

Universidad de Chile Facultad de Ciencias Físicas y Matemáticas Departamento de Ingeniería Eléctrica

# Guía de HSPICE

Preparado por: Jorge Marín N.

Primera Versión: 10 de julio de 2008

## Índice

I.	Intr	oducción	- 2 -
II.	Obj	etivo del manual	- 2 -
III.	Un	Primer Ejemplo	- 3 -
IV.	Vist	ta General de HSPICE	10 -
С	omp	onentes Básicos	10 -
	1.	Fuentes	10 -
	2.	Elementos pasivos	11 -
	3.	Dispositivos activos	12 -
Т	ipos	de Análisis Básicos	14 -
	1.	Análisis Transiente	14 -
	2.	Análisis de Barrido DC (DC Sweep)	14 -
	3.	Análisis AC (en Frecuencia)	15 -
С	omai	ndos de Salida	16 -
	1.	Impresión de resultados	16 -
	2.	Gráfico de Resultados	16 -
	3.	Generación de Archivos de Gráfico	17 -
V.	Sim	ulación en Régimen Transiente	18 -
VI.	Sim	ulación de Barrido DC	21 -
VII.	Sim	ulación en Frecuencia (AC)	26 -
VIII	.Ejeı	rcicios propuestos	30 -
IX.	Bib	liografía	33 -

## I. Introducción

HSPICE (versión comercial del software SPICE, *Simulation Program with Integrated Circuits Emphasis*, desarrollado en 1975 en la Universidad de Berkeley) es una herramienta de la empresa Synopsys para la simulación de circuitos análogos. Es posible simular circuitos electrónicos en régimen permanente, transiente y en el dominio de la frecuencia. Además, es posible efectuar análisis específicos de comportamiento paramétrico y variaciones estadísticas.

HSPICE es un software de uso industrial, es decir, es ampliamente utilizado por fabricantes y diseñadores de circuitos integrados a gran escala. Su utilidad en este ámbito es la posibilidad de obtener comportamientos fidedignos de circuitos (o parte de ellos) conformados por componentes que simulan muy aproximadamente el funcionamiento físico de elementos y dispositivos reales (resistencias, transistores, condensadores, entre otros). Además, permite obtener comportamientos estadísticos de circuitos, útiles sobre todo en análisis relacionados con el proceso de fabricación de éstos, obteniendo medidas de desempeño importantes para la evaluación de desempeño de estos procesos. Todo lo anterior permite diseñar circuitos de manera virtual, disminuyendo en una proporción importante el alto costo de experimentar con dispositivos reales.

## II. Objetivo del manual

El presente manual pretende introducir al alumno en el uso práctico del programa HSPICE. Cuenta con ejemplos simples de las aplicaciones más comunes, así como ayuda para inicializar los programas necesarios para la obtención de resultados visuales y numéricos, como el Terminal de Linux y el visor de formas de onda CosmosScope de Synopsys.

El contenido de esta guía no pretende ser una muestra de todas las funcionalidades que posee HSPICE, sino que incluye las nociones fundamentales de la herramienta que serán utilizadas durante este curso.

## III. Un Primer Ejemplo

Para acceder a las herramientas HSpice y CosmosScope, instalados en el servidor "Tiuque" (cuya ubicación física actual es el Laboratorio de Sistemas Inteligentes), se debe iniciar una sesión en alguna cuenta disponible para los alumnos. Se recomienda el siguiente procedimiento:

- 1. Iniciar una sesión en un PC con sistema operativo LINUX (KDE, Gentoo, Red Hat, Ubuntu, entre otros).
- 2. La herramienta HSpice, funciona a través de líneas de comando, desde el terminal de LINUX. Luego, es necesario inicializar el terminal de LINUX. En general, es posible acceder presionando el ícono que representa una pantalla negra en la interfaz principal del sistema operativo (ver Figura 1). El nombre de la aplicación es "Terminal". Se abrirá una ventana con el nombre de la ubicación actual en las carpetas de Linux y un cursor donde se puede escribir, tal como se ve en la Figura 1.



Figura 1: Interfaz del Sistema Operativo Linux (Red Hat) con la ventana del Terminal abierta

La interfaz de la figura corresponde a la distribución "Red Hat" de Linux, y se puede ver el ícono del terminal encerrado en rojo en la parte inferior izquierda de la figura (esto puede diferir de distribución en distribución). Se utilizarán los siguientes comandos estándar del terminal:

- cd ..: sube un nivel, i.e. nos ubica en la carpeta inmediatamente superior a la actual.
- cd [nombre de subcarpeta]: nos ubica dentro de una subcarpeta de la carpeta actual.
- 1s: permite ver las subcarpetas que contiene la carpeta actual.
- mkdir [nombre de la carpeta]: crea un directorio con nombre especificado.
- 3. Para acceder a la cuenta, es necesario tener un login y un password. Se han habilitado 60 cuentas para el uso de alumnos. El login para estas cuentas es "usrNN", donde NN es un número entre 01 y 60 (nota: en caso de ser un número entre 1 y 9, es necesario anteponer un cero; es decir, en total de dígitos debe ser igual a dos). El password, por otra parte, es idéntico al login: usrNN. El acceso se hace vía SSH, utilizando la siguiente línea de comando:

ssh -X [usrNN]@tiuque.li2.uchile.cl

Aparecerá un mensaje solicitando el password, el cual debe ser ingresado según las instrucciones dadasanteriormente. El terminal desplegará algunas líneas indicando que se ha conectado con éxito.

Para HSpice, el funcionamiento es a través de líneas de comando, desde el terminal de Linux. A continuación, se señalan una serie de pasos para llevar a cabo la simulación de un primer programa en Hspice:

 Para poder simular un circuito en Hspice, primero es necesario poseer un archivo de entrada, el cual debe describir apropiadamente los componentes del circuito que se desea simular, sus conexiones, el tipo de simulación que se hará y diversos parámetros implicados en la operación. Por el momento, usaremos el siguiente texto en el archivo a simular:

\*\*\*acsource.sp
VIN 1 0 SIN 0 5 1meg 0
R1 1 0 5k
.OPTIONS POST
.TRAN 10n 1u
.END

El texto anterior se debe guardar con la extensión ".sp" (en adelante, se usará el nombre "acsource.sp" para este archivo). Por el momento, no nos ocuparemos del por qué de cada línea, sino de lograr simular exitosamente el archivo.

2. Para correr el archivo "acsource.sp" en HSpice, se recomienda seguir los siguientes pasos:

- Abrir un editor de texto en el PC que está utilizando y copiar el texto de la netlist a utilizar. Se recomienda utilizar el nombre acsource.sp al guardar el archivo (la extensión .sp es estrictamente necesaria).
- En el terminal conectado a Tiuque, ejecutar el comando "ls", para ver las carpetas en el directorio actual.
- Crear el directorio "test\_hspice" con el comando "mkdir test\_hspice".
- Hacer "Is". La carpeta creada debería aparecer en la lista.
- Ir a la carpeta "test\_hspice" con el comando "cd test\_hspice". Hacer "ls". La carpeta debería aparecer vacía.
- Enviar el archivo al directorio "test\_hspice" en la cuenta del servidor tiuque que está utilizando, mediante el siguiente comando:

```
scp [[user10]host1]:path_origen [usrNN]0tiuque.li2.uchile.cl:path_destino
```

El texto [[user1@]host1] representa el usuario y el host del PC utilizado. Los textos path\_origen y path\_destino son las direcciones del archivo que se está enviando y del archivo que será guardado en el servidor Tiuque (en este caso, en test\_hspice/ acsource.sp).

• Hacer "ls" en la carpeta "test\_hspice" para asegurarse que el archivo "acsource.sp" está contenido en ella.

La figura 2 ilustra lo anterior paso a paso.

Shell - Konsole -6>	- 0 X
Session Edit View Bookmarks Settings Help	
<pre>41 jorgem@cl01rhx8602:~&gt;ls cscope_2007.12.log Desktop/ vpd.properties work/ 42 jorgem@cl01rhx8602:~&gt;mkdir test_hspice 43 jorgem@cl01rhx8602:~&gt;cd test_hspice/ vpd.properties work/ 44 jorgem@cl01rhx8602:~&gt;cd test_hspice/ 45 jorgem@cl01rhx8602:-&gt;ctest_hspice 45 jorgem@cl01rhx8602:-/test_hspice&gt;ls 46 jorgem@cl01rhx8602:-/test_hspice&gt;ls 47 jorgem@cl01rhx8602:-/test_hspice&gt;</pre>	
A Shell	

Figura 2: Pasos a seguir para generar una carpeta donde guardar el archivo de entrada

 Ejecutar el comando "hspice acsource.sp" para generar los archivos de salida de HSpice. Debería aparecer una larga serie de líneas con texto y números, y finalmente el texto "HSPICE job acsource.sp completed.", seguido por la fecha y la hora. 4. Hacer "Is". Deben haber aparecido los siguientes archivos en la carpeta "test\_hspice": "acsource.ic0", "acsource.st0" y "acsource.tr0", además de "acsource.sp", como se muestra en la figura 3.

Ƴ Shell - Konsole ⊲6>							- 0 X
Session Edit Vie	w Bookmarks Setti	ngs Help					
<pre># nodes =   # diodes= ice = 0</pre>	2 # elements= 0 # bjts =	2 0 # jf	ets =	0 # mosfets =	:	0 # va	▲ dev
analysis	time #	points t	ot. iter	conv.iter			
op point transient readin errchk setup output tot	0.00 0.00 0.00 0.00 0.00 al cpu time job started at job ended at	1 101 0.00 20:12:11 20:12:11	3 74 seconds 03/17/200 03/17/200	37 rev= 8	0		
>info: lic: Release HSPICE job acs Mon Mar 17 20: 48 jorgem@cl01 acsource.ic0 49 jorgem@cl01	***** hspice jo hspice token(s) ource.sp complet 12:11 CLT 2008 rhx8602:~/test_h acsource.sp acs rhx8602:~/test_h	b conclude ed. spice>ls ource.st0 spice>	d acsource	.tr0			
🟯 🔳 Shell							

Figura 3: Terminal luego de simular el archivo "acsource.sp"

Es deseable apreciar gráficamente los resultados de la simulación. Para esto, se utiliza la herramienta CosmosScope. Los siguientes pasos permitirán visualizar las ondas de salida del circuito simulado en CosmosScope.

 Se debe ejecutar la herramienta desde la carpeta en que está instalada, ejecutando el comando "cd [directorio de instalación]/CosmosScope/ai\_bin/" para ir al directorio en que está el programa, y luego iniciarlo con "./cscope", como se muestra en la figura 4. El "directorio de instalación" en este caso es "work", lo cual no será así en el caso general, por lo cual hay que averiguar tal ubicación.

❤ Shell - Konsole <6>				. 🗆 🗙
Session Edit View Book	marks Settings Help			
<pre>41 jorgem@cl01rhx8600 /remote/cl01home/jorg 42 jorgem@cl01rhx8600 ai_aimindex.ai_reg* ai_aimindex.ai_reg* ai_aimpkg.ai_reg* aim.ai_reg* aim.rameway* aimFrameway* aimFrameway.ai_reg* 43 jorgem@cl01rhx8600 ■</pre>	<pre>2:-&gt;cd work/CosmosS gem/work/CosmosScop 2:-/work/CosmosScop aimsh* cscope* cscope.ai_reg* guide* guide.ai_reg* keymanager* keymanager.ai_reg* 2:-/work/CosmosScop</pre>	<pre>cope/ai_bin/ ie/ai_bin/ ie/ai_bin&gt;ls probeserver* probeserver.ai_reg* scope* scopebook* scopebook* scopebook.ai_reg* scopeholl* scopehdl.ai_reg* e/ai_bin&gt;./cscope</pre>	<pre>scope.log sst* sst.ai_reg* start* txTool* txTool.ai_reg*</pre>	
🐣 🔳 Shell				-

#### Figura 4: Pasos para iniciar CosmosScope

2. Se abrirá una ventana como la que se muestra en la figura 5, que corresponde a la interfaz gráfica del programa.



Figura 5: Interfaz gráfica de CosmosScope

3. Para visualizar las curvas de salida, se debe ir a File > Plotfiles en la esquina superior izquierda de la ventana de CosmosScope, como se muestra en la figura 6. Aparecerá una ventana donde se pide buscar el archivo a visualizar.



Figura 6: Menú de opciones para abrir el archivo con curvas de salida de HSpice

4. Se debe seleccionar el archivo "acsource.tr0", el cual contiene las curvas que interesan; se abrirá dos ventanas, que se muestran en la figura 7.

3 美 美 八 🏎 🗛 🗴
♥ Signal Manager □ X
<u>File</u> <u>Plotfile</u> <u>Signals</u> Help
Signal filter Open Plotfiles
Plotfiles Close Plotfiles
(1) acsource.tr0
Setup
Match All
Close
(1) acsource.tr0
File Signal
Filter(delim=";")
→ v(0)
└ <u>↓</u> ↓ v(1)
7
X Axis: t Set Default
Plot Match Deselect Close

Figura 7: Ventanas que se aparecen al abrir el archivo "acsource.tr0"

5. Se debe hacer doble click sobre las señales i(vin), v(0) y v(1) que aparecen listadas en la ventana con el encabezado "(1) acsource.tr0" de la figura 7. Las curvas aparecerán en la ventana de gráficos "Graph0", tal como se muestra en la figura 8.



Figura 8: Formas de onda resultantes de la simulación del archivo "acsource.sp"

## IV. Vista General de HSPICE

Como se vió en el ejemplo anterior, HSPICE requiere un archivo de entrada para su funcionamiento, denominado "netlist" en inglés. Este archivo, como se explicó, debe describir los componentes, conexiones, tipo de simulación y otros parámetros del circuito a analizar. Al correr el software desde la carpeta donde se encuentra el archivo netlist, HSPICE "leerá" el archivo de entrada, calculará iterativamente el resultado de la simulación y arrojará archivos de salida con la información solicitada por la netlist en la carpeta donde está contenida la netlist.

A continuación, se explicarán brevemente las sentencias que debe contener los archivos de entrada (netlist) para hacer las simulaciones deseadas.

## Componentes Básicos

Todo circuito puede contener tres tipos de componentes: fuentes, elementos pasivos y dispositivos activos.

- 1. **Fuentes:** Las fuentes son los componentes que estimulan el circuito de interés con voltages o corrientes de entrada, ya sean estas constantes o variables. Se presentará el caso de fuentes independientes (i.e. que no dependen de variables del sistema simulado).
  - i. Fuente continua: La declaración de este tipo de fuentes, de voltaje o corriente, en el archivo netlist se muestra a continuación:

```
Vxxx n+ n- dcval
Ixxx n+ n- dcval
```

donde:

- Vxxx/Ixxx: Nombre de la fuente. Debe comenzar con la letra "V" para el caso de fuente de voltaje e "I" para la fuente de corriente, seguido por otras letras o números
- n+: Nodo positivo al que se conectará la fuente
- n-: Nodo negativo al que se conectará la fuente
- dcval: Valor en Volts/Ampéres del voltaje/corriente de la fuente

Ejemplos:

• V1 1 GND 5V: Fuente de voltaje contínuo de 5 Volts entre el nodo "1" y el nodo de la tierra del sistema (potencial cero).

- I2 2 0 0.5A: Fuente de corriente corriente de valor VCC, donde VCC es una variable
- ii. Fuente Alterna: La declaración para fuentes de voltaje o corriente alternas para el caso sinusoidal se muestra a continuación:

Vxxx n+ n- SIN vo va freq td Ixxx n+ n- SIN vo va freq td

donde:

- Vxxx: Nombre de la fuente de voltaje; debe comenzar siempre con una V, seguida por letras o números
- n+: Nodo del terminal positivo al que se conectará la fuente; es nombrado por una cadena de letras y números, y será el identificador global que el programa usará para tal nodo
- n-: Nodo del terminal negativo al que se conectará la fuente.
- SIN: Palabra clave que indica que la fuente es de tipo sinusoidal
- vo: Valor offset de la fuente, es decir, valor de voltaje en torno al varía la sinusoide
- va: Voltaje peak de la sinusoide
- freq: Frecuencia de la función sinusoidal
- td: Tiempo de atraso; corresponde al tiempo en el cual comienza la onda
- 2. **Elementos pasivos:** Son los elementos clásicos utilizados en los circuitos pasivos (es decir, que no entregan potencia al sistema): resistencias, condensadores e inductancias.
  - i. Resistencias: Se declaran de la siguiente manera:

Rxxx n1 n2 Rval

- Rxxx: Nombre del elemento; debe comenzar con una R para el caso de las resistencias
- n1: Nodo positivo de la resistencia (al ser un elemento sin polaridad, no importa si se conecta "al revés")
- n2: Nodo negativo de la resistencia
- Rval: Valor de la resistencia en ohms

ii. Condensadores: Se declaran de la siguiente manera:

Cxxx n1 n2 Cval

- Cxxx: Nombre del elemento; debe comenzar con una C para el caso de las capacitancias
- n1: Nodo positivo de la capacitancia (el modelo básico mostrado es un elemento sin polaridad, por lo que no importa si se conecta "al revés")
- n2: Nodo negativo de la capacitancia
- Cval: Valor de la capacitancia en farads
- iii. Inductancias: Se declaran de la siguiente manera:

Lxxx n1 n2 Lval

- Lxxx: Nombre del elemento; debe comenzar con una L para el caso de las inductancias
- n1: Nodo positivo de la inductancia (al ser un elemento sin polaridad, no importa si se conecta "al revés")
- n2: Nodo negativo de la inductancia
- Lval: Valor de la inductancia en henry
- 3. **Dispositivos activos**: Se trata de componentes más complejos: diodos, transistores bipolares (BJT) y transistores MOS.
  - i. Diodos: Se declaran de la siguiente manera:

#### Dxxx n1 n2 dmod

- Dxxx: Nombre del elemento; debe comenzar con una D para el caso de los diodos
- n1: Nodo positivo del diodo (al ser un elemento con polaridad, es importante fijarse el sentido en que se conecta)
- n2: Nodo negativo del diodo
- dmod: Palabra reservada para modelo definido internamente en HSPICE (es posible definir otros modelos, como se verá más adelante)
- ii. Transistor bipolar (BJT): Se declaran de la siguiente manera:

Qxxx nc nb ne model

- Qxxx: Nombre del elemento; debe comenzar con una Q para el caso de los transistores bipolares
- nc: Nodo del terminal de colector del transistor bipolar
- nb: Nodo del terminal de base del transistor bipolar
- ne: Nodo del terminal de emisor del transistor bipolar
- model: Modelo definido internamente en HSPICE o por el usuario (más adelante se verá cómo hacerlo)
- iii. Transistor MOS: Se declaran de la siguiente manera:

Mxxx nd ng ns nb model

- Mxxx: Nombre del elemento; debe comenzar con una M para el caso de los transistores bipolares
- nd: Nodo del terminal de drenaje (drain) del transistor MOS
- ng: Nodo del terminal de puerta (gate) del transistor MOS
- ns: Nodo del terminal de fuente (source) del transistor MOS
- nb: Nodo del terminal de sustrato (bulk) del transistor MOS
- model: Modelo definido internamente en HSPICE o por el usuario (más adelante se verá cómo hacerlo)

## Tipos de Análisis Básicos

A todo circuito se le puede hacer uno o más de los siguientes tipos de análisis: transiente, barrido DC o AC (análisis en frecuencia). Los comandos para ejecutar cada tipo de análisis se listarán a continuación (en las secciones siguientes, se verán ejemplos de cada tipo).

1. **Análisis Transiente**: Se computa la solución del circuito en función del tiempo. Se utiliza, en general, una señal de onda alterna en la entrada. El formato de la declaración es:

.TRAN tstep1 tstop1

donde:

- .TRAN: Comando para comenzar la simulación transiente
- tstep1: Paso de tiempo entre medición y medición
- tstop1: Tiempo final
- Análisis de Barrido DC (DC Sweep): Se computa la solución del circuito para un "barrido" de valores de alguna fuente. Se utiliza una fuente continua, cuyo campo "dcval" debe ser reemplazado por la palabra clave "DC". La salida se debe imprimir con el comando ".PRINT", explicado más adelante. El formato de la declaración es:

.DC var1 start1 stop1 incr1

- .DC: Comando para comenzar la simulación DC
- var1: Variable a la que se le hará el "barrido"
- start1: Valor de comienzo para la variable "var1"
- stop1: Valor de final para la variable "var1"
- incr1: Paso de variación para la variable "var1"

3. Análisis AC (en Frecuencia): Se computa la solución del circuito para el comportamiento en función de la frecuencia (se hace un "barrido" para la frecuencia de alguna fuente). Se utiliza una fuente alterna de voltaje o corriente declarada de la siguiente forma:

```
Vxxx n+ n- AC acval phase Ixxx n+ n- AC acval phase
```

donde:

- AC: Palabra clave para comenzar la simulación AC
- acval: Magnitud de la onda de voltaje/corriente de la fuente
- phase: Ángulo de fase de la onda de voltaje/corriente de la fuente

La salida se debe imprimir con el comando ".PLOT", explicado más adelante. El formato de la declaración es:

.AC DEC 10 fstart fstop

- .AC: Comando para comenzar la simulación AC
- DEC: Palabra clave para determinar una variación por décadas de la frecuencia
- np: 10 puntos calculados por década
- fstart: Frecuencia inicial
- fstop: Frecuencia final

## Comandos de Salida

Para generar la información necesaria en los archivos de salida, se dispone de los siguientes comandos.

1. **Impresión de resultados**: Para el análisis DC, se debe utilizar el comando ".PRINT" para imprimir la información resultante en los archivos de visualización. El formato de la declaración es:

.PRINT DC V(n1)...V(nj)

donde:

- .PRINT: Comando para imprimir los resultados en archivos de salida
- DC: Opción para imprimir los resultados del análisis DC
- V(n1): Opción para medir el voltaje en el nodo n1
- V(nj): Opción para medir el voltaje en el nodo nj
- 2. **Gráfico de Resultados**: Para el análisis AC, se debe utilizar el comando ".PLOT" para generar la información resultante en forma información graficable en los archivos de visualización. El formato de la declaración es:

.PLOT Vdb(n1)...Vdb(nj)

- .PLOT: Comando para generar resultados graficables en archivos de salida
- Vdb(n1): Opción para medir el voltaje en decibeles en el nodo n1
- Vdb(n2): Opción para medir el voltaje en decibeles en el nodo nj

3. **Generación de Archivos de Gráfico**: Para cada tipo de análisis, se debe utilizar el comando ".PLOT" para generar archivos de salida tipo ".trN" que permite graficar los resultados. El formato de la declaración es:

.OPTIONS PLOT

- .OPTIONS: Comando para generar archivo graficable
- PLOT: Opción para generar el archivo ".trN" necesario para visualizar los resultados

## V. Simulación en Régimen Transiente

El circuito simulado en la parte III corresponde a una fuente de voltaje sinusoidal con una resistencia conectada en serie, tal como se muestra en la siguiente figura 9.



Figura 9: Esquema del circuito utilizado en el ejemplo

La simulación es de tipo transiente, es decir, calcula el comportamiento de un circuito en un cierto preríodo definido de tiempo, en general muy corto, en función de la variable temporal. Esto sirve para analizar reacciones particulares del circuito simulado frente a cierto estímulo.

A continuación, se explicará el archivo "acsource.sp" línea por línea:

1. \*\*\*acsource.sp

El símbolo "\*" al comienzo de una línea en una netlist comenta su contenido, es decir, lo ignora. En este caso, esta línea contiene dos asteriscos más y el nombre del archivo, sólo como información adicional. La primera línea siempre es considerada como un "título", es decir, nunca considera su contenido como parte de la simulación.

#### 2. VIN 1 0 SIN 0 5 1meg 0

Esta línea declara la inserción de una fuente de voltaje alterna al circuito. El formato de esta declaración es el de la fuente alterna explicado en la parte IV. Los valores de ejemplo significan, en el orden en que están escritos:

- VIN: Nombre de la fuente de voltaje; comienza con la letra "V" para señalar esto
- 1: Nodo del terminal positivo al que se conectará la fuente
- 0: Nodo del terminal negativo al que se conectará la fuente; en este caso, se utiliza el como nombre "0", con lo cual se establece que este nodo corresponde a la tierra del sistema (NOTA: también se pueden usar las palabras clave "GND" 0 "!GND" para hacer esto)
- SIN: Palabra clave que indica que la fuente es de tipo sinusoidal
- 0: Valor offset de la fuente, es decir, valor de voltaje en torno al varía la sinusoide; en este caso, la sinusoide de voltaje oscila en torno a cero.
- 5: Voltaje peak de la sinusoide; la sinusoide tiene una amplitud de 5 Volts
- freq: Frecuencia de la función sinusoidal, que en este caso, vale 1 MHz (se anota como "1 meg").
- td: Tiempo de atraso, que corresponde al tiempo en el cual comienza la onda; en este caso, la onda comienza apenas se inicia la simulación

#### 3. R1 1 0 5k

Esta declaración inserta una resistencia en el circuito. El formato de la declaración anterior se explicó en la parte IV. Según eso, tenemos que:

- R1: Nombre del elemento; comienza con una R para explicitar que se trata de una resistencia
- n1: Nodo positivo de la resistencia
- n2: Nodo negativo de la resistencia, que en este caso es la tierra (también el nodo negativo de la fuente)
- 5k: Hace que el valor de la resistencia sea de 5 Kohms.

4. .OPTIONS POST

El comando .OPTIONS tiene muchas funcionalidades. En este caso, al estar acompañado por la opción POST, permite generar el archivo "acsource.tr0" con la información para visualizar los resultados en un visor de forma de onda, como CosmosScope.

5. .TRAN 10n 1u

Esta opción comienza un análisis transiente sobre el circuito definido en un tiempo definido.

El formato de la declaración fue explicado en la parte IV. Según eso, tenemos que:

- 10n: Paso de tiempo entre medición y medición, de 10 nanosegundos
- 1u: Tiempo final de 1 microsegundo. Así, considerando el tamaño del paso de tiempo, esta simulación tomará 100 muestras equidistantes en el tiempo del comportamiento del circuito. Además, dada la frecuencia de la sinusoide (1 MHz), sabemos que el tiempo de simulación bastará para visualizar un periodo entero de la señal alterna de entrada
- 6. .END

Esta palabra clave debe ir en la última linea del archivo, para informar su final.

Una vez entendido el archivo, se puede ver que los resultados de la figura 8 son consistentes. En primer lugar, el votaje del nodo 1 es el de la fuente, con 5 volts de amplitud cetrado en cero volts. El voltaje del nodo 0 es constante y nulo, consistentemente con el hecho que este es la tierra del sistema. La corriente del circuito también sinusoidal, lo cual corresponde en el caso de una resistencia con voltaje aplicado sinusoidal. Su valor máximo es de 1 mA, de acuerdo con los valores dados (I = V/R = 5 V/5Kohm = 1 mA).

## VI. Simulación de Barrido DC

Para ejemplificar un análisis tipo "DC Sweep", se utilizará un inversor digital contruído con transistores MOS de tipo N y P (par CMOS), como se muestra en la figura 10.



Figura 10: Esquema del circuito inversor utilizado en el ejemplo

La simulación es de tipo Barrido DC, es decir, calcula el estado de las variables del sistema en función del valor de una única variable; en este caso, esa variable es el voltaje de entrada. Es interesante, para este caso, visualizar el comportamiento DC del voltaje de salida, pues nos mostrará claramente cómo el circuito "invierte" los voltajes.

El código a utilizar en este ejemplo (archivo netlist "inv\_dc.sp") es el siguiente:

```
***inv_dc.sp
MPL OUT IN VDD VDD CMOSP L=0.18u W=1.8u
MNL OUT IN 0 0 CMOSN L=0.18U W=0.9U
VDD VDD 0 1.8
VIN IN 0 DC
.DC VIN 0 2 0.01
CLOAD OUT 0 20fF
R OUT 0 1g
```

.OPTIONS LIST NODE POST

```
.PRINT DC V(IN) V(OUT)
```

.LIB "tsmc\_018um\_model" CMOS\_MODELS

.END

A continuación, se explicará el archivo "inv\_dc.sp" línea por línea:

1. \*\*\* inv\_dc.sp

Comentario con el nombre del archivo.

2. MPL OUT IN VDD VDD CMOSP L=0.18u W=1.8u

Esta línea declara la inserción de un transistor MOS tipo P al circuito. El formato de esta declaración es el del transistor MOS de la parte IV, pero se agregan los valores del ancho y largo del canal del transistor (W y L, respectivamente). Los valores de ejemplo significan, en el orden en que están escritos:

- MPL: Nombre del transistor; comienza con la letra "M" para señalar esto
- OUT: Nodo del terminal drenaje al que se conectará el transistor, que corresponde al terminal de salida del inversor
- IN: Nodo del terminal puerta al que se conectará el transistor, que corresponde al terminal de entrada del inversor
- VDD: Nodo del terminal fuente al que se conectará el transistor, que corresponde al voltaje de polarización
- VDD: Nodo del terminal sustrato al que se conectará el transistor, que corresponde al voltaje de polarización
- CMOSP: Nombre del modelo
- L=0.18u: Largo de canal de 0.18 micrometros
- W=1.8u: Ancho de canal de 0.18 micrometros

3. MNL OUT IN 0 0 CMOSN L=0.18U W=0.9U

Esta línea declara la inserción de un transistor MOS tipo N al circuito. Es análoga a la anterior.

- MNL: Nombre del transistor; comienza con la letra "M" para señalar esto
- OUT: Nodo del terminal drenaje al que se conectará el transistor, que corresponde al terminal de salida del inversor
- IN: Nodo del terminal puerta al que se conectará el transistor, que corresponde al terminal de entrada del inversor

- 0: Nodo del terminal fuente al que se conectará el transistor, que corresponde a la tierra del sistema
- 0: Nodo del terminal sustrato al que se conectará el transistor, que corresponde a la tierra del sistema
- CMOSN: Nombre del modelo
- L=0.18u: Largo de canal de 0.18 micrometros
- W=0.9u: Ancho de canal de 0.9 micrometros

4. VDD VDD 0 1.8

Fuente continua de polarización del circuito. Los significados de sus valores son:

- VDD: Nombre de la fuente de voltaje continuo; comienza con la letra "V" para señalar esto
- VDD: Nodo del terminal positivo de la fuente, correspondiente al del voltaje de polarización del circuito
- 0: Nodo del terminal negativo de la fuente, correspondiente al del voltaje de tierra del circuito
- 1.8: Valor del voltaje continuo de la fuente, 1.8 Volts
- 5. VIN IN 0 DC

Voltaje de entrada del circuito, que será variado para el análisis DC sweep.

- VIN: Nombre de la fuente de voltaje de entrada; comienza con la letra "V" para señalar esto
- IN: Nodo del terminal positivo de la fuente, correspondiente al del voltaje de entrada del circuito
- 0: Nodo del terminal negativo de la fuente, correspondiente al del voltaje de tierra del circuito
- DC: Opción que permite variar la magnitud del voltaje VIN

#### 6. .DC VIN 0 2 0.01

Sentencia que permite hacer el análisis DC sweep.

- .DC: Comando de análisis DC sweep
- VIN: Variable a barrer, en este caso el voltaje de entrada al circuito

- 0: Valor de comienzo de la variable, 0 Volts
- 2: Valor final de la variable, 0 Volts
- 0.01: Valor del paso de barrido, 0.01 Volt

#### 7. CLOAD OUT 0 20fF

Agrega un condensador de carga a la salida del circuito.

- CLOAD: Nombre del condensador; comienza con la letra "C" para señalar esto
- OUT: Nodo del terminal positivo del condensador, correspondiente al del voltaje de salida del circuito
- 0: Nodo del terminal negativo del condensador, correspondiente al del voltaje de tierra del circuito
- 20fF: Valor del condensador, 20 fectoFarad (muy pequeño)

#### 8. R OUT 0 1g

Agrega una resistencia de carga a la salida del circuito.

- R: Nombre de la resistencia; comienza con la letra "R" para señalar esto
- OUT: Nodo del terminal positivo de la resistencia, correspondiente al del voltaje de salida del circuito
- 0: Nodo del terminal negativo de la resistencia, correspondiente al del voltaje de tierra del circuito
- 1g: Valor de la resistencia, 1 gigaOhm (muy grande)

9. .OPTIONS POST

El comando .OPTIONS tiene muchas funcionalidades. En este caso, permite generar el archivo "acsource.tr0" con la información para visualizar los resultados en un visor de forma de onda, como CosmosScope.

10. .PRINT DC V(IN) V(OUT)

Permite imprimir en los archivos de salida el valor del voltaje de entrada y salida.

11. .LIB "tsmc\_018um model" CMOS\_MODELS

Sentencia que llama la librería donde se encuentran los modelos de transistor MOS utilizados.

12. .END

Esta palabra clave debe ir en la última línea del archivo, para informar su final.

Una vez entendido el circuito, es posible simularlo y visualizar la curva de salida en CosmosScope, tal como se ha visto anteriormente. El resultado esperado se muestra en la figura 11.



Figura 11: Gráfico del comportamiento en barrido DC del circuito inversor

Se puede ver que el comportamiento es acorde con la funcionalidad del circuito: cuando el voltaje de entrada VIN es bajo (cercano a 0 Volts), el voltaje de salida VOUT es alto (cercano al valor del voltaje de polarización). Por el contrario, cuando el voltaje VIN es alto y cercano al de polarización, el voltaje de salida VOUT es bajo, cercano a cero. Es decir, el circuito invierte el voltaje de entrada. Se debe notar que la caída de un estado a otro es muy abrupta, y el rango de voltaje de entrada de "incertidumbre" es pequeño, por lo tanto es muy efectivo para aplicaciones digitales.

## VII. Simulación en Frecuencia (AC)

Un ejemplo indicado para comprender el análisis de HSPICE en frecuencia es el de un filtro pasa banda, como se esquematiza en la figura 12.



Figura 12: Esquemático del filtro pasa banda capacitivo

Como se muestra en la figura, el filtro pasa banda a simular es de tipo capacitivo, y conformado por la conexión en cascada de un filtro pasa bajo y un filtro pasa bajo. La fuente Vin, para el caso de la simulación AC, recorrerá un rango de frecuencias, manteniendo una amplitud de 1 V y una forma de onda sinusoidal. En definitiva, lo que importa en este tipo de análisis, será la magnitud del voltaje de salida para cada frecuencia de entrada. En teoría, se debería obtener un intervalo de "paso" de frecuencias, y el resto debería estar atenuado por el efecto del circuito.

El código a utilizar en este ejemplo (en el archivo "lpfilter\_tr.sp") se muestra a continuación:

\*\*\*lpfilter\_fr.sp VIN N1 0 AC 1 R1 N1 N2 200 C1 N2 0 2.5u C2 N2 N3 1u RLOAD N3 0 1k .AC DEC 1 1 100k .OPTION POST .END A continuación, se explicará el archivo "lpfilter\_tr.sp" línea por línea:

1. \*\*\*lpfilter\_fr.sp

Primera línea de comentario con el nombre del archivo, al igual que en los casos anteriores.

2. VIN N1 0 AC 1

Fuente de voltaje alterna que variará su frecuencia en un rango definido.

- VIN: Nombre de la fuente de voltaje de entrada; comienza con la letra "V" para señalar esto
- N1: Nodo del terminal positivo de la fuente, correspondiente al del voltaje de entrada del circuito
- 0: Nodo del terminal negativo de la fuente, correspondiente al del voltaje de tierra del circuito
- AC: Opción que permite variar la frecuencia del voltaje VIN
- 1: Valor de la magnitud del voltaje durante la simulación.
- 3. R1 N1 N2 200

Agrega una resistencia de carga a la parte del filtro pasa bajo del circuito.

- R1: Nombre de la resistencia; comienza con la letra "R" para señalar esto
- N1: Nodo del terminal positivo de la resistencia, correspondiente al del voltaje de entrada del circuito completo
- N2: Nodo del terminal negativo de la resistencia, correspondiente al del voltaje de salida del filtro pasa bajo en cascada
- 200: Valor de la resistencia, 200 Ohm
- 4. C1 N2 0 2.5u

Agrega un condensador de carga en el circuito pasa bajo en cascada.

- C1: Nombre del condensador; comienza con la letra "C" para señalar esto
- N2: Nodo del terminal positivo del condensador, correspondiente al del voltaje de salida del circuito pasa bajo
- 0: Nodo del terminal negativo del condensador, correspondiente al del voltaje de tierra del circuito
- 2.5uF: Valor del condensador, 20 microFarad

#### 5. C2 N2 N3 1u

Agrega un condensador de carga a la salida del circuito.

- C2: Nombre del condensador; comienza con la letra "C" para señalar esto
- N2: Nodo del terminal positivo del condensador, correspondiente al del voltaje de salida del circuito pasa bajo
- N3: Nodo del terminal negativo del condensador, correspondiente al del voltaje de salida del circuito completo
- 1uF: Valor del condensador, 1 microFarad

#### 6. RLOAD N3 0 1k

Agrega una resistencia de carga a la salida del circuito.

- RLOAD: Nombre de la resistencia; comienza con la letra "R" para señalar esto
- N3: Nodo del terminal positivo de la resistencia, correspondiente al del voltaje de salida del circuito
- 0: Nodo del terminal negativo de la resistencia, correspondiente al del voltaje de tierra del circuito
- 1k: Valor de la resistencia, 1 KiloOhm

#### 7. .AC DEC 21 1 100k

Sentencia que permite hacer el análisis AC

- .DC: Comando de análisis AC
- DEC: Opción para fijar un barrido de frecuencia por décadas (logarítmico)
- 21: Cantidad de puntos por década a calcular.
- 1: Valor inicial de la frecuencia, 1 Hertz
- 100k: Valor final de la frecuencia, 100 kiloHertz

#### 8. .OPTION POST

El comando .OPTIONS tiene muchas funcionalidades. En este caso, permite generar el archivo "acsource.tr0" con la información para visualizar los resultados en un visor de forma de onda, como CosmosScope.

9. .END

Esta palabra clave debe ir en la última línea del archivo, para informar su final.

Una vez entendido el circuito, es posible simularlo y visualizar la curva de salida en CosmosScope, tal como se ha visto anteriormente. El resultado esperado se muestra en la figura 13.



Figura 13: Gráfico del comportamiento en barrido DC del circuito inversor

El gráfico muestra el diagrama de bode de la magnitud de la salida de voltaje del circuito en decibeles y de la fase de la onda de salida de voltaje (con respecto a la onda de entrada) en grados hexagesimales. Se puede ver que la magnitud es mayor en un cierto intervalo del dominio de frecuencia, es decir, el circuito está atenuando la magnitud de las componentes de frecuencia fuera del rango definido por las frecuencias de corte (160 y 320 Hertz, aproximadamente, para este circuito).

## VIII. Ejercicios propuestos

1. Hacer una simulación en régimen transiente del circuito inversor de la parte VI de esta guía. Para lo anterior, utilice una fuente de tipo pulso que se declara a continuación:

VIN IN 0 0 PULSE 0 1.8 2n .5n .5n 7n 20n

Además, agregue las siguientes sentencias a su netlist:

.TRAN 200p 20n

.PRINT TRAN V(IN) V(OUT)

Debería obtener un gráfico de entrada y salida como el que se muestra a continuación:



Figura 14: Gráfico de análisis transiente del inversor con fuente de tipo pulso

2. Hacer simulación de tipo AC a un filtro pasa bajos y a un filtro pasa altos capacitivo. Se debería obtener comportamientos como los que se muestran en las figuras 15 y 16.



Figura 15: Gráfico de análisis AC de un filtro pasa altos



Figura 16: Gráfico de análisis AC de un filtro pasa bajos

3. Implementar en HSPICE una netlist que represente el circuito conversor de voltaje alterno a continuo, que sea posible utilizar como fuente de voltaje DC de la fig. 17:



Figura 17: Esquemático de una fuente básica DC

Simular en régimen transiente el circuito, utilizando los siguientes valores: RLOAD =  $1k\Omega$ ; C = 100nF. Utilizar una fuente como la que se describe en las siguientes líneas de comando:

VIN 1 2 SIN 0 5 1meg 0

.TRAN 10n 5u

Utilizar el modelo "dmod" de diodo, para lo cual se debe agregar la siguiente sentencia:

.MODEL dmod d

Al visualizar el voltaje de Vin y RLOAD, se deberían obtener formas de onda como los de la figura:



Figura 18: Ondas de entrada y de salida de la fuente DC

## IX. Bibliografía

Para la confección de esta guía, se utilizaron los siguientes manuales:

- 1. HSPICE® User Guide: Simulation and Analysis, Version A-2008.03, March 2008
- 2. HSPICE® Reference Manual: Commands and Control Options, Version A-2008.03, March 2008
- 3. HSPICE® Reference Manual: Elements and Device Models, Version A-2008.03, March 2008
- 4. CosmosScope<sup>™</sup> User Guide, Version A-2007.12, December 2007
- 5. CosmosScope™ Reference Manual, Version A-2007.12, December 2007

La documentación anterior puede ser obtenida desde el siguiente sitio de Synopsys: <u>https://solvnet.synopsys.com/</u>. Para acceder a él, es necesario solicitar nombre de usuario y una contraseña a través del mismo sitio. Una vez dentro, se debe ingresar a la sección "Documentation", donde hay una lista de programas de Synopsys, entre los cuales se encuentran HSpice y CosmosScope, cuyos enlaces llevan directamente a páginas donjde se pueden descargar todos los manuales de cada programa.