Manual de HSPICE
Rev 1.0

Jorge Chávez

Octubre 1994

Indice

1 Introducción General

Este manual pretende ser una rápida introducción al empleo del simulador de circuitos Hspice . A pesar de ello la compatibilidad con otros simuladores que emplean la notación SPICE es bastante elevada, entre ellos podemos destacar:

- *SPICE2G.6*: simulador original.
- *Spice 3e2* y *Spice 3f2*: simuladores más desarrollados de Universidad de Berkeley
- *PSPICE*: desarrollado por MicroSim. La última versión es la 6.0

Debido a que no se pretende ser exhaustivo, se recomienda en caso de duda consultar manuales y bibliografía al respecto ([1] [2] [3] [4] [5] [6]) para una profundización o bien un tema no tratado en este manual.
Se agradecerán las sugerencias por parte del lector, que pueden ser remitidas a la dirección E-mail: `chavez@gtex02.us.es`

2 Introducción á Hspice

Hspice es un simulador de circuitos analógicos perteneciente a la casa Meta–Software, empleado en ambientes industriales. Permite simulaciones transitorias, regimen permanente y frecuenciales. Los modelos empleados son muy precisos en un amplio dominio de frecuencias que abarcan desde DC hasta frecuencias de microondas ($\simeq 100GHz$).
Este simulador está basado en el trabajo desarrollado inicialmente por la *Universidad de Berkeley* en la creación del simulador SPICE.
Posee integrado interfaces con muchos entorno de diseño microelectrónico como: *Cadence Analog Artist*, *Cadence Edge*, *Mentor MSPICE*,etc.

3 Llamada y ficheros creados

Para invocar al simulador se emplea la orden:
`csh% hspice file.sp > file.lis`
El fichero `file.sp` contiene la descripción del circuito que se desea simular, y mediante la redirección UNIX almacenamos en el fichero `file.lis` la salida por pantalla[1] del programa.
El simulador incorpora un entorno de simulación denominado `gsi`, que permite la edición, invocación, y representación gráfica de los resultados de simulación.
Las convenciones empleadas para los nombres de los ficheros empleados por Hspice es el empleo de los siguientes sufijos:

`.sp` (ó `.cir`): fichero de entrada

`.lis` (ó `.out`): fichero de resultados de simulación

`.cfg` : fichero de configuración

`.tr#` : datos de la simulación transitoria

`.sw#` : datos de la simulación DC

`.ac#` : datos de la simulación AC

`.gr#` : salida gráfica

Estos cuatro últimos ficheros son creados por el simulador cuando se invoca desde otro entorno (p.e. `gsi`).

4 Exponentes

Las cantidades en SPICE se consideran que tienen el siguiente formato:

$$\boxed{\text{number}}\ \boxed{\text{expon}}\ \boxed{\text{comment}}$$

Donde `expon` es cualquiera de los siguientes exponentes literales:

k=10^3	m=10^{-3}
meg=10^6	u=10^{-6}
g=10^9	n=10^{-9}
t=10^{12}	p=10^{-12}
	f=10^{-15}

También puede emplearse el factor:
mil=25.610^{-6}
Considerándose como comentario el resto de la cadena.

`10uAmp` : Equivale á $10\mu == 1e^{-5}$
`-0.5v` : Equivale á -0.5
`1000Farads` : Equivale á `1p`!!

[1] A pesar de la redirección, Hspice también muestra por pantalla el *stderr*, con información sobre el éxito/fracaso de la simulación así como el tiempo de CPU empleado.

5 Fichero de Entrada

El fichero de entrada se puede[2] estructurar empleando el siguiente orden:

- TITULO: Se toma como título de la simulación la primera línea del fichero
- OPCIONES
- ANALISIS a realizar
- SALIDA DESEADA
- CONDICIONES INICIALES
- FUENTES DE ESTIMULO
- NETLIST: Descripción del circuito
- LIBRERIAS Y SUBCIRCUITOS
- `.END`

Se considera como línea de comentario la zona comprendida desde el * hasta el fin de línea. Por tanto son comentarios válidos los siguientes:

```
*Este es un ejemplo de comentario
Rs 1 2 1k * Resistencia de fuente
```

Cuando una orden se desea dividir en más de una línea[3] se emplea en la primera columna de la línea de continuación el carácter +, que indica que esa línea es continuación de la anterior orden.
(Ejemplo):

```
Vs 1 0 dc 0
+ sin(0,5,10kHz)
```

[2] Aunque en otros simuladores como SPICE2G.6 esta estructura debe ser mantenida obligatoriamente

[3] Bien por claridad o debido a que al ser demasiado larga puede ser truncada por el *parser* de Hspice

6 Opciones

La línea de opciones tiene la forma:
`.OPTIONS opt1 opt1`
Pueden añadirse más de una línea de opciones, predominando la última en caso de conflicto de redefinición.
Veamos algunas de las opciones más empleadas:

- `ACCT` : impresión estadísticas
- `CPTIME=secs` : Tiempo máximo de CPU (def: 1e6 secs)
- `LIMPTS=npts` : número maximo de puntos para los comandos `PRINT/PLOT` (def=201).
- `LIST` : Listado del fichero de entrada
- `NMOD`: impresión de los parámetros del modelo.
- `NOPAGE` : Suprime los saltos de página
- `NUMDGT=ndigit` : Número de dígitos de las tablas de salida (def=4).
- `OPTS` : Imprime el valor de todas las opciones
- `POST` : Guarda en fichero la información completa del análisis, para un procesamiento posterior. Esta opción es particular de Hspice .

También se pueden definir **constantes globales**, que pueden usarse a lo largo de todo el circuito.
.PARAM var1=expr1 [var2=exp2 ...]
Pudiendo ser `expr`:

- Un valor de un parámetro
  P.e: `.PARAM F=10KHz PI=3.1415 R=1k Out=2`
- Una expresión algebráica
  P.e: `.PARAM X='Y+3'`
- Una función
  P.e: `.PARAM S(F)='1/(2*PI*F)'`

Pudiéndose posteriormente emplear dichas variables:
(Ejemplo):

```
R1 1 Out R
C1 Out 0 'S(F)/R'
```

Aparte de los operadores algebraicos elementales $(+, -, *, /)$, Hspice soporta además las funciones:

- `sin(x),cos(x),tan(x),atan(x)`
- `sinh(x),cosh(x),tanh(x)`
- `exp(x),log(x),log10(x)`
- `sqrt(x)`
- `abs(x),min(x1,x2),max(x1,x2)`

7 Tablas de Datos

Hspice permite el empleo de datos provinientes de tablas contenidas internamente ó de fichero externos:

7.1 Tabla Interna

.DATA data_name
+ pnam1 pnam2 pnam3 ...
+ pval1 pval2 pval3 ...
+ pval1 pval2 pval2 ...
.....
.ENDDATA
Para emplearla es necesario decir el análisis en el que se va a emplear.
(Ejemplo):

```
.DATA dsrc
+ TIME pv1 pv2
+ 0n 5v 0v
+ 5n 0v 5v
+ 8n 0v 5v
.ENDDATA
.TRAN DATA=dsrc
v1 1 0 PWL(TIME,pv1)
.END
```

7.2 Tabla en fichero

Existe la posibilidad de emplear ficheros con datos para crear una tabla. La tabla resultado puede ser:

1. **Tabla consecutiva**: es decir, varios ficheros (con el mismo número de columnas pero no necesariamente el mismo número de filas) constituyen una tabla con el número de columnas de cualquiera de ellos y número de filas la suma de los ficheros individuales.
   La sintaxis es:
   .DATA dataname MER
   +FILE = 'fileA' pnam1=nc1 pnam2=nc2 pnam3=nc3
   +FILE= 'fileB'
   +FILE= 'fileC'
   .ENDATA
   De esta forma creamos una tabla cuyas columnas `nc1`, `nc2` y `nc3` tienen como identificadores `pnam1`, `pnam2` y `pnam3` respectivamente.
   Consideremos los ficheros:

fileA	fileB	fileC
a10 a20 a30	b10 b20 b30	c10 c20 c30
a11 a21 a31		c11 c21 c31
a21 a22 a32		

   La tabla resultado que considerada sería:
   ```
   a10 a20 a30
   a11 a21 a31
   a21 a22 a32
   b10 b20 b30
   c10 c20 c30
   c11 c21 c31
   ```

2. **Tabla horizontal**: es decir, varios ficheros (con el mismo número de filas pero no necesariamente el mismo número de columnas) constituyen una tabla con el mismo número de filas de cualquiera de ellos y número de columnas la suma de las columnas de los ficheros individuales.
   La sintaxis es:
   .DATA dataname LAM
   +FILE = 'fileE' pnam1=nc1 pnam2=nc2 pnam3=nc3
   +FILE = 'fileF' pnam4=nc4
   +FILE = 'fileG' pnam5=nc5 pnam6=nc6
   .ENDATA
   De esta forma creamos una tabla cuyas columnas `nc1`, `nc2`, `nc3` ... tienen como identificadores `pnam1`, `pnam2`, `pnam3`, `pnam3` ... respectivamente.
   Consideremos los ficheros:

fileE	fileF	fileG
a10 a20 a30	b10	c10 c20
a11 a21 a31	b11	c11 c21
a21 a22 a32	b21	c21 c22

   La tabla resultado que considerada sería:
```
a10 a20 a30 b10 c10 c20
a11 a21 a31 b11 c11 c21
a21 a22 a32 b21 c21 c22
```

8 Tipos de Análisis

8.1 Comando .OP

.OP
Punto de operación de los elementos del circuito:

- Tensiones en todos los nudos
- Intensidades de las fuentes de tensión
- Tensiones de las fuentes de intensidad
- Punto de trabajo de elementos no lineales

8.2 Comando .DC

.DC var1 start1 stop1 inc1 [var2 start2 stop2 inc2]
Realiza un análisis estático (*sweep analysis*), es decir calcula el punto de operación del circuito para distintos valores de las fuentes. Considera que la fuente `var1` cambia desde `start1` hasta `stop1` con un incremento `inc1`. Puede especificarse una segunda fuente y dejando una fija modifica la segunda.
(Ejemplos):

```
.dc Vs 0 5 .1
.dc Vce 0 10v .5v Ib 0mA 1mA 50uA
```

8.3 Comando .AC

.AC type np fstart fstop
Realiza un análisis de pequeña señal. Esto es computa los valores en pequeña señal de las variables de salida en función de las variables de entrada.
Es necesario que al menos exista una fuente con estímulo AC para que el análisis tenga sentido. Veamos a continuación el significado de los parámetros:

`type` puede tomar uno de los valores siguientes:

- `DEC` :División en décadas (cada una con `np` puntos)
- `OCT` :División en octavas (cada una con np puntos)
- `LIN` :División en `np-1` partes iguales

`np` Número de puntos.
`fstart` frecuencia inicial.
`fstop` frecuencia final.

(Ejemplo):

```
.ac dec 10 1Hz 1GHz
```

Como una particularidad de Hspice , se puede realizar un análisis modificando: tensiones/intensidades independientes ó un parámetro de un modelo:
(Ejemplo):

```
.AC dec 10 1 10k SWEEP cload LIN 5 1pF 10pF
```

8.4 Comando .TRAN

.TRAN tplot tstop [[tstart] tstep] [UIC]
Realiza un análisis transitorio del circuito, desde el instante inicial (def: 0 segs, ó `tstart`) hasta `tstop`, imprimiendo cada `tplot` el resultado. El parámetro `tstep` acota superiormente el paso de integración.
Con la opción `UIC`, se simula con las condiciones iniciales que se hayan descrito, en vez de calcular el punto de equilibrio (`.OP`)
(Ejemplo):

```
.TRAN 1u 3m UIC
.TRAN 1mseg 2seg 1seg .1mseg
```

Como una particularidad de Hspice , se puede realizar un análisis modificando: tensiones/intensidades independientes ó un parámetro de un modelo:
(Ejemplo):

```
.TRAN 5u 4m SWEEP cload LIN 3 2p 3p
```

8.5 Comando .PZ

.PZ output_variable input_source
El análisis polo–cero permite la obtención de la función de transferencia del circuito:

$$H(s) = \frac{a_o}{b_o} \frac{(s+z_1)(s+z_2)...(s+z_m)}{(s+p_1)(s+p_2)...(s+p_n)} \tag{1}$$

Los parámetros del análisis son:

> `input_source`: fuente independiente de tensión/intensidad que se toma como excitación de entrada.
> `output_variable`: tensión ó intensidad de elemento que se toma como variable de salida.

(Ejemplos):

```
.PZ V(10) Vin
.PZ I(RL) Vs
.PZ V(2,3) Iin
```

8.6 Comando .TF

.TF output_variable input_source
Calcula las resistencias de entrada y de salida del circuito.Los parámetros del análisis son:

> `input_source`: Fuente independiente de tensión ó intensidad considerada como la excitación de entrada.
> `output_variable`: tensión ó intensidad de elemento que se toma como variable de salida.

(Ejemplos):

```
.TF V(5,3) Vin
.TF I(Vload) Vin
```

8.7 Comando .FOUR

.FOUR freq out1 [out2 out3...]
Realiza un análisis de Fourier sobre un análisis transitorio del mismo fichero. Este análisis calcula la componente DC así como las 9 primeras componentes AC (tomando como frecuencia fundamental `freq`) de las señales de salida especificadas: `ov1, ov2, ...`:

$$g(t) = \sum_{m=0}^{9} C_m\, cos(m\ t) + \sum_{m=0}^{9} D_m\, sin(m\ t) \tag{2}$$

$$C_m = \frac{1}{\pi}\int_{-\pi}^{\pi} g(t)\ cos(m\ t)\ dt \tag{3}$$

$$D_m = \frac{1}{\pi}\int_{-\pi}^{\pi} g(t)\ sin(m\ t)\ dt \tag{4}$$

$$R_m = \sqrt{C_m^2 + D_m^2} \tag{5}$$

Asimismo calcula la *distorsión armónica* (THD: Total Harmonic Distortion):

$$THD = \frac{1}{R_1}\sqrt{\sum_{m=2}^{9} R_m^2}\ 100\% \tag{6}$$

(Ejemplo):

```
.FOUR 100k v(5)
```

8.8 Comando .NOISE

.NOISE output_variable input_source [interval]
Realiza un análisis de ruido en el circuito. Debe ser empleado junto con un análisis AC.
Donde:

`output_variable`: Es la tensión del nodo considerado como variable de salida
`input_source`: Es la fuente independiente de tensión/intensidad que se toma como referencia de intrada del ruido.
`interval`: Es el intervalo de impresión. Si se emplea, en el fichero de resultados se muestra una tabla con las contribuciones de todos los nodos de salida al ruido global. Alternativamente se puede emplear el comando `.PLOT` para mostrar los resultados.

Los dispositivos generadores de ruido son las resistencias y dispositivos semiconductores. Para cada frecuencia del análisis AC, se considera para cada uno de los generadores de ruido la contribución así como la propagación hasta los nodos de salida (donde se considera una adición RMS). Se calcula la ganancia desde la fuente de entrada hasta la tensi'on de salida.
La expresión del ruido total en un nodo de salida queda:

$$out\_noise = \sum_{n=1}^{n} |Z_n\ I_n|^2 \tag{7}$$

Donde:

`I`: Fuente de corriente equivalente, causadas por: *thermal noise*, *shot noise* ó *flicker noise*.
`Z`: Transimpedancia equivalente desde la fuente de ruido y la salida.
`n`: Número de fuentes de ruido, asociadas á resistencias, MOSFET, diodos, JFET y BJT.

Las dimensiones de la fuente de entrada son: $volt/hertz^{\frac{1}{2}}$ y $amp/hertz^{\frac{1}{2}}$ (según que la misma de tensión ó intensidad, respectivamente) y de esta forma el ruido de salida siempre tiene como dimensiones: $amp/hertz^{\frac{1}{2}}$.
(Ejemplos):

```
.NOISE V(5) Vin 20
.NOISE v(4,5) Is
```

9 Generación de Resultados

Pasemos a continuación a enumerar las diferentes formas que tiene Hspice de mostrar los resultados.

9.1 Variables de Salida

Las variables de salida pueden ser:

- Tensiones nodales: es decir desde un determinado nodo a tierra.
  P.e: `V(2)`
- Tensiones de rama*: es decir la caida de tensión entre dos puntos: $V(5,8) = V(5) - V(8)$
  P.e: `V(4,3)`
- Intensidades de fuentes de tensión:
  P.e: `I(Vs)`
- Intensidades de elementos*:
  P.e: `I(Rs)`
- Para referenciar a un elemento de un subcircuito se emplea
  `nombre_subcircuito.nodo` ó `nombre_subcircuito.nombre`
  P.e.: `V(X1.3)`
- En el caso del análisis AC, debido a que la salida son vectores de variable compleja puede aplicarse:
  - `V()` ó `I()`: Valor complejo
  - `VR()` ó `IR()`: Valor real
  - `VI()` ó `II()`: Valor imaginario
  - `VM()` ó `IP()`: Magnitud
  - `VDB()` ó `IDB()`: Magnitud en Decibelios
  - `VP()` ó `IP()`: fase
- Expresión Algebraica con vectores. Empleando la misma sintaxis que se describe en la sección 6:
  P.e.: `PAR('5*sqrt(v(3))')`

(*) Opciones propias de Hspice

9.2 Formato de Salida

El número de variables que aparecen en el fichero de salida por línea depende del número de columnas. La opción `CO=num`, permite modificar el valor del número de caracteres por columna (def `C0=80`), hasta un máximo de `132` caracteres.
En caso de que el número de variables lo superara, Hspice crea una nueva tabla a continuación con el resto de variables.
Los exponentes empleados son los que aparecen en la sección 4. Hspice además emplea adicionalmente el exponente `X` como sinónimo de `MEG`.
El tipo de formato viene especificado por la opción tt INGOLD=val (def `INGOLD=0`) que puede tomar los valores:

- `INGOLD=0`: *Engineering Format*
  P.e: `1.234K,123M`
- `INGOLD=1`: *Fixed & Exponential* (formato G del lenguage C)
  P.e: `1.234e+03,0.123`

- `INGOLD=2`: *Exponential*
  P.e: `1.234e+03,1.23e-01`

El primer formato tiene la ventaja de ofrecer 2 ó 3 dígitos extra para el mismo espacio, y además es tres veces más rápido.

9.3 Comando .PRINT

.PRINT type var1 ...
Este comando crea una tabla con los valores numéricos de las variables `var1 ...` correspondientes al análisis `type` (que puede tomar como valores: `DC`, `AC`, `NOISE` ó `TRAN`).
(Ejemplos):

```
.PRINT TRAN v(4) i(Vin) PAR('v(2)-3')
.PRINT AC vm(10) vp(10) vr(2) vi(4)
.PRINT NOISE Inois
.PRINT DC v(out) i(r1) v(2,4)
```

9.4 Comando .PLOT

.PLOT type var1 ... [(lower,upper)]
Este comando crea una gráfica ascii con los valores de las variables `var1 ...` correspondientes al análisis `type` (que puede tomar como valores: `DC`, `AC`, `NOISE` ó `TRAN`).
Opcionalmente se puede añadir el intervalo de las variables de salida que se desea analizar.
(Ejemplos):

```
.PLOT TRAN V(1) v(2) (0.5v 4.6v)
```

9.5 Comando .GRAPH

.GRAPH type [MODEL=mname]variable [(lower,upper)]
Este comando crea un fichero (por omisión en formato PostScript) con la gráfica de las variables seleccionadas correspondientes al análisis `type` (que puede tomar como valores: `DC`, `AC`, `NOISE` ó `TRAN`).
Opcionalmente se puede añadir el intervalo de las variables de salida que se desea analizar.
(Ejemplos):

```
.GRAPH V(2)
```

Para modificar las escalas de la gráfica es necesario emplear la sentencia:
.MODEL mname PLOT par1=val1 [par2=val2 ...]
Donde:

`mname`: denominación del modelo
`par1=val1`: parámetro cuyo valor puede ser cualquiera de los siguientes:

- `XMIN/XMAX`: Límites horizontales de la gráfica. Si no se especifican son calculados automáticamente.
- `YMIN/YMAX`: Límites verticales de la gráfica. Si no se especifican son calculados automáticamente.
- `XSCAL/YSCAL`: Tipo de escala en el eje horizontal/vertical Puede tomar los valores:
  `1` : Escala lineal, es el modo por defecto
  `2` : Escala logarítmica
  `3` : Escala en decibelios
- `TIC=bool`: Muestra las marcas en los ejes (por omisión es igual á `1`)

- `FREQ=npts`: Muestra cada `npts` un símbolo. Permite saber la precisión de la gráfica.

(Ejemplo):

```
.MODEL ac_graf PLOT XSCAL=2
.GRAPH MODEL=ac_graf vdb(10) vp(10)
```

10 Condiciones Iniciales

.IC V(node1)=val1 [V(node2)=val2 ...]

Aunque como comentaremos más adelante es posible establecer las condiciones iniciales cuando definamos los elementos.

(Ejemplo):

```
.IC V(11)=5v V(4)=-5v
```

11 Fuentes de Estímulo

Veamos los elementos que permiten introducir estímulos en el circuito, esto es las fuentes de tensión/intensidad. Podemos distinguir:

- Fuentes independientes: Su comportamiento temporal sólo depende del tiempo.
- Fuentes dependientes: Su comportamiento temporal depende de la tensión/intensidad en otro elemento del circuito.

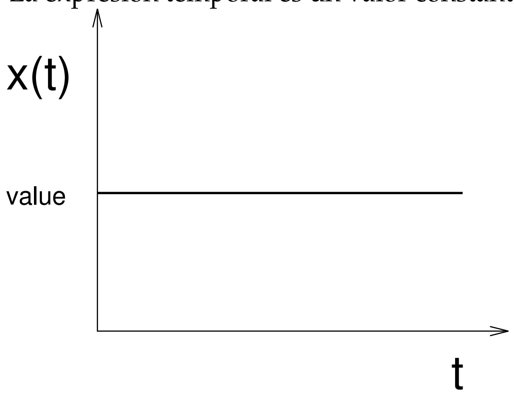
Estas fuentes pueden aplicar (según el análisis que se realize) uno ó más de los siguientes estímulos.

11.1 Diferentes estímulos

11.1.1 Estímulo DC

Es el valor considerado cuando se realiza un análisis del punto de trabajo (*Bias Point*).
DC value

Si no se expecifica se considera que las fuentes tienen un nivel de continua nulo (`DC=0`).
La expresión temporal es un valor constante `value` a lo largo del tiempo.



11.1.2 Estimulo AC

Es el valor considerado cuando se realiza un análisis en pequeña señal del cicuito.
AC mag phase
Producirá un valor senoidal de pequeña señal de amplitud `mag`, fase `phase` en grados.
Si no se especifica se considera que las fuentes tienen una pequeña señal de amplitud nula y fase nula.
Al ser un análisis de pequeña señal, se consideran los modelos linealizados de los dispositivos no lineales en el punto de polarización, siendo los valores de tensiones e intensidades calculados las amplitudes de las senoides. Por tanto no es de extrañar que los valores de las mismas puedan sobrepasar el valor de la alimentación, debido a que en el modelo de pequeña señal es bien sabido (!!!) que las fuentes de alimentación DC son cortocircuitos.
Si sólo se emplea una fuente de tensión (intensidad) no tiene utilidad emplear un estímulo distinto de:
`AC 1 0`
La utilidad de ambos parámetros aparece cuando existe más de una fuente de excitación y existe una diferencia en la magnitud ó en la fase de ambas y se desea analizar su comportamiento.
Una ventaja que tiene emplear la sentencia anterior es que de esta forma la medida realizada en cualquier nodo es directamente la ganancia.

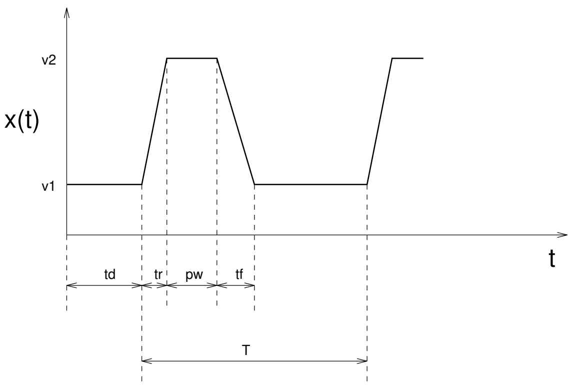
11.1.3 Estímulo PULSE

Es un posible estímulo que se considera en un análisis transitorio.
La forma de onda generada es un tren de pulsos. También permite modelar un escalón.
PULSE v1 v2 td tr tf pw T
Donde:

- `v1` y `v2`: valores entre los que cambia la señal
- `td`: Retraso del primer cambio (*delay*)
- `tr`: Tiempo de subida (*raise*)
- `tf`: Tiempo de bajada (*fall*)
- `pw`: Duración del pulso (*pulse width*)
- `T`: Periodo de la señal

La expresión de la señal es:

Tiempo	**Valor**
0	v1
td	v1
td+tr	v2
td+tr+pw	v2
td+tr+pw+tf	v1



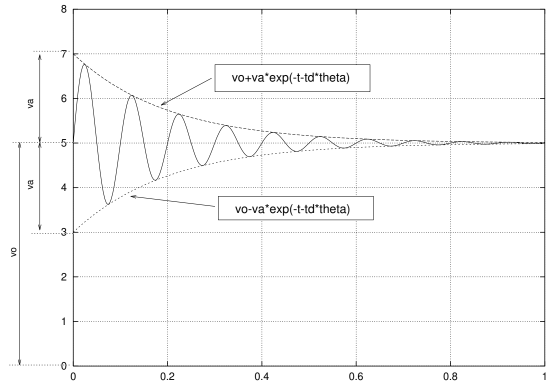
11.1.4 Estímulo SIN

Es un posible estímulo que se considera en un análisis transitorio.
La forma de onda generada es una senoide que puede estar amortiguada con el tiempo.
SIN v0 va freq td theta phi
Produce una senoide amortiguada donde:

- `v0`: Valor de Continua
- `va`: Amplitud de la senoide no atenuada
- `freq`: Frecuencia de la senoide
- `td`: Retraso de la señal
- `theta`: Factor de amortiguación (*damping factor*)
- `phi`: Desfase en grados

La expresión de la señal es:

Tiempo	**Valor**
$0 < t < td$	$vo + va\ sin\left(\frac{2\pi phi}{360}\right)$
$td < t$	$vo + va\ exp(-(t - td\ theta))\ sin\left(2\ \pi\left(freq\ (t - td) + \frac{phi}{360}\right)\right)$



En caso de que se desee una senoide sin amortiguación basta emplear `theta=0`, o bien no añadirlo (su valor por defecto es 0).

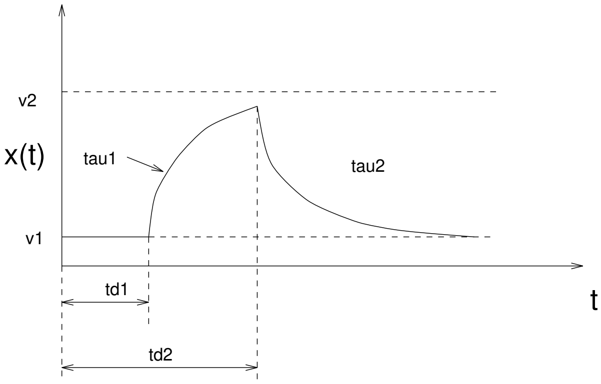
11.1.5 Estímulo EXP

Es un posible estímulo que se considera en un análisis transitorio.
La forma de onda generada es una exponencial de subida y otra de bajada.
EXP v1 v2 td1 tau1 [td2 tau2]
Permite definir una exponencial (o sucesión de dos exponenciales) cuyos parámetros son:

`v1`: tensión inicial y segundo valor asintótico
`v2`: primer valor asintótico
`td1`: retraso de la primera exponencial
`tau1`: constante de tiempo de la primera exponencial
`td2`: retraso de la segunda exponencial
`tau2`: constante de tiempo de la segunda exponencial

La expresión de la señal es:

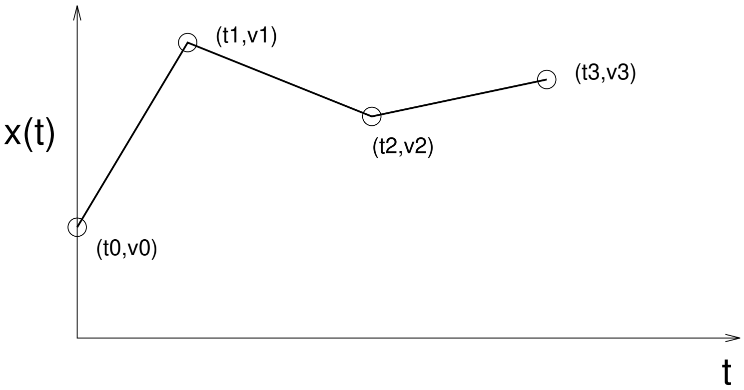
Tiempo	**Valor**
$0 < t < td1$	$v1$
$td1 < t < td2$	$v1 + (v2 - v1)\left(1 - exp\left(-\frac{t-td1}{tau1}\right)\right)$
$td2 < t$	$v1 + (v2 - v1)\left(1 - exp\left(-\frac{t-td1}{tau1}\right)\right) + (v1 - v2)\left(1 - exp\left(-\frac{t-td2}{tau2}\right)\right)$



11.1.6 Estímulo PWL

Es un posible estímulo que se considera en un análisis transitorio.
La forma de onda generada es un función lineal a trozos.
PWL t1 v1 [t2 v2 t3 v3 ...][R=repeat][TD=delay]
Permite definir una función lineal, especificando pares (v_k, t_k) por donde se desea pase la curva, interpolando linealmente entre ellos. Los parámetros empleados son:

`tk, vk ...`: valores de tensión en el tiempo especificado
`repeat`: tiempo a partir del cual la onda se repite
`delay`: retraso para comenzar



Existe la alternativa de definirla empleando una tabla:
`PWL (time_label,volt_label)`
Donde la tabla se define con la sentencia `DATA` comentada en la sección 7.1, siendo válido el ejemplo que figura allí.

11.1.7 Estímulo SFFM

Es un posible estímulo que se considera en un análisis transitorio.
La forma de onda generada es una señal modulada en frecuencia única (*Single Frecuency FM*).
SFFM co ca [fc mdi fs]
Donde:

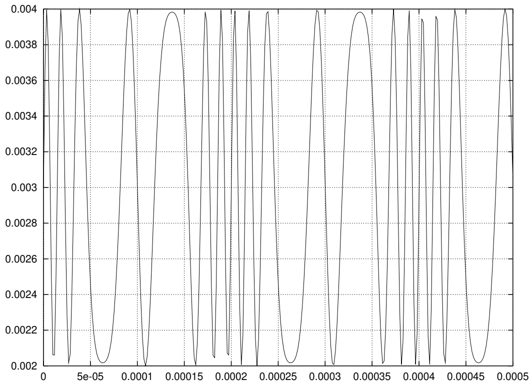
`vo`: Valor de Offset
`va`: Amplitud
`fc`: Frecuencia de la portadora en Hz
`mdi`: Indice de modulación
`fs`: Frecuencia de la señal en Hz

La expresión sería:

$$vo + va\ sin\,(2\ \pi\ \ fc\,t + mdi\ \ sin\,(2\ \pi\ \ fs\ t))$$

(Ejemplo):

```
SFFM(3M,1M,20K,10,5k)
```



11.1.8 Estímulo AM

Es un posible estímulo que se considera en un análisis transitorio.
La forma de onda generada es una señal modulada en amplitud (*Amplitude Modulation*).
AM sa oc fm fc td
Donde:

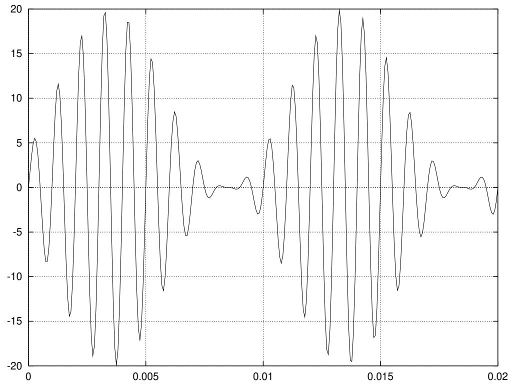
- `sa`: Amplitud de la señal
- `oc`: Constante de offset
- `fm`: Frecuencia de Modulación
- `fc`: Frecuencia de la Portadora
- `td`: Retraso de la señal

La expresión sería:

$$sa\ \ (oc + sin\,(2\ \pi\ fm(t - td)))\ \ sin\,(2\ \pi\ fc\ (t - td))$$

(Ejemplo):

```
AM(10 1 100 1K 1M)
```



11.2 Tipos de Fuentes

11.2.1 Fuente Independiente de Tensión

V<name> <node+> <node-> [[DC=]dcval][AC=acmag,acphase][tranfun]
Donde:

`[DC=] dcval`: Es el valor de continua de la fuente en un análisis DC. Si aparece un valor y no se especifica a qué estímulo se debe, se considera DC.
(Ejemplo):

```
V1 10 12 5
```

Es una fuente de tensión con offset de `5v` entre los nodos `10` y `12`.
`AC=acmag,acphase`: son los parámetros de la fuente para un análisis de pequeña señal
`tranfun`: Es la función que describe el comportamiento transitorio de la fuente (AM, DC, EXP, PULSE, PWL, SFFM, SIN)

(Ejemplo):

```
Vx 1 0 ac 1 0
Vb 2 0 dc=10v
Vin 13 2 1m ac 1 sin(0 1 1meg)
```

11.2.2 Fuente Independiente de Intensidad

I<name> <node+> <node-> [[DC=]dcval][AC=acmag,acphase][tranfun]
El significado de los parámetros es igual que en el caso anterior:
(Ejemplo):

```
Ix 3 1 10uAmp
Ib 2 0 dc=5m
Iin 13 2 1m pulse(0,5,0,1n,1n,0.5m,1m)
```

11.2.3 Polinomiales

Las fuentes dependientes pueden poseer una dependencia lineal con respecto a la variable de control:
$Xdep(t) = gain\ Xind(t)$
Pero también puede expresarse una dependencia polinomial. Hspice permite que esta dependencia sea respecto a una, dos ó tres variables independientes:

1. Función unidimensional: POLY(1) p0 p1 ...
   La expresión de la variable dependiente en función de la independiente `X` es:
   $p0 + p1\ X + p2\ X^2 + p3\ X^3 + ...$
   (Nota): Si la polinomial es de orden 1, y sólo aparece un coeficiente Hspice supone que es `p1`.
2. Función bidimensional: POLY(2) p0 p1 ...
   La expresión de la variable dependiente en función de las independientes `X`, `Y` es:
   $p0 + p1\ X + p2\ Y + p3\ X^2 + p4\ X\ Y + p5\ Y^2 + p6\ X^3 + p7\ X^2\ Y + p8\ X\ Y^2 + p9\ Y^3 + ...$
3. Función tridimensional: POLY(3) p0 p1 ...
   La expresión de la variable dependiente en función de las independientes `X`, `Y`, `Z` es:
   $p0 + p1\ X + p2\ Y + p3\ Z + p4\ X^2 + p5\ X\ Y + p6\ X\ Z + p7\ Y^2 + p8\ Y\ Z + p9\ Z^2 + p10\ X^3 + ...$

11.2.4 Fuente de Tensión dependiente de Intensidad

Caso lineal:
H<name> <node+> <node-> <cvs> gain
+ [MAX=maxval][MIN=minval]
Caso Polinomial:
H<name> <node+> <node-> POLY(ndim) <cvs1> [<cvs2> <cvs3>]
+ [MAX=maxval][MIN=minval] val1 val2 ...
Permite definir una CCVS (*Current Controlled Voltage Source*). Fuente dependiente cuya tensión depende de la intensidad que circule por la fuente de tensión `cvs` (`control_voltage_source`).
La ganancia `gain` tiene dimensiones de `volt/Amp`.
Los parámetros opcionales `MAX=maxval` y `MIN=minval` permite saturar la señal de salida si ésta alcanza dichos valores.
(Ejemplo):

```
Hsense 1 2 Vsense 10
Hamp 13 0 POLY(1) Vin 1 2k 10
```

En este último caso la expresión de la tensión de la fuente `Hamp` sería:
$V_{Hamp}(t) = 1 + 2e3 * I_{Vin} + 10 * I_{Vin}^2$

11.2.5 Fuente de Tensión dependiente de Tensión

Caso Lineal:
E<name> <node+> <node-> <c+> <c->
+ gain [MAX=maxval][MIN=minval]
Caso Polinomial:
E<name> <node+> <node-> POLY(ndim) <c1+> <c1-> [<c2+> <c2-> <c3+> <c3->]
+ gain [MAX=maxval][MIN=minval] val1 val2 ...
Permite definir una VCVS (*Voltage Controlled Voltage Source*). Fuente dependiente cuya tensión depende de la tensión que exista entre los nodos `c+` y `c-`.
La ganancia `gain` no tiene dimensiones, `volt/volt`.
(Ejemplo):

```
Ebuff 1 2 10 11 1
Ebuff 10 0 1 2 1000 MAX=5 MIN=0
```

En este último caso la tensión $V_{(10,0)}$ depende proporcionalmente de la tensión $V_{(1,2)}$, con una ganancia 1000, no permitiéndose que la primera ($V_{(10,0)}$) supere el valor de 5v ni disminuya por debajo de 0v, saturando en dicho caso.

11.2.6 Fuente de Intensidad dependiente de Intensidad

Caso Lineal:
F<name> <node+> <node-> <cvs>
+ gain [MAX=maxval][MIN=minval]
Caso Polinomial:
F<name> <node+> <node-> POLY(ndim) <cvs1> [<cvs2> <cvs3>]
+ gain [MAX=maxval][MIN=minval] val1 val2...
Permite definir una CCCS (*Current Controlled Current Source*). Fuente dependiente cuya intensidad depende de la intensidad que circule por la fuente de tensión `cvs` (`control_voltage_source`).
La ganancia `gain` no tiene dimensiones, `Amp/Amp`.
(Ejemplo):

```
Fsense 1 2 Vsense 10
Famp 13 0 POLY(1) 500
```

11.2.7 Fuente de Intensidad dependiente de Tensión

Caso Lineal:
G<name> <node+> <node-> <control+> <control->
gain + [MAX=maxval][MIN=minval]
Caso Polinomial:
G<name> <node+> <node-> POLY(ndim) <c1+> <c1-> [<c2+> <c2-> <c3+> <c3->]
gain [MAX=maxval][MIN=minval] val1 val2 ...
Permite definir una VCCS (*Voltage Controlled Current Source*). Fuente dependiente cuya intensidad depende de la tensión que exista entre los nodos `c+` y `c-`.
La ganancia `gain` tiene dimensiones `Amp/volt`.
(Ejemplo):

```
Gbuff 1 2 10 11 1.0 Gnonlin 100 101 POLY(2) 3 0 4 0 0.0 13.6m 0.2m
0.005m
```

En este último caso la Intensidad `Gnonlin` tiene una expresión:
$I_{Gnonlin} = 0.0 + 13.6e - 3 * V_{(3,0)} + 0.2e - 3 * V_{(4,0)} + 0.005e - 3 * V_{(3,0)}^2$

12 Elementos

12.1 Comando .MODEL

Permite definir una serie de parámetros del modelo analítico del dispositivo, que permite ser referenciado por un elemento. La sintaxis empleada para definir un modelo es:
.MODEL <model_name> type_name([par1=val1 par2=val2 ..])
Donde:

`model_name`: es el nombre que permitirá posteriormente referenciar a dicho modelo.
`type_name`: es el identificador del dispositivo que se esté modelando. Lo identificadores válidos son:

RES	resistor model
CAP	capacitor model
D	diode model
NMOS	N-channel MOSFET model
PMOS	P-channel MOSFET model
NPN	NPN BJT model
PNP	PNP BJT model
URC	Uniform Distributed RC model
NJF	N-channel JFET model
PJF	P-channel JFET model
NMF	N-channel MESFET model
PMF	P-channel MESFET model
SW	voltage controlled switch
CSW	current controlled switch
LTRA	Uniform RLC/RC/LC/RG transmission line model

12.2 CONDENSADOR

C<name> <node+> <node-> [model_name] <cvalue> [IC=<vo_value>]
Siendo `cvalue` la capacidad del condensador en faradios. Opcionalmente se puede incluir la condición inicial en tensiones `vo_value`[4] así como el modelo del condensador `model_name`. Los parámetros del modelo del condensador se muestran en la tabla [1]. Siendo la fórmula empleada para computar la capacidad en dicho caso:
`CAP=CJx(LENGTH-NARROW)x(WIDTH-NARROW)+2xCJSWx(LENGTH+WIDTH-2*NARROW)`
(Ejemplos):

```
Cload 15 0 20pF
C2 1 2 .2e-12 IC=1.5volts
```

Ident.	Parameter	Units
CJ	junction bottom capacitance	F/meters2
CJSW	junction sidewall capacitance	F/meters
DEFW	default device width	meters
GNARROW	narrowing due to side etching	meters

Tabla 1: Parámetros del modelo del condensador

[4]Recordar añadir en el análisis `TRAN` la sentencia `UIC` para que emplee dichas condiciones iniciales

12.3 DIODO

D<name> <anodo> <catodo> <model> [area]

La intensidad directa circula desde `anodo` á `cátodo`.
Siendo `model` el identificador del modelo empleado, los parámetros del modelo se muestran en la tabla [2]. Opcionalmente se puede añadir el `area`, que escala `IS`, `RS`, `CJO` y `IBV`.

Ident.	Parameter	Units
IS	saturation current	amp
N	emission coefficient	
RS	parasitic resistance	ohm
CJO	zero-bias p-n capacitance	farad
VJ	p-n potential	volt
M	p-n grading coefficient	
FC	forward-bias depletion cap. coef.	
TT	transit time	sec
BV	reverse breakdown voltage	volt
IBV	reverse breakdown current	amp
EG	bandgap voltage	eV
XTI	saturation current temperature exponent	
KF	flicker noise coef.	
AF	flicker noise exponent.	

Tabla 2: Parámetros del modelo del diodo

(Ejemplos):

```
Dsignal 14 0 D1N4148
* Modelo de diodo ideal
.model Dideal D
* Modelo elemental de diodo de se nal
.model D1N4148 D(Is=0.1p Rs=16 Cjo=2p Tt=12n Bv=100 Ibv=0.1p)
* Modelo elemental de diodo Zener con Bv=4.7v
Dzener 1 3 D1N750
.model D1N750 D(Is=880.5E-18 Rs=.25 Cjo=175p Vj=.75 Bv=4.7 Ibv=20.245m)
```

12.4 Inductancia

L<name> <node+> <node-> <Lvalue> [IC=io_value]

Siendo `lvalue` la inductancia de la bobina. Opcionamente se puede incluir la condición inicial en intensidades `io_value`.
(Ejemplos):

```
Lload 15 0mH
Lsense 5 12 20mHenry ic=2mA
```

12.5 MOSFET

M<name> <dnode> <gnode> <snode> <bnode> <model>
+ [L=length][[W=width]
+ [AD=d_area][AS=s_area]
+ [PD=d_perim][PS=s_perim]

Donde:

> `model`: es el modelo empleado. Los parámetros del modelo del MOSFET aparecen en la tabla [3].
> Existen cuatro modelos fundamentales:

LEVEL=1 Shichman-Hodges

LEVEL=2 MOS2 (descrito en [[1]])

LEVEL=3 MOS3, un modelo semi-empirico (descrito en [[1]])

LEVEL=4 BSIM (descrito en [[2]])

> `length, width`: son las dimensiones geométricas del canal del MOSFET.
> `d_area, s_area`: son las dimensiones geométricas del area de difusión de drenador y fuente respectivamente.
> `p_area, p_area`: son las dimensiones geométricas del perímetro de difusión de drenador y fuente respectivamente.

(Ejemplos):

```
M1 14 2 13 0 Pmos W=12u L=4u
```

Ident.	Parameter	Units
LEVEL	model index	-
VTO	zero-bias threshold voltage	V
KP	transconductance parameter	A/V**2
GAMMA	bulk threshold parameter	V**0.5
PHI	surface potential	V
LAMBDA	channel-length modulation (MOS1 and MOS2 only)	1/V
RD	drain ohmic resistance	Ohm
RS	source ohmic resistance	Ohm
CBD	zero-bias B-D junction capacitance	F
CBS	zero-bias B-S junction capacitance	F
IS	bulk junction saturation current	A
PB	bulk junction potential	V
CGSO	gate-source overlap capacitance per meter channel width	F/m
CGDO	gate-drain overlap capacitance per meter channel width	F/m
CGBO	gate-bulk overlap capacitance per meter channel length	F/m
RSH	drain and source diffusion sheet resistance	Ohm/sq.
CJ	zero-bias bulk junction bottom cap. per sq-meter of junction area	F/m**2
MJ	bulk junction bottom grading coef.	-
CJSW	zero-bias bulk junction sidewall cap. per meter of junction perimeter	F/m
MJSW	bulk junction sidewall grading coef.	-
JS	bulk junction saturation current per sq-meter of junction area	A/m**2
TOX	oxide thickness	meter
NSUB	substrate doping	1/cm**3
NSS	surface state density	1/cm**2
NFS	fast surface state density	1/cm**2
TPG	type of gate material:	-
XJ	metallurgical junction depth	meter
LD	lateral diffusion	meter
UO	surface mobility	cm**2/V-s
UCRIT	critical field for mobility degradation (MOS2 only)	V/cm
UEXP	critical field exponent in mobility degradation (MOS2 only)	-
UTRA	transverse field coef (mobility) (deleted for MOS2)	-
VMAX	maximum drift velocity of carriers	m/s
NEFF	total channel charge (fixed and mobile) coefficient (MOS2 only)	-
KF	flicker noise coefficient	-
AF	flicker noise exponent	-
FC	coefficient for forward-bias depletion capacitance formula	-
DELTA	width effect on threshold voltage (MOS2 and MOS3)	-
THETA	mobility modulation (MOS3 only)	1/V
ETA	static feedback (MOS3 only)	-
KAPPA	saturation field factor (MOS3 only)	-

Tabla 3: Parámetros del modelo del MOSFET niveles 1 á 3

12.6 RESISTENCIA

R<name> <node+> <node-> [<model_name>] <rvalue>

Donde:

`model_name`: es el identificador del modelo de resistencia empleado. En la tabla [4], se muestran los parámetros del modelo de la resistencia que es posible emplear.
La ecuación que permite con estos valores calcular la resistencia nominal es:
`R=RSHxW-NARROW`
La influencia de la temperatura en el valor de la resistencia viene dado por:
`RES(temp)=RES(tnom)x(1+TC1x(temp-tnom)+TC2*(temp-tnom)2)` `rvalue`: es el valor de la resistencia en ohmios

(Ejemplos):

```
R1 2 3 100k
```

Ident.	Parameter	Units
TC1	first order temperature coeff.	Ohm/C
TC2	second order temperature coeff.	Ohm/C2
RSH	sheet resistance	Ohm/[]
DEFW	default width	meters
NARROW	narrowing due to side etching	meters

Tabla 4: Parámetros del modelo resistivo

12.7 BJT

Q<name> <cnode> <bnode> <enode> [<snode>] <model> [area_value]

Los parámetros del modelo del transistor BJT aparecen en la tabla [5].
(Ejemplos):

```
* Modelo ideal de npn con beta=100
.model Qideal NPN(bf=100)
Qnom 10 1 3 q2n2222
.model Q2N2222A NPN(Is=14.34f Xti=3 Eg=1.11 Vaf=74.03 Bf=255.9 Ne=1.307
+ Ise=14.34f Ikf=.2847 Xtb=1.5 Br=6.092 Nc=2 Isc=0 Ikr=0 Rc=1
+ Cjc=7.306p Mjc=.3416 Vjc=.75 Fc=.5 Cje=22.01p Mje=.377 Vje=.75
+ Tr=46.91n Tf=411.1p Itf=.6 Vtf=1.7 Xtf=3 Rb=10)
```

Id.	Parameter	units
IS	transport saturation current	A
BF	ideal maximum forward beta	-
NF	forward current emission coefficient	-
VAF	forward Early voltage	V
IKF	corner for forward beta high current roll-off	A
ISE	B-E leakage saturation current	A
NE	B-E leakage emission coefficient	-
BR	ideal maximum reverse beta	-
NR	reverse current emission coefficient	-
VAR	reverse Early voltage	V
IKR	corner for reverse beta high current roll-off	A
ISC	B-C leakage saturation current	A
NC	B-C leakage emission coefficient	-
RB	zero bias base resistance	Ohms
IRB	current where base resistance falls halfway to its min value	A
RBM	minimum base resistance at high currents	Ohms
RE	emitter resistance	Ohms
RC	collector resistance	Ohms
CJE	B-E zero-bias depletion capacitance	F
VJE	B-E built-in potential	V
MJE	B-E junction exponential factor	-
TF	ideal forward transit time	sec
XTF	coefficient for bias dependence of TF	-
VTF	voltage describing VBC dependence of TF	V
ITF	high-current parameter for effect on TF	A
PTF	excess phase at freq=1.0/(TF*2PI) Hz	deg
CJC	B-C zero-bias depletion capacitance	F
VJC	B-C built-in potential	V
MJC	B-C junction exponential factor	-
XCJC	fraction of B-C depletion capacitance connected to internal base node	-
TR	ideal reverse transit time	sec
CJS	zero-bias collector-substrate capacitance	F
VJS	substrate junction built-in potential	V
MJS	substrate junction exponential factor	-
XTB	forward and reverse beta temperature exponent	-
EG	energy gap for temperature effect on IS	eV
XTI	temperature exponent for effect on IS	-
KF	flicker-noise coefficient	-
AF	flicker-noise exponent	-
FC	coefficient for forward-bias depletion capacitance formula	-

Tabla 5: Parámetros del modelo *Gummel-Poon* del transistor BJT

13 Subcircuitos

Spice permite la creación de bloques contituidos por elementos que pueden ser incorporados como bloques primitivos, esto es permite el empleo de estructuras jerárquicas en la construcción del *Netlist*.

13.1 Comando .SUBCKT

La forma general para la definición de un subcircuito es:
.SUBCKT sub_name n1 n2 .. [par1=val1] ...
<elements>
.ENDS [sub_name]
Donde:

`sub_name`: es el nombre del subcircuito
`n1 n2 ...`: son los identificadores **locales** de los nodos del subcircuito.
`par1=val1 ...`: son las asignaciones de parámetros **locales** del subcircuito.

Es opcional emplear el subcircuito con la sentencia `.ENDS`, y permite al `parser` de Hspice comprobar la correcta definición.
Es posible que un subcircuito contenga a otro, aunque no a sí mismo!!!
(Ejemplo):

```
.subckt rc_net 1 2 R=1k S=1M
r1 1 2 R
c1 2 0 'S/R'
.ends
```

13.2 Llamada a un Subcircuito

Para invocar a un subcircuito definido:
X<name> n1 n2 ... <sub_name> [par1=val1...]
Donde:

`X<name>`: es el identificador del elemento
`n1 n2 ...`: son los nodos externos (evidentemente no tienen por qué coincidir con los locales del subcircuito, aunque tienen que ser iguales **en cantidad**.
`sub_name`: identificador del subcircuito empleado.
`par1=val1 ...`: asignaciones de parámetros válidas unicamente en el subcircuito invocado. Reemplazan a los que tuviera internamente el subcircuito.

(Ejemplo):

```
X1 100 101 rc_net
```

14 Librerías

Para la inclusión de modelos de elementos previamente definidos (librería) se pueden emplear dos comandos:

14.1 Comando .INC

.INC 'filename'

Donde `filename` es el nombre de un fichero (con su *PATH* si es necesario). Este fichero será incluido por el preprocesador en la posición indicada y puede tener cualquier tipo sentencias Hspice . A su vez éste puede invocar a otros ficheros de forma recursiva (excepto a sí mismo!!!).
(Ejemplo):

.INC es2.cmos

14.1.1 Comando .LIB

.LIB filename entry

Es análogo a `.INC`, pero permite seleccionar los modelos del fichero `filename` que desean incorporarse.
(Ejemplo):

.LIB 'modelos' q2n2222a

(NOTA) El nombre del fichero en ambos casos debe aparecer entre comillas simples para que se respeten las mayúsculas/minúsculas del mismo. Si no se hace esto el nombre del fichero que se busca será el indicado pero convertido a minúsculas.

Indice Alfabético

Bibliografía

[1] A. Vladimirescu and S. Liu, "The simulation of mos integrated circuits using spice2." ERL Memo No. ERLM80/7,Electronics Research Laboratory, University of California, Berkeley, Oct. 1980.

[2] D. L. S. B. J. Sheu and P. K. Ko, "Spice2 implementation of bsim." ERL Memo No. ERL M85/42,Electronics Research Laboratory, University of California, Berkeley, May 1985.

[3] J. R. Pierret, "A mos parameter extraction program for the bsim model." ERL Memo Nos. ERL M84/99 and M84/100, Electronics Research Laboratory, University of California,Berkeley, Nov. 1984.

[4] META-Software, *HSPICE User's Manual. H9001*. Meta-Software, Inc, 1300 White Oaks Road, Campbell, CA 95008, 1990.

[5] P. Antognetti and G. Massobrio, *Semiconductor Device Modeling with SPICE*. McGraw-Hill, 1988.

[6] P. A. R. L. Geiger and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*. McGraw-Hill, 1990.