1 Introduzione

1.1 Il modello

I valori standard presi come riferimento sono:

T [K]	W [m]	L [m]	Vt0n [V]	Vt0p [V]	n	tch	Term_V [m/s]	Cinv [F/m²]	Kds	rpara [Ω]
300	1e-6	30e-9	0,3	-0,3	1	0,7	1,23e5	0,02	0,2	60

Una simulazione può avere parametri differenti dai precedenti al fine di enfatizzare un determinato comportamento del modello impiegato. Per utilizzare un modello parametrico configurabile attraverso uno script file SIMetrix è necessario commentare (e quindi escludere) i valori riportati nel modello stesso, come viene riportato di seguito:

PMOS:

```
.subckt SUB_PMOS drainin gatein sourcein
.param T = 300
.param W = 1e-6
.param L = 30e-9
.param Vt0 = -0.35
.param n=1
.param tch = 0.70
.param Term_V = 1.23e5
.param Cinv = 0.020
.param Kds = 0.04
.param rpara = 60
.param Vt0n = 0.35
.param Vt0p = -0.35
*.param Vt0 = {{(Vt0p)}}
.param Vth= {BOLTZ * {(T)} / ECHARGE }
.param Vt= { {(Vt0)} - ({(Kds)} * V(drain,source)) }
[…]
```

```
.subckt SUB_PMOS drainin gatein sourcein
*.param T = 300
*.param W = 1e-6
*.param L = 30e-9
*.param Vt0 = -0.35
*.param n=1
*.param tch = 0.70
*.param Term_V = 1.23e5
*.param Cinv = 0.020
*.param Kds = 0.04
*.param rpara = 60
*.param Vt0n = 0.35
*.param Vt0p = -0.35
.param Vt0 = {{(Vt0p)}}
.param Vth= {BOLTZ * {(T)} / ECHARGE }
.param Vt= { {(Vt0)} - ({(Kds)} * V(drain,source)) }
[…]
```

NMOS:

```
.subckt SUB_NMOS drainin gatein sourcein
.param T = 300
.param W = 1e-6
.param L = 30e-9
.param Vt0 = 0.35
.param n=1
.param tch = 0.70
.param Term_V = 1.23e5
.param Cinv = 0.020
.param Kds = 0.04
.param rpara = 60
.param Vt0n = 0.35
.param Vt0p = -0.35
*.param Vt0 = {{(Vt0n)}}
.param Vth= {BOLTZ * {(T)} / ECHARGE}
.param Vt= {{(Vt0)}-{Kds}*V(drain,source)}
[…]
```

```
.subckt SUB_NMOS drainin gatein sourcein
*.param T = 300
*.param W = 1e-6
*.param L = 30e-9
*.param Vt0 = 0.35
*.param n=1
*.param tch = 0.70
*.param Term_V = 1.23e5
*.param Cinv = 0.020
*.param Kds = 0.04
*.param rpara = 60
*.param Vt0n = 0.35
*.param Vt0p = -0.35
.param Vt0 = {{(Vt0n)}}
.param Vth= {BOLTZ * {(T)} / ECHARGE}
.param Vt= {{(Vt0)}-{Kds}*V(drain,source)}
[…]
```

Commentando i parametri è quindi possibile definire delle variabili globali all'interno dell'ambiente SIMetrix da modificare a piacimento.

1.2 Gli script file

SIMetrix permette l'utilizzo di particolare Script (scritti in un linguaggio simile al BASIC) per automatizzare diverse operazioni. I file di scripting sono utilizzati ampiamente per definire dinamicamente parametri simulativi, tipologie di simulazioni e tutto il necessario all'ambiente simulativo per compiere le operazioni necessarie. I cicli vengono utilizzati per eseguire simulazioni ricorsive su parametri ottenuti da simulazioni precedenti. Per quanto riguarda la sintassi si rimanda alle pagine[1] dedicate.

Siccome il nostro modello non possiede più al suo interno le definizioni dei vari parametri dei transistori, è necessario, prima di lanciare una simulazione, eseguire uno script file che contiene la definizione di tutti i parametri globali, pena l'arresto del processo di simulazione a causa della mancanza di alcune variabili.

In particolare, si consiglia di utilizzare il seguente snippet come incipit di ogni script file, in modo da non "dimenticarsi" di inizializzare le variabili:

```
let global:T = 300
let global:W = 1e-6
let global:L = 30e-9
let global:n = 1.3
let global:tch = 0.70
let global:Term_V = 1.23e5
let global:Cinv = 0.020
let global:Kds = 0.2
let global:rpara = 60
let global:Vt0n = 0.3
let global:Vt0p = -0.3
```

1.3 Command Shell

La command shell di Simetrix permette di visualizzare parametri simulativi e di eseguire tutti i comandi compatibili con gli script file. In particolare, è possibile visualizzare "run-time" tutte le grandezze derivate da una simulazione con il comando "Display"

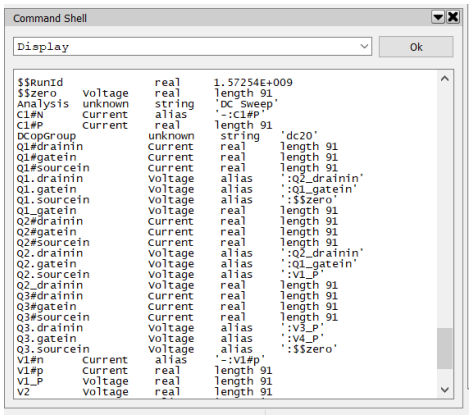


Figura 1

Per visualizzare i dati grezzi dei vari vettori di tensione è possibile utilizzare il comando "Show":

Show <nome del dispositivo>.<nome porta>

Per quanto riguarda le correnti:

Show <nome del dispositivo>#<nome porta>

[1] https://help.simetrix.co.uk/8.0/simetrix/simetrix_docs.htm#mergedProjects/script_manual/topics/sr_com_curve.htm

2 Analisi dei tempi di propagazione FO4

Una figura di merito molto importante per caratterizzare il tempo di propagazione di un circuito è la misura del tempo FO4 (Fan Out 4). Lo scopo di questa indagine è appunto simulare il comportamento di un invertitore che pilota 4 altri invertitori simili a lui.

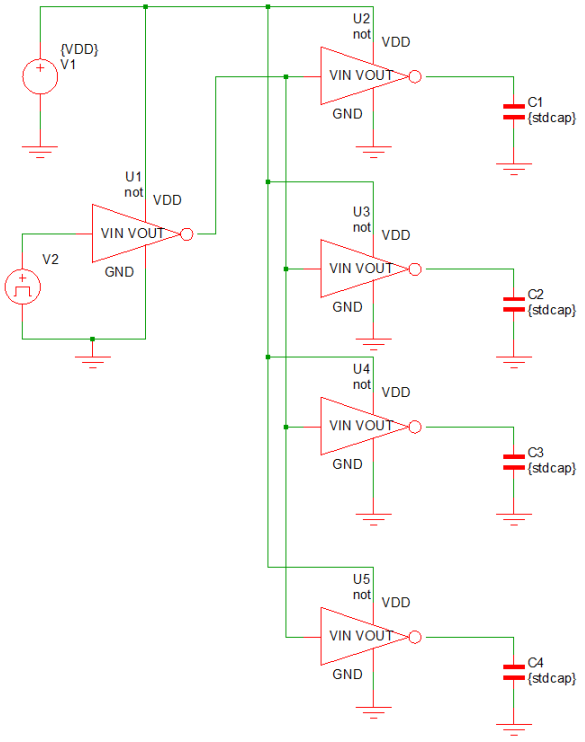


Figura 2

Notare i valori dei condensatori che sono settati al parametro {stdcap}, mentre la tensione di alimentazione è settata a {VDD}. Tale modifica è necessaria quando si vuole automatizzare l'analisi di un circuito. In questo caso i valori dei condensatori e della tensione di alimentazione verranno modificati dallo script file riportato di seguito. Il generatore di forme d'onda V2 deve avere i seguenti parametri:

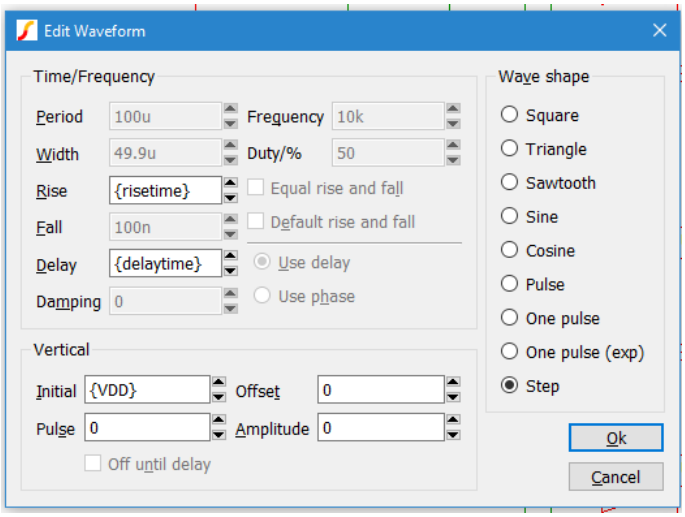


Figura 3

Lo script file necessario per avviare la simulazione è il seguente:

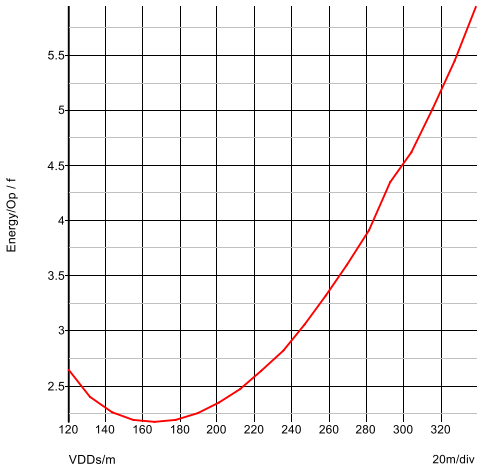
```
*transistor params
let global:T = 300
let global:W = 1e-6
let global:L = 30e-9
let global:n = 1.0
let global:tch = 0.70
let global:Term_V = 1.23e5
let global:Cinv = 0.020
let global:Kds = 0.2
let global:rpara = 60
let global:Vt0n = 0.3
let global:Vt0p = -0.3

* device params
let global:stdcap = 20f
let global:risetime = 0.1p
let global:delaytime= 0p

* simulation params
let global:simtime = 20p
let global:maxstep = 20f

let VDDs = [ 0.3 0.4 0.5 0.6 0.7]
let FO4t = vector(length(VDDs))
for iVDD = 0 to length(VDDs)-1
  let global:VDD = VDDs[iVDD]
  Netlist design.net
  Run /noerr /file design.net /an {'.TRAN 0 '& STR(simtime) &' 0 '& STR(maxstep)}
  let FO4t[iVDD] = XfromY(U1.VOUT, 0.9*VDD) - delaytime
  Echo
  Echo FO4 time: {FO4t[iVDD]}
  if iVDD == 0 then
    Plot U1.VOUT
  else
    Curve U1.VOUT
  endif
next iVDD
Plot XY(FO4t, VDDs)
```

---

3 Analisi energetica di un Full Adder

3.1 Circuito e simulazione

In questa esercitazione si andrà ad analizzare l'andamento dell'energia assorbita da un circuito al variare della tensione di alimentazione. Spesso si è portati a pensare che scalando la tensione di alimentazione di un circuito si riduce anche l'energia necessaria a svolgere la computazione. Questo è vero fino a che il tempo necessario per eseguire un certo numero di operazioni non implica un consumo statico più alto dell'energia necessaria ad eseguire le commutazioni stesse. Tale condizione accade per tensioni di alimentazione molto basse. Se viene riportata l'energia per operazione in funzione della tensione di alimentazione si può ottenere il seguente grafico:



Per estrarre in maniera efficace la curva Energia per operazione – tensione di alimentazione è necessario analizzare un circuito complesso, nel nostro un FULL ADDER. Per cui è necessario creare le seguenti porte logiche:

la porta AND

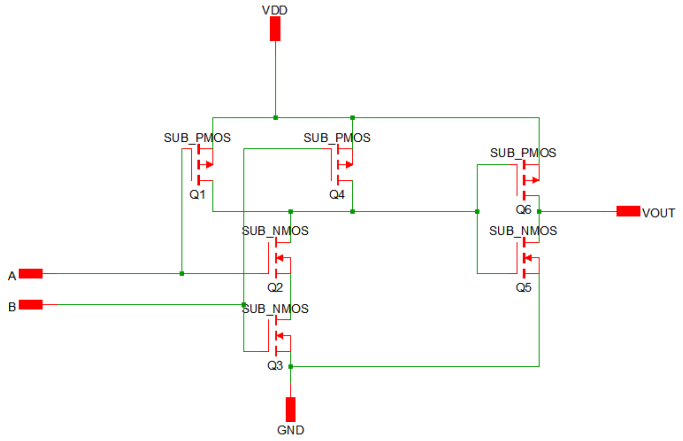


Figura 4

La porta OR:

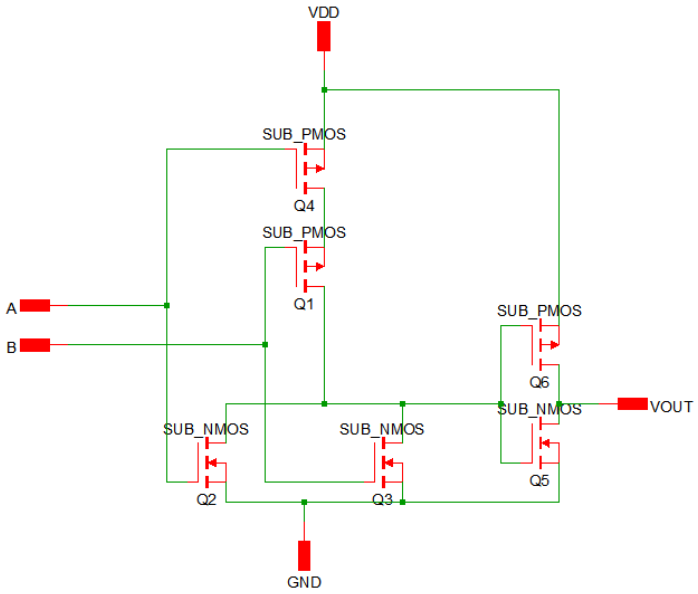


Figura 5

La porta XOR:

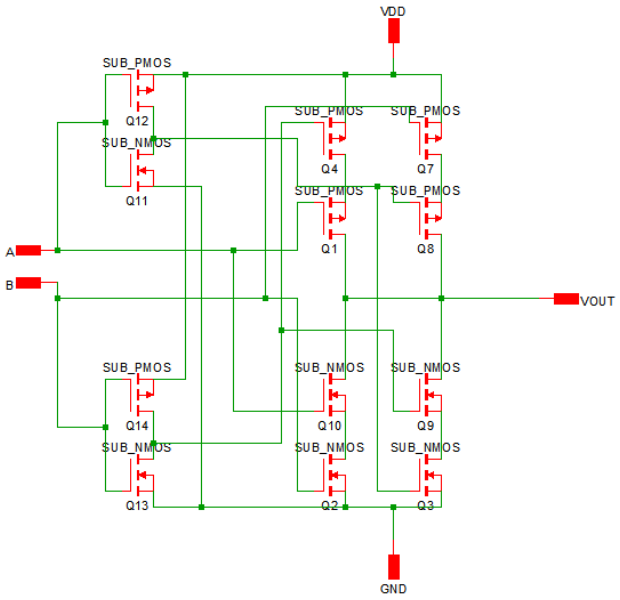


Figura 6

Un HALF ADDER:

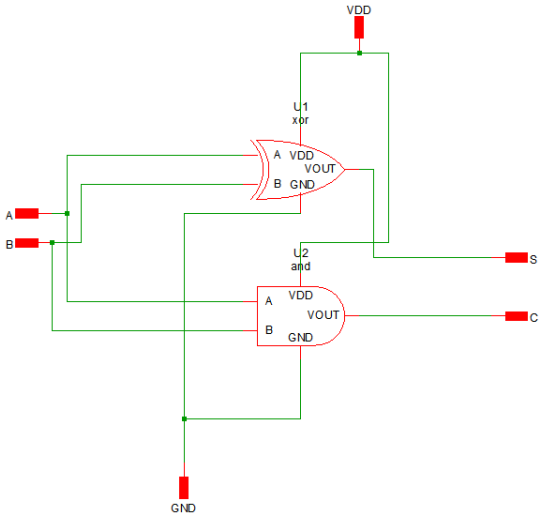


Figura 7

E quindi di un FULL ADDER, ad un singolo bit:

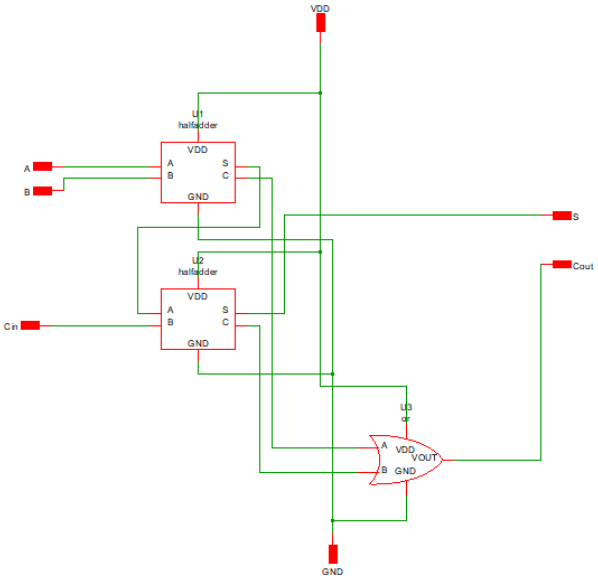


Figura 8

Connettendo in cascata 8 full adder è possibile simulare un sommatore a 8 bit (ci fermeremo al FULL ADDER a singolo bit).

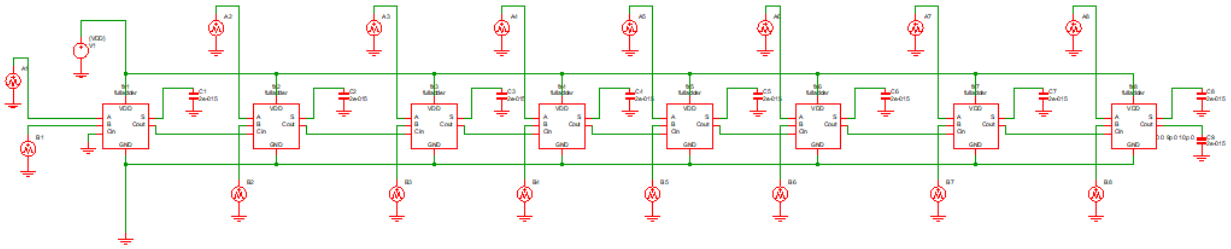


Figura 9

Tale circuito è sufficientemente complesso per analizzare alcuni caratteri generali dei circuiti digitali al variare di determinati parametri, come ad esempio la tensione di alimentazione (VDD) e/o la pendenza di sottosoglia del transistore (variando il fattore di idealità n), che con circuiti più semplici sarebbero difficili da interpretare.

L'idea di base è simulare una serie di somme, ed analizzare l'energia consumata del dispositivo per operazione, al fine di ottenere un grafico al variare della tensione di alimentazione del circuito. Per fare ciò è necessario seguire un certo iter:

STEP 1 – Ricerca del tempo di propagazione

Il circuito di test è il seguente:

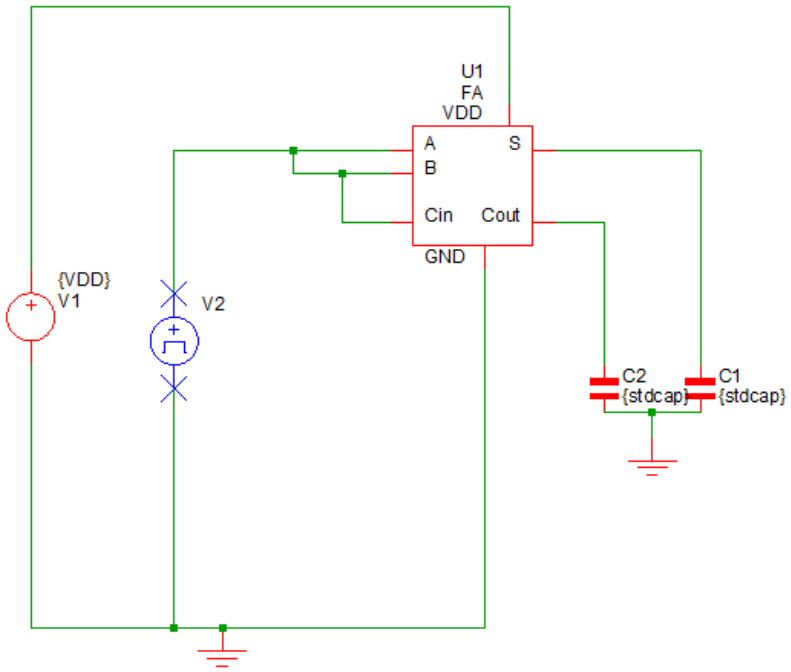


Figura 10

Dove V2 deve essere:

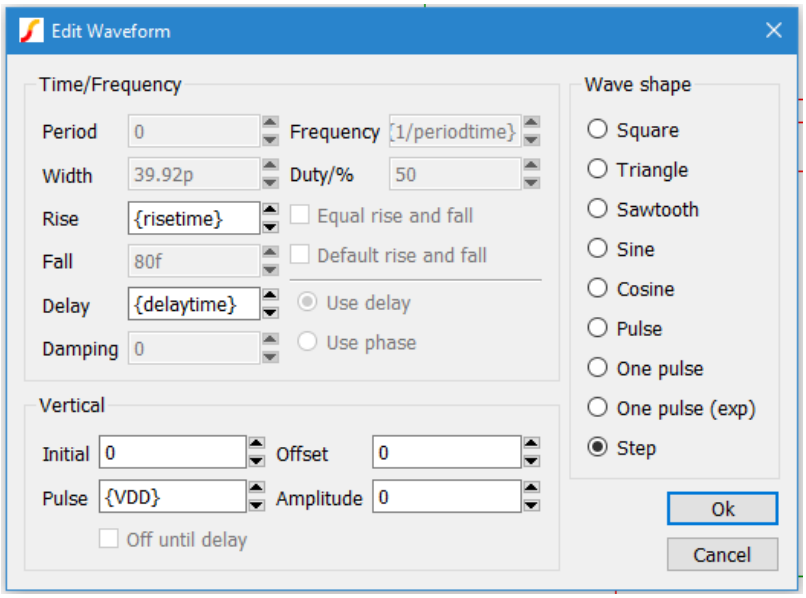


Figura 11

Tramite il seguente script file sarà possibile analizzare i tempi di propagazione del circuito:

```
*transistor params
let global:T = 300
let global:W = 1e-6
let global:L = 30e-9
let global:n = 1.3
let global:tch = 0.70
let global:Term_V = 1.23e5
let global:Cinv = 0.020
let global:Kds = 0.2
let global:rpara = 60
let global:Vt0n = 0.3
let global:Vt0p = -0.3
```

```
* device params
let global:stdcap = 20f
let global:delaytime= 0
let global:risetime = 0.3p

* simulation params
let global:simtime = 70n
let global:maxstep = 0.1p

let VDDs = [ 0.12 0.14 0.16 0.18 0.2 0.25]
let proptime = vector(length(VDDs))

for iVDD = 0 to length(VDDs)-1
let global:VDD = VDDs[iVDD]
Netlist design.net
Run /noerr /file design.net /an {'.TRAN 0 '& STR(simtime) &' 0 '& STR(maxstep)}
let proptime[iVDD] = XfromY(U1.S, 0.9*VDD) - delaytime
Echo
Echo Propagation time: {proptime}

if iVDD == 0 then
Plot U1.S
else
Curve U1.S
endif
next iVDD
Plot XY(proptime, VDDs)
```

Una volta ottenuti i tempi di propagazione è necessario riportarli nella tabella in fondo a questo documento: ci serviranno per lo step 2.

STEP 2 – Determinazione dell'energia per operazione

Modificare il generatore di forme d'onda V2 come riportato:

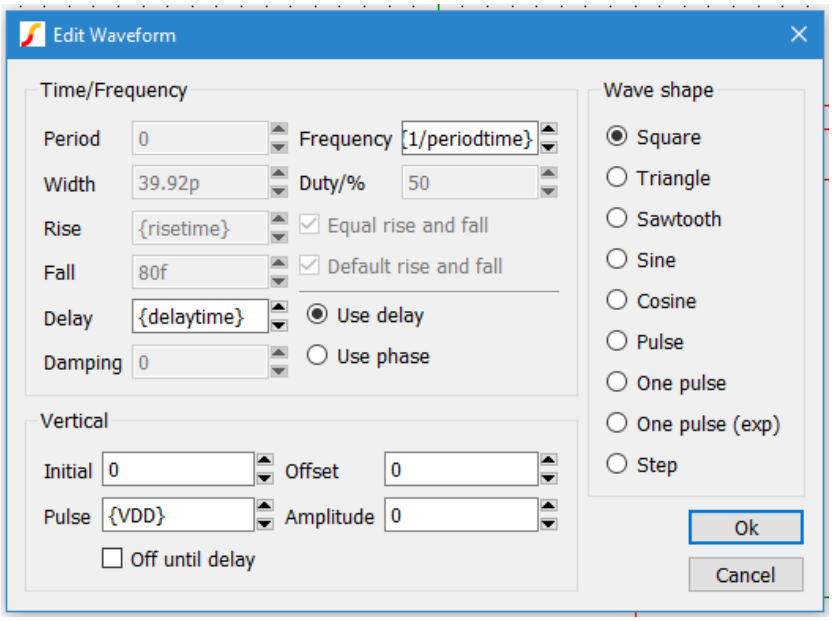


Figura 12

Il seguente script file, una volta modificato con i dati che sono stati ricavati dallo STEP 1, fornirà in uscita l'energia necessaria ad eseguire le 10 operazioni. (Questo script va eseguito per ogni tensione di alimentazione e quindi per ogni tempo di propagazione trovato nello STEP 1)

```
*transistor params
let global:T = 300
let global:W = 1e-6
let global:L = 30e-9
let global:n = 1.3
let global:tch = 0.70
```

```
let global:Term_V = 1.23e5
let global:Cinv = 0.020
let global:Kds = 0.2
let global:rpara = 60
let global:Vt0n = 0.3
let global:Vt0p = -0.3


********** DA MODIFICARE *******
let global:VDD = <<VDD da inserire>>
let global:proptime = <<tprop da inserire>>
********************************

* device params
let global:stdcap = 20f
let global:delaytime= 0
let global:N_op = 5
let global:maxstep = proptime/25000

* simulation params
let global:simtime = proptime*2 * 2 * N_op
let global:periodtime = proptime*2 * 2


Netlist design.net
Run /noerr /file design.net /an {'.TRAN 0 '& STR(simtime) &' 0 '& STR(maxstep)}
Echo
Echo Propagation time: {proptime}
Plot U1.S
let pow = U1.VDD *abs(U1#VDD)
let energy = integ(pow)
plot integ(pow)
Echo
echo Energy: {energy[length(energy)-1]}
```

Cosa abbiamo ottenuto?

$VDD\ [V]$	$t_{prop}\ [ns]$	$Energia\ [fJ]$