



UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI  
INGEGNERIA INFORMATICA,  
MODELLISTICA, ELETTRONICA  
E SISTEMISTICA

DIMES

## **Report Tecnico 1**

# **Ricavare curva energy-delay ottima per una catena di inverter**

*Corso di Progettazione Low Power*

*Corso di Laurea Magistrale in Ingegneria Elettronica*

*Unical, aa 2020/2021*

Andrea Alecce Matricola 214611

Prof. F. Frustaci

# Sommario

- 1. Design e analisi inverter ..... 3
  - 1.1. Misura del ritardo ed energia dissipata da un inverter ..... 5
  - 1.2. Dimensionamento e calcolo dei ritardi ..... 6
  - 1.3. Modello del ritardo ..... 8
  - 1.4. Modello dell'energia ..... 10
    - 1.4.1. Calcolo capacità intrinseca ..... 10
    - 1.4.2. Calcolo  $\gamma e$  ..... 12
- 2. Curva energy-delay ottimale per un buffer a tre stadi..... 13
  - 2.1. Modello di ritardo e energia ..... 15
  - 2.2. Ottimizzazione in Matlab ..... 16

## 1. Design e analisi inverter

Inizialmente viene simulato un singolo inverter, ricavando il dimensionamento minimo dello stesso: in questa fase usiamo una tecnologia basata su modello non realmente esistente.

Oltre alla misura del delay/ritardo dell'inverter, si analizza anche il concetto di inverter minimo. Si è interessati alla curva energy/delay ottima; il fatto di poter modellizzare analiticamente (espressione matematica) il sistema risulta di interesse non solo per avere una soluzione in maniera veloce (abbiamo i simulatori per questo in maniera più accurata). Ma cercare una relazione (ad esempio la proporzionalità tra delay e capacità oppure energia e  $V_{DD}$ ) permette di ricavare un andamento veritiero del delay in funzione della  $C_L$  o dell'energia in funzione della  $V_{DD}$  e così via. Anche se questi modelli non sono assolutamente accurati, lo è la loro dipendenza (lineare, esponenziale etc...). Ciò permette di ricavare una espressione analitica della curva, cosa che non viene fornita dai tool software.

Si effettua una simulazione su un singolo inverter, verificandone il funzionamento, misurando il ritardo e l'energia dissipata.

Il circuito è il seguente:

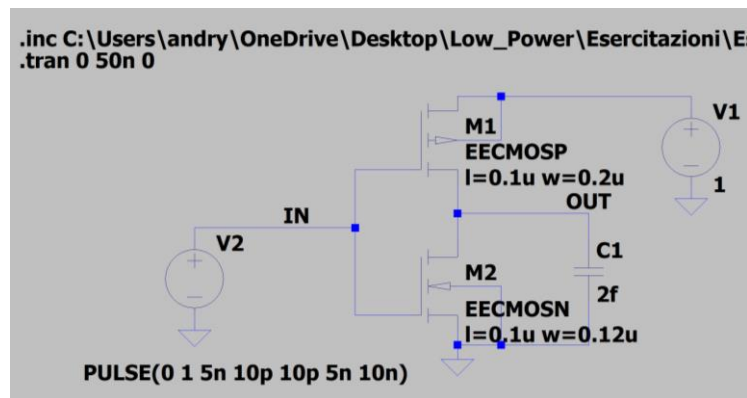


Figura 1 - Inverter in LTSpice

Dove sono stati rinominati i transistor N e P come:

- **EECMOSN**
- **EECMOSP**

Questo perché si intende usare il modello presente nel file **RIT\_Models\_For\_LTSPICE.txt**, bisogna usare l'esatto riferimento dato dal parametro **.model**. Il file fornisce alcune info sul modello: ad esempio, si tratta di una tecnologia a **100nm**, quindi la dimensione minima per il canale è 100nm. Inoltre, indica altri parametri importanti con i relativi valori settati, tra cui:

- **TOX=5E-9**
- **VTH0=0.4**
- **LEVEL=8**

L'ultimo parametro sta a indicare il livello di dettaglio di quello specifico modello. Un livello 8 è poco dettagliato, si arrivano ad avere modelli con livello 50.

Si inserisce l'alimentazione e il segnale in ingresso ad onda quadra. Infine, si identificano le net di ingresso e uscita, per visualizzarle facilmente in fase di simulazione.

Infine, dobbiamo indicare dove recuperare il modello del mosfet e che tipo di simulazione avviare. Per fare ciò, definiamo la direttiva **.inc** seguito dal path in cui è salvato il file (senza spazi).

Il tipo di simulazione è nel dominio del tempo (simulazione in transitorio), useremo la direttiva **.tran**. La durata complessiva è di 50ns.

Si avvia la simulazione e si mostrano i segnali di interesse, ovvero  $V_{IN}$ ,  $V_{OUT}$  e  $I(V_1)$  la corrente in uscita dall'alimentazione.

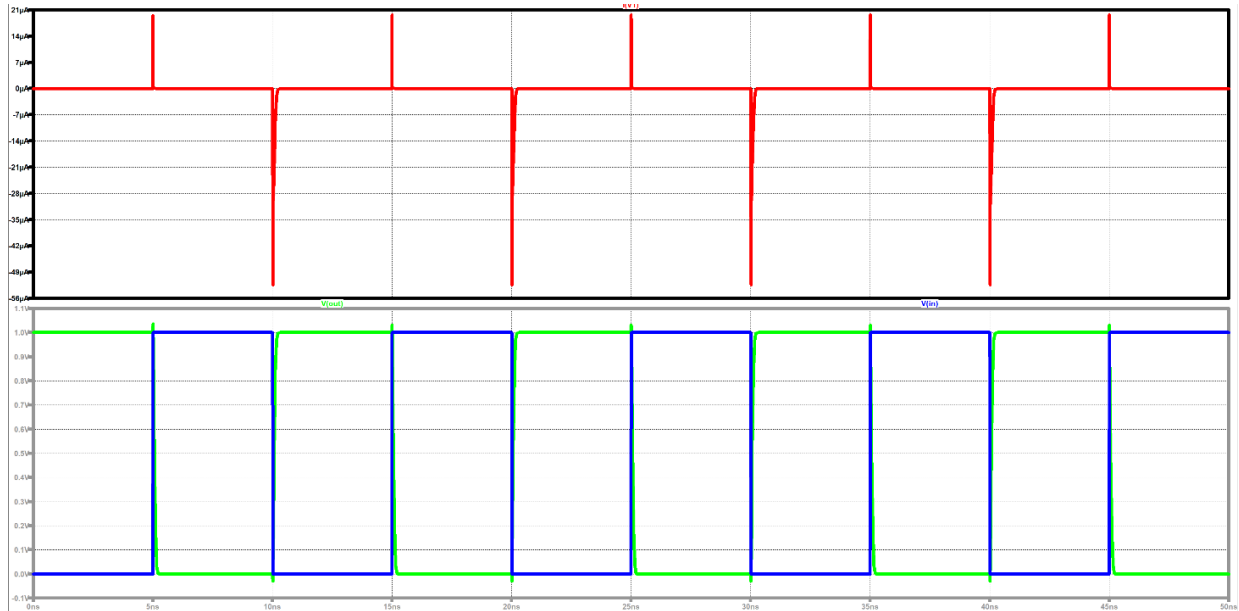


Figura 2 - Grafici ricavati in LTSpice. In alto: corrente di alimentazione; in basso: segnali di ingresso e uscita

Con uno zoom sulla transizione:

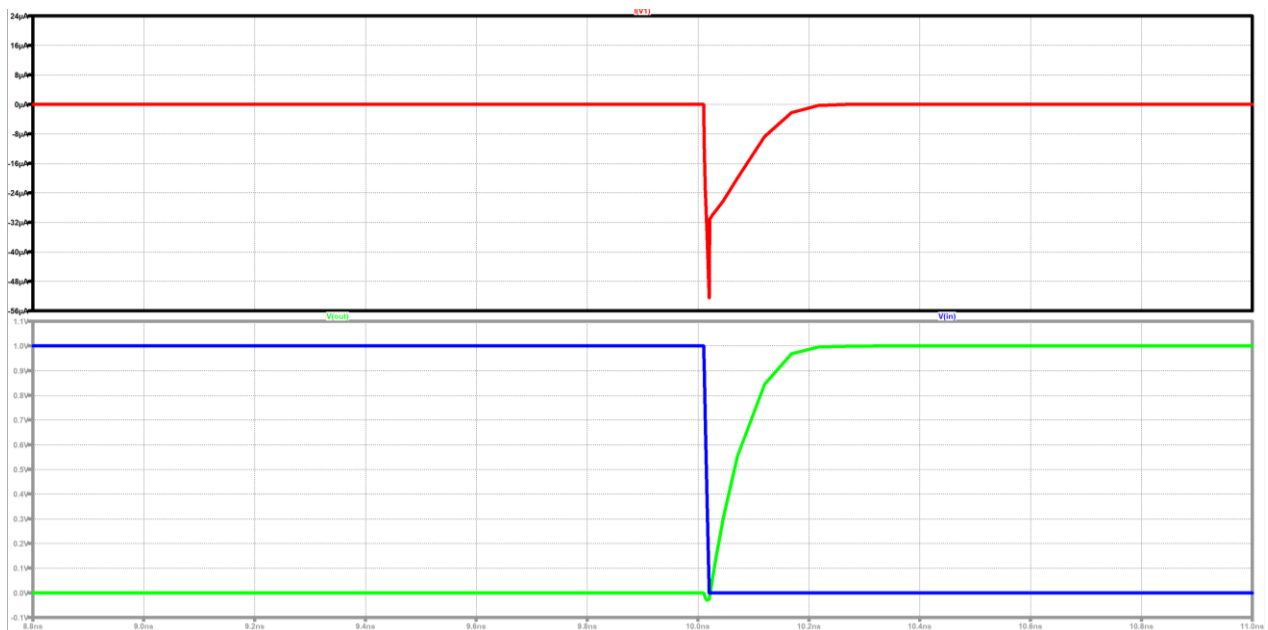


Figura 3 - Dettaglio dei grafici ricavati in LTSpice. In alto: corrente di alimentazione; in basso: segnali di ingresso e uscita

Come atteso dal comportamento dell'inverter notiamo che l'uscita è esattamente il negato dell'ingresso. Anche l'andamento della corrente è regolare.

Ci si chiede dov'è che l'inverter dissipa energia. La risposta è durante la fase di carica della capacità, ovvero durante la transizione 0-1 dell'uscita (a 10ns, 20ns etc...). Quando la capacità si carica, il

grafico della corrente evidenzia un picco negativo, rispettando la convenzione secondo cui una corrente uscente da un generatore ha segno negativo.

Nelle transizioni opposte si nota un picco positivo, quindi una corrente entrante. L'unico path che fisicamente possa far scorrere una corrente entrante nel generatore è quella dall'ingresso. Quando c'è una transizione (variazione molto veloce) del segnale in ingresso, la capacità Gate-Source ha una bassa impedenza (corto circuito) provocando un collegamento per la corrente.

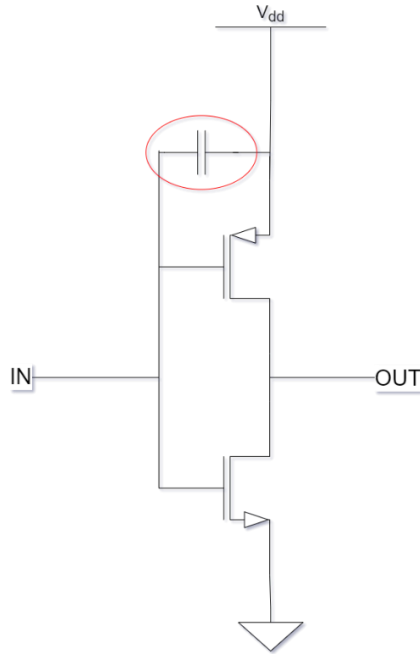


Figura 4 - Inverter CMOS. In evidenza, la capacità tra ingresso e alimentazione

La corrente fornita dall'ingresso può essere vista come la corrente in uscita da un'altra porta logica. Questa corrente sarà a sua volta fornita dalla relativa alimentazione.

Si conclude che entrambi i picchi (positivi e negativi) sono sempre potenza dissipata, in quanto provengono sempre dall'alimentazione (in comune).

### 1.1. Misura del ritardo ed energia dissipata da un inverter

Ci si chiede ora di misurare il ritardo e l'energia dissipata. È possibile farlo graficamente, tramite i cursori messi a disposizione da LTSpice. Ciò comporta però problemi di fattibilità (su 1000 simulazioni diventa impossibile) sia di precisione. Altro metodo è l'utilizzo di alcuni script. Nel caso in esame, si usa un semplice file *Schema1\_delay\_energy\_script.txt* al cui interno si trovano le seguenti direttive:

- **.measure tran ritardo\_rise trig v(In) val=0.5 fall=1 targ v(Out) val=0.5 rise=1**
- **.measure tran ritardo\_fall trig v(In) val=0.5 rise=1 targ v(Out) val=0.5 fall=1**
- **.measure tran energia INTEG -1\*(1\*i(V1)) from=7.5n to=12.5n**

Le prime due sono misure di ritardo, mentre l'ultima è una misura di energia.

Per il ritardo di salita, si ha:

- **.measure** che è una parola chiave ad indicare che si stanno effettuando delle misure;
- **Tran** è la tipologia di simulazione (transitorio, nel dominio del tempo);
- **Ritardo\_rise** o **ritardo\_fall** è il nome della simulazione

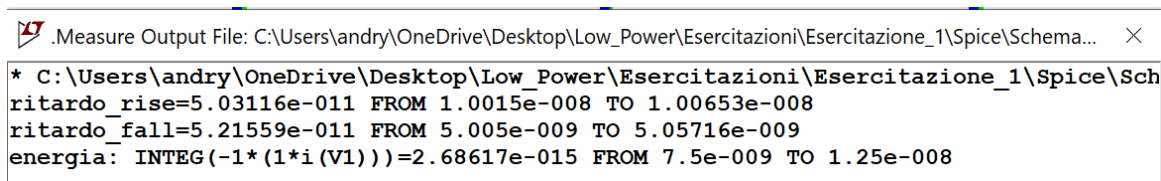
- **Trig** è l'istante di trigger, ovvero il momento in cui parte la misura. Nel caso in esame quando V(In) passa per il valore val=0.5 sul primo fronte di discesa (fall=1)
- **Targ** è l'istante di target, ovvero il momento in cui la misura deve terminare. Nel caso in esame quando V(Out) passa per il valore val=0.5 sul primo fronte di salita (rise=1)

Analogamente per il ritardo di discesa, con l'accortezza di invertire rise e fall.

Per la misura dell'energia, si indica l'integrale definito del prodotto tra l'alimentazione  $V_1$  (che è 1V) e la corrente  $I(V_1)$ . Gli estremi di integrazione da 7.5ns a 12.5ns si ricavano graficamente considerando un intervallo di tempo sufficientemente ampio da racchiudere il picco di corrente di interesse.

Per lanciare lo script, si torna su Spice, poi File – Execute .MEAS Script (in genere gli script hanno formato **.Meas** per essere riconosciuti, ma funziona anche come .txt) e si apre il file.

Vengono subito mostrati i risultati:



```

* C:\Users\andry\OneDrive\Desktop\Low_Power\Esercitazioni\Esercitazione_1\Spice\Schema...
ritardo_rise=5.03116e-011 FROM 1.0015e-008 TO 1.00653e-008
ritardo_fall=5.21559e-011 FROM 5.005e-009 TO 5.05716e-009
energia: INTEG(-1*(1*i(V1)))=2.68617e-015 FROM 7.5e-009 TO 1.25e-008
  
```

Figura 5 - Risultato dello script per la misura dei ritardi e dell'energia dissipata

Si ottiene:

**ritardo\_rise=5.03116e-011 FROM 1.0015e-008 TO 1.00653e-008**

**ritardo\_fall=5.21559e-011 FROM 5.005e-009 TO 5.05716e-009**

**energia: INTEG(-1\*(1\*i(V1)))=2.68617e-015 FROM 7.5e-009 TO 1.25e-008**

## 1.2. Dimensionamento e calcolo dei ritardi

Risulta interessante ricavare il valore di  $W_p$  (PMOS) tale da garantire gli stessi tempi di salita e discesa. A tal fine, si sfruttano le simulazioni parametriche, ovvero un modo per velocizzare le misurazioni quando esse differiscono tra loro solo per il valore di un unico parametro, definito a priori il circuito. Si è interessati a cambiare la  $W$  del PMOS, lasciando la  $W_N$  del NMOS al valore minimo e ricavando il rapporto d'aspetto ottimo che garantisca uguale delay sia in falling che rising.

Per parametrizzare il valore, lo si racchiude tra parentesi graffe indicandone il nome: ad esempio **{wp}**.

Vanno poi aggiunti gli statements relativi al modello di transistor usato e alla direttiva sulla simulazione, come visto precedentemente. Inoltre, si aggiunge una direttiva per andare a modificare il valore di wp parametrico, ovvero:

**.step param wp 0.12u 0.36u 0.01u**

Dove si indica il nome del parametro, il valore iniziale, quello finale e il passo: le simulazioni saranno circa 24.

Avviando la simulazione e considerando i grafici relativi alla  $V_{in}$  e  $V_{out}$ , si può notare la presenza di più grafici sovrapposti, come in figura:

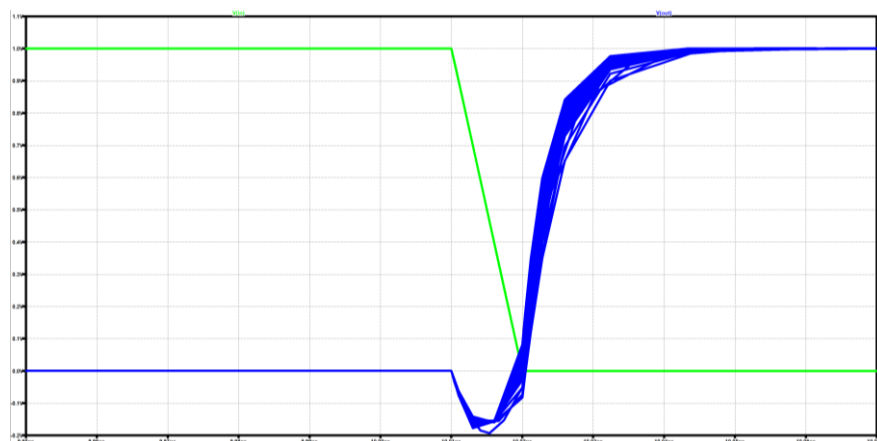


Figura 6 - Grafici ricavati in LTSpice. Dettaglio dei segnali di ingresso e uscita

Ogni simulazione è caratterizzata da una Wp differente. Il segnale in ingresso rimane identico in quanto fornito da un generatore d'onda ideale, non soggetto a variazioni rispetto a wp.

Bisogna ora confrontare i valori di ritardo di salita e discesa in modo da ricavare la wp tale da fornire un ritardo uguale. A tal fine si utilizza lo stesso algoritmo precedentemente implementato per i delay.

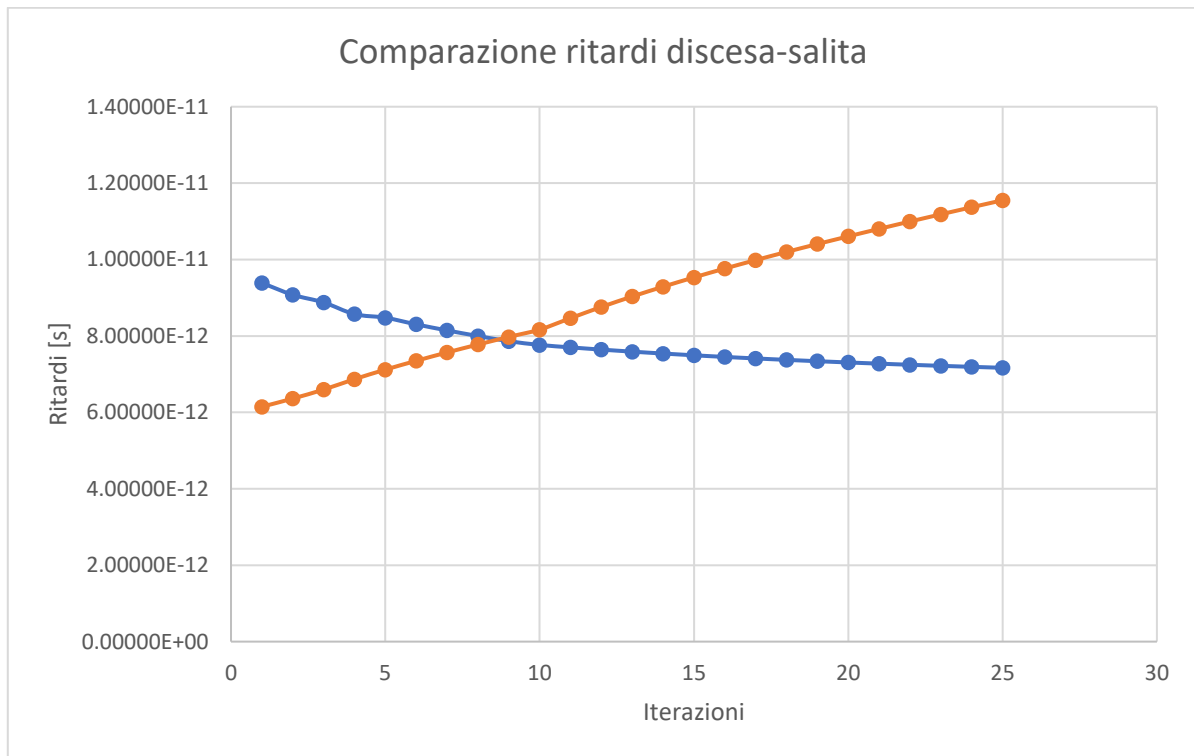
Il risultato è:

Measurement: ritardo_rise			
step	ritardo_rise	FROM	TO
1	9.38725e-012	1.0015e-008	1.00244e-008
2	9.07744e-012	1.0015e-008	1.00241e-008
3	8.878e-012	1.0015e-008	1.00239e-008
4	8.57165e-012	1.0015e-008	1.00236e-008
5	8.47718e-012	1.0015e-008	1.00235e-008
6	8.30383e-012	1.0015e-008	1.00233e-008
7	8.14506e-012	1.0015e-008	1.00231e-008
8	7.99875e-012	1.0015e-008	1.0023e-008
9	7.86338e-012	1.0015e-008	1.00229e-008
10	7.76529e-012	1.0015e-008	1.00228e-008
11	7.70133e-012	1.0015e-008	1.00227e-008
12	7.64287e-012	1.0015e-008	1.00226e-008
13	7.58917e-012	1.0015e-008	1.00226e-008
14	7.53963e-012	1.0015e-008	1.00225e-008
15	7.49374e-012	1.0015e-008	1.00225e-008
16	7.45108e-012	1.0015e-008	1.00225e-008
17	7.41128e-012	1.0015e-008	1.00224e-008
18	7.37404e-012	1.0015e-008	1.00224e-008
19	7.33909e-012	1.0015e-008	1.00223e-008
20	7.3062e-012	1.0015e-008	1.00223e-008
21	7.27517e-012	1.0015e-008	1.00223e-008
22	7.24582e-012	1.0015e-008	1.00222e-008
23	7.21801e-012	1.0015e-008	1.00222e-008
24	7.19159e-012	1.0015e-008	1.00222e-008
25	7.16644e-012	1.0015e-008	1.00222e-008
Measurement: ritardo_fall			
step	ritardo_fall	FROM	TO
1	6.14695e-012	5.005e-009	5.01115e-009
2	6.36136e-012	5.005e-009	5.01136e-009
3	6.59585e-012	5.005e-009	5.0116e-009
4	6.86919e-012	5.005e-009	5.01187e-009
5	7.11814e-012	5.005e-009	5.01212e-009
6	7.34893e-012	5.005e-009	5.01235e-009
7	7.56811e-012	5.005e-009	5.01257e-009
8	7.77529e-012	5.005e-009	5.01278e-009
9	7.97475e-012	5.005e-009	5.01297e-009
10	8.16778e-012	5.005e-009	5.01317e-009
11	8.46494e-012	5.005e-009	5.01346e-009
12	8.75917e-012	5.005e-009	5.01376e-009
13	9.03179e-012	5.005e-009	5.01403e-009
14	9.28886e-012	5.005e-009	5.01429e-009
15	9.53227e-012	5.005e-009	5.01453e-009
16	9.76368e-012	5.005e-009	5.01476e-009
17	9.98528e-012	5.005e-009	5.01499e-009
18	1.01988e-011	5.005e-009	5.0152e-009
19	1.04055e-011	5.005e-009	5.01541e-009
20	1.06066e-011	5.005e-009	5.01561e-009
21	1.08029e-011	5.005e-009	5.0158e-009
22	1.09952e-011	5.005e-009	5.016e-009
23	1.11838e-011	5.005e-009	5.01618e-009
24	1.13692e-011	5.005e-009	5.01637e-009
25	1.15517e-011	5.005e-009	5.01655e-009

Figura 7 - Risultato delle misure dei ritardi

Si nota subito che il ritardo\_rise diminuisce all'aumentare della wp, fenomeno atteso in quanto il pmos è responsabile della carica del nodo di uscita: più è conduttivo più veloce è la carica. In maniera diametralmente opposta si nota sul ritardo\_fall: in questo caso perché aumenta la capacità intrinseca dell'inverter.

Per individuare il valore di wp, soprattutto in casi con un numero di simulazioni elevato, è utile usare tool come *Excel*. Plottando i due grafici relativi ai delay in salita e discesa sovrapposti, si trova un unico valore in cui si intersecano, ovvero 9.



Questo valore, moltiplicato per il passo scelto (che era 0,01u) e sommato al Wmin del NMOS fornisce il valore di wp del PMOS che garantisce stesso ritardo di salita e discesa: in questo caso è pari a 0,21u.

### 1.3. Modello del ritardo

Ricordando che il modello di ritardo è:

$$\tau = \tau_{nom} \left( 1 + \frac{1}{\gamma} \frac{C_{i+1}}{C_i} \right)$$

Dove  $\tau_{nom}$  è il ritardo di propagazione dell'inverter minimo a vuoto. Quindi, oltre ad aver ricavato la wp, dalle simulazioni fatte si è ricavato anche  $\tau_{nom}$ , ovvero il valore di ritardo ottenuto dall'intersezione delle due curve. Nel caso in esame, circa 8 ps.

Ritornando al modello di ritardo, bisogna ora trovare dalla simulazione il parametro  $\gamma$  (che, teoricamente, si può ricavare dalla tecnologia utilizzata). Altro parametro che è possibile ricavare dalla simulazione è  $\tau$ . Delle due capacità rimaste come incognite, si potrebbe fissare una e ricavare l'altra.

Ad esempio, dal circuito finora analizzato si può porre in uscita una capacità nota come carico (es. 1fF) e dalla simulazione si ricava il ritardo  $\tau$ .



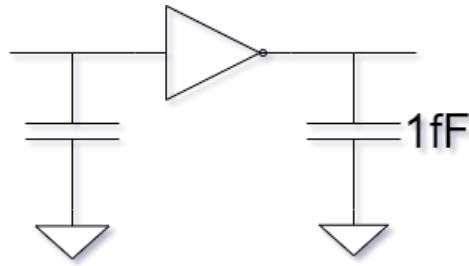


Figura 8 - Inverter e relative capacità

Manca ricavare il rapporto  $\frac{C_{i+1}}{C_i}$  che equivale al rapporto tra la capacità di carico e quella di ingresso. Ma con questo approccio non abbiamo conoscenza del valore della capacità di ingresso, dunque è da escludere.

Una soluzione è quella di utilizzare un secondo inverter come carico del primo. Il primo avrà dimensione minima mentre il secondo avrà dimensione, ad esempio, pari a 5 volte quello minimo: in questo modo è facile ricavare che il rapporto  $\frac{C_{i+1}}{C_i}$  sarà pari a 5.

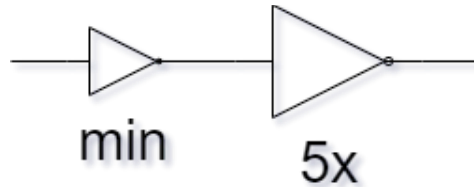


Figura 9 - Inverter minimo e di carico

Ovviamente al variare del rapporto  $\frac{C_{i+1}}{C_i}$  varierà anche la  $\gamma$  (anche se di poco), perché questo modello non è preciso. Una idea sarebbe quella di effettuare più simulazioni al variare della dimensione dell'inverter di carico e fare una media dei valori di  $\gamma$  ricavati.

Si realizza il nuovo circuito:

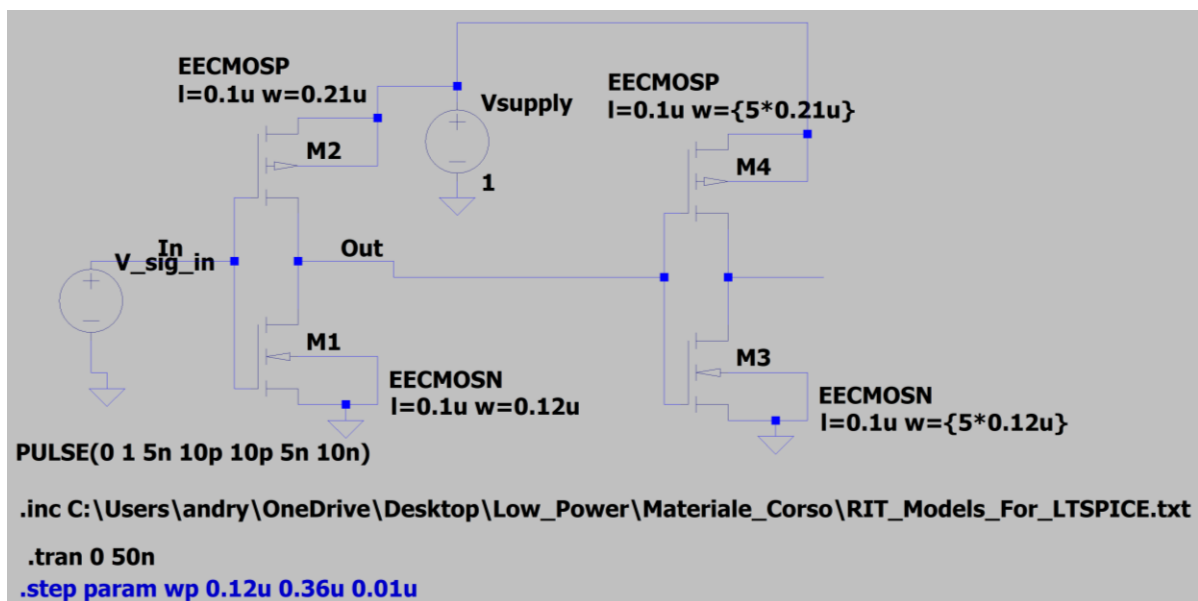
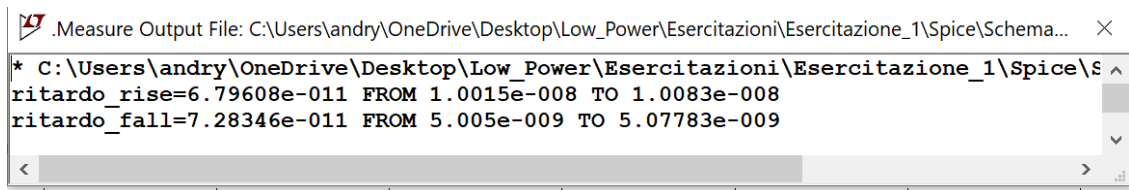


Figura 10 - Realizzazione in LTSpice di un inverter minimo e di carico

E si effettua la simulazione. In teoria ci si aspetta che il ritardo di rise e fall siano uguali, a meno di piccole incertezze. Si effettua il processo iterativo tramite lo stesso algoritmo di misura di prima, ottenendo i seguenti ritardi:



```

* C:\Users\andry\OneDrive\Desktop\Low_Power\Esercitazioni\Esercitazione_1\Spice\S
ritardo_rise=6.79608e-011 FROM 1.0015e-008 TO 1.0083e-008
ritardo_fall=7.28346e-011 FROM 5.005e-009 TO 5.07783e-009

```

Figura 11 - Risultato dello script per la misura dei ritardi

Considerando il ritardo più grande (fall) come valore di  $\tau$ . La formula del modello del ritardo diventa:

$$73p = 8p \left( 1 + \frac{1}{\gamma} * 5 \right)$$

Da cui:

$$\gamma_d = 0,615$$

#### 1.4. Modello dell'energia

Discorso analogo può essere fatto per il modello dell'energia: il buffer i-esimo contribuisce al consumo di energia totale per un valore pari a:

$$E_{dyn} = (\gamma C_i + C_w + C_{i+1}) V_{DD}^2$$

Dove si trascura  $C_w$  (capacità parassita della linea), la  $\gamma$  farà riferimento all'energia. Le uniche incognite sono  $C_i$  (capacità intrinseca dell'inverter minimo) e  $\gamma$ .

Per ottenere  $\gamma$ , bisogna ricavare la capacità intrinseca dell'inverter minimo che, divisa per quella di ingresso ci fornisce proprio  $\gamma$ .

Il problema si riduce quindi nell'ottenere il valore di questi due parametri, ricavabili da una misura dell'energia.

##### 1.4.1. Calcolo capacità intrinseca

Considerando l'inverter minimo, ci si interessa della capacità di ingresso e quella intrinseca, si effettua una simulazione di carica con una capacità di carico nota, ricavando una energia che teoricamente sarà pari a:

$$E = C V_{DD}^2 = (C_{int} + C_{load}) V_{DD}^2$$

Da cui si può ricavare  $C_{int}$ .

In tutto ciò si è trascurato l'effetto della corrente di corto circuito, che può essere modellizzata come contributi in termini di capacità. Per ovviare a ciò, si effettuano una serie di simulazioni con  $C_{load}$  differenti e se ne ricava una media.

Di seguito, il circuito simulato e i grafici ottenuti:

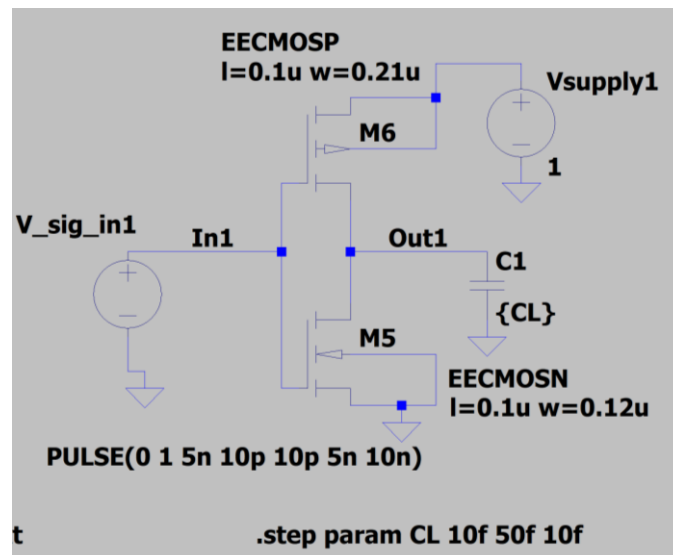


Figura 12 – Realizzazione in LTSpice di un Inverter con Cload parametrico

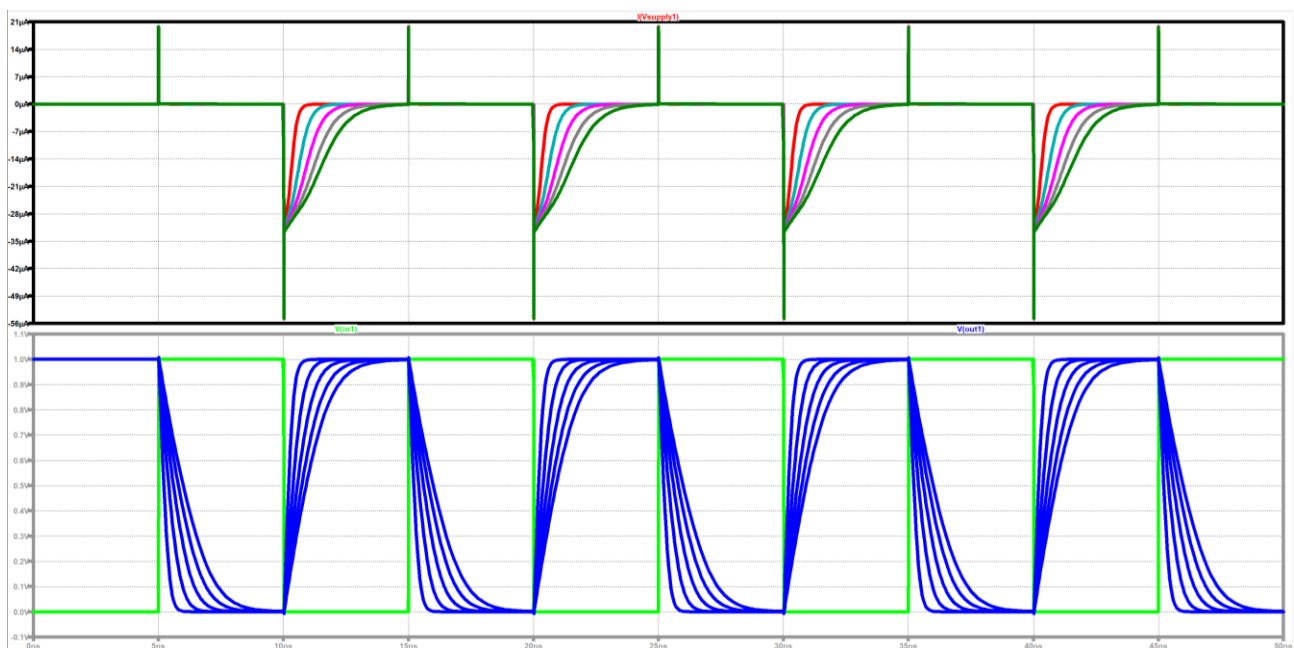


Figura 13 - Grafici ricavati in LTSpice di un Inverter con Cload parametrico. In alto: corrente di alimentazione; in basso: segnali di ingresso e uscita

Si avvia nuovamente lo script per la misura dell'energia. Dalla formula inversa del modello dell'energia si ricava l'espressione per la C<sub>int</sub> di ciascuno step. Infine, si effettua una media dei valori. Il tutto riportato in tabella:

step	energia	Cint	Cint media
1	1.06E-14	6.04E-16	5.38E-16
2	2.06E-14	5.88E-16	
3	3.06E-14	5.79E-16	
4	4.06E-14	5.53E-16	
5	5.04E-14	3.67E-16	

Per cui la  $C_{int}=0.53fF$ .

#### 1.4.2. Calcolo $\gamma_e$

Per ricavare la capacità di ingresso dell'inverter minimo, nota la capacità intrinseca appena ricavata, si può effettuare una ulteriore simulazione. Si realizza un circuito con un inverter minimo e un inverter di carico di dimensioni nota e pari, ad esempio, a 5 volte quello minimo.

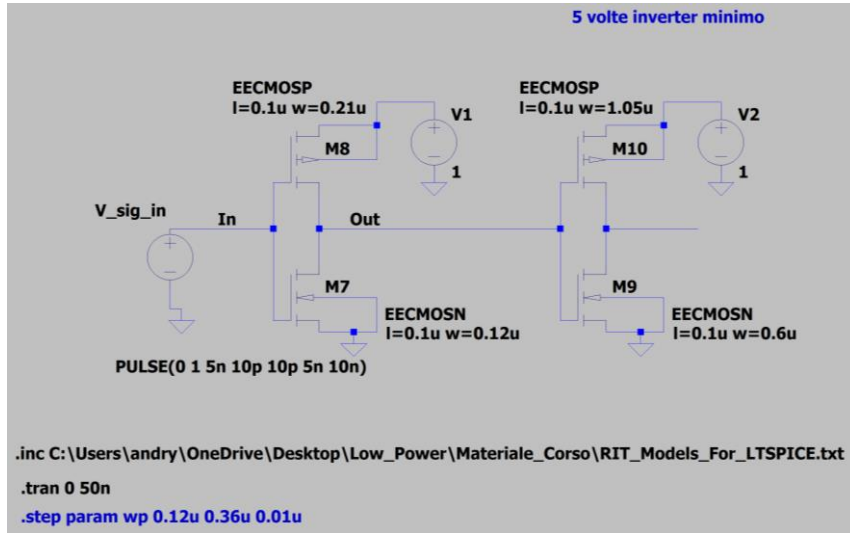


Figura 14 - Realizzazione in LTSPice di un Inverter con inverter di carico

L'energia necessaria a caricare l'uscita dell'inverter minimo sarà pari a:

$$E = (C_{int} + 5 * C_{ing,min}) V_{DD}^2 = 3.85 fJ$$

Dunque, conoscendo la E ricavabile da simulazione, la  $V_{DD}$  e la  $C_{int}$ , si ottiene la  $C_{ing,min}$ .

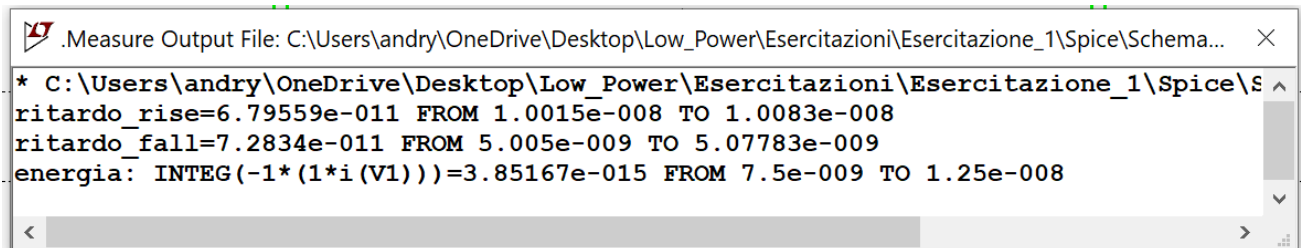


Figura 15 - Risultato dello script per la misura dei ritardi

$$C_{ing,min} = \frac{3.85 - 0.53}{5} = 0,66 fF$$

Infine, è possibile ricavare  $\gamma_e$  come

$$\gamma_e = \frac{C_{int}}{C_{ing,min}} = 0,8$$

Anche in questo caso è possibile iterare più volte con diversi valori, ad esempio, del dimensionamento dell'inverter di carico e ricavare una media dei valori di  $C_{ing,min}$  e  $\gamma_e$ .

## 2. Curva energy-delay ottimale per un buffer a tre stadi

Buffer a tre stadi con carico, di cui:

- il primo è un inverter minimo
- il secondo è dimensionato  $x1$  volte l'inverter minimo
- il terzo è dimensionato  $x2$  volte l'inverter minimo
- inverter di carico, dimensionato 50 volte l'inverter minimo

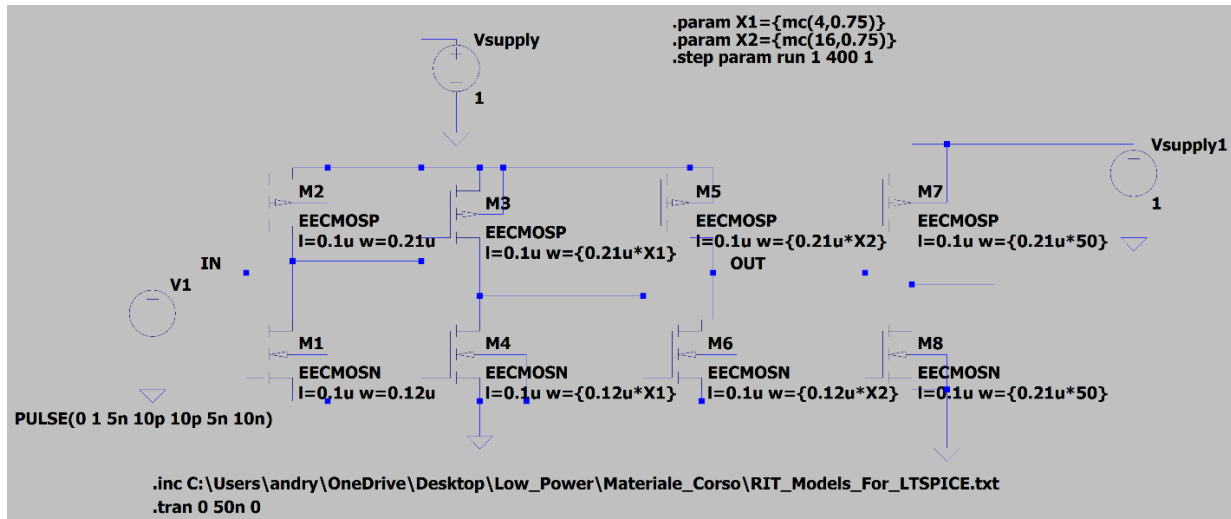


Figura 16 - Buffer a tre stadi con carico realizzato in LTSpice

Ovviamente l'alimentazione dell'inverter di carico è differente da quella del buffer, in modo tale da agevolare la misura dell'energia del solo buffer.

Il nostro obiettivo è trovare una serie di coppie energy-delay da graficare, sfruttando simulazioni montecarlo, ovvero una successione di simulazioni dove i valori di  $x1$  e  $x2$  sono scelti in maniera stocastica, ma con una distribuzione di probabilità ragionevole.

A tal proposito si utilizza la direttiva:

```
.param X1={mc(4,0.75)}
```

```
.param X2={mc(16,0.75)}
```

In LTSpice per indicare il parametro di interesse, con distribuzione uniforme intorno ad un valor medio pari a 4 e 16 rispettivamente con deviazione standard di 0.75. Il valore scelto per  $x1$  e  $x2$  deve rispettare unicamente il crescente valore in dimensione rispetto all'inverter minimo.

Infine, si indica il numero di simulazioni da fare tramite la direttiva:

```
.step param run 1 400 1
```

Avviate le simulazioni, si attende un certo intervallo di tempo affinché vengano eseguite tutte. Lo stato delle simulazioni viene notificato e aggiornato sullo schermo di LTSpice in basso a destra.

Di seguito, vengono mostrati i grafici relativi alla corrente di alimentazione:

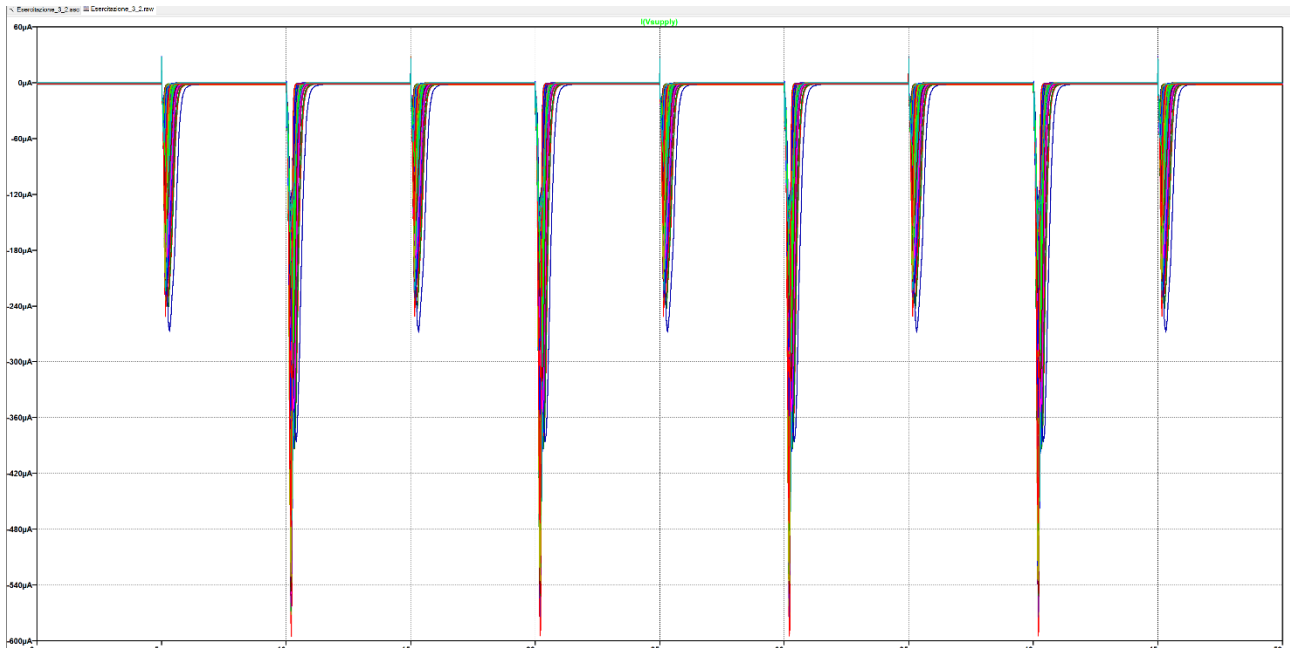


Figura 17 - Grafico ricavato in LTSpice della corrente di alimentazione di un buffer a tre stadi

Da cui bisogna ricavare i valori di energia e ritardo, ottenibili facilmente tramite script utilizzando le direttive:

**.measure tran energia\_bis INTEG 0.5\*1\*(abs(i(Vsupply))-i(Vsupply)) from=4n to=14n**

**.measure tran ritardo\_fall trig v(IN) val=0.5 rise=1 targ v(OUT) val=0.5 fall=1**

In linea teorica i ritardi di fall e rise sono uguali, per cui è presente la misura di solo uno dei due.

Da evidenziare il fatto che, rispetto al singolo inverter, la corrente assorbita ha picchi negativi, mentre prima era possibile osservare sia picchi negativi che positivi: ciò è dovuto proprio al fatto che ci si trova una catena di inverter, per cui mentre un inverter sta scaricando, altri due stanno caricando, e viceversa. Dunque, nel calcolo dell'energia, non è più consentito considerare un unico picco, bensì due.

Ecco perché l'intervallo di integrazione per la misura dell'energia è from=4n to=14n, in modo da prendere i due picchi di assorbimento.

Inoltre, nella direttiva è implementata una differente formula per il calcolo dell'energia. Dal grafico è possibile notare alcuni spike positivi, dovuti alle capacità parassite sugli ingressi. Integrando semplicemente, andrei a considerare anche questi picchi il cui contributo farebbero diminuire il valore di energia calcolata. Siccome questi picchi sono "fittizi" (nonostante siano piccoli e trascurabili) è opportuno eliminarli nel calcolo.

Per eliminarla, considero la forma d'onda iniziale f1 e il valore assoluto della stessa f2. Ne faccio la sottrazione, per cui i picchi entrambi positivi si eliminano, mentre quelli discordi si sommano. Da questo risultato divido per due per ottenere il segnale originario a meno del piccolo spike.

Avviato lo script, si ottengono i 400 valori di energia e ritardo, importabili in Excel. Da questi valori viene generato un grafico a punti, come segue:

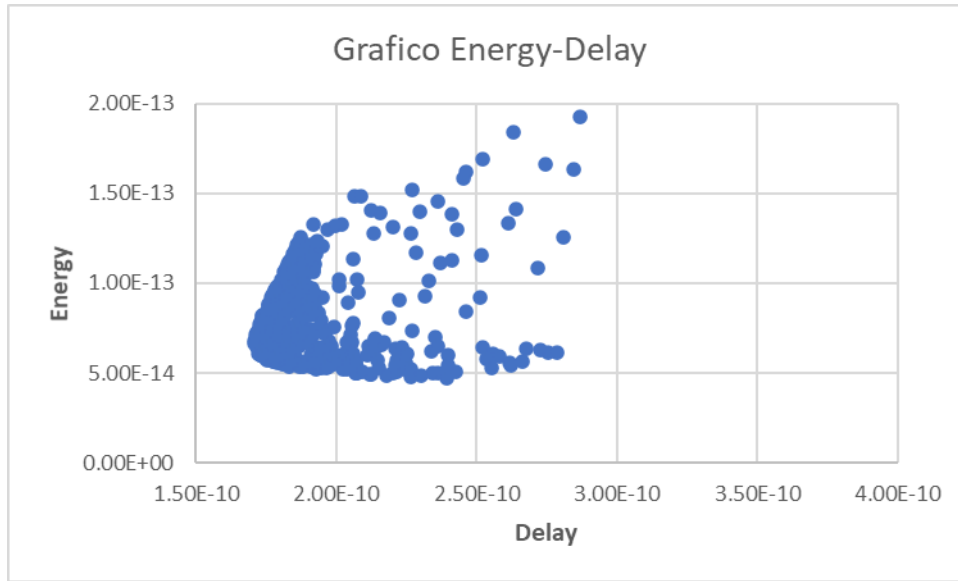


Figura 18 - Grafico dello spazio Energy-Delay

Da cui si può notare la curva, o comunque il luogo dei punti ottimo a cui sarei interessato durante la progettazione. Ad esempio, per un delay di 200ps ho tanti punti corrispondenti di energia, ma l'ottimo è situato sulla curva.

Il grafico ha dimostrato l'esistenza di una curva energy-delay ottima, ma non permette di ricavare informazioni consistenti sulla coppia di parametri di interesse. Per ricavare ulteriori informazioni, è possibile utilizzare i tool di Matlab.

Il problema che si vuole risolvere di minimizzazione interessa:

$$\begin{cases} \min E(x_1, x_2) \\ \tau(x_1, x_2) = D_0 \end{cases}$$

che risulta essere analiticamente complessa da risolvere.

## 2.1. Modello di ritardo e energia

Riprendendo il modello del ritardo, in generale il ritardo è dato dal contributo di ogni inverter, dunque:

$$\tau = \sum_i \tau_i$$

$$\tau = \tau_{nom} \left( 1 + \frac{1}{\gamma_d} x_1 \right) + \tau_{nom} \left( 1 + \frac{1}{\gamma_d} \frac{x_2}{x_1} \right) + \tau_{nom} \left( 1 + \frac{1}{\gamma_d} \frac{50}{x_2} \right)$$

Analogamente, l'energia complessiva è data dal contributo di ogni inverter, dunque:

$$E = \sum_i E_i$$

1. Il primo stadio è responsabile solo della sua capacità intrinseca.
2. Il secondo stadio è responsabile della sua capacità intrinseca più quella di ingresso.
3. Il terzo stadio è responsabile della sua capacità intrinseca, quella di ingresso e quella di carico (50x).

Per il primo inverter:

$$E_1 = V_{DD}^2 C_{intr} = V_{DD}^2 \gamma_e C_{in\ inv\ min} = V_{DD}^2 \gamma_e k$$

Dove:

$$C_{intr} = \gamma_e C_{in\ inv\ min} = \gamma_e k$$

Per il secondo inverter:

$$E_2 = V_{DD}^2 x_1 (1 + \gamma_e) k$$

Per il terzo inverter:

$$E_3 = V_{DD}^2 x_2 (1 + \gamma_e) k$$

Per il carico:

$$E_{Load} = V_{DD}^2 C_L = V_{DD}^2 50k$$

L'espressione complessiva sarà:

$$E = V_{DD}^2 \gamma_e k + V_{DD}^2 x_1 (1 + \gamma_e) k + V_{DD}^2 x_2 (1 + \gamma_e) k + V_{DD}^2 50k$$

Di cui si conoscono i valori di:

- $k = C_{ing,min} = 0,66fF$
- $\gamma_e = 0.8$
- $\tau_{nom} = 8ps$
- $\gamma_d = 0,615$
- $V_{DD} = 1$

## 2.2. Ottimizzazione in Matlab

Una volta ottenute le formule che descrivono l'energia e il ritardo del buffer sono state implementate in Matlab tramite i seguenti script che sfruttano dei tool di ottimizzazione. Teoricamente, si dovrebbe derivare le espressioni dell'energia in funzione di  $x_1$  e  $x_2$ , uguagliandole e sfruttando l'espressione del ritardo per ottenere la formulazione analitica. Sfruttando i tool tutto ciò avverrà implicitamente.

Per richiamare il tool, è necessario dare la direttiva ***optimtool***:



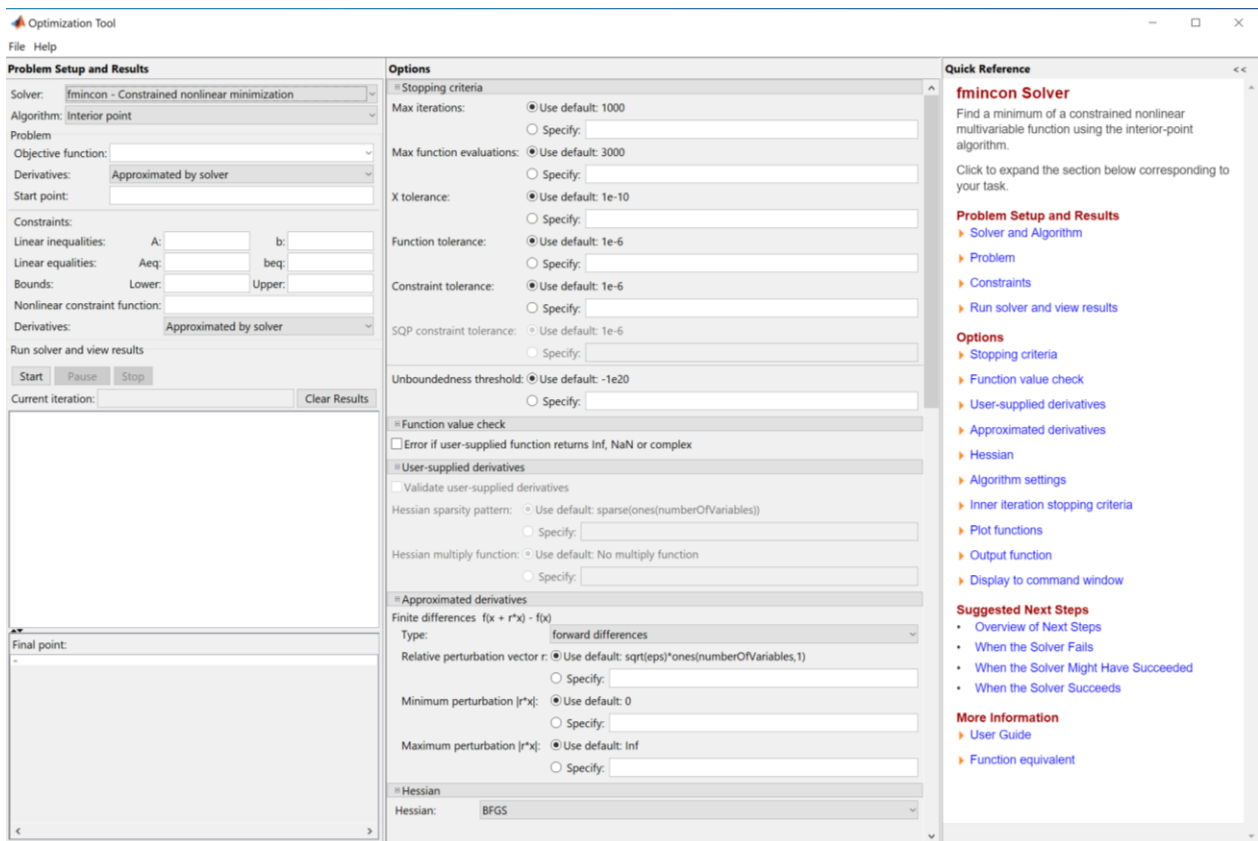


Figura 19 - Finestra in Matlab relativa al tool di ottimizzazione

Prima di utilizzare il tool, occorre creare due file nei quali vengono descritte le equazioni espresse precedentemente:

- **Energy.m**

```

1 function [res] = Energy(W)
2
3     GammaE=0.8;
4     Vdd=1;
5     K=0.66;
6     Cload=50;
7     res=K*Vdd*Vdd*(GammaE+(1+GammaE)*W(1)+(1+GammaE)*W(2)+Cload);
8     %l'energia risultante sarà espressa in fJ

```

Figura 20 - Funzione Energy.m in Matlab

Dove res è la variabile in uscita dalla funzione. W(1) e W(2) sono esattamente  $x_1$  e  $x_2$ , espressi come un array di due elementi in ingresso alla funzione.

- **DelayConstraint.m**

```

1 function[c, ceq] = DelayConstraint(W)
2
3 tau0=8;
4 GammaD=0.61;
5 Cload=50;
6 D0=210;%il ritardo lo esprimo in ps
7 ceq=tau0*(1+W(1)/GammaD + 1 +W(2)/(GammaD*W(1)) + 1 + Cload/(GammaD*W(2)))-D0;
8 % Vincoli di uguaglianza non lineari ( visti come ceq ==0)
9 %c = -W(1);
10 c = 1-W(1);

```

Figura 21 - Funzione DelayConstraint in Matlab

Dove compare, oltre la funzione ritardo, anche il vincolo sul ritardo. Nel caso in esame, si sceglie:

$$\tau(x_1, x_2) = D_0 = 210ps$$

Da cui

$$\tau - 210ps = 0$$

che è il metodo in cui Matlab accetta l'espressione.

Ulteriore vincolo è di tipo geometrico e vincola il tool a trovare soluzioni solo maggiori dell'inverter minimo, che ha valore 1. Il vincolo è quindi  $W(1) > 1$ . Siccome Matlab accetta le espressioni del tipo  $c \leq 0$ , si adatta il vincolo a questa regola, ottenendo  $c = 1 - W(1)$ .

Ultimati i file, sul tool bisogna fornire alcuni parametri. Inizialmente, è necessario specificare la funzione obiettivo, cioè la funzione che si vuole minimizzare. Per minimizzare l'energia, bisogna inserire: @Energy. Nella casella Nonlinear Constraint Function bisogna fornire il vincolo sul delay, dunque: @DelayConstraint. Inoltre, nella casella Start Point va inserito un valore iniziale: supponendo che il buffer deve caricare una capacità di 50x, i valori di  $x_1, x_2$  saranno compresi tra 1 e 50, fornisco un valore arbitrario di [10;10].

Nel tool, sulla destra, è possibile settare alcune opzioni relative alla tolleranza, e al numero di iterazioni. Alcune vengono lasciate di default, altre modificate come segue:

The screenshot shows the 'Options' dialog box with the following settings:

- Max iterations:** ☒ Specify: 10000
- Max function evaluations:** ☒ Use default: 3000
- X tolerance:** ☒ Specify: 0.01
- Function tolerance:** ☒ Specify: 0.01
- Constraint tolerance:** ☒ Use default: 1e-6
- SQP constraint tolerance:** ☒ Use default: 1e-6
- Unboundedness threshold:** ☒ Use default: -1e20

Figura 22 - Finestra in Matlab relativa alle opzioni del tool di ottimizzazione

Avviando il tool, il suo output è:

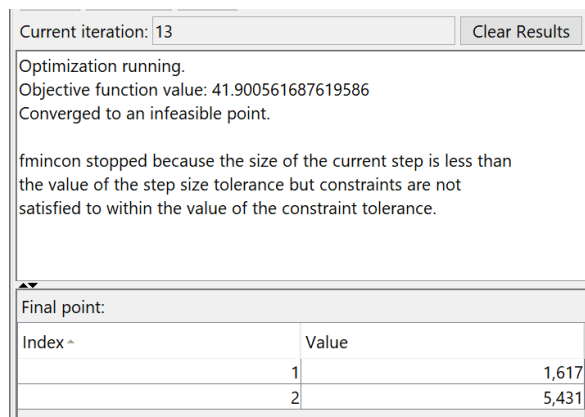


Figura 23 - Finestra in Matlab relativa al risultato dell'ottimizzazione

Dove i due valori  $x_1 = 1.617$  e  $x_2 = 5.431$  minimizzano l'energia per il ritardo pari a 210ps.

Si può ora ricavare la curva effettiva: considerando l'insieme di punti ricavati nel grafico dello spazio energy-delay, si estrapolano i valori di ritardo massimo e minimi, che risultano essere circa:

- Dmin=160ps
- Dmax=250ps

Facendo variare il vincolo  $D_0$  all'interno di questo range e iterando il tool di ottimizzazione, si ricava per ognuna una coppia  $x_1, x_2$ . Per ciascuna coppia, relativa al vincolo in ritardo, si sostituiscono i valori in LTSpice e si ottengono i valori di delay e energia. Tutti questi dati ricavati sono riportati in tabella:

D0	x1	x2	Energia[fJ]	Delay[ps]
160	3.64824902	13.3276134	6.79E-14	1.71E-10
170	3.28287199	11.5143122	6.27E-14	1.71E-10
180	2.47035149	8.17721728	5.51E-14	1.80E-10
190	2.08538763	6.87041418	5.22E-14	1.89E-10
200	1.82543892	6.03074796	5.05E-14	1.99E-10
210	1.61696199	5.43064886	4.93E-14	2.08E-10
220	1.50263421	4.91527959	4.85E-14	2.18E-10
225	1.38414689	4.76047488	4.82E-14	2.22E-10
235	1.27777231	4.39786004	4.75E-14	2.32E-10
240	1.22701168	4.24254845	4.72E-14	2.37E-10
250	1.11067366	3.99082455	4.68E-14	2.47E-10

Figura 24 - Tabella riassuntiva della coppia Energy-Delay ricavati post-ottimizzazione

I punti ottenuti vengono infine plottati in un grafico e la loro linea di tendenza rappresenta la curva energy-delay ottima, perfettamente compatibile con il grafico ottenuto inizialmente, come visibile in figura:

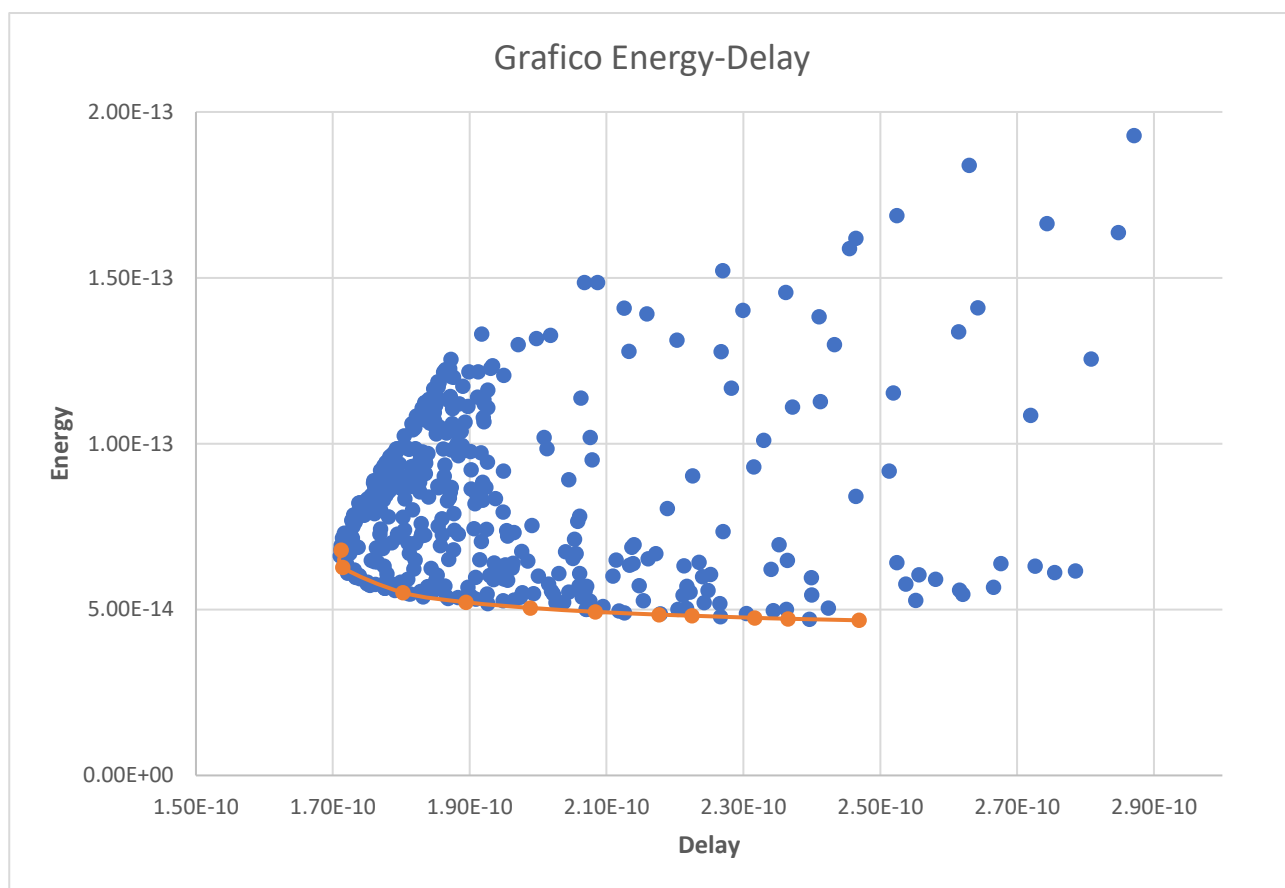


Figura 25 - Grafico dello spazio Energy-Delay con relativa curva ottimizzata