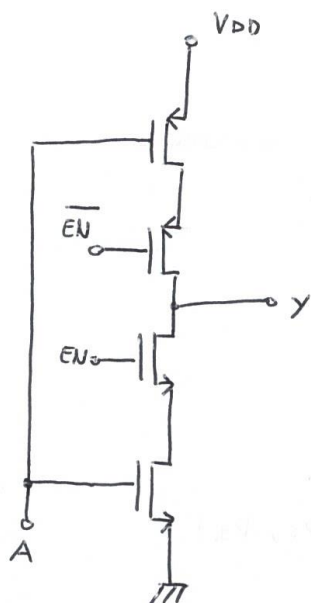
$$\beta_p = \beta_p' \cdot \left(\frac{W}{L}\right)_p = \frac{\beta_m'}{2,5} \cdot \left(\frac{W}{L}\right)_p$$

$$\rightarrow C_{im_m} = \frac{\epsilon W \cdot L}{d} \quad \rightarrow C_{im} = 3,5 C_{im_m}$$

(16)

CMOS TECHNOLOGY

INVERTER THREE-STATE



EN	A	y
0	0	z
0	1	z
1	0	1
1	1	0

SWITCH MODEL N-MOS TRANSISTOR



$$V_{GS} < V_{tn}$$



$$V_{GS} > V_{tn}$$



SWITCH MODEL P-MOS TRANSISTOR



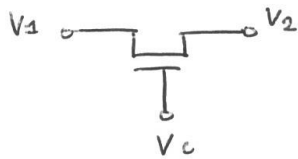
$$|V_{GS}| < |V_{tp}|$$



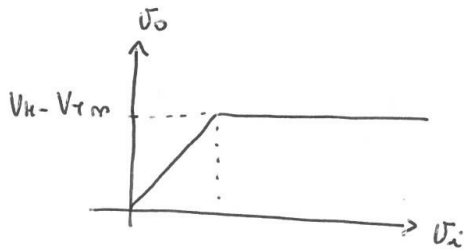
$$|V_{GS}| > |V_{tp}|$$



PASS TRANSISTOR N

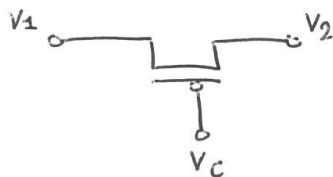


$$V_0 = V_S = \min(V_1, V_2)$$

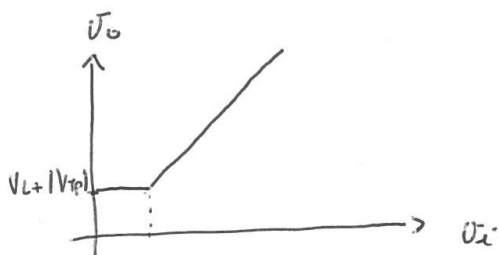


Degradazione segnale
logico alto.

PASS TRANSISTOR P

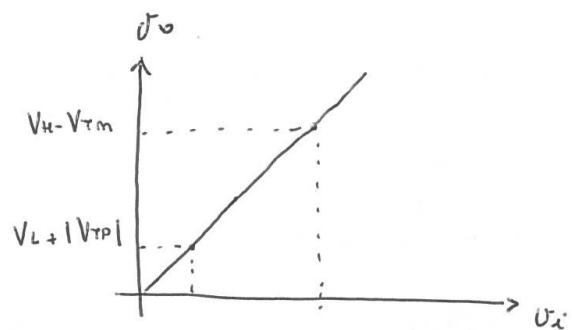
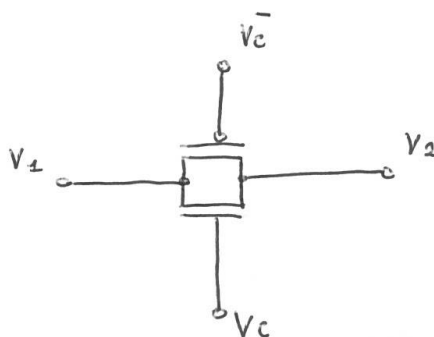


$$V_0 = V_S = \max(V_1, V_2)$$

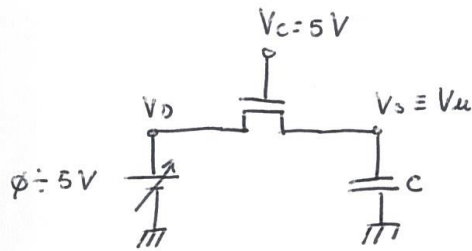


Degradazione segnale
logico basso.

PASS GATE



PASS TRANSISTOR A RESISTENZA EQUIVALENTE



$$\beta_m = \mu_m C_{ox} \frac{W}{L}$$

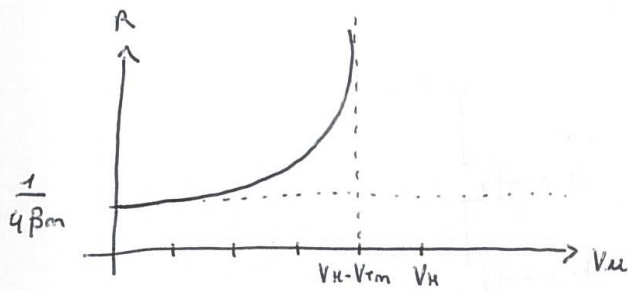
$$I_D = \beta_m \left[(V_{GS} - V_t) V_{DS} - \frac{V_{DS}^2}{2} \right]$$

$$\sim \beta_m (V_{GS} - V_t) V_{DS}$$

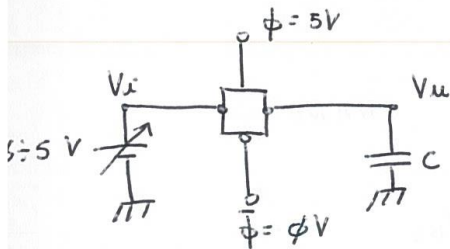
$$\frac{1}{R} = \frac{dI_D}{dV_{DS}} = \beta_m (V_{GS} - V_t)$$

$$R = \frac{1}{\beta_m (V_{GS} - V_t)} = \frac{1}{\beta_m (V_U - V_t)}$$

$$V_t = \pm V$$



PASS GATE RESISTENZA EQUIVALENTE

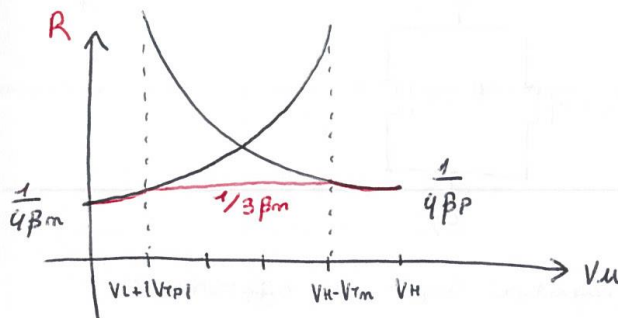


$$R_m = \frac{1}{\beta_m (V_U - V_t)}$$

$$R_p = \frac{1}{\beta_p (V_U - V_t)}$$

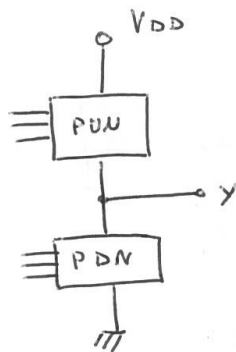
$$V_{tp} = -V$$

$$V_{tm} = V$$



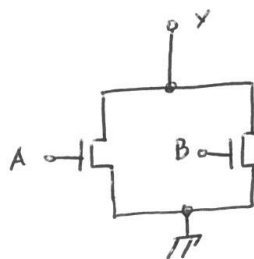
$$\beta_m = \beta_p = \beta$$

COMPLEMENTARY CMOS

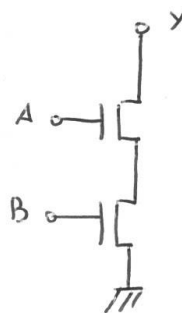


PDN

$$\bar{y} = A + B \quad (\text{NOR})$$

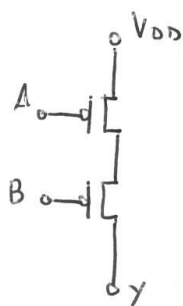


$$\bar{y} = A \cdot B \quad (\text{NAND})$$

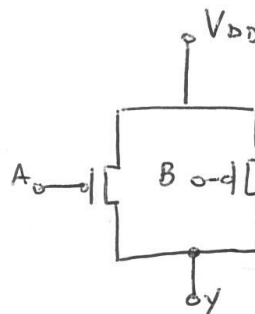


PUN

$$\bar{y} = A + B \quad (\text{NOR})$$



$$\bar{y} = A \cdot B \quad (\text{NAND})$$



$$\bar{y} = f_{PDN}(A, B) \quad \text{funzione negata degli in affermati.}$$

$$y = f_{PUN}(\bar{A}, \bar{B}) \quad \text{funzione affermata degli in negati.}$$

(20)

Funzione implementabile in logica PDN + PUN:

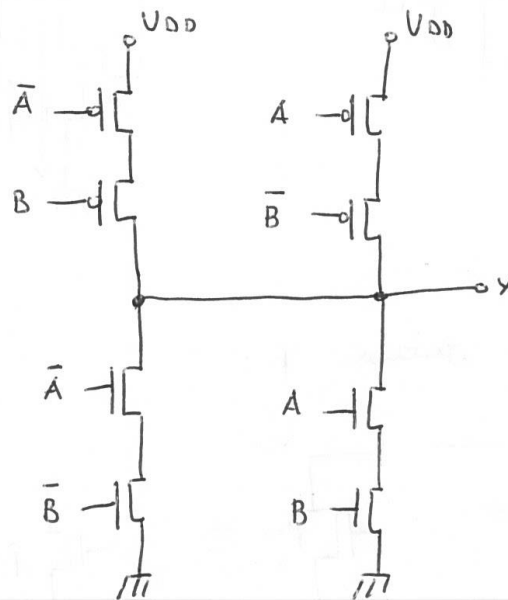
$$Y = \overline{A(B + CD)}$$

Funzione non implementabile in logica PDN + PUN:

$$Y = A \oplus B = \bar{A}B + A\bar{B}$$

Come lo realizzo?

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

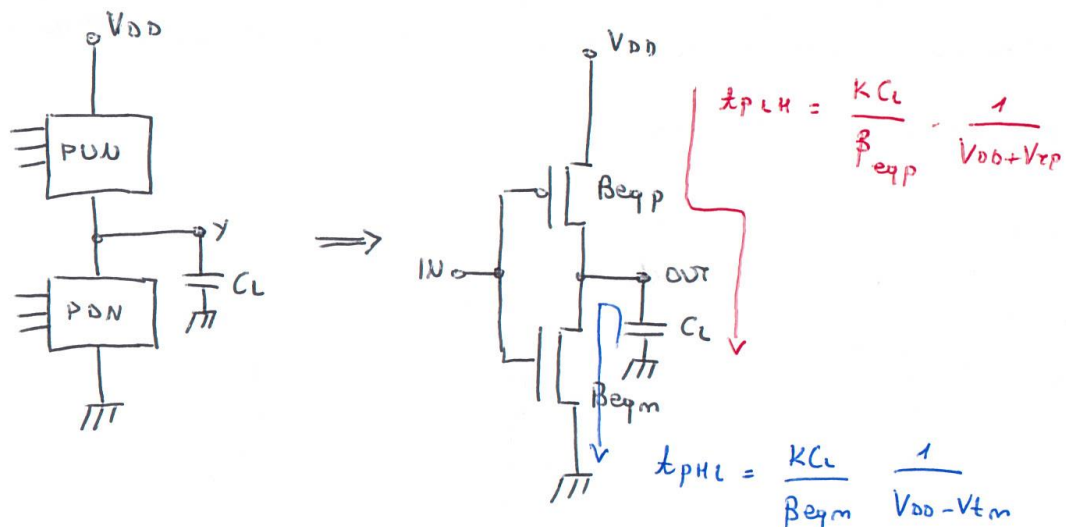


Questa è la LOGICA COMPLEX GATE

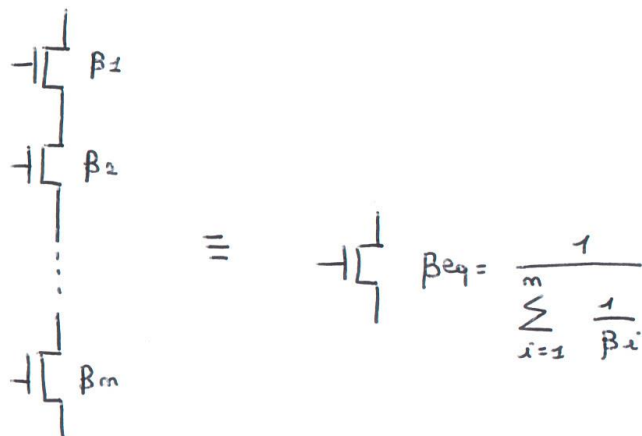
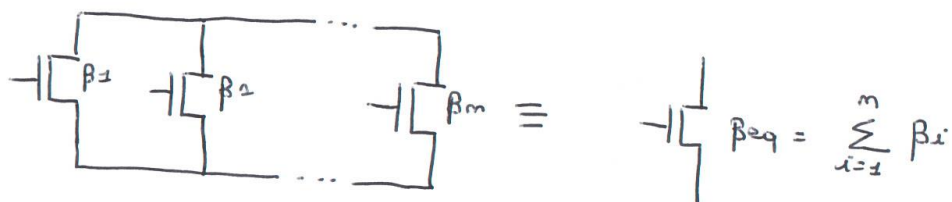
→ mintermini che portano a VDD + mintermini che portano a massa.

Analisi Dinamica di una rete PUN + PUN :

→ Ci si riconduce ad uno schema INVERTER.



Analisi dei fattori β :

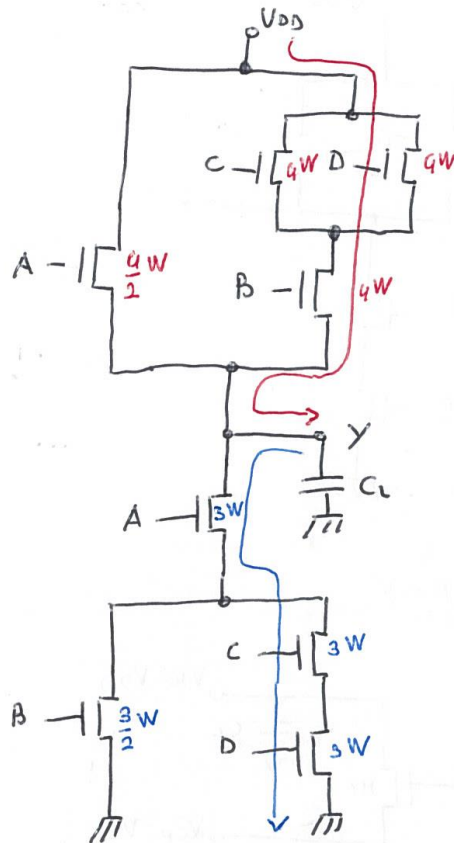


Esercizio:

$$Y = \overline{A(B+CD)}$$

A	B	C	D	Y	β
0	0	0	0	1	$5/3 \cdot \beta_P$
0	0	0	1	1	$3/2 \cdot \beta_P$
0	0	1	0	1	$3/2 \cdot \beta_P$
0	0	1	1	1	$1 \cdot \beta_P$
0	1	0	0	1	$1 \cdot \beta_P$
0	1	0	1	1	$1 \cdot \beta_P$
0	1	1	0	1	$1 \cdot \beta_P$
0	1	1	1	1	$1 \cdot \beta_P$
1	0	0	0	1	$2/3 \cdot \beta_P$
1	0	0	1	1	$1/2 \cdot \beta_P$
1	0	1	0	1	$1/2 \cdot \beta_P$
1	0	1	1	0	$1/3 \cdot \beta_m$
1	1	0	0	0	$1/2 \cdot \beta_m$
1	1	0	1	0	$1/4 \cdot \beta_m$
1	1	1	0	0	$1/4 \cdot \beta_m$
1	1	1	1	0	$1/5 \cdot \beta_m$

CASI
PEGGIORI



$$\beta_{PEQ} = \frac{1}{2} \beta_P = \frac{1}{2} \cdot \mu_P \cdot \frac{W_P}{L_P}$$

$$\beta_{MEQ} = \frac{1}{3} \beta_m = \frac{1}{3} \cdot \mu_m \cdot \frac{W_m}{L_m}$$

$$t_{PLH} = \frac{K C_L}{\beta_{EQP}} ; \quad t_{PHL} = \frac{K C_L}{\beta_{EQM}}$$

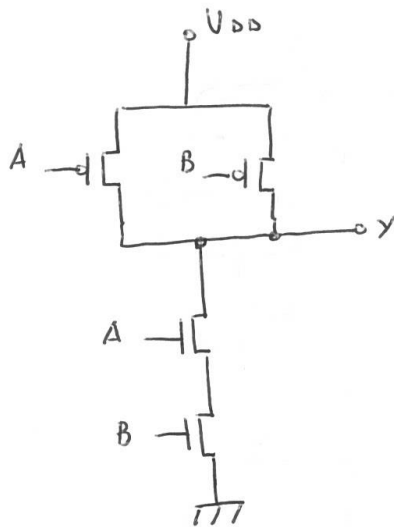
$$\Rightarrow t_{PLH} = t_{PHL} \rightarrow \frac{1}{2} \mu_P \frac{W_P}{L_P} = \frac{1}{3} \mu_m \frac{W_m}{L_m}$$

Supponendo $\mu_P = \mu_m/2$

$$L_m = L_P = L_{min} \Rightarrow \frac{W_P}{W_m} = \frac{4}{3}$$

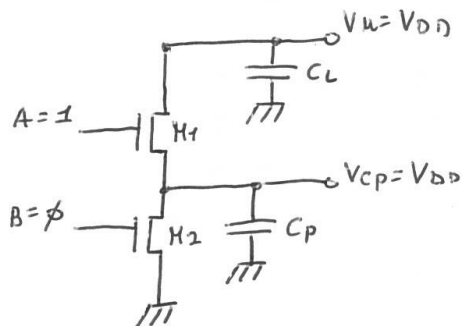
CHOS BODY EFFECT

Porta NAND

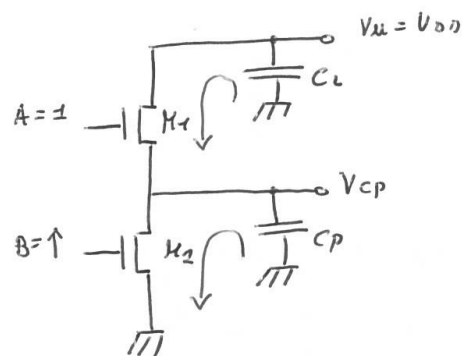


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

$A=1$, $B=0$



$A=1$, $B=\uparrow$

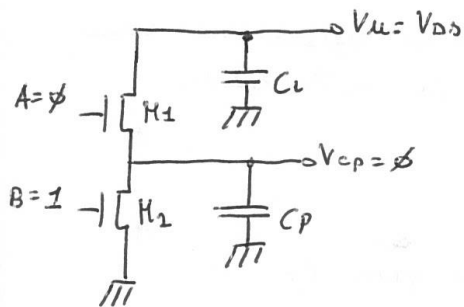


Inizialmente C_P si scarica su M_2 .

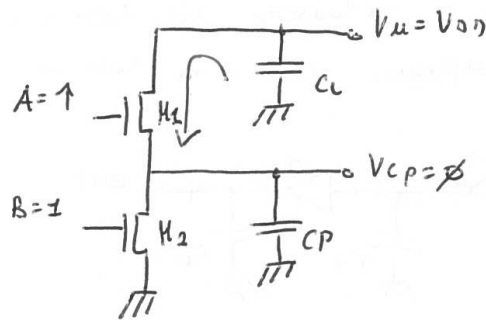
M_1 inizia lo scaricamento con $V_{S_1} = V_{DD}$ e quindi l'assenza, evidente a causa dell'effetto body.

Il tempo di propagazione è più lungo a questo stimolo con β_{MEQ} .

$A = \emptyset, B = 1$



$A = 1, B = 1$



Adesso è source di H_1 R_{on} $V_{S1} = \emptyset$.

L'effetto body è ridotto e sarà presente una commutazione più veloce.

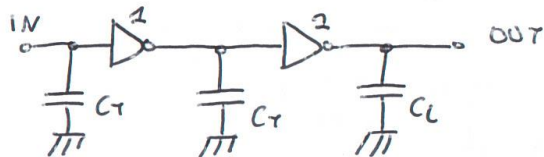
La commutazione dell'uscita è più veloce quando varia un ingresso logicamente più vicino all'uscita.

Quando commuta A, qui varia più velocemente di quando commuta B.

Il tempo di propagazione aumenta sensibilmente con l'aumentare del FAN IN.

INVERTER CHAIN : PILOTARE UN CARICO CON ALTA CAPACITÀ

Oiettivo : ridurre il tempo di propagazione nel pilotaggio di un carico con C_L elevato.



$$C_L = 1000 \cdot C_T$$

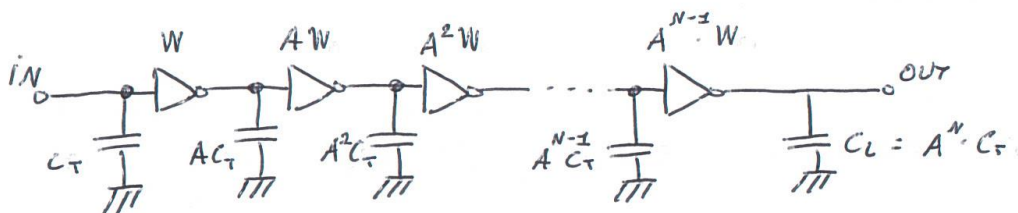
$$t_{p2} = \frac{K C_L}{\beta} = 1000 \cdot \frac{K C_T}{\beta} = 1000 \cdot t_{pAIF}$$

$$t_{pAIF} = \frac{K C_T}{\beta}$$

Se $W_2 = 1000 \cdot W_{AIF} \rightarrow t_{p2} = t_{pAIF}$

ma $C_{im2} \neq C_T$, $C_{im2} = 1000 \cdot C_T \Rightarrow t_{p1} = 1000 \cdot t_{pAIF}!$

non ho risolto il problema.



$$t_{p1} = K \cdot \frac{A C_T}{\beta} = A \cdot t_{pAIF}$$

$$t_{p2} = K \cdot \frac{A^2 C_T}{A \cdot \beta} = A \cdot t_{pAIF} = t_{p1}$$

$$t_{pTOT} = N \cdot A \cdot t_{pAIF} = \frac{\ln\left(\frac{C_L}{C_{im}}\right)}{\ln(A)} \cdot A \cdot t_{pAIF}$$

$$\frac{d}{dA} t_{pTOT} = 0 \rightarrow A = e = 2,71$$

*Per le regole di Layout
A = 3.*

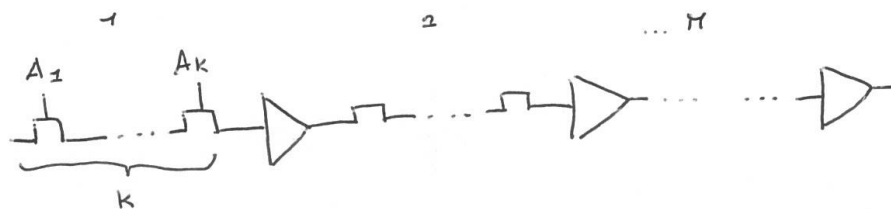
$$C_L = e^N \cdot C_T \rightarrow N = \ln \frac{C_L}{C_{im}} \rightarrow N = \frac{\ln\left(\frac{C_L}{C_{im}}\right)}{\ln(3)}$$

PASS TRANSISTOR CHAIN

Se si utilizzano transistor di tipo n si ha il degrado del livello logico vero.

Ottimizzazione del ritardo:

La rete di transistor viene spezzata da un certo numero di Buffer.



k transistor per blocco.

M blocchi.

N : numero totale di transistor

$$N = k \cdot M, \quad M = \frac{N}{k}$$

$$T_{tot} = M \cdot T_B + M \cdot (k^2 \cdot T_P) = \frac{N}{k} \cdot T_B + N k T_P$$

$$\frac{dT_{tot}}{dk} = 0 \Rightarrow k = \sqrt{\frac{T_B}{T_P}}$$

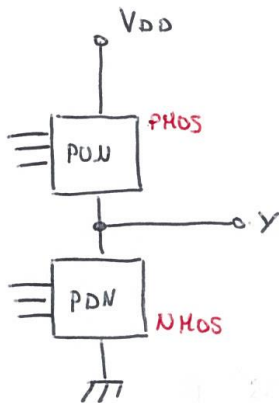
Nei reti di transistor il ritardo cresce in proporzione a k^2 non k .

Ottimizzazione: PASS GATE CHAIN

- No degrado livello logico;
- Ritardo T_P minore;

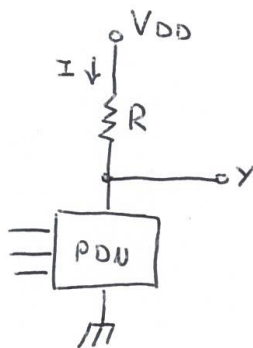
CELLE LOGICHE IN LOGICA STATICA

CMOS COMPLEMENTARE



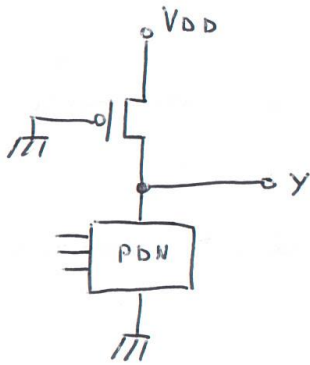
- Ottimo noise margin
- $P_{statica} = \emptyset$
- Complementari per funzione:
2-N MOS

LOGICA A RAPPORTO



- Complementari per funzione:
NMOS + R
- $V_{OR} = V_{DD}$
- $V_{OL} = \frac{R_{PDN}}{R_{PDN} + R} \cdot I$
- Risposta Asimmetrica
- $P_{statica_{LOW LEVEL}} \neq \emptyset$

LOGICA PSEUDO N-KOS

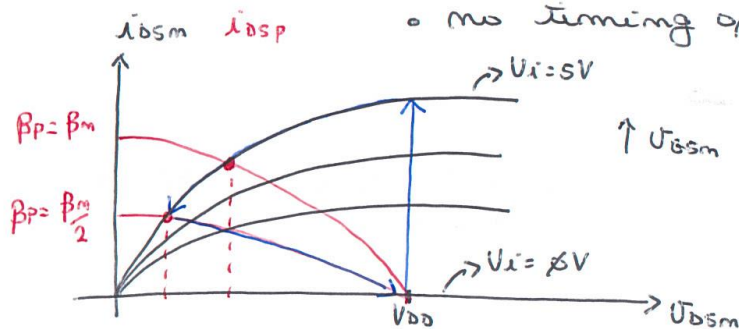


Vantaggi:

- $N+1$ MOS
- Area ridotta
- C_{in} minore

Svantaggi:

- $P_{statica} \neq 0$
- Dipendenza da β
- no timing optimization



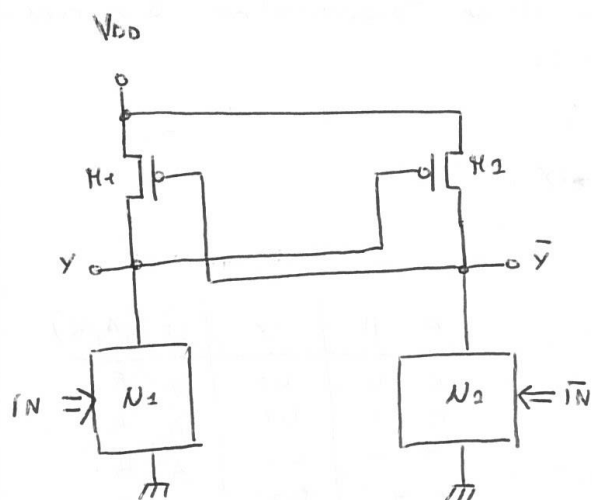
$$\beta_M \gg \beta_P.$$

Riducendo β_P non degrado le linee logiche.

Riducendo β_P , riduco i_{dsp} e quindi $P_{statica}$.

In compenso aumenta t_{PLH} .

DUAL CASCADE VOLTAGE SWITCH LOGIC



- No dissipazione statica
- No degradazione livello logico basso
- $2N_{MOS} + 2P_{MOS}$
- Temporizzazione delle uscite Y e $\bar{Y}$ quasi simultanea a differenza del CMOS.
- Prestazioni del sistema dipendono da β

Se $Y=1$, $\bar{Y}=\phi \Rightarrow K_2 \text{ OFF}, N_2 \text{ conduce}$
 $K_1 \text{ ON}, N_1 \text{ interdetta}$

Commutazione ingressi per cui risultata $Y=\phi$ e $\bar{Y}=1$:

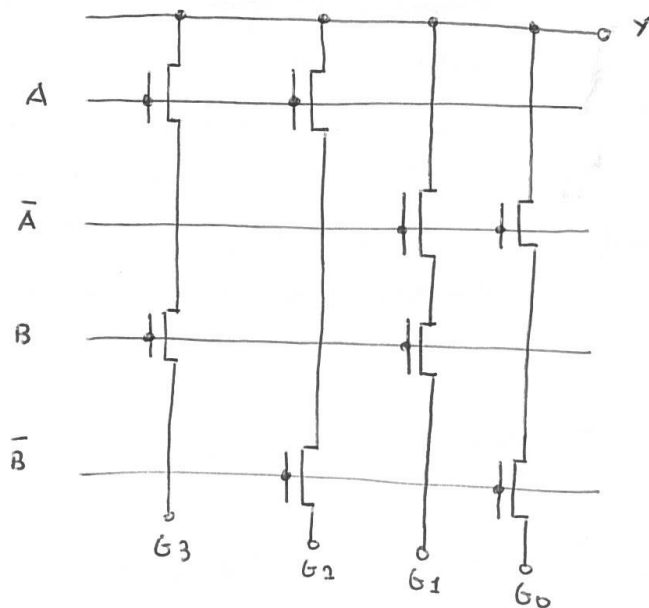
U_1 inizia a condurre ma K_2 comincia a condurre solo quando $|V_{GS}| > |V_{tp}|$ ovvero $Y < V_{DD} - |V_{tp}|$.

$\bar{Y}$ rimane momentaneamente pari a 0 perché N_2 si è interdetto.

Nei momenti in cui $Y < V_{DD} - |V_{tp}|$, K_2 conduce e $\bar{Y}=1$.

PASS TRANSISTOR LOGIC

Si realizza una catena di Pass Transistor dipendenti
ingressi ed ingressi negati.



A	B	y	F(A, B)
0	0	G ₀	$\bar{A} \cdot \bar{B}$
0	1	G ₁	$\bar{A} \cdot B$
1	0	G ₂	$A \cdot \bar{B}$
1	1	G ₃	$A \cdot B$

Tramite G si possono ottenere 2^4 combinazioni delle uscite

DEC	G ₃	G ₂	G ₁	G ₀	y
0	0	0	0	0	ϕ
1	0	0	0	1	$\overline{A+B}$
2	0	0	1	0	$\bar{A} \cdot B$
3	0	0	1	1	$\bar{A}$
4	0	1	0	0	$A \cdot \bar{B}$
5	0	1	0	1	$\bar{B}$
6	0	1	1	0	$A \oplus B$
7	0	1	1	1	$\overline{A \cdot B}$
8	1	0	0	0	$A \cdot B$
9	1	0	0	1	$A \oplus B$
10	1	0	1	0	B
11	1	0	1	1	$\bar{A} + B$
12	1	1	0	0	A
13	1	1	0	1	$A + \bar{B}$
14	1	1	1	0	$A + B$
15	1	1	1	1	1

Limiti della PCL:

• minimizzazione logica inefficiente

Se sono presenti N ingressi ed 1 uscita:

- 2^N LINEE

- $2 \cdot N$ MOS

• Ingressi negati ed affermati

• Degradamento velocità logico vero

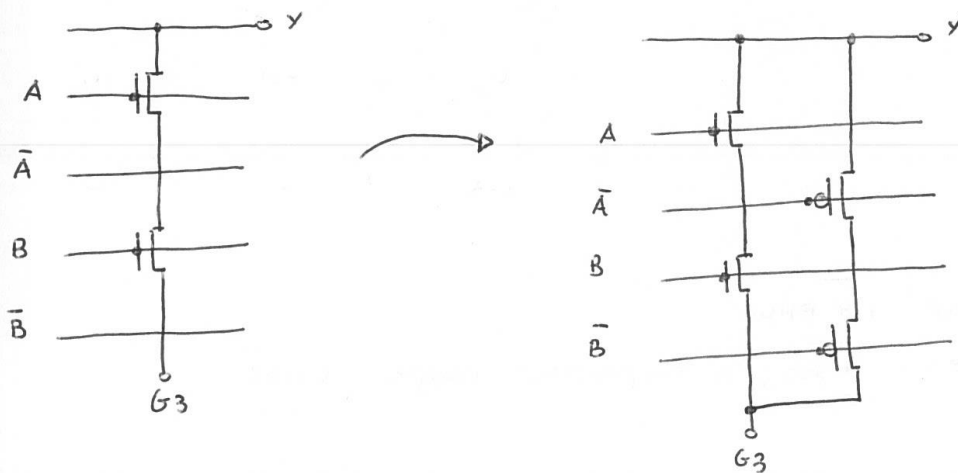
• Tempo di ritardo elevato

• Effetto crowbar

In pratica si ricorre per basse velocità di funzionamento e ridotto numero di ingressi.

Per eliminare il degrado dei livelli logici si sostituisce la logica Pass Transistor con la seguente

Passo Gate:

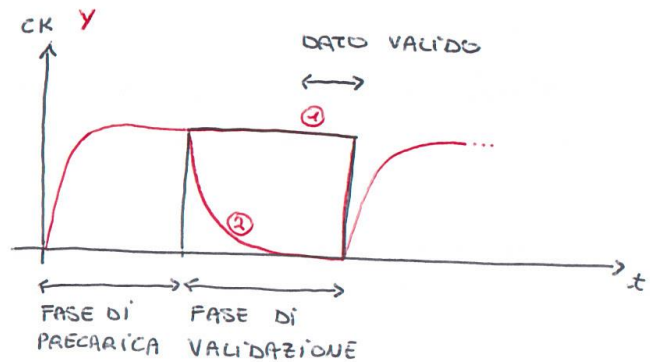
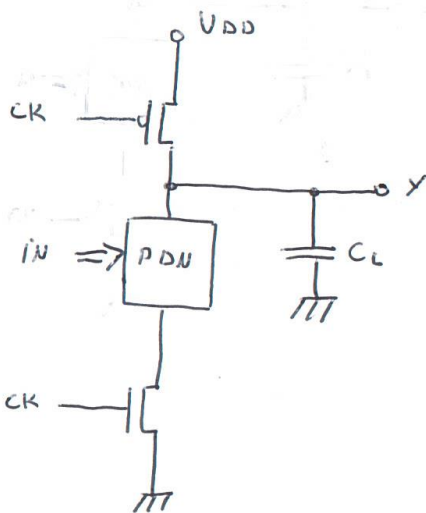


CELLE LOGICHE IN LOGICA DINAMICA

Il segnale logico di uscita è mantenuto su una capacità parassita.

Ci sono dei limiti temporali per cui il dato può essere considerato valido dettati da un segnale di sincronismo.

LOGICA DYNAMIC



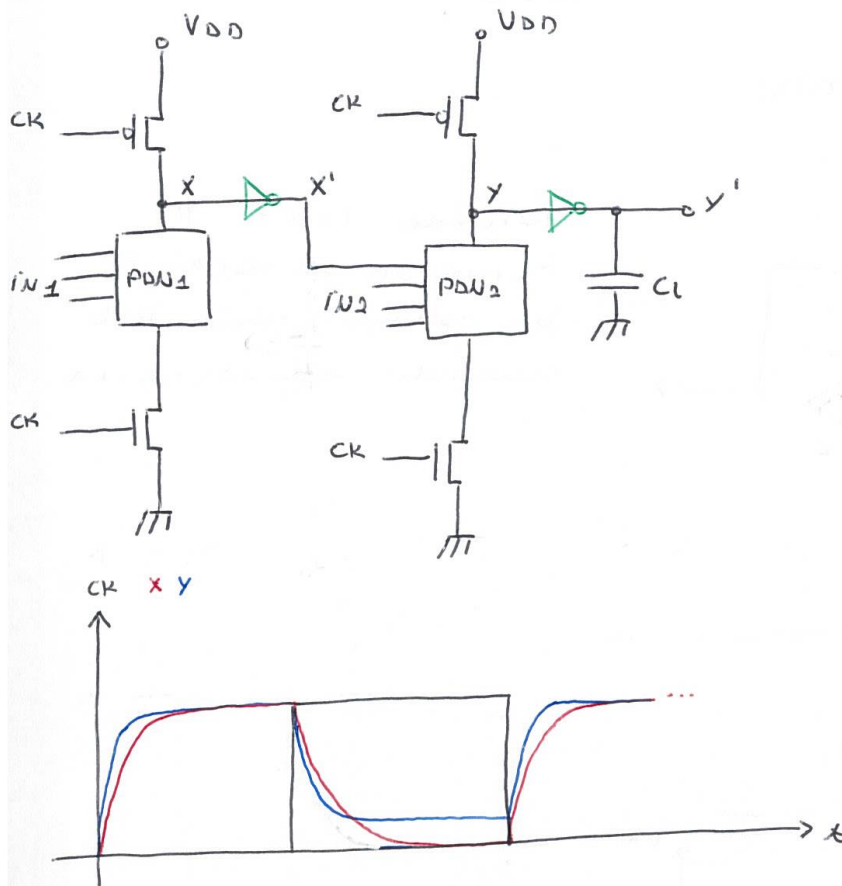
Vantaggi:

- 1 NMOS + 1 NMOS + 1 PMOS
- C_m ridotte: solo la capacità degli NMOS

Svantaggi:

- Il dato valido solo per frazioni del semiperiodo in cui CK = 1.

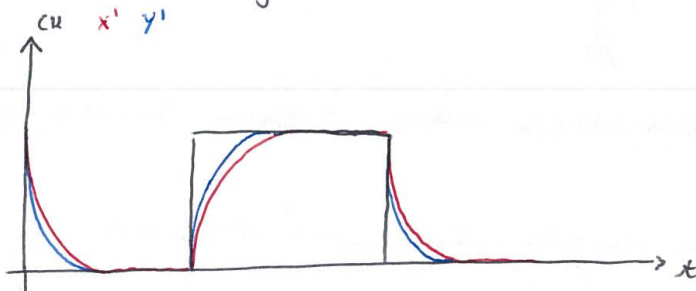
LOGICA NORINO IN CASCATA



La causa della propagazione ad 1, PDN2 lavora come se avesse un ingresso zero e supponiamo che la logica imponga sempre $Y = \phi$.

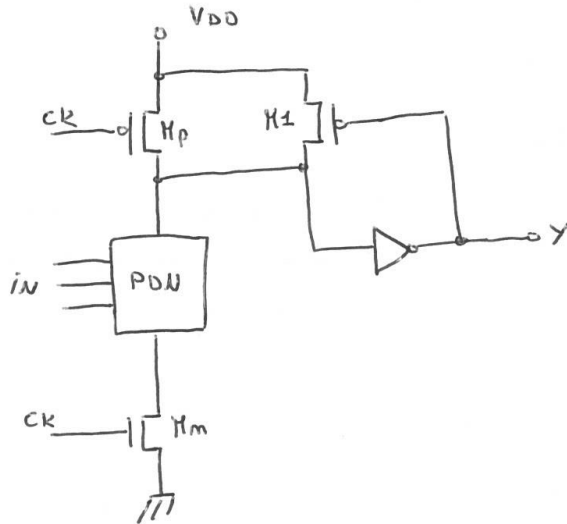
In realtà $X = \phi$ che impone $Y = 1$ (sempre per supposizione) $\rightarrow$ c'è una scarica parziale dell'uscita.

Ulteriore una fase di propagazione a ϕ .



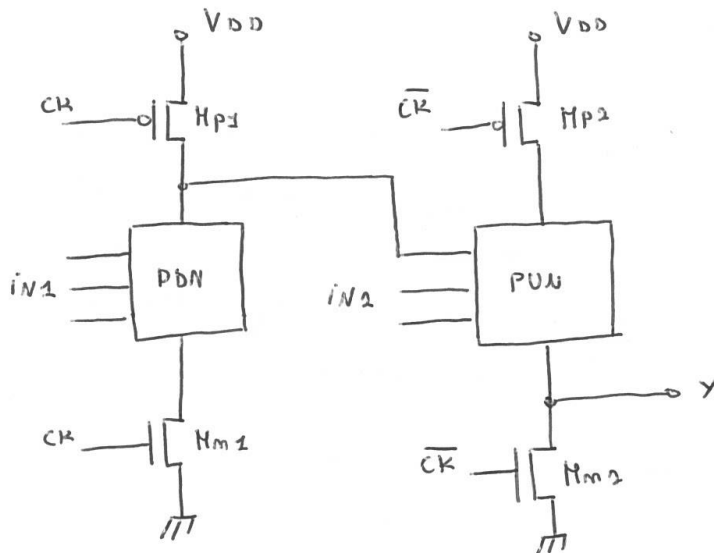
È possibile realizzare automaticamente funzioni non negaste.

LOGICA DOMINIO STATICA



Quando $CK = 1$, M_1 compensa le variazioni di resistenza dello PUN quando è interdetta.

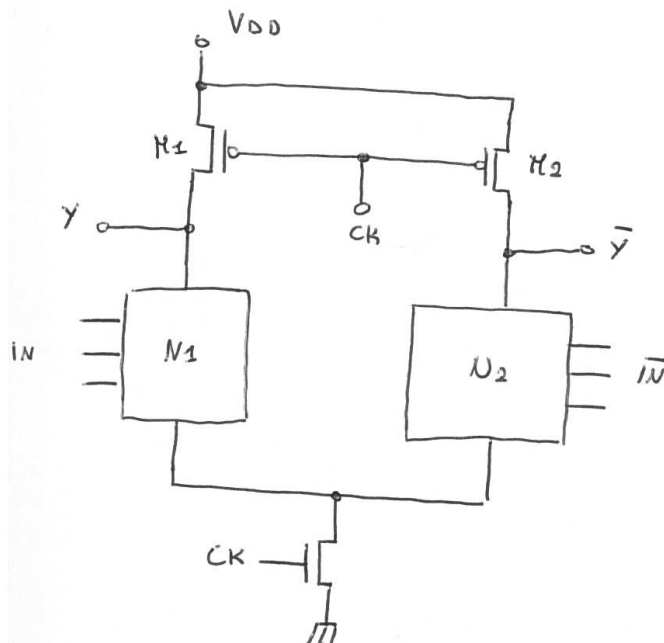
LOGICA DOMINIO P-N



Da un'architettura in deterministica allo stile dominio P-N.

Permette di realizzare anche funzioni negaste.

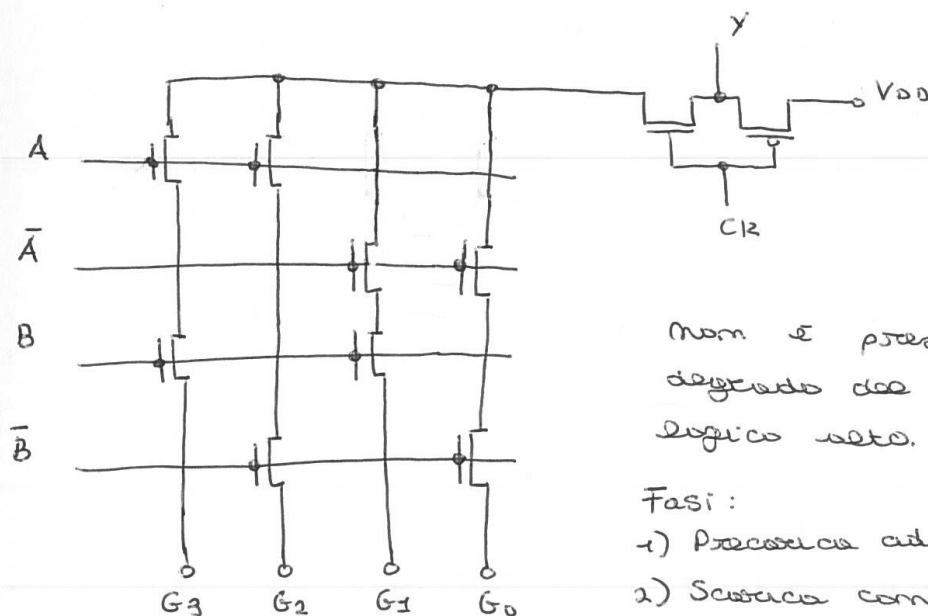
DUAL CASCADE VOLTAGE SWITCH LOGIC DINAMICA



$CK = \phi$ il sistema è in precarica con $Y = 1$.

La precarica avviene a livello logico alto, per cui il collegamento in cascata è consentito con l'inserimento di un invertitore.

PASS TRANSISTOR LOGIC DINAMICA



Non è presente il degrado del livello logico alto.

Fasi:

- 1) Precarica ad 1;
- 2) Selezione condizionata;

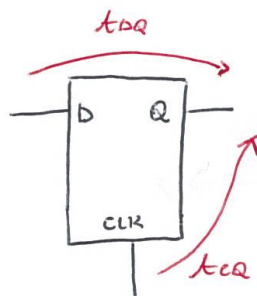
nel caso non sia prevista programmazione, si possono eliminare tutti i termi $G_i = V_{DD}$.

PROBLEMI DI TEMPORIZZAZIONE

LATCH

Sono dispositivi sensibili al livello (LOW or HIGH).

Quando un latch si trova in modalità trasparente ogni variazione dell'ingresso viene riportata in uscita.

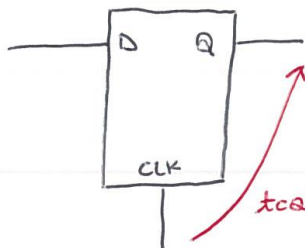


t_{DQ} : tempo che intercorre tra la variazione di D e l'uscita Q stabile.

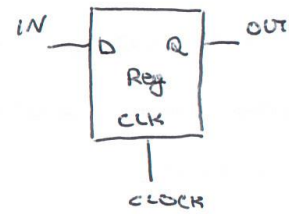
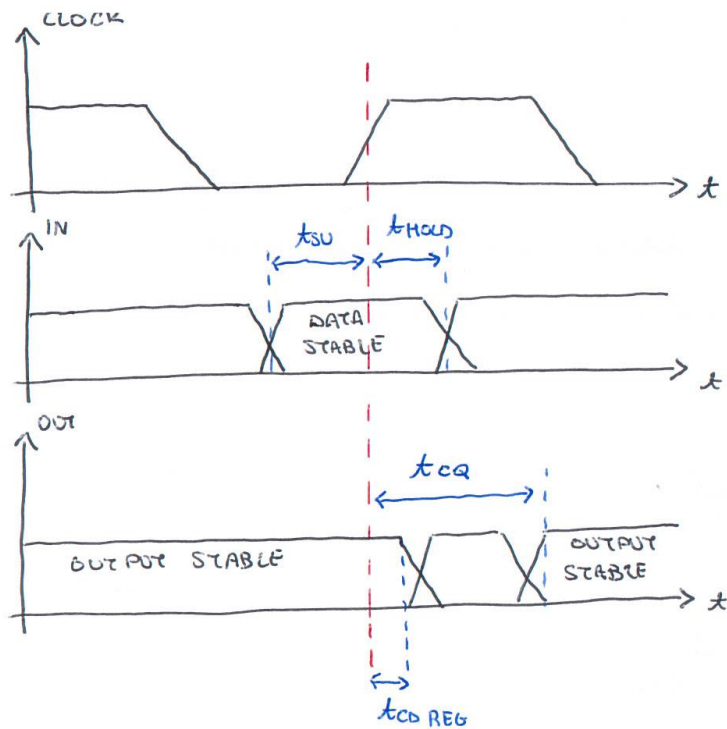
t_{CQ} : tempo che intercorre tra la variazione di CLK e l'uscita Q stabile.

FLIP - FLOP

Sono dispositivi sensibili al fronte di variazione di un segnale di riferimento.



t_{CQ} : tempo che intercorre tra la variazione di CLK e l'uscita Q stabile.

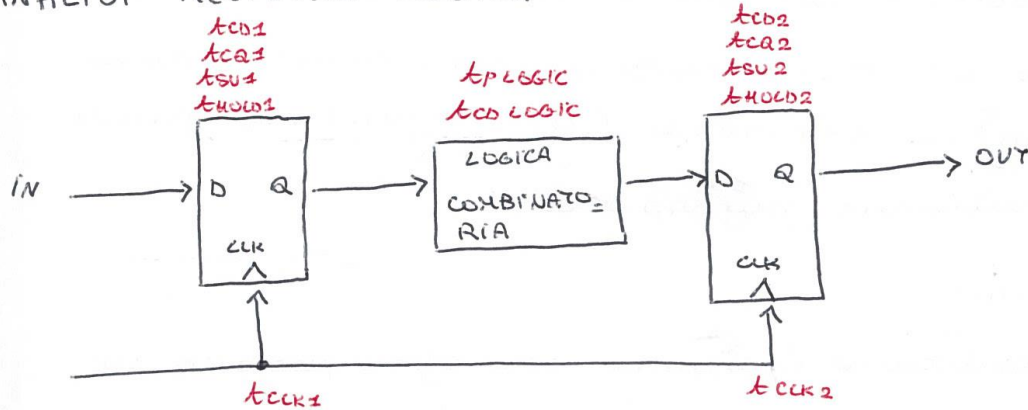


t_{SU} : Set Up: tempo di mantenimento di IN stabile prima della variazione di CLK

t_{HOLD} : tempo di mantenimento di IN stabile dopo la variazione di CLK

$t_{CO REG}$: Contamination Delay: tempo che intercorre tra la variazione di CLK e la variazione dell'uscita Q.

ANALISI REGISTRO - LOGICA - REGISTRO



$t_{PLLOGIC}$: tempo di propagazione del segnale
 nella logica $\rightarrow$ tempo tra variazione
 ingresso e uscita stabile.

$$T = \frac{1}{f_{CLK}} \quad , \quad T \geq t_{CQ1} + \underset{\substack{\uparrow \\ t_{PLLOGIC} = \max \begin{cases} t_{PLH} \\ t_{PLL} \end{cases}}}{t_{PLLOGIC}} + t_{SU2}$$

$$t_{HOLD2} \leq t_{CD LOGIC} + t_{CD1}$$

$t_{CD LOGIC}$: tempo tra variazione ingresso e
 variazione uscita.

Formule valide sotto condizioni ideali per cui
 risulta: $t_{CLK1} = t_{CLK2}$.

Analisi sotto condizioni Reali: $t_{clk1} \neq t_{clk2}$

Le analisi di clock subisce variazioni spaziali (CLOCK SKEW) e temporali (CLOCK JITTER) nei fronti di commutazione significative.

CLOCK SKEW

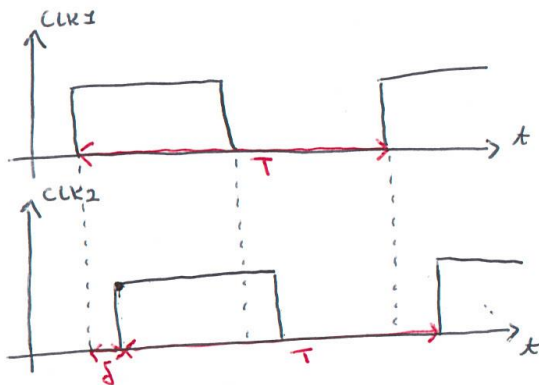
È una variazione costante per ogni periodo di clock.

Sorgenti:

- Variazioni dei valori nominale nella costruzione dei clock driver.
- Variazioni di temperatura e di alimentazione.

CLOCK SKEW POSITIVO

Clock e dati scattano nella stessa direzione.



$$T + \delta \geq t_{cq1} + t_{plogic} + t_{su0}$$

$$\rightarrow T \geq t_{cq1} + t_{plogic} + t_{su0} - \delta$$

$$t_{hold2} + \delta \leq t_{cdlogic} + t_{cd1}$$

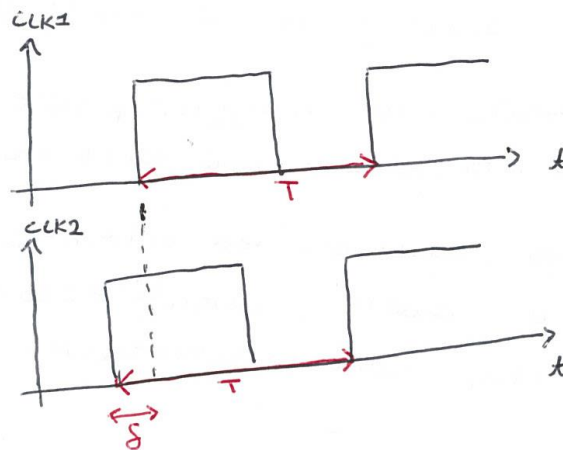
$$\rightarrow t_{hold2} \leq t_{cdlogic} + t_{cd1} - \delta$$

Si può migliorare le prestazioni del circuito ma rende più difficile rispettare t_{hold2} .

⇒ RACE CONDITIONS più stringenti

CLOCK SKEW NEGATIVO

Clock e data scendono in direzione opposta.



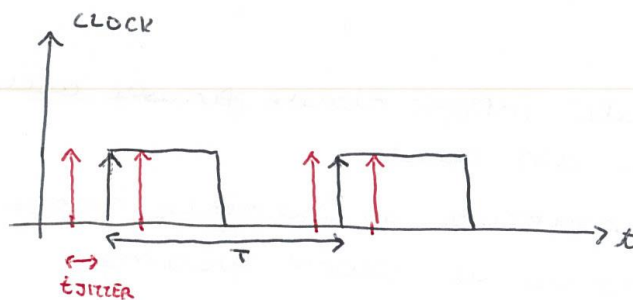
$$T \geq t_{cq1} + t_{plogic} + t_{su2} - \delta$$

$$t_{hold2} \leq t_{co logic} + t_{cd1} - \delta$$

$\delta < \phi$ peggiora le prestazioni del circuito ma rende più facile rispettare t_{hold2} .
 $\Rightarrow$ RACE CONDITIONS più elusive.

CLOCK JITTER

È una variazione di tempo per ogni ciclo di clock.



$$T: T - 1 t_{jitter} \geq t_{cq1} + t_{plogic} + t_{su2}$$

$$\rightarrow T \geq t_{cq1} + t_{plogic} + t_{su2} + 1 t_{jitter}$$

$$t_{hold2}: t_{hold2} + 1 t_{jitter} \leq t_{co logic} + t_{cd1}$$

$$\rightarrow t_{hold2} \leq t_{co logic} + t_{cd1} - 1 t_{jitter}$$

RETE DI DISTRIBUZIONE DEL CLOCK

È necessario progettare una clock network in grado di minimizzare clock skew e clock jitter.

Negli High-speed processor, la maggior parte della potenza dinamica è consumata nel clock network.

Clock gating: tecnica per ridurre la potenza dissipata nella linea di clock, disabilitando il clock per aree del chip non utilizzate.

TECNICHE PER LA DISTRIBUZIONE DEL CLOCK

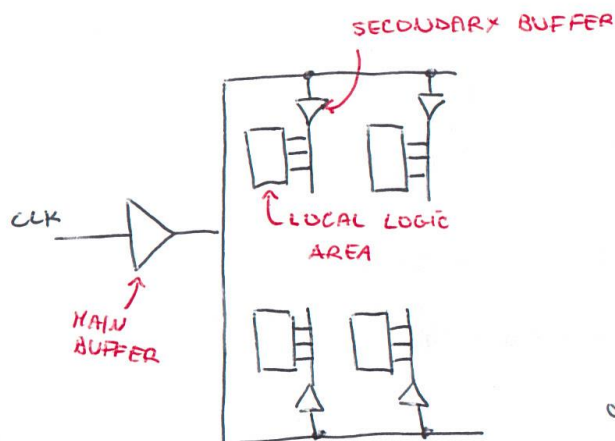
BALANCED PATHS (H-TREE NETWORK)

È in grado di eliminare lo skew clock ma potrebbe essere necessario più area di clock per propagare il segnale a tutti i linee del chip.

CLOCK GRIDS

Tipicamente utilizzato negli array di logica dove la rete di distribuzione del clock.

È in grado di minimizzare il ritardo assoluto e di rendere più semplice il clock gating.



Secondary Buffer usano le porte di clock locali da queste gate, riducendo l'effetto carico ed amplificando il segnale di clock.

Lo skew clock è presente solo nella LOCAL LOGIC AREA. (42)

Ridurre Clock Skew e Clock Jitter:

- usare una rete per la distribuzione del clock bilanciata.
- Routing data e clock in direzioni opposte in modo da eliminare RACE CONDITIONS.
- Routing clock per ridurre potenza dinamica dissipata.

Usare delle linee di protezione accanto al segnale di clock in modo da ridurre accoppiamenti capacitivi.

- Eliminare il rumore nelle linee di alimentazione.

PROGETTAZIONE LOW POWER

Perché optare per una progettazione LOW POWER:

- Costo del Package;
- Power supply rail design;
- Costi di raffreddamento del chip;
- Efficienza del sistema ed affidabilità;
- Durata della batteria;

CMOS ENERGY AND POWER EQUATIONS

$$P = P_D + P_{SC} + P_S$$

$$P_D = C_L \cdot V_{DD}^2 \cdot f_{\phi \rightarrow 1} \quad (\text{Potenza Dinamica})$$

$$P_{SC} = t_{SC} V_{DD} I_{PEAK} \cdot f_{\phi \rightarrow 1} \quad (\text{Potenza di corto circuito})$$

$$P_S = V_{DD} \cdot I_{LEAKAGE} \quad (\text{Potenza Statica})$$

$$f_{\phi \rightarrow 1} = P_{\phi \rightarrow 1} \cdot f_{CLOCK}$$

$P_{\phi \rightarrow 1}$: activity factor $\rightarrow$ probabilità di una commutazione $\phi \rightarrow 1$.

$$P_D = C_L \cdot V_{DD}^2 \cdot P_{B \rightarrow 1} \cdot f_{CLOCK}$$

$$C_L = FAN\ OUT \cdot C_T$$

P_D dipende da :

- FAN OUT (dispositivi più grandi dissipano di più)
- Tensione di alimentazione
- Velocità del circuito
- Frequenza del clock

$$P_{SC} = t_{SC} \cdot V_{DD} \cdot I_{PEAK} \cdot P_{B \rightarrow 1} \cdot f_{CLOCK}$$

La pendenza non infinita di una variazione dell'ingresso provoca la momentanea conduzione dei NMOS e PMOS con un conseguente picco tra V_{DD} e GND.

P_{SC} dipende da :

- Durata transizione
- Tensione di alimentazione
- Corrente di picco che dipende a sua volta dalla dimensione dei transistor.
- Velocità del circuito
- Frequenza del clock

$$P_S = V_{DD} \cdot I_{LEAKAGE}$$

Dipende da :

- Gate Leakage
- Drain Junction Leakage
- Corrente sottosoglia (FATTORE DOMINANTE)

$$V_{t1} < V_{t2} \Rightarrow I_{LEAKAGE1} > I_{LEAKAGE2}$$

(15)

Con il ridursi della tensione di alimentazione e la conseguente diminuzione della tensione di soglia, la corrente sottosoglia è diventata la principale fonte di consumo di potenza.

ANALISI DELLA SWITCHING ACTIVITY $P_{\phi \rightarrow 1}$

$P_D \propto P_{\phi \rightarrow 1}$

$P_{\phi \rightarrow 1}$ ha due componenti:

- Componente Statica: Funzione della topologia logica.
- Componente Dinamica: Funzione del comportamento temporale degli ingressi.

Componente Statica:

$$P_{\phi \rightarrow 1} = P_{\text{out}=\phi} \cdot P_{\text{out}=1} = P_{\text{out}=\phi} \cdot (1 - P_{\text{out}=\phi})$$

Tecniche per ridurre $P_{\phi \rightarrow 1}$:

- Logic Re Structuring: Cambiare la topologia logica di una rete per ridurre le transizioni $\phi \rightarrow 1$.

Per input random, l'implementazione a catena ha una switching activity minore dell'implementazione ad albero;

- Input ordering: Si cerca di porre segnali con alta transition rate;

Componente Dinamica:

A causa del ritardo di propagazione delle porte si presenteranno transizioni spurie (GLITCHES)

GLITCH: Un nodo è sottoposto a molteplici variazioni durante un ciclo di clock prima di raggiungere il valore logico corretto.

Tecniche per ridurre i glitch:

- esempio dovuto a mismatch nelle lunghezze dei path è necessario rendere uguali la durata dei timing path attraverso l'inserimento di logica, o la modifica della stessa.

TECNICHE PER RIDURRE IL CONSUMO DI POTENZA

MULTI V_{DD}

$$P_D \propto V_{DD}^2$$

$$t_{PHL}, t_{PLH} \propto V_{DD}^{-1}$$

Diminuendo la tensione di alimentazione costante di riduzione il consumo di potenza ma riduce le prestazioni della logica.

Si determina il comminuto critico e si usa la tensione di alimentazione maggiore per i transistor in questo percorso.

Per gli altri transistor si usa una tensione di alimentazione minore, in modo da rendere tutti i comminuti critici.

Con questa soluzione sono necessari dei buffer inverter quando un transistor con bassa alimentazione deve pilotare uno con alta alimentazione (STEP UP CONVERSION).

Convertitori non necessari nella STEP DOWN CONVERSION.

MULTI V_T

Ridurre V_T aumenta la corrente di leakage sotto soglia, ma allo stesso tempo migliora le performance del circuito

$$t_{PHL} \propto \frac{K \cdot C_L}{\beta_n (V_{DD} - V_{Tn})}$$

$$t_{PLH} \propto \frac{K \cdot C_L}{\beta_p (V_{DD} + |V_{Tp}|)}$$

Si determina il commutatore critico e si usa una tensione di soglia bassa per i transistor in questo percorso.

Per gli altri transistor si usa una V_T maggiore in modo da rendere tutti i commutatori critici.

Non sono necessari level converter.

BUS MULTIPLEXING

Il bus sono sorgente significativa di consumo di potenza per via della elevata switching activity e della grande capacità di carico.

→ Si procede a condensare i bus più lunghi con una tecnica di TIME MULTIPLEXING.

N.B.: Se le informazioni precedentemente contenute su 2 bus, riunite in un singolo bus grazie al time multiplexing, sono correlate, allora si ha un aumento della switching activity con un conseguente aumento di potenza dissipata.

RIDUZIONE DEI GLITCH TRAMITE PIPELINING

I glitch dipendono dalla profondità della logica progettata $\rightarrow$ più un dispositivo si trova all'interno della logica e più questo è soggetto a glitch.

Si procede a ridurre la profondità della logica aggiungendo registri di pipeline.

CLOCK GATING

È il metodo più utilizzato per ridurre i consumi sulla linea di clock e di unità funzionali.

Si procede a disattivare il clock per unità non funzionanti.

Problemi:

• unità Potenti

meccanismo logico per generare il segnale di disinibizione

- aumenta complessità
- aumentano consumi
- aumentano ritardi anche nella diffusione del segnale di clock.

DYNAMIC FREQUENCY SCALING (DFS)

DYNAMIC VOLTAGE SCALING (DVS)

Algoritmi software o logici che consento di ridurre run-time la tensione di alimentazione e la frequenza di clock in modo da ridurre le perdite per surriscaldamento la batteria.

RIASSUNTO :

Tecniche per ridurre potenza in fase di progettazione

- Riprogettamento logico (P_0)
- Riduzione V_{DD} (P_0)
- Sizing logico (P_0)
- Multi V_{DD} (P_0)
- Multi V_t (P_1)

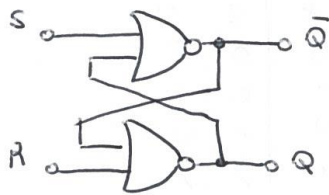
Tecniche per ridurre potenza run-time :

- Clock gating (P_0)
- DFS, DVS (P_0)

Per logiche di piccole dimensioni preferisco usare una tecnica multi V_t piuttosto che multi V_{DD} , in quanto non è necessario aggiungere i transistori di sneaker, che a sua volta consumano potenza.

FLIP-FLOP : SOLUZIONI STATICHE

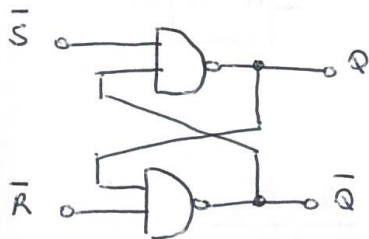
LATCH SR (NOR2 BASED)



S	R	Q	Q ¹
0	0	Q	Q ¹
0	1	0	1
1	0	1	0
1	1	/	/

8 MOS

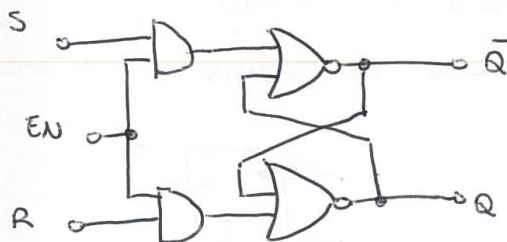
LATCH SR (NAND 2 BASED)



S ⁻	R ⁻	Q	Q ⁻
1	1	Q	Q ⁻
1	0	0	1
0	1	1	0
0	0	/	/

12 MOS

LATCH SR CON ENABLE (NOR2 BASED)

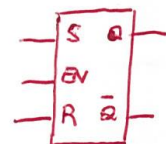


EN	S	R	Q	Q ¹
0	x	x	Q	Q ¹
1	0	0	Q	Q ¹
1	0	1	0	1
1	1	0	1	0
1	1	1	/	/

20 MOS

Soluzione ottimizzata:

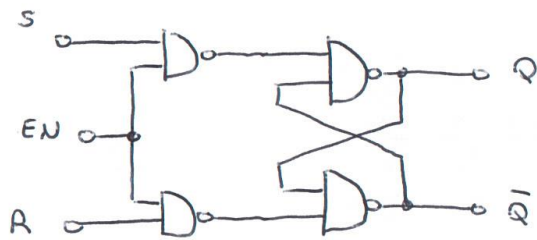
$$Y = \bar{Q} = \overline{S \cdot EN + Q}$$



Sono necessari 6 MOS per realizzare $\bar{Q}$ (3 PMN + 3 PMN)
e 6 MOS per realizzare Q (3 PMN + 3 PMN)

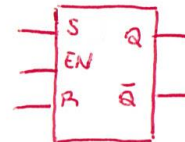
→ 12 MOS

LATCH SA CON ENABLE (NAND 2 BASED)

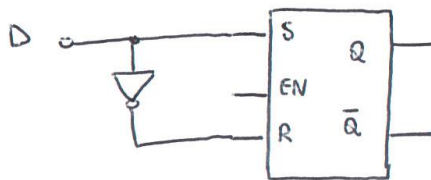


16 MOS

EN	S	R	Q	$\bar{Q}$
0	x	x	Q	$\bar{Q}$
1	0	0	Q	$\bar{Q}$
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1



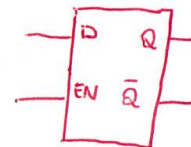
LATCH D



EN	D	Q	$\bar{Q}$
0	x	Q	$\bar{Q}$
1	0	0	1
0	1	1	0

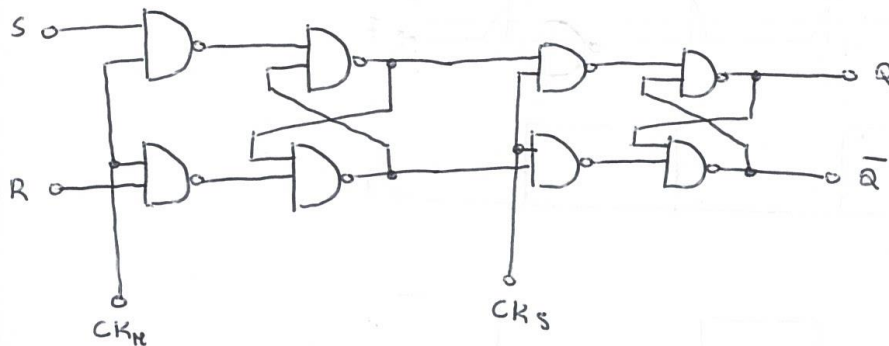
Se Realizzato con tecnologia NOR2 : 14 MOS

Se Realizzato con tecnologia NAND2 : 18 MOS



FLIP FLOP SR : LATCH SR EDGE TRIGGERED (NAND 2 BASED) IN ARCHITETTURA MASTER-SLAVE

NEGATIVE EDGE TRIGGERED

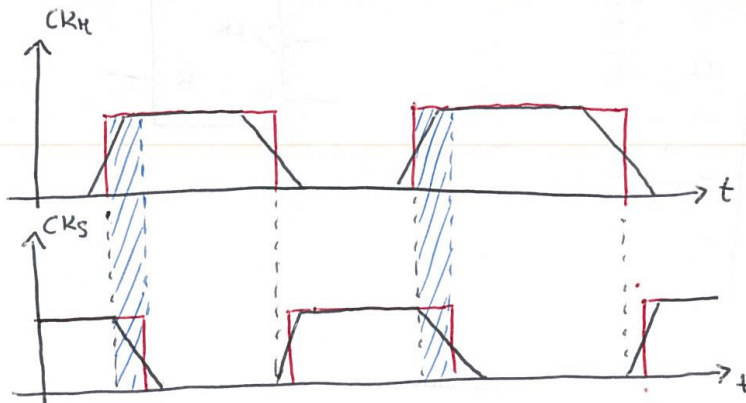


32 MOS

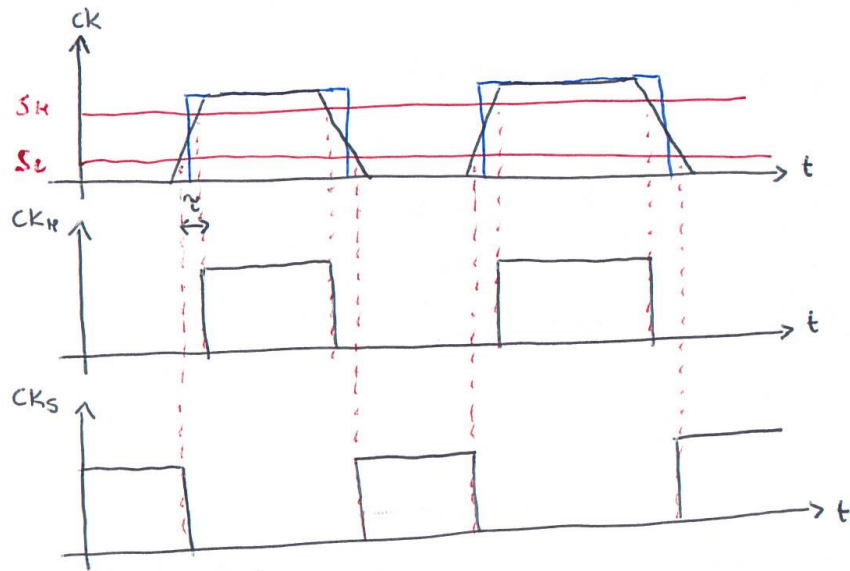
È necessario utilizzare un clock con fasi non sovrapposte.

$CK_H \rightarrow \neg CK_S$ non va bene!

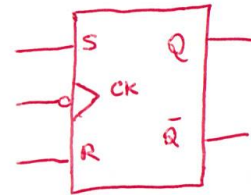
CK_H non può essere a livello alto insieme a CK_S .



uso un clock a due fasi non sovrapposte:
TECNICA DI GENERAZIONE A SOGLIA

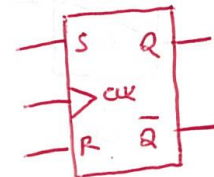


CLK	S	R	Q	$\bar{Q}$
0	x	x	Q	$\bar{Q}$
1	x	x	Q	$\bar{Q}$
↑	x	x	Q	$\bar{Q}$
↓	0	0	Q	$\bar{Q}$
↓	0	1	0	1
↓	1	0	1	0
↓	1	1	1	1

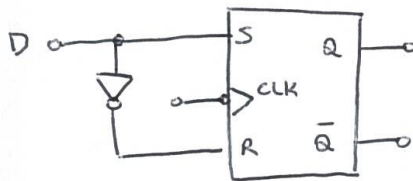


FLIP FLOP SR POSITIVE EDGE TRIGGERED

CLK	S	R	Q	Q'
0	x	x	Q	Q'
1	x	x	Q	Q'
↓	x	x	Q	Q'
↑	0	0	Q	Q'
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	-	-

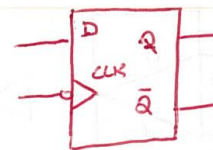


FLIP FLOP D NEGATIVE EDGE TRIGGERED (NAND2 BASED)

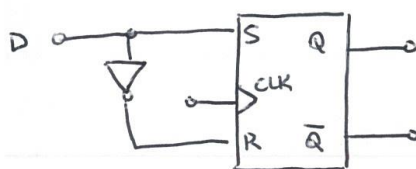


CLK	D	Q
0	x	Q
1	x	Q
↑	x	Q
↓	0	0
↓	1	1

34 MOS

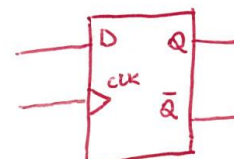


FLIP FLOP D POSITIVE EDGE TRIGGERED (NOR2 BASED)

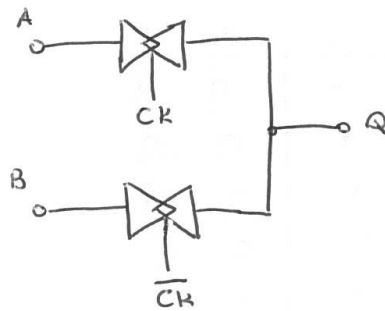
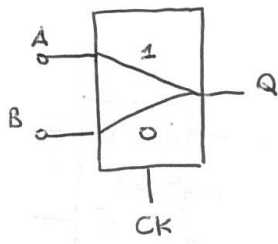


CLK	D	Q
0	x	Q
1	x	Q
↓	x	Q
↑	0	0
↑	1	1

26 MOS

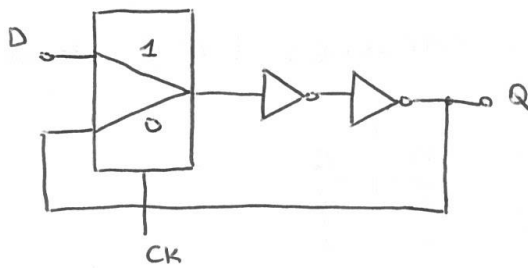


MUX A 2 VIE



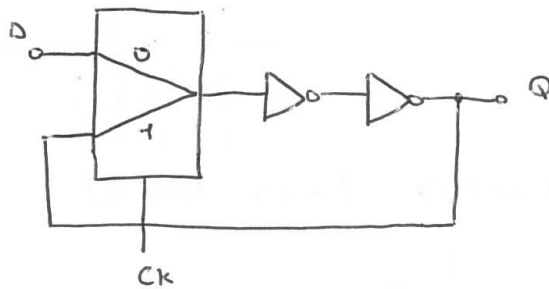
6 MOS

LATCH D POSITIVE



10 MOS

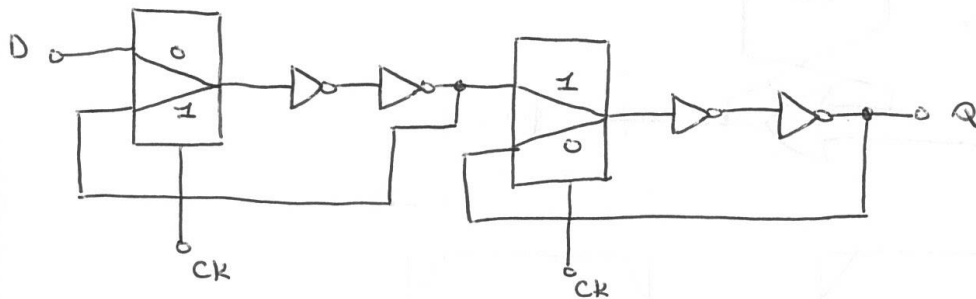
LATCH D NEGATIVE



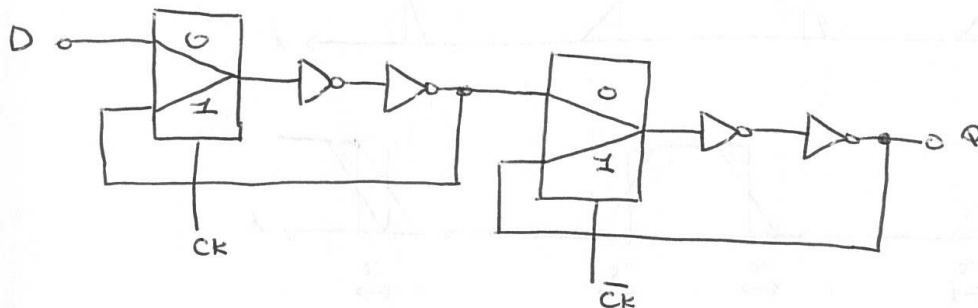
10 MOS

FLIP FLOP D POSITIVE EDGE TRIGGERED

ARCH. 1 : 20 MOS

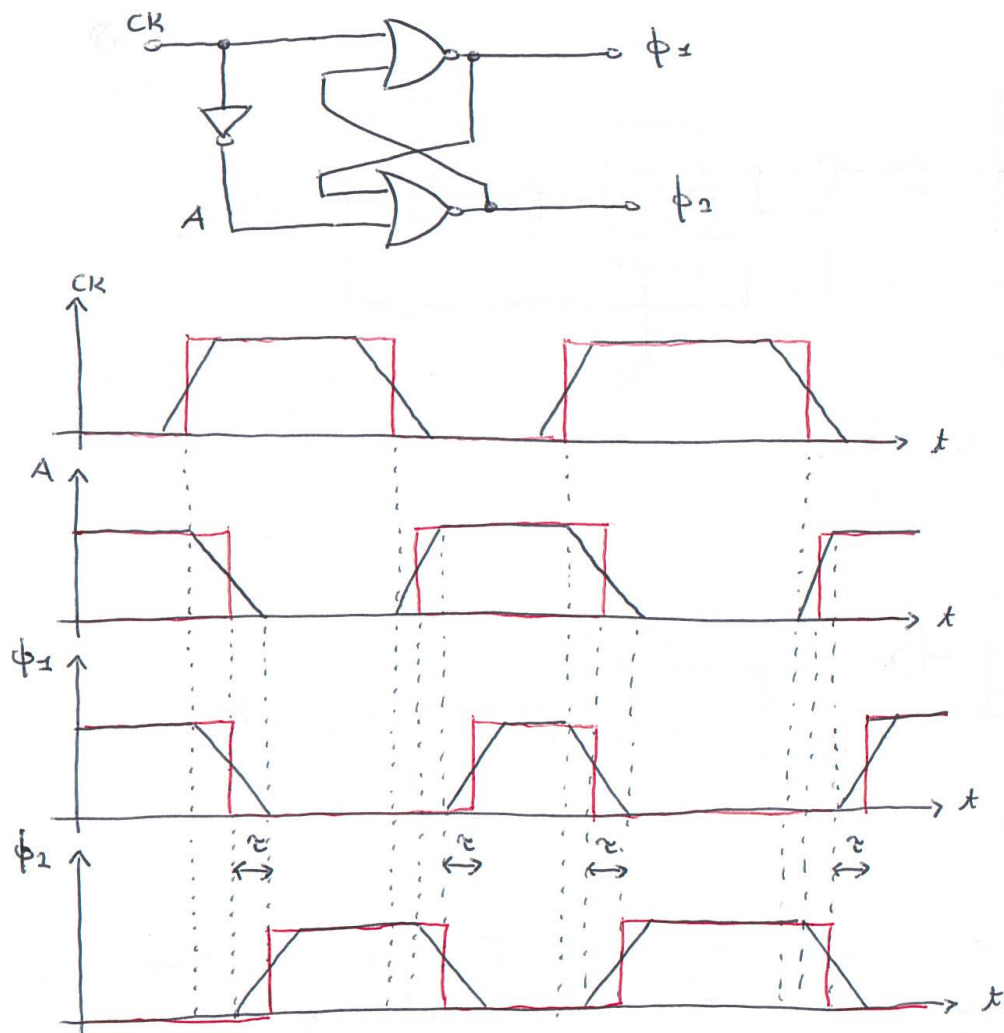


ARCH. 2 : 20 MOS

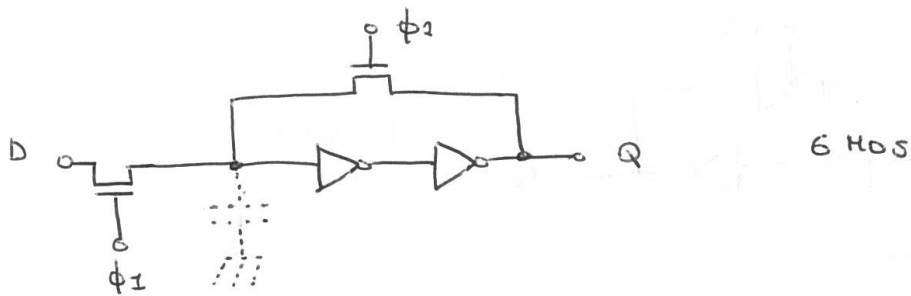


Bisogna assicurare che CK e $\overline{CK}$ non possano essere nullo contemporaneamente

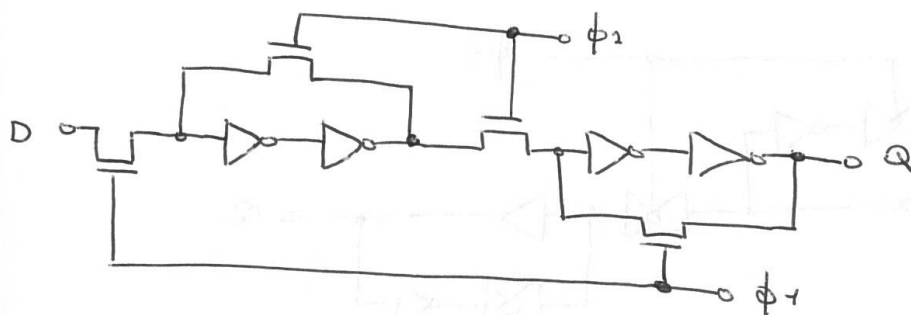
GENERATORE DI CLOCK A DUE FASI TECNICA DI GENERAZIONE A SOGLIA



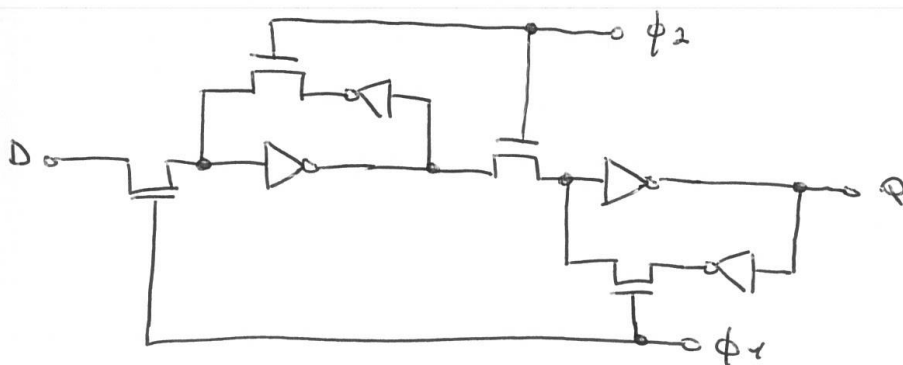
LATCH D CON PASS TRANSISTOR



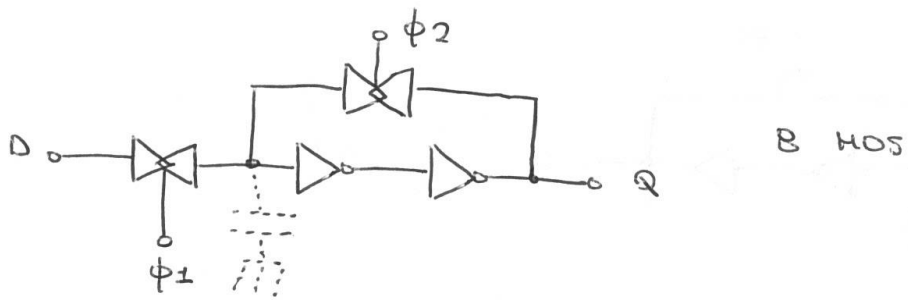
FLIP FLOP D CON PASS TRANSISTOR



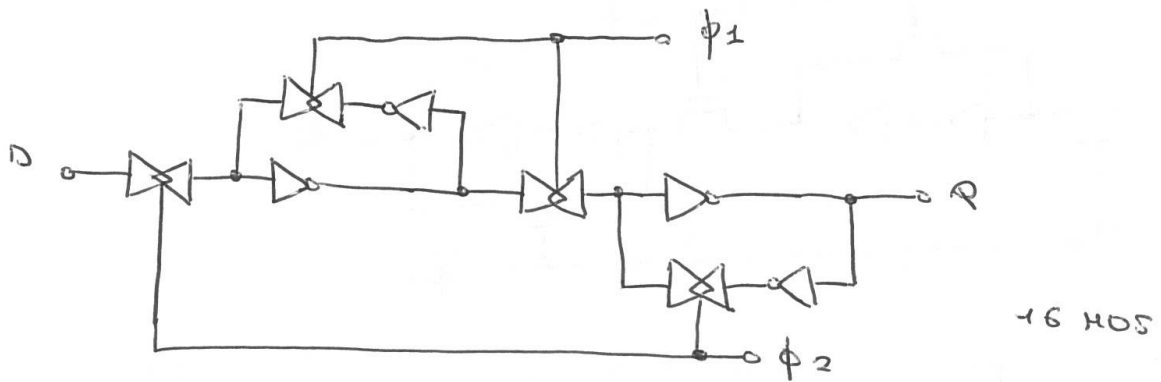
Soluzione per minimizzare il ritardo:



LATCH D CON PASS GATE

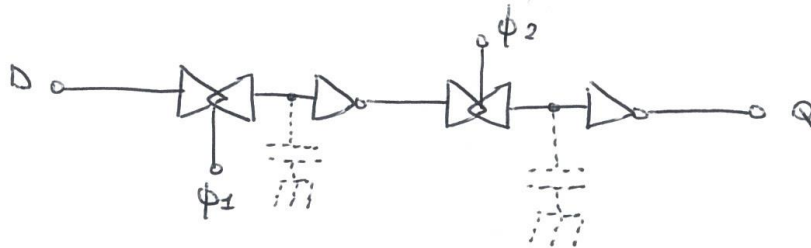


FLIP FLOP D CON PASS GATE



FLIP FLOP : SOLUZIONI DINAMICHE

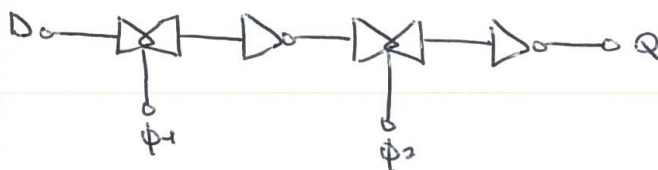
FLIP FLOP D



Nammo enitati i percorsi teorapanti :

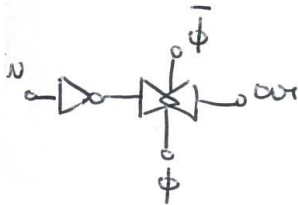
- $\phi_1 = \phi$, $\phi_2 = \phi$;
- $\phi_1 = 1$, $\phi_2 = 1$;
- $\phi_1 = \phi$, $\phi_2 = 1$;
- $\phi_1 = 1$, $\phi_2 = \phi$;

CELLA BASE SHIFT REGISTER

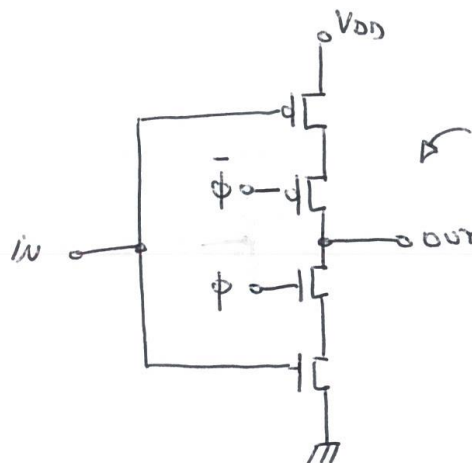


È un flip flop D dinamico.

Ottimizzazione:



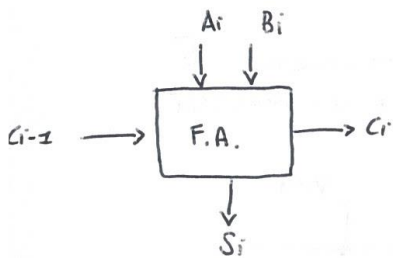
=>



Invertitore 3 state.

LOGICA DIGITALE

1 BIT FULL ADDER



A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

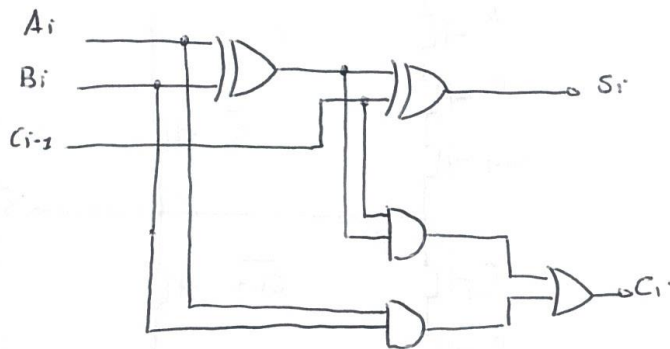
$$C_i = A_i B_i C_{i-1} + A_i B_i \overline{C_{i-1}} + A_i \overline{B_i} C_{i-1} + \overline{A_i} B_i C_{i-1} \\ = A_i B_i + C_{i-1} (A_i \oplus B_i)$$

$$G_i = A_i B_i = 1$$

→ CARRY GENERATE

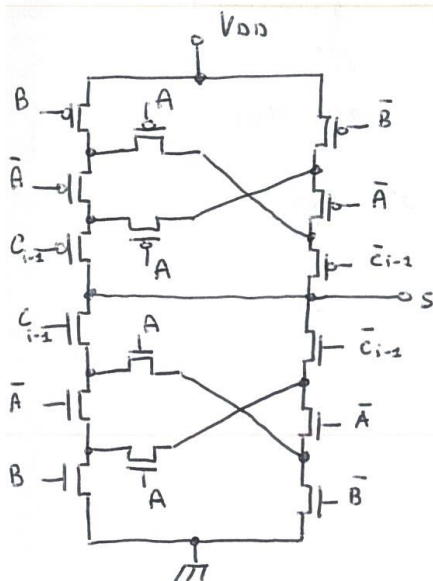
$$P_i = A_i \oplus B_i = 1$$

→ CARRY PROPAGATE



Realizzazione S:

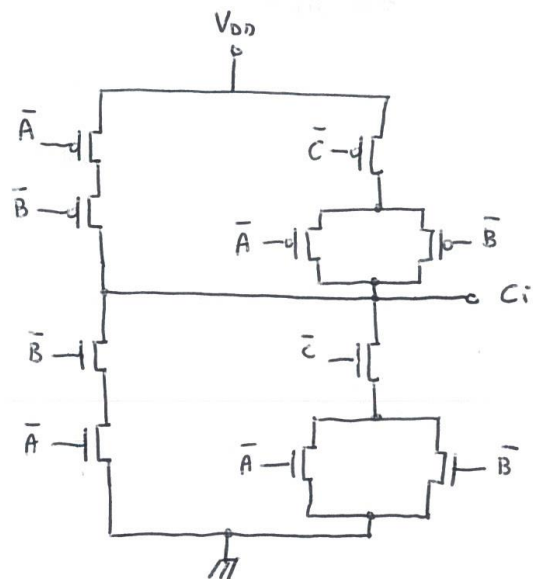
COMPLEX GATE



16 MOS

Realizzazione C:

COMPLEX GATE



10 + 2*3 = 16 MOS

(63)

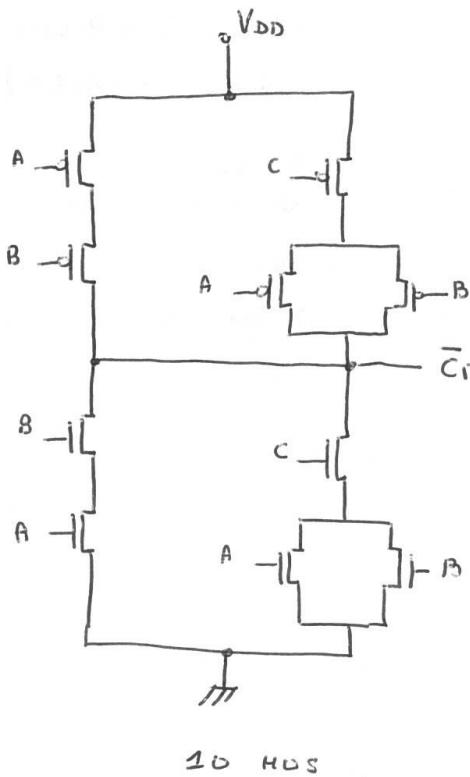
tot : 32 MOS

Ottimizzazione: S può essere ricavata da C

$$S = \overline{C_i} \cdot (\overline{A \cdot B \cdot C}) + ABC = \overline{C_i} \cdot (A+B+C) + ABC$$

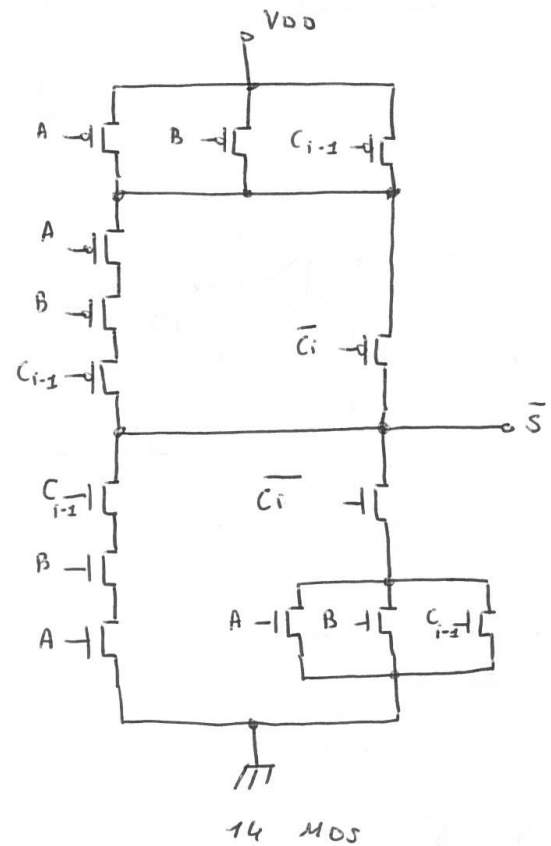
Realizzazione $\overline{C_i}$.

COMPLEX GATE



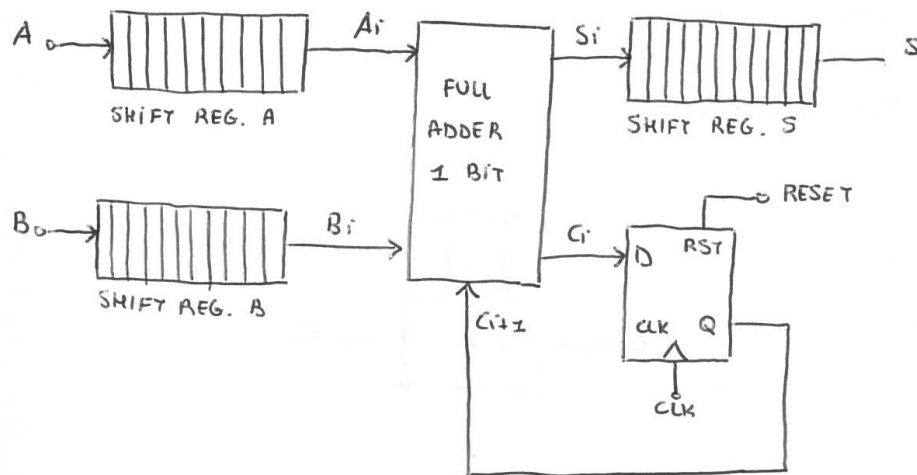
Realizzazione $\overline{S}$:

CMOS-COMPLEMENTARE



$$\text{tot: } 10 + 14 + 2 \cdot 2 = 28 \text{ MOS}$$

SERIAL ADDER



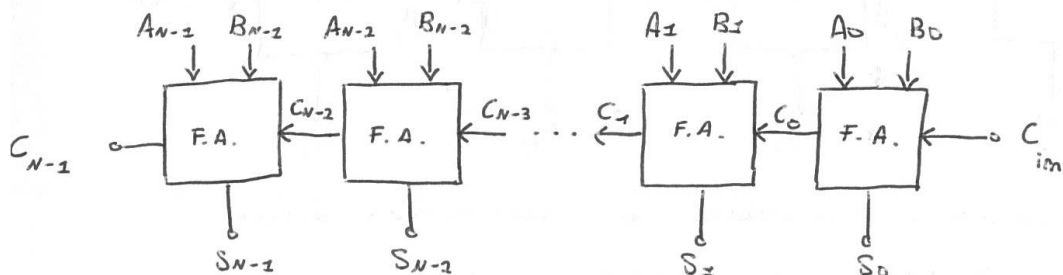
Gli Shift Register sono ad N Bit.

$$t_{TOT} = N \cdot T_{CLK}$$

Bassa complessità logica ma lungo tempo di esecuzione.

PARALLEL ADDER

RIPPLE CARRY ADDER (CARRY PROPAGATE ADDER)



t_c : Ritardo propagazione del carry

$$\rightarrow t_c > t_s$$

t_s : Ritardo propagazione della somma

$$t_{TOT} = N \cdot t_c$$

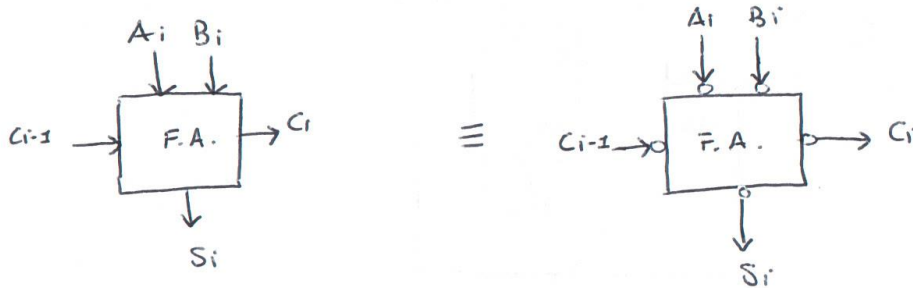
(65)

È più veloce di un Serial Adder ed ha il vantaggio

di avere un semplice sistema modulare.

Ottimizzazione eseguita per la minimizzazione del cammino critico:

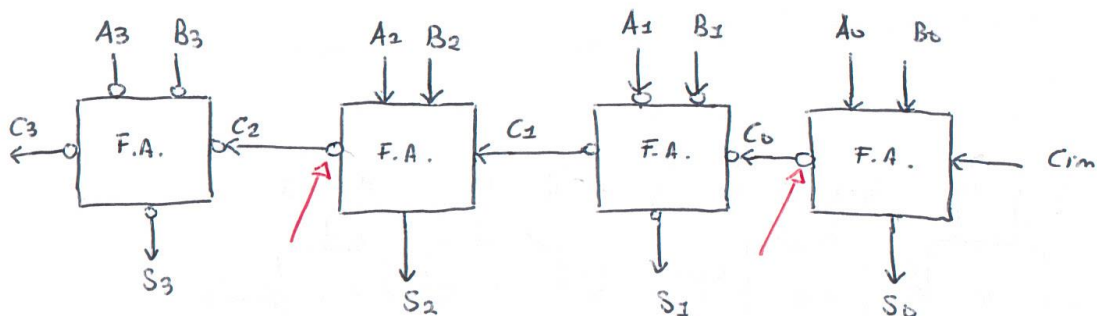
Proprietà di Inversione:



$$S(A_i, B_i, C_{i-1}) = \bar{S}(\bar{A}_i, \bar{B}_i, \bar{C}_{i-1})$$

$$C_i(A_i, B_i, C_{i-1}) = \bar{C}_i(\bar{A}_i, \bar{B}_i, \bar{C}_{i-1})$$

Per $N=4$:



Ho soppressato 2 inverters nel cammino critico riducendo il tempo di propagazione.

CARRY LOOK AHEAD ADDER

Generazione del carry:

$$C_i = A_i B_i C_{i-1} + A_i B_i \overline{C_{i-1}} + A_i \overline{B_i} C_{i-1} + \overline{A_i} B_i C_{i-1} = A_i B_i + C_{i-1} (A_i \oplus B_i)$$

$$G_i = A_i B_i = 1 \quad \text{Carry Generate} \quad (C_i = 1)$$

$$P_i = A_i \oplus B_i = 1 \quad \text{Carry Propagate} \quad (C_i = C_{i-1})$$

nella generazione del carry posso fare questa semplificazione:

$$P_i = A_i \oplus B_i \Rightarrow P_i = A_i + B_i$$

nel caso $A_i = 1, B_i = 1$ si deve generare facendo risultare comunque C_i corretto.

$$C_0 = G_0 + P_0 C_{in}$$

$$C_1 = G_1 + P_1 C_0 = G_1 + P_1 G_0 + P_1 P_0 C_{in}$$

$$C_2 = G_2 + P_2 C_1 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{in}$$

$$= G_3 + P_3 C_2 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{in} \dots$$

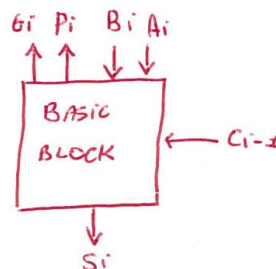
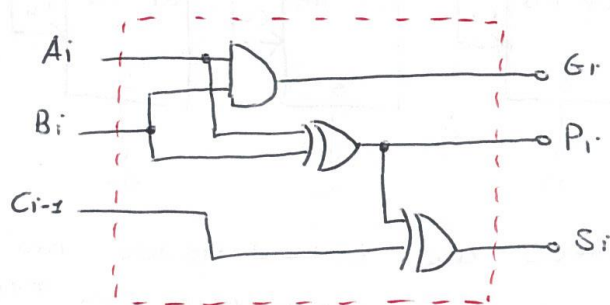
tutti i simboli di carry sono generati allo stesso tempo.

Blocco Base:

$$G_i = A_i B_i$$

$$P_i = A_i \oplus B_i$$

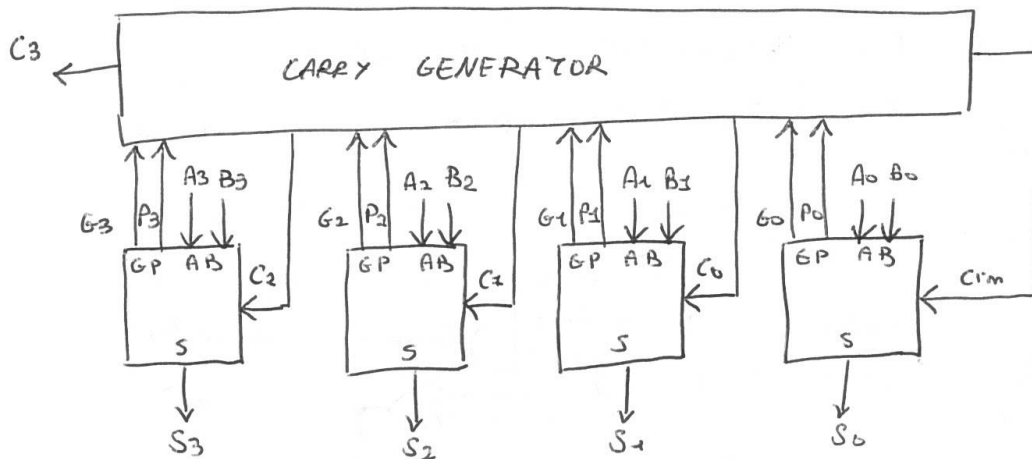
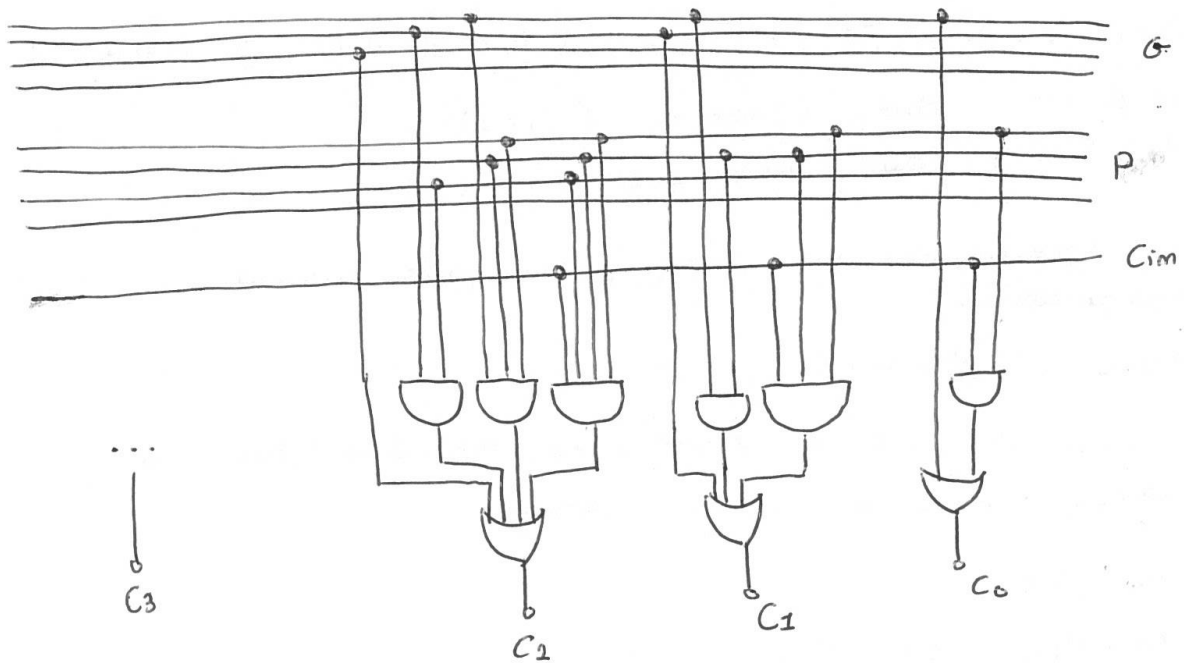
$$S_i = P_i \oplus C_{i-1}$$



(6)

Per $N=4$:

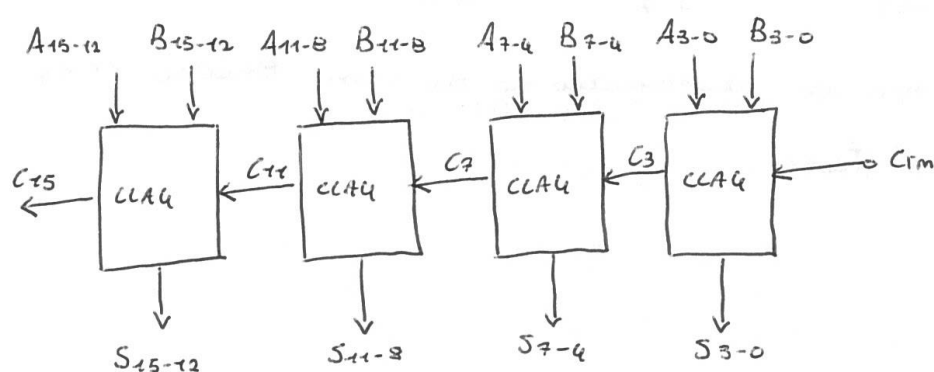
generazione del carry :



È più veloce del Serial Adder ma richiede dei bus
grandi e delle porte AND con FAN IN via via
crescente.

Quando $N > 4$, per evitare di usare porte AND con più di 5 ingressi, e quasi ridimensionando notevolmente la velocità del circuito, si usano altre soluzioni basate sul CARRY LOOK AHEAD ADDER.

CLA RIPPLE CARRY SOLUTION



T_{CLA4} : tempo di propagazione di un blocco CLA4

$$T_{tot} = \frac{N}{4} \cdot t_{CLA4}$$

CLA SOLUZIONE GERARCHICA

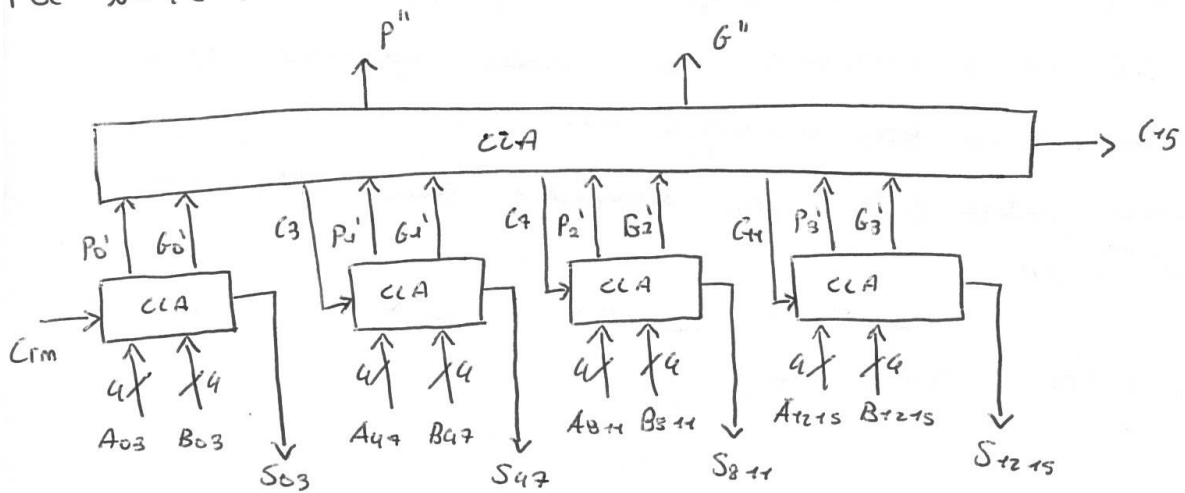
Si estraggono dal Carry Generator del Carry Look Ahead Adder due segnali G' e P' :

$$\begin{aligned} C_3 &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{in} \\ &= G' + P' C_{in} \end{aligned}$$

$$G' = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0$$

$$P' = P_3 P_2 P_1 P_0$$

Per $N = 16$:



T_{CLA4} : tempo di propagazione di un blocco CLA4

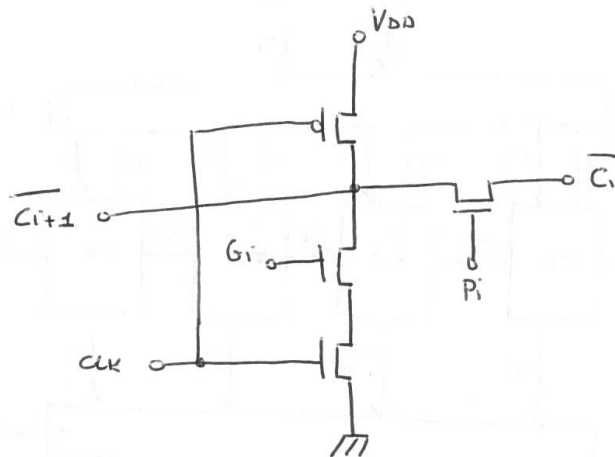
$$T_{tot} = \log_4(N) T_{CLA4}$$

MANCHESTER CARRY ADDER

Si realizza con degli interruttori controllati da A_i e G_i

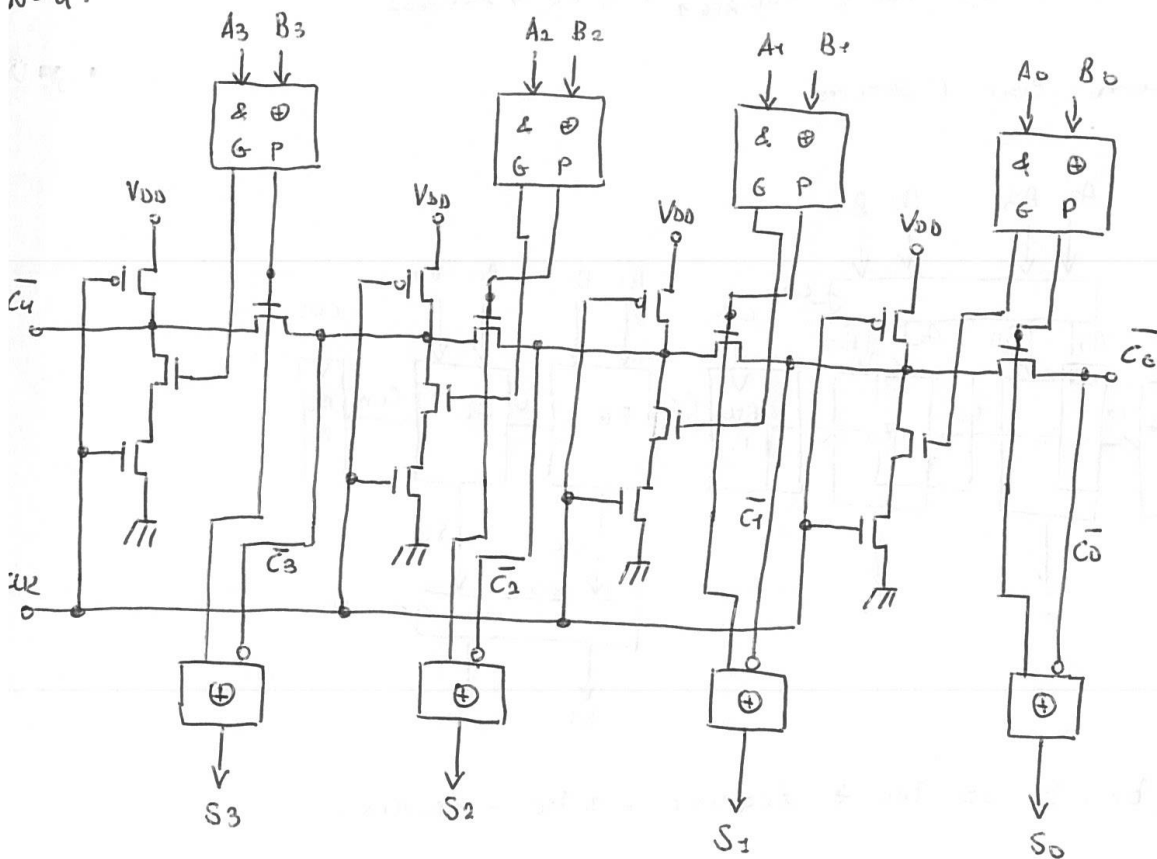
$$G_i = A_i B_i$$

$$P_i = A_i \oplus B_i$$



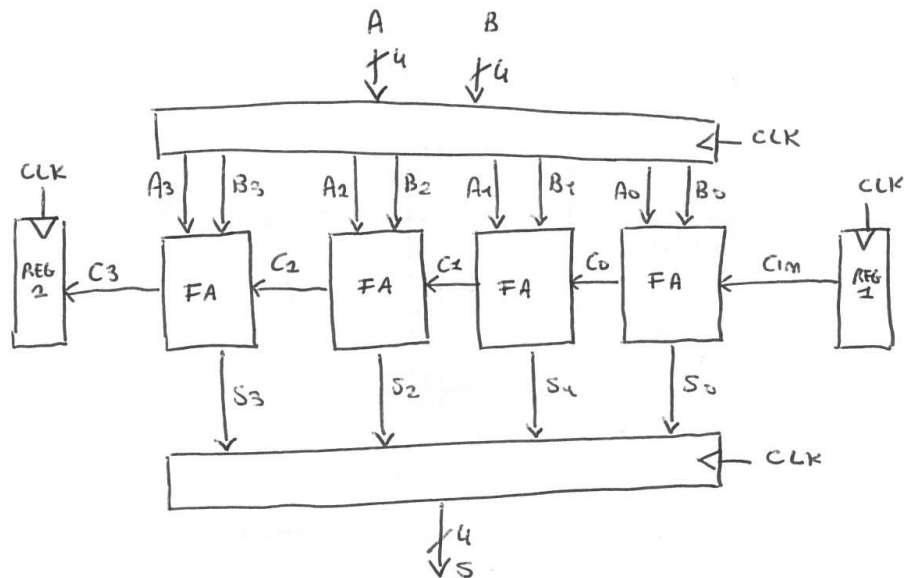
Propaga $\bar{C}_i \neq \text{non } C_i$ per non avere degradazione del livello logico visto $\rightarrow$ Propagazione senza $\text{ack} = 1$.

$N=4$:



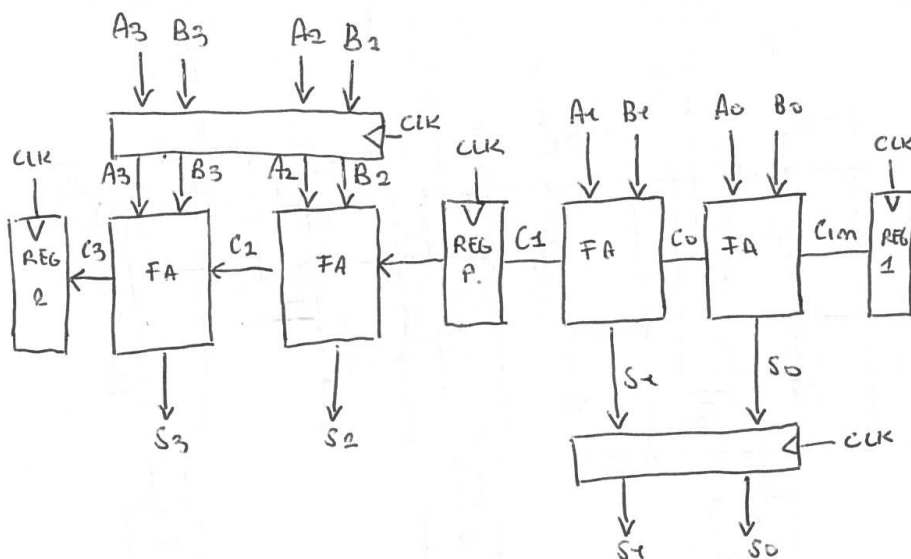
RIPPLE CARRY ADDER PIPELINED

Soluzione senza Pipeline:



$$t_c > t_s \Rightarrow T_{CK} \geq t_{CQ, REG1} + 4t_c + t_{SU, REG1}$$

Soluzione con Pipeline



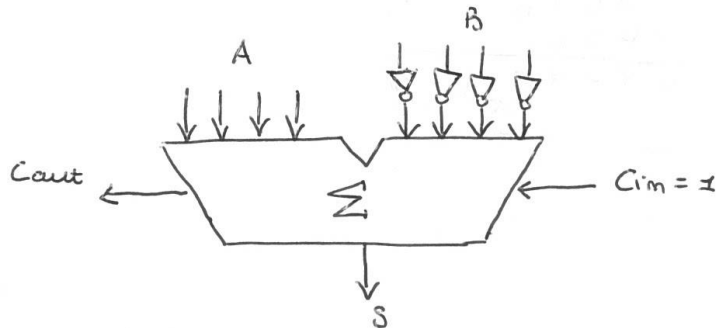
$$t_c > t_s \Rightarrow T_{CK} \geq t_{CQ, REG1} + 2t_c + t_{SU, REG1, P}$$

SOTTRATTORE

$$D = A - B = A + (-B)$$

$-B$: Complemento a 2 di $B \rightarrow$ negazione + 1.

$N=4$:



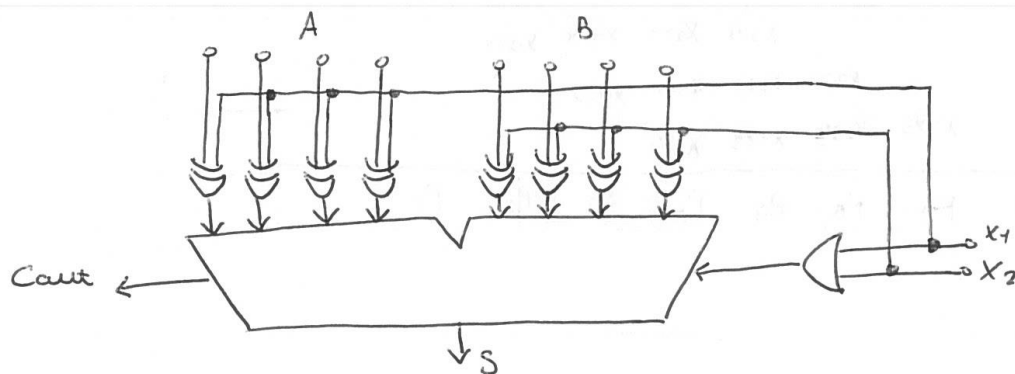
Utilizzo di un sommatore come: Sommatore - Sottrattore

A	x	U
0	0	0
0	1	1
1	0	1
1	1	0

$$\text{Se } x = 0 \Rightarrow U = A$$

$$\text{Se } x = 1 \Rightarrow U = \bar{A}$$

$N=4$:



X1	X2	S
0	0	A + B
0	1	A - B
1	0	B - A

MOLTIPLICATORE

Moltiplicazione tra interi: $P = A \cdot B$

$$A = 1010$$

$$B = 0110$$

$$P = 0111100$$

$$\begin{array}{r} 1010 \cdot \\ 0110 = \\ \hline 0000 \\ 1010 \\ 1010 \\ 0000 \\ \hline 0111100 \end{array}$$

$$X = \sum_{i=0}^{m-1} x_i 2^i$$

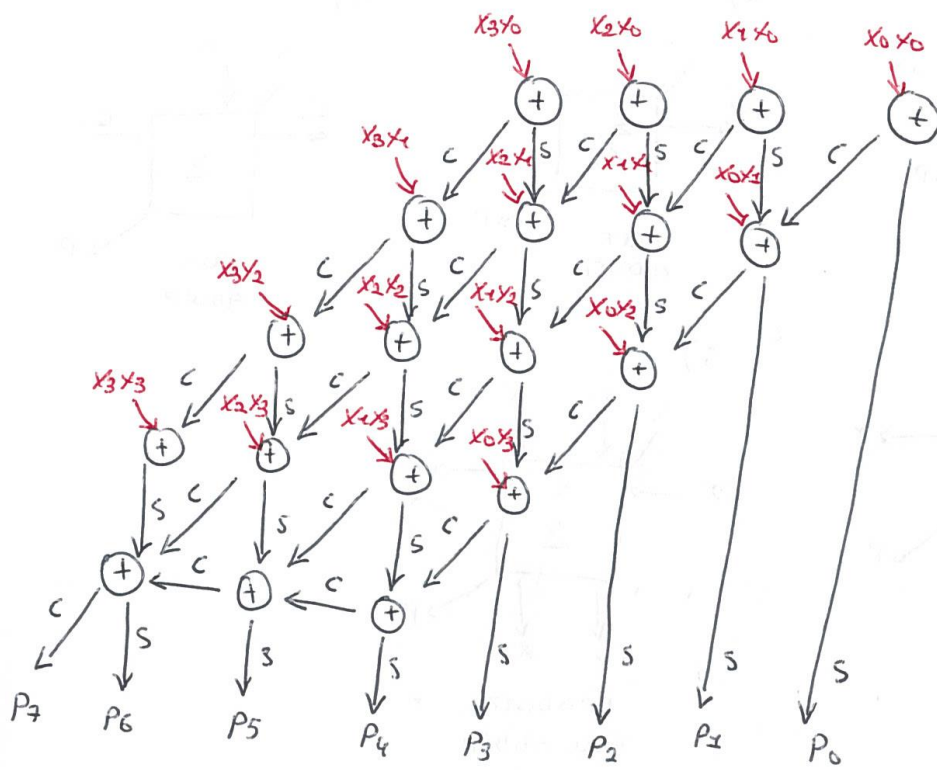
$$Y = \sum_{j=0}^{m-1} y_j 2^j$$

$$P = \sum_{i=0}^{m-1} x_i 2^i \cdot \sum_{j=0}^{m-1} y_j 2^j = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} (x_i \cdot y_j) \cdot 2^{i+j} = \sum_{k=0}^{m+m-1} p_k \cdot 2^k$$

Se $m = n = 4$:

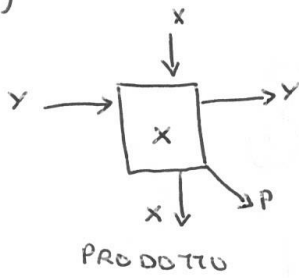
$$\begin{array}{r} x_3 \quad x_2 \quad x_1 \quad x_0 \cdot \\ y_3 \quad y_2 \quad y_1 \quad y_0 = \\ \hline x_3 y_0 \quad x_2 y_0 \quad x_1 y_0 \quad x_0 y_0 \\ x_3 y_1 \quad x_2 y_1 \quad x_1 y_1 \quad x_0 y_1 \\ x_3 y_2 \quad x_2 y_2 \quad x_1 y_2 \quad x_0 y_2 \\ x_3 y_3 \quad x_2 y_3 \quad x_1 y_3 \quad x_0 y_3 \\ \hline p_7 \quad p_6 \quad p_5 \quad p_4 \quad p_3 \quad p_2 \quad p_1 \quad p_0 \end{array}$$

SCHEMA :

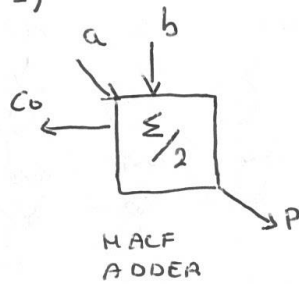


MOLTIPLICATORE PARALLELO

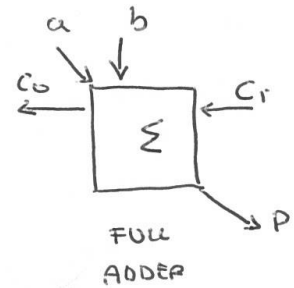
1)



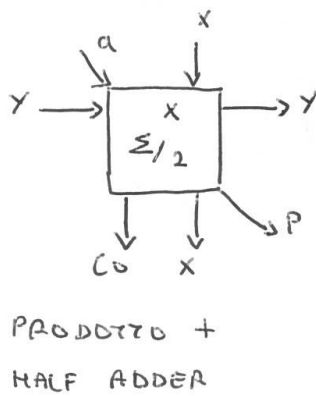
2)



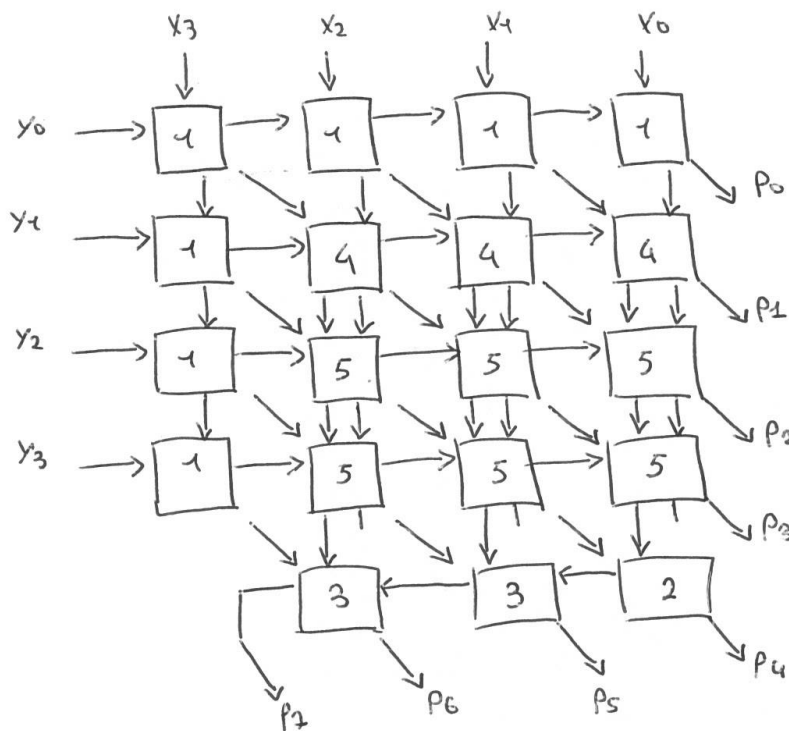
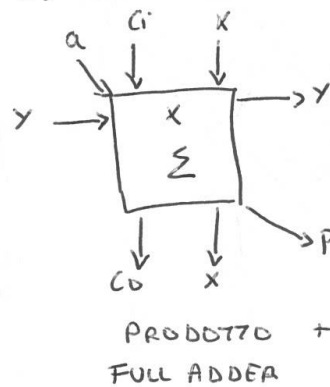
3)



4)



5)



N HALF ADDER
N(N-2) FULL ADDER
N² AND

$$T_{tot} = 2(N-1)T_B, \quad T_B: \text{single cell propagation delay.}$$

(76)

QUANTIZZAZIONE DIGITALE

INTEGER REPRESENTATION

Con N bit è possibile rappresentare numeri interi positivi da $\emptyset$ a 2^{N-1} .

$$UN(N) = [\emptyset; 2^{N-1}] \quad , \quad N = \lceil \log_2(x+1) \rceil$$

addizione:

$$UN(N) + UN(N) \Rightarrow UN(N+1)$$

sottrazione:

$$UN(N) - UN(N) \Rightarrow UN(N) \quad \text{se } \geq \emptyset$$

$$UN(N) - UN(N) \Rightarrow SG(N+1) \quad \text{se } < \emptyset$$

$$UN(N) - UN(N) \Rightarrow UN(N+N)$$

Rappresentazione Signed in complemento a 2:

Dati N bit, i numeri negativi sono rappresentati con $2^N - |x|$.

MSB svolge la funzione di segno.

$$SG(N) = [-2^{N-1}; 2^{N-1}-1]$$

$$x \in SG(N) = \begin{cases} x & \text{se } \emptyset \leq x < 2^{N-1} \\ 2^N - |x| & \text{se } -2^{N-1} \leq x < \emptyset \end{cases}$$

$$N = \begin{cases} \lceil \log_2(x+1) \rceil + 1 & \text{se } x \geq \emptyset \\ \lceil \log_2(|x|) \rceil + 1 & \text{se } x < \emptyset \end{cases}$$

addizione: $SG(N) + SG(N) = SG(N+1)$

→ se carry out non fa più significato in questa operazione.

I circuiti digitali esistono in maniera intrinseca col complemento a 2 per addizioni e sottrazioni.

(74)

moltiplicazione: $SG(N) \cdot SG(N) \Rightarrow SG(2N)$

I circuiti digitali non lavorano intrinsecamente col complemento a 2 nell'operazione di moltiplicazione.

→ Si moltiplicano i valori assoluti e si aggiunge il segno.

SIGNED WORD EXTENSION VS UNSIGNED WORD EXTENSION

UN: Propaga ϕ logico.

SG: Propaga MSB.

REAL REPRESENTATION

FLOATING POINT

$$x \in \mathbb{R} : x = M \cdot \beta^E$$

M: mantissa

β : base

E: esponente

SINGLE PRECISION

1 BIT SEGNO

8 BIT ESPONENTE

23 BIT MANTISSA

BASE 2

BIAS: 127

32 BIT

DOUBLE PRECISION

1 BIT SEGNO

11 BIT ESPONENTE

52 BIT MANTISSA

BASE 2

BIAS: 1023

Distribuzione dei valori non uniforme.

FIXED POINT

$$x^* \in \mathbb{R} : x = x^* \cdot \text{LSB}$$

Consiste esattamente come una rappresentazione integer ma pesa per LSB.

Se LSB è una potenza di 2:

$$\text{LSB} = 2^{-F} : x = \underbrace{b_{N-1} b_{N-2} \dots b_1 b_0}_{\text{Integer Part}}, \underbrace{b_{-1} \dots b_{-(F-1)} b_{-F}}_{\text{Fractional Part}}$$

Distribuzione dei valori uniforme.

STRATEGIE DI QUANTIZZAZIONE

N : numero di Bit

$$\text{LSB} = \frac{\max(\text{value})}{2^{N-1}}$$

se si usa una quantizzazione in complemento a 2 asimmetrica.

$$\text{LSB} = \frac{\max(\text{value})}{2^{N-1} - 1}$$

se si usa una quantizzazione in complemento a 2 simmetrica.

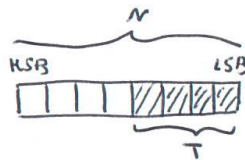
$$\text{value}_q = \text{round} \left(\frac{\text{value}}{\text{LSB}} \right) \cdot \text{LSB}$$

$$\epsilon_A = |\text{value} - \text{value}_q|$$

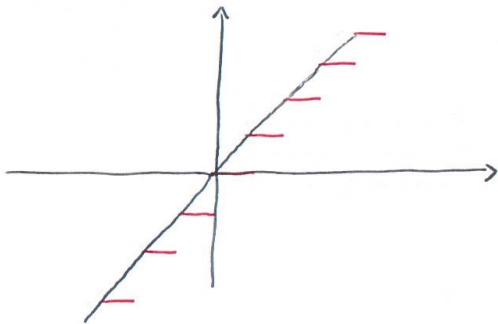
$$\text{Se } \text{LSB} = 2^{-F} \text{ e } \text{LSB} \leq \epsilon_A \text{ dato} : F = \lceil \log_2 \left(\frac{1}{\epsilon_A} \right) \rceil$$

Strategie di Troncamento

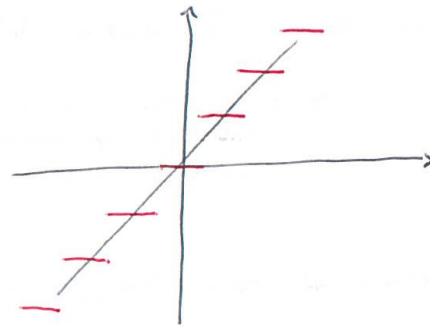
$$LSB_t = LSB \cdot 2^T$$



$$\text{value}_t = \text{floor} \left(\frac{\text{value}}{2^T} \right)$$

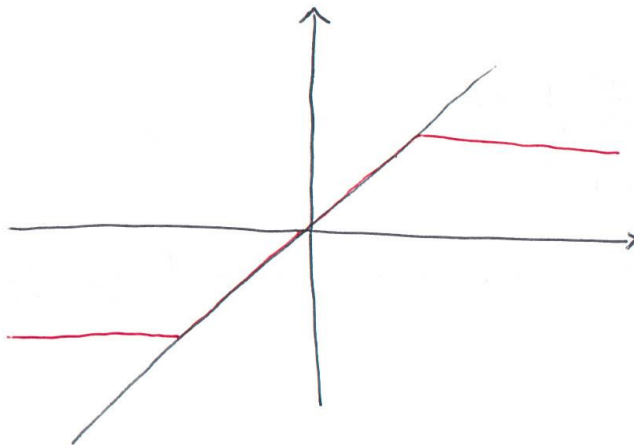
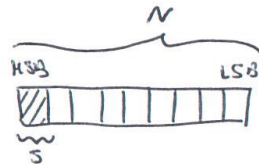


$$\text{value}_t = \text{round} \left(\frac{\text{value}}{2^T} \right)$$



Strategie di Saturazione :

$$\text{value}_s = \begin{cases} x & \text{se } x \text{ altrimenti} \\ -2^{N-1-S} & \text{se } x < -2^{N-1-S} \\ 2^{N-1-S} & \text{se } x > 2^{N-1-S} \end{cases}$$



N.B. nelle operazioni di somma e sottrazione LSB deve essere lo stesso.

nelle operazioni di moltiplicazione LSB si moltiplica.

FILTRI DIGITALI

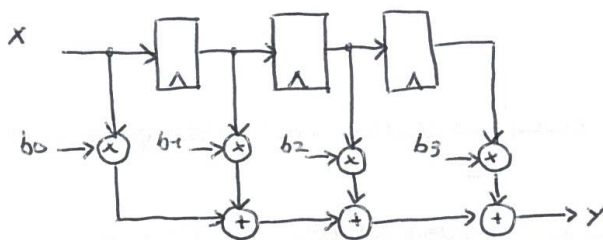
FIR

N : Ordine del filtro

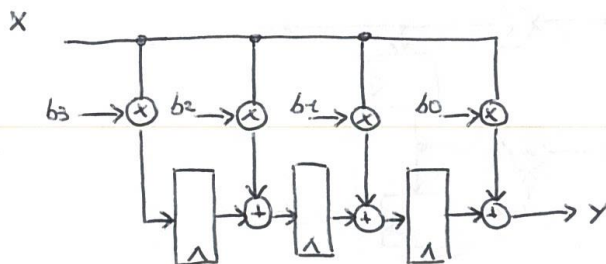
$N+1$ tappi del filtro

Filtro sempre stabile con possibilità di avere risposta in fase lineare.

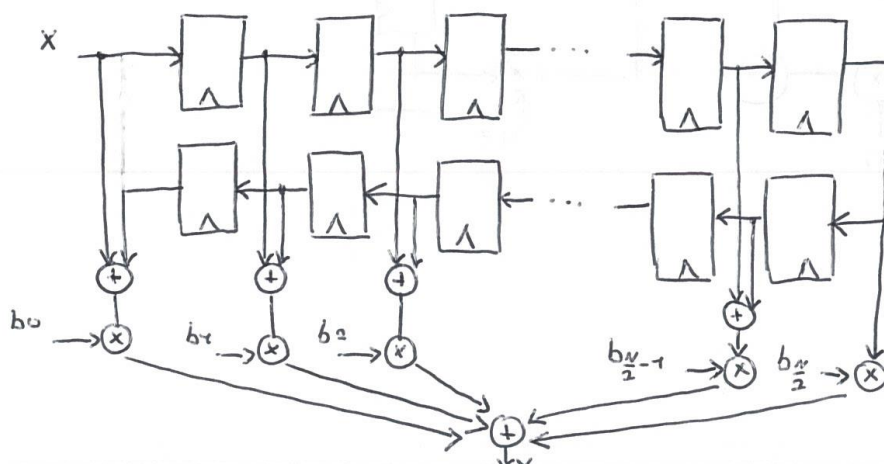
FORMA DIRETTA



FORMA TRASPOSTA

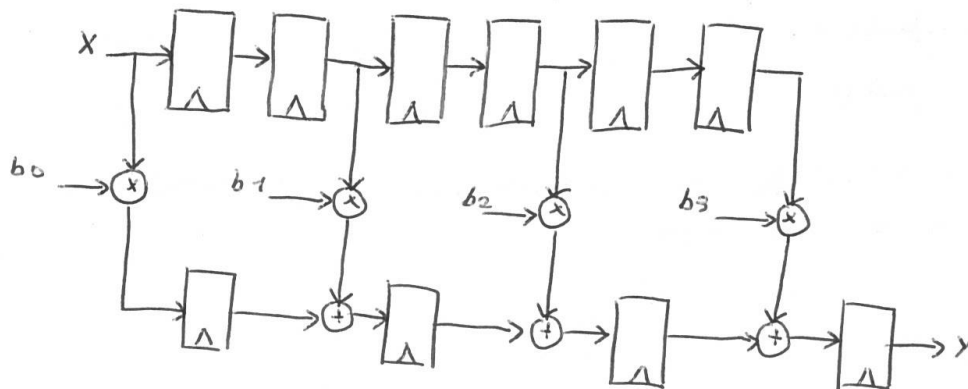


FORMA SIMMETRICA



84

FORMA DIRETTA CON PIPELINE



IIR

È presente una reazione all'uscita nei confronti dell'ingresso.

È un filtro che può essere instabile a causa dei poli ed ha una risposta non lineare in fase.

FORMA DIRETTA 1

