



Politecnico di Bari

Corso di Laurea in Ingegneria Informatica
Anno Accademico 2009/2010

Disciplina: Elettronica 2
Docente: V. Passaro

[RELAZIONE DI LABORATORIO DI ELETTRONICA 2]

INVERTITORE, PORTE LOGICHE C-MOS, FLIP FLOP, MULTIPLEXER, DECODIFICATORE E SOMMATORE

Indice

•	INDICE	2
•	PREFAZIONE	3
1.	INVERTITORE.....	4
1.1.	Cos'è un invertitore?	4
1.2.	Descrizione logica o comportamentale	5
1.3.	Descrizione a livello di Transistor	7
1.3.1.	Analisi Statica	12
1.3.2.	Analisi Dinamica	12
1.4.	Descrizione layout	14
1.4.1.	Estrazione dei parametri parassiti	16
1.5.	Oscillatore ad anello	17
2.	PORTE LOGICHE C-MOS	20
2.1.	Cosa sono le porte logiche?	20
2.2.	Descrizione a livello transistor	22
2.2.1.	Analisi Statica	25
2.2.2.	Analisi Dinamica	29
2.3.	Progetto Layout	33
3.	FLIP FLOP JK	35
3.1.	Cos' è un flip flop?	35
3.2.	Descrizione a livello logico	38
3.3.	Descrizione a livello layout	40
3.3.1.	Analisi degli stati logici	41
3.3.2.	Analisi dinamica	42
4.	MULTIPLEXER	44
4.1.	Cos'è un multiplexer?	44
4.2.	Simulazione logica	45
5.	DECODIFICATORE	49
5.1.	Cos'è il decodificatore?	49
5.2.	Simulazione Logica.....	52
6.	SOMMATORE	53
6.1.	Cos'è il sommatore?	53
6.2.	Simulazione Logica	57

PREFAZIONE

Nella relazione vengono presentati i concetti base gli argomenti proposti nell'intestazione in modo che chiunque abbia modo di leggere questo referto può comprendere a fondo tutti i valori che tale esercitazione possa predisporre.

Alla base di tutto ci sono due prove tenutesi il 21 Ottobre 2009 e il 4 Novembre 2009, nel Laboratorio di Cad presso il dipartimento del DEE del Politecnico di Bari, nella quale usando la guida proposta dal docente, in formato HTML, con l'ausilio di un PC e di alcuni Software si è portata a termine.

L'esercitazione come da intestazione proponeva alcuni modi sull'uso dell'invertitore, porte logiche c-mos, flip flop, multiplexer e decodificatori e a scelta dello studente un sommatore, il tutto però attraverso l'ausilio dei programmi di simulazione.

I software usati sono:

- MicroSim
- SwitchCad
- MicroWind 2

La relazione è strutturata secondo le istruzioni del docente, in modo che il fine da raggiungere non perda il valore della definizione.

E' necessario precisare che per portare a termine l'opera, sono stati di ausilio libri di testo, appunti, manuali, enciclopedie e siti internet al fine di dare una giusta motivazione a fronte delle varie decisioni prese durante l'esercitazione.

A tal punto si augura una buona lettura.

Angelo Antonio Salatino

aas88ie@gmail.com



1 INVERTITORE

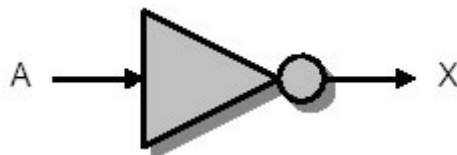
1.1 Cos'è un invertitore?

L'invertitore per definizione è un'entità o sistema che fornisce in uscita una grandezza opposta a quella immessa in ingresso. L'esempio più banale per comprendere tale concetto lo si può fare in riferimento all'automobile dove quest'ultima procede in avanti per un determinato tempo e ad un tratto si ferma per immettere la retromarcia. Come ben sappiamo indipendentemente dalla marcia con cui sta viaggiando l'automobile, il motore gira sempre nello stesso verso, quando si va ad immettere la retromarcia non si fa altro che innescare un meccanismo che tende a far girare le ruote in modo opposto alla rotazione fornita dal motore.

Ma che funzionalità può assumere un invertitore in elettronica digitale?

Un invertitore in elettronica digitale è usato per implementare la funzione NOT, utile per i calcoli digitali che a loro volta si basano sull'uso del sistema binario formato da uno 0 e un 1.

La funzione NOT ha un solo ingresso ed una sola uscita ed è caratterizzata dal fatto che l'uscita assume valore alto (ovvero 1) se e solo se l'ingresso non assume valore alto (o anche 0) e viceversa. Il simbolo per indicare la funzione NOT è il seguente:



Riportando in tabella i valori forniti dalla funzione NOT si ottiene la seguente True Table (tabella delle verità):

A	X
0	1
1	0

La funzione NOT quindi inverte il segnale d'ingresso e il circuito che realizza tale funzione è detto invertitore.

Il comportamento della funzione NOT detta anche porta logica è ovviamente una schematizzazione di ciò che avviene in un circuito reale costituito da transistor. Il comportamento ideale dell'invertitore, peraltro, è una buona approssimazione del comportamento di un inverter reale. In particolare bisogna definire con cura, affinché si

ottengano gli effetti desiderati, gli intervalli efficaci di funzionamento per le tensioni che stabiliscono lo stato logico di 1 e 0 detti anche margini di rumore rispettivamente alto e basso (NM_H e NM_L), si deve in più considerare le resistenze di canale e le capacità parassite.

Esistono sul mercato molti modelli di invertitori catalogati in famiglie logiche, le quali tendono a risolvere diversamente problemi legati alla potenza dissipata e al tempo di commutazione. Esistono quattro famiglie logiche definite a seconda della tecnologia usata e dell'approccio circuitale che utilizzano:

NMOS (MOSFET a canale n), CMOS (Complementary MOSFET, ovvero transistor a canale n e p ottenuti nello stesso dispositivo), TTL (Transistor-Transistor Logic, utilizza transistor BJT) e infine ECL (Emitter Coupled Logic, utilizza transistor BJT), etc.

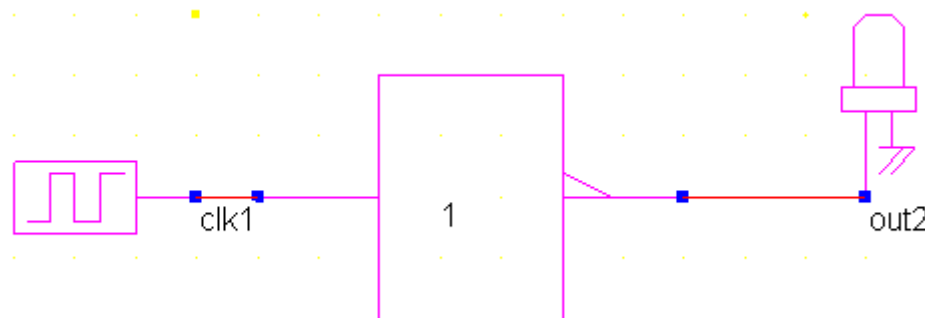
Fatta questa breve trattazione su cosa sia e dove sia usato un invertitore si può passare a descrivere dettagliatamente l'oggetto della prova.

1.2 Descrizione logica o comportamentale

In questo argomento si è passati da un discorso teorico formato da tabelle di verità e ritardi di propagazione ad un livello più pratico dove, usando il simulatore DSCH, abbiamo opportunamente inserito i componenti nell'area di lavoro e effettuati i collegamenti tra i vari componenti.

Il primo circuito formato da tre soli componenti:

- Led
- Logic Gate Not
- Clock



Tale circuito permette di visualizzare attraverso il led come si comporta la porta logica NOT a seguito di un ingresso variabile generato dal Clock. Ma ciò che è essenziale ai fini di questa prova è effettivamente a quanto ammonta il ritardo di propagazione e che influenza ha il numero di fanout sulla porta stessa.

A livello teorico sappiamo già che il fanout è il numero massimo di porte logiche dello stesso tipo che possono essere collegate in uscita. Tutto ciò comporta un limite, perché se si supera il tale numero di o per lo meno lo si raggiunge accade che la porta in questione risulta essere la più lenta di tutta la rete combinata.

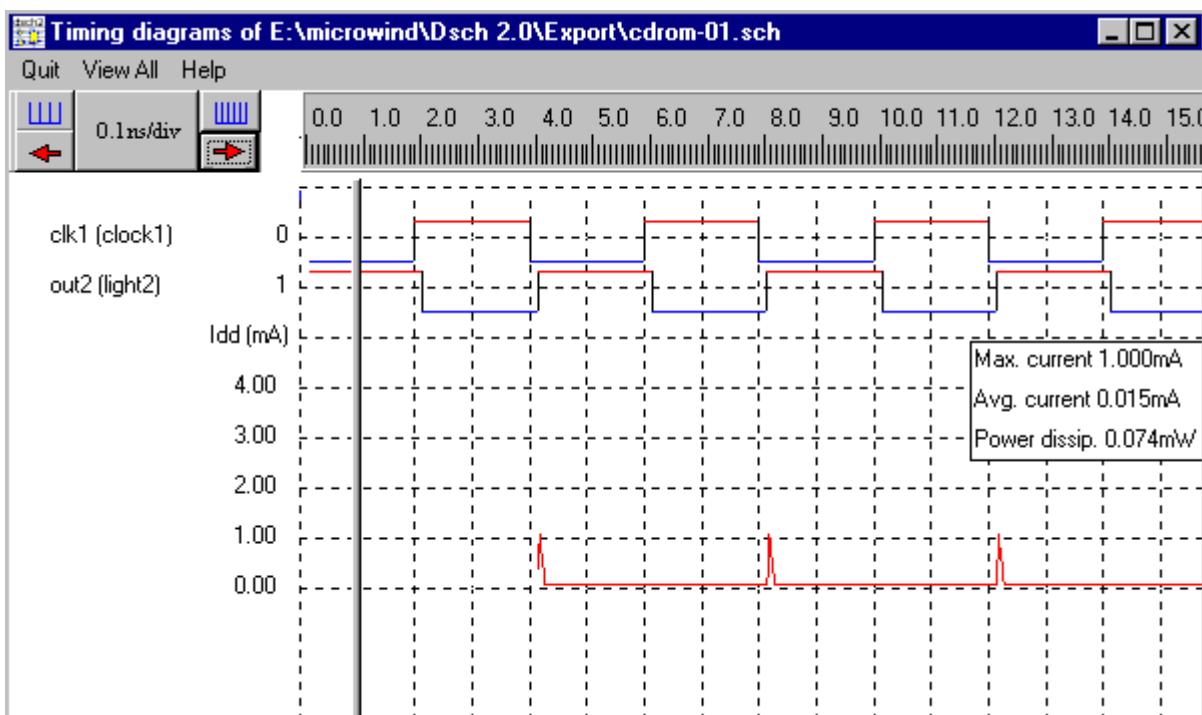
Con esattezza il tempo di propagazione vale:

$$T_p = T_{\text{delay}} + (T_{\text{wiredelay}} * N)$$

dove il T_{delay} sta per il ritardo intrinseco di propagazione e che ammonta a 0,1ns mentre il $T_{\text{wiredelay}}$ è il ritardo introdotto dal fanout che in generale vale 0,05ns. Dalla precedente relazione si evince molto facilmente il legame tra il fanout e il ritardo di propagazione.

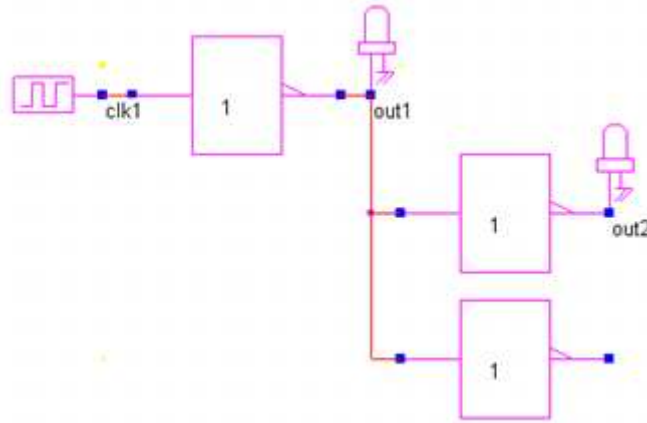
Al termine dell'assemblaggio e configurazione dei vari componenti si è passati alla simulazione dove con il Timing Diagrams, ovvero il programma per tracciare i grafici, abbiamo potuto visualizzare la netta disparità tra il segnale d'ingresso e il segnale d'uscita sull'asse dei tempi, mentre per quanto riguarda l'ampiezza e la forma, il segnale non ha subito modifiche sostanziali.

Ecco una immagine dove viene mostrato l'elaborato del timing diagrams:



Il passo successivo a questo è stato quello di mettere in uscita ad una NOT altre due porte NOT, per stimare approssimativamente il ritardo di propagazione visto che quest'ultimo dipende fortemente dal numero di fanout.

Il circuito descritto è il seguente:



Il risultato è che partendo dal generatore di clock, la prima NOT che si incontra ha due porte NOT collegate in uscita e presenta un tempo di propagazione più elevato circa 0,2ns e quindi risulta essere relativamente più lenta delle altre due porte messe in uscita che continueranno ad avere un tempo di propagazione pari a 0,15ns.

A rigor di logica tra il segnale d'ingresso e il segnale d'uscita prelevato da una delle due porte logiche avrà un ritardo di propagazione complessivo pari a :

$$T_p = 0,2\text{ns} + 0,15\text{ns} = 0,35\text{ns}.$$

In conclusione siamo riusciti a dimostrare praticamente attraverso una simulazione come varia il tempo di propagazione al variare del circuito.

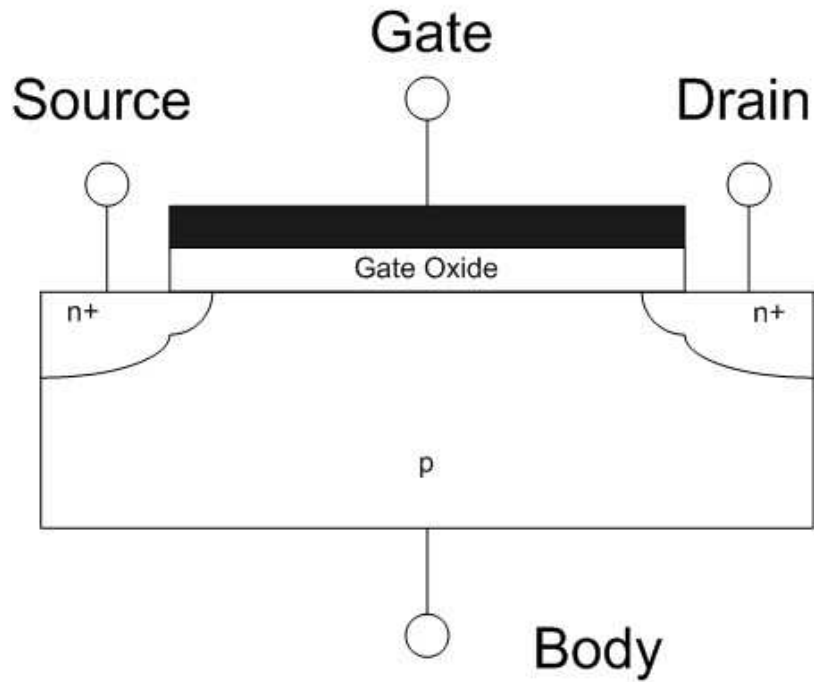
1.3 Descrizione a livello transistor

Come già esplicitato in precedenza esistono tante tecnologia con cui costruire un invertitore, la maggior parte di esse è formata per lo più da transistori, ciò che cambia da una tecnologia all'altra è il tipo di transistor usato. Nella prova è stato preso in considerazione un transistor C-MOS.

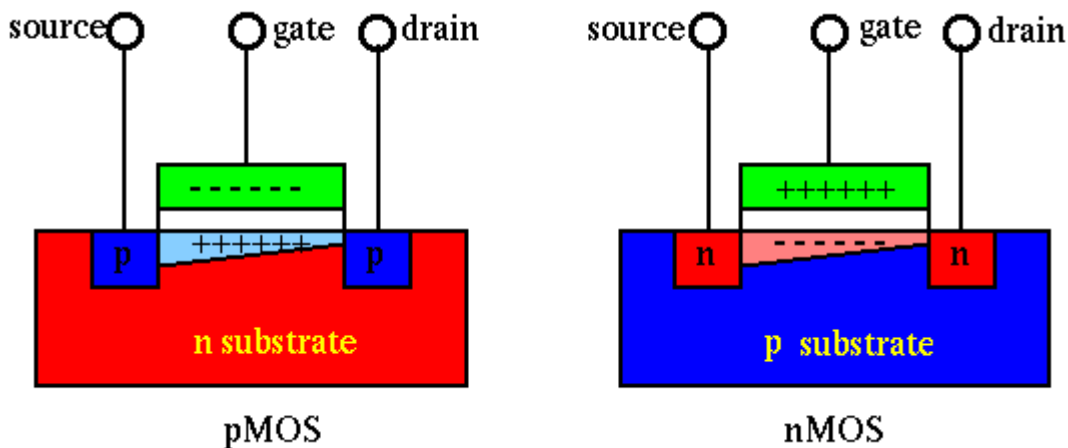
Il transistor C-MOS detto anche Complementary MOSFET ha la peculiarità di avere due transistor alloggiati in un'unica entità. Questi due transistor non sono i soliti BJT, formati da un'unica barra di silicio avente le tre zone con un drogaggio differente ma si tratta di transistor dove è presente un substrato, due sacche, un canale dove vi scorre della corrente, infine

questo canale e soggetto ad allargamenti o restrizioni a seconda della tensione di pilotaggio. Esiste anche una zona alloggiata sul canale che è il punto di contatto per il GATE, ma tra questo morsetto e il canale esiste uno strato metallo ossido semiconduttore(MOS).

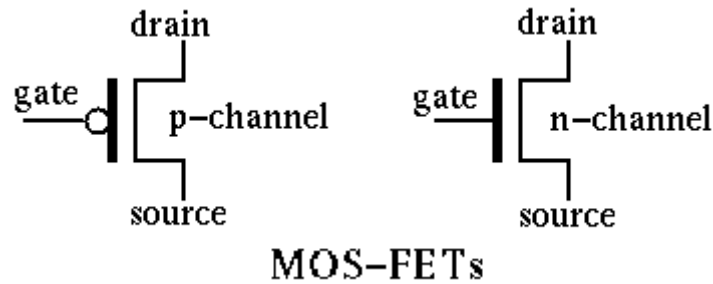
Nella seguente immagine è possibile visionare la forma di un MOSFET:



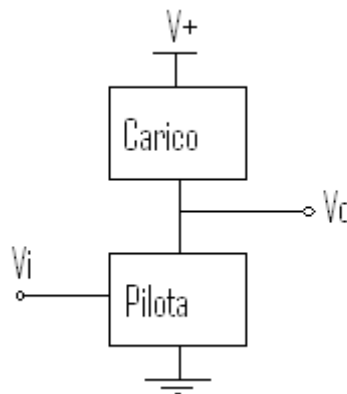
Come i BJT esistono in generale due tipi di MOSFET, uno a canale N l'altro a canale P. Il C-MOS ne contiene uno di ogni tipo.



Nella precedente figura viene mostrata la differenza tra i due transistori, dove è facile intuire le posizioni del substrato, delle sacche e del canale. Il loro simbolo circuitale è il seguente:



Al termine di questo breve cenno su questo tipo di tecnologia possiamo pensare realizzare il nostro invertitore attraverso uno schema a blocchi e precisamente si farà uso di due semplici blocchi fondamentali, uno è denominato “carico” e l’altro “pilota”. Nella seguente figura si mostra con dettaglio il concetto precedentemente esposto:



Alla base delle considerazioni su fatte si può affermare che per realizzare un invertitore è sufficiente un solo C-MOS, questo lo si può sfruttando il P-MOS(mosfet a canale P) per il carico e N-MOS(mosfet a canale N) per il pilota.

Per un progettista non rimane altro che progettare con cura il transistore facendo in modo che le correnti di Pull-Up risultino essere uguali alle correnti di Pull-Down.

Per chiarezza si descriveranno molto brevemente i concetti di corrente di pull-up e di pull-down.

La corrente di pull-up è quella corrente che si genera quando in ingresso all’invertitore viene immesso un valore di tensione basso, tale da mantenere spento il pilota e fare in modo che in uscita venga fornita una tensione e una corrente derivanti dal generatore $V+$.

La corrente di pull-down, invece, è quella corrente che permette di scaricare il carico di uscita quando all’ingresso dell’invertitore viene immesso un valore alto di tensione tale da accendere il pilota e permettere che il tutto venga scaricato a massa.

Ritornando alle questioni del progettista, quest'ultimo deve fare, come su preannunciato, in modo che le due correnti siano uguali. Ovvero deve fare in modo che vengano rispettate le condizioni di accordamento:

$$K_p = K_n$$

$$V_{tp} = V_{tn}$$

Le tensioni V_{tp} e V_{tn} sono dette tensioni di soglia e sono approssimativamente simili, quindi in buona sostanza una condizione è possibile trascurarla.

Per quanto riguarda la prima uguaglianza bisogna prestare attenzione alle misure del transistore, perché questi due fattori dipendono fortemente dalla larghezza del canale e soprattutto dalla mobilità di elettroni o protoni a seconda se sono rispettivamente a canale n o canale p.

Abbiamo quindi:

$$K_n = \mu_n \epsilon \omega_n / 2t_{ox} L \quad \text{e} \quad K_p = \mu_p \epsilon \omega_p / 2t_{ox} L$$

$$\text{ma } K_p = K_n \Rightarrow$$

$$\mu_n \omega_n = \mu_p \omega_p$$

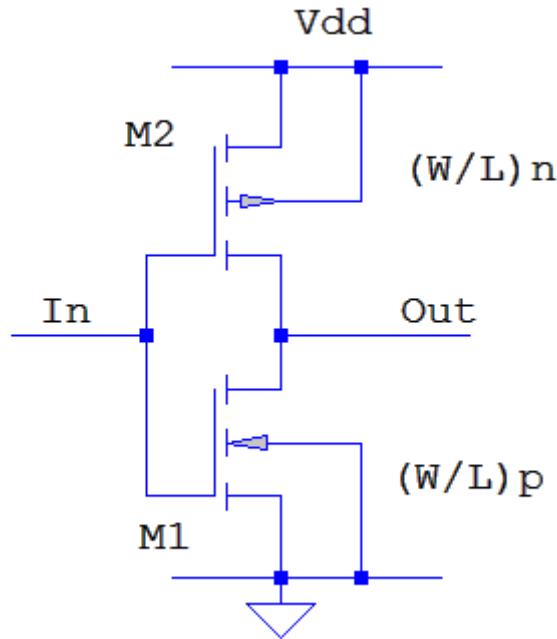
$$\mu_n / \mu_p = \omega_p / \omega_n \cong 2.5$$

A rigore di quanto visto la larghezza del canale p del transistor P-MOS deve essere circa 2.5 volte superiore alla larghezza del canale n del transistore N-MOS. Questa disparità serve a compensare la differente mobilità di elettroni e lacune.

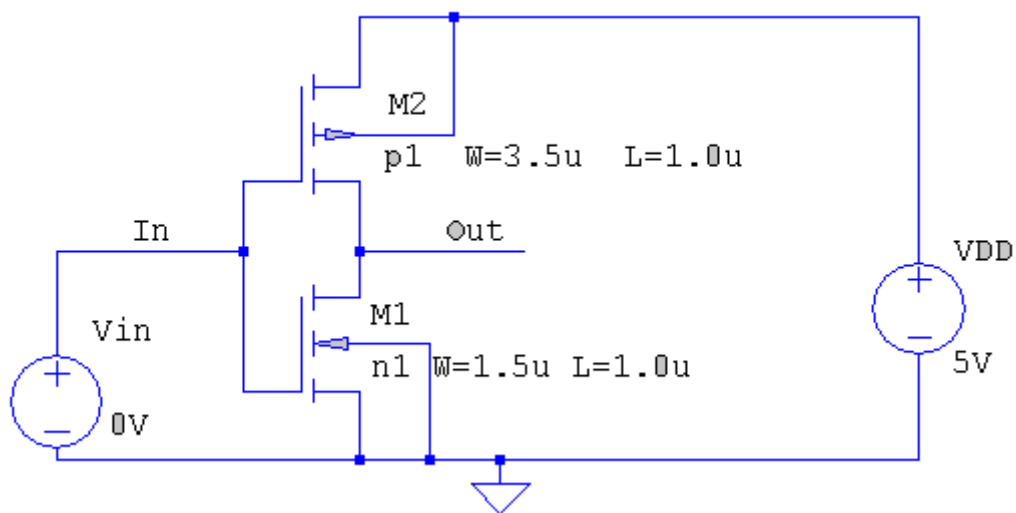
Al termine di questo fondamento teorico si è passati alla procedura di simulazione con un nuovo software.

Il software in questione è: Linear Technology Switcher CAD.

Nell'area di lavoro del suddetto software abbiamo inserito due transistori uno P-MOS e uno N-MOS e abbiamo effettuato i collegamenti secondo il seguente schema, dato che il C-MOS non è presente nel set di componenti:



Costruito il circuito, ovviamente corredato della tensione d'ingresso e della tensione di alimentazione, successivamente abbiamo modificato i valori della larghezza e della lunghezza del canale dei transistor in modo tale da avere le stesse correnti di pull-up e pull-down, rispettando le condizioni di accordamento. Riportandoci così al seguente schema:

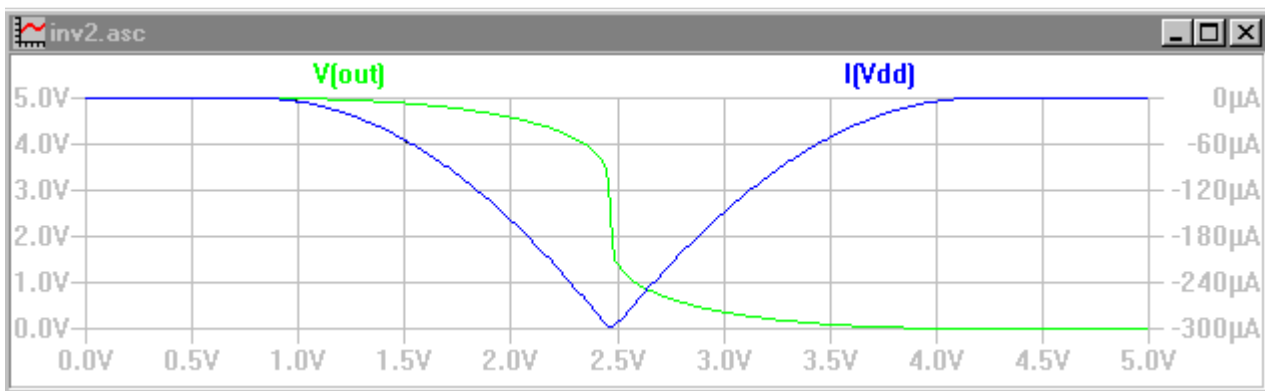


Infine abbiamo creato il file Netlist contenente tutti i comandi utili per la simulazione sia statica che dinamica.

1.3.1 Analisi Statica

L'analisi statica di questa esercitazione comprende l'analisi delle caratteristiche statiche dell'invertitore ovvero studiare le caratteristiche dell'invertitore in assenza di un carico in uscita, questo tipo di studio verrà affrontato nell'analisi dinamica.

Successivamente alla precedente prova si è passati alla simulazione impostando da prima i valori dei generatori. Di conseguenza con il software per tracciare i grafici abbiamo visualizzato la caratteristica della corrente della tensione di alimentazione e la tensione di uscita, ottenendo il seguente grafico:

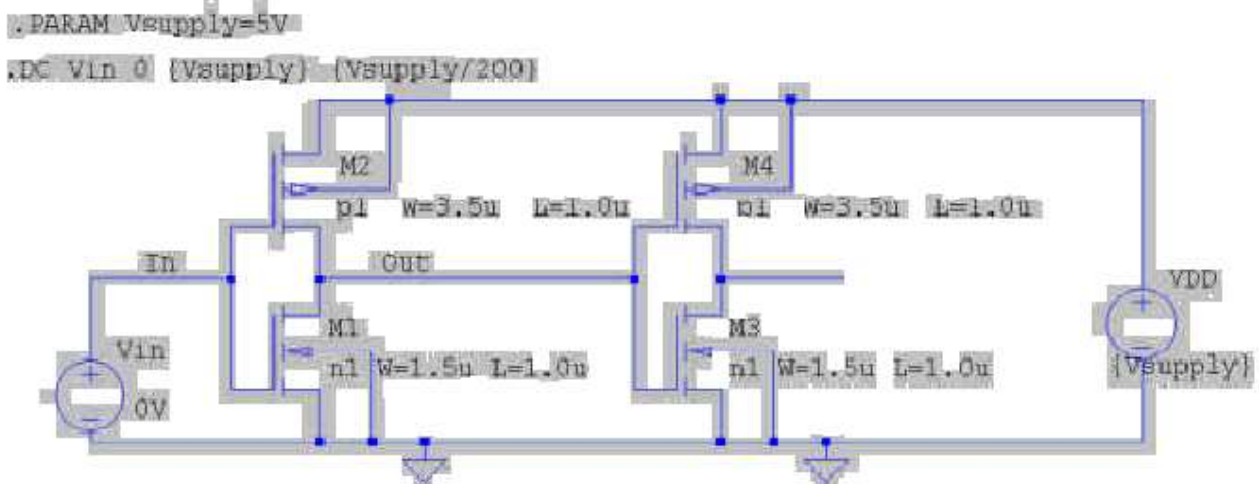


Oltre al precedente abbiamo provato ad impostare determinate tensioni di ingresso per visualizzare la risposta del nostro circuito al variare di tale tensione.

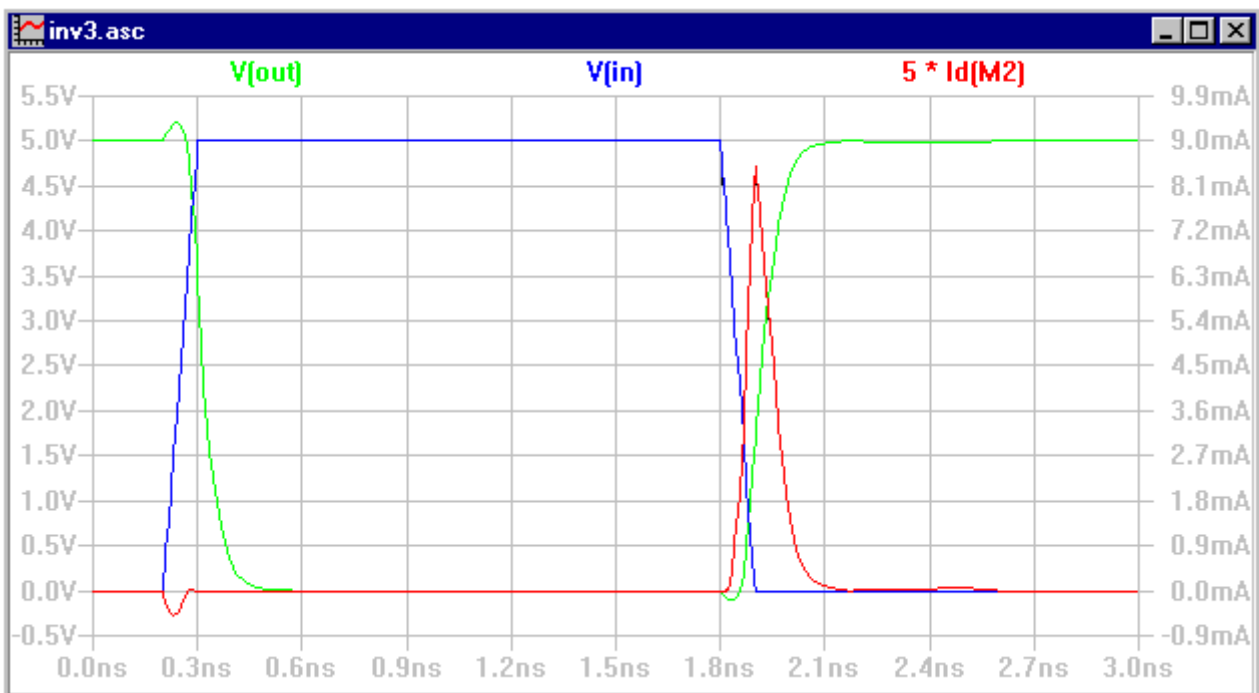
1.3.2 Analisi Dinamica

Alla base di quello che è stato fatto sin'ora è stato aggiunto un altro invertitore con le stesse caratteristiche in cascata al precedente allo scopo di avere un carico in uscita al primo invertitore.

Con questa operazione si è costretti a stimare il ritardo di propagazione considerando che il fanout = 1 e quest'ultimo contribuisce a tale ritardo. Il circuito di tale analisi è il seguente:



All'atto della simulazione le caratteristiche richieste erano la tensione d'uscita la tensione d'ingresso e la corrente sul transistor M2.

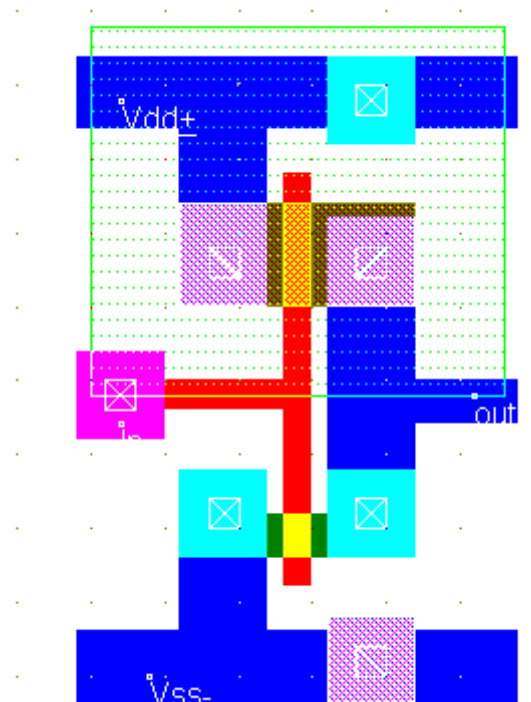


Si nota chiaramente che la corrente ha dei picchi in corrispondenza della commutazione della porta logica mentre in condizioni stazionarie la sua corrente è pressoché nulla o verisimilmente dell'ordine dei micro ampere. Tutto ciò dimostra il fatto che con questa tecnologia si può risparmiare sull'energia dissipata avendo solo consumi e surriscaldamenti in corrispondenza di commutazioni diversamente dalle altre tecnologie. Da notare anche il ritardo di propagazione che presenta il segnale d'uscita rispetto al segnale d'ingresso. In questo caso risulta che il ritardo di propagazione è circa 0,1 ns.

1.4 Descrizione a livello Layout

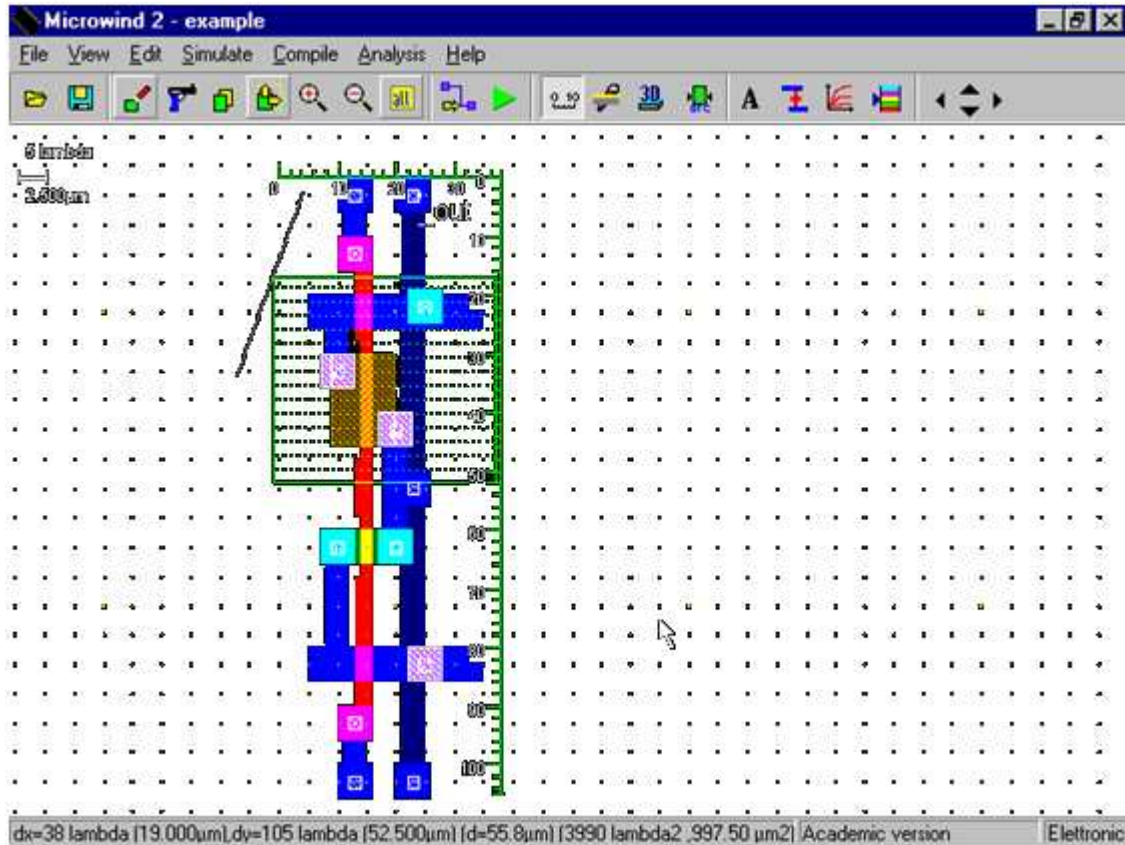
Dopo aver approfondito con un approccio pratico ciò che è stato visto a livello teorico, si è passati alla simulazione o meglio dire una *“realizzazione artigianale”* di un invertitore sempre con la tecnologia C-MOS ma a livello di layout, ovvero alla costruzione vera e propria dei substrati, dei contatti, dei legami metallici interni al transistor.

Il software per effettuare tale simulazione è il Microwind 2 dove nell’area di lavoro sono presenti una serie di bottoni con cui attivare i vari componenti costitutivi del transistor, per l’esattezza si ha il bottone metal2 (blu), metal1 (azzurro), contatti metal1-diffusioni (bianco), polisilicio (rosso), diffusione P (giallo), diffusione N (verde), Nwell (verde tratteggiato). Le dimensioni dei dispositivi restano invariate rispetto alle misure usate in precedenza. Con le direttive imposte dalla guida passo dopo passo siamo arrivati al seguente circuito:



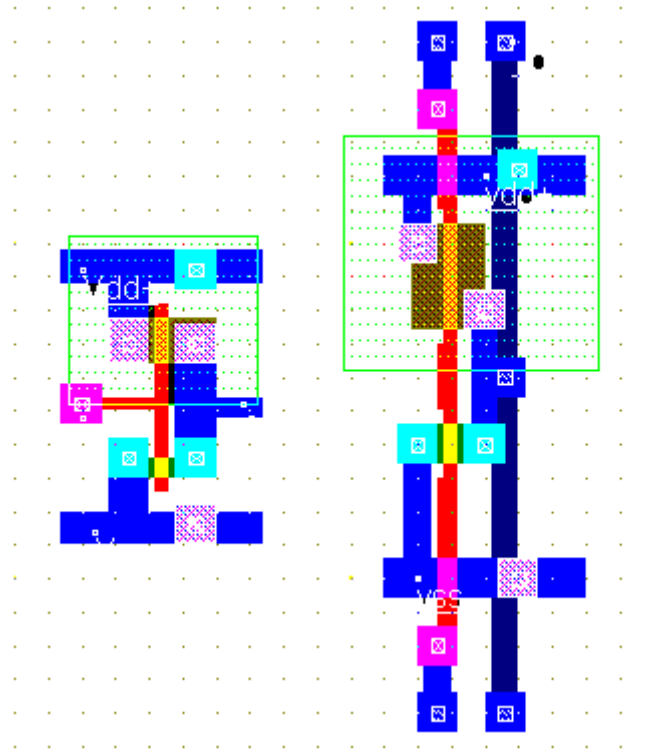
Il precedente è disposto in modo tale da semplificare i collegamenti e di disporre un numero adeguato di prese di substrato in prossimità dei dispositivi, al fine di prevenire fenomeni di latch-up.

Terminata l’operazione di assemblaggio e verificato con l’apposito tool che il tutto stia rispettando le regole di progetto si è passati al Compile OnLine, e inserito *“out = / in”* nella textbox il compilatore mostra la seguente figura, ovvero definita come il progetto *“full custom”*:



A differenza del progetto *full custom*, sia il terminale di ingresso che quello di uscita sono accessibili in metal1 sia sul lato superiore che sul lato inferiore della cella.

Per il segnale di uscita, si noti l'utilizzo di una linea in metal2 necessaria per "scavalcare" le due linee di alimentazione e di massa. Si noti che l'area occupata dal circuito è di poco inferiore a 4000 lambda quadri ed è pertanto circa tre volte maggiore rispetto a quella del circuito *full custom* (in cui avevamo peraltro utilizzato dispositivi con larghezze di canale molto inferiori). La figura seguente raffronta le dimensioni dei due layout:



Dopo aver costruito il transistor è possibile così passare alla fase di estrazione del circuito dal layout e alla sua simulazione.

1.4.1 Estrazione dei parametri parassiti

Come precedentemente accennato si passerà in questo punto ad estrarre il circuito di un invertitore a partire dal layout già elaborato e studiarne quelli che sono gli elementi parassiti introdotti dagli elementi che compongono il layout stesso.

Come in precedenza fatto per la simulazione con i transistor dove abbiamo usato come carico un altro invertitore, anche qui immetteremo in cascata un'ulteriore copia di quanto già fatto.

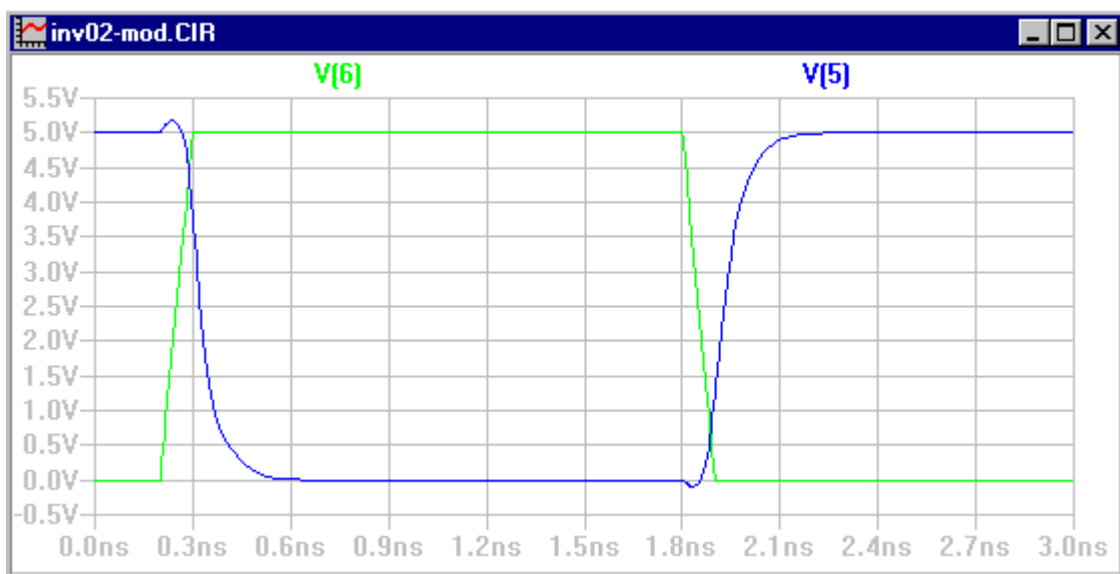
Controllando ovviamente che il layout rispetti le regole di progetto, si è passati a consultare quali potrebbero essere le capacità che si vengono a creare.

Cliccando due volte su un nodo, appare una finestra dove vengono mostrate la capacità, la resistenza e l'induttanza che si viene a creare su quel nodo. Il nodo che prenderemo in considerazione sarà quello della connessione effettuata tra l'invertitore e il carico di uscita.

La capacità al nodo risulta essere di 26fF di cui 10.5fF sono derivanti dalla capacità di gate dell'invertitore di carico, mentre per quanto riguarda l'altra parte la si deve grazie alle regioni diffuse di drain dei due mos.

Di conseguenza si è passati ad estrarre una netlist, che in seguito sarà modificata per poter effettuare la simulazione in SwitchCad.

Aperta la netlist abbiamo inserito il generatore di tensione d'ingresso, modificato le schede model dei MOS e il comando .TRAN. Salvata la netlist, chiudendo il programma mowind e avviando lo switch cad, apriamo la netlist precedentemente modificata e eseguiamo la simulazione dove con il programma per visualizzare le caratteristiche vediamo la tensione d'ingresso e la tensione d'uscita.

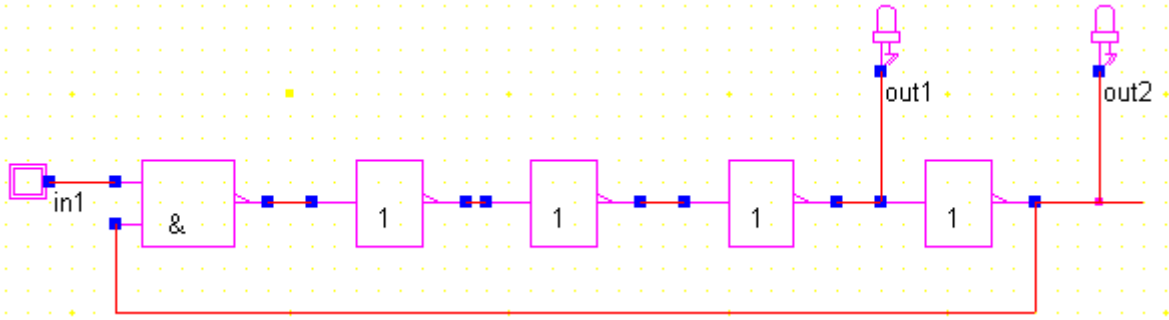


Nella precedente figura è più visibile il ritardo di propagazione che vale circa 0.10 ns. come visto in precedenza nel caso dell'analisi dinamica dell'invertitore.

1.5 Oscillatore ad anello

Quest'ultima prova di simulazione comporta nuovamente l'uso del software MicroSim per effettuare una simulazione simile a quella fatta nella descrizione logica comportamentale.

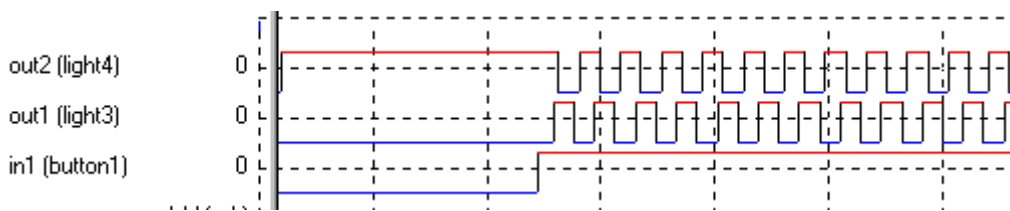
In questo caso avremo una struttura ad anello formata da tante NOT in numero dispari, chiusi ad anello:



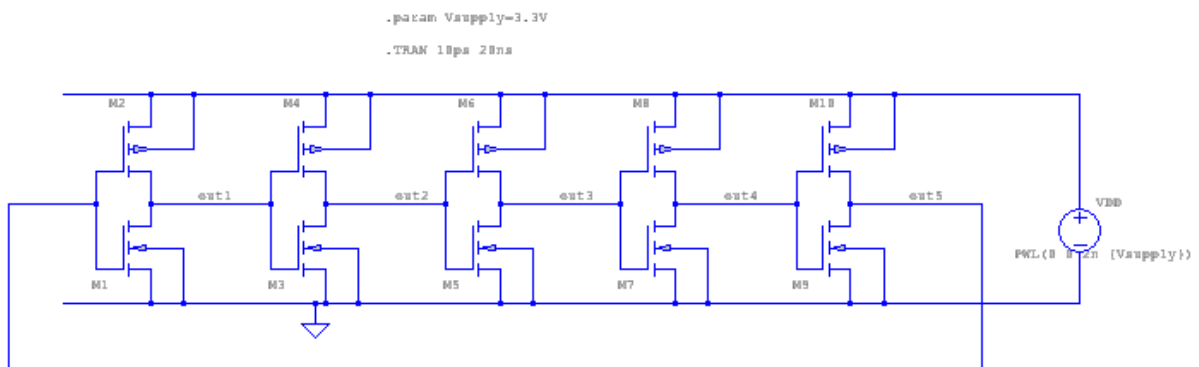
In questo circuito quando non vi è alcuna configurazione iniziale, tutto è fermo e le uscite sono basse.

Quando in ingresso si applica con il “bottono” un livello logico alto, il circuito si comporta come un oscillatore, ovvero attraverso questa configurazione fa in modo che i Led si accendano in modo alternato.

Il risultato della simulazione è riportato di seguito:

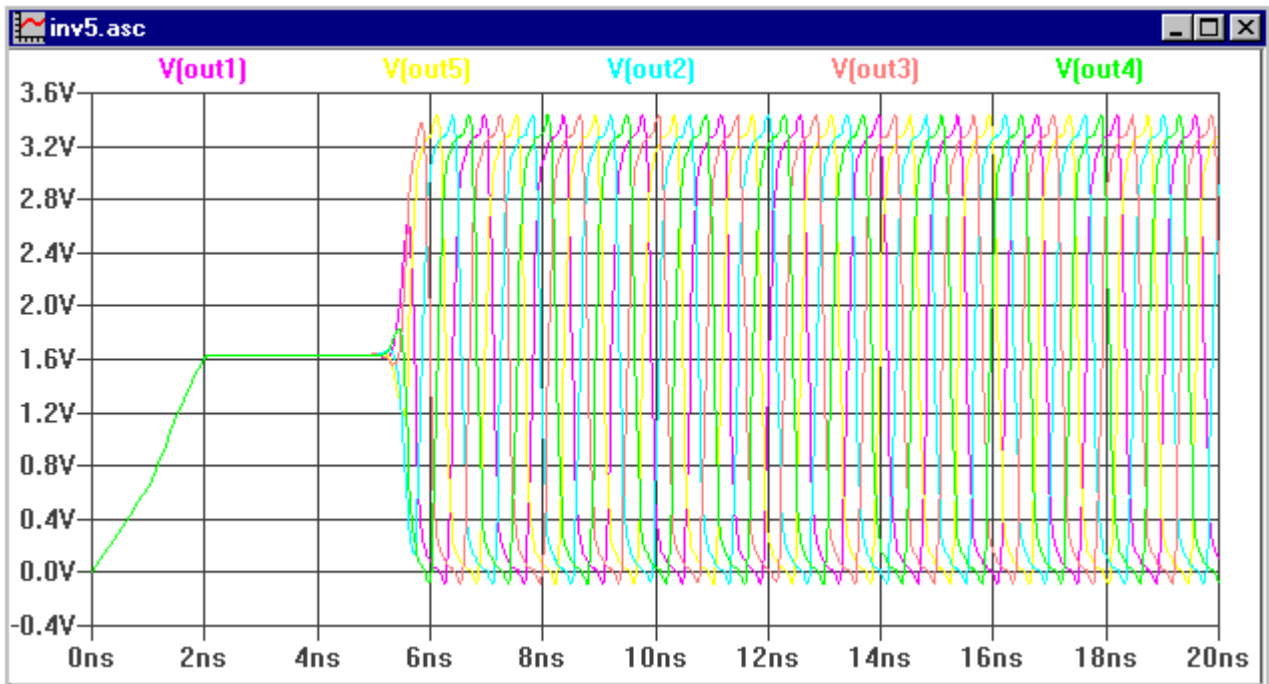


L’analisi a livello transistor di questo circuito è il seguente:



Questo circuito è costituito da 5 invertitori in cascata, dove come ingresso comune è stato immesso un generatore lineare a tratti, dove è stato settato per 0V a t=0 e Vsupply a t=2ns, dove è destinato a permanere.

Il responso è il seguente:



La Figura seguente mostra il risultato della simulazione in transitorio. Dopo il transitorio iniziale di 2ns, necessario per alimentare il circuito, le tensioni di tutti i nodi da out1 ad out4 si portano ad un valore prossimo a $V_{dd}/2$ (in questa simulazione $V_{dd}=3.3V$). Questa condizione rappresenta un punto di equilibrio instabile del circuito. Ogni minimo disturbo, indotto nella pratica da rumore o da accoppiamenti parassiti e nella simulazione da errori numerici, comporta l'allontanamento dal punto di equilibrio instabile con innesco di oscillazioni. Nella simulazione le oscillazioni si evidenziano a partire dal tempo $t=6ns$.

Con questo si dichiara conclusa l'esercitazione sugli invertitori, dove con vari processi di simulazione si è andati a comprendere meglio quelli che sono i comportamenti degli invertitori reali.

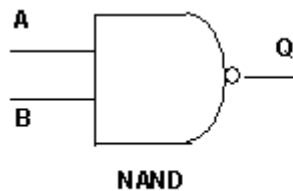
2 PORTE LOGICHE C-MOS

2.1 Cosa sono le porte logiche?

Questo argomento è stato già espletato nella parte relativa all’invertitore, ma soffermandoci un istante su questo argomento si può dire che una porta logica(logic gate) è un circuito che realizza una funzione logica elementare.

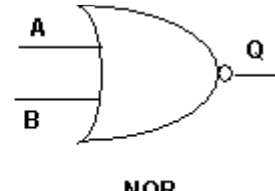
Le funzioni logiche elementari a noi più note sono la NOR, NAND, NOT.

Della NOT ci siamo già occupati quando si è parlato dell’invertitore, non ci rimane che parlare della NOR e della NAND:



NAND

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



NOR

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

Come già specificato queste funzioni logiche possono essere realizzate a livello elettronico grazie a dei circuiti digitali integrati a loro volta suddivisi in famiglie logiche:

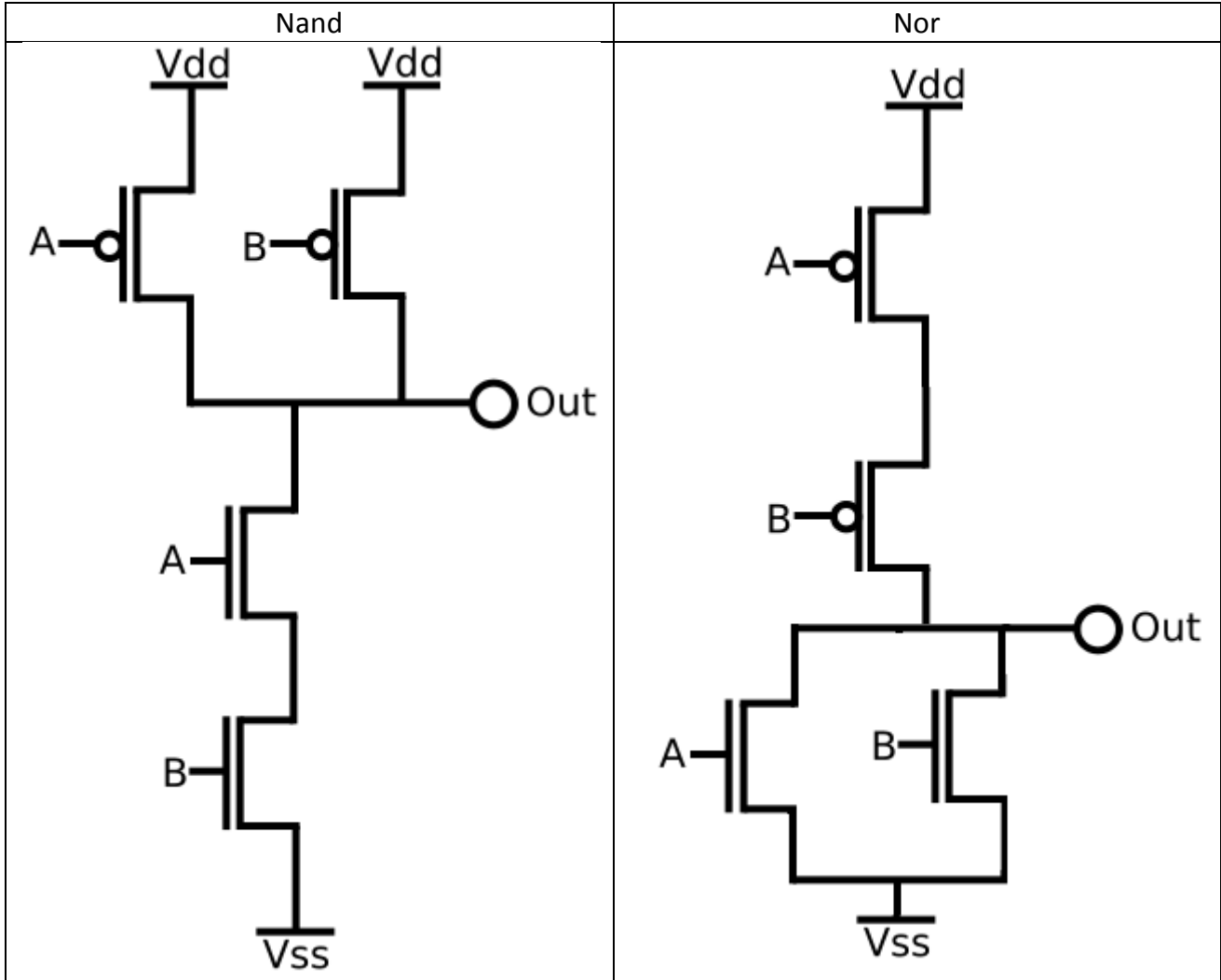
N-MOS, C-MOS, TTL, RTL, Etc..

Ci soffermeremo principalmente su quelle che sono le porte logiche C-MOS perché estremamente vantaggiose in termini di consumi dove tale tecnologia in condizioni statiche, la corrente che attraversa i transistori è molto bassa provocando così una dissipazione di potenza molto contenuta rispetto alle altre famiglie. Un’ altra parametro che rende vantaggioso l’uso di questa famiglia sta nel fatto che possiede parametri e caratteristiche di trasferimento simmetriche. La famiglia CMOS(metallo ossido semiconduttore complementari) è realizzata da due transistor Mosfet(ad effetto di campo), uno con canale di tipo N e l’altro con canale di tipo P e usando il sistema carico pilota visto già per l’invertitore è possibile realizzare le porte logiche NAND e NOR.

Nel nostro caso siamo in presenza di due ingressi e quindi a rigor di logica sia il carico che il pilota dovrà contenere due transistori rispettivamente di tipo P e due di tipo N. In definitiva saranno necessari due C-MOS visto che quest’ultima ha la capacità di contenerne uno di ogni

tipo. Si noti che questo ragionamento va fatto per qualsiasi tipo di porta logica con n ingressi, dove saranno necessarie n transistori di tipo P e n transistori di tipo N.

Con l'ausilio delle tabelle di verità si arriva ai seguenti circuiti:



Osservando la NAND, si nota che quando A e B si trovano entrambi al livello logico alto abbiamo la corrente di pull down che tende a scaricare la tensione d'uscita a massa mentre per qualsiasi altra configurazione abbiamo la corrente di pull up che tiene alto il valore della stessa uscita.

Osservando la NOR, invece, abbiamo che quando A e B si trovano entrambi al livello logico basso abbiamo una corrente di pull up che carica l'uscita fornendogli un valore alto, mentre per qualsiasi altra combinazione abbiamo che necessariamente in N-MOS si accende per dar vita ad una corrente di pull down e così scaricare a massa l'uscita.

Si deve notare un particolare che nel momento in cui andiamo a considerare queste due porte abbiamo che in alcuni casi le correnti di pull up e pull down sono differenti perdendo la caratteristica di simmetria che il componente ci offre difatti non abbiamo più l'uguaglianza tra due tempi di propagazione $T_{\text{phl}} \neq T_{\text{plh}}$. Consideriamo la NOR nella condizione A=1 B=1 abbiamo che gli N-MOS sono accesi, con una conseguente corrente di pull down che si andrà a ripartire nei due transistor in parallelo.

Abbiamo quindi : $k_n \neq k_p$

$$k_p = k_{neq} = k_{n1} + k_{n2}$$

$$k_n = \frac{1}{2} \frac{\mu_n \varepsilon \omega_n}{t_{ox} L} ; k_p = \frac{1}{2} \frac{\mu_p \varepsilon \omega_p}{t_{ox} L}$$

$$k_p = k_{neq} = k_{n1} + k_{n2} \Rightarrow \mu_p \omega_p = 2 \mu_n \omega_n \Rightarrow \frac{\omega_p}{\omega_n} = \frac{2 \mu_n}{\mu_p} \cong 2 \cdot 2,5$$

Avremo che i P-MOS saranno più grandi, conseguentemente saranno tanto più grandi quanti più ingressi ci saranno. Stesso e identico ragionamento vale per la NAND solo che qui si deve porre A=0 B=0 avendo così una corrente di pull up che si ripartisce nei due P-MOS

$$k_n = k_{peq} = k_{p1} + k_{p2}$$

$$\Rightarrow 2 \mu_p \omega_p = \mu_n \omega_n \Rightarrow \frac{\omega_p}{\omega_n} = \frac{\mu_n}{2 \mu_p} \cong \frac{2,5}{2}$$

Dove la larghezza del canale P-MOS tende a ridursi all'aumentare del numero dei canali.

Chiarito questo concetto si può dichiarare conclusa quella che è la parte teorica afferente a questo argomento lasciando spazio all'esercitazione pratica svolta in laboratorio.

2.2 Descrizione a livello transistor.

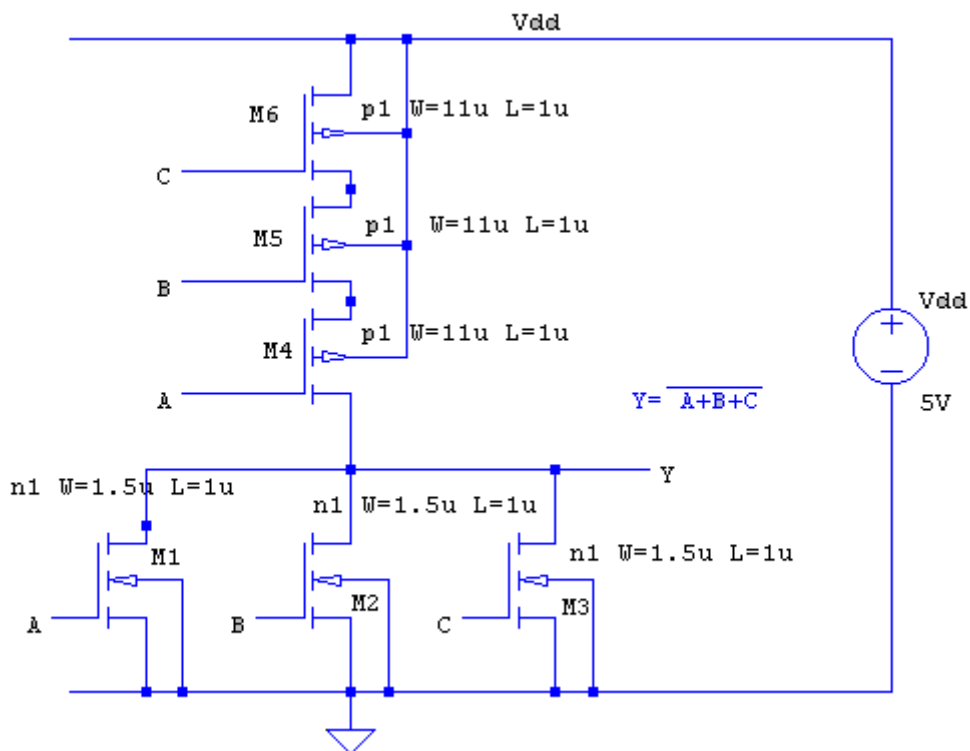
Lo scopo dell'esercitazione rivolta alle porte logiche C-MOS comprende da quanto esplicitato in precedenza la comprensione della porta logica NOR e NAND.

Nel nostro caso con esattezza useremo porte logiche a tre ingressi, con un conseguente uso di sei transistor dal motivo su specificato.

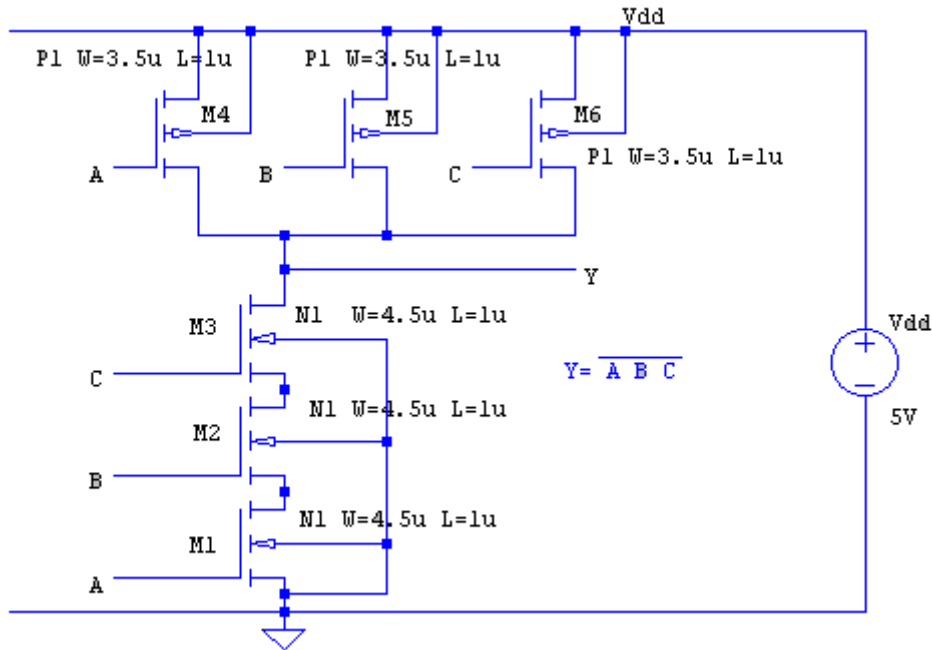
A	B	C	AND	OR
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

La precedente tabella di verità mette in mostra le relazioni tra ingresso e uscita della nostra NAND e NOR a tre ingressi.

Avremo quindi il circuito della porta NOR:



Invece per la porta NAND:



Dagli schemi circuitali si può comprendere l'uso e la disposizione dei transistor. Tutto il circuito sia per la NOR che NAND è alimentato da una sorgente in continua a 5V. I segnali d'ingresso possono essere inseriti sui morsetti A,B e C, mentre l'uscita può essere prelevata dal morsetto Y.

In entrambi gli schemi circuitali, si può notare che per ogni transistor sono riportati dei dati:

f.ex.: N1 W=4,5u L=1u

N1 →Modello

W→ Larghezza Canale

L→ Lunghezza Canale

A seguito di questa breve descrizione possiamo dire che la lunghezza del canale è uguale per tutti mentre la larghezza varia a seconda del circuito perché come citato in precedenza ci si deve porre nel caso peggiore e porre l'uguaglianza tra i due tempi di propagazione.

Inoltre si è usato il criterio dell'uguaglianza del tempo di propagazione tra i due circuiti: $t_{pnand} = t_{pnor}$ assumendo anche che i transistor a canale N della porta NOR abbia la larghezza del canale più stretta:

$$\frac{W}{L} = \frac{3\lambda}{2\lambda}$$

Nel caso del circuito della porta logica NOR abbiamo:

$$\mu_p \omega_p = 3 \mu_n \omega_n \Rightarrow \frac{\omega_p}{\omega_n} = \frac{3 \mu_n}{\mu_p} \cong 3 \cdot 2,5 \Rightarrow \omega_p = 7,5 \omega_n$$

$$\begin{cases} \omega_p = 11 \\ \omega_n = 1,5(\text{arbitrario}) \end{cases}$$

Nel caso del circuito della porta logica NAND abbiamo:

$$3 \mu_p \omega_p = \mu_n \omega_n \Rightarrow \frac{\omega_p}{\omega_n} = \frac{\mu_n}{3 \mu_p} \cong \frac{2,5}{3} \Rightarrow \omega_p = 0,83 \omega_n$$

$$\begin{cases} \omega_p = 3,5 \\ \omega_n = 4,5(\text{arbitrario}) \end{cases}$$

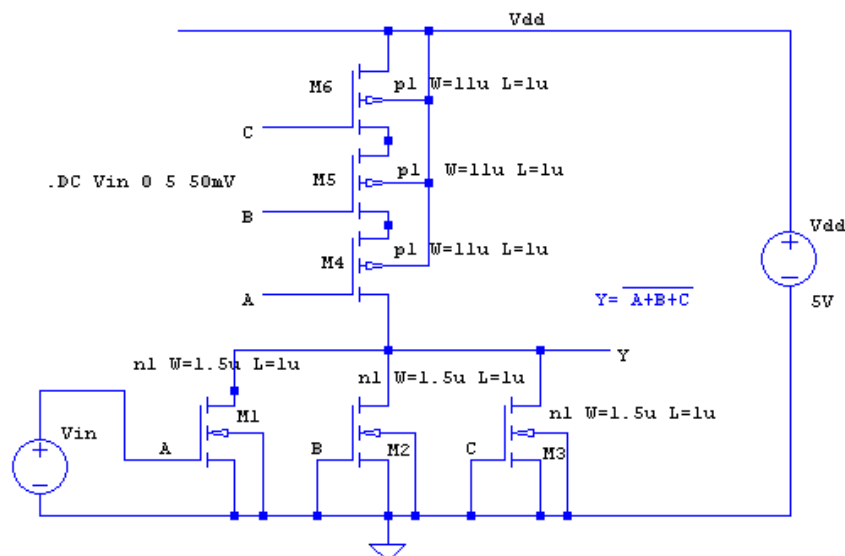
Terminate queste considerazioni si passa all'analisi statica della prova.

2.2.1. Analisi Statica

L'analisi statica consiste nel valutare le caratteristiche d'uscita dei circuiti in esame, si dovrà quindi inserire una tensione d'ingresso per poi valutare gli andamenti alla commutazione della porta logica. Si inizierà con la NOR per poi completare con la NAND.

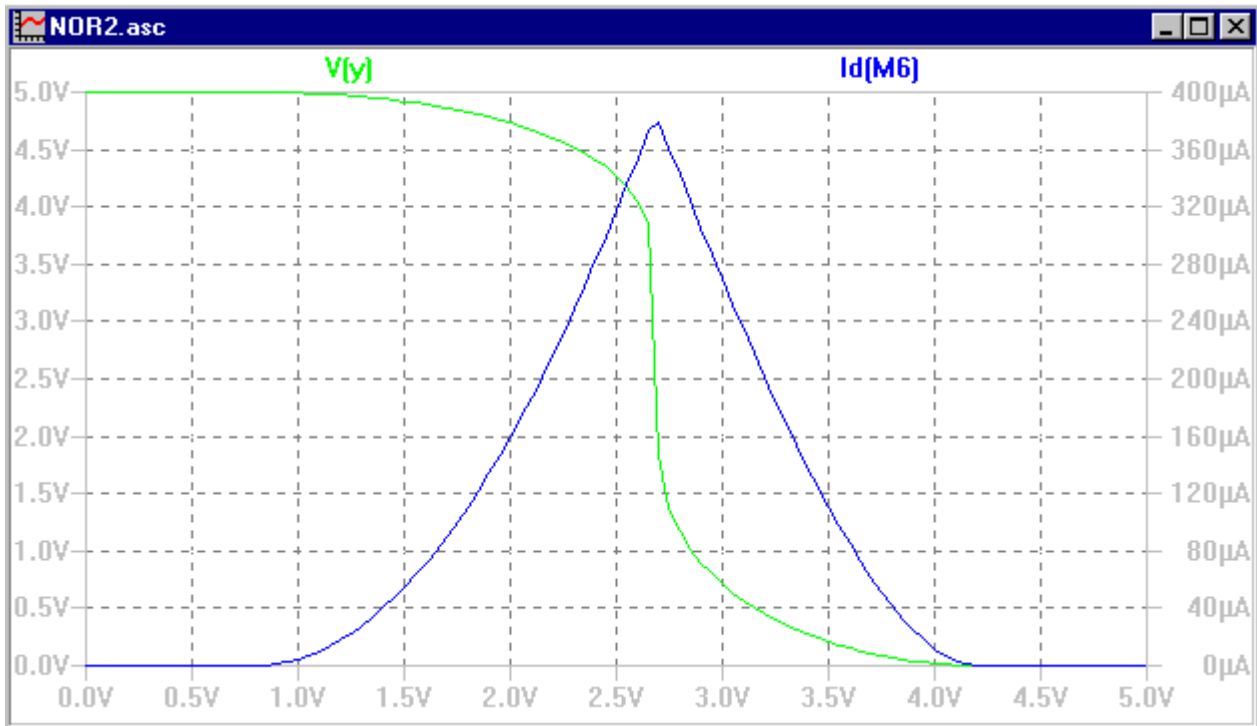
Inizialmente abbiamo il circuito alimentato ma privi di tensione d'ingresso quindi è come se gli ingressi fossero tutto ad un livello logico basso, quindi l'uscita del nostro circuito presenta un livello logico alto.

Si procede quindi ad alimentare un ingresso come mostrato nella figura:



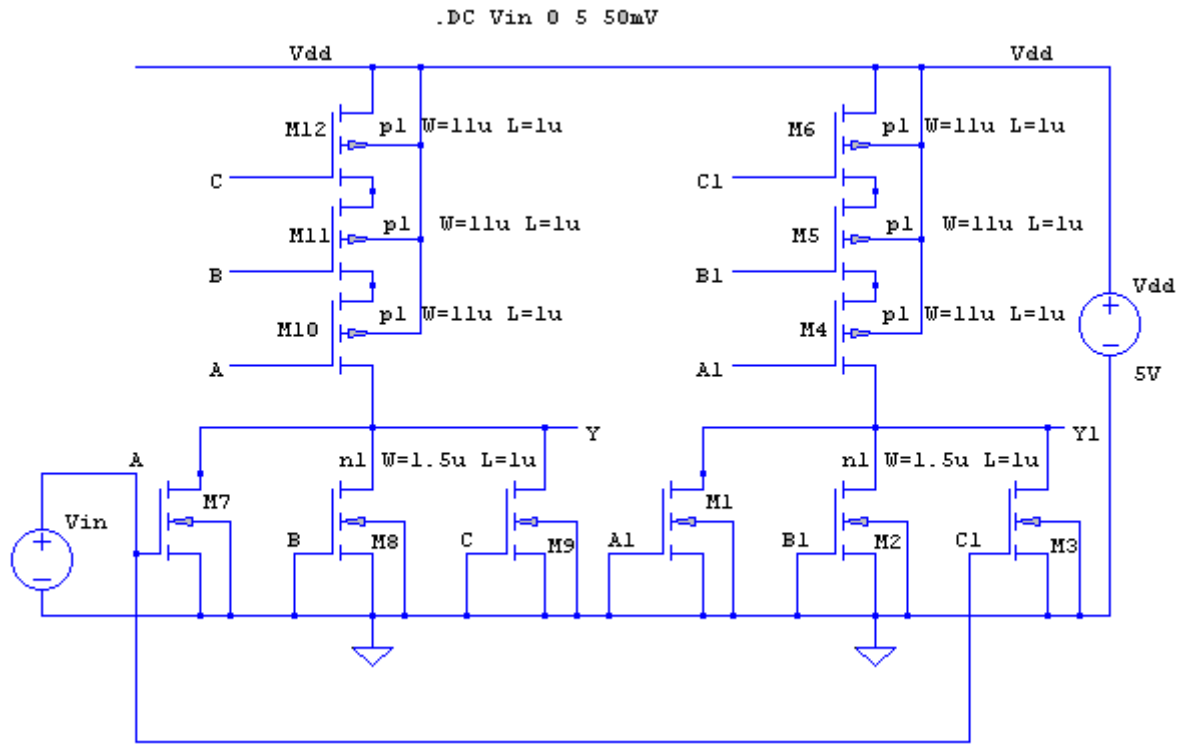
$$V_A = V_{in}; \quad V_B, V_C = 0$$

ottenendo la seguente caratteristica in uscita:



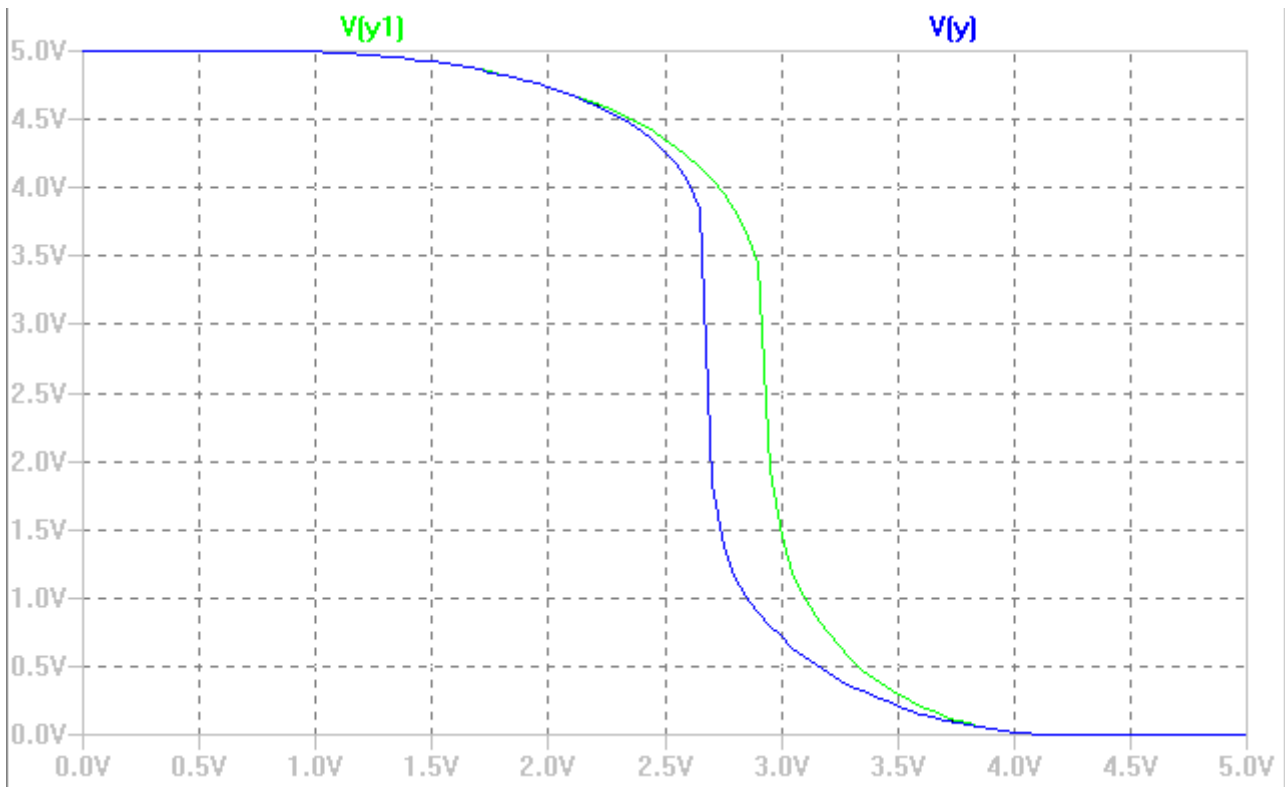
dove è semplice notare che $V_{oh}=5v$ mentre $V_{ol}=0$.

E' utile confrontare le caratteristiche che si hanno alimentando differientemente i tre ingressi, infatti con l'ausilio di un'altra porta NOR possiamo valutare la caratteristica d'uscita quando tutti gli ingressi sono posti a zero tranne C, come nel circuito alla pagina seguente:



$V_{A1}, V_{B1} = 0 \quad V_{C1} = V_{in}$

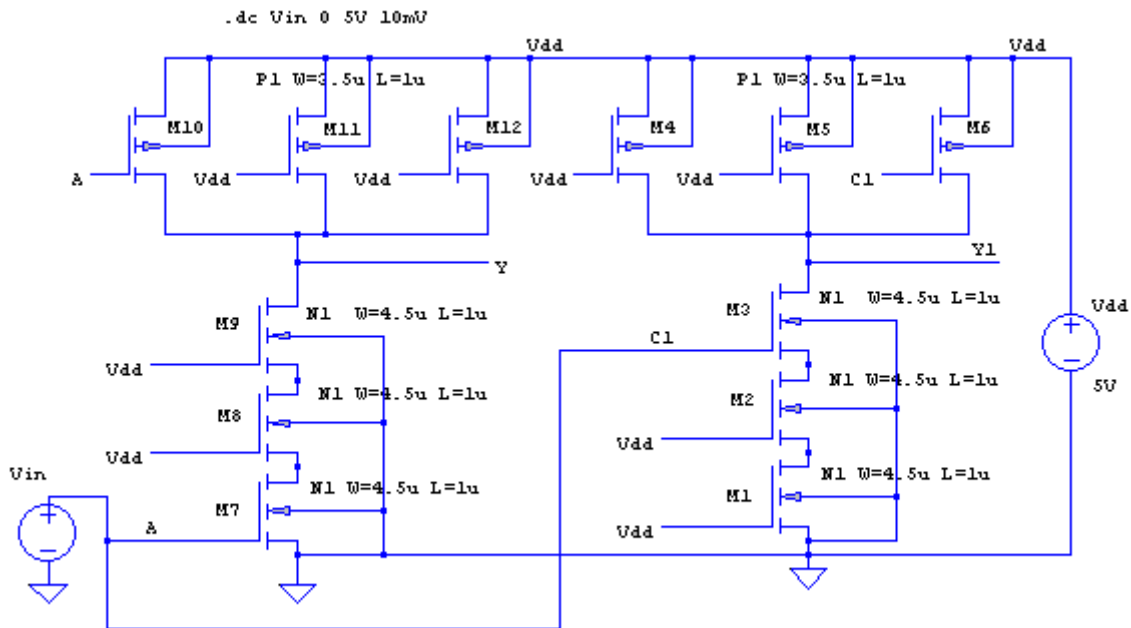
ottenendo le seguenti curve:



Sebbene le due caratteristiche siano simili l'uscita che vede l'ingresso alto sul morsetto C mostra uno shift di 250mV in più rispetto alla curva ottenuta con il solo ingresso alto sul morsetto A. Questo dipende da molti fattori circa le regioni di funzionamento dei transistor. In pratica abbiamo che il transistor P-MOS dell'ingresso C è collegato direttamente all'alimentazione mentre il P-MOS dell'ingresso A è collegato direttamente all'uscita, favorendo un tempo di scarica inferiore, dimostrando quindi la disparità incontrata, e favorendo così anche la disuguaglianza sui due tempi di propagazione sia di salita che di discesa.

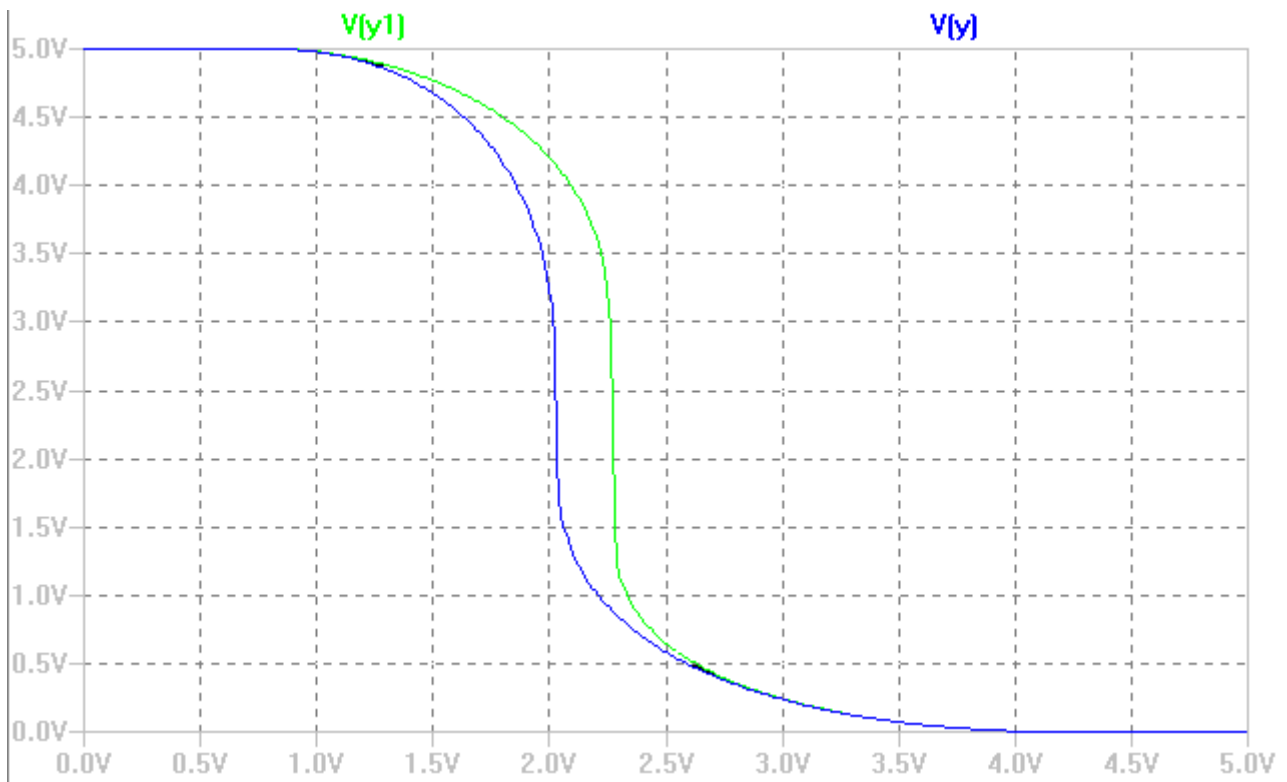
Di seguito sarà riportata l'analisi sulla porta NAND, seguendo a specchio quella svolta precedentemente.

Si ometterà quella con il singolo ingresso andando a testare, invece, quando due porta NAND ricevono due ingressi diversi, come mostra il seguente circuito:



VA = Vin VB,VC = Vdd VA1,VB1 = Vdd VC1 = Vin

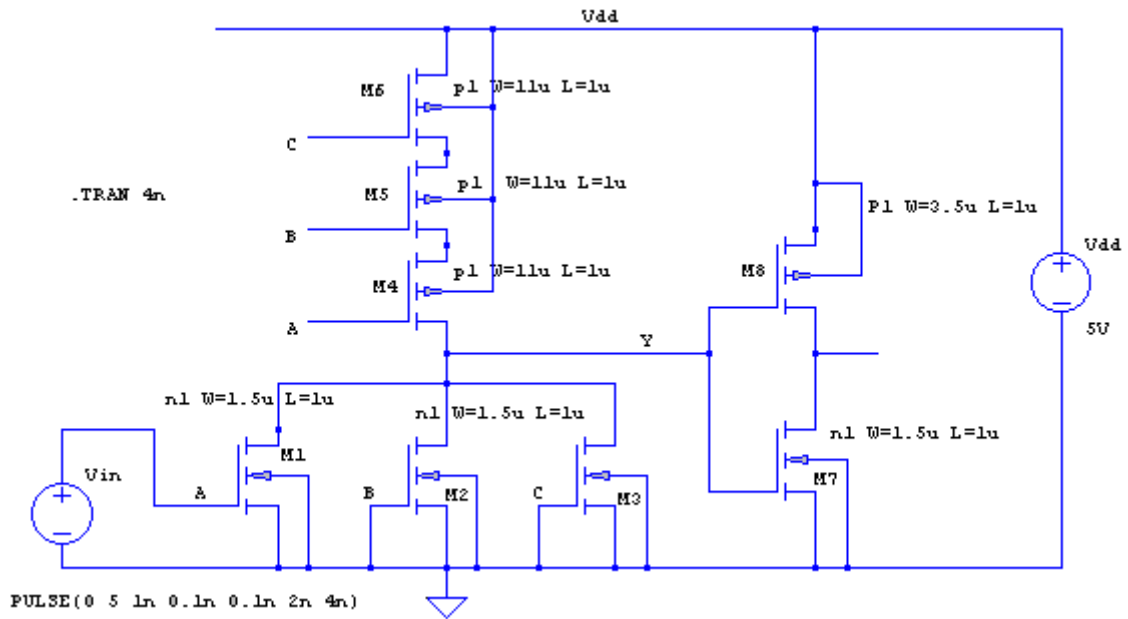
ottenendo le seguenti caratteristiche:



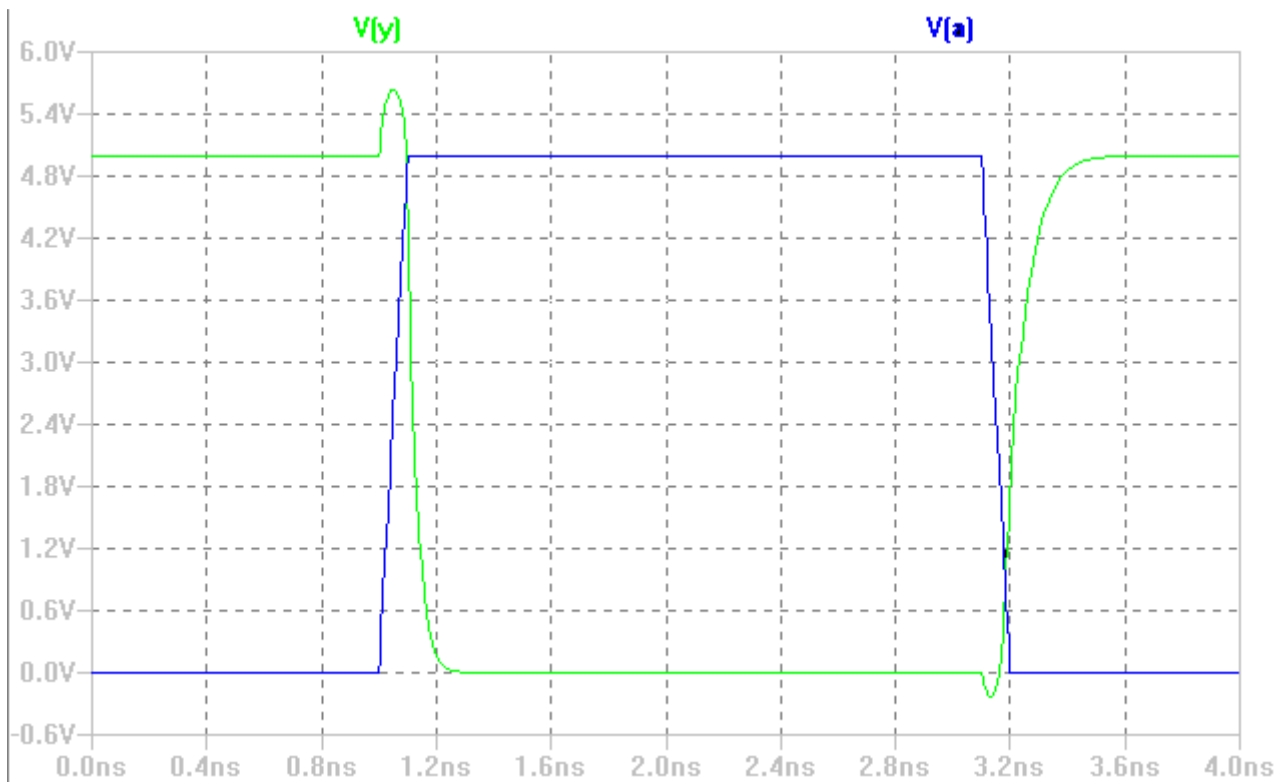
Osservando anche qui una determinata disparità tra le due caratteristiche, anch'essa determinata da quelle che sono le regioni di funzionamento dei transistor sopra descritte.

2.2.2 Analisi Dinamica

L'analisi dinamica consiste nel collegare in uscita alla porta logica un elemento circuitale che renda un effetto capacitivo o per lo meno che tendi a rendere dinamico il circuito, in modo tale da ottenere una caratteristica differente. Si noti che il nostro elemento circuitale è un invertitore, perché più elementi di questo tipo introducono effetti parassiti e quindi capacitivi, tali da soddisfare le nostre esigenze, come mostra la seguente figura:

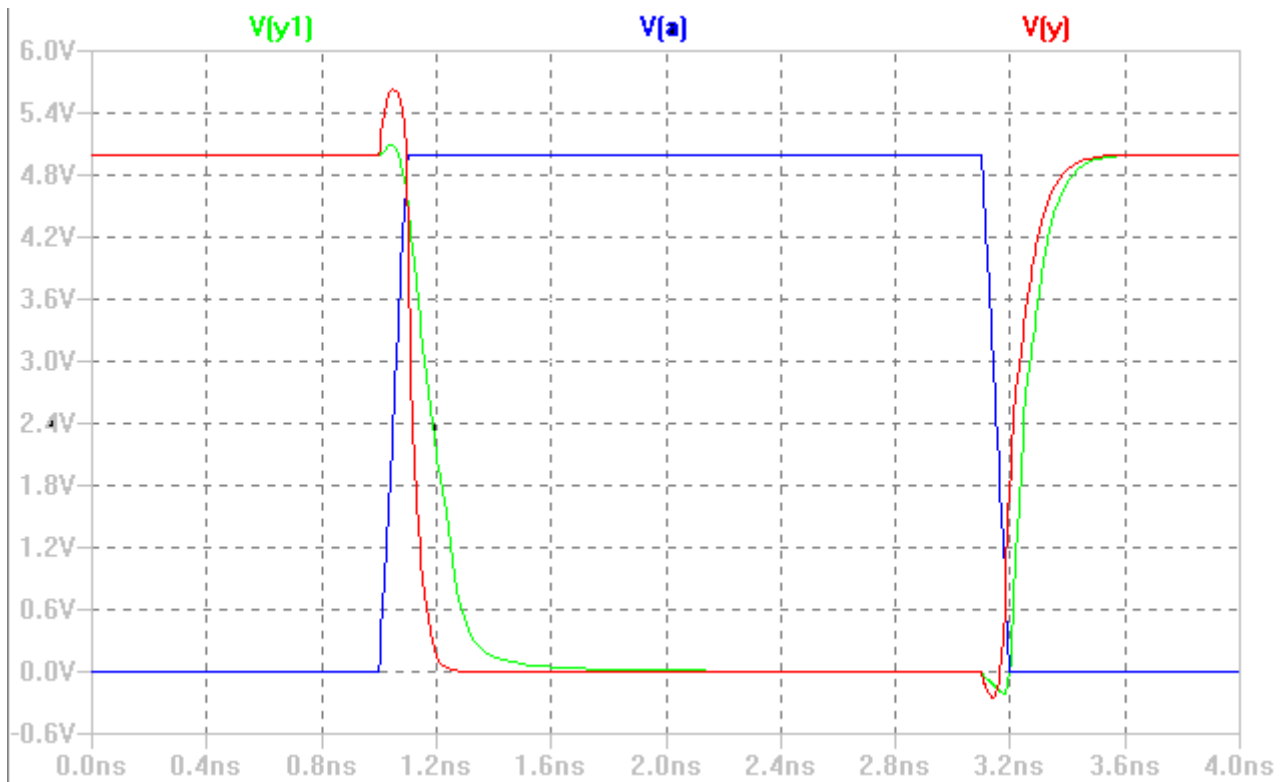


Ottenendo come risultato della simulazione le seguenti caratteristiche:



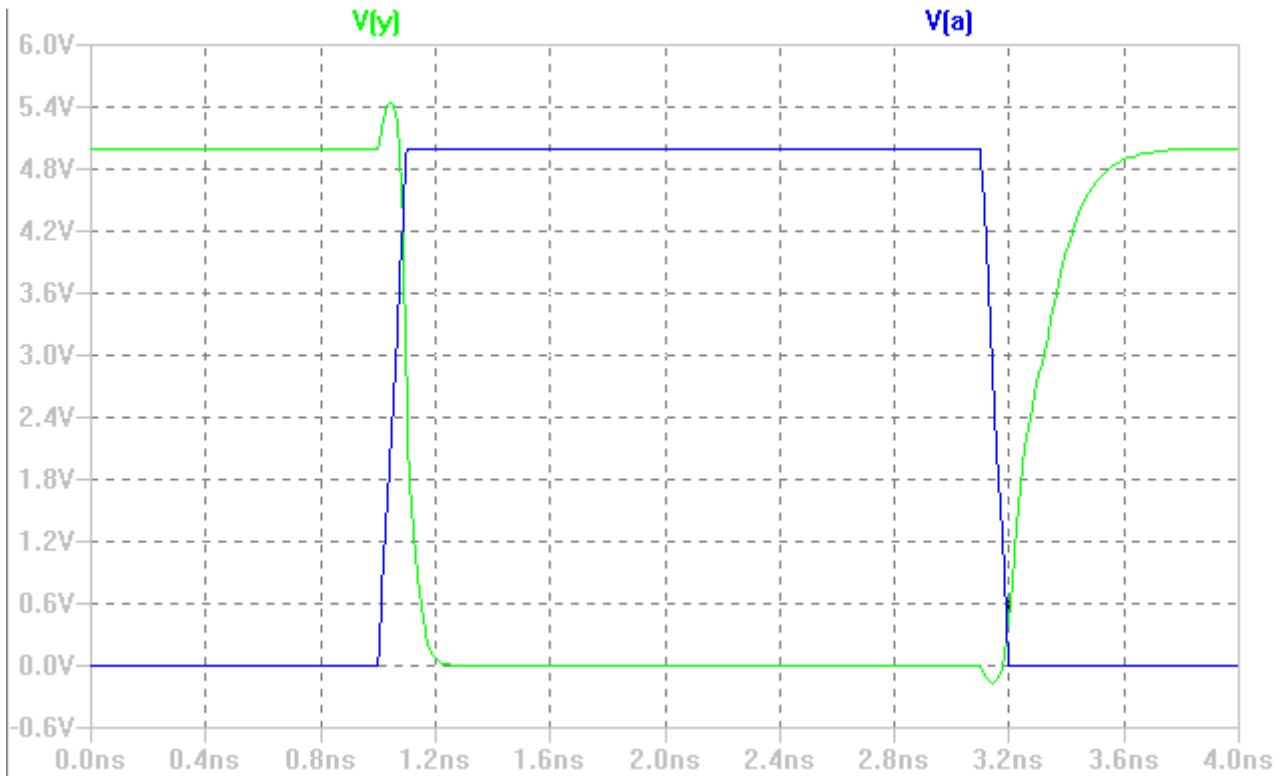
Nella precedente abbiamo visto che la caratteristica di colore verde mostra l'andamento della porta NOR a seguito di due commutazioni, dando la possibilità di stimare il tempo di propagazione. La curva disegnata di colore blu indica invece l'uscita dell'invertitore usato come carico costituito anch'esso da transistori C-MOS

Se tentiamo di confrontare le varie caratteristiche al variare degli ingressi, ci accorgiamo che anche nell'analisi dinamica abbiamo degli shift, come mostra la seguente figura:

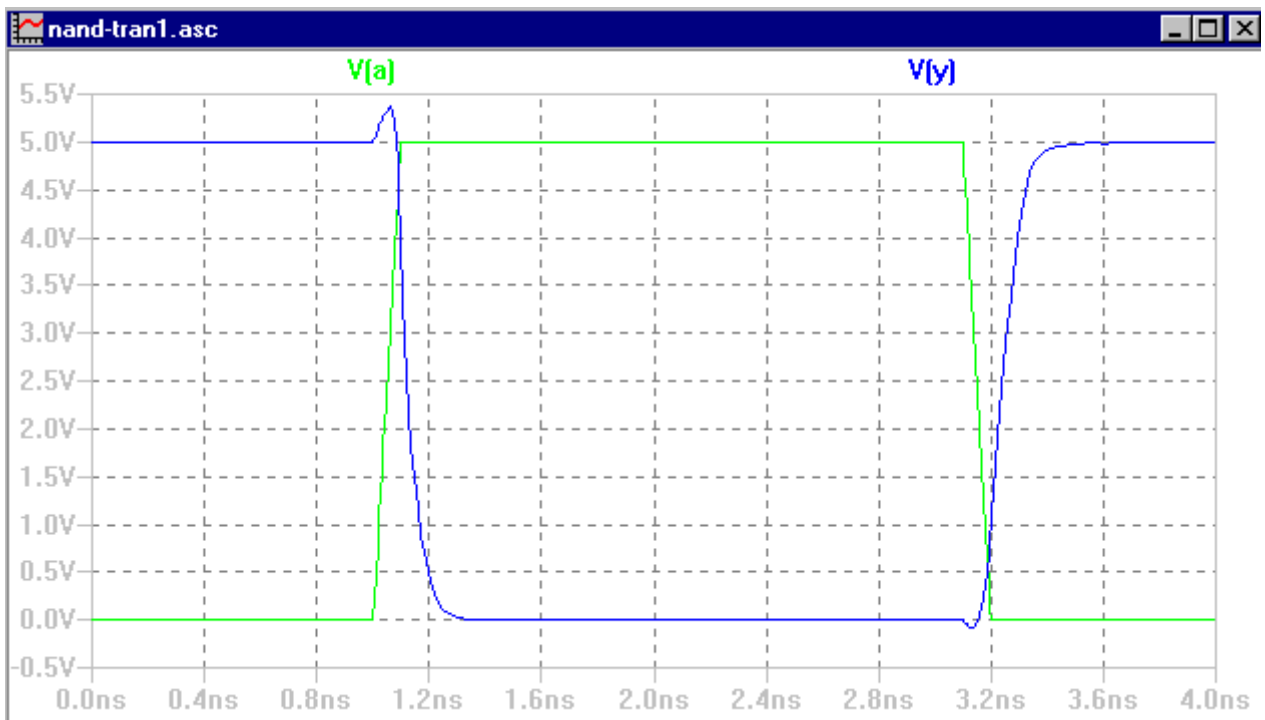


dove $V(y)$ è la caratteristica ottenuta mantenendo alto il valore dell'ingresso A, mentre la curva $V(y1)$ è la caratteristica ottenuta mantenendo alto il valore dell'ingresso C.

Nel caso si prova a modificare le dimensioni dei transistori portando tutti gli N-MOS $W=1,5\mu$ e i P-MOS con $W=3,5\mu$ ci si accorge che i tempi di propagazione diventano ancor più asimmetrici come mostra la seguente figura:



Per quanto riguarda la simulazione per l'analisi dinamica della porta NAND abbiamo riproposto tutte le precedenti prove, ma soffermandoci per lo più su quella dove impostando tutte le dimensioni dei transistor come fatto in precedenza otteniamo la seguente caratteristica:

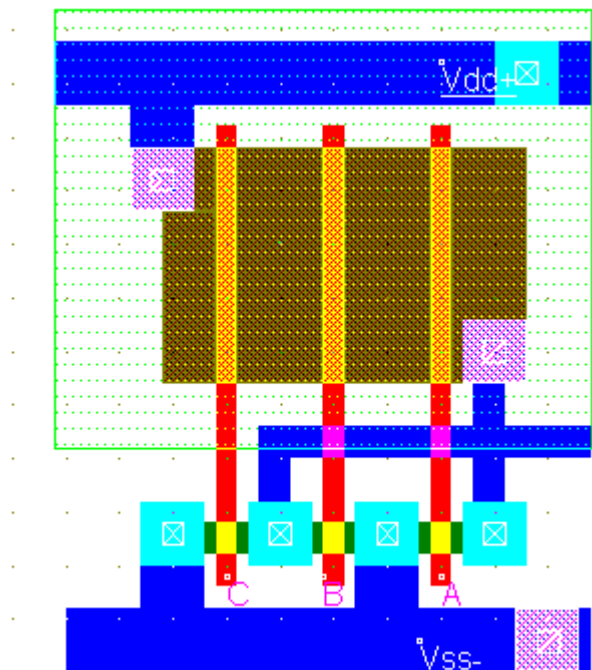


dove è possibile notare che in questo caso a differenza del precedente abbiamo che i tempi di propagazione sono simmetrici, in accordo con le specifiche di progetto.

Anche nel caso della porta NAND il transistor più sfavorito, ovvero che presenta tempi di propagazione più asimmetrici sono quelli collegato per l'ingresso C.

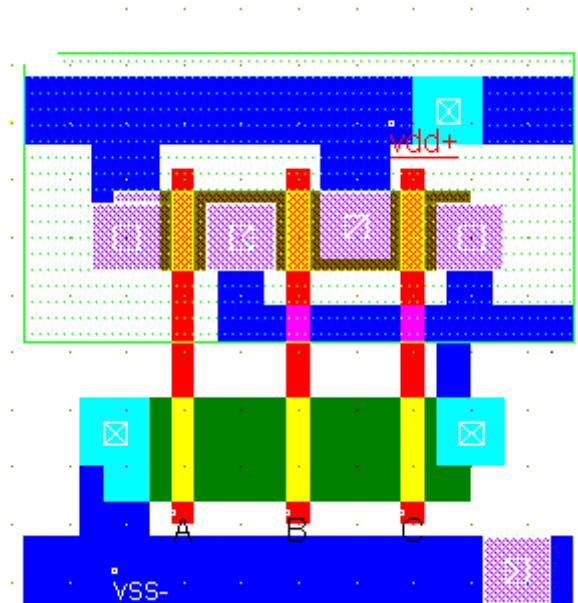
2.3 Progetto Layout

Il circuito NOR a tre ingressi visto precedentemente con gli stessi parametri usati, ha il seguente layout:



I morsetti di ingresso della porta NOR sono accessibili tramite le linee di polisilicio mentre l'uscita, in metal1, è disposta sul lato destro del layout. Dalla figura si possono notare due linee in metal1 orizzontali utili, già incontrate nell'invertitore di cui una dedicata alla tensione di alimentazione (Vdd) e l'altra alla massa. I tre PMOS sono disposti in prossimità della linea di alimentazione; i dispositivi sono in serie e sono realizzati mediante un'unica diffusione di tipo P intersecata da tre linee parallele di polisilicio. In questo modo due PMOS adiacenti condividono la stessa regione diffusa, che costituisce sia il drain di un MOS che il source del MOS adiacente. I MOS a canale N sono collocati vicino alla linea di massa. Anche nella rete di *pull-down*, dispositivi adiacenti condividono la stessa regione diffusa. In questo caso si rende necessaria l'introduzione di contatti in ogni regione di diffusione N in modo da poter collegare i terminali di source o di drain dei dispositivi o a massa o all'uscita.

La porta NAND sempre a tre ingressi ha il seguente layout:



In questo caso gli NMOS sono in serie e sono realizzati mediante un'unica diffusione N intersecata da tre linee parallele di polisilicio. Per i PMOS è necessaria l'introduzione di contatti in ogni regione diffusa per collegare i terminali di source o di drain dei dispositivi o al terminale di alimentazione o all'uscita.

Con questo si ritiene conclusa la prova sulle porte logiche introducendo a presto i Flip Flop.

3 FLIP FLOP JK

3.1. Cosa sono i Flip Flop?

Il flip flop è un elemento circuitale che si fa da intermediario tra la logica combinatoria e la logica sequenziale, anche se viene classificato come elemento sequenziale per le sue nature.

Prima di addentrarci in quelli che sono i dettagli dei flip flop conviene soffermarsi sulle definizioni di circuiti combinatori e sequenziali.

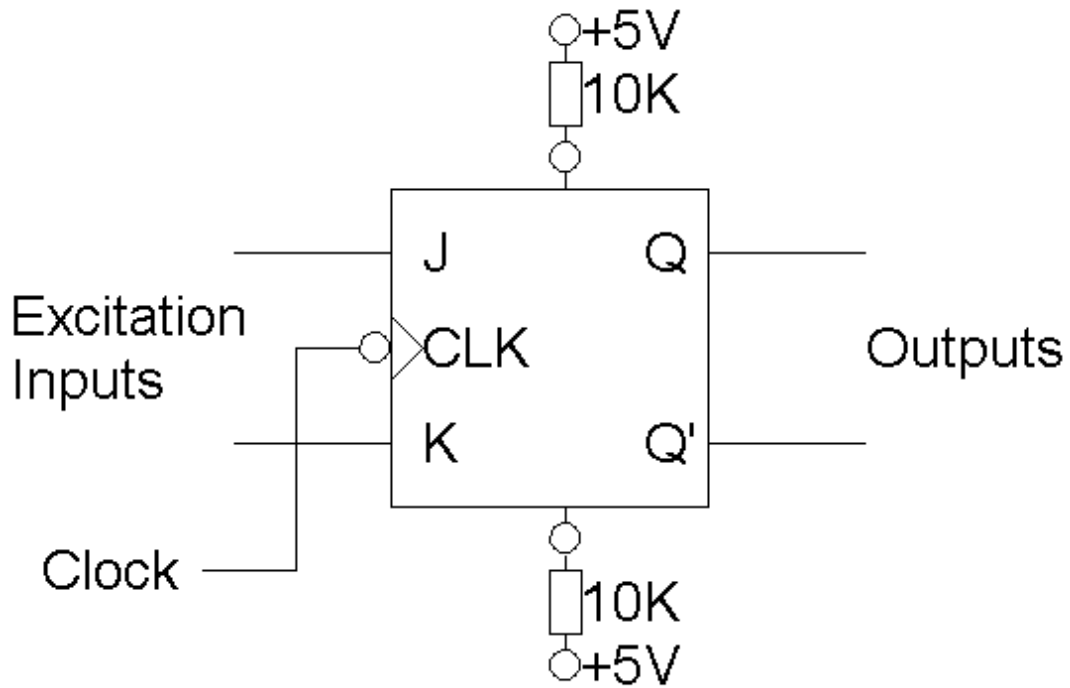
Def:

E' definito circuito combinatorio è un sistema logico la cui uscita dipende unicamente dagli ingressi allo stesso istante a meno dei ritardi dovuti alla propagazione dell'informazione.

A differenza del precedente un circuito sequenziale è una rete più complessa ed in particolare si ha che l'uscita dipende dagli stati logici attuali e dagli stati precedenti, definito a sua volta come circuito dotato di memoria.

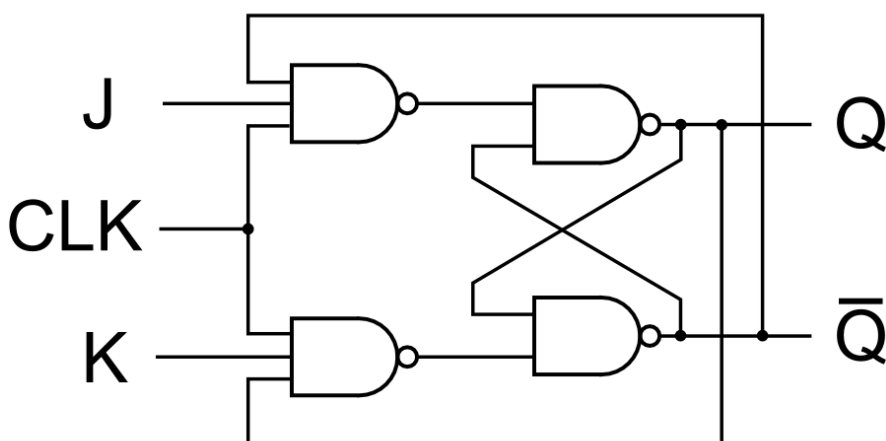
La configurazione assunta dalle uscite di un circuito sequenziale dovrà essere determinata non solo dalla configurazione degli ingressi come un circuito combinatorio, ma si dovranno prendere in considerazione anche la configurazione dello stato precedente.

Ritornando ai flip flop si afferma che sono in grado di memorizzare un bit ed esistono svariati tipi di flip flop che differiscono sostanzialmente dal metodo di memorizzazione.



Nella precedente figura è mostrato il simbolo del flip flop. Il flip flop al suo interno può essere costituito da porta logiche spiegando la funzione da intermediario tra due logiche, oppure da componenti attivi discreti come il BJT e FET.

Ecco il circuito di un flip flop ottenuto con porte logiche, insieme ad un package del flip flop:

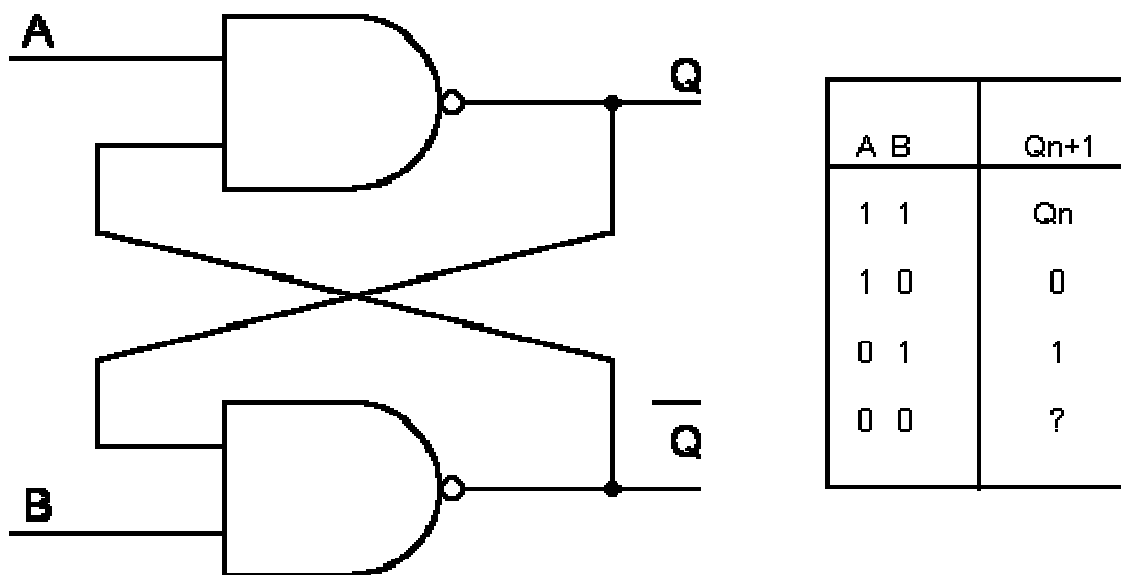


Tornando un po' indietro circa i circuiti combinatori abbiamo che questi ultimi possono essere realizzati in due modi:

- Retroazione
- Con elemento capacitivo

I circuiti con elementi capacitivi sono in gradi di memorizzare dati grazie alla presenza di una capacità posta in uscita.

I circuiti con retroazione riguarda proprio il nostro caso dove vi si presenta un latch SR, come mostra la seguente figura:



Questo latch detto anche latch SR è caratterizzato da una retroazione cosiffatto in grado di memorizzare una informazione binaria a un bit. Esso consta in due ingressi A(set) e B(reset) e di due uscita una la negata dell'altra. Non appena vi si applica un ingresso a tale circuito ci sarà un breve transitorio di varie combinazioni d'uscita, fino a quando non si stabilizza al valore finale.

Possiamo sfruttare la tabella per conoscere lo stato futuro della nostra macchina, solo che nella configurazione "00" abbiamo una situazione di continua instabilità, per questo viene apportata una leggera modifica a questo latch riportandoci a quello che è il vero è proprio circuito flip flop jk, su mostrato.

Il nostro flip flop Jk in questione ha la seguente tabella di verità:

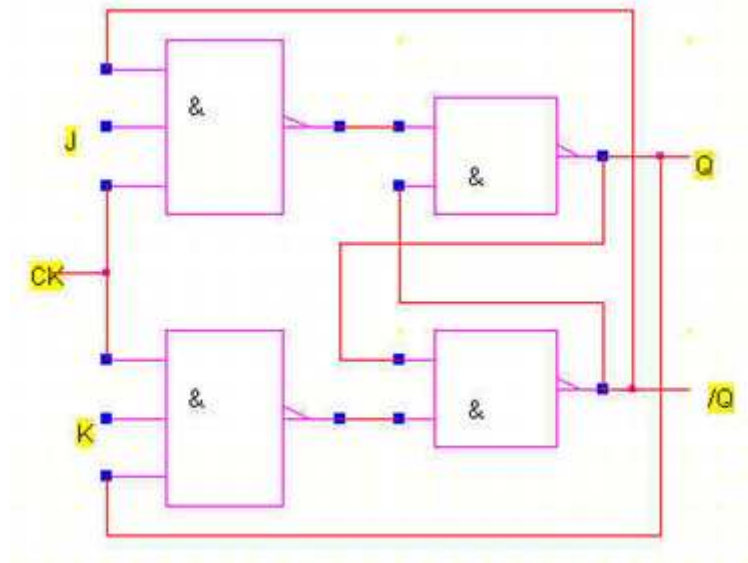
JK Flip Flop									
Tabella delle caratteristiche					Tabella di eccitazione				
J	K	Q_{next}	Commento	Q	Q_{next}	J	K	Commento	
0	0	Q_{prev}	Mantiene lo stato	0	0	0	X	Non cambia	
0	1	0	reset	0	1	1	X	Set	
1	0	1	set	1	0	X	1	Reset	
1	1	Q_{prev}	Commuta	1	1	X	0	Non Cambia	

dove è possibile capire lo stato futuro dagli stati attuali oppure gli stati precedenti dall'uscita attuale.

Dopo aver spaziato un po' su quelli che sono i cenni teorici riguardanti i flip flop si è dediti alla simulazione.

3.2 Descrizione a livello logico

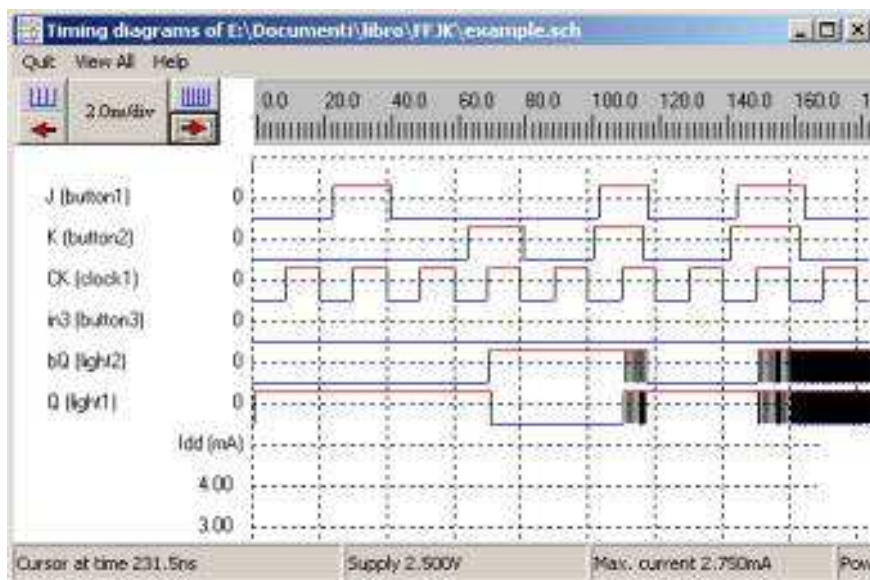
Questa parte di simulazione comprende un po' quello che è stato detto in precedenza, circa la composizione interna del flip flop. Di seguito viene mostrata una figura circa la circuiteria interna del flip flop:



Si nota dalla “&” negli oggetti e dal triangolino in uscita degli stessi che il circuito è fatto in Nand-Nand, poi c’è un clock che sincronizza le operazioni con i segnali J e K, in modo tale da fornire le uscite Q e \bar{Q} secondo le tabelle di verità dettate in precedenza.

Il modello precedente possiede le riporti mediante delle retroazioni le due uscite in ingresso, come già spiegato risolvendo il problema della meta stabilità nel caso in cui il flip flop cada nella combinazione “11” ricadendo in uno stato non predicibile detto corsa critica.

Avviata la simulazione con il programma dei grafici, Timing Diagrams, otteniamo tutte le curve di nostro interesse come mostra la seguente figura:



Dove sono riportati gli ingressi come il clock, generato automaticamente da un oggetto circuitale apposito, e gli ingressi J e K abilitati mediante dei pulsanti.

Si nota che il grafico presenta in uscita diagrammi chiari di comprensione ad eccezion fatta quando J e K sono entrambi ad un livello logico alto dove c'è un'oscillazione continua.

3.3 Descrizione a livello layout

Nel precedente punto abbiamo visto come è possibile ottenere un flip flop attraverso l'uso delle sole porte Nand, con esattezza c'è ne sono due a due ingressi e due a tre ingressi, adesso verrà mostrato a livello progettuale come realizzare un flip flop a livello layout.

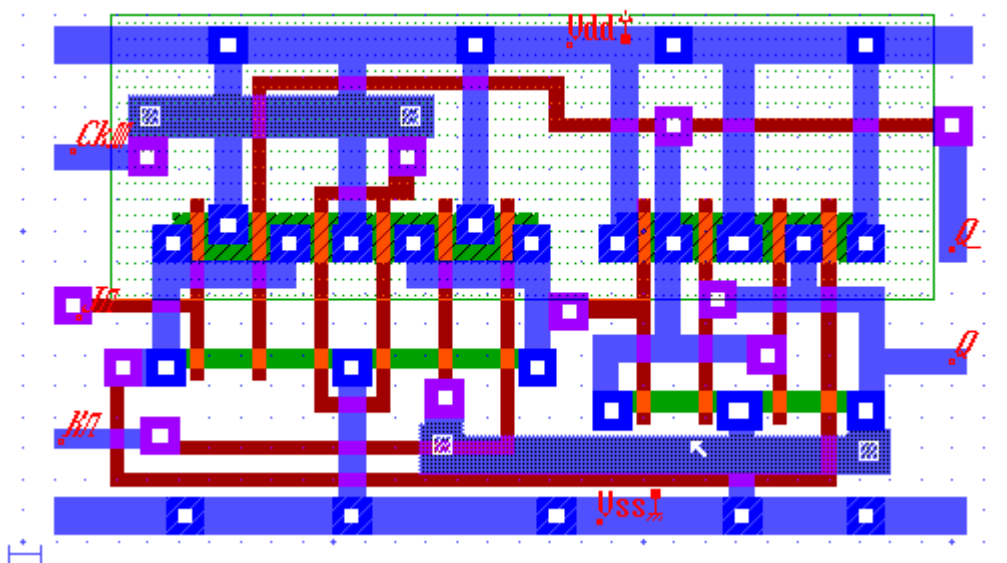
Innanzitutto si parte con il fare delle precisazione ovvero imponendo che le porte Nand a due ingressi presenti un tempo di propagazione(t_{plh}) pari a 1ns quando è caricata da una capacità di 0,5 pF, mentre le Nand a tre ingressi dovranno avere un t_{plh} non superiore ai 200ps considerando che ha come carico proprio le porte nand a due ingressi.

Mettendoci nel caso peggiore delle configurazioni otteniamo:

Porta Nand 3 ingressi	NMOS	PMOS
W	3λ	8λ
L	2λ	2λ

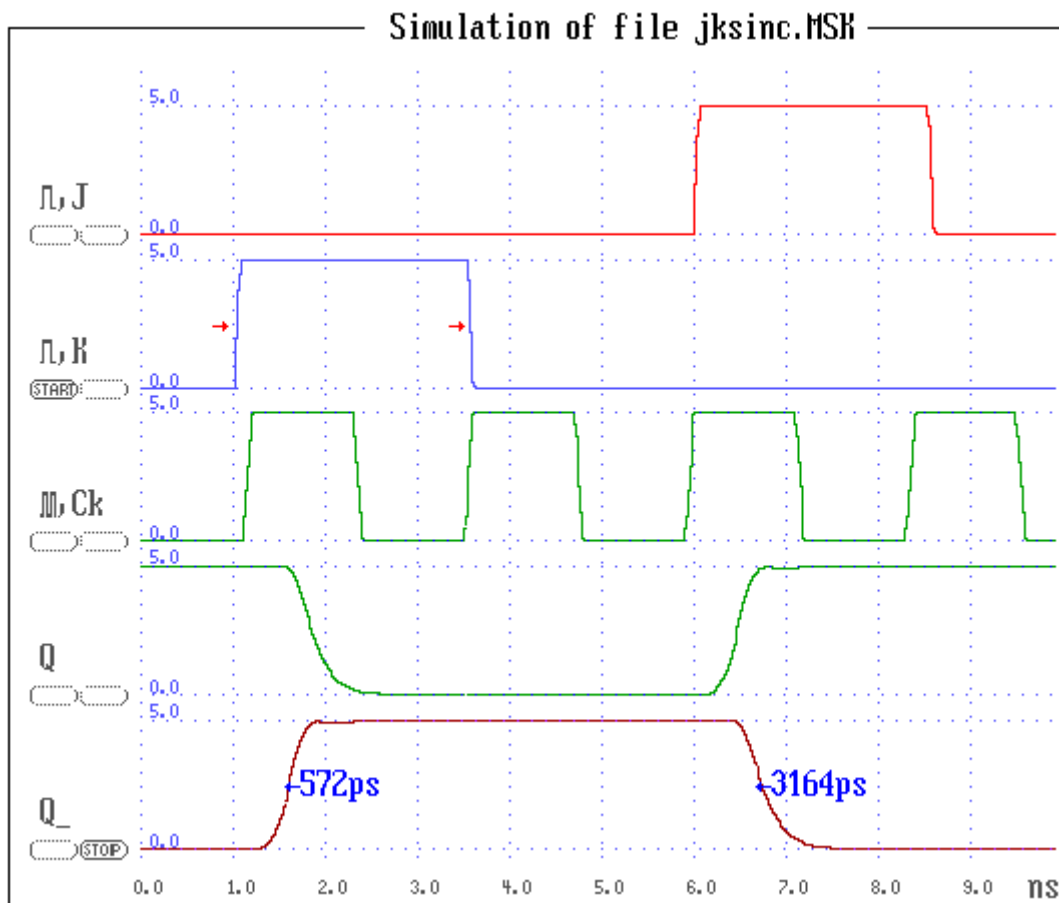
Porta Nand 2 ingressi	NMOS	PMOS
W	3λ	8λ
L	2λ	2λ

Alla base dei precedenti calcoli effettuati, andiamo a combinare i vari elementi per formare il flip flop, introducendo tutti i vari transistor, piste di collegamento e piste per la tensione cercando di ottimizzare gli spazi per ottenere il seguente circuito:



3.3.1. Analisi degli stati logici

Mandando in simulazione il precedente elaborato attraverso il MICRO otteniamo i seguenti grafici:



E' stato scelto un segnale di clock opportuno mentre i due segnali d'ingresso J e K vengono scelti in modo tale da non trovarsi mai contemporaneamente al livello logico alto.

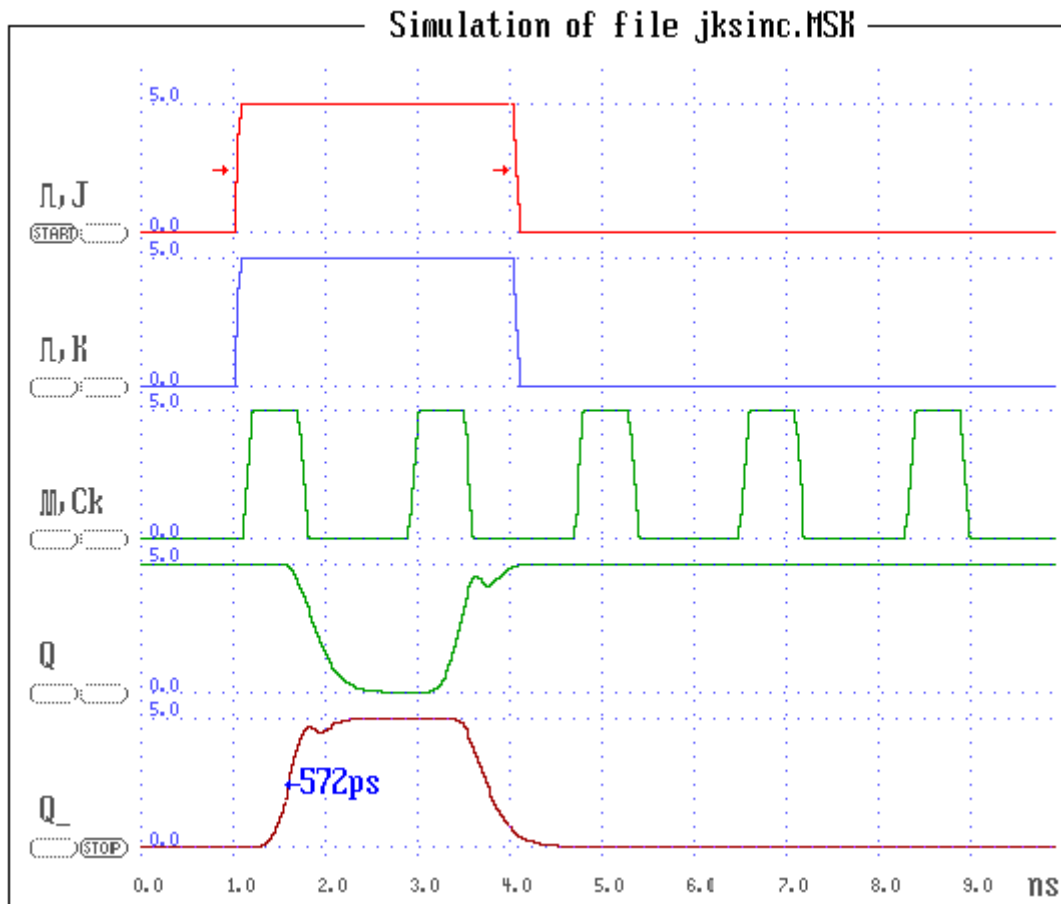
Da notare la coerenza tra le tabelle di verità e i grafici temporali ottenuti:

$$\begin{cases} J = 0 \\ K = 1 \end{cases} \Rightarrow \text{Operazione di Reset: } \begin{cases} Q = 0 \\ \bar{Q} = 1 \end{cases}$$

$$\begin{cases} J = 1 \\ K = 0 \end{cases} \Rightarrow \text{Operazione di Set: } \begin{cases} Q = 1 \\ \bar{Q} = 0 \end{cases}$$

3.3.2. Analisi Dinamiche

In questa parte si va a controllare la situazione delle commutazioni a ridosso della configurazione "11". Ottenendo il seguente grafico:



Il segnale immesso attraverso i due ingressi dura almeno due colpi di clock, e per quanto riguarda l'uscita abbiamo che :

$$\begin{cases} J = 1 \\ K = 1 \end{cases} \Rightarrow \text{Operazione di Toogle: } \begin{cases} Q = \bar{Q}_{prec} \\ \bar{Q} = Q_{prec} \end{cases}$$

Si nota dal grafica che proprio in prossimità di questo tipo di configurazione abbiamo delle situazioni instabili prima di raggiungere la completa stabilità e quindi la commutazione rispetto allo stato precedente.

Si dimostra che all'aumentare della frequenza di clock, implicante della riduzione della durata dell'impulso di clock, si ha che il sistema non è più in grado di reagire e quindi di far sì che avvenga una commutazione tra lo stato presente e lo stato futuro. Tali problemi sono risolvibili

attraverso l'uso di circuiti dello stesso tipo ma più complessi come la configurazione basata sullo schema master-slave.

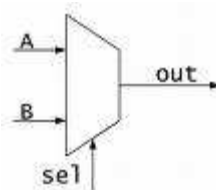
Dopo le precedenti prove si passa all'esercitazione che vede sotto esame un multiplexer.

4. MULTIPLEXER

4.1. Cos'è un multiplexer?

Un multiplexer, comunemente chiamato MUX è un circuito combinatorio, formato al suo interno da porte logiche. Esso ha la caratteristica di trasferire in uscita uno dei segnali presenti in ingresso. Chi gestisce questa preferenza sono i selettori. In definitiva per n selettori ci potranno essere 2^n ingressi, ma una sola uscita.

Il multiplexer è raffigurato di seguito:

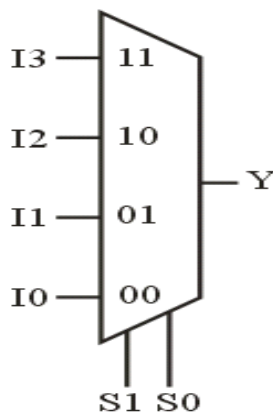


dove il selettore è uno solo, gli ingressi come dettato dalla formula saranno al più due, mentre l'uscita è unica. Il multiplexer raffigurato può avere la seguente tabella di verità:

Ingresso Selettore	Out
0	A
1	B

Dalla tabella si evince che nel momento in cui vi si presenta uno zero sul selettore il multiplexer replicherà il valore d'ingresso presente su A in uscita, stesso discorso per B quando sul selettore vi arriva un 1.

Pensando ad un multiplexer con due selettori abbiamo il seguente:

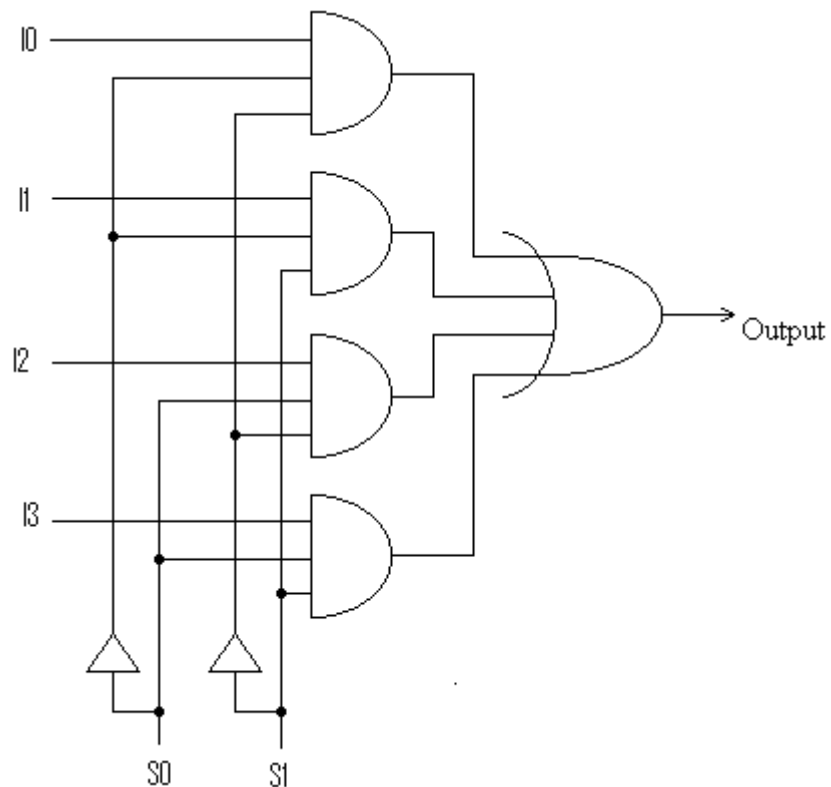


S1	S0	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

E' ovvio che a parità di valore dei selettori si possono scambiare i valori degli ingressi da replicare in uscita, perche questa è una scelta del tutto arbitraria.

La precedente tabella di verità può essere espressa sottoforma di una funzione che ha come incognita l'uscita Y:

avente il seguente circuito combinatorio:

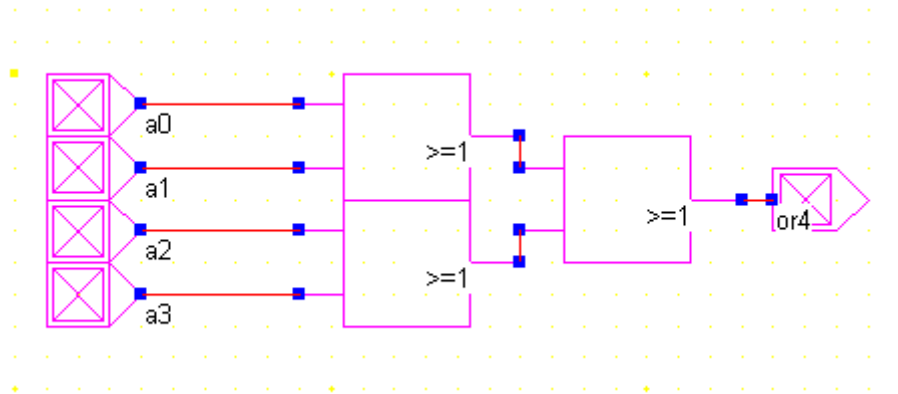


4.2. Simulazione Logica

La simulazione logica comporta l'uso del multiplexer attraverso la costruzione di quest'ultimo con le porte logiche.

Si vuole creare un multiplexer a due selettori e quindi quattro ingressi, come mostra la figura precedente.

Nella realizzazione si incorre in un problema, perché le librerie del software non comprendono una porta OR a 4 ingressi, di conseguenza si procederà alla costruzione della porta stessa come mostra il seguente circuito:



Il circuito è formato da una OR di due OR, con degli oggetti tutti simili detti terminali. Dopo aver attivato il comando :

File → Schema to New Symbol

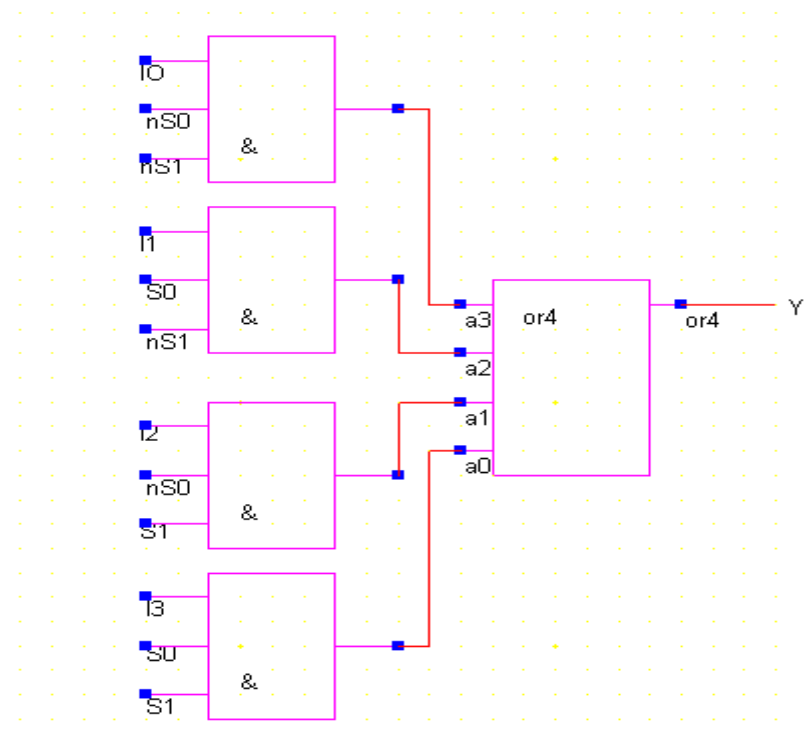
lo salviamo, in modo tale che sia subito utile per il nostro multiplexer.

Dopo questo passo lo andiamo a prelevare dalla directory principale con estensione *.sym e lo inseriamo nell'area di lavoro attraverso il comando:

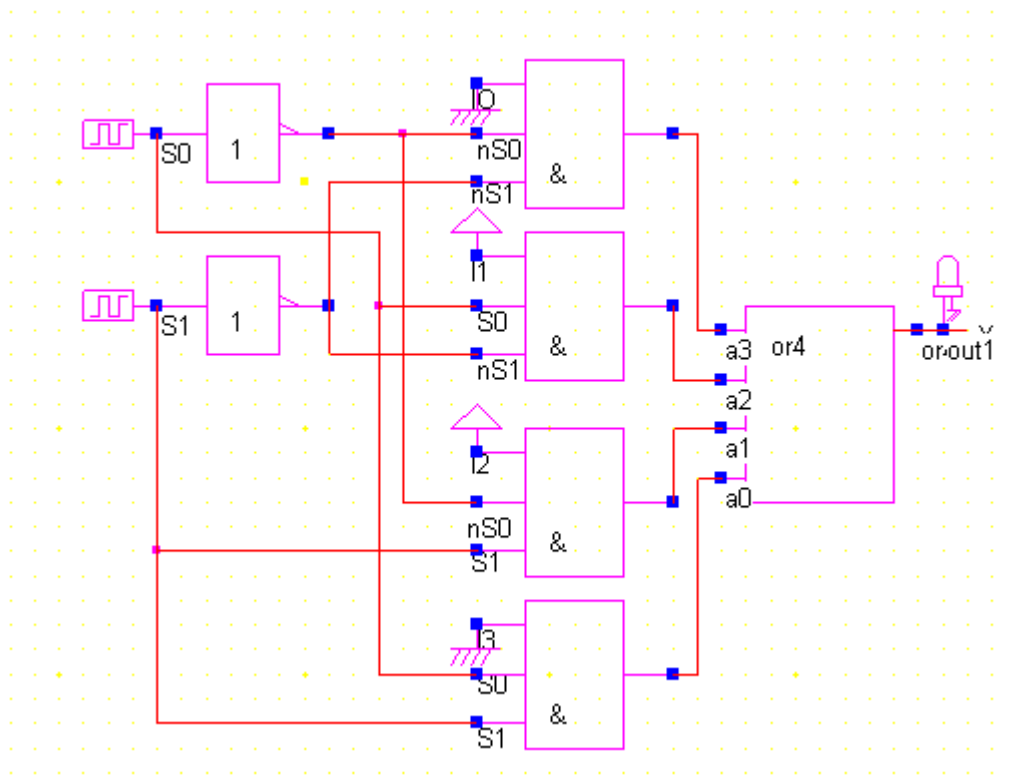
Insert → User Symbol

insieme ad altre quattro porte AND a tre ingressi.

Li disponiamo in modo tale da ottenere il seguente circuito:



Dopo di che lo colleghiamo ad altri apparati esterni per verificarne il funzionamento attraverso la simulazione, come mostra la seguente figura:



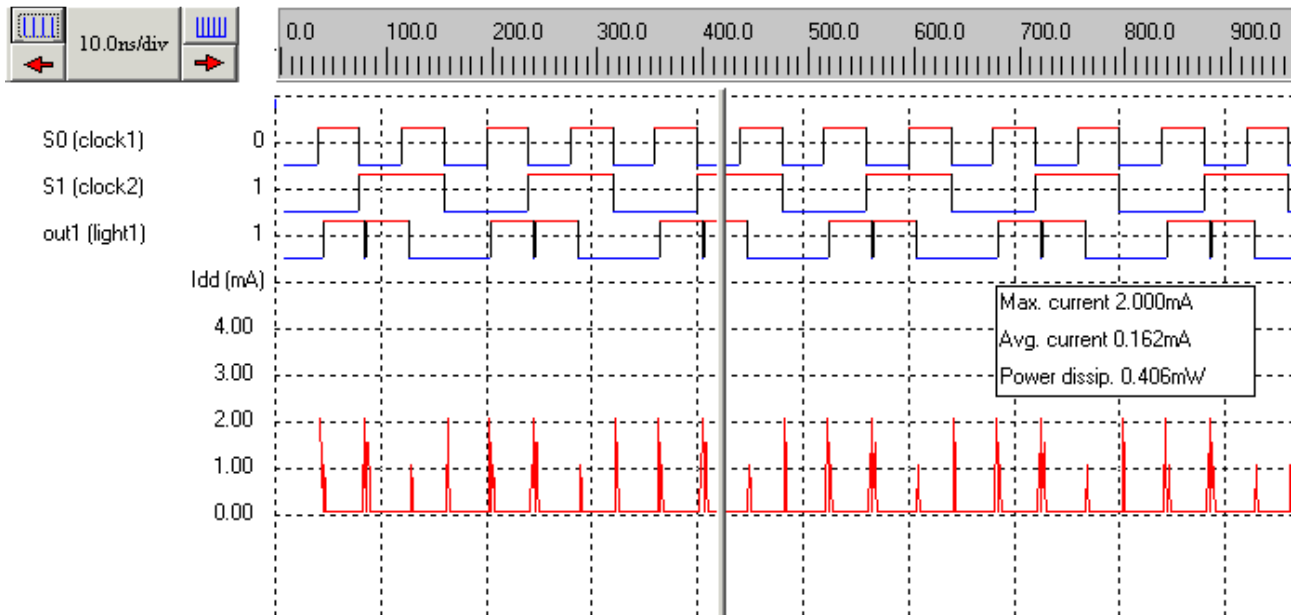
Dove abbiamo inserito due clock in modo tale che fungano da selettori, impostando che il clock collegato in S0 abbia frequenza doppia rispetto al clock inserito su S1, coprendo ciclicamente valori da

$$00 \rightarrow 01 \rightarrow 10 \rightarrow 11$$

I segnali di ingresso invece sono applicati a coppia alti e a coppia bassi, in modo da avere delle combinazioni del tutto casuali.

In uscita è stato inserito un led o spia luminosa che indichi visivamente l'istante in cui l'uscita è alta.

Avviata la simulazione con il programma che effettua in grafici, otteniamo le seguenti risposte:



Le caratteristiche d'uscita sono concentrate sui clock che fungono da ingresso e sull'uscita, che mostra valori altro per ogni volta che attraverso il circuito combinatorio si trasmette il valore alto, il precedente è anche visibile attraverso la spia luminosa, ma essendo alimentata ad alta frequenza non permette di apprezzare al meglio queste variazioni.

Nel riquadro del grafico è possibile notare anche una piccola finestrella dove viene mostrato numericamente il valore della corrente massima e una stima della potenza dissipata.

Con quest'ultimo passo si ritiene conclusa la parte esercitativa sui multiplexer, lasciando spazio alla prossima simulazione.

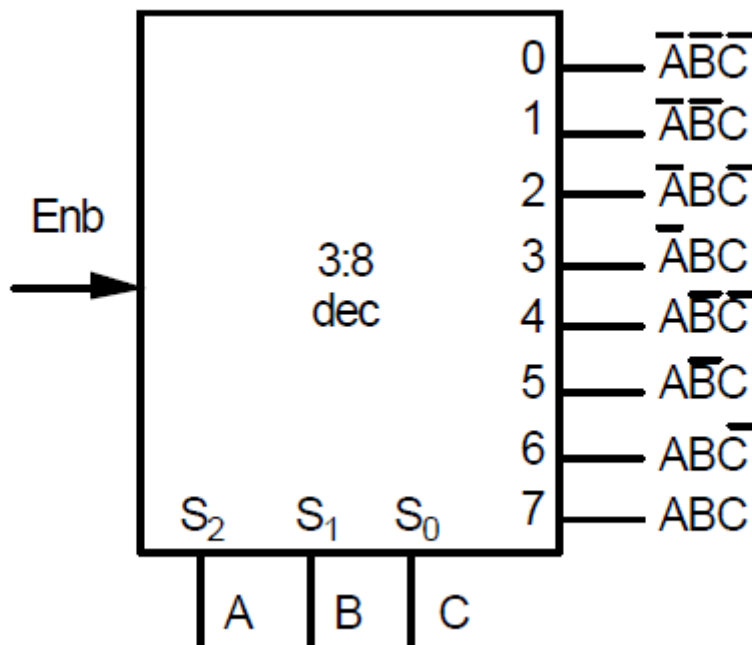
5. IL DECODIFICATORE

5.1. Cos'è il decodificatore?

Il decodificatore, detto anche decoder, è un circuito combinatorio che possiede N ingressi e 2^N uscite.

Data una combinazione di bit in ingresso il decoder attraverso una serie di transistori, solitamente ad effetto di campo, fornirà un valore alto a una delle 2^N uscite.

Nella seguente figura è riportato il simbolo logico del decoder:



A	B	C	U0	U1	U2	U3	U4	U5	U6	U7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

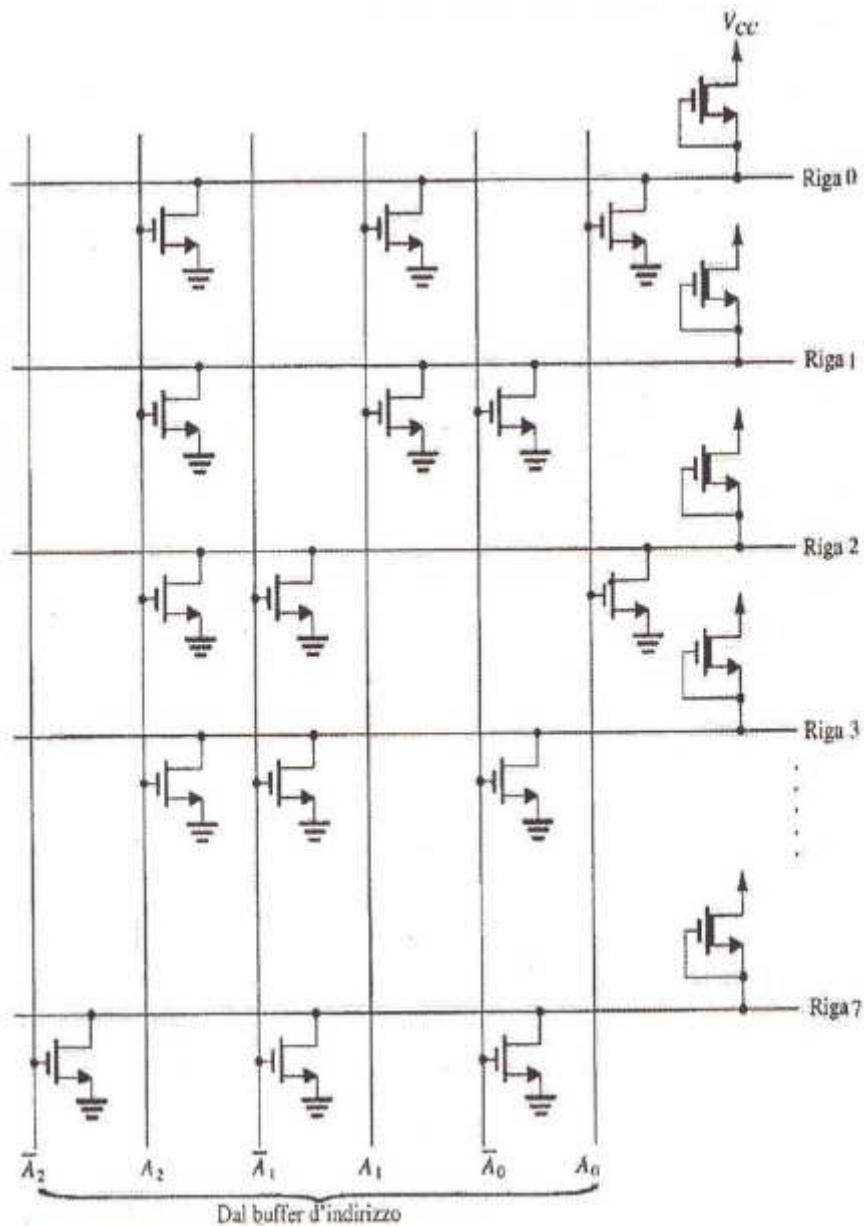
Nella precedente tabella di verità viene mostrato quanto su detto, ovvero che per ogni combinazione di bit in ingresso si ottiene il valore alto su una delle uscite.

Il decoder è in grado di convertire il valore in ingresso ad esempio binario in un numero decimale come nel seguente caso:

D C B A	 	0	1	2	3	4	5	6	7	8	9
0 0 0 0	 	1	0	0	0	0	0	0	0	0	0
0 0 0 1	 	0	1	0	0	0	0	0	0	0	0
0 0 1 0	 	0	0	1	0	0	0	0	0	0	0
0 0 1 1	 	0	0	0	1	0	0	0	0	0	0
0 1 0 0	 	0	0	0	0	1	0	0	0	0	0
0 1 0 1	 	0	0	0	0	0	1	0	0	0	0
0 1 1 0	 	0	0	0	0	0	0	1	0	0	0
0 1 1 1	 	0	0	0	0	0	0	0	1	0	0
1 0 0 0	 	0	0	0	0	0	0	0	0	1	0
1 0 0 1	 	0	0	0	0	0	0	0	0	0	1
1 0 1 0	 	0	0	0	0	0	0	0	0	0	0
1 0 1 1	 	0	0	0	0	0	0	0	0	0	0
1 1 0 0	 	0	0	0	0	0	0	0	0	0	0
1 1 0 1	 	0	0	0	0	0	0	0	0	0	0
1 1 1 0	 	0	0	0	0	0	0	0	0	0	0
1 1 1 1	 	0	0	0	0	0	0	0	0	0	0

Oltre al precedente, il decoder è in grado di convertire da bcd a decimale (4 a 10), da binario a ottale (3 a 8), da binario esadecimale (4 a 16), codice da gray a decimale, etc.

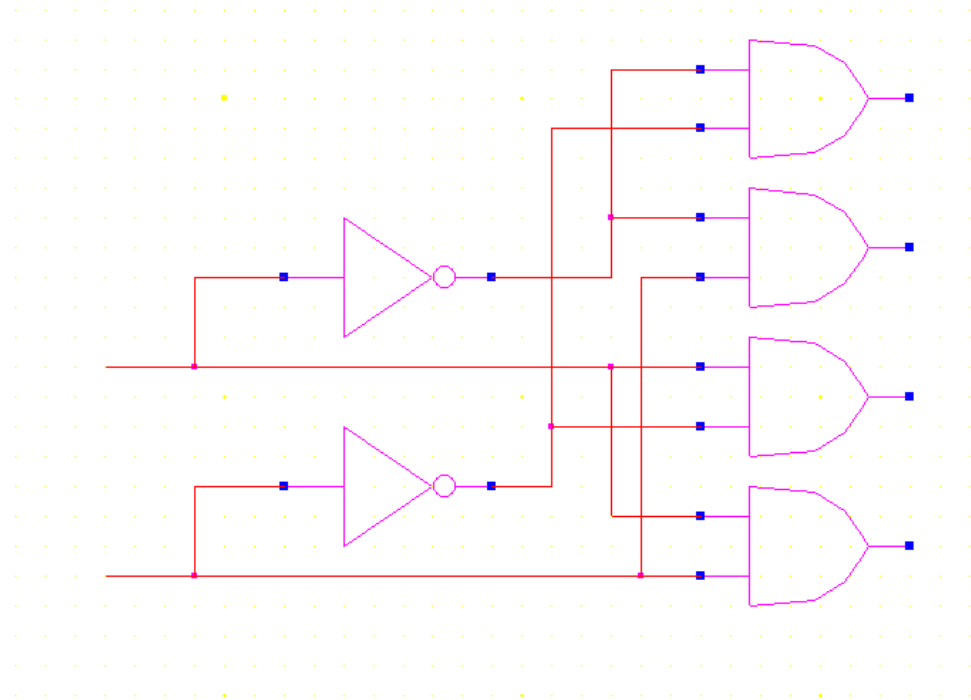
Nella seguente figura viene riportato lo schema circuitale a livello transistor:



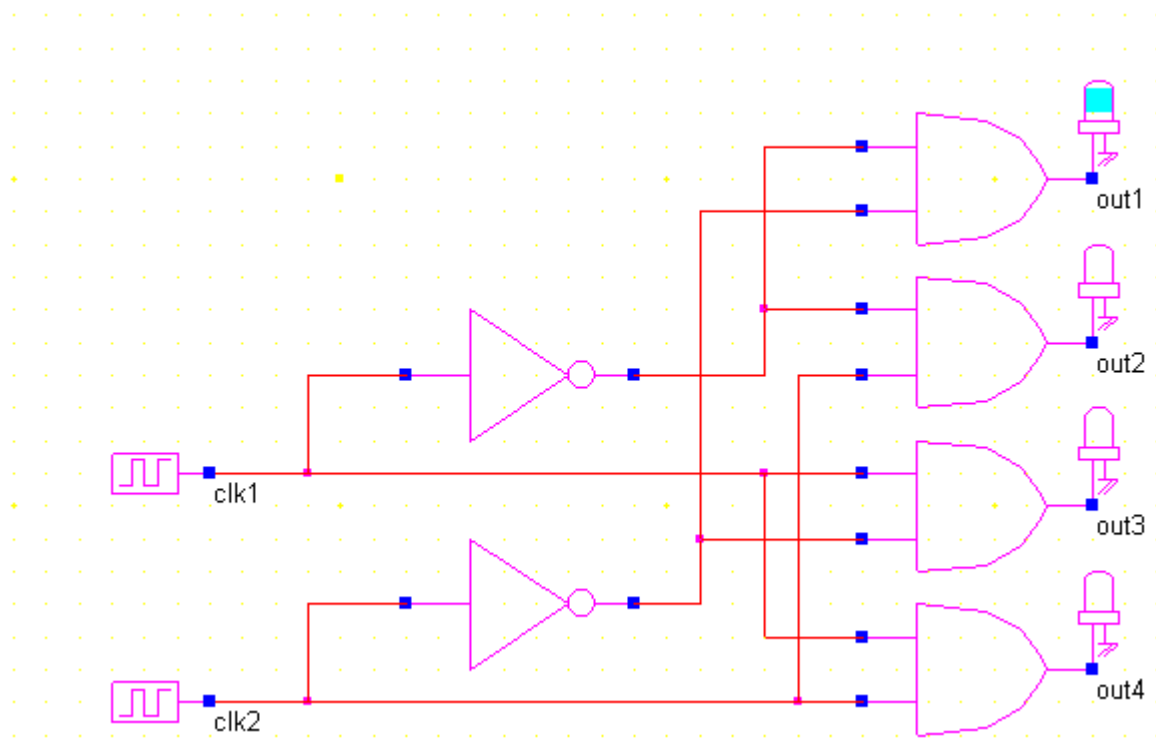
Dove è abbastanza evidente che alla base della parola in ingresso (*Dal buffer d'indirizzo*), una delle 2^N uscite, già prevaricate non vengono scaricate a terra attraverso un transistor.

5.2. Simulazione Logica

La simulazione logica di questo decoder comprende l'uso dello stesso componente con due ingressi e quattro uscite come mostra la seguente figura:

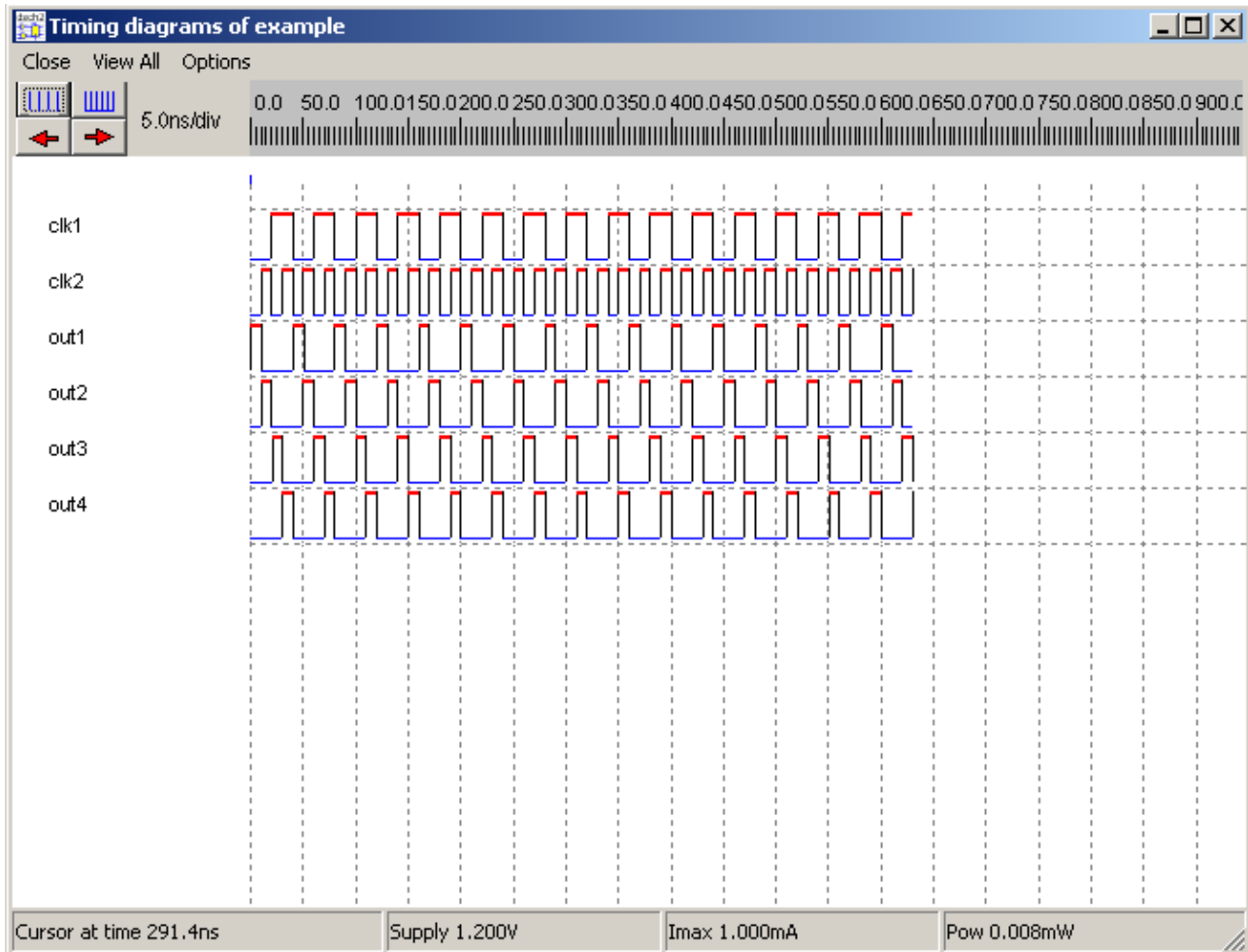


Il precedente circuito verrà collegato a due clock in ingresso uno con frequenza il doppio dell'altro in modo tale da coprire tutte e quattro le combinazioni possibili in ingresso, in uscita sono stati inseriti delle spie luminose(LED), come mostra, la seguente figura:



Nel nostro caso clk2 ha frequenza doppia a clk1, con periodo di 20ns, mentre il clk1 ha un periodo di 40ns.

Avviata la simulazione otteniamo le seguenti uscite:



Dal precedente grafico si evince la seguente tabella di verità:

clk2	clk1	1
0	0	Out1
0	1	Out2
1	0	Out3
1	1	Out4

6. SOMMATORE

6.1. Cos'è il sommatore?

Il sommatore oggetto di questa simulazione è anch'esso un circuito combinatorio, formato da porte logiche. Esso è caratterizzato dall'aver in ingresso due sequenze di bit e in uscita fornisce un'altra sequenza che è la somma delle due ricevute in ingresso, come mostra il seguente esempio:

$$100101 + 001101 = 110010$$

Ne esistono di due tipi:

- half adder
- full adder.

L'half adder, detto anche semisommatore esegue la somma di due bit A e B che arrivano in ingresso e la presenta sull'uscita S. Oltre ad S viene calcolato anche il riporto C, senza tener conto di un riporto precedente. La somma binaria di due bit si calcola allo stesso modo della somma di due numeri decimali, per cui il riporto è generato se e solo se i due addendi sono pari a 1, come mostra la seguente tabella:

A	B	Somma
0	0	0
0	1	1
1	0	1
1	1	10

Nella combinazione 11:

$$\begin{cases} S = 0 \\ C = 1 \end{cases}$$

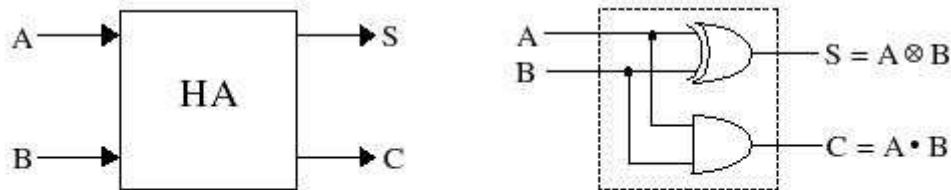
Per tutte le altre configurazioni:

$$\begin{cases} S = \text{Somma} \\ C = 0 \end{cases}$$

A rigore di quanto è stato citato, C è alto quando A=B=1, mentre la S è alta quando i due ingressi sono l'uno il complementare dell'altro, comportandosi come una XOR.

$$\begin{cases} C = A \cdot B \\ S = A \oplus B \end{cases}$$

Il circuito in esame è il seguente:



Simbolo logico e realizzazione circuitale di un HA

Il full adder è una semplice estensione dell'half adder, dove l'insieme di due half-adder e una porta logica OR, opportunamente collegati, restituisce un full-adder. Il full adder esegue la somma tra A_n, B_n e C_{n-1} , dove C_{n-1} è il riporto del blocco di sommatore posto a monte del presente.

In uscita fornisce S_n che è la somma tra i ingressi, e C_n che è il riporto di uscita in funzione sia di A_n, B_n e di C_{n-1} .

Da precisare che un sommatore half adder o full adder riesce a fare la somma di un solo bit per volta ($A + B$), quindi per far la somma di due sequenze ($S1 + S2$) con 6 bit si dovrà disporre di 6 full adder o al più 5 full adder e un half adder.

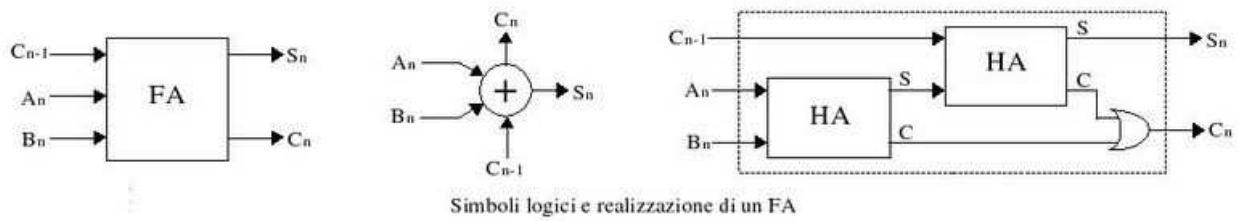
Il full adder ha la seguente tabella di verità:

A	B	C_{n-1}	S_n	C_n
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

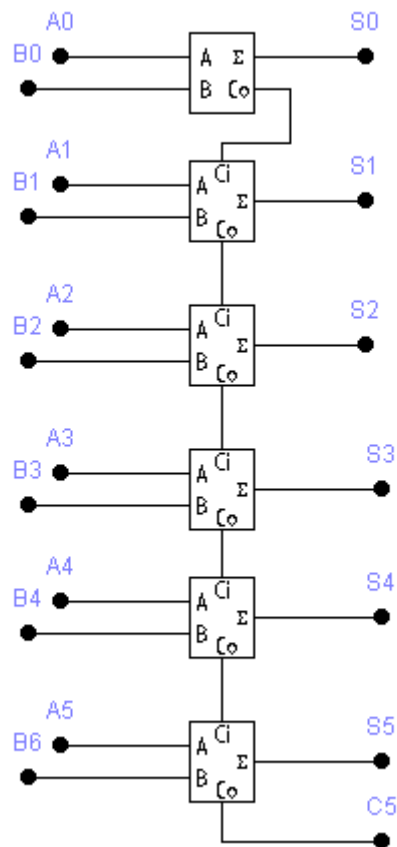
$$\begin{cases} S_n = A \oplus B \oplus C_{n-1} \\ C_n = A \cdot B + C_{n-1}(A \oplus B) \end{cases}$$

Si nota che C_n è la somma dei due riporti.

Ecco il full adder:



Il sommatore necessario per fare la somma di due sequenze a sei bit è il seguente:

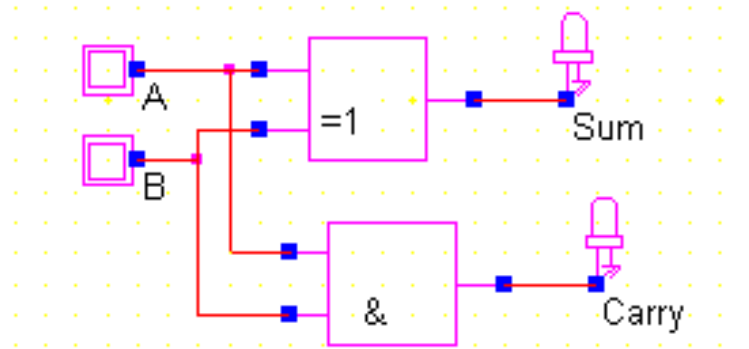


Dove il primo è un half adder perché non necessita di ingresso con riporto, mentre i successivi full adder che devono calcolare la somma in funzione del riporto generato dal blocco precedente.

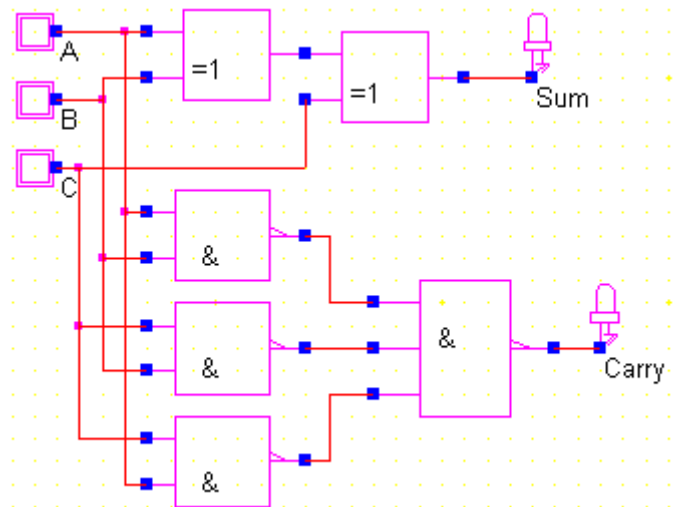
Nella simulazione logica si farà un sommatore a due bit.

6.2. Simulazione logica

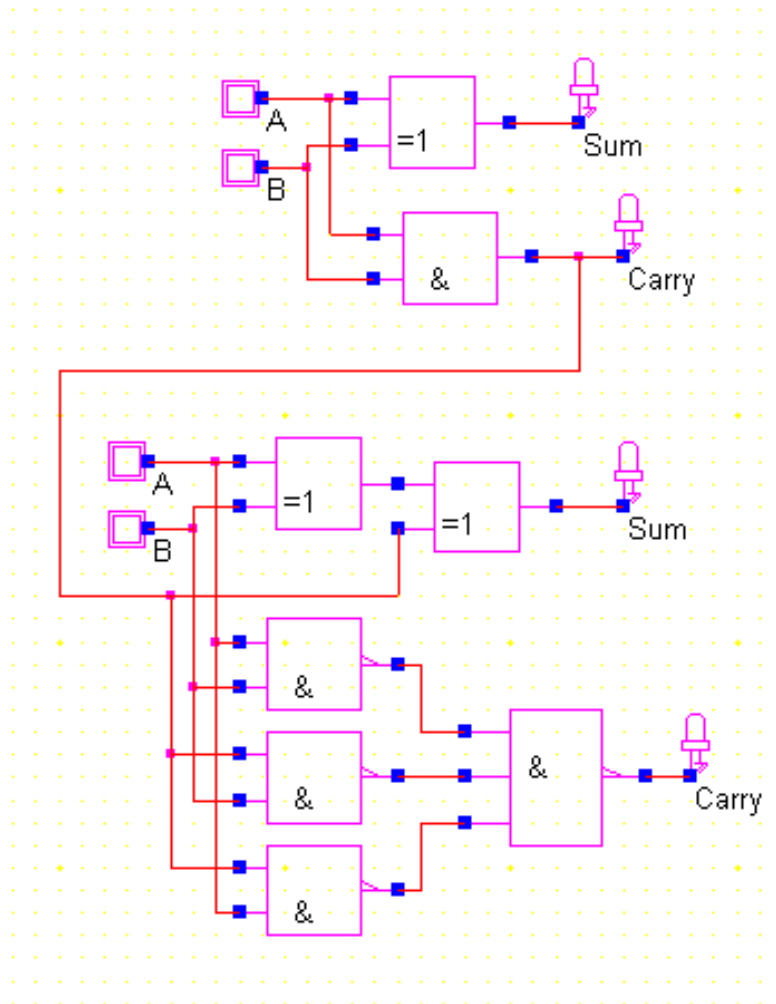
Si simulerà un sommatore di sequenze a due bit, formato da un half adder e un full adder. Per prima cosa si costruisce un half adder:



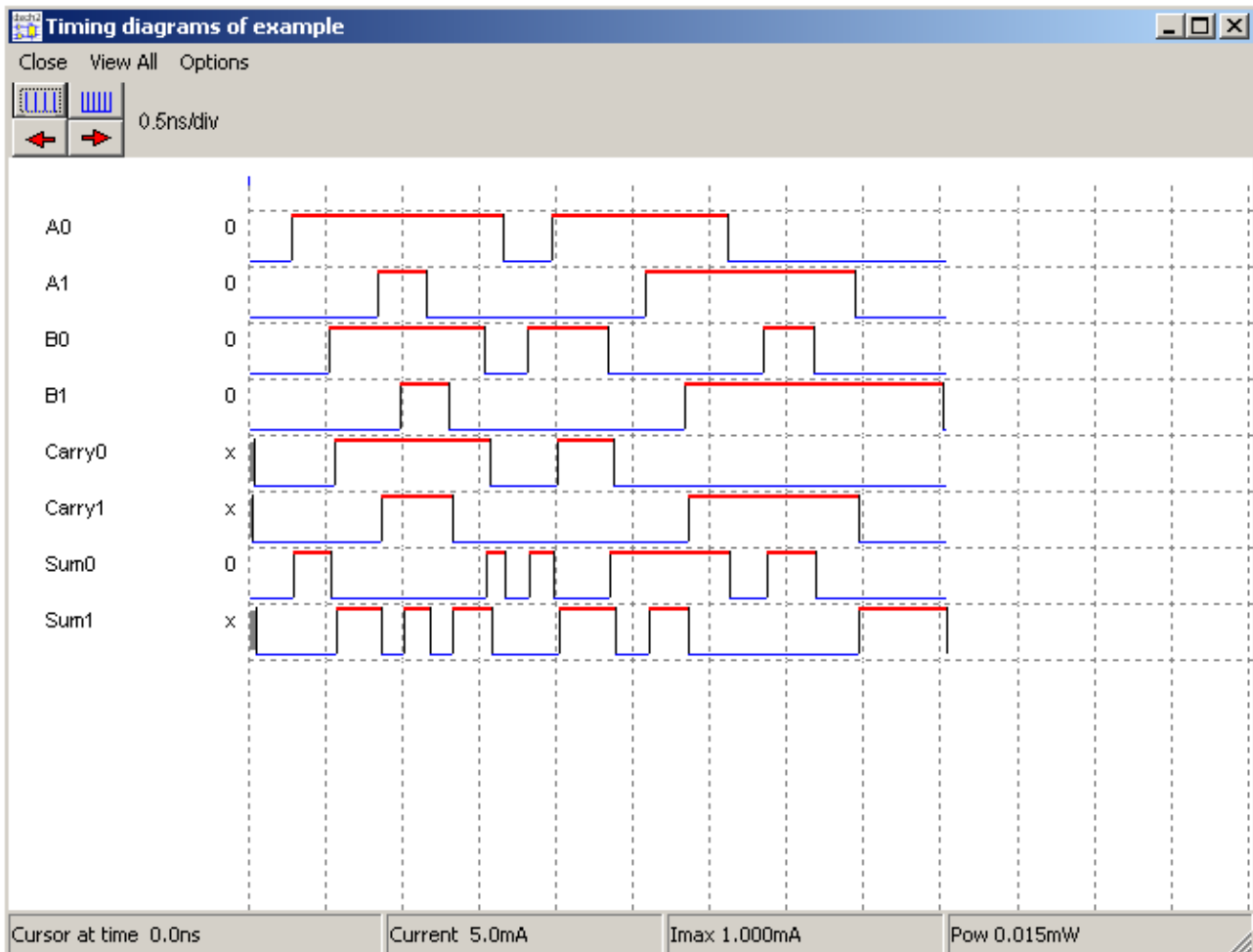
Poi un full adder:



Li componiamo ottenendo il seguente circuito:



Mandandolo in simulazione si ottiene:



Il precedente diagramma temporale mostra che al variare degli ingressi A e B dei due sommatore, si hanno svariate combinazioni di uscita.

Ad esempio quando A0 e B0 sono entrambi alti si vede che Carry0 passa al livello logico alto, stessa considerazione per Carry1. In definitiva il diagramma presentato dimostra quanto specificato teoricamente e rispetta fedelmente le true table.

Al termine di questa prova, si ritiene conclusa la parte relativa al laboratorio di Elettronica 2, dove concedendomi di poter esprimere un giudizio positivo, in merito al lavoro fatto in laboratorio, posso ritenere che la conoscenza e l'esperienza relativa a suddetta disciplina è rafforzata, considerando utile il laboratorio al fine dell'apprendimento.