

# Appunti di Electronics

LEZIONI DEL PROF. FANUCCI AA 2023-2024

SAVERIO MOSTI

## Sommario

INTRODUZIONE .....	6
RECAP DI ELETTRONICA DIGITALE .....	7
Cross-section di un NMOS .....	7
Approssimazione di un NMOS ad uno switch.....	7
Approssimazione di un PMOS ad uno switch .....	8
Pass gate.....	8
Inverter realizzato con Switch e CMOS.....	10
Ritardo di propagazione dell'inverter .....	11
Consumo di Potenza .....	11
PROCESSO DI FABBRICAZIONE DEI CMOS.....	12
Qualità del processo costruttivo .....	12
Litografia .....	12
Photolithography.....	13
Cross-section e layout dell 'inverter .....	13
FABBRICAZIONE DI CIRCUITI INTEGRATI .....	15
Wafer .....	15
Ossidazione .....	15
Photoresist .....	15
Litografia.....	16
Incisione .....	16
Rimozione del photoresist.....	16
Inserimento dello "n-well" .....	17
Rimozione dell'ossido .....	17
Inserimento polisilicio .....	17
Litografia del polisilicio .....	17
Self-Aligned Process .....	18
N-diffusion .....	18
P-Diffusion .....	18
Contatti .....	19
Metallizzazione .....	19
REGOLE DI DESIGN .....	20
Why Have Design Rules?.....	20

ELECTRONIC ABSTRACTION LEVELS.....	20
INTEGRATED CIRCUIT DESIGN STYLE .....	20
FULL-CUSTOM DESIGN .....	21
<b>Specifiche del progetto</b> .....	21
<b>Progettazione del circuito</b> .....	21
<b>Layout del circuito</b> .....	21
<b>Verifica e simulazione</b> .....	22
Design Rule Check [DRC] .....	22
Layout vs Schematic [LVS].....	23
Layout Parasitic Extraction [LPE] .....	23
SEMI-CUSTOM DESIGN .....	24
<b>Gate Array Structure</b> .....	24
Matrici di porte logiche .....	24
Personalizzazione limitata .....	24
Vantaggi della Gate Array Structure .....	24
<b>Standard Cell Structure</b> .....	25
Librerie di celle standard .....	25
Personalizzazione delle interconnessioni .....	25
Vantaggi della Standard Cell Structure .....	25
Svantaggi della Standard Cell Structure.....	25
ASIC (APPLICATION-SPECIFIC INTEGRATED CIRCUIT).....	26
<b>Single gate Structure</b> .....	26
<b>Gate array interconnection</b> .....	26
Matrice di porte Logiche programmabili .....	26
Interconnessione programmabile .....	26
Efficienza e Flessibilità .....	26
<b>Sea-of-gate structure</b> .....	27
Matrice Regolare .....	27
Interconnessione Programmabile .....	27
Efficienza.....	27
Applicazioni .....	27
LOGIC SYNTHESIS.....	28
<b>Register Transfer Level [RTL]</b> .....	28
<b>Place and Route</b> .....	28

<b>Costo di un circuito integrato</b> .....	<b>28</b>
Non-Recurrent engineering Cost [NRE].....	28
Recurrent Engineering Cost [RE] .....	28
Costo totale di produzione .....	29
<b>FPGA (FIELD PROGRAMMABLE GATE-ARRAY UNIT)</b> .....	<b>30</b>
Da cosa sono fatti? .....	30
Vantaggi di FPGA .....	30
<b>Programmare un FPGA</b> .....	<b>30</b>
FPGA SRAM Base (Xilinx) .....	30
FPGA basato su tecnologie anti-fused .....	32
Segnale di clock negli FPGA .....	33
<b>DSP (DIGITAL SIGNAL PROCESSOR)</b> .....	<b>33</b>
<b>RIASSUNTO DEGLI STILI DI DESIGN DEI CIRCUITI INTEGRATI</b> .....	<b>34</b>
<b>The “Energy-Flexibility” Gap</b> .....	<b>34</b>
<b>SoC (System-on-Chip)</b> .....	<b>35</b>
<b>LA FOUNDRY LIBRARY</b> .....	<b>35</b>
<b>TIPOLOGIE DI LOGICA</b> .....	<b>36</b>
Logica combinatoria .....	36
Logica sequenziale .....	36
Macchina di Mealy .....	36
Macchina di Moore.....	37
<b>NODI</b> .....	<b>38</b>
Hard Node .....	38
Soft Node.....	38
Databook di un nodo .....	38
Propagation delay.....	38
Alimentazione e ritardo di propagazione .....	39
Soglia e ritardo di propagazione.....	39
Interconnessione .....	40
Strenght.....	40
Leakage .....	41

Layout e cell area.....	41
Wafer Cross-Section.....	41
Symbol of the node .....	41
True Table .....	41
Capacità.....	41
Strength.....	41
Transition Time (Ritardo di propagazione) .....	42
Technology corners.....	42
Typical.....	42
Worst .....	42
Best.....	42
Rumore.....	43
Comportamento statico (Steady state parameters).....	43
Soglie .....	43
Margini di rumore .....	44
TRANSISTORS .....	44
CMOS Complementare .....	44
Pull-down Network (PDN) .....	44
Pull-up Network (PUN) .....	45
NAND2.....	45
NOR2 .....	46
Funzioni complesse (Teoria della dualità) .....	46
XOR .....	47
Complementary CMOS e il ritardo di propagazione.....	47
Analisi Dinamica (NAND) .....	49
Percorso critici di una rete combinatoria .....	52
CMOS gate – Body effect .....	54
Fan-in.....	56
Esempio di analisi dinamica (NAND) .....	57
Catena di inverter .....	58
TIMING .....	59
Latch.....	59
Flip Flop Edge Triggered (Registro) .....	59
Clock Skew.....	59
Gals (Global Asynchronous Locally Synchronous) .....	59
Classificazione dei tempi.....	60
Tsetup e Thold .....	60

Tc-q .....	60
Tplogic .....	60
<b>Timing violations .....</b>	<b>61</b>
Setup time violation .....	61
Hold time violation .....	62
Come evitarle? .....	62
<b>Timing e strumenti di sintesi.....</b>	<b>63</b>
Regola generale quando un circuito non funziona .....	64
<b>IL SEGNALE DI CLOCK.....</b>	<b>65</b>
<b>Clock skew .....</b>	<b>65</b>
<b>Clock Jitter .....</b>	<b>66</b>
<b>Clock skew e clock jitter combinati .....</b>	<b>67</b>
<b>Generatore di Clock .....</b>	<b>67</b>
Nota sulla Zibo board .....	67
PPL (Anello di Blocco di Fase) .....	67
Distribuire il segnale di clock .....	68

## Introduzione

Questi appunti sono nati da una grande disperazione, acidi e voglia di picchiare. L'uso di intelligenza artificiale è stato massivo ma mantenendo sempre una certa eleganza.

Veramente, col cuore, vi auguro uno studio decente.

I primi capitoli non servono ad una mazza...

## Recap di elettronica digitale

### Cross-section di un NMOS

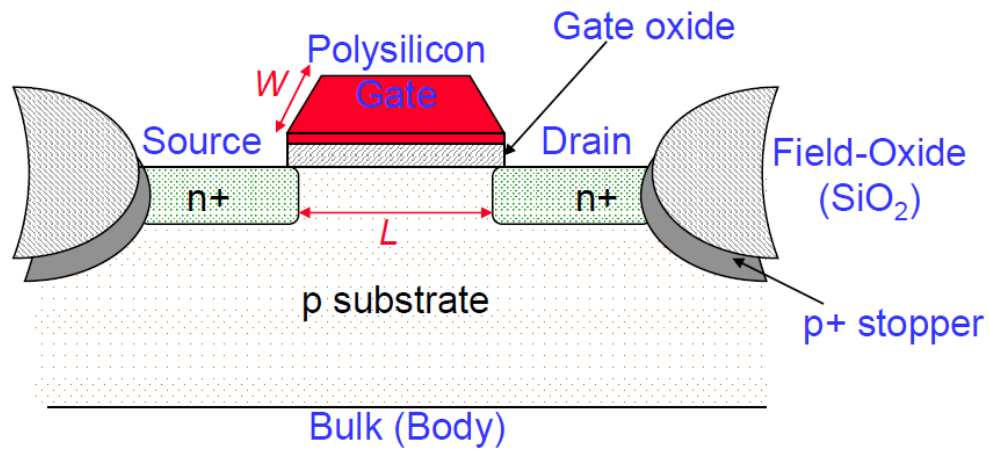


Figure 1- Struttura di un transistor.

### Approssimazione di un NMOS ad uno switch

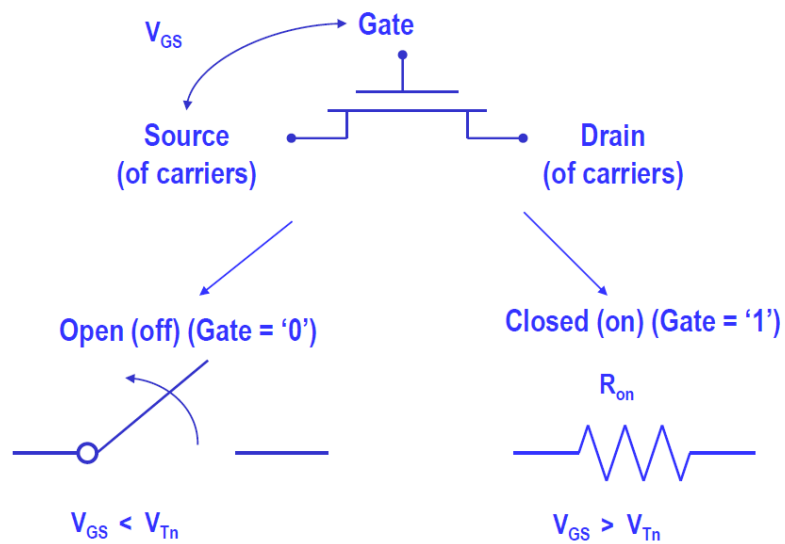


Figure 2- Approssimazione NMOS



Approssimazione di un PMOS ad uno switch

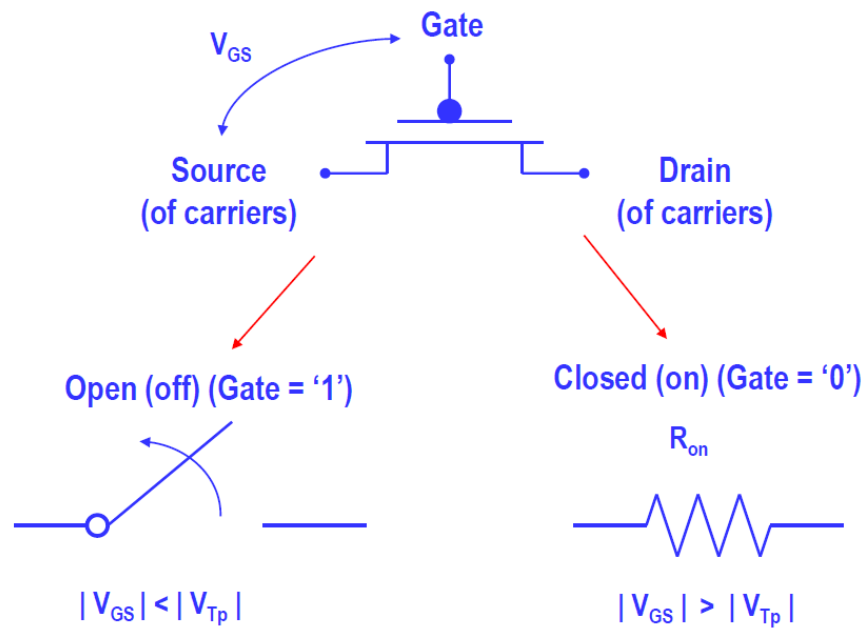
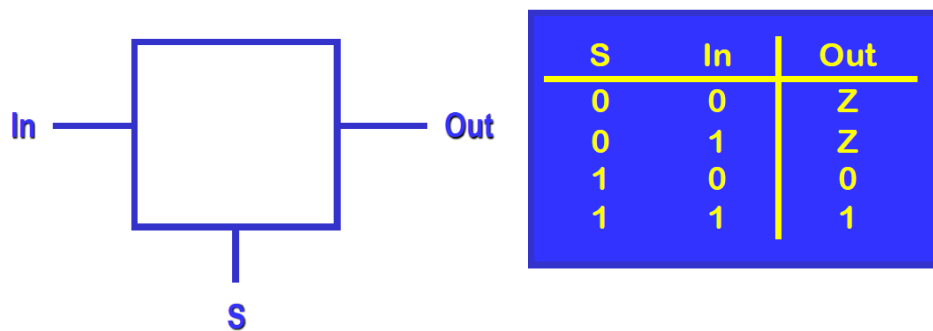


Figure 3 - Approssimazione PMOS

Pass gate

Three-state logic gates: high (H), low (L) and high-impedance (Z)



La resistenza equivalente di questo dispositivo è la seguente:

$$I_D = \beta_n \left[ (V_{GS} - V_T) V_{DS} - \frac{V_{DS}^2}{2} \right] \approx \beta_n (V_{GS} - V_T) V_{DS}$$

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

$$R = \frac{1}{\beta_n (V_{GS} - V_T)} = \frac{1}{\beta_n (4 - V_u)}$$

$V_G = 5V$   
 $V_T = 1V$

Figure 4 - Calcolo resistenza equivalente

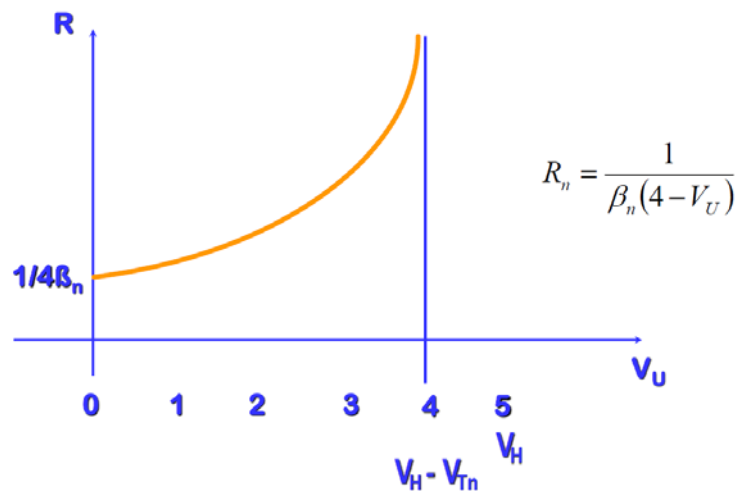
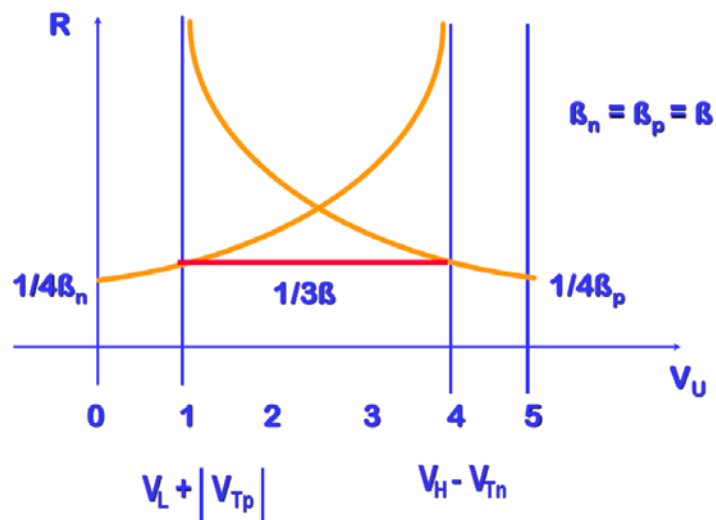
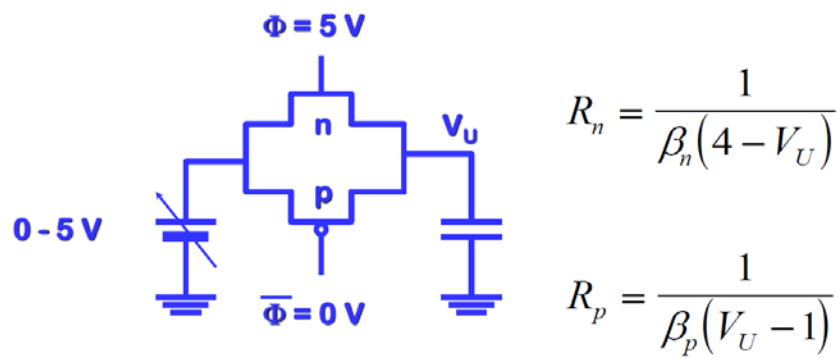


Figure 5 - Come varia la resistenza equivalente.

La resistenza equivalente varia in funzione della tensione di ingresso.



Inverter realizzato con Switch e CMOS

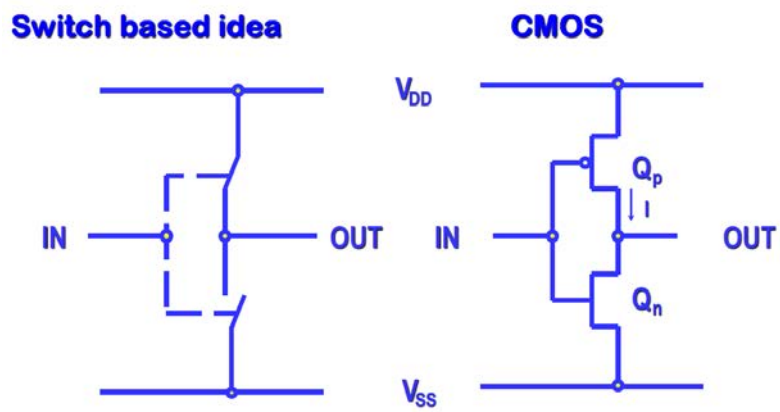


Figure 6 - Le due implementazioni degli inverter.

Il comportamento dell'uscita dipende (ovviamente) dalla tensione di ingresso.

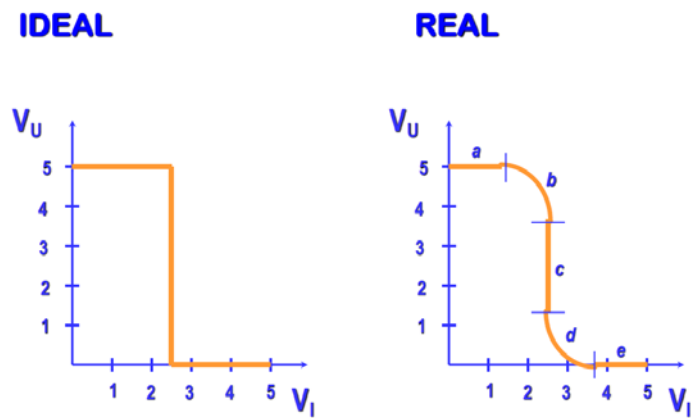


Figure 7- Caratteristica ideale e reale dell'inverter.

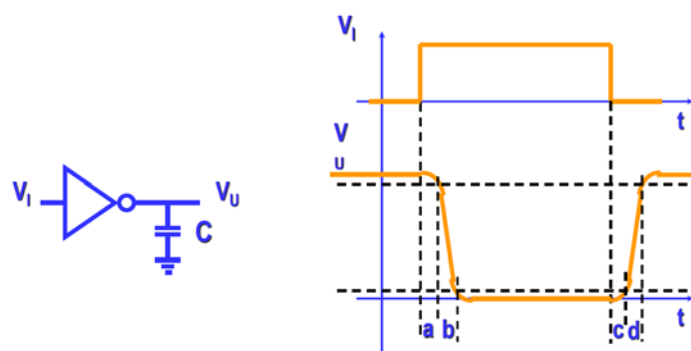
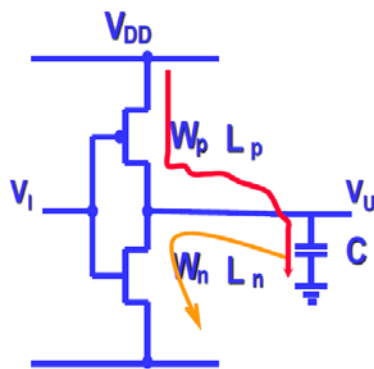


Figure 8- Caricare una capacità.

Ritardo di propagazione dell'inverter



$$t_{pHL} \propto \frac{KC}{\beta_n} \frac{1}{V_{DD} - V_{Tn}}$$

$$t_{pLH} \propto \frac{KC}{\beta_p} \frac{1}{V_{DD} + V_{Tp}}$$

$$\beta_n = \mu_n C_{ox} \frac{W_n}{L_n} \quad \beta_p = \mu_p C_{ox} \frac{W_p}{L_p}$$

Figure 9 - Calcolo del ritardo di propagazione in funzione delle caratteristiche dell'inverter.

Consumo di Potenza

- ✧ **Charge:** Capacitor store energy  $E_C = \frac{1}{2} CV^2$
- ✧ **Power Supply** provides energy  $E_f = CV^2$
- ✧ **Discharge:** Capacitor gives its energy to the inverter
- ✧ In one cycle the inverter consumes an energy equal to  $E_D = CV^2$
- ✧ In the time T the inverter consumes a power  
consumption of  $P_D = E_D/T = P_D = CV^2/T = CV^2f$

## Processo di fabbricazione dei CMOS

La fabbricazione dei transistor CMOS parte dai “wafer” di silicio.

### Qualità del processo costruttivo

La qualità del processo costruttivo di un circuito (*e quindi la qualità dei circuiti stessi*) dipende da questi fattori:

- **Tolleranza:** verso i requisiti di costruzione, più siamo intransigenti e meglio verranno (*però più costoso sarà il processo*).
- **Consistenza dei chip:** La differenza tra due chip prodotti deve essere minima.
- **Rate di produzioni fallite:** La percentuale di chip malfunzionanti prodotti, più sono e più costoso sarà il processo.
- **Durabilità del chip:** La scelta dei materiali gioca un ruolo fondamentale, un chip di alta qualità non si rompe dopo pochi secondi di funzionamento.
- **Conformità agli standard di produzione.**

### Litografia

Processo simile alla stampante 3D. Ad ogni passo, vengono depositati materiali diversi. Strato dopo strato il circuito viene alla luce.

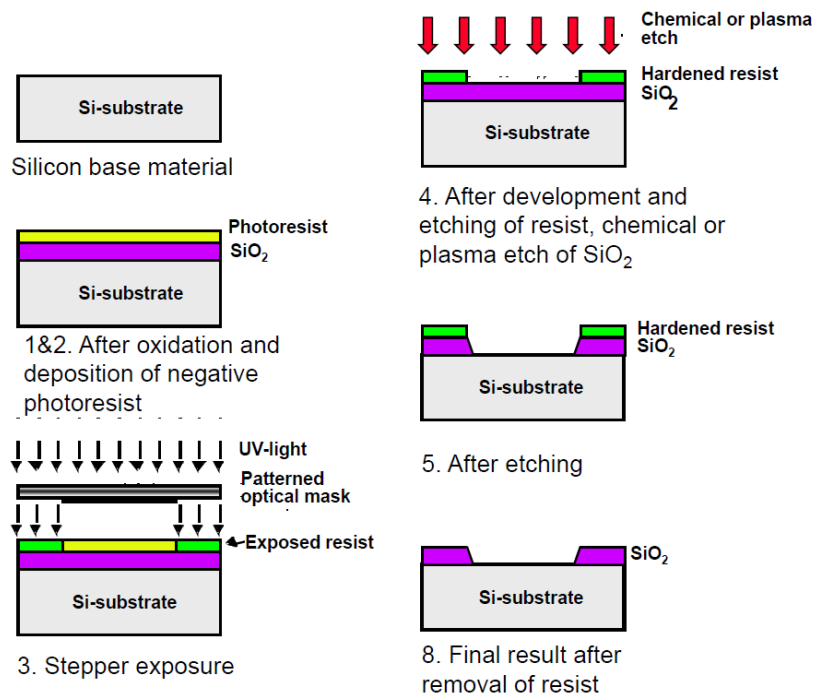


Figure 10 - Litografia.

Nello step tre la luce UV serve per esporre specifiche aree ad una certa reazione chimica che le rende più facili da rimuovere.

## Photolithography

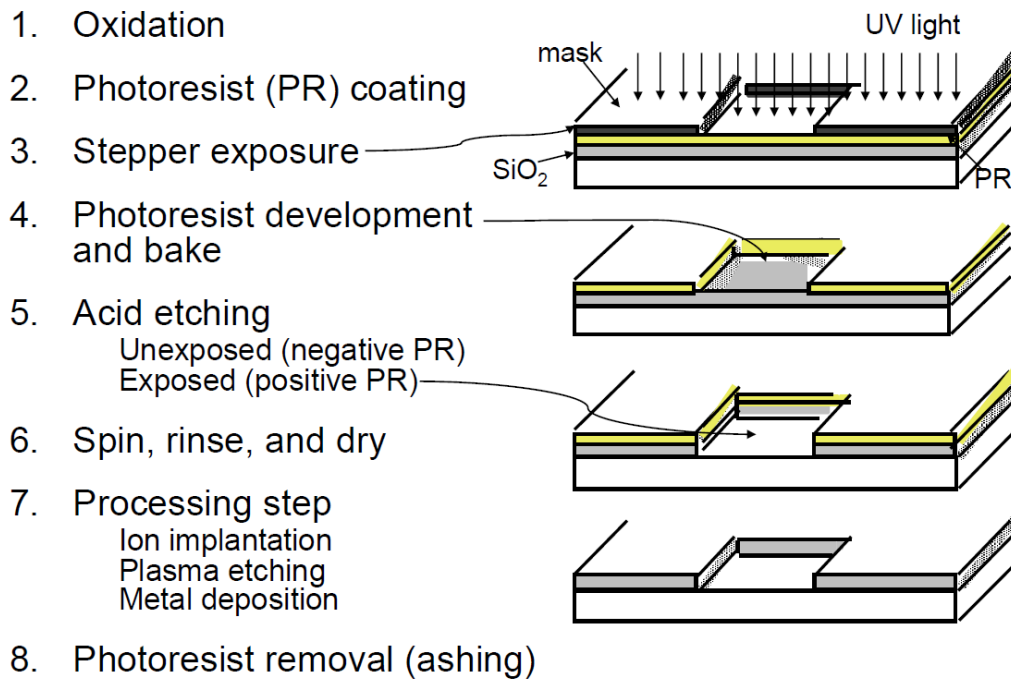


Figure 11- Processo della Photolithography.

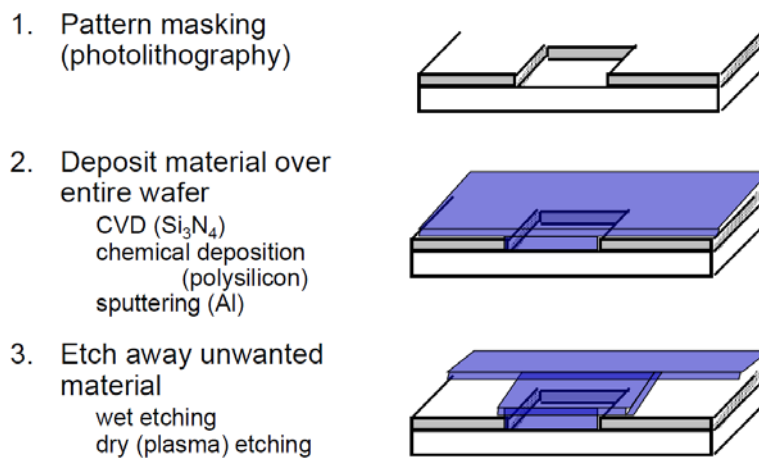


Figure 12 - In questa figura si nota come funziona il processo base: prima si deposita il materiale sopra la maschera e poi si incide.

## Cross-section e layout dell'inverter

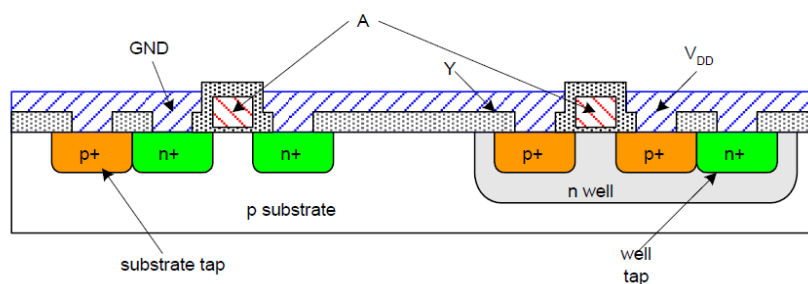


Figure 13- Cross-section dell'inverter.

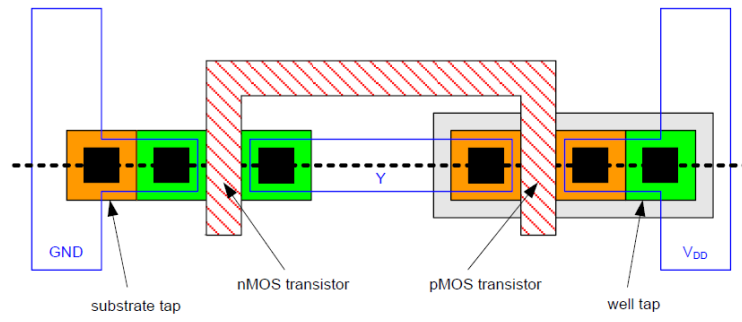


Figure 14 - Layout dell'inverter

## Fabbricazione di circuiti integrati

Vediamo ora come si crea un inverter passo dopo passo:

L'inverter verrà costruito dal basso verso l'alto.

### Wafer

Si inizia con un wafer di silicio (detto anche "blank wafer") esso è come una tela bianca nella quale incideremo il nostro dispositivo. Questo viene drogato in modo da farlo diventare carico di lacune (tipo P) esso sarà il nostro substrato.



Figure 15- Creazione del substrato di tipo P.

- First step will be to form the n-well:
  - Remove layer where n-well should be built
  - Implant or diffuse n dopants into exposed wafer
  - Strip off SiO<sub>2</sub>

### Ossidazione

Si copre il wafer con un ossido protettivo (SiO<sub>2</sub>).

Il processo è eseguito a 900 –1200 °C con acqua o ossigeno in una apposita fornace di ossidazione.



Figure 16 - L'ossido ricopre il substrato di tipo P.

### Photoresist

Sopra l'ossido viene messo un polimero organico sensibile alla luce chiamato "Photoresist".

Questo materiale si ammorbidisce quando è esposto alla luce.

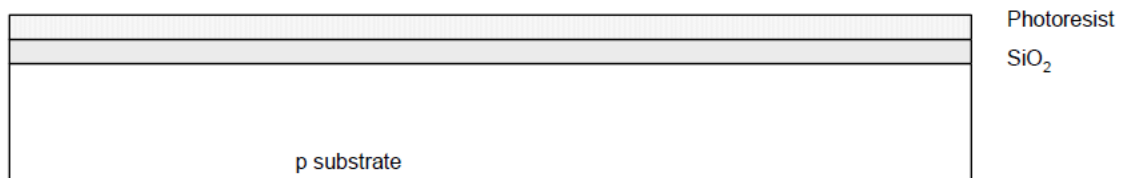


Figure 17- Il photoresist ricopre l'ossido.



## Litografia

Una parte del Photoresist viene incisa per fare spazio ad una maschera di silicio drogata (tipo N) chiamata "maschera".

La maschera serve solo a prendere le misure.

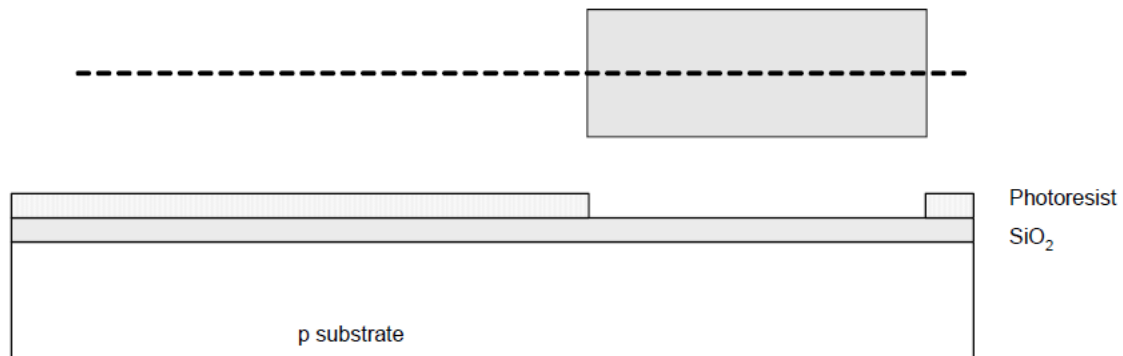


Figure 18 - La maschera viene adagiata sopra l'ossido dopo aver fatto spazio tra il photoresist.

## Incisione

Tramite acido fluoridrico si incide l'ossido. L'acido non attacca il photoresist; quindi, solo la parte esposta dell'ossido viene incisa.

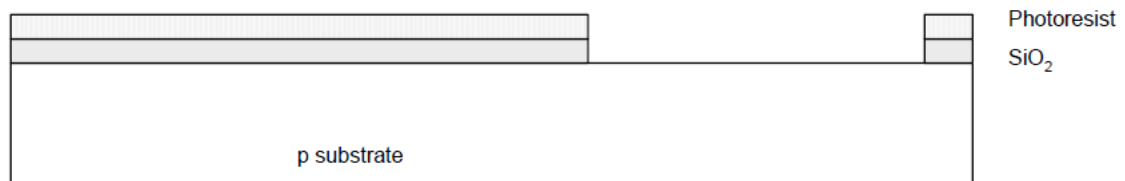


Figure 19- Incisione dall'ossido.

## Rimozione del photoresist

Usando una mix di acidi (chiamato piranha) si rimuove il photoresist.



Figure 20- Il piranha morde come un pazzo furioso il photoresist e lo ammazza male.

### Inserimento dello "n-well"

n-well è la parte di ossido drogata N e viene applicata tramite diffusione o "Ion implantation".

#### Diffusione

Si piazza il wafer tramite gas arsenico, il quale carica la parte esposta di elettroni (drogaggio di tipo N).

#### Ion Implantation

Si spara contro al wafer un raggio di ioni, i quali rimangono entrano e rimangono bloccati nella parte di silicio esposta.

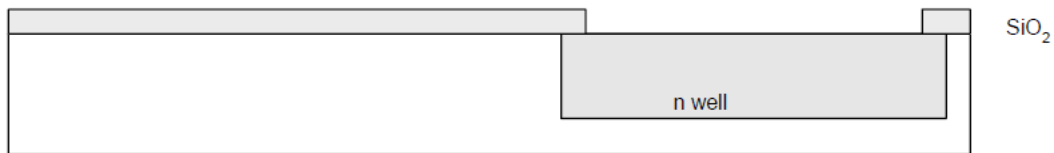


Figure 21- Risultato finale della diffusione o Ion Implantation

### Rimozione dell'ossido

Usando dell'acido fluoridrico si rimuove il resto dell'ossido.



Figure 22 - Dopo aver rimosso l'ossido.

### Inserimento polisilicio

Si deposita uno strato sottilissimo di poli silicio, il quale fungerà da gate del transistor.

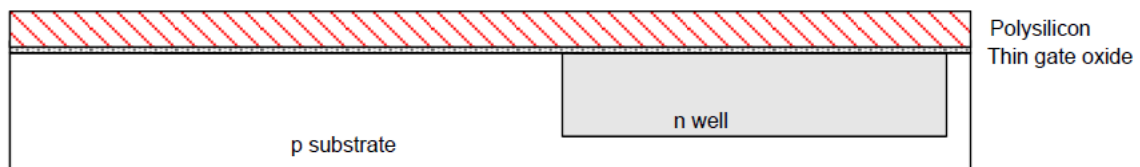


Figure 23- Dopo aver messo i polisilicio.

### Litografia del polisilicio

Tramite il processo di Litografia, si ottengono 2 pin, uno attaccato alla parte con drogaggio P e l'altra con drogaggio N.

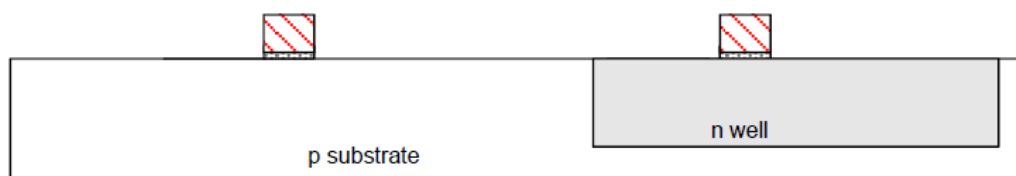


Figure 24 - Litografia del polisilicio.

### Self-Aligned Process

Si ricopre il tutto con dell'ossido.

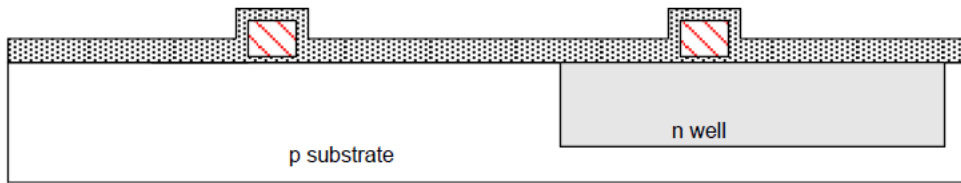


Figure 25 – L'Ossido ricopre la superficie.

### N-diffusion

Si incidono sull'ossiduo 3 aree nelle quali andrò (come prima) a creare zone di drogaggio di tipo N+.

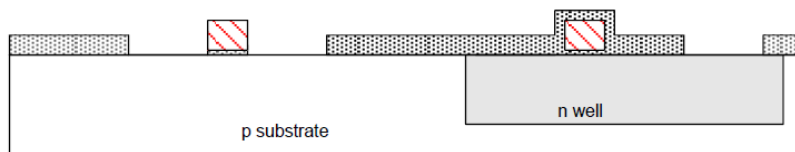


Figure 26 - Dopo aver inciso l'ossido n+.

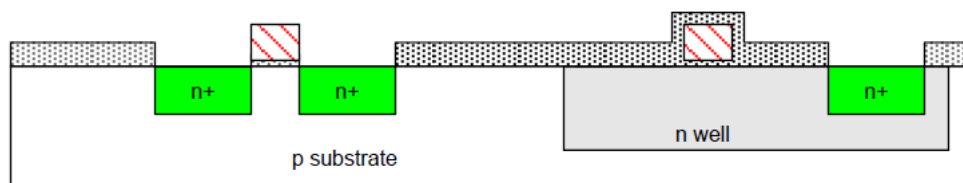


Figure 27- Dopo il processo di diffusione.

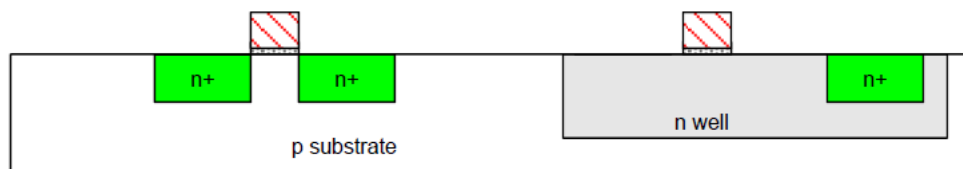


Figure 28 - Dopo aver rimosso l'ossido.

### P-Diffusion

Identico alla N-diffusion, ma ora si droga con un drogaggio di tipo P.

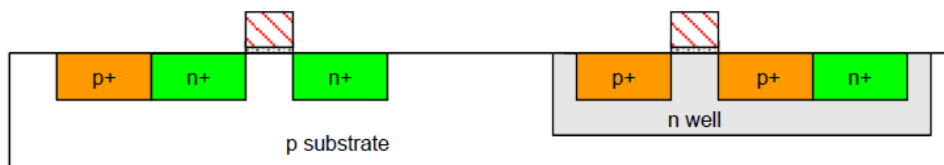


Figure 29- Dopo la P-diffusion.

## Contatti

Si ricopre tutto con uno spesso strato di ossido, poi si incide per fare spazio ai contatti.

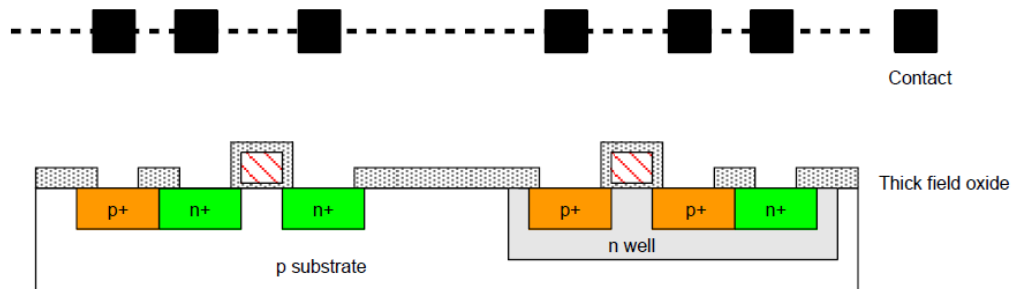


Figure 30 - Dove piazzerò i contatti.

## Metallizzazione

Ricopro di alluminio l'intero wafer.

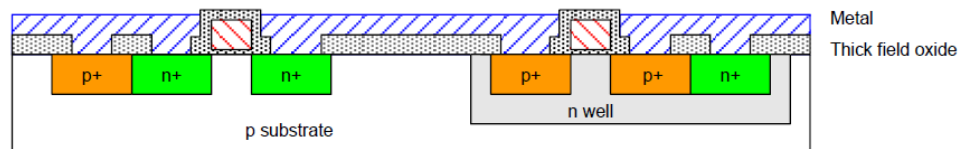


Figure 31- Metallizzazione.

## Regole di design

Interface between the circuit designer and process engineer;

Guidelines for constructing process masks;

Unit dimension: minimum line width:

scalable design rules: lambda parameter

absolute dimensions: micron rules

Rules constructed to ensure that design works even when small fab errors (within some tolerance) occur;

A complete set includes:

set of layers

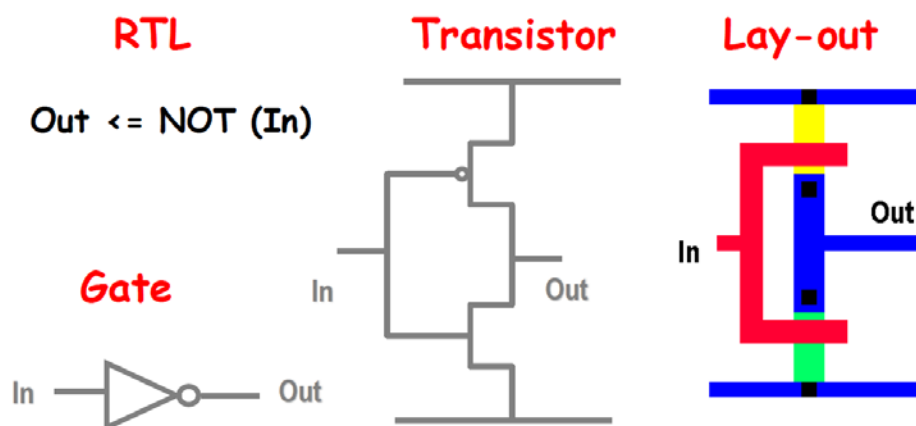
intra-layer: relations between objects in the same layer

inter-layer: relations between objects on different layers

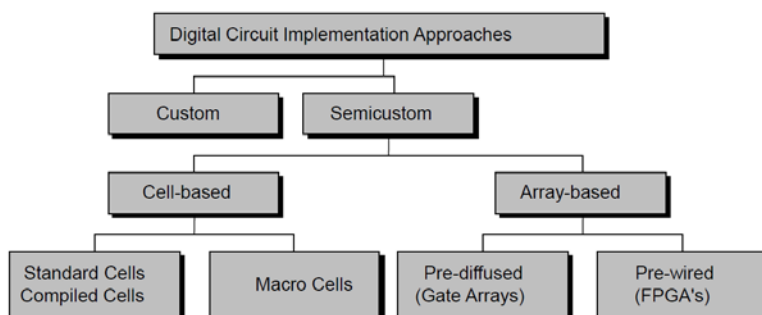
## Why Have Design Rules?

- To be able to tolerate some level of fabrication errors such as:
  - Mask misalignment
  - Dust
  - Process parameters (e.g., lateral diffusion)
  - Rough surfaces

## Electronic abstraction levels



## Integrated circuit design style



## Full-Custom Design

Prevede la creazione di un layout personalizzato per ogni componente del circuito, compresi transistor, resistori, condensatori e interconnessioni. In altre parole, ogni singolo elemento del circuito è progettato e posizionato manualmente per ottimizzare le prestazioni, la potenza e l'area del chip.

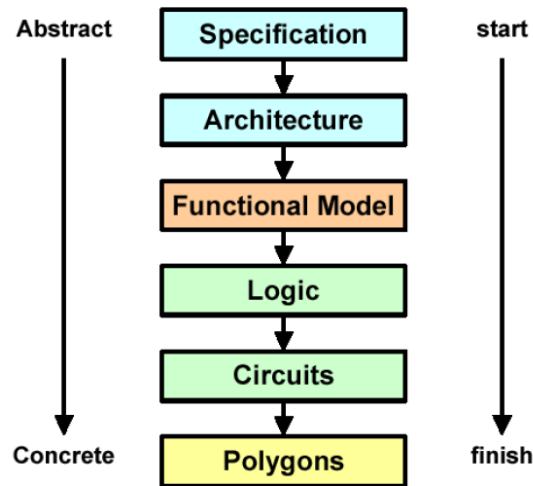


Figure 32- Workflow della full-custom design.

### Vantaggi

Il design full-custom offre la massima flessibilità durante la progettazione. Inoltre, il circuito prodotto avrà la massima ottimizzazione in termini di area, potenza e prestazioni.

### Svantaggi

Richiede un notevole sforzo di progettazione e un costo più elevato. È spesso utilizzato per applicazioni ad alte prestazioni (e ad alti volumi di vendita), dove le prestazioni, l'area del chip e la potenza sono critici.

### Specifiche del progetto

La prima fase del processo full-custom coinvolge la definizione delle specifiche del progetto, inclusi i requisiti di prestazioni, funzionalità, budget, potenza e spazio.

### Progettazione del circuito

Si selezionano i componenti del circuito e si produce il modello funzionale (detto anche "schematico") che riassume la funzionalità del circuito.

### Layout del circuito

I progettisti traducono il progetto schematico in un layout fisico dettagliato, che definisce la posizione e le dimensioni esatte di ogni componente del circuito, nonché le interconnessioni tra di essi. Questa fase coinvolge l'utilizzo di strumenti di layout CAD per posizionare e collegare i componenti in modo ottimale.

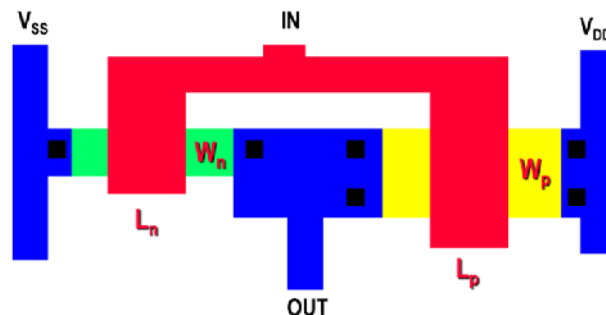


Figure 33- Layout di un inverter.

## Verifica e simulazione

Verifiche **per garantire che il design soddisfi le specifiche del progetto e funzioni correttamente** in tutte le condizioni di funzionamento previste. Questa fase viene portata avanti tramite tool che producono grafici come quello mostrato sotto.

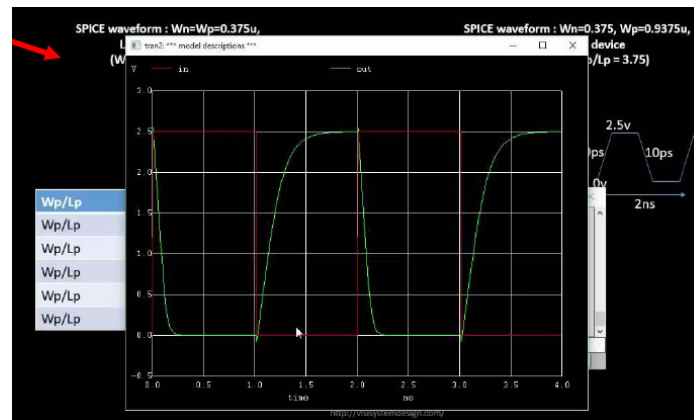


Figure 34 -Esempio di waveform.

## Design Rule Check [DRC]

La DRC è un controllo automatico eseguito da software per verificare che il layout (o altri aspetti) del circuito soddisfi le regole di progettazione.

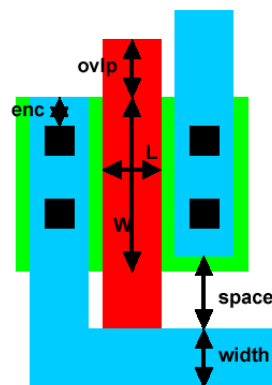


Figure 35 - Regole di progettazione che riguardano l'area del layout.

Le regole di progettazione possono riguardare parametri come le dimensioni delle strutture del circuito, le distanze minime tra le linee, le aree di contatto, le proporzioni dei dispositivi elettronici, le interconnessioni e molti altri aspetti del layout del circuito.

## Software di controllo della DRC - CAD

La DRC viene solitamente eseguita in modo automatico tramite software di progettazione assistita da computer (CAD) specificamente progettati per il design dei circuiti integrati.

### Layout vs Schematic [LVS]

La LVS è una fase nella quale, una volta completato il layout, si confronta il layout fisico con lo schematico per assicurarsi che corrispondano esattamente.

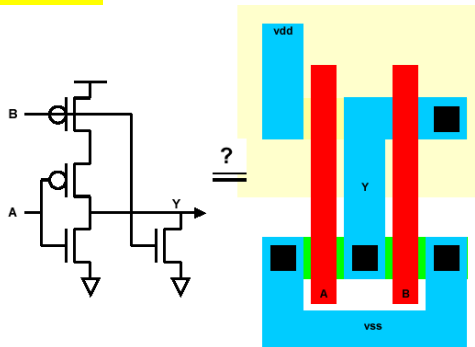


Figure 36 - Confronto tra lo schematico (sinistra) e il layout (destra).

La LVS garantisce che il layout sia corretto e funzionante.

### Layout Parasitic Extraction [LPE]

Fase che si concentra sull'estrazione e sulla modellazione dei parametri parassitici presenti nel layout.

Questi parametri parassitici possono includere capacità, induttanze, resistenze e altre caratteristiche che influenzano le prestazioni e il comportamento del circuito. Consente ai progettisti di valutare e ottimizzare le prestazioni del circuito tenendo conto di tali effetti.

Questa fase serve principalmente per compilare il datasheet del componente.



## Semi-Custom Design

Approccio intermedio tra la progettazione full custom e la progettazione standard cell (*cellule standard*), ed è ampiamente utilizzata nell'industria dei circuiti integrati per ottimizzare i tempi di sviluppo e i costi di progettazione. Le aziende produttrici forniscono blocchi "preconfezionati" e generici che noi utilizziamo per creare circuiti più grossi.

### Gate Array Structure

Questo approccio combina elementi di personalizzazione limitata con una vasta disponibilità di celle logiche predefinite, consentendo una progettazione più rapida ed economica rispetto alla progettazione full custom. La Gate Array Structure si basa sulle matrici di porte logiche:

#### Matrici di porte logiche

Matrice regolare di celle logiche ripetute su tutta l'area del chip. Queste celle logiche possono includere porte logiche, flip-flop, latch e altri elementi di base utilizzati nella costruzione di circuiti digitali.

Le connessioni tra le celle logiche sono predefinite e fissate durante la fase di progettazione della struttura a gate array.

La fabbrica confeziona questo insieme di porte logiche e che poi i progettisti collegano tramite canali.

#### Personalizzazione limitata

Nonostante la presenza di una vasta matrice di celle logiche, la personalizzazione all'interno di una struttura a gate array è limitata. I progettisti possono personalizzare solo una piccola parte del chip, nota come "zona di personalizzazione" o "zona di interconnessione". In questa zona, i progettisti possono creare circuiti più complessi, utilizzando interconnessioni personalizzate tra le celle logiche.

#### Vantaggi della Gate Array Structure

- **Costi ridotti:** La disponibilità di una vasta gamma di celle logiche predefinite riduce i costi di progettazione e produzione, in quanto non è necessario progettare ogni componente da zero.
- **Velocità di sviluppo:** Grazie alla struttura regolare e alle celle logiche predefinite, la progettazione di un chip basato su gate array è più rapida rispetto alla progettazione full custom.
- **Flessibilità limitata:** Sebbene la personalizzazione sia limitata rispetto alla progettazione full custom, la presenza di una zona di interconnessione consente comunque una certa flessibilità nella progettazione di circuiti specifici.

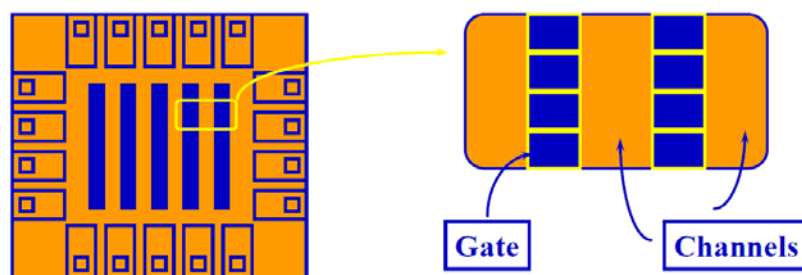


Figure 37 - Esempio di Gate Array- le parti blu sono le interconnessioni.

## Standard Cell Structure

Fornisce una selezione più limitata di celle logiche predefinite, ma offre maggiore flessibilità e prestazioni rispetto alla gate array.

### Librerie di celle standard

La caratteristica principale di una struttura a cella standard è l'uso di librerie di celle standard predefinite. Queste librerie contengono una vasta selezione di blocchi funzionali, come porte logiche, flip-flop, latch e altre componenti di base utilizzate nella costruzione di circuiti digitali. Le celle in queste librerie sono progettate per essere facilmente configurabili e riutilizzabili in diverse applicazioni di progettazione.

### Personalizzazione delle interconnessioni

A differenza della gate array, in cui le interconnessioni sono predefinite e fisse, nella struttura a cella standard, si ha maggiore flessibilità nella personalizzazione delle interconnessioni tra le celle. Ciò consente la creazione di circuiti più complessi e ottimizzati per le prestazioni, poiché le interconnessioni possono essere progettate in base alle specifiche esigenze del circuito.

### Vantaggi della Standard Cell Structure

- **Flessibilità:** I progettisti possono selezionare e configurare le celle standard in modo da soddisfare le esigenze specifiche del circuito.
- **Prestazioni ottimizzate:** Poiché i progettisti hanno maggiore controllo sulle interconnessioni tra le celle, è possibile ottimizzare le prestazioni del circuito, riducendo la propagazione del segnale, minimizzando i ritardi e migliorando l'efficienza complessiva.

### Svantaggi della Standard Cell Structure

- **Velocità di sviluppo:** Richiede più tempo rispetto alla gate array.

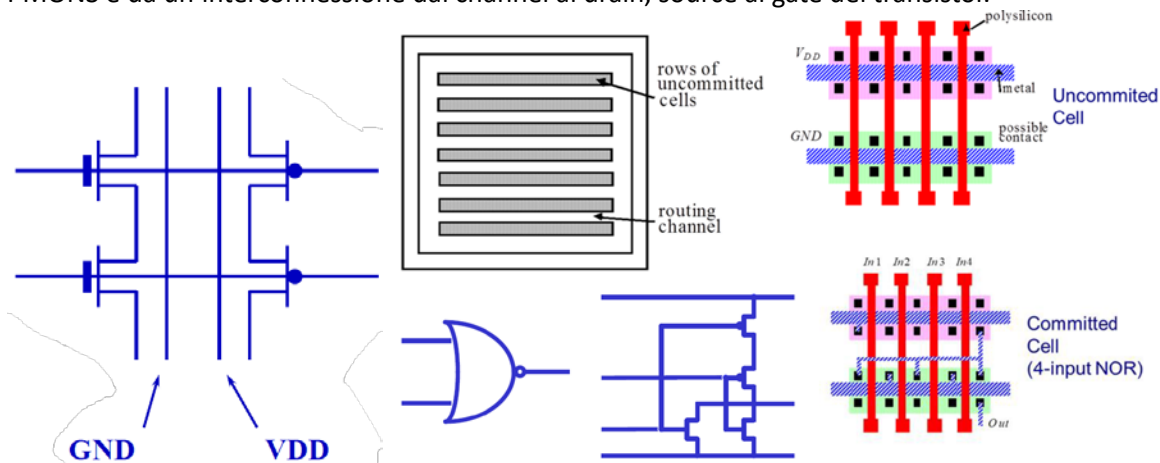
## ASIC (Application-Specific Integrated Circuit)

**Circuito creato appositamente per risolvere un problema:** consente di raggiungere delle prestazioni in termini di velocità di processamento e consumo elettrico difficilmente ottenibili con l'uso di soluzioni più generiche (*general purpose*).

**Sviluppo molto costoso e per questo motivo il loro impiego è limitato a campi in cui possano essere usati in maniera massiccia, ovvero alti volumi di mercato, come l'elettronica di consumo.**

### Single gate Structure

Tipo specifico di circuito integrato programmabile progettato per implementare una singola funzione logica o un singolo gate logico. Questo tipo di struttura è spesso utilizzato in applicazioni in cui è necessaria una piccola quantità di logica programmabile o una funzione logica specifica. È composta da 2 NMOS a da 2 PMONS e da un'interconnessione dal channel al drain, source al gate del transistor.



### Gate array interconnection

Connessione di una matrice di porte logiche programmabili (gate array) all'interno di un circuito integrato applicativo specifico (ASIC). In una struttura di gate array ASIC, una vasta matrice di porte logiche (come porte AND, OR, NOT, ecc.) è predisposta su un chip in un modello regolare.

### Matrice di porte Logiche programmabili

La base di una ASIC gate array è una matrice di porte logiche. Queste porte sono generalmente organizzate in una griglia regolare e possono essere configurate per eseguire varie funzioni logiche.

### Interconnessione programmabile

Le porte logiche all'interno della matrice sono interconnesse attraverso una rete di collegamenti programmabili. Questi collegamenti possono essere configurati in modo da collegare le porte logiche in modalità specifiche per implementare la funzionalità desiderata.

Gli ingegneri possono configurare i collegamenti tra le porte logiche in modo da soddisfare le specifiche dell'applicazione. Questo offre un alto grado di flessibilità nella progettazione di circuiti personalizzati senza dover progettare e fabbricare un circuito integrato da zero.

### Efficienza e Flessibilità

L'utilizzo di una matrice di porte logiche programmabili offre un compromesso tra l'efficienza di una soluzione ASIC specificamente progettata e la flessibilità di una FPGA (*Field-Programmable Gate Array*). È possibile ottenere prestazioni ottimizzate per l'applicazione specifica senza sacrificare la capacità di personalizzazione.

In conclusione, la "ASIC Gate Array Interconnection" si riferisce alla configurazione programmabile delle interconnessioni tra le porte logiche all'interno di una matrice di porte logiche su un chip ASIC.

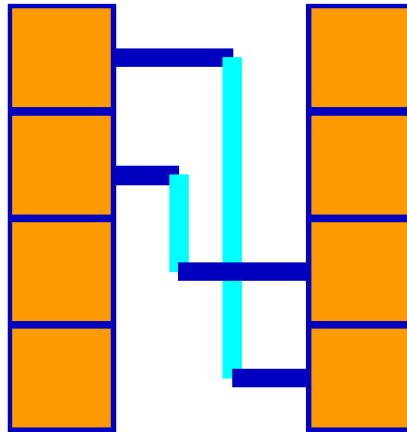


Figure 38- Interconnessioni tra porte logiche di due matrici.

### Sea-of-gate structure

Si basa su una matrice regolare di celle di base contenenti porte logiche. Prevede la disposizione di un gran numero di celle di base all'interno di una griglia su un chip.

#### Matrice Regolare

La caratteristica principale di una struttura Sea-of-Gates è la sua disposizione regolare. Le celle di base, ognuna contenente una o più porte logiche, sono organizzate in una griglia uniforme su tutto il chip.

#### Interconnessione Programmabile

Le celle di base sono collegate tra loro attraverso una rete di interconnessione programmabile. Questo consente agli ingegneri di configurare le connessioni tra le porte logiche per implementare la logica desiderata per l'applicazione specifica e in modo da soddisfare le specifiche dell'applicazione, consentendo la creazione di circuiti integrati altamente personalizzati.

#### Efficienza

La disposizione regolare delle celle di base e la possibilità di condividere le risorse tra di esse può portare a una maggiore efficienza in termini di area di chip, consumo di energia e prestazioni rispetto ad altre soluzioni di progettazione ASIC.

#### Applicazioni

La struttura Sea-of-Gates è ampiamente utilizzata in una vasta gamma di applicazioni, inclusi dispositivi elettronici di consumo, apparecchiature di rete, sistemi embedded, dispositivi di elaborazione di segnali digitali, e altro ancora. È particolarmente adatta per applicazioni in cui è richiesta una combinazione di prestazioni ottimizzate e flessibilità di progettazione.

## Logic Synthesis

Processo compiuto da software specializzati, i quali richiedono:

- Descrizione VHDL del circuito.
- Librerie di tecnologie che voglio usare nel circuito.
- Requisiti di design e obiettivi, come ad esempio:
  - $\text{Area} < X \text{ pm}$
  - $\text{Power Consumption} < Y \text{ mW}$
  - $F_{\text{max}} (\text{frequenza massima di clock}) > Z \text{ MHz}$

Non è sicuro che il prodotto del software rispetterà tutti i vincoli, questo perché non c'è alcun controllo sull'effettiva fattibilità del circuito. Il tool potrebbe anche accettare vincoli impossibili (tipo  $\text{Area} = 0$ ).

### Register Transfer Level [RTL]

Livello di astrazione (*modello*) per descrivere il comportamento di un circuito digitale in termini di trasferimenti di dati tra i registri. In pratica, l'RTL definisce come i dati vengono trasferiti da un registro all'altro all'interno del circuito in risposta a determinati eventi di controllo.

Nel modello RTL, il circuito è descritto come una serie di registri e percorsi di trasferimento dei dati tra di essi. Ogni registro memorizza temporaneamente un valore e i trasferimenti di dati avvengono tra i registri in base a segnali di controllo.

RTL è un livello di descrizione più basso rispetto a livello di astrazione più elevati, come il behavioral level, ma più alto rispetto al gate level. Ciò significa che offre una visione più dettagliata del circuito rispetto a livelli più astratti, ma non specifica l'implementazione hardware a livello di singoli transistor.

RTL è comunemente espresso utilizzando linguaggi di descrizione hardware (HDL) come Verilog o VHDL. Questi linguaggi forniscono una sintassi e una semantica per descrivere il comportamento dei circuiti digitali a un livello di astrazione RTL.

### Place and Route

In questo step, il circuito sintetizzato (*quindi dopo la sintesi logica fatta dal software*) viene mappato fisicamente su un chip, dove vengono assegnate le posizioni fisiche ai componenti del circuito e vengono tracciate le interconnessioni tra di essi. Questo processo è fondamentale per garantire che il circuito funzioni correttamente e rispetti i requisiti di performance, consumo energetico e layout.

La fase di Place and Route è essenziale per garantire che il circuito integrato funzioni correttamente e soddisfi i requisiti di performance, consumo energetico e area. Un'efficace Place and Route può influenzare significativamente le prestazioni e la producibilità del chip finale.

### Costo di un circuito integrato

Una delle metriche da considerare ovviamente:  $\text{Costo Totale} = \text{NRE} + (N * \text{RE})$

#### Non-Recurrent engineering Cost [NRE]

Sono i costi di design e progettazione. Sono costi indipendenti dal numero di unità prodotte.

- Licenza dei tool usati.
- Stipendi.
- Costo dovuto allo stampo per produrre i circuiti.
- Tool di test per gli ASIC.

#### Recurrent Engineering Cost [RE]

Sono i costi di produzione. Quindi dipendenti dal numero e dalla grandezza delle unità prodotte.

- **Materie prime:** I wafer di silicio e altri metalli.
- **Processo di produzione:** Acqua, elettricità e manutenzione.
- **Package:** Il box che conterrà il chip.
- **Test**

- Ogni singolo circuito in uscita va testato con delle grosse e costose macchine, ovviamente le noleggiamo; quindi, le paghiamo per unità di tempo.
- Di solito per testare un circuito ci si mette qualche secondo.

Costo totale di produzione

Consideriamo che produrremo N unità:  $C_t = NRE + n \cdot RE$

*Costo per unità*

$$C_1 = (C_t / n) = (RE + NRE/n)$$

Da notare che  $C_1$  è sempre maggiore di  $RE$ .

*Come risparmiare*

*Miglior stile di design con N piccolo?* Gate-array, Poiché ha NRE bassi.

Performa bene con N basso, ma non posso ottimizzare l'area.

*Stile di design che richiede meno area?* Full-Custom, perchè ottimizzo tutto però conviene solo a fronte di N molto grande (*posso ammortizzare*), poiché NRE è altissimo.

## FPGA (Field Programmable Gate-Array unit)

Dispositivo di circuito integrato che può essere programmato e configurato dopo la sua produzione. Contrariamente ai circuiti integrati tradizionali, come ASIC (Application-Specific Integrated Circuit), che sono fabbricati per uno scopo specifico durante la produzione e non possono essere modificati, **gli FPGA possono essere programmabili per svolgere varie funzioni specifiche dopo essere stati fabbricati.**

Da cosa sono fatti?

Gli FPGA sono costituiti da una matrice di blocchi logici configurabili (*come porte logiche, latch e flip-flop*) e interconnessioni programmabili che permettono agli utenti di implementare qualsiasi funzione logica desiderata. Possono essere programmati utilizzando linguaggi di descrizione hardware come VHDL.

Vantaggi di FPGA

I vantaggi principali degli FPGA includono la flessibilità e la riconfigurabilità, che li rendono adatti a una vasta gamma di applicazioni, dalla prototipazione rapida alla produzione di dispositivi su larga scala.

Programmare un FPGA

FPGA SRAM Base (Xilinx)

Lo stile di sviluppo SRAM-based (Static Random-Access Memory) per FPGA si riferisce a un tipo di tecnologia utilizzata per la realizzazione di FPGA, in cui **la configurazione del dispositivo è memorizzata all'interno di celle di memoria statica volatile (SRAM).**

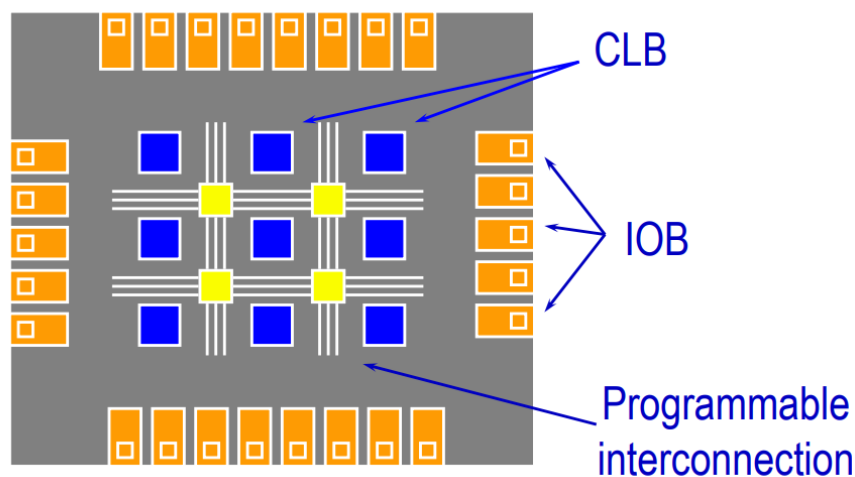


Figure 39- FPGA Block.

*CLB (Configurable Logic Block)*

Blocco hardware dove sono implementate delle funzioni logiche. Ogni CLB è una rete sequenziale sincronizzata con clock, enable ed elementi di memoria. Ogni CLB ha anche una PROM (*la quale permette di implementare qualsiasi funzione logica*).

*IOB (Input-Output-Circuits)*

Interfacce I/O programmabili.

*Configurazione temporanea (Programmable interconnection)*

La configurazione del dispositivo viene caricata da un dispositivo di memoria esterna (*come un chip di memoria SRAM*) ogni volta che il dispositivo viene avviato o reset, consentendo una maggiore flessibilità e programmabilità rispetto ad altre tecnologie di FPGA. La configurazione del dispositivo FPGA viene memorizzata temporaneamente in una memoria SRAM esterna o all'interno del dispositivo FPGA stesso.

### General interconnection

Vengono utilizzati gli switch (chiamati anche *exchange matrix*) del board.

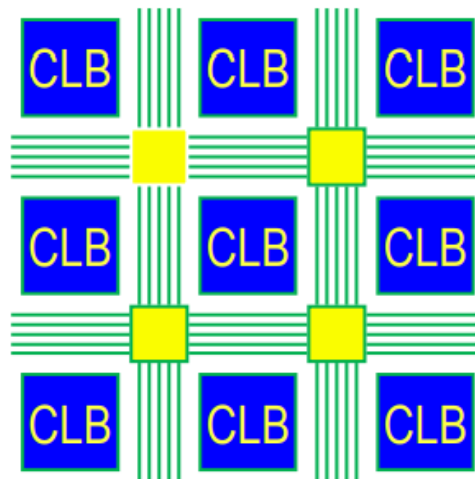


Figure 40- Situazione iniziale, nei blocchi gialli (*exchange matrix*) verranno implementate le interconnessioni.

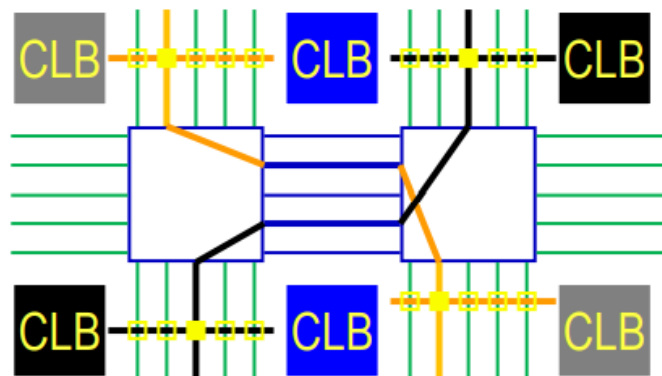


Figure 41- Situazione finale, le interconnessioni sono state programmate.

### Direct interconnection

Connessioni dedicate tra CLB adiacenti. Permette di avere ritardo di propagazione minimo e spesso anche calcolabile a priori. Non usa risorse generali.

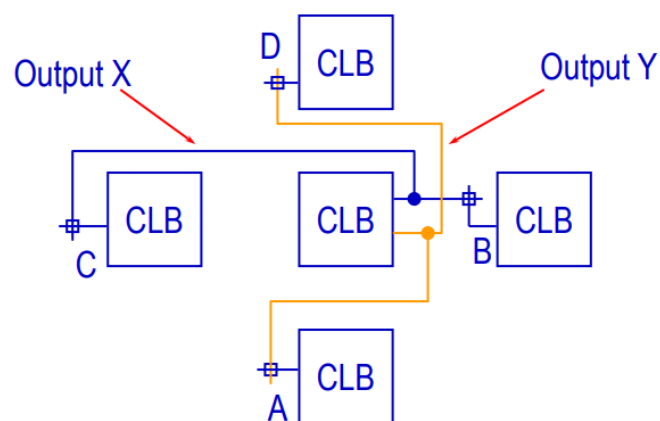


Figure 42 - Connessione diretta.



### Global interconnection

Vengono definite lunghe linee che attraversano tutto il circuito senza usare delle exchange Matrix. Ho il minimo ritardo di propagazione del clock (quindi è il modo migliore per rendere i CLB sincronizzati).

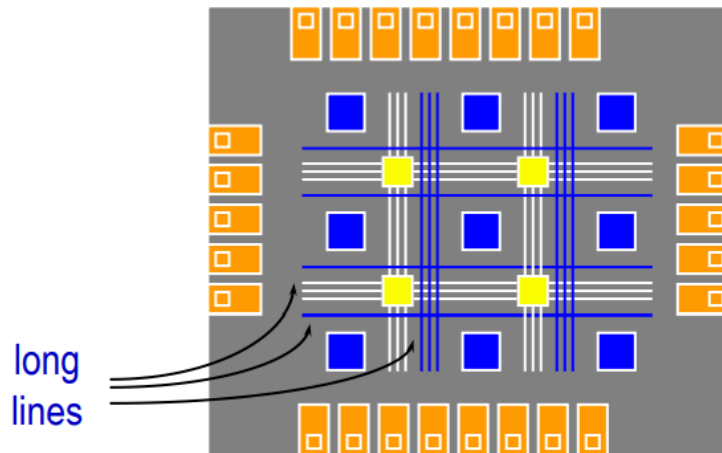


Figure 43- Global interconnection.

### Flessibilità

Poiché la configurazione è memorizzata in una memoria esterna, può essere modificata o riscritta più volte senza danneggiare il dispositivo FPGA stesso. Questo consente una maggiore flessibilità nel processo di sviluppo e iterazione del progetto.

Inoltre, alcuni FPGA possono essere riconfigurati dinamicamente durante l'esecuzione, consentendo di adattare il dispositivo alle esigenze specifiche dell'applicazione senza dover interrompere il funzionamento del sistema.

### Facilità di sviluppo

Questo stile di sviluppo semplifica il processo di progettazione e sviluppo, poiché consente ai progettisti di testare rapidamente diverse configurazioni e iterare sul design senza dover ricorrere a costosi processi di produzione di chip personalizzati.

### FPGA basato su tecnologie anti-fused

La configurazione del dispositivo è memorizzata in modo permanente all'interno di cellule di memoria che possono essere programmate attraverso la "fusione" o la "antifusione" di determinati collegamenti.

### Configurazione permanente

Una volta che le connessioni sono fuse, la configurazione del dispositivo FPGA diventa permanente e non può essere modificata. Questo differisce dagli FPGA SRAM-based, dove la configurazione può essere modificata dinamicamente.

### Robustezza e sicurezza

Offre un'elevata sicurezza e robustezza contro tentativi di hacking o manipolazioni non autorizzate, poiché non è possibile modificare la configurazione una volta che è stata impostata.

## Segnale di clock negli FPGA

### Generazione

Il segnale di clock viene tipicamente generato da un oscillatore esterno o da un circuito PLL (Phase-Locked Loop) all'interno del FPGA stesso. Questo segnale di clock ha una frequenza e una fase specifiche stabilite dall'utente o dal progettista del circuito.

### Distribuzione e divisione in aree

Il segnale di clock viene quindi inviato a un insieme di buffer all'interno dell'FPGA. Questi buffer servono a garantire che il segnale di clock sia adeguatamente distribuito in tutto il dispositivo (*in modo da evitare jitter o disturbi*).

Il board degli FPGA viene divisa in aree e il clock è gestito internamente in ogni area. Questo è uno dei motivi per cui non è possibile utilizzare il clock skew tra aree diverse in un FPGA.

## DSP (Digital Signal Processor)

Dispositivi progettati per elaborare segnali digitali in tempo reale. Questi chip sono ottimizzati per eseguire vari e complessi algoritmi di elaborazione dei segnali, come filtraggio, trasformazioni di Fourier, modulazione, demodulazione e altro ancora.

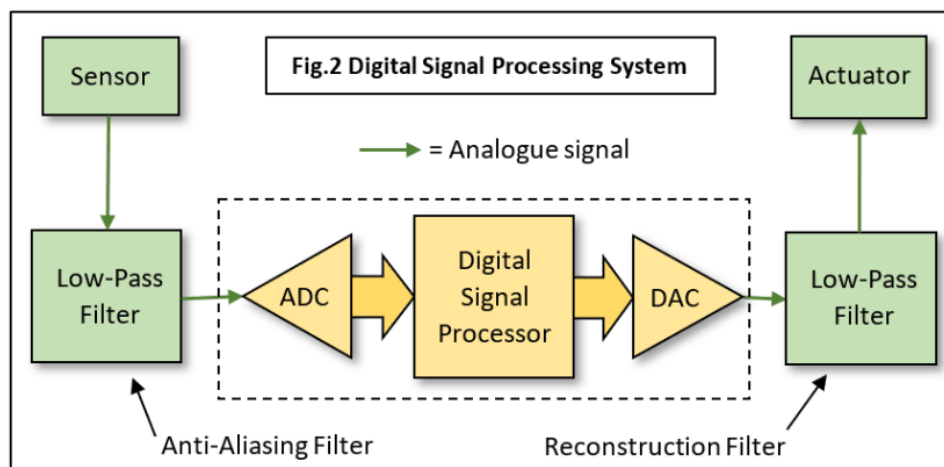


Figure 44 - Diagramma di un DSP.

E' l'hardware usato dalle SDR (Software Defined Radio) per processare digitalmente i segnali.

Un DSP ha bisogno di hardware piuttosto generico come ALU (*Arithmetic Logic Unit*), unità di moltiplicazione-accumulo (MAC), registri, banchi di memoria e altro ancora.

Sebbene i DSP siano progettati per eseguire specifici algoritmi di elaborazione dei segnali, molti offrono anche una certa flessibilità attraverso l'utilizzo di architetture programmabili o di istruzioni di set di istruzioni (ISA) che consentono di implementare una varietà di algoritmi.

In sintesi, i DSP appartengono alla branca "Software defined" insieme alle CPU general purpose.

## Riassunto degli stili di design dei circuiti integrati

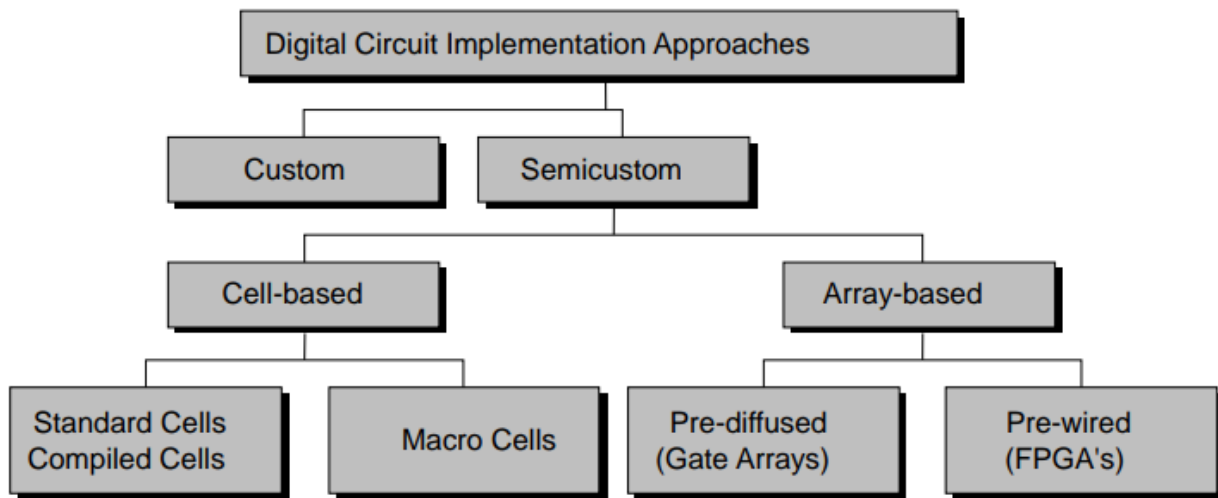


Figure 45- Stili di design divisi per famiglie.

### The “Energy-Flexibility” Gap

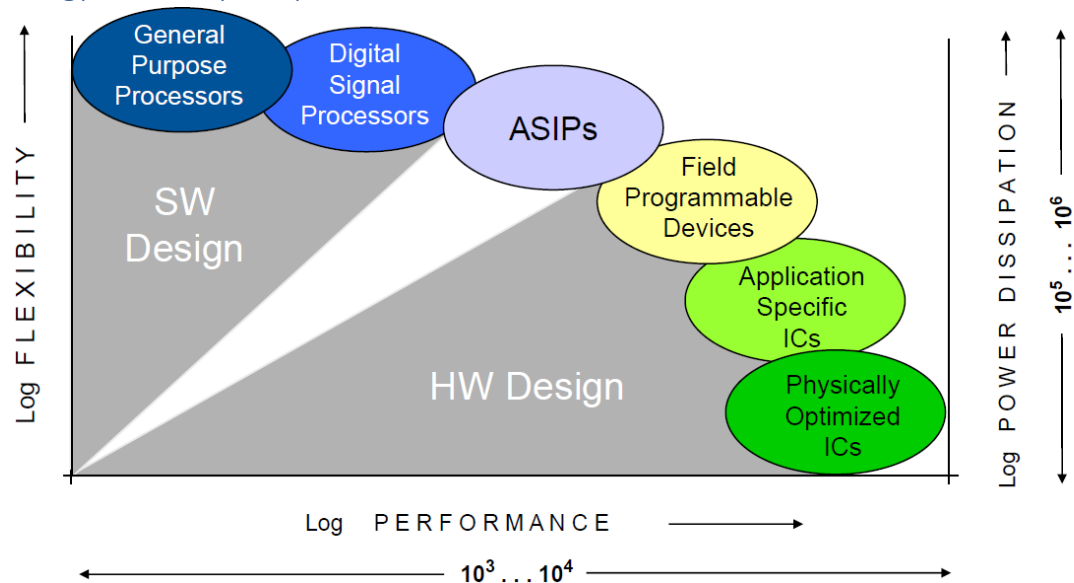


Figure 46- Trade-off di ogni possibile stile di design.

Come si può vedere, la full custom design offre la maggiore ottimizzazione di area e performance ma molta meno flessibilità. Invece FPGA (field-programmable design) presenta minore capacità di performance ma più flessibilità. Il Massimo della flessibilità la si ottiene da un processore general purpose, il quale ha la capacità di eseguire software e quindi potenzialmente fare qualsiasi cosa; però è la soluzione più lenta e ingombrante.

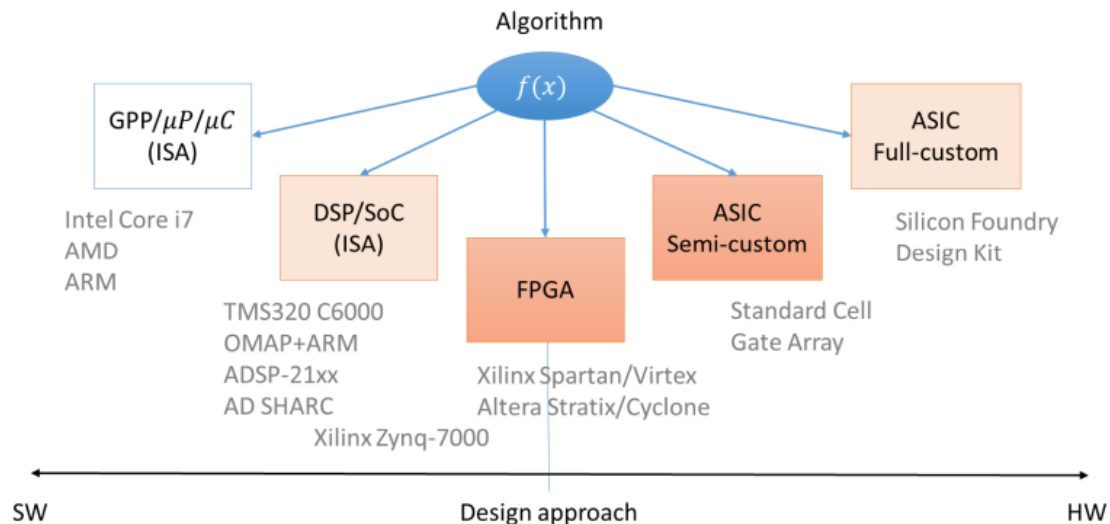


Figure 47- Filosofia di ogni stile di design.

## SoC (System-on-Chip)

Gli elementi SoC (System-on-Chip) sono circuiti integrati che integrano tutti o la maggior parte dei componenti di un sistema elettronico su un singolo chip. Questi componenti possono includere processori centrali (CPU), unità di elaborazione grafica (GPU), DSP (*Digital Signal Processor*), memoria, interfaccia di I/O, controller di periferiche, unità di elaborazione specializzate e altro ancora.

L'obiettivo degli SoC è quello di fornire una soluzione completa e altamente integrata per una vasta gamma di applicazioni, consentendo un design più compatto, efficiente ed economico dei dispositivi elettronici. Gli SoC sono ampiamente utilizzati in dispositivi quali smartphone, tablet, smartwatch, dispositivi IoT (Internet of Things), sistemi embedded.

Grazie alla loro elevata integrazione e alla capacità di ospitare una vasta gamma di funzionalità su un singolo chip, gli SoC consentono di ridurre i costi di produzione, migliorare le prestazioni e l'efficienza energetica, e semplificare lo sviluppo e la produzione di dispositivi elettronici complessi.

programmable	reconfigurable	fixed
DSP, $\mu$ C, RISC, GPP, ...	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	

## La foundry library

Libreria di elementi di base fornita dalla fabbrica di semiconduttori (*appunto la foundry*) che produce il chip FPGA. Questa libreria contiene una vasta gamma di componenti e blocchi di base che possono essere utilizzati dai progettisti durante lo sviluppo del circuito su un FPGA.

La library è il set di hardware di cui dispone il FPGA per emulare la descrizione VHDL.

### Porte logiche

Include porte logiche elementari come AND, OR, NOT, NAND, NOR, XOR, e XNOR. Queste porte logiche sono gli elementi di base utilizzati per costruire circuiti più complessi.

### Flip-flop e latch

Questi sono gli elementi di memoria utilizzati per memorizzare informazioni all'interno del circuito. Possono includere flip-flop di tipo D, JK, SR, e latch.

### Multiplexer e demultiplexer

Questi componenti consentono la selezione e il routing di segnali all'interno del circuito.

### Contatori e registri

Utilizzati per contare e memorizzare dati sequenziali all'interno del circuito.

### Memorie RAM e ROM

Blocchi di memoria utilizzati per memorizzare dati in modo temporaneo o permanente.

### Adattatori di interfaccia

Forniscono interfacciamento con altri componenti del sistema, come bus di comunicazione o interfacce di input/output.

### Convertitori di segnale

Includono convertitori di segnale analogico-digitale (ADC) e digitale-analogico (DAC) utilizzati per interfacciare il circuito FPGA con il mondo esterno.

La foundry library fornisce una base essenziale per lo sviluppo di circuiti su FPGA, poiché offre una serie di componenti predefiniti e ottimizzati per le specifiche del chip FPGA in questione. I progettisti possono utilizzare questi componenti come mattoni di base per costruire circuiti più complessi, risparmiando tempo e sforzo nello sviluppo di soluzioni su misura.

## Tipologie di Logica

### Logica combinatoria

L'output corrente dipende solo dall'input corrente.

### Logica sequenziale

L'output corrente dipende dall'input corrente e anche dagli input passati.

#### Asynchronous Sequential Logic

L'output corrente dipende dall'input corrente e anche dagli input passati, però l'output è trasparente, quindi se l'input cambia anche l'output cambia immediatamente.

#### Synchronous Sequential Logic

L'output corrente dipende dall'input corrente e anche dagli input passati, però l'output è non trasparente; quindi, se l'input cambia l'output non cambia immediatamente.

L'output cambia a seconda del valore del clock (*in comune a tutti i componenti di questo tipo*).

### Macchina di Mealy

Classico esempio di logica asincrona. L'elemento di memoria è un D-Flip-Flop positive Edge triggered.

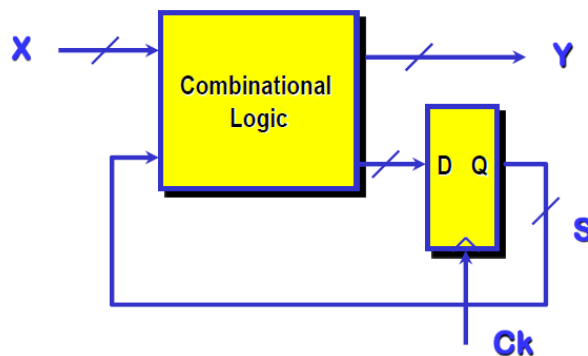


Figure 48 - Macchina di Mealy

Macchine di Mealy in cascata possono dare problemi a causa della loro trasparenza. Se l'input della prima balla, allora l'output della prima balla e così via.

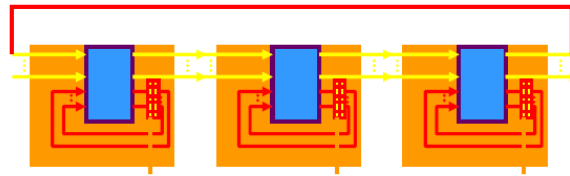


Figure 49- macchine di Mealy in cascata

### Macchina di Moore

Classico esempio di logica sincrona. L'elemento di memoria è un D-Flip-Flop positive Edge triggered.

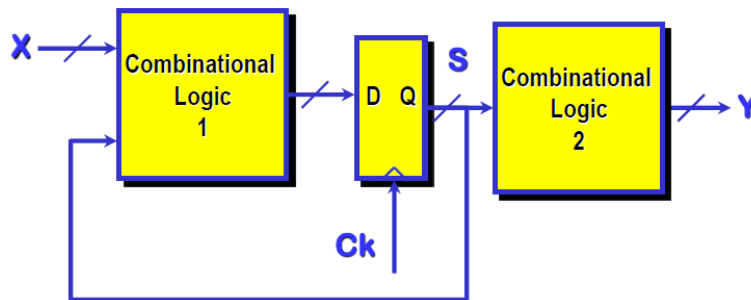


Figure 50 - Esempio di macchina di Moore.

Macchine di Moore in cascata non danno il problema delle macchine di Mealy.

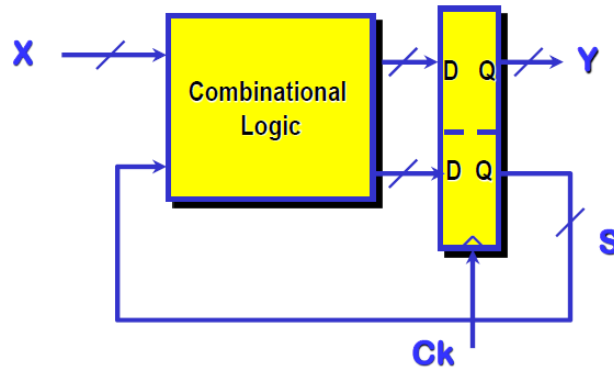


Figure 51 - Esempio di macchina di Moore.

## Nodi

### Hard Node

Se l'input non cambia, l'output non cambierà mai (es. Inverter).

### Soft Node

Se l'input non cambiasse, l'output potrebbe comunque cambiare (es. memoria dinamica).

Questo accade quando utilizzo una cella di memoria dinamica: In questo tipo di cella, memorizziamo il bit in una capacità, eventualmente, la capacità si svuoterà e perderò le informazioni.

Questa è la situazione in cui memorizziamo il valore su una capacità parassita, un esempio potrebbe essere quando carichiamo un nodo di output attraverso un transistor di passaggio, la carica in questo caso si dissiperà lentamente a causa della corrente parassita, anche se manteniamo l'alimentazione accesa, non è garantito che il valore venga mantenuto, possiamo risolvere la situazione aggiornando il valore nella capacità.

### Databook di un nodo

Informazioni date dal produttore. Come, ad esempio, lo schematico o la tabella di verità. Di solito contiene anche informazioni più particolari che andremo ad elencare.



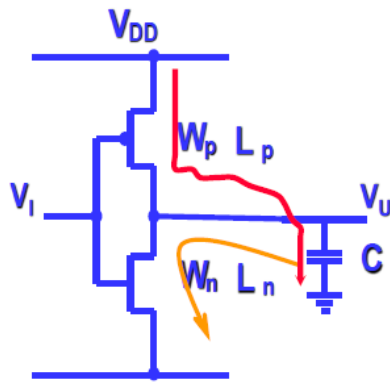
Figure 52- Schematico e tabella di verità

### Propagation delay

Il propagation delay indica quanto tempo ci vuole affinché un cambiamento applicato al nodo di input si rifletta sul nodo di output del circuito. È un parametro critico nelle prestazioni di un circuito, specialmente in applicazioni ad alta velocità o ad alta frequenza, poiché influisce direttamente sulla velocità massima a cui il circuito può operare.

Questo ritardo è dovuto al tempo impiegato dal segnale per attraversare i vari componenti del circuito, come porte logiche, interconnessioni, buffer, ecc. Infatti, può essere influenzato da vari fattori, tra cui la complessità del circuito, la tecnologia di produzione, la lunghezza e la capacità delle interconnessioni, la velocità dei componenti logici utilizzati, la temperatura e altri fattori ambientali.

Misurare e minimizzare il propagation delay è importante per garantire prestazioni ottimali del circuito e soddisfare i requisiti di timing dell'applicazione. Questo può essere fatto tramite tecniche di progettazione appropriata, selezione accurata dei componenti e delle tecnologie utilizzate, e utilizzo di strumenti di simulazione per analizzare e ottimizzare il comportamento del circuito.



$$t_{pHL} \propto \frac{KC}{\beta_n} \frac{1}{V_{DD} - V_{Th}}$$

$$t_{pLH} \propto \frac{KC}{\beta_p} \frac{1}{V_{DD} + V_{Tp}}$$

$$\beta_n = \mu_n C_{ox} \frac{W_n}{L_n} \quad \beta_p = \mu_p C_{ox} \frac{W_p}{L_p}$$

Abbiamo delle formule in cui possiamo sfruttare tutti i possibili trade-off (Area, Consumo di Potenza, Soglia dei transistor, ecc.).

Alimentazione e ritardo di propagazione

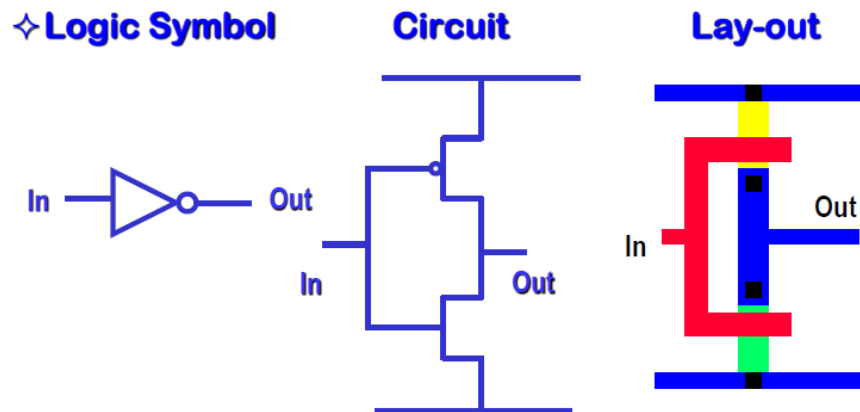
Dalle formule possiamo vedere come **aumentando l'alimentazione, riduciamo il ritardo di propagazione.**

Soglia e ritardo di propagazione

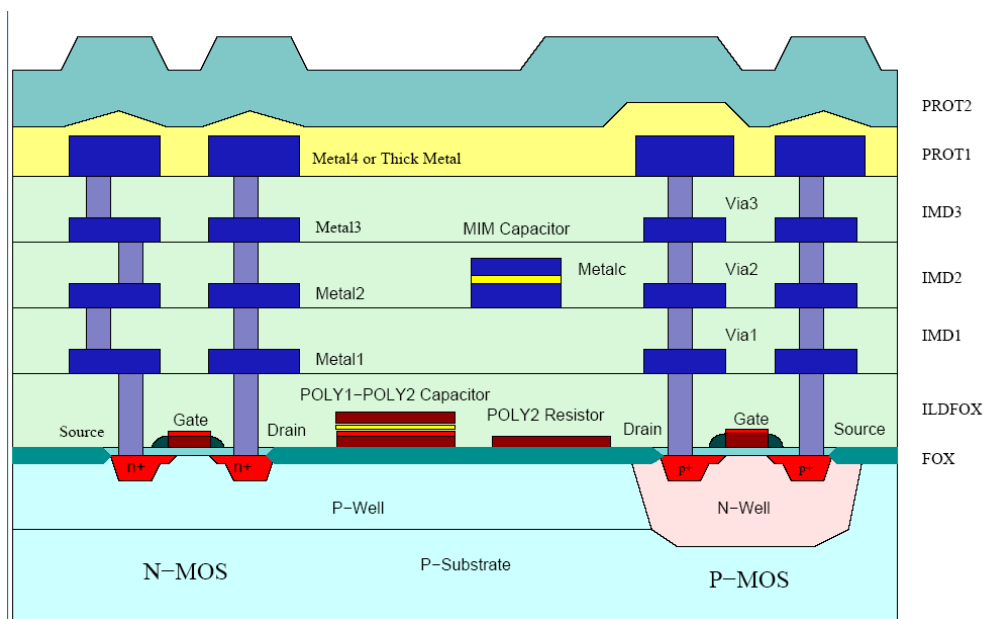
**Se riduciamo la soglia del transistor, diminuiamo il ritardo di propagazione,** ma aumentiamo anche la corrente di dispersione e quindi il consumo energetico.



## Interconnessione



Qui abbiamo i metalli (rettangoli blu) nelle interconnessioni, essi garantiscono maggiore flessibilità per le interconnessioni, ma sono necessarie le maschere che sono costose. Se riduciamo il numero di metalli, riduciamo anche il numero dei costi.



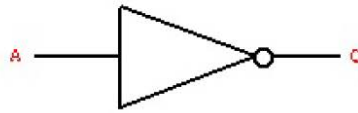
## Strenght

Più è alta la forza, più è alta la corrente in uscita, e quindi possiamo caricare o scaricare più velocemente la capacità in uscita.

Quindi con maggiore forza ma con la stessa capacità di carico, otteniamo un ritardo di propagazione più basso, ma abbiamo una dimensione maggiore.

Un transistor più largo fornisce una corrente più alta e un minore ritardo di propagazione.

Strength	1
Cell Area	29.120 um <sup>2</sup>
Equation	Q = "!A"
Type	Combinational
Input	A
Output	Q



State Table	
A	Q
L	H
H	L

Propagation Delay [ns]					
Input Transition [ns]		0.01		4.00	
Load Capacitance [fF]		5.00	100.00	5.00	100.00
A to Q	fall	0.04	0.41	-0.36	0.63
	rise	0.06	0.67	0.83	1.88

Output Transition [ns]					
Input Transition [ns]		0.01		4.00	
Load Capacitance [fF]		5.00	100.00	5.00	100.00
A to Q	fall	0.04	0.54	0.70	1.43
	rise	0.08	1.03	0.62	1.58

### Leakage

La potenza perduta a causa di perdite.

### Layout e cell area

Forma del circuito e area occupata dal layout del circuito, di solito espresso in picometri.

### Wafer Cross-Section

Come i livelli sono posizionati sul wafer.

### Technology Overview

Tabella che mostra i materiali e le tecniche usati per produrre il circuito.

### Symbol of the node

Simbolo del nodo.

### True Table

Esprime cosa fa il nodo.

### Capacità

Capacità parassite di ingresso e uscita.

### Strength

INVX1, il "1" rappresenta la forza del nodo. La forza è la capacità di pilotare il carico in uscita; quindi, alla quantità di corrente che il nodo è in grado di fornire in uscita. Una forza maggiore implica un minore ritardo di propagazione ma maggiore capacità e quindi maggiore consumo energetico.

## Transition Time (Ritardo di propagazione)

**Il tempo che il nodo impiega per cambiare il valore in uscita.**

Il ritardo di propagazione dipende dalle condizioni operative (temperatura, alimentazione, processo, ecc.). Ma il valore in questa tabella è calcolato utilizzando determinati "technology corners".

### Technology corners

Come posso stabilire le prestazioni di un nodo?

- **Fattori di produzione:** Alla fine del processo di produzione non tutti i nodi prodotti hanno le stesse caratteristiche, alcuni sono più veloci e altri sono più lenti.
- **Ritardo di propagazione:** Può dipendere da fattori esterni come la temperatura operativa (quindi la dalla mobilità degli elettroni), oppure l'alimentazione.

Quindi, i "technology corners" sono presenti perché come designer dobbiamo verificare il sistema al fine di tener conto di tali variazioni.

Propagation Delay [ns]					Output Transition [ns]						
Input Transition [ns]		0.01		4.00	Input Transition [ns]		0.01		4.00		
Load Capacitance [fF]		5.00	100.00	5.00	100.00	Load Capacitance [fF]		5.00	100.00		
A to Q	fall	0.04	0.41	-0.36	0.63	A to Q	fall	0.04	0.54	0.70	1.43
	rise	0.06	0.67	0.83	1.88		rise	0.08	1.03	0.62	1.58

Figure 53- Statistiche di prestazione di un dispositivo.

	Best	Typical	Worst
Temperatura	-40 - 0 C°	25 C°	85 - 125 C°
Alimentazione	1.1 * VDD	VDD	0.9 * VDD
Processo	FAST	TYPICAL	SLOW

Figure 54- Esempio di technology corners.

### Typical

In questo caso quando abbiamo una condizione tipica, abbiamo valori di alimentazione e temperatura dell'ambiente tipici.

### Worst

In questo caso se la condizione è peggiore, abbiamo:

- Un processo più lento.
- Una temperatura operativa molto alta.
- Alimentazione più bassa del 10%.

### Best

In questo caso, se la condizione è migliore, abbiamo:

- Un processo più veloce.
- Una temperatura operativa molto bassa.
- Alimentazione più alta del 10%.

Ad alta velocità, V è più lento, soglia più bassa, bassa dispersione, ha una propagazione più lunga.

Maggiore capacità, maggiore consumo di energia e forse l'area con maggiore forza.

Quindi, per questo ritardo, nel Databook abbiamo i numeri per queste 3 condizioni (technology corners).

I ritardi di propagazione si dividono ancora in 2 numeri: H → L e L → H.

Le celle standard devono avere alta velocità, bassa dispersione e alta densità.

## Rumore

Variazioni non volute del voltaggio e corrente nei nodi logici. Il rumore è dovuto agli elementi capacitivi (*rumore nel voltaggio*) e/o induttivi (*rumore nella corrente*) che si trovano tra due nodi.

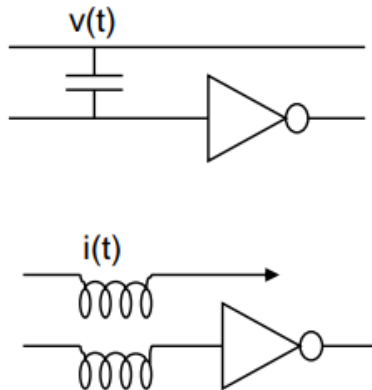


Figure 55- Tipiche situazioni in cui posso trovare rumore.

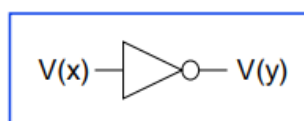
Il rumore può influenzare l'input di un nodo e creare comportamenti inaspettati.

## Comportamento statico (Steady state parameters)

Dice quando il circuito è resistente alle variazioni dovute dal processo di manifattura e dal rumore. Una variabile logica è associata a due soglie di voltaggio.

Soglie

$$1 \Leftrightarrow V_{OH} \text{ and } 0 \Leftrightarrow V_{OL}$$



$$V_{OH} = ! (V_{OL})$$

$$V_{OL} = ! (V_{OH})$$

Figure 56- Soglie dei valori logici.

$$V_{sw} = V_{OH} - V_{OL}$$

Figure 57- Signal swing.

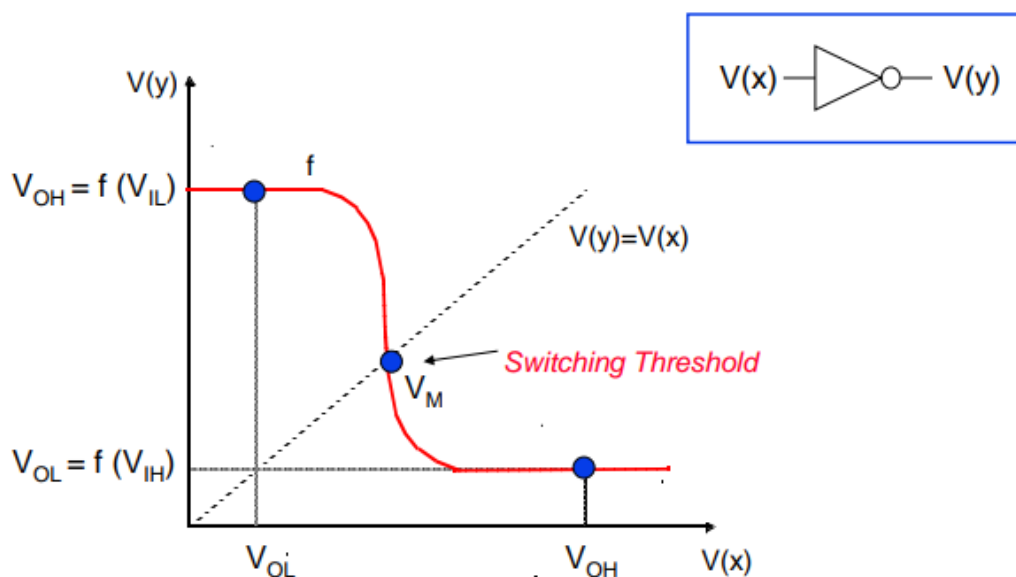


Figure 58- La posizione delle soglie determina la sensibilità dell'inverter, superata la soglia di switching il valore in uscita cambia.

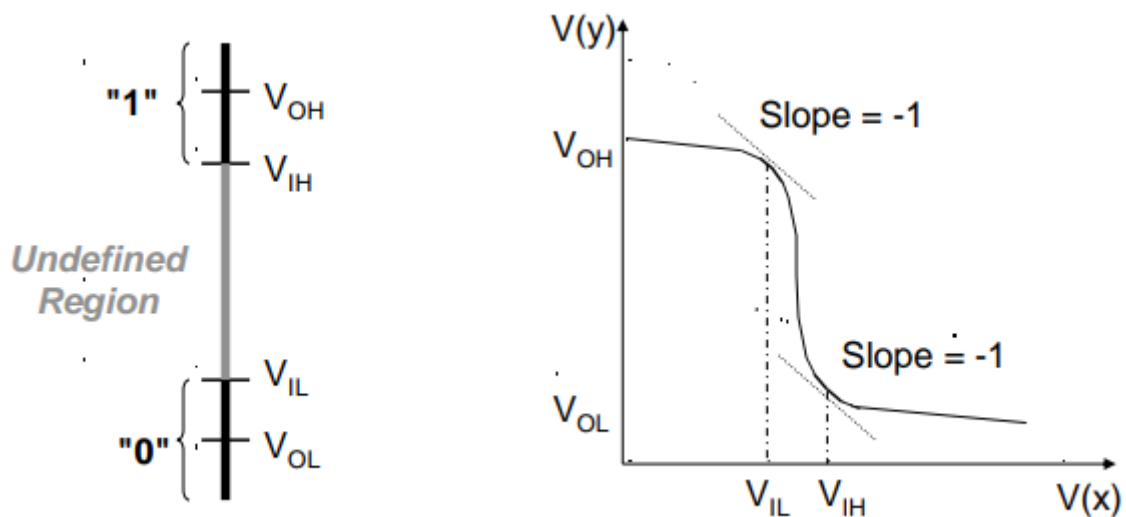


Figure 59- Le soglie non vengono scelte a caso.

La zona indefinita è il range di valori dell'input per cui ho un output non prevedibile.

#### Margini di rumore

Il nostro obiettivo è definire le soglie in modo che il nodo sia robusto a rumore in entrata. Quindi fare in modo che le zone definite (*quelle in nero*) siano più grandi possibile.

## Transistors

### CMOS Complementare

Con questa tecnica possiamo implementare qualsiasi funzione logica.

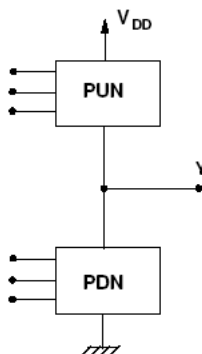


Figure 60- Schema di una rete implementata con complementary CMOS.

Dividiamo il nodo in due parti:

- **Pull-up network:** Connette l'output con l'alimentazione.
- **Pull-down network:** Connette l'output con la massa.

#### Pull-down Network (PDN)

Di solito la prima cosa che si fa è la PDN, che deriva direttamente dalla funzione logica da implementare.

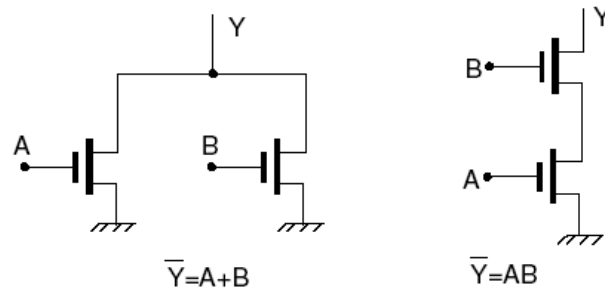


Figure 61- Esempio di pull-down network.

Nel primo caso (*quello a sinistra*), se  $A=1$  OR  $B=1$ , colleghiamo l'uscita Y a massa. Se  $A=1$ , il transistor sinistro è approssimabile ad un cortocircuito, a cui colleghiamo la terra con Y (*la stessa cosa per B e il transistor destro*).

Pull-up Network (PUN)

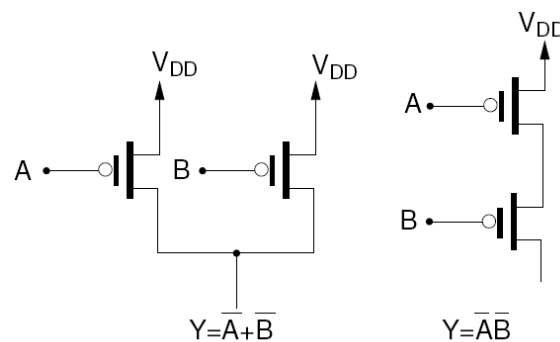


Figure 62- Esempio di pull-up network.

Nel primo caso (*a sinistra*), se  $A=1$  OR  $B=1$ , colleghiamo l'uscita Y all'alimentazione.

$$\bar{Y} = f_{PDN}(A, B)$$

$$Y = f_{PUN}(\bar{A}, \bar{B})$$

Figure 63- La PDN e la PUN implementano due funzioni logiche diverse che se unite producono la funzione logica voluta.

NAND2

Implementiamo una NAND a due ingressi.

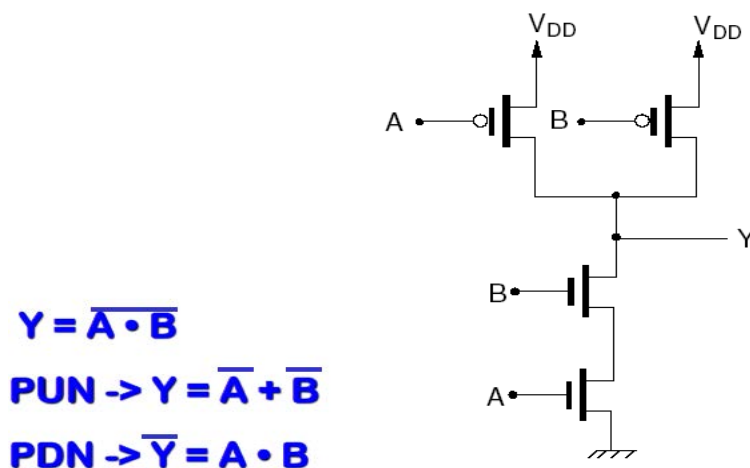


Figure 64- Schema della NAND.

## NOR2

Implementiamo una NOR a due ingressi.

$$Y = \overline{A + B}$$

$$\text{PUN} \rightarrow Y = \overline{A} \cdot \overline{B}$$

$$\text{PDN} \rightarrow \overline{Y} = A + B$$

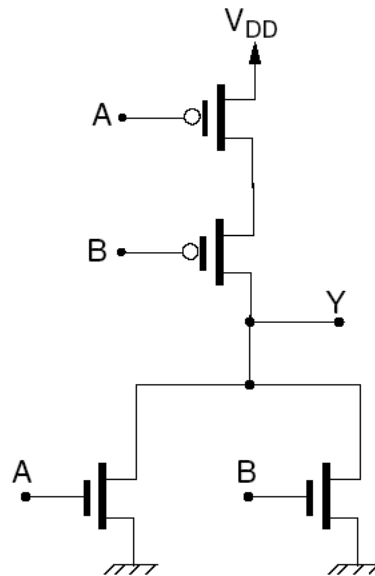


Figure 65- Schema della NOR.

Funzioni complesse (Teoria della dualità)

Una funzione complessa può essere prodotta con la legge della dualità.

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

$$\begin{aligned} \text{PUN} \rightarrow Y &= \overline{A(B + CD)} \\ &= \overline{A} + \overline{B + CD} \\ &= \overline{A} + \overline{B} \cdot \overline{CD} \\ &= \overline{A} + \overline{B}(\overline{C} + \overline{D}) \end{aligned}$$

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

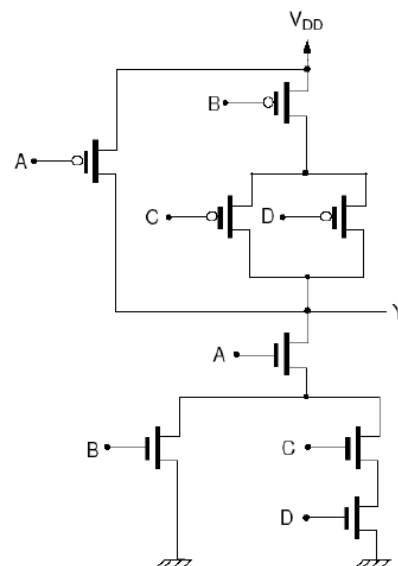


Figure 66- Esempio di utilizzo della teoria della dualità.

## XOR

L'XOR non rispetta la teoria della dualità, quindi non può essere implementato con questa tecnica. Dobbiamo implementarlo manualmente. L'XOR non rispetta la teoria della dualità, quindi non può essere implementato con questa tecnica. Dobbiamo implementarla manualmente.

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

## XOR

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

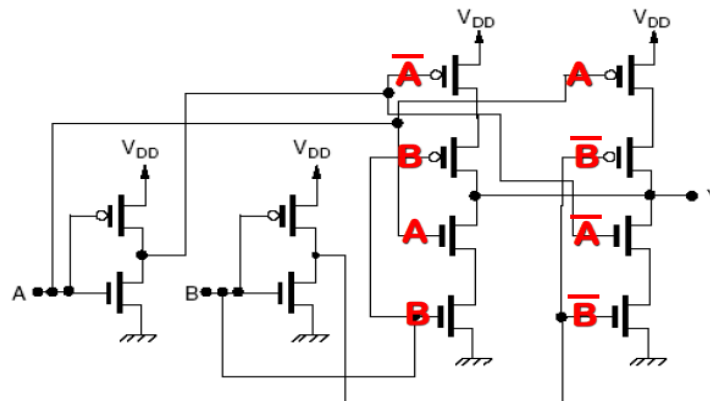


Figure 67- XOR implementata manualmente.

## Complementary CMOS e il ritardo di propagazione

Dopo aver costruito una rete utilizzando questa tecnica come possiamo calcolare il ritardo di propagazione? Nel caso dell'inverter abbiamo queste due formule:

**Dynamic A** 
$$t_{pHL} \propto \frac{KC}{\beta_n} \frac{1}{V_{DD} - V_{Tn}}$$

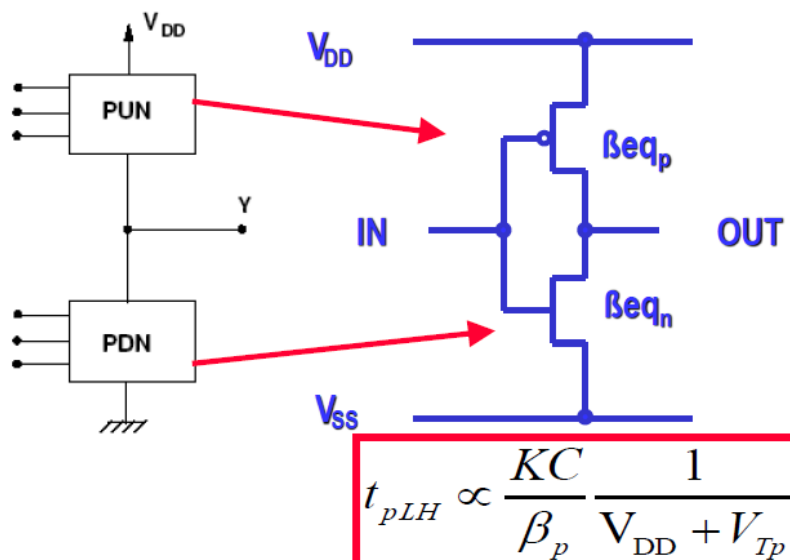


Figure 68- Formule dei due ritardi di propagazione.



$$N < 4$$

#### Transistor in parallelo

La conduttanza equivalente è la somma delle conduttanze.

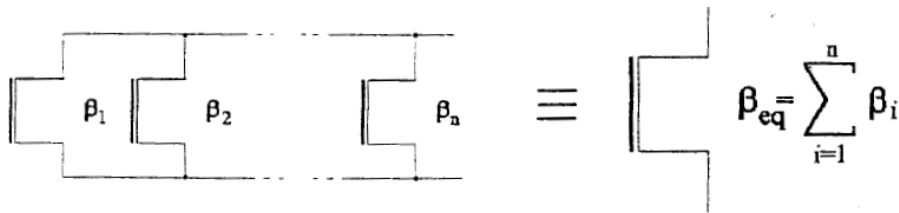


Figure 69- Transistor in parallelo.

#### Transistor in serie

La conduttanza equivalente è il reciproco della somma dei reciproci delle conduttanze.

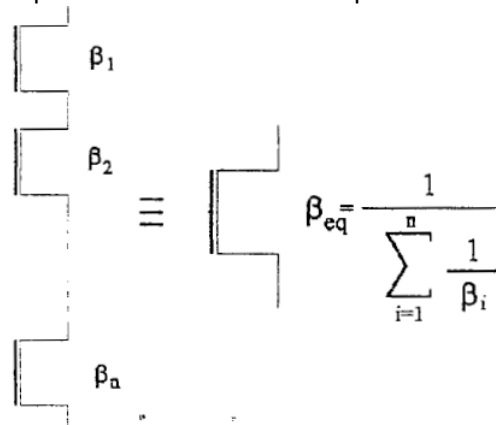
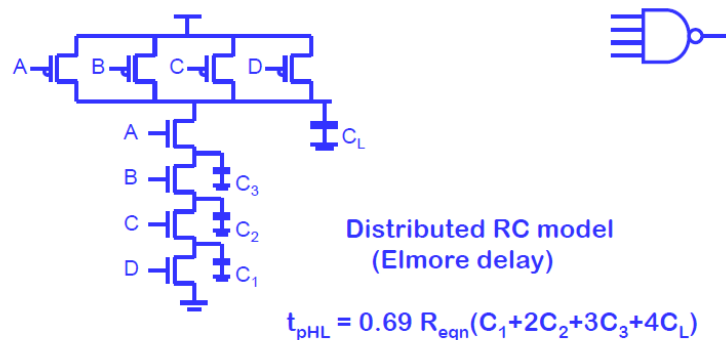


Figure 70- Transistor in serie.

Queste formule sono valide solo per un numero di transistor in parallelo/serie minore di 4, poiché oltre al 4 non posso più approssimare i transistor a componenti passivi.

$$N \geq 4$$

Se abbiamo 4 o più transistor, il ritardo aumenta in modo quadratico, poiché dobbiamo considerare anche la capacità (*non sono più approssimabili ad elementi passivi*):



Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case

64

Se ho  $M$  transistor in parallelo, considero 1 capacità.

Se ho  $N$  transistor in serie, considero  $(N - 1)$  capacità.

## Analisi Dinamica (NAND)

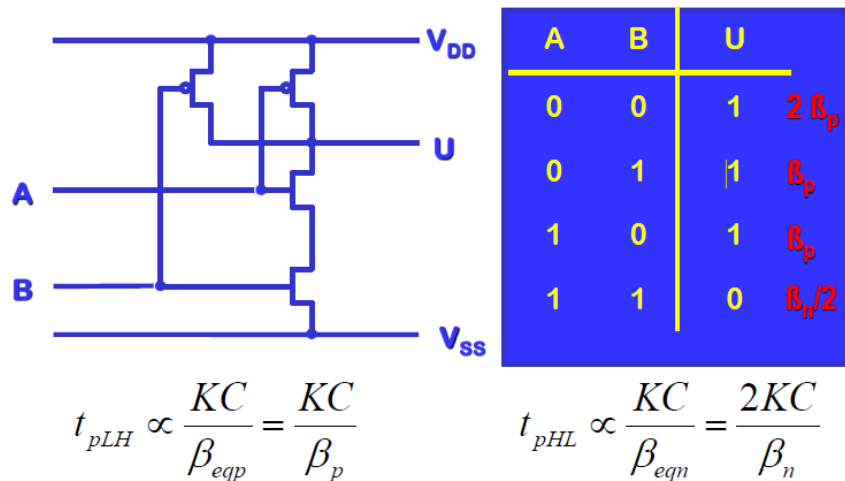


Figure 71- Analisi dinamica di una NAND.

**Il mio obbiettivo è calcolare il  $\beta_{eq}$  per ogni percorso.**

Per il percorso  $\{A, B\} = \{0, 0\}$  solo i PMOS saranno attivi.

I PMOS hanno  $\beta = \beta_p$  e ho 2 PMOS in parallelo; quindi, il  $\beta_{eq}$  sarà la somma di tutti i  $\beta$ , che è  $2\beta_p$ .

Per il percorso  $\{A, B\} = \{0, 1\}$  solo un PMOS (A) e un NMOS (B) saranno attivi.

Il PMOS e il NMOS non sono collegati; quindi, il  $\beta_{eq}$  sarà uguale  $\beta_p$ , perché è il percorso tra Vdd e U.

Per il percorso  $\{A, B\} = \{1, 0\}$  solo un PMOS (B) e un NMOS (A) saranno attivi.

Il PMOS e il NMOS sono collegati, ma il NMOS è in un circuito aperto quindi il GND non può raggiungere U e l'output sarà 1.

Il  $\beta_{eq}$  sarà uguale a  $\beta_p$ , perché è il percorso tra Vdd e U.

Per il percorso  $\{A, B\} = \{1, 1\}$  solo gli NMOS saranno attivi.

Il NMOS ha  $\beta = \beta_n$  e ho 2 NMOS in serie, quindi il  $\beta_{eq}$  sarà: il reciproco della somma dei reciproci dei  $\beta_n$ .

### Consiglio per l'analisi dinamica

Gestendo l'area possiamo fare in modo che il ritardo  $H \rightarrow L$  sia uguale a quello  $L \rightarrow H$ .

$$t_{pLH} \propto \frac{KC}{\beta_{eqp}} = \frac{KC}{\beta_p} \quad t_{pHL} \propto \frac{KC}{\beta_{eqn}} = \frac{2KC}{\beta_n}$$

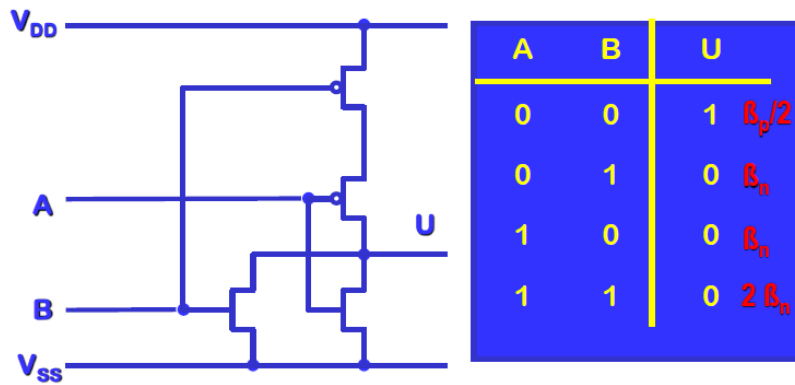
✧  **$t_{pHL} = t_{pLH}$  in the worst case**

$$\beta_p = \frac{\beta_n}{2} \rightarrow \beta_n = 2\beta_p \rightarrow \mu_n \frac{W_n}{L_n} = 2\mu_p \frac{W_p}{L_p}$$

**If**  $\mu_p = \frac{\mu_n}{2} \quad L_n = L_p = L_{\min} \quad \rightarrow \quad W_n = W_p$

Figure 72- Modificando l'area del PMOS e/o NMOS si può fare in modo che il rapporto nella seconda riga sia lo stesso.

### Analisi dinamica di una NOR



**52 If**  $\mu_p = \frac{\mu_n}{2} \quad L_n = L_p = L_{\min} \quad \rightarrow \quad W_p = 4W_n$

Figure 73- Analisi dinamica di una NOR

### Confrontare due soluzioni: NOR vs NAND

Applichiamo il consiglio di prima, sia nella NOR che nella NAND. Quindi il ritardo di propagazione  $H \rightarrow L$  è uguale a quello  $L \rightarrow H$ .

$$\beta_{eqNAND} = \beta_p^{NAND} = \frac{\beta_n^{NAND}}{2}$$

$$\beta_{eqNOR} = \beta_n^{NOR} = \frac{\beta_p^{NOR}}{2}$$

$$W_n^{NOR} = W \quad W_p^{NOR} = 4W$$

$$\beta_n^{NAND} = 2\beta_n^{NOR} \Rightarrow W_n^{NAND} = 2W_n^{NOR} = 2W$$

$$\beta_p^{NAND} = \frac{\beta_p^{NOR}}{2} \Rightarrow W_p^{NAND} = \frac{W_p^{NOR}}{2} = \frac{4W}{2} = 2W$$

Per avere la simmetria dei due ritardi di propagazione in una porta NAND, dobbiamo garantire che le condizioni di ingresso che attivano ciascun percorso attraverso la porta siano equivalenti. In altre parole, sia il percorso attraverso l'ingresso PMOS che quello attraverso l'ingresso NMOS devono avere la stessa complessità e/o lo stesso numero di transistor attivi.

Ciò significa che dovremmo avere lo stesso numero di transistor PMOS e NMOS attivi in ciascun percorso o comunque che la complessità dei percorsi sia sempre la stessa. Questo assicurerà che i ritardi di propagazione siano simmetrici.

Quando entrambe le porte, NAND e NOR, hanno la stessa velocità, ciò significa che i loro ritardi di propagazione sono uguali o simili. Questo implica che il ritardo attraverso i percorsi critici di entrambe le porte è bilanciato, il che può avvenire regolando le dimensioni dei transistor e altri parametri del circuito.

At the end of the story

NOR  $\rightarrow 4W + W \rightarrow (x2) \rightarrow 10W$

NAND  $\rightarrow 2W + 2W \rightarrow (x2) \rightarrow 8W$

La porta NOR è più complessa della NAND.

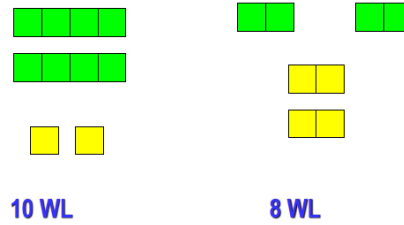


Figure 74 - A sinistra la NOR e a destra la NAND.

Percorso critici di una rete combinatoria

Colorati in rosso e verde abbiamo i 2 percorsi più critici, ossia quelli che attraversano più Transistors.

A	B	C	D	Y	B
0	0	0	0	1	5/3
0	0	0	1	1	3/2
0	0	1	0	1	3/2
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	2/3
1	0	0	1	1	1/2
1	0	1	0	1	1/2
1	0	1	1	0	1/3
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	4/3

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

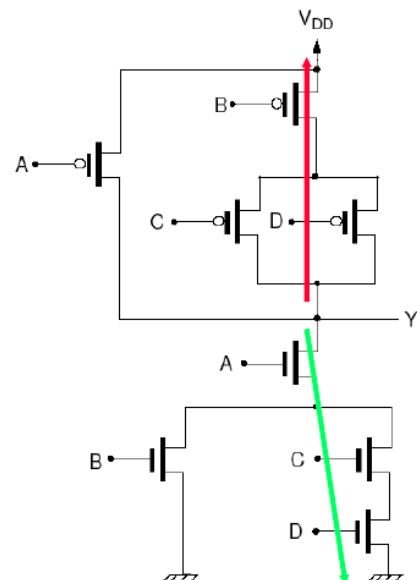


Figure 75- I percorsi critici.

Vediamo come aggiustare le cose:

$$t_{pLH} \propto \frac{KC}{\beta_{eq_p}} \qquad t_{pHL} \propto \frac{KC}{\beta_{eq_n}}$$

$$\beta_{eq_p} = \frac{\beta_p}{2} = \frac{\mu_p}{2} \frac{W_p}{L_p} \qquad \beta_{eq_n} = \frac{\beta_n}{3} = \frac{\mu_n}{3} \frac{W_n}{L_n}$$

✧  $t_{pHL} = t_{pLH}$  in the worst case

$$\frac{\mu_p}{2} \frac{W_p}{L_p} = \frac{\mu_n}{3} \frac{W_n}{L_n}$$

If  $\mu_p = \frac{\mu_n}{2} \quad L_n = L_p = L_{\min} \quad \longrightarrow \quad \frac{W_p}{W_n} = \frac{4}{3}$

Figure 76- Calcoli

Analizziamo i percorsi critici:

In quello dei PMOS abbiamo un PMOS in serie a 2 PMOS in parallelo, quindi  $\beta_{eq}$  è  $\beta_p/2$ .

In quello dei NMOS abbiamo 3 NMOS in serie, quindi  $\beta_{eq}$  è  $\beta_n/3$ .

I due  $\beta_{eq}$  devono essere uguali, quindi per renderli tali operiamo in  $W_p$  e  $W_n$ .

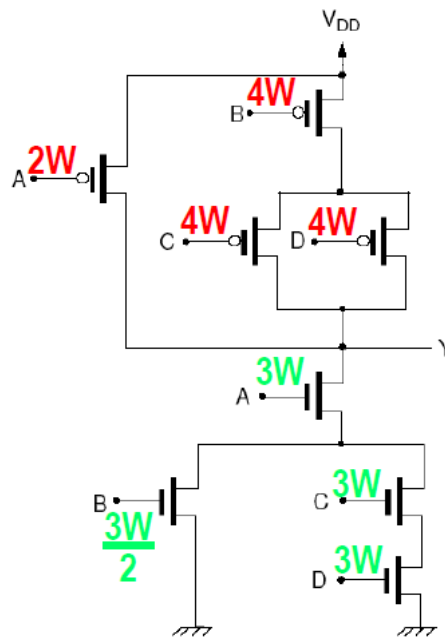


Figure 77- Soluzione ai percorsi critici.

Aggiustando l'area di ogni transistor posso fare in modo che ogni percorso abbia la stessa complessità e che quindi non ne esista uno critico.

### CMOS gate – Body effect

Il Body Effect nei circuiti CMOS è un fenomeno che si verifica a causa della differenza di potenziale tra il substrato di silicio (body) e la sorgente del transistor MOSFET.

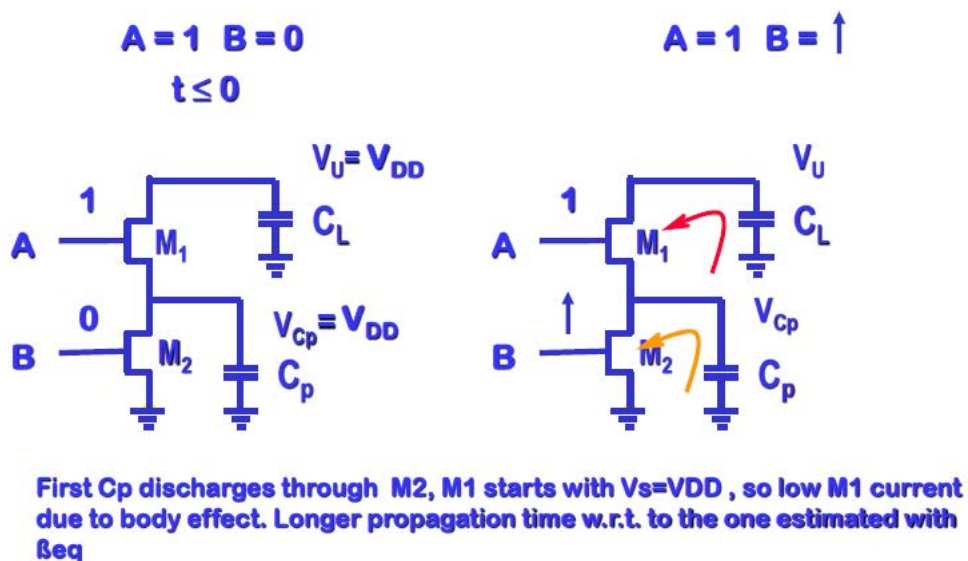
Nei transistor MOSFET, la regione di silicio sotto il canale è di solito connessa al substrato, che può essere portato a un potenziale diverso rispetto alla sorgente del transistor. Questa differenza di potenziale può influenzare le caratteristiche elettriche del transistor.

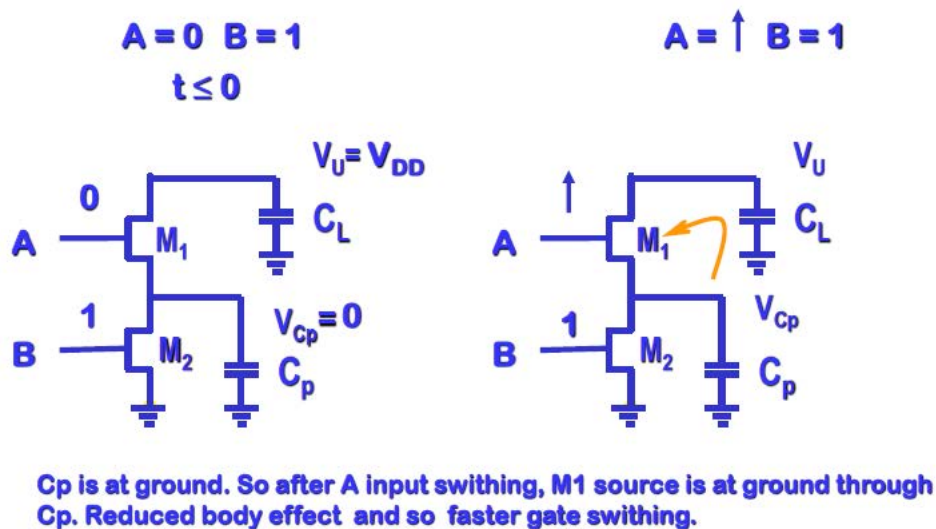
Nel caso dei transistor CMOS, che utilizzano sia transistor N-channel (NMOS) che transistor P-channel (PMOS) all'interno della stessa cella logica, l'effetto di corpo può influenzare in modo significativo le prestazioni del circuito.

L'effetto di corpo è particolarmente rilevante nei transistor P-channel, dove il substrato di silicio è generalmente collegato al potenziale di alimentazione positivo (Vdd). Quando il transistor P-channel è acceso, il substrato assume il potenziale di Vdd, il che crea una differenza di tensione tra il substrato e la sorgente del transistor. Questa differenza di tensione riduce la larghezza del canale effettivo del transistor, aumentando così la resistenza del canale e riducendo la conducibilità del transistor stesso. Ciò porta a una riduzione della corrente di drain (ID) del transistor P-channel, influenzando negativamente le prestazioni del circuito.

Per mitigare l'effetto di corpo, possono essere adottate diverse strategie, tra cui l'uso di strutture di transistor speciali, la modifica delle tensioni di polarizzazione o l'utilizzo di tecniche di progettazione specifiche che tengono conto di questo fenomeno.

È importante considerare l'effetto di corpo durante la progettazione di circuiti CMOS per garantire che le prestazioni del circuito siano robuste e coerenti, specialmente nelle applicazioni ad alta precisione o a bassa potenza.





Le conseguenze dell'effetto di corpo (Body Effect) nei transistor CMOS possono essere diverse e includono:

**Variazione delle caratteristiche di guadagno e corrente:** L'effetto di corpo può influenzare le caratteristiche di guadagno e corrente dei transistor CMOS, specialmente nei transistor P-channel. Questo può portare a una variazione delle prestazioni del circuito, come una maggiore resistenza di canale e una minore corrente di drain del transistor P-channel, influenzando il funzionamento del circuito.

**Variazione del ritardo di propagazione:** L'effetto di corpo può influenzare il ritardo di propagazione dei segnali attraverso i transistor CMOS, poiché **può influenzare la velocità di commutazione dei transistor**. Ciò può portare a una variazione del ritardo di propagazione tra i percorsi critici del circuito, influenzando il tempo di risposta complessivo del circuito.

**Effetto sulla tensione di soglia:** L'effetto di corpo può influenzare la tensione di soglia dei transistor CMOS, specialmente nei transistor P-channel. Questo **può portare a una variazione della tensione di soglia dei transistor e quindi delle loro caratteristiche operative**, influenzando le prestazioni del circuito.

**Consumo di potenza:** L'effetto di corpo può influenzare il consumo di potenza dei transistor CMOS, specialmente nei transistor P-channel, poiché può portare a una riduzione della corrente di drain e quindi a una maggiore resistenza di canale. Questo può aumentare la potenza dissipata nei transistor e ridurre l'efficienza energetica complessiva del circuito.



### Fan-in

Indica il numero di input (ingressi) che possono essere collegati a una porta logica senza comprometterne il funzionamento corretto. In altre parole, il numero massimo di segnali di input che può gestire contemporaneamente.

Influisce sulla complessità del circuito, sulle prestazioni e sulla velocità di commutazione. Idealmente, **si desidera mantenere il fan-in il più basso possibile** per evitare ritardi eccessivi e mantenere la stabilità del circuito.

Ad esempio: in una porta logica AND con un fan-in di 2, significa che la porta può accettare fino a due segnali di input. Se si tenta di collegare più di due segnali di input, la porta potrebbe non funzionare correttamente o potrebbe verificarsi un ritardo maggiore nella propagazione del segnale attraverso la porta.

### Consiglio sul fan-in

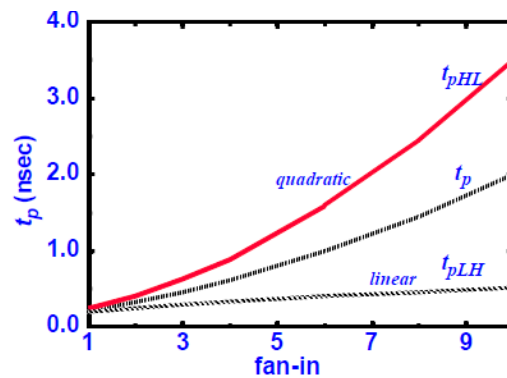


Figure 78- Effetti del fan-in sul ritardo di propagazione.

Porte con un fan-in maggiore di 4 dovrebbero essere evitate, perché può portare ad un aumento esponenziale del ritardo di propagazione.

Esempio di analisi dinamica (NAND)

$$Y = \overline{A_1 \cdot A_2 + B_1 \cdot B_2}$$

PDN  $\overline{Y} = A_1 \cdot A_2 + B_1 \cdot B_2$

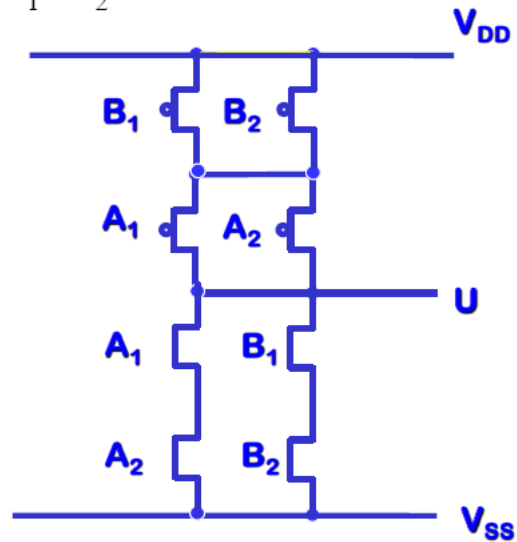


Figure 79 - Schematico della NAND.

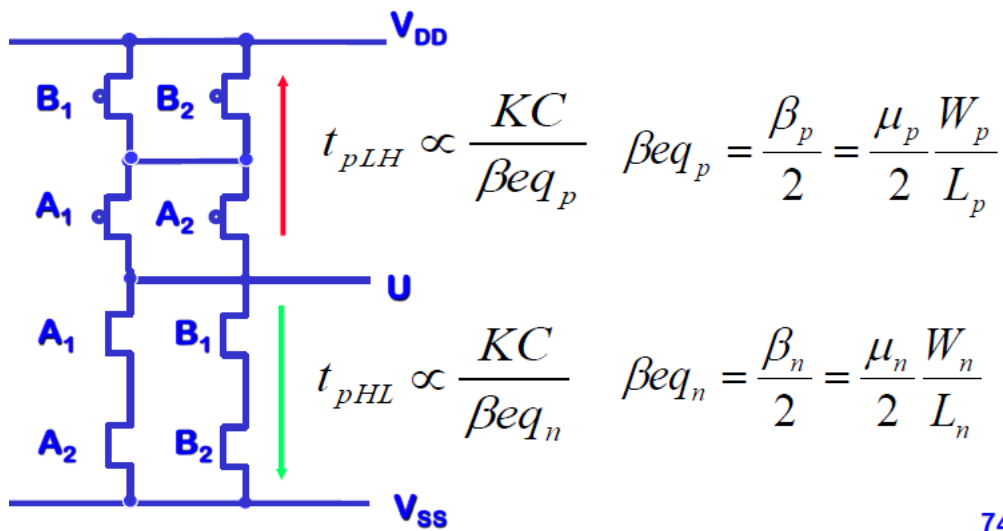


Figure 80 - Calcoli.

74

Catena di inverter

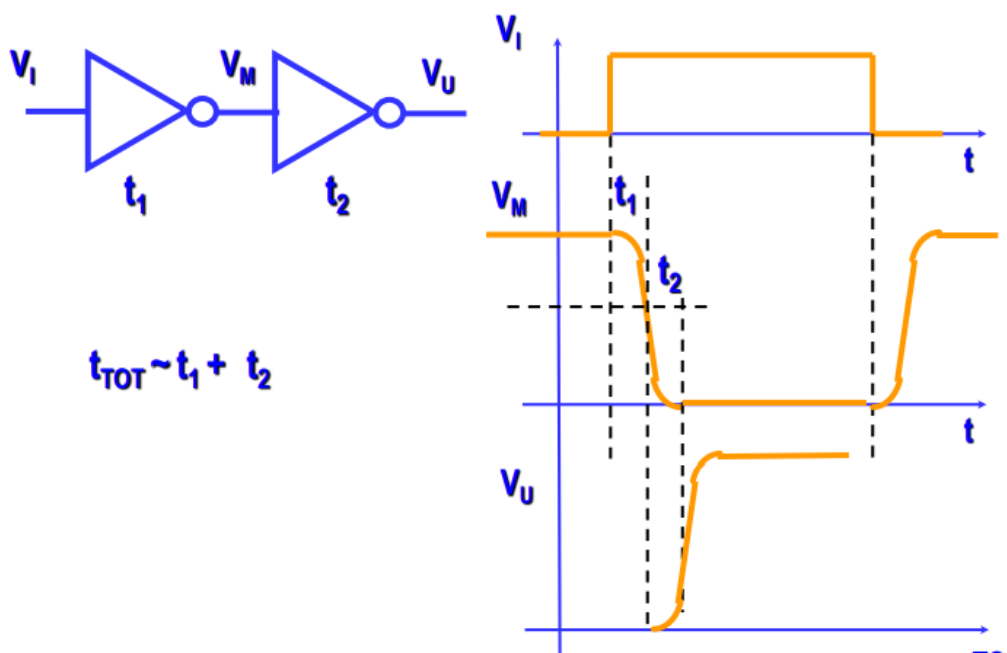


Figure 81- Una catena composta da due inverter e il comportamento equivalente risultante.

## Timing

Il timing è molto importante quando abbiamo sistemi con registri. Per timing si intende il come regolare il clock, il quale è in comune per ciascun registro.

Analizzando i percorsi RLR (*registro-logica-registro*) possiamo verificare facilmente i requisiti di timing del sistema.

### Latch

I latch sono sensibili al valore del clock.

- Se il clock è 1, output trasparente all'input.
- Se il clock è 0, output non è trasparente all'input e mostra l'ultimo valore dell'input (*memorizzazione*).

### Flip Flop Edge Triggered (Registro)

I flip-flop sono sensibili a cambio di valore del clock, l'output cambia solo al fronte positivo del clock (quando diventa 1).

- Quando il clock è 1 e c'è una transizione, l'uscita assume il valore dell'input corrente.
  - o Durante questo tempo, il registro è trasparente.
- Quando il clock è 0, l'input non viene più ascoltato e l'uscita è conservata.

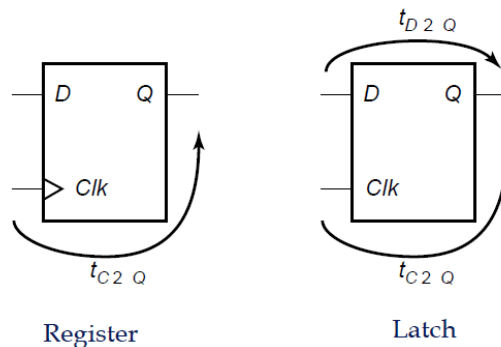


Figure 82 - Un registro e un latch a confronto.

### Clock Skew

La conseguenza di usare i registri è che tutti i circuiti che li includono dipendono dal clock.

I registri devono vedere tutti lo stesso fronte di salita del clock.

Questa condizione viene meno a fronte di in una grande area; in questo caso c'è uno sfasamento del clock, che è un delta (ritardo) tra il timing del clock di due registri posizionati in due punti distanti del circuito.

Anche un piccolo ritardo può causare problemi di desincronizzazione, i quali producono effetti imprevedibili.

### Gals (Global Asynchronous Locally Synchronous)

Approccio usato negli FPGA. Dividiamo il chip in aree, ognuna delle quali ha il suo clock. Ogni area è collegata con altre aree attraverso una comunicazione asincrona. Questo approccio serve a ridurre lo sfasamento del clock. Il nuovo problema è quello di gestire la comunicazione tra le aree, che funziona utilizzando diverse frequenze di clock.

## Classificazione dei tempi

Supponiamo che dobbiamo impostare il timing tra due registri per un'operazione di registro-registro. Dobbiamo comprendere i vincoli temporali:

### Tsetup e Thold

L'input deve essere stabile per un tempo TSetup prima del fronte di salita del clock e per un tempo THold dopo il fronte di salita del clock.

### Tc-q

Il tempo tra il fronte di salita del clock e l'istante in cui l'uscita diventa stabile.

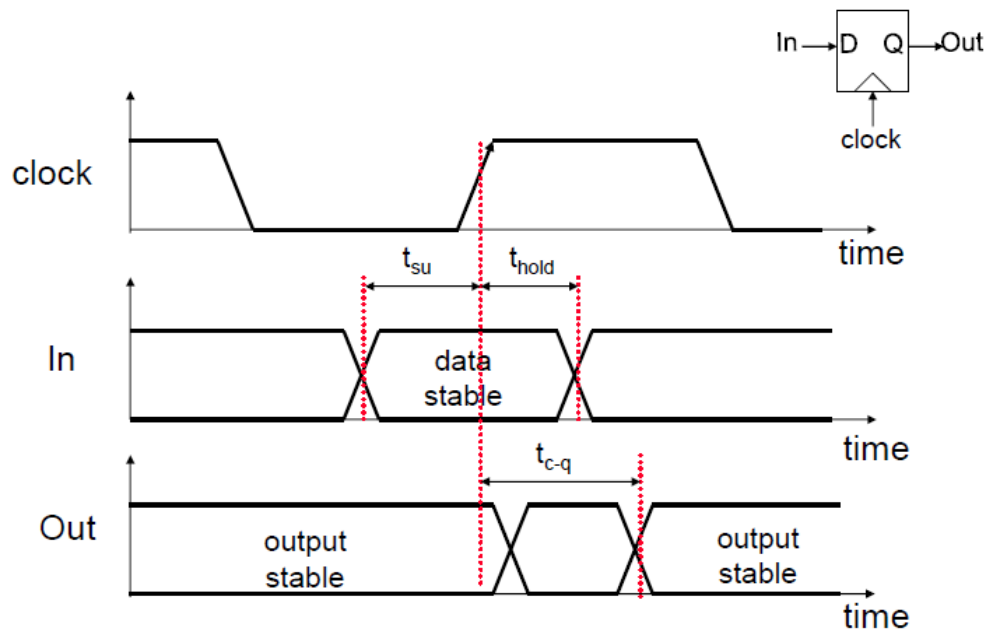


Figure 83- TSetup, THold e Tc-q in azione.

### Tplogic

Una rete logica possiede un tempo di attraversamento (*tempo di propagazione*) che va contato e che di sicuro è maggiore rispetto al tempo di propagazione su un semplice filo.

### Worst possible propagation delays

Definiamo anche il ritardo di propagazione  $t_{PHL}$  e  $t_{PLH}$ , che sono i peggiori ritardi di propagazione possibili.

### Tcdlogic

In una rete logica posso avere glitch o comunque instabilità temporanee sull'output. Definiamo il ritardo di contaminazione ( $t_{cdlogic}$ ) come il ritardo dalla transizione dell'input al PRIMO cambiamento dell'output. Nota che il primo cambiamento non è quello "buono" ma è quando l'output viene contaminato a causa del cambiamento dell'input, non è quando l'output diventa stabile.

## Timing violations

Supponiamo di avere due registri R1 e R2, collegati da una rete combinatoria.

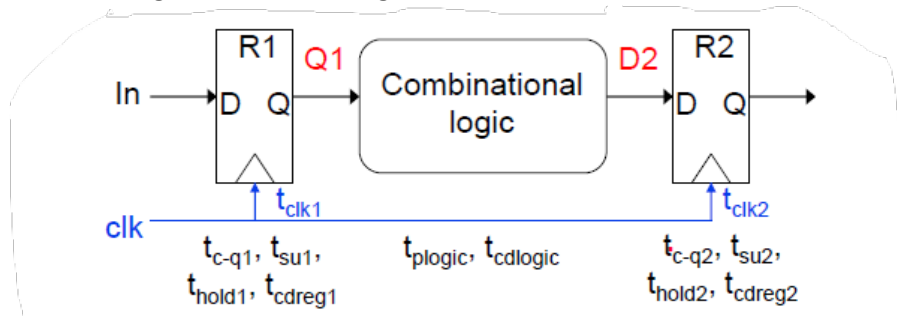


Figure 84- Rete di esempio.

### Supposizioni

- R1 e R2 hanno lo stesso clock con periodo T.
- Non ci importa (e non sappiamo nulla) di cosa succede prima di R1.
- La logica combinatoria introduce un ritardo T<sub>plogic</sub>.

Supponiamo che al fronte di salita del clock l'input di R1 cambi.

- 1 Prima del fronte di salita del clock, IN deve essere stabile per T<sub>setup1</sub>.
  - 2 Fronte di salita del clock → R1 e R2 sono in ascolto degli ingressi.
  - 3 Dopo T<sub>cq1</sub> → Q1 cambia e diventa stabile.
  - 4 D2 (input di R2) deve essere stabile per almeno T<sub>hold2</sub>.
  - 5 Dopo T<sub>cq2</sub> → Q2 cambia e diventa stabile.
  - 6 Fronte di discesa del clock → R1 e R2 non sono più in ascolto degli ingressi.
- Dobbiamo fare in modo che T sia abbastanza grande da soddisfare questi due requisiti.

### Setup time violation

$$T \geq t_{cq1} + t_{plogic} + t_{setup2}$$

Quando si viola la prima regola: **D2 cambia subito prima del fronte di salita.**

- T<sub>cq1</sub> → Tempo richiesto da R1 per adeguare l'uscita all'ingresso.
- T<sub>plogic</sub> → Tempo richiesto per attraversare la rete combinatoria.
- T<sub>setup2</sub> → Tempo richiesto da R2 per campionare in modo corretto.

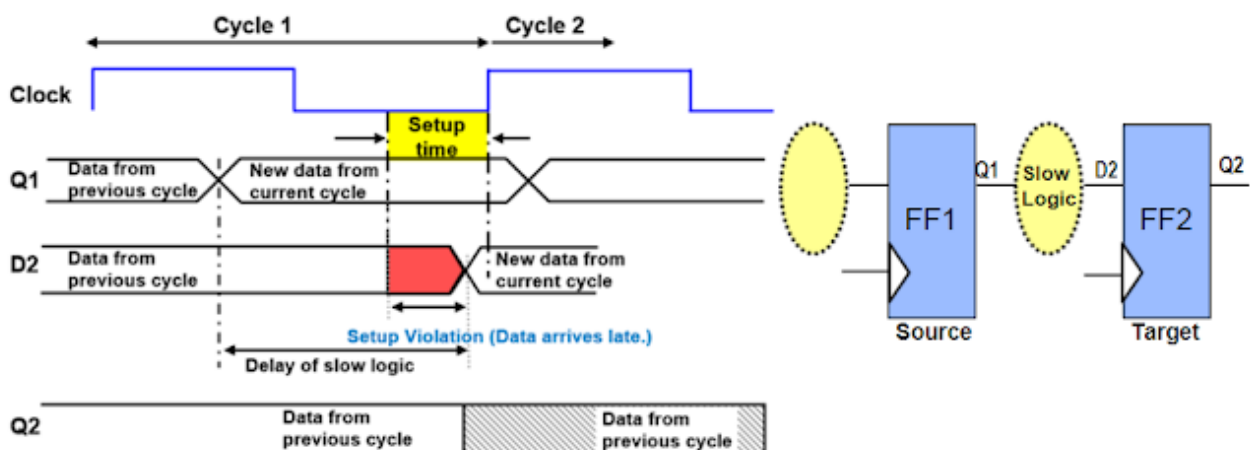


Figure 85- Esempio di setup violation, in questo caso causata da una rete combinatoria molto lenta.

Hold time violation

$$t_{hold2} \leq t_{cdReg1} + t_{cdLogic}$$

Quando si viola la seconda regola: **D2 cambia subito dopo al fronte di salita.**

- $t_{cdReg1}$  → Tempo richiesto a D1 per propagarsi attraverso R1 e uscire come segnale di uscita stabile (ossia quanto tempo serve al registro per aggiornare l'uscita).
- $t_{cdLogic}$  → Tempo di contaminazione della rete combinatoria, ossia il tempo che trascorre dalla variazione dell'input alla stabilizzazione dell'output della rete combinatoria (ossia quanto è reattiva la rete).

Se la condizione è rispettata, allora (supponendo che il segnale D1 cambi contemporaneamente al fronte di salita del clock) la propagazione del nuovo segnale sarà troppo lenta per influire sul campionamento del segnale precedente e quindi non avrà conseguenze indesiderate (ossia non influirà nel campionamento del valore precedente).

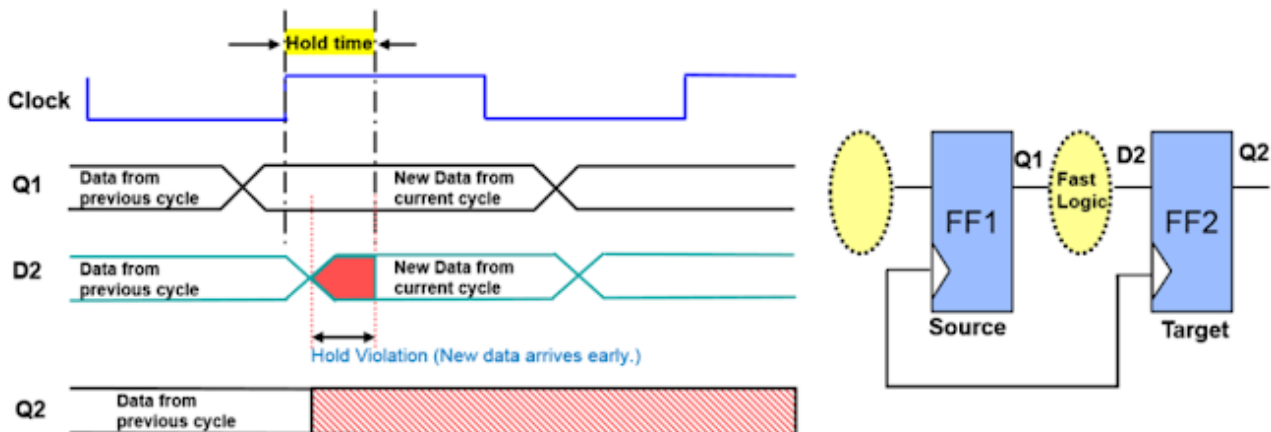


Figure 86- Hold violation, causato da una rete combinatoria troppo veloce.

Come evitarle?

Per evitare queste due problematiche è possibile impostare bene il periodo di clock oppure utilizzando tecniche di sincronizzazione, come il ritardo dei segnali di dati rispetto al segnale di clock (*clock skew*), per garantire il corretto funzionamento del registro e prevenire violation.

Per evitare la violazione della seconda regola è possibile anche rallentare la logica tra i due registri tramite l'inserimento di un buffer.

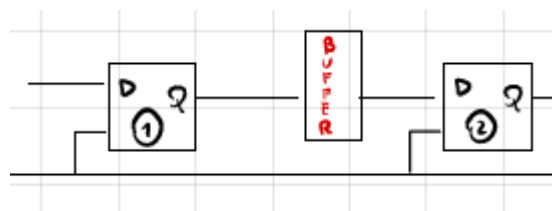


Figure 87- Il buffer ritarda il segnale tra i due registri.

## Timing e strumenti di sintesi

1. Lo strumento trova ogni possibile percorso che un segnale potrebbe seguire.
2. Per ogni percorso, calcolerà il tempo di propagazione lungo quel percorso.
3. **Il percorso critico è quello con il maggiore tempo di propagazione**, che determina anche la durata minima del periodo di clock.

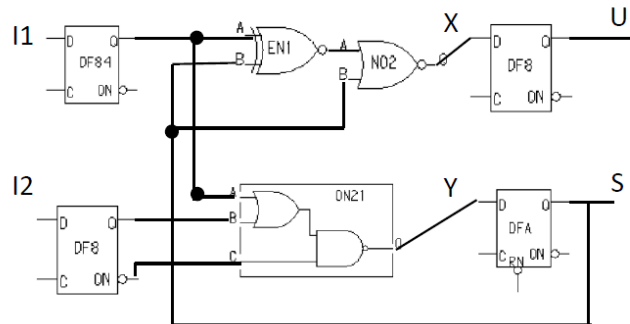


Figure 88 - Rete di esempio.

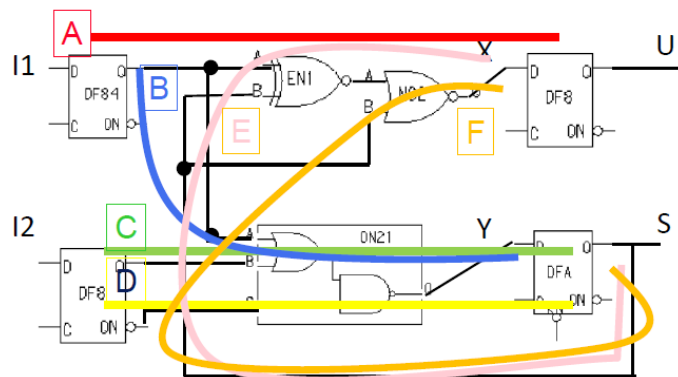
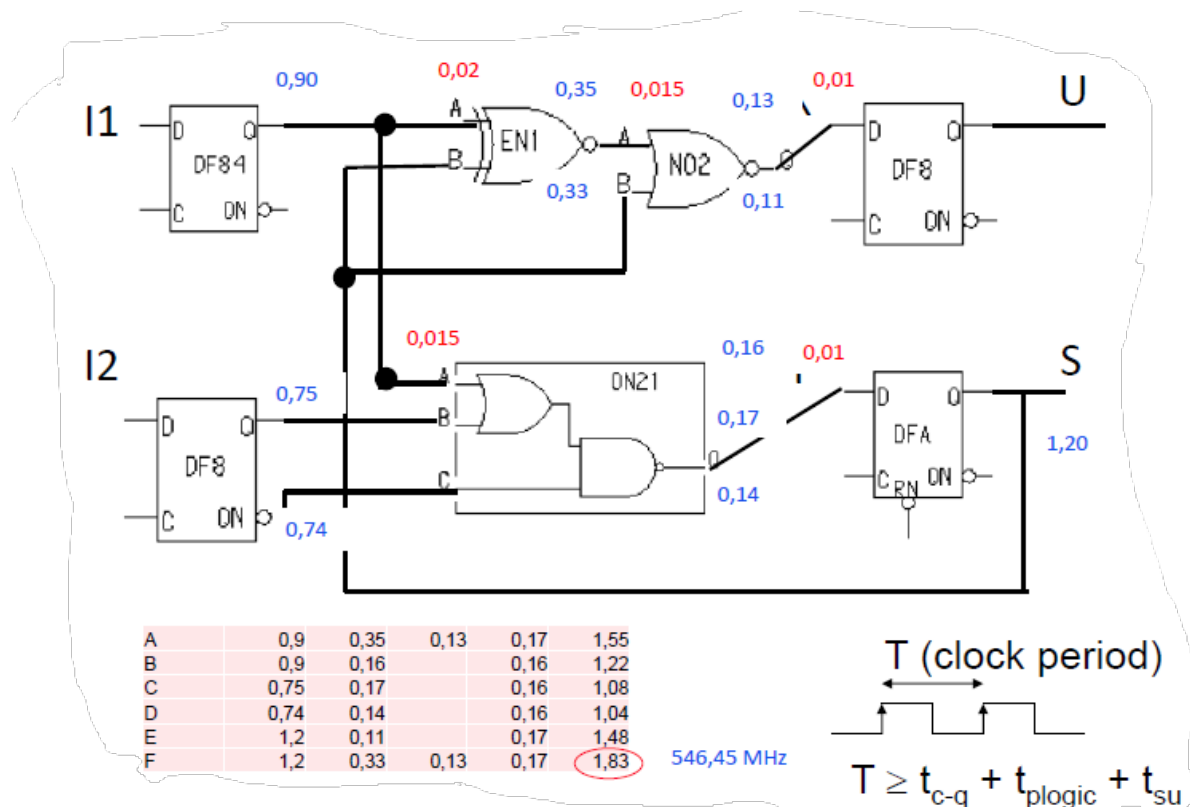


Figure 89 - Calcolo di ogni possibile percorso RLR (Register-Logic-Register).

Per ogni componente, il tool possiede il databook dal quale legge ogni possibile informazione utile:

- Tabella di verità → Per escludere percorsi che non avverranno mai.
- Area e processo di fabbricazione.
- Potenza dissipata.
- Tempo di propagazione per ogni ingresso per ogni uscita.
- Capacità di **ingresso** e di **uscita**.
- Nel caso dei registri: Tsetup e Thold.





In questo caso lo strumento ci dice che il percorso F è quello che richiede più tempo; quindi, lui è quello critico e il periodo clock deve essere maggiore di quel tempo di propagazione.

$$T \geq t_{c-q} + t_{logic} + t_{su}$$

$$t_{hold} \leq t_{cdlogic} + t_{cdreg}$$

Figure 90 - Le due regole.

Regola generale quando un circuito non funziona

Quando si ha un circuito che ad una data frequenza di clock funziona male, molto probabilmente si ha una setup time violation e/o una hold time violation. Nel primo caso basta abbassare la frequenza di clock. Il secondo caso è quello più complicato, si può risolvere utilizzando il clock skew a nostro favore oppure rallentando la logica.

## Il segnale di clock

Supponiamo che il clock arrivi allo stesso istante presso R1 e R2.

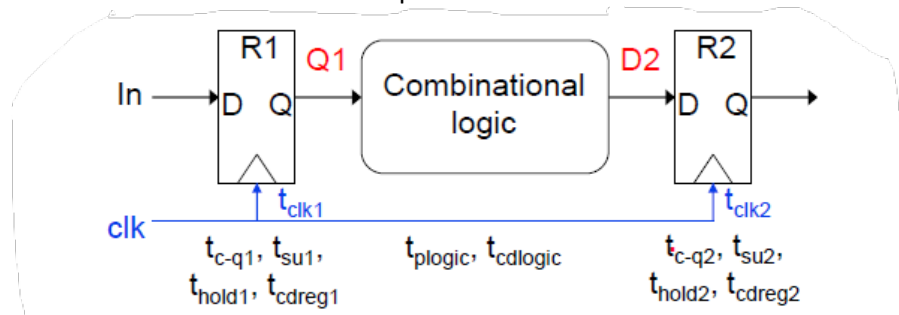


Figure 91- Rete di esempio

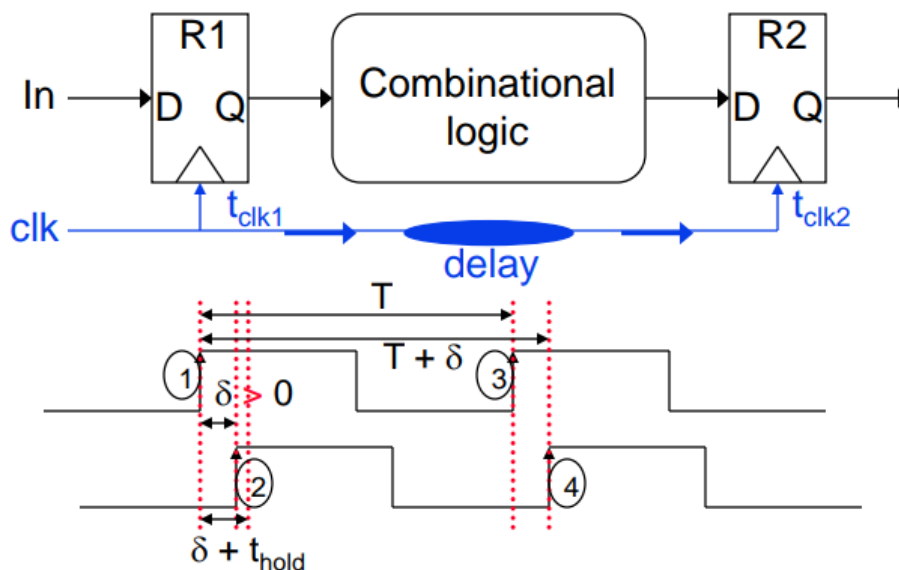
Se vogliamo analizzare queste equazioni, dobbiamo considerare lo scostamento (Skew) di clock e il jitter del clock.

### Clock skew

Parametro statico dovuto alle lunghezze diverse dell'interconnessione (*dei fili*), carichi diversi e da agenti esterni come la temperatura o l'alimentazione.

Questo fenomeno non sempre è negativo, a volte può giocare a nostro favore per far rispettare i requisiti di timing.

#### Positive clock skew



$$T: \quad T + \delta \geq t_{c-q} + t_{plogic} + t_{su} \quad \text{so} \quad T \geq t_{c-q} + t_{plogic} + t_{su} - \delta$$

$$t_{hold}: \quad t_{hold} + \delta \leq t_{cdlogic} + t_{cdreg} \quad \text{so} \quad t_{hold} \leq t_{cdlogic} + t_{cdreg} - \delta$$

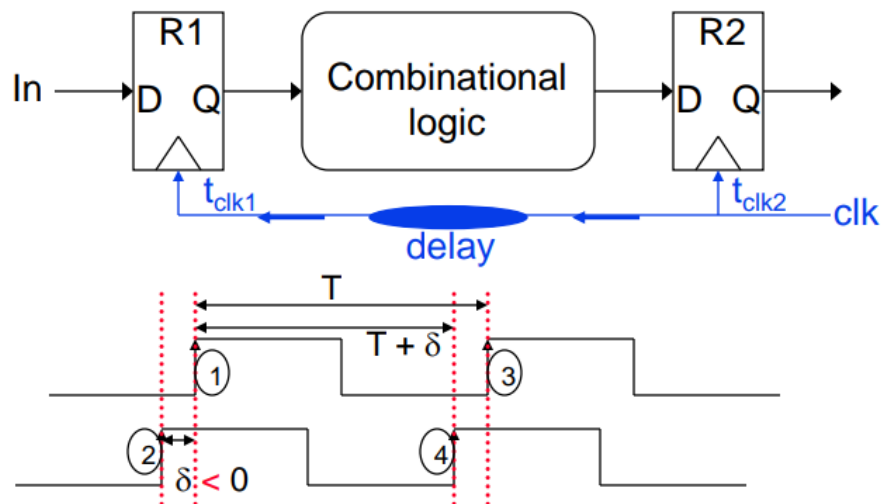
Figure 92- Effetti positivi del clock skew sulla 1° regola.

Come si può vedere il fronte di salita del clock arrivano sempre un poco dopo a R2.

Ciò ci permette di avere un periodo di clock più basso perché R1 ha più tempo per stabilizzare gli ingressi (è come se  $T_{setup2}$  fosse aumentato).

Ma questo fenomeno rende la seconda regola più difficile da rispettare, perché dal punto di vista di R2 gli input rimangono stabili per meno tempo; quindi, è come se la rete che produce il suo ingresso abbia aumentato la velocità.

### Negative clock skew



$$T: \quad T + \delta \geq t_{c-q} + t_{plogic} + t_{su} \quad \text{so} \quad T \geq t_{c-q} + t_{plogic} + t_{su} - \delta$$

$$t_{hold}: \quad t_{hold} + \delta \leq t_{cdlogic} + t_{cdreg} \quad \text{so} \quad t_{hold} \leq t_{cdlogic} + t_{cdreg} - \delta$$

Figure 93 – Effetti positivi del clock skew a favore della 2° regola.

Come si può vedere le variazioni clock arrivano sempre un poco dopo a R1. Ciò ci obbliga di avere un periodo di clock più alto e quindi una frequenza più bassa. Ma questo fenomeno rende la seconda regola più facile da rispettare.

### Come posso produrre artificialmente il clock skew?

I modi sono i seguenti:

- Fare in modo che il collegamento che porta il clock ai due registri sia molto lungo, in questo modo il segnale di clock ci metterà più tempo per arrivare a R2.
- Inserire dei buffer per rallentare il clock.

### Clock Jitter

Il jitter è dovuto al modo in cui generiamo internamente il clock, ossia attraverso il PLL (*anello di blocco di fase*). Il modo in cui lo generiamo non è perfetto e ad ogni ciclo di clock potresti avere una piccola variazione che (*fronte di salita un poco prima di quando dovrebbe essere e fronte di discesa un poco dopo*) che rende la durata effettiva del periodo di clock leggermente maggiore. Ciò gioca a nostro sfavore ovviamente.

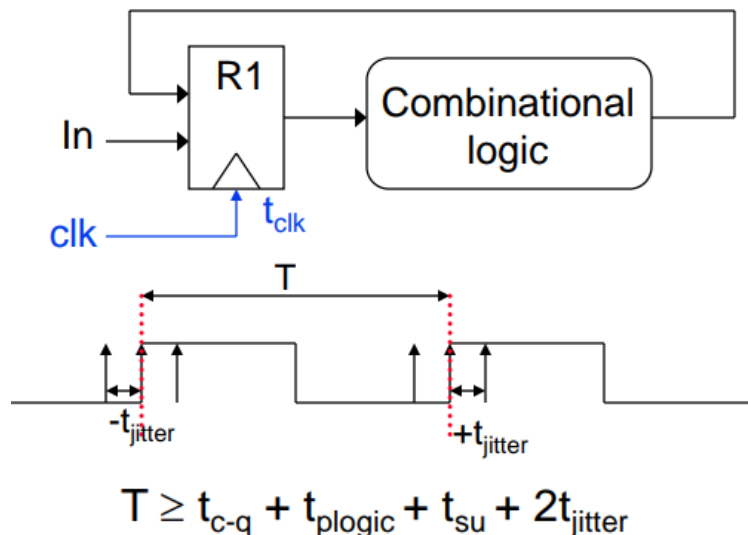


Figure 94- Effetti negativi del clock jitter sulla 1° regola.

## Clock skew e clock jitter combinati

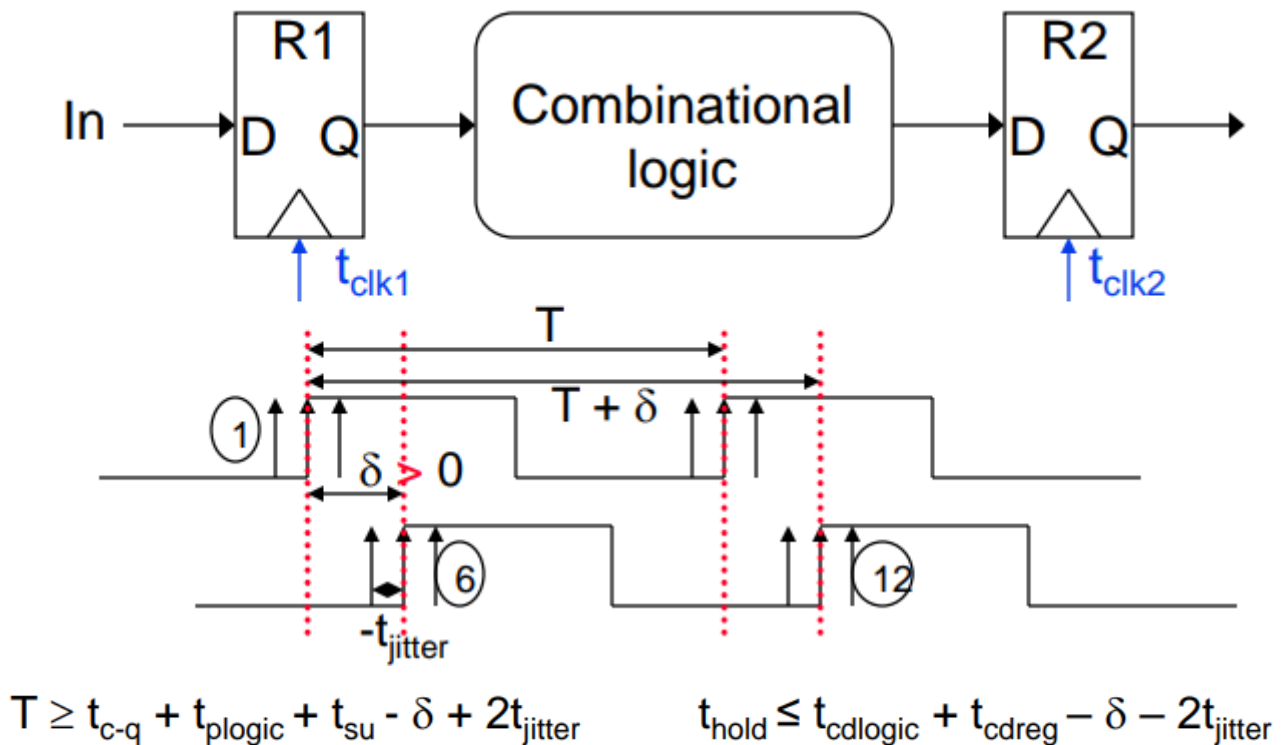


Figure 95 - effetti dello skew e del jitter insieme su entrambe le regole.

Il clock skew può rendere più facile da rispettare solo una delle due regole, mentre il jitter è un fenomeno puramente negativo che abbassa le prestazioni.

### Generatore di Clock

Nota sulla Zibo board

La scheda Zibo ha un FPGA e un clock di ingresso (*non ad alta frequenza, tipicamente 125 MHz*).

Come funziona il PLL? È un loop dove abbiamo il segnale di ingresso che è di 125 MHz, molto stabile, e poi se vogliamo ottenere ad esempio un clock di 400 MHz partendo da 100 MHz, usiamo un VCO (*oscillatore controllato in tensione*), dividiamo 400 MHz per 4, otteniamo un clock diviso. Possiamo avere due situazioni per il clock: se il clock è in ritardo dobbiamo aumentare la tensione (accelerare), se il clock è in anticipo dobbiamo diminuire la tensione. Questo cambiamento nella frequenza del clock generatore è chiamato jitter, che è un parametro statistico. Il clock q è stabile. Lo strumento si occupa di questo, non devi pensarci mentre progetti l'hardware.

### PPL (Anello di Blocco di Fase)

Un PLL è un circuito elettronico che genera un segnale di uscita con una frequenza e una fase controllate rispetto a un segnale di riferimento di ingresso. È comunemente utilizzato per generare segnali di clock stabili e precisi per vari scopi nelle applicazioni digitali e analogiche.

## Distribuire il segnale di clock

### Balanced paths

Questa tecnica mira a garantire che **tutti i percorsi del segnale di clock abbiano lunghezze simili**, riducendo al minimo i ritardi e le discrepanze temporali tra i vari componenti.

1. I percorsi che il segnale di clock deve percorrere vengono suddivisi in segmenti più brevi e bilanciati in modo che la lunghezza di ciascun segmento sia il più possibile simile.
  - a. Questa cosa può essere ottenuta tramite la disposizione strategica dei componenti e dei collegamenti sulla scheda o nel circuito.
2. Stimare i ritardi associati a ciascun segmento.
  - a. Si tiene conto delle caratteristiche del materiale, delle lunghezze e della velocità di propagazione dei segnali.
3. Se necessario, vengono implementate tecniche di compensazione dei ritardi per garantire che tutti i segmenti del percorso del clock siano allineati temporalmente.
  - a. Come l'aggiunta di linee di ritardo o di buffer in punti strategici lungo il segmento.
4. Verifica tramite simulazione del perfetto bilanciamento dei percorsi.

### Clock grids

Dividiamo il nostro chip in diverse aree locali, ognuna delle quali ha il suo clock locale.

Ogni area è collegata con altre aree attraverso una comunicazione asincrona.

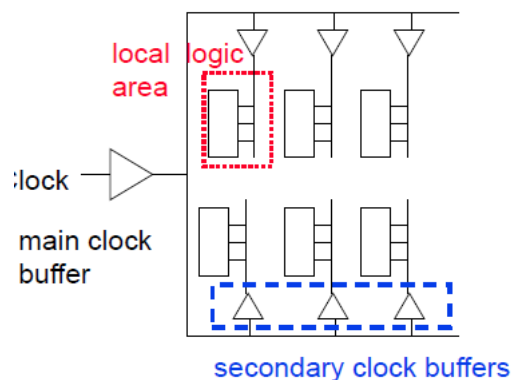


Figure 96 – Diagramma di una clock grid.

Questo approccio serve a **ridurre lo sfasamento del clock**. Il nuovo problema è quello di gestire la comunicazione tra le aree, che funziona utilizzando diverse frequenze di clock.

### H-Tree Clock Network

Strategia utilizzata per distribuire in modo efficiente il segnale di clock in un sistema digitale, specialmente in sistemi integrati complessi come processori, FPGA o ASIC.

Questa tecnica prende il nome dalla sua struttura a forma di "H", che è simile alla forma di un albero invertito.

#### Come viene progettata la rete

La rete del clock viene progettata come una struttura ad albero forma di "H", con un segmento verticale centrale (tronco) e diversi rami orizzontali che si estendono da entrambi i lati.

Questa configurazione permette di distribuire il segnale di clock in modo uniforme e simmetrico a tutti i componenti del sistema.

#### Segmento verticale

Il segmento verticale dell'H-Tree rappresenta il percorso principale del segnale di clock. È generalmente progettato con un'ampiezza maggiore rispetto ai rami laterali per ridurre i ritardi e migliorare l'integrità del segnale.

#### Rami orizzontali

I rami orizzontali dell'H-Tree si estendono da entrambi i lati del tronco centrale e si suddividono ulteriormente in segmenti più piccoli man mano che si ramificano verso l'esterno. Questi rami distribuiscono il segnale di clock a livelli più bassi di gerarchia, raggiungendo eventualmente tutti i componenti del sistema.

#### Segmentazione e ripetizione

Lungo il percorso dell'H-Tree, vengono utilizzati buffer di segnale o elementi di ritardo per compensare i ritardi e garantire che il segnale di clock venga distribuito uniformemente e senza perdite di integrità.

#### Simmetria e bilanciamento

L'H-Tree viene progettato per garantire simmetria e bilanciamento nella distribuzione del clock. Tutti i rami dell'albero hanno lunghezze simili e ritardi minimi, e se i percorsi sono perfettamente bilanciati, il clock skew diventa trascurabile.

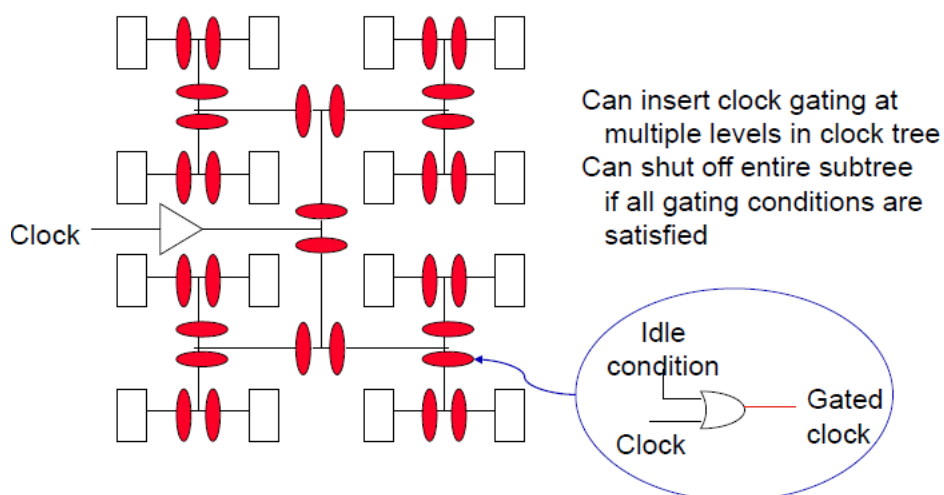


Figure 97 - Rete progettata con questa tecnica.