

SPICE

PROGRAMACION en SPICE. Fichero *.CIR	2
1. INTRODUCCION	2
2. ELEMENTOS: DESCRIPCION Y MODELOS	4
2a. Resistencias	5
2b. Condensadores e Inductores	6
2c. Subcircuito	7
2ch. Fuentes Independientes	8
2d. Fuentes Dependientess	9
2e Otros elementos	10
2f. Diodos	11
2g. Transistor Bipolar	12
2g. JFET	13
2h. MOSFET	13
3. LAS LIBRERIAS	15
4. TIPOS DE ANALISIS	15
5. SENTENCIAS UTILES	17
6. SALIDA DE RESULTADOS	19
7. GRAFICADOR PROBE	19
8. EJEMPLOS	21
 SPICE-DOS	 25
1. INSTALACION Y USO	25
2. CONFIGURACION	25
3. Entrega de Resultados	26
 SPICE-WINDOWS	 26
1. INSTALACION	26
2. CREACION DEL CIRCUITO: "SCHEMATICS"	27
3. GRAFICADOR PROBE	28
4. Otras Cuestiones	29

Bibliografía útil:

- "Spice. A guide to circuit simulation et analysis using spice" P.W.Tuinenga Prentice-Hall 1995.
- "Semiconductor device modeling with Spice" G.Massobrio P.Antognetti McGraw-Hill 1993
- "Simulación Electrónica con PSpice" J.D.Aguilar A.Domenech J.Garrido RA-MA
- "Guía básica de PSpice 5.0" Manual_PSpice.pdf J.D.Aguilar

La versión PSpice 5.1 de DOS, está muy bien, tú dominas el programa, no al revés. Pero es una versión de 16 bits, y por tanto funciona en sistemas operativos de 16 y 32 bits, pero no en Windows 64 bits, ya que Microsoft decidió aniquilar los programas antiguos impidiendo que los programas buenos, efectivos y ya pagados funcionasen en sus nuevos sistemas operativos. Es algo que se está extendiendo desde el poder: poner caducidad a los teléfonos, coches, dinero, personas, para obligar a pagar y someterse a lo "autorizado" desde arriba.

PROGRAMACION en SPICE. Fichero *.CIR

1. INTRODUCCION

En este documento, hablaremos de "Spice", pero refiriéndonos a "PSpice". PSpice es una implementación para PC, derivada de la versión Spice-2. Existen más versiones derivadas de Spice-2, como Spice-3 o HSpice. Según dicen los entendidos, no existen muchas diferencias de fondo entre Spice-3 y Spice-2, puesto que comparten el núcleo de cálculos matemáticos y circuitales. HSpice es una versión profesional para entorno industrial.

El programa SPICE simula el funcionamiento de un circuito utilizando las ecuaciones de corriente de Kirchhoff. A cada nudo se le asigna un número único como identificación (entero y positivo); el nudo cero siempre debe corresponder a tierra. Los nodos también podrían nombrarse con una cadena de texto, pero eso complica las referencias. Todos los nodos deben tener un camino de continua hacia tierra, y ésta debe existir necesariamente.

- Si se está en la versión DOS, se crea manualmente el fichero de programa del tipo *.cir.
- Si se usa la versión windows, el primer paso es construir el circuito visualmente. Este fichero se guarda en un fichero tipo *.sch. Con *.sch se generará el programa de Spice que se graba en un fichero *.cir.

Tras la simulación, aparecerán los resultados en un fichero de tipo texto, con la extensión ".out". En el fichero *.out, también se encuentra una indicación de dónde pueden estar los errores. De haberlos, se marca la posición con "----\$". Otro fichero de resultados (*.dat), pero en formato binario, contiene la información para que el programa graficador "PROBE" pueda representar los resultados de forma gráfica.

El programa SPICE (*.cir) siempre empieza con un título que se interpreta como un comentario. La última línea del fichero debe ser ".END" (no puede haber líneas en blanco tras ".end"). Entre estas dos sentencias debe estar, sin importar el orden:

- La descripción del circuito (los elementos con sus nodos).
 - La definición de dispositivos no elementales (.MODEL, .LIB).
 - El tipo de análisis (.AC, .DC, .TRAN, .OP, .TEMP etc.).
 - Las variables que se desean calcular o mostrar (.PRINT, .PLOT, .PROBE).
 - Comentarios
- Los comandos o sentencias de control, comienzan por ".". El resto serían elementos del circuito, o son ignoradas (comentarios, título, líneas en blanco)
 - Las líneas de comentarios, deben empezar por "!", pero si se quieren añadir comentarios a una sentencia se utiliza ";". Si una línea es muy larga (más de 80 caracteres), la línea de continuación debe empezar por "+". Deben utilizarse caracteres tipo ASCII (código menor que 128).
 - Spice siempre es indiferente a mayúsculas y minúsculas.
 - Los distintos campos se separan por: uno o más blancos/espacios, comas, signo "=" o paréntesis. Todos ellos son equivalentes a blancos/espacios.

El uso de esos separadores depende del usuario, y por tanto se verá en los *.cir distintas costumbres.

- Todos los nodos deben tener un camino de continua hacia tierra. Además deben tener al menos dos conexiones, salvo algunos casos de líneas de transmisión, y el sustrato de un mosfet.
- Los nodos pueden nombrarse también como una cadena alfanumérica, siempre que no empiecen por un número, por ejemplo: N002, Ncolector, Nsalida. Pero cuando se usa esta nomenclatura, en Probe se usa V(N002), pero en .PRINT/.PLOT hay que usar V([N002]) lo cual suele inducir a errores.

- En Spice, las intensidades entrantes a un elemento siempre son positivas, salvo en el caso de elementos de dos terminales, que son positivas si entran por el nodo positivo (n+), y salen por el negativo (n-).

Formato Numérico

Los valores pueden ser enteros (2, -35), en coma flotante (3.16, -6.5) o en formato exponencial (-1.2E-4 , 2.34E11). Recordar que los ingleses utilizan un punto en lugar de la coma decimal usada en el resto de países.

Factores de escala:

k = 1E3	(kilo)	Meg = 1E6	(Mega)	G = 1E9	(Giga)	T = 1E12	(Tera)
m = 1E-3	(mili)	u = 1E-6	(micro)	n = 1E-9	(nano)	p = 1E-12	(pico)
f = 1E-15	(femto)						

p.ej. 2.3K equivale a 2300 , 450m equivale a 0,450

- Tras los factores de escala, se ignora cualquier caracter inmediatamente seguido
- Las letras que van inmediatamente tras un número, y que no sean factores de escala se ignoran. También se ignoran si van inmediatamente tras un factor de escala. Por tanto 10V y 10Voltios es 10, y 1nFaradio es 1n. Además, "10V" será 10 voltios, si es una magnitud de tensión, pero será 10Ω si la cadena se asigna a una magnitud de resistencia.
- Recuerde que Spice siempre es indiferente a mayúsculas y minúsculas. Aquí usaremos la distinción para facilitar la lectura.
- Un condensador de 0.01F no es de 0,01 faradios, sino de 0,01*10⁻¹⁵ faradios.
- Spice siempre usa las fases en grados sexagesimales. Además siempre hace los cálculos considerando la temperatura nominal, que está definida por defecto en 27°C. La temperatura nominal se puede cambiar con la sentencia ".OPTIONS". Pero como PROBE tiene la capacidad de derivar, hay que recordar que la derivada de sinusoidales se hace respecto al ángulo en radianes.
- Se admiten "parámetros", que actúan como variables globales, y por tanto, válidas también dentro de un subcircuito. Más adelante se verá la capacidad de hacer barridos sobre fuentes de tensión/intensidad independientes, o sobre los parámetros de los dispositivos (resistencias, diodos, JFET, BJT, etc.). Pues bien, spice también puede hacer barridos sobre los "parámetros" que funcionan como variables globales. Las llamadas a los valores de los parámetros se hacen entre llaves, y dentro de ellas se pueden hacer operaciones matemáticas. Las funciones admitidas son:

```
+ - * /      abs() sqrt() exp() log() log10()
sin() cos() tan() asin() acos() atan() arctan() (siempre en radianes)
limit(x,min,max) si x>max => max, si x<min => min, otro caso => x
pwr(x,y)=|x|^y    pwrs(x,y)=(signo-x)|x|^y    table(x, x1,y1, x2,y2, etc.)
```

Table, permite interpolar el resultado de x con la linea quebrada definida por los pares x_i,y_i.

Ejemplo simple: R3 valdrá 3kΩ, y R4 valdrá 16kΩ.:

```
R3 2 1 {valor8}
R4 3 0 {10k+2*valor8}
.PARAM valor8=3K pi2={20*3.14159/10}
```

- Se pueden crear funciones matemáticas personalizadas que actúen sobre los parámetros. Por ejemplo, en la siguiente sentencia se crea la funcion ff2(x,y,z)=x*(100y+z):

```
.FUNC ff2(x,y,z) { x * (100*y + z) }
```

En el siguiente ejemplo, se hace un barrido desde 1kΩ hasta 9kΩ en la resistencia R₂:

```
R2 2 0 {variable1} ; si variable1=20k R1=20K
.PARAM variable1=20K
.DC LIN PARAM variable1 1k 9k 1k ; de 1kΩ a 9kΩ
```

2. ELEMENTOS: DESCRIPCION Y MODELOS

Las líneas que describen los elementos empiezan por una letra fija (R para resistencias) y un máximo de 7 caracteres cualesquiera (xxxxxxx), luego vienen los números de nodos (n+ y n-) y al final el valor del elemento o el modelo de dispositivo.

Elementos:

Tipo:	Formato:	nodos:				valor/multiplicador:
Resistencia	Rxxxxxxx	n+	n-	[*]		valor (Ohmios)
Condensador	Cxxxxxxx	n+	n-	[*]		valor (Faradios) [[IC=]vv]
Inducción	Lxxxxxxx	n+	n-	[*]		valor (Henrios) [[IC=]ii]
Diodo	Dxxxxxxx	n+	n-			nombre de modelo [área]
BJT	Qxxxxxxx	nc	nb	ne	[ns]	nombre de modelo [área]
JFET	Jxxxxxxx	nd	ng	ns		nombre de modelo [área]
MOSFET	Mxxxxxxx	nd	ng	ns	nb	nombre de modelo [M=área] [x]

- En la resistencia, condensador e inducción se puede incluir un nombre de modelo, en la posición marcada con [*]. Si se omite, se usa el que modelo por defecto.
- En el diodo, n+ y n- son los nodos del lado P y del lado N.
- En el BJT, nc, nb y ne, se refieren a los nodos de Colector, Base y Emisor.
- En los FETs, nd, ng, ns y nb, se refieren a los nodos de Drenador, puerta (Gate), fuente (Source) y sustrato (Bulk o Body).

Si en los dispositivos semiconductores se añade un valor de [área] se considera como un aumento de área del dispositivo (como si se conectasen varios en paralelo). El [área] es equivalente a un "multiplicador" de dispositivo. En el Mosfet, para simular a varios Mosfet en paralelo, se añade un "multiplicador" con una M=, además se pueden añadir algunos parámetros adicionales para distinguir distintos subtipos de Mosfet que comparten el mismo modelo (en la posición marcada con [x]).

De los siguientes elementos, el más útil es el subcircuito. El subcircuito permite definir bloques circuitales, que pueden ser usados en distintos sitios del circuito principal, sin necesidad de repetir el subcircuito. Los nodos n1, n2, n3, etc. permiten el contacto eléctrico con el circuito principal.

Subcircuito	Xxxxxxxx	n1	n2	n3...	nombre-modelo	
FET de GaAs (canal-N)	Bxxxxxxx	nd	ng	ns	nombre de modelo [área]	
Línea de Transmisión	Txxxxxxx	n1	n2	n3 n4	z0=xx	td=xx
Acoplamiento-Inductores	Kxxxxxxx	L1	L2	[k=valor-acoplamiento 0<k≤1]		
Conmutador controlado por tensión	Sxxxxxxx	n+	n-	nc+	nc-	nombre-modelo
Conmutador contr. por intensidad	Wxxxxxxx	n+	n-	vnom	nombre-modelo	

MODELOS de DISPOSITIVOS

No todas las resistencias son iguales. Hay resistencias que apenas dependen de la temperatura, otras con coeficiente positivo, y otras con coeficiente negativo. Igualmente no todos los condensadores son iguales. Algunos tienen una capacidad muy fija, en otros depende de la temperatura, o del voltaje aplicado, y en estos casos, puede depender de forma lineal o parabólica. No todos los diodos son iguales, hay diodos Schottky, de Silicio y diversos LEDs. No todos los transistores son iguales, los hay con baja ganancia pero que manejan gran potencia, y al revés. Por ello, Spice permite definir **modelos** con la etiqueta .MODEL. De esta forma, en un circuito pueden coexistir distintos diodos, condensadores, transistores y subcircuitos. Pero aun existiendo distintos modelos de diodos, transistores, etc. sí existen diferencias fundamentales entre diodos y transistores, o entre resistencias y condensadores. Por tanto cada familia de modelos, se agrupa en un *tipo de modelo*: "RES" para resistencias, "D" para diodos, "NPN" para BJTs de tipo npn, o "PJF" para JFET de canal P.

La sentencia .MODEL define las características de un dispositivo concreto. El nombre que se dará al nuevo dispositivo es *nombre-modelo*, y *tipo-modelo* es uno de los tipos predefinidos. Cada *tipo_de_modelo* tiene unos parámetros *pnom1*, *pnom2*, etc., a los que hay que dar valores

concretos *pval1*, *pval2*, etc. Estos valores sólo se darían en el caso de que los valores predefinidos por spice no nos interesen.

```
.MODEL nombre-modelo tipo-modelo ( pnom1=pval1 pnom2=pval2 ...)
```

Spice considera los caracteres () = , como blancos. Se suelen incluir para facilitar la lectura.

Tipos de modelo posibles:

RES	para Resistencia (R)
CAP	para condensador (C)
IND	para inductor (L)
D	para diodo (D)
NPN	para BJT de tipo npn (Q)
PNP	para BJT de tipo pnp (Q)
LPNP	para BJT de tipo pnp lateral (Q)
NJF	para JFET de canal n (J)
PJF	para JFET de canal p (J)
NMOS	para MOSFET de canal n (M)
PMOS	para MOSFET de canal p (M)
CORE	para acoplamiento magnético (K)
VSWITCH	para conmutador por tensión (S)
ISWITCH	para conmutador por corriente (W)
GASFET	para MESFET-N de GaAs (B)

2a. Resistencias

Si no se desea un modelo específico de resistencia, la definición es muy simple. En el ejemplo, R_{OUT} , entre los nodos 3 y 4, vale 7kΩ. La intensidad $I(R_{OUT})$ es positiva si va del nodo 3 al 4.

```
Rout 3 4 7K
R23 7 5 4K TC=0.001 0.0002
R44 8 6 5K TCE=0.01
```

Tras la definición de una resistencia se pueden añadir los coeficientes de temperatura para modelar la dependencia con la temperatura. Recordemos, que el "=" puede ser reemplazado por un espacio. Además, si se deseara, podrían rodearse los coeficientes de temperatura con paréntesis, ya que Spice los considera equivalentes a espacios.

R_{23} está entre los nodos 7 y 5, tiene un valor de 4kΩ a la temperatura nominal (si no se cambia, Spice toma la temperatura nominal como 27°C). Además, R_{23} tiene unos coeficientes de temperatura $TC1=0,001$ (lineal) y $TC2=0,0002$ (cuadrático). Cuando se activan los coeficientes $TC1$ y $TC2$, la dependencia con la temperatura quedaría:

$$R(T) = \langle \text{valor} \rangle \times R_0 \times [1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2]$$

R_{44} vale 5kΩ y tiene una dependencia exponencial con la temperatura. Cuando se activa el coeficiente TCE , la dependencia con la temperatura quedaría:

$$R(T) = \langle \text{valor} \rangle \times R_0 \times \left[1.01^{TCE \cdot (T - T_{nom})} \right]$$

Donde R_0 es el valor básico asignado en la definición del modelo de resistencia. Si no se coge ningún modelo, el valor básico es de 1Ω. Ese valor básico, siempre es multiplicado por el $\langle \text{valor} \rangle$ que se indica al final de la línea que invoca a la resistencia.

```
R55 2 0 Rmia 3 ; R55=6K
R77 2 1 Rmia 10 ; R77=20K
.MODEL Rmia RES ( R=2K TC1=0.01 TC2=0.005 )
```

En el anterior ejemplo se han definido dos resistencias R_{55} y R_{77} particulares basadas en un modelo que el usuario ha llamado "Rmia". Como el modelo es para una resistencia se usa la palabra clave "RES" que indica que el tipo de modelo es una resistencia. Aunque es costumbre que el nombre elegido para el modelo("Rmia") empiece por la letra (R) para que recuerde que es una resistencia, no es obligatorio. Cuando se llame los elementos, se debe tener en cuenta su valor básico. Como se puede observar en R_{55} su valor de "3", multiplica al valor básico definido en el modelo "Rmia", que es 2kΩ.

Algunos de los parámetros básicos de una resistencia, junto con su nombre, y valor predefinido:

Parámetro:	Nombre	Valor
Valor básico de la resistencia	R	1 Ω
Coeficiente lineal de temperatura	TC1	0 °C ⁻¹
Coeficiente cuadrático de temperatura	TC2	0 °C ⁻²
Coeficiente exponencial de temperatura	TCE	0 °C ⁻¹

2b. Condensadores e Inductores

El uso de condensadores e inductores es muy similar, por eso nos centramos en condensadores, y serán evidentes los cambios que habría que hacer para inductores. No se pueden poner dos condensadores en serie. No se pueden poner dos inductores en paralelo.

El condensador C₁₁ entre los nodos 8 y 9 tiene una capacidad de 10 μ F.

```
C11 8 9 10uF ; 10uF = 10 micro faradios
C33 5 4 2nF 1V ; la tensión inicial del condensador de 2nF es 1V
C33 5 4 2nF IC=1 ; caso idéntico al anterior
L55 8 9 330uF ; inductor de 330 micro faradios
L77 9 6 500uH 0.1m ; corriente inicial de 0,1mA en el inductor de 500 $\mu$ H
```

Tras la definición de un condensador o inductor, se puede añadir el valor inicial. Si es un condensador, sería el valor inicial de tensión en el condensador. Si es una inducción, sería el valor inicial de corriente. Para que se apliquen las anteriores condiciones iniciales, en .TRAN debe añadirse la opción UIC.

Los condensadores e inductores, pueden depender lineal o parabólicamente de la tensión aplicada (condensadores) o de la intensidad que circula (inductores). Por tanto, en los modelos, aparecen coeficientes para modelar tales dependencias. Además podría haber también dependencia parabólica con la temperatura. El modelo "CAP" para condensador podría tener una dependencia con la tensión y temperatura como:

$$C(V,T) = \langle \text{valor} \rangle \times C_0 \times [1 + VC1 \cdot V + VC2 \cdot V^2] \times [1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2]$$

Donde C₀ es el valor básico asignado en la definición del modelo de condensador (si no se modifica, es 1 faradio). Ese valor básico, siempre es multiplicado por el <valor> que se indica al final de la línea que invoca al condensador. En un inductor no existen VC1 ni VC2, sino IL1 e IL2 que modelan la dependencia de un inductor L(I,T) de forma análoga a la mostrada con el condensador C(V,T).

```
C77 3 2 Cmio 7 ; C77 = 7nF
L88 9 1 Lmio 22 ; L88 = 220uH
.MODEL Cmio CAP ( C=1nF VC1=0.02 VC2=0.001 )
.MODEL Lmio IND ( L=10uH IL1=0.02 IC2=0.001 TC1=0.001 )
```

En el anterior ejemplo se han definido un condensador C₇₇ y un inductor L₈₈ particulares basados en modelos que el usuario ha llamado "Cmio" y "Lmio". Aunque es costumbre que el nombre elegido empiece por "C" para condensadores y "L" para inductores, no es necesario, pero es conveniente para evitar errores por despiste. Cuando se llame los elementos, se debe tener en cuenta su valor básico. Como se puede observar en C₇₇ su valor de "7", multiplica al valor básico definido en el modelo "Cmio", que es 1nF. Además C₇₇ es un condensador cuya capacidad depende parabólicamente de la tensión aplicada. El inductor L₈₈ también depende parabólicamente de la intensidad que circula a su través (coeficientes IL1 e IL2), pero además depende linealmente con la temperatura (coeficiente TC1).

Algunos de los parámetros básicos de un condensador, junto con su nombre, y valor predefinido:

Parámetro:	Nombre	Valor
Valor básico del condensador	C	1 F
Coeficiente lineal de tensión	VC1	0 V ⁻¹
Coeficiente cuadrático de tensión	VC2	0 V ⁻²
Coeficiente lineal de temperatura	TC1	0 °C ⁻¹
Coeficiente cuadrático de temperatura	TC2	0 °C ⁻²

Se pueden usar condensadores e inductores no lineales, que dependen de la tensión e intensidad aplicada, respectivamente (IC son las condiciones iniciales):

```
Cxxxxxx  n+  n-  POLY  C0  C1  C2  ...  [[IC=]vv]
Lxxxxxx  n+  n-  POLY  L0  L1  L2  ...  [[IC=]ii]
```

$$C(V) = C0 + C1 \times V + C2 \times V^2 + \dots ; \quad L(I) = L0 + L1 \times I + L2 \times I^2 + \dots$$

2c. Subcircuito

Es posible descomponer circuitos complejos en asociaciones de circuitos más simples o subcircuitos. El nodo 0, que corresponde a tierra es global, y por tanto es común al circuito principal y a todos los subcircuitos. Los nodos definidos en un subcircuito son locales, sólo son accesibles en ese subcircuito. Aún así, para evitar confusiones, es recomendable que no compartan nombres de nodo con otros subcircuitos o con el circuito principal.

LLamada a un subcircuito:

```
Xnombre n1 n2 n3  modelo_subcircuito
```

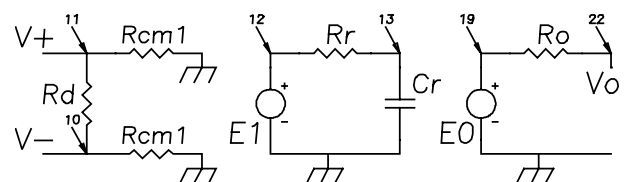
Definición de un subcircuito:

```
.SUBCKT modelo_subcircuito n1 n2 n3 ...
[definición del subcircuito]
.ENDS
```

En este ejemplo, mostramos cómo utilizar los subcircuitos para crear un modelo simplificado de amplificador operacional (no depende de la alimentación, y por tanto no se satura). Este modelo simple se muestra en la figura, y su código mostrado aquí, está en la librería FTC.lib. Luego se ve la aplicación de ese amplificador operacional simplificado en un programa (etapa de ganancia 20). En ese mismo programa, hay una etapa de ganancia 30 que usa el subcircuito de amplificador operacional de EVAL2.lib proporcionado por PSpice-5.1.

*** Subcircuito simple que simula un operacional *** (definido en FTC.lib)

```
*      patas:      +      -      salida
.subckt ampop 11 10 22
Rd 11 10 5Meg
Rcm1 10 0 500Meg
Rcm2 11 0 500Meg
E1 12 0 11 10 1
Rr 12 13 100k
Cr 13 0 100nF
E0 19 0 13 0 200k
Ro 19 22 75
.ends ampop
```



subcircuito ampop de FTC.lib

E1 es una fuente de tensión dependiente de (V(11)-V(10)), es decir de (V+ – V-). Esa etapa proporciona un comportamiento en frecuencia similar al de un amplificador operacional compensado como el 741. R_D y R_{cm} , son las resistencias de entrada diferencial y en modo común. R_R y C_R fijan la frecuencia de corte del amplificador compensado. E0 es una fuente de tensión dependiente de V(13), y es la etapa que da ganancia (200.000), y tiene una resistencia de salida (R_o) de 75Ω. Este operacional, no depende de la alimentación, y por tanto no se satura nunca.

*** Programa de dos etapas amplificadoras con ganancias de 20 y 30.

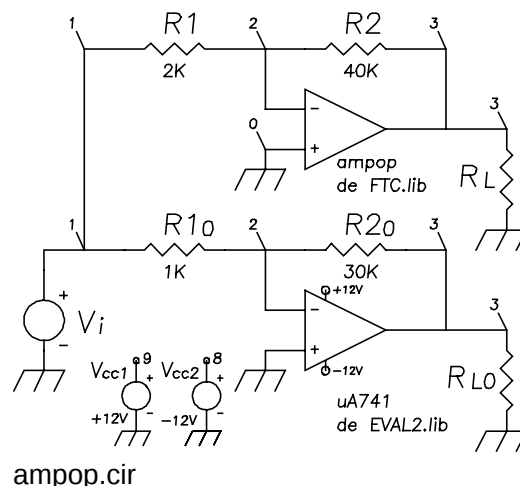
* Usa dos subcircuitos de amplificador operacional: ampop y ua741

```
Vcc1 9 0 +12V ; +Vcc Alimentacion (+)
Vcc2 8 0 -12V ; -Vcc Alimentacion (-)
Vi 1 0 AC 1 ; Señal de entrada: módulo=1 fase=0°
* 1ª Etapa: Ganancia 20 que usa ampop de FTC.lib
R1 1 2 2k
R2 2 3 40k
RL 3 0 10k
xA20 0 2 3 ampop ; patas: + - salida
```

```

* 2ª Etapa: Ganancia 30 y usa uA741 de EVAL2.lib
R10 1 4 1k
R20 4 5 30k
RL0 5 0 10k
XA30 0 4 9 8 5 uA741 ;patas + - +Vcc -Vcc salida
.AC DEC 8 1 10MegHz ; Analisis AC desde
* [ 1Hz a 10MHz (8 pt/dec)
.PRINT AC Vdb(3) Vdb(5) Vp(5) ; Salida (dB)
* [ de V(3) y V(5). Fase v(5)
.lib FTC.lib ; libreria con ampop
.lib EVAL2.lib ; libreria con el uA741
.options nomod nopage
.probe
.end

```



2ch. Fuentes Independientes

Hay fuentes independientes de tensión y de intensidad. Dentro de cada tipo, pueden ser de continua, es decir constantes en el tiempo (DC), de alterna sinusoidal especificada como fasor (módulo-fase) (AC) y dependiente del tiempo en el que se especifica la dependencia temporal (que se usará en el análisis TRAN)

```

Tension      Vxxxxxxx n+ n- [[DC] valorDC] [AC valorAC fase] [ValorTemporal]
Intensidad   Ixxxxxxx n+ n- [[DC] valorDC] [AC valorAC fase] [ValorTemporal]

```

Los tres tipos de etiquetas pueden coexistir, y dependiendo del tipo de análisis (.DC, .AC y .TRAN) se activará un grupo u otro de etiquetas.

La etiqueta DC, puede omitirse, e indica que la fuente de continua tiene un valor igual a "valorDC".

La etiqueta AC, indica una fuente de alterna especificada como fasor, de amplitud (módulo-magnitud) igual a "valorAC" y un desfase en grados de "fase". Si se omite "valorAC", se toma 1V. Si se omite la fase, se supone fase 0°.

```

Vcc 8 0 DC 6 ; fuente de tensión continua de 6V entre los nodos 8 y tierra
Vcc 8 0 6 ; igual a la anterior
Vii 2 1 AC 1 ; fuente de tensión alterna de amplitud 1V y fase 0°
Vii 2 1 AC 1 0 ; igual a la anterior

```

La dependencia específica con el tiempo de la fuente de tensión/intensidad, se define con "ValorTemporal". Se puede usar dependencia temporal con forma sinusoidal "SIN", cuadrada "PULSE", lineal a tramos "PWL", pulso-exponencial "EXP" o modulada en frecuencia "SFFM":

SIN . Sinusoidal:

```

SIN( offset amplitud frecuencia tiempo-retardo amortiguamiento fase)

```

La onda senoidal es 0V hasta que se llega a tiempo-retardo. El amortiguamiento (s) es exponencial:

$$\text{SIN}(t) = \text{offset} + \text{amplitud} + \text{sen}[2\pi \text{frecuencia}(t - \text{retardo}) + \text{fase}] \times e^{-(t - \text{retardo}) \cdot \text{amortiguamiento}} \quad \text{para } t > (t - \text{retardo})$$

Ejemplo de senoidal con amplitud 2V y frecuencia 1 kHz y fase 45°:

```

Vin 3 0 SIN( 0 2V 1kHz 0 0 45)

```

PULSE . Onda Cuadrada:

```

PULSE( V1 V2 t-cambio t-12 t-21 ancho-pulso-2 período)

```

La onda cuadrada comienza en V1 hasta que se llega a t-cambio; en ese momento pasa a V2, y continúa en V2 durante "ancho-pulso-2". Los tiempos de subida y de bajada se especifican con t-12 (tiempo que tarda en cambiar de V1 a V2) y t-21 (tiempo que tarda en cambiar de V2 a V1). En V1 permanece un tiempo igual al periodo menos la suma de el tiempo en V2 y los tiempos de cambio (t-12 y t-21).

Ejemplo de cuadrada entre $V1=-2V$ y $V2=4V$ con periodo 3s. Comienza en $V1=-2V$. En $t=0,4s$ comienza a subir a $V2=4V$ y tarda 0,3 segundos. Permanece 1s en $V2=4V$, y entonces tarda 0,7s en bajar a $-2V$:

```
Vcua 7 4 PULSE( -2 4 0.4 0.3 0.7 1s 3s)
```

PWL . Onda Quebrada:

```
PWL( t1 V1 t2 V2 ... tn Vn )
```

La onda quebrada se especifica por los pares de valores t_i, V_i que la conforman. Al acabar el último par t_i, V_i , la señal continúa en el último valor de V_i . Siempre inicia en el primer V_i aunque esté antes del primer tiempo t_i .

Recorrido de (t_i, V_i) en PWL: $(-\infty 1)(0 1)(0,5 2)(1 2)(1,5 1)(2 1,5)(4 0,5)(5 2,5)(\infty 2,5)$:

```
Vp 1 0 PWL( 0s 1V 0.5s 2V 1s 2V 1.5s 1V 2s 1.5V 4s 0.5 5s 2.5)
```

EXP . Exponencial:

```
EXP( V1 V2 t-exp-1 cte-1 t-exp-2 cte-2)
```

La señal comienza en $V1$ hasta llegar a la activación de la primera exponencial (de $V1$ a $V2$) en $t\text{-exp-1}$ con constante de tiempo $cte-1$. Al llegar a $t\text{-exp-2}$ se activa una segunda exponencial (de $V2$ a $V1$) con constante de tiempo $cte-2$ que se suma a los términos anteriores.

La siguiente señal empieza en 2V. En 0,3s se suma una exponencial con constante de tiempo 1s y amplitud $(V2-V1)=(4V-2V)$ (en este caso, sería una subida). Al llegar a 1s, se suma una segunda exponencial con constante de tiempo 0,2s, y amplitud $(V1-V2)=(2V-4V)$ (aquí sería una bajada):

```
V3 3 0 EXP( 2 4 0.3s 1.0s 1s .2s )
```

SFFM . Frecuencia Modulada:

```
SFFM( offset amplitud frecuencia-portadora índice-modulación frec-señal)
```

Genera una señal de tipo FM (frecuencia modulada). Las señales AM se pueden conseguir con fuentes dependientes (se verá un ejemplo).

Las tres formas de fuentes independientes (*DC AC ValorTemporal*) pueden aparecer combinadas en una misma fuente independiente. El tipo de análisis que posteriormente se haga será el que active cada forma definida en la fuente. DC se activa con ".DC", AC se activa con ".AC" y ValorTemporal se activa con ".TRAN".

2d. Fuentes Dependientess

Son fuentes de tensión o de intensidad controladas por una tensión o intensidad en otra parte del circuito.

Tensión	cont. por tensión	Exxxxxxx	n+	n-	nc+	nc-	valor (V)
Intensidad	cont. por tensión	Gxxxxxxx	n+	n-	nc+	nc-	valor (A)
Tensión	cont. por intensidad	Hxxxxxxx	n+	n-	vnom	valor (V)	
Intensidad	cont. por intensidad	Fxxxxxxx	n+	n-	vnom	valor (A)	

En las fuentes dependientes de tensión (Exxx y Gxxx) $nc+$ y $nc-$ son los nodos controladores, es decir, los nodos de los cuales depende la fuente con nodos $n+$ y $n-$.

Las fuentes Hxxx y Fxxx, dependen de la intensidad que circula por una rama. Para marcar dicha rama, se incluye en esa rama una fuente de tensión de valor 0V. El nombre dado a esa fuente sería el dado en la posición "vnom", y por tanto Hxxx y Fxxx dependen de la intensidad que pasa por la fuente de "vnom" (usualmente 0V).

En Spice es usual insertar fuentes de tensión de valor 0V para utilizarlas como amperímetros, basta pedir la intensidad que circula por la fuente de tensión.

- Spice admite fuentes dependientes no lineales, polinómicas de una o más dimensiones. Las siguientes fuentes son de tensión entre los nodos 6 y 0. E_{11} depende de $V_x=V(9,0)=V(9)$. E_{22} además depende de $V_y=V(8,0)=V(8)$. E_{33} además depende de $V_z=V(7,0)=V(7)$:

$$\begin{aligned}
&E_{11} \ 6 \ 0 \ \text{POLY}(1) \ 9 \ 0 \ (\ 5 \ 4 \ 3 \ 2 \ 1 \) \\
&E_{11} \ 6 \ 0 \ 9 \ 0 \ (\ 5 \ 4 \ 3 \ 2 \ 1 \) \\
&E_{11} = 5 + 4 \cdot V_x + 3 \cdot V_x^2 + 2 \cdot V_x^3 + 1 \cdot V_x^4 \\
&E_{22} \ 6 \ 0 \ \text{POLY}(2) \ 9 \ 0 \ 8 \ 0 \ (\ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \) \\
&E_{22} = 1 + 2 \cdot V_x + 3 \cdot V_y + 4 \cdot V_x^2 + 5 \cdot V_x V_y + 6 \cdot V_y^2 \\
&E_{33} \ 6 \ 0 \ \text{POLY}(3) \ 9 \ 0 \ 8 \ 0 \ 7 \ 0 \ (\ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \) \\
&E_{33} = 1 + 2 \cdot V_x + 3 \cdot V_y + 4 \cdot V_z + 5 \cdot V_x^2 + 6 \cdot V_x V_y + 7 \cdot V_x V_z + 8 \cdot V_y^2 + 9 \cdot V_y V_z + 10 \cdot V_z^2
\end{aligned}$$

Los ejemplos son autoexplicativos. Los paréntesis no son necesarios, pero facilitan la lectura. Si la fuente dependiente tiene una sola dimensión, se puede eliminar *POLY(1)*. La siguiente es una señal de amplitud modulada. La portadora es $V_x = V(9,0)$ y la moduladora es $V_y = V(8,0)$:

$$\begin{aligned}
&E_{AM} \ 6 \ 0 \ \text{POLY}(2) \ 9 \ 0 \ 8 \ 0 \ (\ 0 \ 1 \ 0 \ 0 \ 0.5 \ 0 \) \\
&E_{AM} = V_x + 0.5 \cdot V_x V_y = V_x \cdot (1 + 0.5 V_y)
\end{aligned}$$

Las fuentes dependientes polinómicas de una sola dimensión sirven para modelar resistencias no lineales. Bastaría usar que la variable controladora sea igual a la controlada.

$$\begin{aligned}
&G_{44} \ 6 \ 0 \ \text{POLY}(1) \ 9 \ 0 \ (\ 1 \ 2 \ 3 \) \\
&I(G_{44}) = 1 + 2 \cdot V_x + 3 \cdot V_x^2 \\
&G_{55} \ 9 \ 0 \ \text{POLY}(1) \ 9 \ 0 \ (\ 0 \ 0 \ 3 \) \\
&I(G_{55}) = 0 + 0 \cdot V_x + 3 \cdot V_x^2
\end{aligned}$$

La fuente de intensidad G_{44} depende de V_x de forma polinómica, y G_{55} simula un dispositivo con una intensidad que depende cuadráticamente de la tensión V_x en sus extremos.

$$\begin{aligned}
&F_{66} \ 6 \ 0 \ \text{POLY}(1) \ V_1 \ (\ 4 \ 5 \ 6 \) \\
&I(F_{66}) = 4 + 5 \cdot I(V_1) + 6 \cdot I(V_1)^2 \\
&H_{77} \ 6 \ 0 \ \text{POLY}(2) \ V_1 \ V_2 \ (\ 0 \ 0 \ 0 \ 3 \ 5 \ 7 \) \\
&V_{77} = V(6,0) = H_{77} = 3 \cdot I(V_1)^2 + 5 \cdot I(V_1) \cdot I(V_2) + 7 \cdot I(V_2)^2
\end{aligned}$$

2e Otros elementos

Acoplamiento-Inductores:

Kxxxxxxx L1 L2 [k=valor-acoplamiento 0<k≤1]

Acoplamiento entre los inductores L1 y L2, con un coeficiente de acoplamiento $k=0,89$ ($0 < k < 1$)

L1 2 0 220uH ; inductor de 220 μ H
 L2 3 0 330uH ; inductor de 330 μ H
 KL1conL2 L1 L2 0.89 ; Acoplamiento 0,89 (entre 0 y 1) de L1 con L2

Conmutadores controlados:

Sxxxxxxx n+ n- nc+ nc- nombre-modelo ; controlado por tensión $V(nc+, nc-)$
 Wxxxxxxx n+ n- vnom nombre-modelo ; controlado por la intensidad en vnom

Los conmutadores se pueden usar como resistencias controladas por tensión/intensidad, ya que dependiendo de los valores de conmutación $V_{ON}-V_{OFF}$ ($I_{ON}-I_{OFF}$) cambia de resistencia ($R_{ON}-R_{OFF}$).

.MODEL Smio Vswitch (2ohm 2Megohm 7V 2V)

El conmutador definido como Smio, enciende a 7V ($R_{ON}=2\Omega$) y se apaga a 2V ($R_{OFF}=2M\Omega$)

Línea de Transmisión:

Txxxxxxx n1 n2 n3 n4 z0=xx td=xx
 Txxxxxxx n1 n2 n3 n4 z0=xx f=xx nl=xx

Terminales de la puerta 1: n1 n2. De puerta 2: n3 n4. Impedancia característica: z_0 . Longitud de la línea de transmisión, bien con el tiempo de retardo (T_D) o con la combinación de la frecuencia (F) y la longitud eléctrica normalizada de la línea a esa frecuencia (NL).

2f. Diodos

Obviamente, aquí no trataremos a fondo los parámetros o modelos de dispositivos semiconductores (para ello, ver el libro de Massobrio). Además, se pondrá la parte más básica de los modelos de PSpice, que pueden ser muy distintos a los modelos de Spice, o a los de otras versiones de Spice.

Para invocar a un diodo, debe darse inicialmente un nombre *Dxxxxxxx* que comience por "D", luego siguen los nodos del ánodo *n+* (lado P) y del cátodo *n-* (lado N). Sigue el nombre de modelo de diodo, y luego el factor de área, que indica el número de dispositivos en paralelo equivalente.

Dxxxxxxx n+ n- nombre de modelo [área] [OFF] [IC=Vd]

Si se omite el *área*, se supone que es 1, es decir, que sólo hay un diodo, y no hay varios en paralelo. Si se añade la opción *OFF* el proceso cálculo de la solución comienza suponiendo al diodo en corte. Si se añade la condición inicial *IC=0.5* implica que en el análisis transitorio se comenzará con una tensión en el diodo de 0,5V (usar UIC en .TRAN).

```
D11 3 2 Dmio
D22 9 8 Dmio 2.0
.MODEL Dmio D ( Is=1E-10 N=1.5 Rs=5 Ibv=1E-3 BV=5.6V Eg=1.11)
```

En el anterior ejemplo, usamos dos diodos del tipo "Dmio". *D₁₁* está entre los nodos 3 y 2, y *D₂₂* entre los nodos 9 y 8. *D₂₂* equivale a dos diodos *D₁₁* en paralelo. Ambos diodos comparten el tipo de diodo, que se define con la sentencia .MODEL. Dmio tiene una corriente inversa de saturación de *I_S=1E-10A*, un coeficiente de emisión $\eta=1,5$, una resistencia serie de 5Ω. Además es de tipo zener, pues tiene una tensión de ruptura *V_B=BV=5,6V* con una corriente de *I_{BV}=1E-3A*. Por defecto *E_G* es 1,11eV, que corresponde a silicio. Aquí se recuerda que es de silicio, puesto que es el valor por defecto, pero podría haberse puesto 0,67eV para diodos de germanio, o 0,69eV para diodos schottky.

Algunos parámetros básicos de un diodo, junto con su nombre, valor predefinido, y un ejemplo:

Parámetro:	Nombre	Valor	Ejemplo
Corriente de saturación	IS	1E-14 A	1.0E-13A
Coeficiente de emisión (η)	N	1	2.0
Resistencia óhmica serie	RS	0 Ohm	10 Ohm
Tensión de ruptura inversa	BV	infinito V	5.6 V
Corriente de ruptura	IBV	1E-3 A	9E-3 A
I saturación recombinación	ISR	0 A	1.0E-15A
Coef. emisión recombinac. (η_R)	NR	2	2.0
I umbral alta inyección	IKF	0 A	40 mA
Ancho banda prohibida <i>E_G</i>	EG	1.11 eV	.67 eV
Exponente de temperatura	XTI	3.0	3.0
Capacidad de depleción a 0V	CJ0	0 F	2 pF
Tensión de la unión PN	VJ	1 V	0.35V
Coeficiente de graduación	M	0.5	0.5
Tiempo de tránsito	TT	0 s	0.1ns

Ecuaciones:

Basado en la ecuación de Shockley modificada, tendríamos tres componentes. *I_{FWD}* es la propia corriente de la ecuación de Shockley. *I_{REC}* es la corriente de generación-recombinación. *I_{BV}* es la corriente inversa al iniciarse la ruptura en la tensión fijada por BV. *K_{FWD}* tiene en cuenta los efectos de alta inyección si se fija un *I_{KF}≤0*. A este modelo se le puede añadir una resistencia serie *R_S*.

$$I_d = \langle \text{área} \rangle \times (K_{FWD} \cdot I_{FWD} + K_{REC} \cdot I_{REC} - I_{REV})$$

$$I_{FWD} = I_S \left(e^{\frac{V_d}{nV_T}} - 1 \right) ; I_{REC} = I_{SR} \left(e^{\frac{V_d}{n_R V_T}} - 1 \right) ; I_{REV} = I_{BV} e^{-\frac{(V_d + BV)}{V_T}}$$

$$K_{FWD} = \sqrt{\frac{I_{KF}}{I_{KF} + I_{FWD}}} \quad \text{si } I_{KF} > 0 \quad ; \quad K_{FWD} = 1 \quad \text{si } I_{KF} \leq 0$$

$$K_{REC} = \left[\left(1 - \frac{V_d}{V_j} \right)^2 + 0,005 \right]^{\frac{M}{2}} \quad ; \quad V_T = \frac{kT}{q}$$

2g. Transistor Bipolar

Para usar un transistor bipolar (BJT), debe darse inicialmente un nombre Qxxxxxxx que comience por "Q", luego siguen los nodos del colector *nc*, base *nb* y emisor *ne*, y opcionalmente el del sustrato *ns*. Sigue el nombre de modelo del BJT, y luego el factor de área, que indica el número de dispositivos en paralelo equivalente.

Qxxxxxxx nc nb ne [ns] nombre de modelo [área] [OFF] [IC=Vbe,Vce]

Si se omite el nodo de sustrato, se supone conectado a tierra. Si se omite el área, se supone que es 1, es decir, que sólo hay un BJT, y no hay varios en paralelo. Si se añade la opción *OFF* el proceso cálculo de la solución comienza suponiendo al BJT en corte. Si se añade la condición inicial *IC=0.6,5.2* implica que en el análisis transitorio se comenzará con una tensión $V_{BE}=0,6V$ y $V_{CE}=5,2V$ (usar *UIC* en *.TRAN*).

```
Q11 3 2 1 Qmio IC=0.6,5.0
Q22 9 8 7 6 Qmio 2
.MODEL Qmio NPN ( Is=1E-14 Bf=250 Vaf=40V )
```

En el anterior ejemplo, definimos un BJT-NPN de nombre "Qmio", con todas los parámetros por defecto, salvo que la corriente I_S es $6E-14$ amperios, tiene una $\beta_F=250$, y una tensión Early directa de $40V$. Q_{11} tiene el nodo de sustrato "sin conectar", es decir, conectado a tierra, y Q_{22} lo tiene conectado al nodo 6. Q_{22} es el doble de ancho que Q_{11} , y en éste se fijan condiciones iniciales para el transitorio.

Algunos parámetros básicos de un diodo, junto con su nombre, y valor predefinido:

Parámetro:	Nombre	Valor
Corriente de saturación	IS	1E-16 A
Coeficiente de emisión n_F	NF	1
Beta directa máxima	BF	100
Tensión Early (directa)	VAF	infinito
Coeficiente de emisión n_R	NR	1
Beta inversa máxima	BR	1
Tensión Early inversa	VAR	infinito
Ancho banda prohibida E_g	EG	1.11 eV
Exponente de temperatura	XTI	3.0
Capacid. base-emisor a $0V$	CJE	0 F
Potencial base-emisor	VJE	0.75 V
Coeficiente de graduación	MJE	0.33
Tiempo de tránsito directa	TF	0 s
Capacid. base-colector a $0V$	CJC	0 F
Potencial base-colector	VJC	0.75 V
Coeficiente de graduación	MJC	0.33
Tiempo de tránsito inversa	TR	0 s
Resistencia serie base	RB	0 Ω
Resistencia serie emisor	RE	0 Ω
Resistencia serie colector	RC	0 Ω
Tiempo transito directo	TF	0 s

2g. JFET

Para usar un transistor JFET, debe darse inicialmente un nombre *Jxxxxxxx* que comience por "J", luego siguen los nodos del drenador *nd*, puerta *ng* y fuente *ns*. Sigue el nombre de modelo del JFET, y luego el factor de área, que indica el número de dispositivos en paralelo equivalente.

Jxxxxxxx nd ng ns nombre de modelo [área] [OFF] [IC=Vds,Vgs]

Si se omite el área, se supone que es 1, es decir, que sólo hay un JFET, y no hay varios en paralelo. Si se añade la opción *OFF* el proceso cálculo de la solución comienza suponiendo al JFET en corte. Si se añade la condición inicial *IC=3.3,-1.1* implica que en el análisis transitorio se comenzará con una tensión $V_{DS}=3.3V$ y $V_{GS}=-1.1V$ (usar UIC en .TRAN).

```
J11 3 2 1 Jmio IC=3.3, -1.1
J22 9 8 7 6 Jmio 2
.MODEL Jmio NJF ( VT0=-2.0V Beta=2E-4 Lambda=1E-4 Rd=6 Rs=8 )
```

En el anterior ejemplo, se usa un JFET de canal N (NJF), con tensión umbral de -2V, transconductancia $\beta=2E-4A/V^2$, y la modulación del canal ($\lambda=10^{-4}V^{-1}$) que es como el efecto Early en BJTs. Las resistencias serie de drenador y fuente son de 6Ω y 8Ω respectivamente. *J₂₂* es dos veces más ancho que *J₁₁*.

Parámetro:	Nombre	Valor
Tensión umbral (pinch-off)	VT0	-2 V
Transconductancia	BETA	1E-4 A/V ²
Modulación longitud canal	LAMBDA	0 V ⁻¹
Coeficiente de emisión	N	1
Tensión de la unión PN	PB	1 V
Resistencia de drenador	RD	0 Ω
Resistencia de fuente	RS	0 Ω
Resistencia de puerta	RG	0 Ω
Corriente de saturación GS	IS	1E-14 A
I saturación recombinación	ISR	0 A
Coef. emisión recombinac.(η_R)	NR	2
Capacid. de unión GS	CGS	0 F
Capacid. de unión GD	CGD	0 F

Ecuaciones:

(No hemos puesto la contribución de la ionización.)

$$\begin{aligned}
 I_{DS} &= 0 & V_{GS} \leq V_{T0} & \text{ corte} \\
 I_{DS} &= \beta \cdot (V_{GS} - V_{T0})^2 \cdot [1 + \lambda \cdot V_{DS}] & (V_{GS} - V_{DS}) \leq V_{T0} & \text{ saturación} \\
 I_{DS} &= \beta \cdot \left[2(V_{GS} - V_{T0})V_{DS} - V_{DS}^2 \right] \cdot [1 + \lambda \cdot V_{DS}] & V_{T0} < (V_{GS} - V_{DS}) & \text{ triodo}
 \end{aligned}$$

2h. MOSFET

Para usar un transistor MOSFET, debe darse inicialmente un nombre *Mxxxxxxx* que comience por "M", luego siguen los nodos del drenador *nd*, puerta *ng*, fuente *ns*, y sustrato *nb*. Sigue el nombre de modelo de mosfet, luego el factor de área, que indica el número de dispositivos en paralelo equivalente.

Mxxxxxxx nd ng ns nb nombre de modelo [M=área] [L=x][W=x] ...
... [AD=x][AS=x][PD=x][PS=x] [OFF] [IC=Vds,Vgs,Vbs]

Se puede fijar la longitud y ancho del canal para cada mosfet con L y W. Si se omiten se toma la opción por defecto: la fijada en el modelo, la de spice (100μm), o la definida con DEFL y DEFW en .OPTIONS). Se puede fijar el área y perímetro de las difusiones de drenador y fuente con AD, AS, PD, PS.

Si se omite el *área*, se supone que es 1, es decir, que sólo hay un mosfet, y no hay varios en paralelo. Si se añade la opción *OFF* el proceso cálculo de la solución comienza suponiendo al transistor en corte. Si se añade la condición inicial *IC=3.3,2.2,0.1* implica que en el análisis transitorio se comenzará con una tensión $V_{DS}=3,3V$, $V_{GS}=2,2V$ y $V_{BS}=0,1V$ (usar UIC en .TRAN).

```
M11 13 12 11 0 Mmio IC=3.3,2.2,-0.1
M22 23 22 21 8 Mmio L=3u W=5u
M33 33 32 31 8 Mmio 3u 5u
.MODEL Mmio NJF ( VT0=2.0V Kp=3E-4 Lambda=1E-4 Rd=2 Rs=3 )
```

M_{11} tiene los nodos de drenador, puerta, fuente, sustrato igual a 13, 12, 11, 0. Se especifican las condiciones iniciales de V_{DS} , V_{GS} y V_{BS} . M_{22} y M_{33} tienen el sustrato conectado al nodo 8, y ambos tienen la misma longitud de canal ($3\mu m$) y ancho de canal ($5\mu m$). M_{11} tiene un ancho y longitud de canal de $100\mu m$, puesto que al no definirlo, se toma el valor por defecto de spice. Los tres mosfet usan el nivel 1 de mosfet, que es predefinido en spice. Los tres tienen una tensión umbral de 2V, con la transconductancia de $3e-4 A/V^2$ y la modulación del canal ($\lambda=10^{-4}V^{-1}$) que es como el efecto Early en BJTs. Las resistencias serie de drenador y fuente son de 2Ω y 3Ω respectivamente.

Parámetro:	Nombre	Valor
Nivel de modelo	LEVEL	1
Longitud del canal	L	defl*
Ancho del canal	W	defw*
Tensión umbral	VT0	0 V
Transconductancia	KP	$2E-5 A/V^2$
Modulación longitud canal	LAMBDA	$0 V^{-1}$
Coef. de emisión sustrato	N	1
Resistencia de drenador	RD	0Ω
Resistencia de fuente	RS	0Ω
Resistencia de puerta	RG	0Ω
Resistencia de sustrato	RB	0Ω
R paralelo drenador-fuente	RDS	0Ω
Difusión lateral	LD	$0 m$
Corriente de sat-sustrato	IS	$1E-14 A$
Potencial inversión superf.	PHI	$0,6V$
Param. efecto sustrato	GAMMA	0
Capacid. de sustrato-d	CBD	$0 F$
Capacid. de sustrato-s	CBS	$0 F$

Los valores *defl* y *defw* son la longitud y ancho por defecto, que si no se cambian son $100\mu m$. Se pueden cambiar con el comando .OPTIONS, o bien, en los parámetros que se añaden al llamar al mosfet.

Ecuaciones en nivel 1 (LEVEL=1):

$$I_{DS} = 0 \quad V_{GS} \leq V_{T0} \quad \text{corte}$$

$$I_{DS} = \frac{W K_p}{L^2} \cdot (V_{GS} - V_{T0})^2 \cdot [1 + \lambda \cdot V_{DS}] \quad (V_{GS} - V_{DS}) \leq V_{T0} \quad \text{saturación}$$

$$I_{DS} = \frac{W K_p}{(L - L_D)^2} \cdot \left[2 (V_{GS} - V_{T0}) V_{DS} - V_{DS}^2 \right] \cdot [1 + \lambda \cdot V_{DS}] \quad V_{T0} < (V_{GS} - V_{DS}) \quad \text{triódo}$$

3. LAS LIBRERIAS

Las librerías son ficheros que contienen definiciones de subcircuitos y dispositivos (diodos, transistores, etc.). Estas definiciones se pueden utilizar en los circuitos, siempre que el programa (*.CIR) incluya una llamada a la librería que contiene las definiciones.

Las librerías pueden contener los dispositivos distribuidos por la versión de Spice que se use; pueden contener los dispositivos modelados por algún fabricante, o bien, ser diseñados a gusto del usuario.

Las librerías son ficheros de tipo texto, legibles y editables por el usuario. Algunas versiones de Pspice crean un fichero binario equivalente (*.IND) para agilizar la carga y lectura de la librería por el programa.

- La sentencia ".LIB" es equivalente a ".LIB NOM.LIB", y llama a la librería NOM.LIB que vuelve a llamar a otras (DNOM.LIB, QNOM.LIB, MNOM.LIB, OPNOM.LIB, CPNOM.LIB y BJT.LIB). NOM.LIB es un fichero editable por el usuario, que incluirá las librerías más necesarias o habituales.

- También se puede llamar por separado a una o varias librerías, usando el nombre: .LIB miLibreria.lib. En el caso de que la librería no esté en el directorio actual o en el "path", se puede incluir la ruta: .LIB c:\directorio\BJT.lib.

- La versión 5.1 de Pspice suministrada en esta página web, además de las básicas, tiene muchísimas librerías con una enorme variedad de dispositivo (subdirectorio .LIB\). También hay otro subdirectorio (.LIB0\) con otras librerías (más antiguas), y con EVAL2.lib que es de PSpice 5.1-win. En .LIB2\ se añaden librerías de LT-Spice-II, que se pueden adaptar fácilmente a PSpice.

Contenido de algunas librerías de Pspice:

DNOM.LIB (diodos): D1N914, D1N916 Zener: D1N752 5,2V, D1N754 6,4V, D1N759 11,4V

QNOM.LIB (BJT): 2N2222 (NPN, $\beta=217$) 2N2907 (PNP, $\beta=197$).

BJT.LIB (BJT): BC107B (NPN, $\beta=300$) BD557 (PNP, $\beta=30$), LM394 (NPN, $\beta=600$), BC547B (NPN, $\beta=300$), BC557B (PNP, $\beta=344$), BD237 (NPN, $\beta=108$).

MNOM.LIB (mosfet): IRF150 (canal N) IRF9130 (canal P)

OPNOM.LIB (amp.op): uA741 (mejor usar versiones de Linear.lib o Eval2.lib)

CPNOM.lib (comparador): LM111

EVAL2.lib (pspice-5.1-win):

Diodos:	D1N4148	MBD101	Zener:	D1N750
BJT-NPN:	Q2N2222A	Q2N3904	BJT-PNP:	Q2N2907A Q2N3906
JFET-N:	J2N3819	JFET-P:	J2N4393	
MOSFET-N:	IRF150	MOSFET-P:	IRF9140	
Amp.Op:	LM324	UA741	Comparador:	LM111

4. TIPOS DE ANALISIS

Existen básicamente tres tipos de análisis:

- de Continua
- de Pequeña Señal
- de Tiempo o Transitorio.

En el análisis de pequeña señal, siempre se comienza haciendo un análisis de continua, aunque no se pida, pues debe calcular los modelos linealizados de los dispositivos no lineales (modelos de pequeña señal). Puede especificarse que, además, se haga un análisis de ruido o de distorsión con .NOISE o .DISTO.

En el análisis de tiempo o transitorio, aunque no se pida, se hace un análisis de continua, para determinar el estado inicial de continua en tiempo $t=0s$. Siempre calcula desde $t=0s$, aunque se diga que comience después, porque "empezar después" en "t-start" significa que spice no mostrará los cálculos realizados en el intervalo $[0s, t-start]$. Si se tienen simulaciones sinusoidales de gran señal, se puede pedir que haga un análisis de Fourier de la señal de salida (.FOUR). Si se desea

hacer un análisis de tiempo, pero con los modelos de pequeña señal de los dispositivos, debe añadirse que calcule el punto de operación (.OP).

En el análisis de continua se puede pedir que haga un barrido de continua de una variable respecto a los valores de una fuente de continua (.DC), o que calcule los modelos lineales de pequeña señal con .OP. También se puede pedir que calcule el valor de pequeña señal de la función de transferencia y resistencias de entrada y salida con .TF. Si se añade .SENS, puede hacer un análisis de sensibilidad de una variable respecto a algún parámetro del circuito.

De barrido en continua (DC): Se analiza el circuito mientras se hace variar el valor de continua de alguna fuente de tensión o de intensidad.

```
.DC Vx1 valor_inicio valor_final dc_incremento
```

La orden de la línea anterior, hace variar el valor de la fuente de tensión Vx1 desde *valor_inicio* hasta *valor_final*, en pasos de *dc_incremento*. Admite un doble bucle, como en el siguiente ejemplo, que barre la fuente V1 de 0V a 5V, para cada valor de la fuente Ib, que varía de 0 mA a 100 mA:

```
.DC V1 0V 5V 1V Ib 0mA 100mA 20mA
```

Más ejemplos:

```
.DC V1 2 12 1V ; desde 2V a 12V paso=1V
.DC LIN V1 2 12 2 ; desde 2 a 12 con paso 2V
.DC DEC V1 0.1 100 4 ; Logaritmico de 0,1 a 100 4pt/Decada
.DC OCT V1 0.1 100 4 ; Logaritmico de 0,1 a 100 4pt/Octava
.DC TEMP 10 60 5 ; desde 10°C a 60° paso=5°C
.DC LIN TEMP 10 60 5 ; desde 10°C a 60° paso=5°C
.DC DEC TEMP 1 100 4 ; Logaritmico de 1°C a 100°C 4pt/Decada
.DC OCT TEMP 1 100 4 ; Logaritmico de 1°C a 100°C 4pt/Octava
```

También se pueden especificar listas de valores;

```
.DC V1 LIST 2 3 6 7 11 12 ; V1 = 2V 3V 6V 7V etc.
.DC TEMP LIST 10 15 30 35 55 60 ; T= 10°C 15° 30° 35° etc.
```

Se podría hacer un barrido en parámetros de dispositivos:

```
.DC LIN NPN Qmio Bf 10 210 50 ; beta directa de Qmio de 10 a 210
.DC DEC NPN Qmio (Is) 1E-18 1E-15 5 ; barre Is con 5 pt/decada
```

De barrido en frecuencias (AC): Analiza la respuesta del circuito al variar la frecuencia de las fuentes de señal alterna (AC):

```
.AC LIN npuntos frec_inicio frec_final
```

El comando precedente, realiza un análisis en alterna, calculando los resultados pedidos en un barrido en total de *npuntos* frecuencias, de forma LINEal empezando por la frecuencia *frec_inicio* hasta *frec_final*. Si en la orden anterior se sustituyese "LIN" por "DEC", se realizaría un barrido logarítmico con *npuntos* por cada DECada. Con "OCT" sería un barrido con *npuntos* por cada OCTava.

Como el análisis AC es de pequeña señal, antes de hacer el análisis en frecuencia, el circuito es linealizado en la zona del punto de operación. Esa es la razón de que el valor dado en una fuente (con AC) sea indiferente, y se suela coger igual a "1" por comodidad.

De barrido en el tiempo (TRANSitorio): Analiza la evolución temporal de magnitudes del circuito.

```
.TRAN tiempo_paso tiempo_fin tiempo_inicio tiempo_incremento
```

La orden anterior, realiza un análisis transitorio, calculando los resultados desde un tiempo igual a *tiempo_inicio* hasta *tiempo_fin*, con incremento igual a *tiempo_paso*. Si la resolución gráfica es pobre, se debe incluir *tiempo_incremento* ("Step_Ceiling") con valor menor de $(tiempo_final - tiempo_inicio)/50$.

De temperatura: Calcula los resultados pedidos a diferentes temperaturas.

```
.TEMP 50 75 Realiza el análisis a temperaturas de 50°C y 75°C.
```

Análisis de Sensibilidad (.SENS): Interesante para ver la influencia (sensibilidad) de una variable del circuito frente a parámetros (valores de componentes) del circuito.

`.SENS V(2,1) V(3) I(R1)` Calcula la sensibilidad absoluta y relativa de la tensión V_{21} , de V_3 y de la intensidad en R1.

De función de transferencia (.TF): Calcula la función de transferencia de pequeña señal, las resistencias de entrada y de salida en corriente continua.

`.TF V(3,2) Vin`

Donde $V_{32}=V(3,2)$ sería la variable de salida (con el mismo formato de `.PRINT`), y V_{in} sería la señal de entrada, que puede ser una fuente de tensión o de intensidad, pero independientes.

De Fourier (.FOUR): Especifica que se hará junto al análisis de tiempo (`.TRAN`) un análisis de Fourier sobre las señales de tensión $V(3)$ y $V(5)$, teniendo como frecuencia fundamental 2kHz:

`.FOUR 2k V(3) V(5)`

De punto de operación (.OP): Muestra la información de los modelos de pequeña señal de los dispositivos no lineales en un punto de trabajo.

Paramétrico (.STEP): Permite realizar todos los análisis del circuito variando el valor de una fuente independiente (tensión/intensidad), un parámetro de un dispositivo (resistencia, diodo, transistor, etc.), de temperatura, o incluso un parámetro definido por nosotros.

```
.STEP LIN variable valor_inicio valor_final incremento
.STEP variable valor_inicio valor_final incremento
.STEP OCT variable valor_inicio valor_final nº-puntos
.STEP DEC variable valor_inicio valor_final nº-puntos
.STEP variable LIST valor1 valor2 ...
```

Si la variable es la temperatura, poner TEMP en el lugar de la variable. El análisis STEP es incompatible con un análisis de tipo `.DC`

```
.STEP PARAM variable1 1k 10k 3k ; de 1k a 10k en paso de 3k
.STEP PARAM variable1 LIST 1K 5K 10K ; lista de valores
.STEP TEMP LIST 20 60 80 ; 20°C 60°C 80°C
.STEP DEC NPN Qm1o(Is) 1e-16 1e-13 6
.STEP RES Rm1a(R) 1k 10k 1k
.STEP LIN RES Rm1a(R) 1k 10k 1k
.STEP Vcc1 10V 18V 2V ; Vcc1 de 10 a 18V paso=2V
.STEP LIN Is1 1mA 8mA 1mA ; Is1 de 1mA a 8mA paso=1mA
```

5. SENTENCIAS UTILES

- Muchas veces se tiene una parte fija que se incluye en todos los `*.cir` que uno hace. En esos casos se usa un "include fichero". En la siguiente orden se incluye el contenido del fichero.txt en el `*.cir`:

`.INC fichero.txt`

- Con la sentencia `.IC` se fijan las condiciones iniciales en ciertos nodos. Se dan las tensiones de nodo iniciales en ciertos nodos. Así, durante un análisis `.TRAN`, se verá la evolución del transitorio partiendo de esas condiciones iniciales. La sentencia `.IC` no afecta a un análisis de continua.

`.IC V(2) 3 V(4)=6V` la condición inicial de $V(2)$ es 3V y de $V(4)$ es 6V

- Por defecto, Spice admite ficheros de entrada con una anchura de 80 caracteres y con una salida de 80 caracteres. Eso se puede cambiar con la sentencia `.WIDTH` que especifica un ancho de entrada xxi, y un ancho de salida xxo (xxo sólo puede tomar los valores 80 o 132. Habitualmente sólo se necesita un ancho extra en el fichero de salida para las tablas de datos, por eso, lo que se suele ver es: `.WIDTH 132`

`.WIDTH IN=xxi OUT=xxo`

El comando .OPTIONS tiene muchas utilidades, y una de ellas es fijar la anchura, por eso, muchas veces no se usa .WIDTH, sino la opción equivalente de .OPTIONS.

- Existe otra sentencia que es muy útil, y que configura el funcionamiento de Spice: .OPTIONS. Tras .OPTIONS se pueden añadir los siguientes modificadores:

- ACCT Muestra en el fichero .out, las estadísticas del tiempo de cálculo
- NOMOD No muestra los parámetros de los modelos en el fichero .out
- OPTS En el fichero *.out, se muestran las opciones activadas.
- NODE Muestra la tabla de nodos y los elementos conectados a ellos.
- TNOM=xx Fija la temperatura nominal a xx°C. Por defecto es 27°C
- DEFL=xx Fija la longitud de canal de un Mosfet. Por defecto es 0,1mm=100µm
- DEFW=xx Fija la anchura de canal de un Mosfet. Por defecto es 0,1mm=100µm
- NOPAGE Elimina los códigos de control de cambio de página, y sus títulos correspondientes.

Por tanto, muy útil, salvo que el fichero de salida .out vaya a imprimirse en papel.

- NUMDGT=xx Especifica el número de dígitos significativos en los resultados numéricos. En Spice xx debe cumplir: $0 < xx < 8$, pero en Pspice, admite cualquier número entero positivo, incluyendo el 0. Si $xx=0$, no se da ninguna cifra significativa, lo que es ridículo. Si xx es mayor que 11, no da error, simplemente da 11 cifras significativas.

- RELTOL=xx Fija el error relativo (%) máximo en los cálculos. Por defecto es 0,001 (0,1%)

- LIMPTS=xx Límite de puntos calculado en .PRINT/.PLOT. Por defecto es 201. El máximo, según versiones, es 16.000/32.000. Si $xx=0$ es número ilimitado.

- TRTOL=xx Fija la tolerancia al error en análisis de tiempo .TRAN. Por defecto es 7 (7%)

- LVLTIM=xx Especifica cómo proceder cuando las iteraciones tienen fácil/difícil el cálculo de los puntos en un análisis de tipo .TRAN

- ITL4=xx Controla el número de iteraciones al resolver los cálculos en análisis .TRAN.

- Las opciones LVLTIM, TRTOL e ITL4 (por defecto: 201, 7% y 10), se aplican en los análisis de tipo .TRAN. Como son para controlar la convergencia de los cálculos, se podría pensar que no son importantes, pero bastantes veces es necesario activarlas. En circuitos muy simples, pueden verse que las senoidales se ven quebradas, sin su curvatura característica, como si faltaran puntos de cálculo. Estos efectos son debidos a que Spice2 (y por tanto Pspice) reduce o aumenta el intervalo de tiempo de cálculo según vea fácil o difícil la convergencia. Así, en casos de convergencia fácil, aumenta el paso de cálculo, es decir el intervalo de tiempo entre puntos de cálculo. Concretamente duplica el tiempo entre dos puntos de cálculo. Por eso, en circuitos muy simples puedes ver que las curvas tienen poquísimos puntos calculados. Pero si la convergencia es difícil, reduce (en 1/8) el paso de cálculo.

Si LVLTIM=2 (por defecto), TRTOL="transient tolerance" es un parámetro importante. Si TRTOL se aumenta, se permite a Spice aumentar el error, y por tanto hace la convergencia más fácil, con lo que se aumenta el paso de cálculo, y las gráficas saldrán con menos puntos, más pobres, mas bastas. Pero si el circuito tiene una situación de convergencia difícil, un valor alto de TRTOL, impediría que Spice se pare al dar un error de convergencia.

Si LVLTIM=1, TRTOL, no tendría ninguna influencia. En caso de fácil convergencia, el paso de cálculo se duplicaría. En caso de difícil convergencia, el paso de cálculo se dividiría por 8. En estos casos, el número de iteraciones sólo dependería del número asignado a ITL4 (mayor, facilita la convergencia)

6. SALIDA DE RESULTADOS

Los resultados pueden obtenerse en forma de tabla numérica con `.PRINT`, de esa forma se pueden hacer cálculos matemáticos, estadísticos o graficarlos según el gusto personal. También se pueden obtener en forma gráfica dentro de un fichero de texto con `.PLOT`. Antiguamente, cuando los terminales de los ordenadores, incluso las impresoras eran de tipo texto, era una opción muy útil. En los tiempos actuales con las capacidades gráficas de pantallas, ordenadores, impresoras, `.PLOT` no es nada útil. Además, PSpice tiene un graficador muy bueno, con muchas posibilidades matemáticas: `PROBE`, que veremos en el siguiente apartado. `Probe`, permite incluso exportar la gráfica a formatos modernos (PCL, PS, HP-GL, etc.) e incluso permite exportar los datos numéricos de la gráfica como una tabla de datos, para luego operar sobre ellos.

`PRINT` y `PLOT`, generan sus tablas de datos y gráficas en el fichero de texto `*.out`. `Probe` es muy útil, por eso se suele añadir el comando `".PROBE"`, que almacena todos los datos del circuito para ser vistos luego con `Probe`.

```
.PRINT  tipo_salida  variables_salida
.PLOT   tipo_salida  variables_salida
.PROBE
```

Donde `tipo_salida` indica el tipo de datos que se han de mostrar, y puede ser:

- DC se muestran los valores de continua de las variables de salida.
- TRAN se muestran los valores de variables-salida en función del tiempo.
- AC se muestran los valores de alterna de variables_salida.

y `variables_salida` puede ser :

- V(n1,n2) caída de tensión entre los nodos n1 y n2
- V(n3) caída de tensión entre n3 y tierra (tierra siempre es nodo 0)
- I(Vs) intensidad que circula por la fuente de tensión Vs
- VM(n1) caída de tensión en Módulo/Magnitud. Es igual a V(n1)
- VDB(n1,n2) caída de tensión en decibelios (dB) (para obtener Bode en módulo)
- VP(n1,n2) Fase de la caída de tensión (°) (para diagrama de Bode en fase)
- VR(n1) parte Real de la caída de tensión. VI(n1) sería la parte imaginaria
- IB(Q1) intensidad que entra a la base de Q1
- VB(Q2) caída de tensión desde la base de Q2 hacia tierra
- VBE(Q3) caída de tensión Base-Emisor en el transistor Q3
- VGS(J4) caída de tensión Puerta-Fuente en el JFET J4
- IG(J5) intensidad que entra a la puerta del JFET J5

Es importante recordar que si se dan nombres no numéricos a los nodos (p.ej. N003), debe obligatoriamente emplearse corchetes, es decir `V([N003])`. Además, recordar que no deberían emplearse corchetes en las señales de `PROBE`. Conclusión: Mejor usar sólo nodos numéricos.

7. GRAFICADOR PROBE

PSpice está escrito en C, pero sus algoritmos, cálculos y especificaciones se basan en SPICE-2 (escrito en Fortran). Aun así, existen algunas mejoras de Pspice sobre Spice-2. Una de esas utilidades añadidas, es el graficador `PROBE`, que puede ser utilizado como un *super osciloscopio gráfico*. Otra de las utilidades añadidas por Pspice es `"PARTS"`, que sirve para obtener el modelo de spice de un dispositivo. `"Parts"`, de forma simple, te pide algunos datos de las especificaciones del fabricante del dispositivo, y él va construyendo el modelo de spice.

Los comandos de `Probe` están en inglés, y se explican perfectamente. Al añadir curvas, se pueden incluir las variables `V(2,1)`, `V(3)` (diferencia de potencial entre los nodos 2 y 1, y los nodos 3 y 0), y expresiones como `V(2,1)-2*V(3)`.

Dispone además de funciones como seno (sin), coseno (cos), tangente (tan), raíz cuadrada (sqrt), derivada (d), integral (s), potencia ($p(x,y)=|x|^y$), mínimo (min), máximo (max), valor absoluto (abs), exponencial (exp), logaritmo natural y decimal (log y log10) y algunas otras más. Estas operaciones matemáticas no se pueden hacer en los comandos .PRINT/.PLOT. Probe tiene mucha más flexibilidad en el cálculo de señales y su nomenclatura que .PRINT/.PLOT. Las siguientes líneas muestran operaciones posibles en PROBE, pero que no se pueden hacer en .PRINT/PLOT:

```
d( V(2) )           derivada de V(2)
d( V(N003) - 10 * V(2) )   derivada de ( VN003 - 10*V2 )
db( V(2) - V(5) )   V(2)-V(5)  módulo en decibelios
M( V(5) )           módulo-magnitud de V(5)
P( V(5) )           fase-argumento de V(5)
R( V(5) )           parte real de V(5)
IMG( V(5) )         parte imaginaria de V(5)
```

En la versión 5.1 de PSpice para DOS, existe el ejecutable setupdev.exe que permite escoger de forma simple el driver de pantalla y de impresora más adecuados, creando un fichero pspice.dev con la configuración elegida. También se puede editar directamente, pues es de tipo texto. Por ejemplo, en el siguiente pspice.dev:

```
Display = IBMVGA
Hard-Copy = FILE,HP8_GL2-PCL
Hard-Copy = FILE, TABLE
Hard-Copy = FILE, PS
Hard-Copy = PRN:,HPLJ
```

Se define la pantalla IBMVGA. Existen más opciones con mejores resoluciones (VGA800 VGA1024), pero a mí me han dado problemas con mis ordenadores. Como se observa, puede haber varias impresoras definidas. Al hacer un "HardCopy", PROBE, permitirá elegir cuál es la impresora de salida. Una opción muy útil es "imprimir a fichero" con FILE, de esa forma podemos generar un fichero de tipo ploter (HP8_GL2-PCL), postscript (PS), PCL de una HP-Laserjet (HPLJ) o incluso una tabla numérica (TABLE) para ser procesada en una hoja de cálculo. Para mí, las más útiles son las mostradas en el ejemplo.

.PROBE/CSDF (sin espacio entre ".probe" y "/csdf"). Además del fichero *.dat con los datos para el graficador Probe, se crea un fichero con esos datos en formato texto, legibles por el usuario.

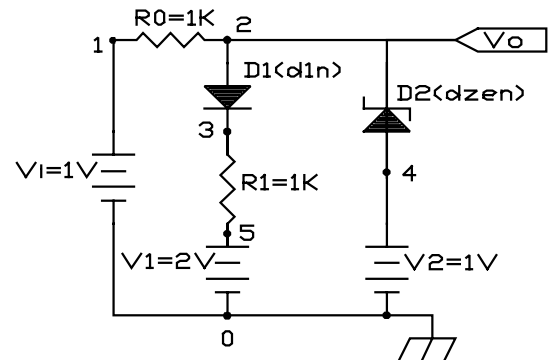
8. EJEMPLOS

Los ficheros mostrados se dan con la versión DOS de SPICE.

EJ1_DC.CIR:

* Ejemplo 1: Barrido en DC. Calculo de la característica de transferencia

```
Vi 1 0 1V
R0 1 2 1Kohmio
D1 2 3 D1n
R1 3 5 1Kohmio
V1 5 0 2V
D2 4 2 Dzen
V2 4 0 1V
; [ paso=0,1V
.DC Vi -2 10 .1 ; barre Vi desde -2V a 10V
.print DC v(2) v(d1) i(d1) i(v1) i(R1)
.print DC i(R0) v(d2) i(d2) i(v2)
.lib FTC.lib ; los diodos estan en FTC.lib
.options nomod nopage
.probe
.end
```

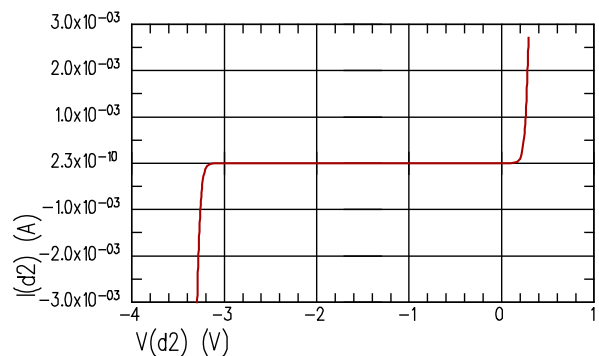
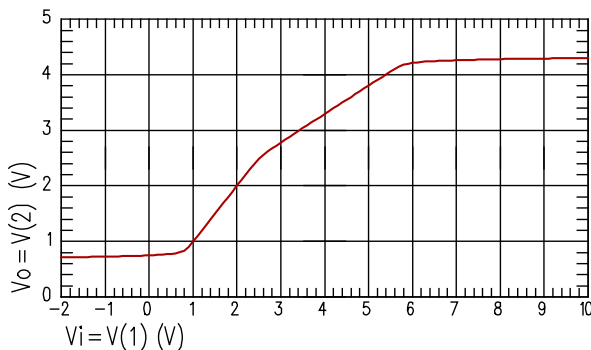


Ejemplo 1

EJ1_TRAN.CIR:

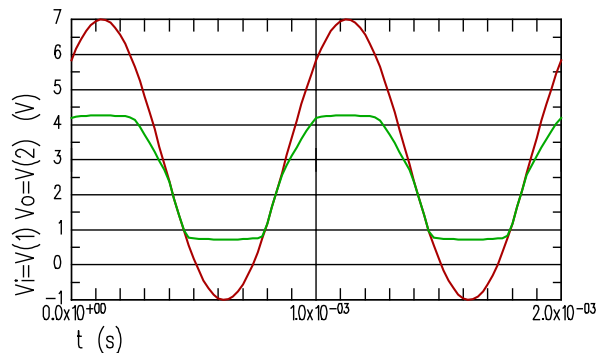
* Ejemplo 1: Transitorio. Calculo de la salida en funcion del tiempo

```
Vi 1 0 SIN(3 4V 1KHz 0 0 45) ; Amplitud=4V Frecuencia=1kHz Fase=45°
R0 1 2 1Kohmio ; Nivel de continua (offset) = 3V
D1 2 3 D1n
R1 3 5 1Kohmio
V1 5 0 2V
D2 4 2 Dzen
V2 4 0 1V
.TRAN 0.02ms 2ms 0s .01m ; desde 0 segundos a 2ms paso=0.02ms
* (con 0,01ms hemos forzado a que pspice tenga mejores curvas)
.print TRAN v(1) v(2) v(d1) i(d1) i(v1)
.print TRAN i(R0) i(R1) v(d2) i(d2) i(v2)
.lib FTC.lib ; los diodos estan en esa libreria
.options nomod nopage ; trtol=0.01
.probe
.end
```



Las dos primeras gráficas se obtienen con EJ1_DC.cir. La primera es la curva de transferencia $V_o = V(2)$ frente a $V_i = V(1)$. La segunda muestra la flexibilidad de .print o de .probe, pues al representar $I(D2)$ frente a $V(D2)$ se obtiene la curva $I_d - V_d$ del diodo.

La gráfica de la derecha se obtiene con EJ1_TRAN.cir. Representa la tensión de salida $V_o = V(2)$ y de entrada senoidal $V_i = V(1)$ frente al tiempo. Las gráficas se ven muy bien pues se ha tenido el cuidado de poner un paso de cálculo 0,01ms, menor que el de visualización (0,02ms).

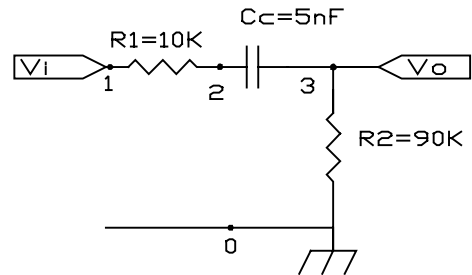


EJ2_TRAN.CIR:

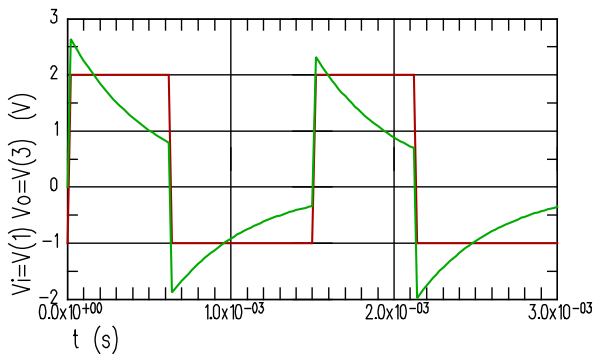
```
* Ejemplo 2: Transitorio. Respuesta de un circuito RC en funcion del tiempo
* Pulse=onda cuadrada desde -1V a +2V. Periodo: 1,5ms. Ancho del pulso: 0.6ms
Vi 1 0 Pulse( -1V 2V 0 0 0 0.6ms 1.5ms )
R1 1 2 10Kohmio
Cc 2 3 5nF
R2 3 0 90K
.TRAN 0.02ms 3ms 0s .01m ; desde 0 segundos a 3ms paso=0.02ms
* (con 0,01ms hemos forzado a que pspice tenga mejores curvas)
.print TRAN v(1) v(2) v(3)
.options nomod nopage ; trtol=0.01
.probe
.end
```

EJ2_AC.CIR:

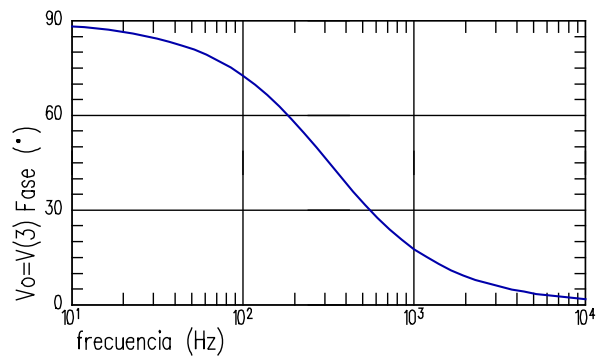
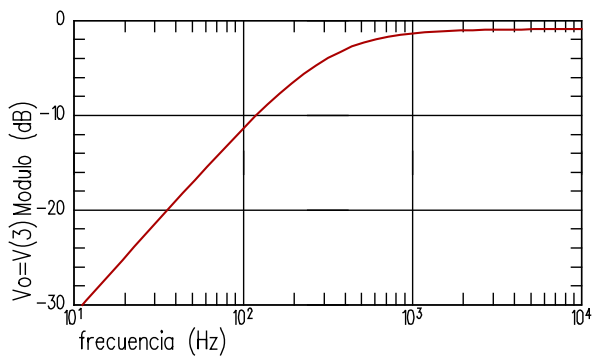
```
* Ejemplo 2: Analisis AC Diagrama de Bode en modulo
* y fase de un circuito RC
Vi 1 0 AC 1 ; Vi es fuente alterna
R1 1 2 10Kohmio
Cc 2 3 5nF
R2 3 0 90K ; [...14 puntos por decada
*El circuito se simula desde 10Hz a 10kHz con ...
.AC DEC 14 10 10000
.print AC v(1) v(2) v(3) vdb(3) vp(3)
.options nomod nopage
.probe
.end
```



Ejemplo 2



La gráfica de la derecha corresponde a la dependencia temporal de la tensión de salida $V_o=V(3)$ y de entrada cuadrada $V_i=V(1)$ en EJ2_TRAN.cir. Luego aparece el diagrama de Bode en módulo (en decibelios) y de fase (en grados sexagesimales) obtenido con el análisis .AC de EJ2_AC.cir.



EJ3_BJT.CIR:

* Ejemplo 3 Transistor Bipolar. Polarizacion de un par Darlington

V1 1 0 DC 2V

Rb 1 2 2.2MEGohmio

Q1 5 2 3 BC107 ; colector base emisor

Q2 5 3 0 BD237 ; BD237 se define en .model

Rc 6 5 1Kohmios

Vmd 8 6 0V ; sirve para medir la intensidad

Vcc 8 0 12V

.MODEL BD237 NPN(IS=3E-12 BF=90 NF=0.9 VAF=54

+ RB=8 TF=1e-08)

* Para que el .print actue, no basta hacer un .DC:

* - hacer un barrido .DC con: .DC V1 2 2 .1

* - hacer un analisis .DC a solo un valor: .DC V1 LIST 2

* - usar .OP, pero .OP solo da 3/4 cifras significativas

* aunque se use .options numdgt=6 (numero-digitos=6)

* - hacer un analisis .DC pero a temperatura ambiente: .DC TEMP LIST 27 (27°C)

.OP ; para comparar

.DC TEMP LIST 27 ; usamos

* Vbe(Q1)=V(2,3) Vce(Q1)=V(5,3)

* Vbe(Q2)=V(3)=V(3,0) Vce(Q2)=V(5,0)

* I(Vmd)=Ic(Q1)+Ic(Q2)=-I(Vcc)=I(Rc)

.print DC Vbe(Q1) Vce(Q1) Ib(Q1) Ic(Q1) Ie(Q1)

.print DC Vbe(Q2) Vce(Q2) Ib(Q2) Ic(Q2) Ie(Q2)

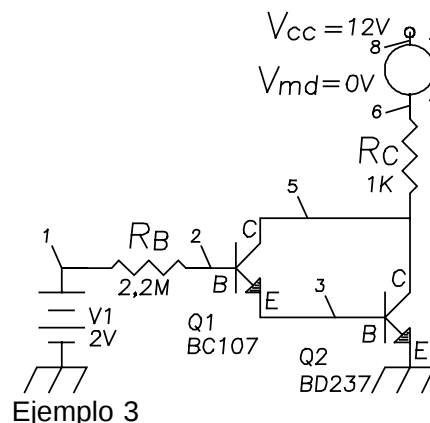
.print DC I(Rb) I(Rc) I(Vmd) I(Vcc)

.print DC V(1) V(Vcc) V(Rb) V(Rc)

.lib FTC.lib ; el BC107 esta en esa libreria

.options nomod nopage numdgt=6 width=132

.end



EJ4_AC.CIR:

* Ejemplo 4. Bode de configuracion inversora de amplificador operacional. Barrido de ganancia

Vcc1 9 0 +12V ; +Vcc Alimentacion (+)

Vcc2 8 0 -12V ; -Vcc Alimentacion (-)

Vi 1 0 AC 1 ; Señal-entrada: módulo=1 fase=0°

* Etapa con ganancia (-R2/R1)=(-Rf/R1)

R1 1 2 1k

R2 2 3 {Rf} ; R2=Parametro(Rf)

RL 3 0 10k

xA30 0 2 9 8 3 uA741 ; patas: + - +Vcc -Vcc output

.PARAM Rf=3K

* Rf=2k 10k 100k 10000k => distintas ganancias

.STEP PARAM Rf LIST 2k 10k 100k 10000k

.AC DEC 9 1 10MegHz ; Analisis AC desde 1Hz a 10MHz (9 pt/dec)

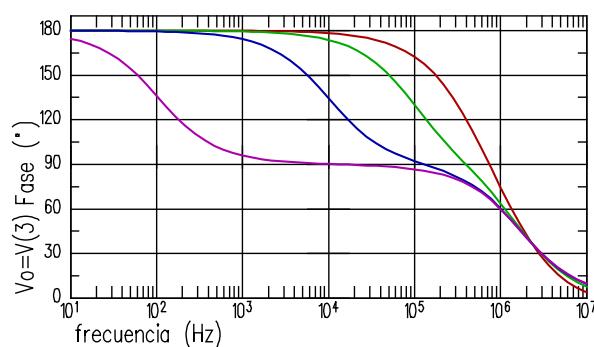
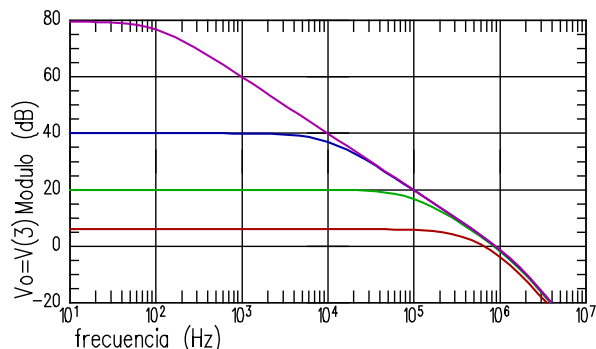
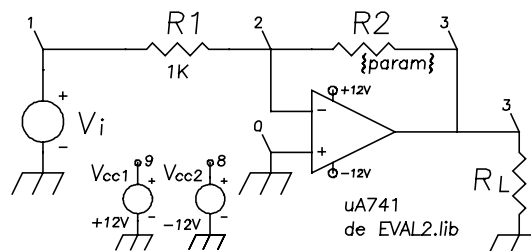
.PRINT AC V(3) Vdb(3) Vp(3) ; Salida: Modulo Modulo-dB Fase

.lib EVAL2.lib ; libreria que contiene al uA741

.options nomod nopage

.probe

.end



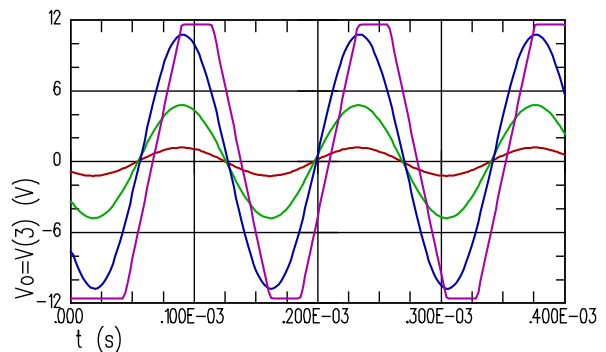
Las dos primeras gráficas son el diagrama de Bode en módulo (dB) y fase (°) obtenidas con EJ4_AC.cir. En ellas se ve cómo el amplificador uA741 es de tipo "compensado", y el ancho de banda aumenta al disminuir la ganancia.

EJ4_TRAN.CIR:

```
* Ejemplo 4. Bode de configuracion inversora de
* amplificador operacional. Barrido de ganancia.
Vcc1 9 0 +12V ; +Vcc Alimentacion (+)
Vcc2 8 0 -12V ; -Vcc Alimentacion (-)
Vi 1 0 SIN(0 1.2 7kHz 0 0 45) ; Amplitud=1.2V Frecuencia=7kHz Fase=45°
* Etapa con ganancia (-R2/R1)=(-Rf/R1)
R1 1 2 1k
R2 2 3 {Rf} ; R2=Parametro(Rf)
RL 3 0 1000k
xA30 0 2 9 8 3 uA741 ; patas: + - +Vcc -Vcc output
.PARAM Rf=3K
* Rf=1k 4k 9k 18k => distintas ganancias (-R2/R1)
.STEP PARAM Rf LIST 1k 4k 9k 18k
.TRAN 0.002ms .4ms 0s .001m ; desde 0 segundos a 2ms paso=0.01ms
* (con 0,01ms hemos forzado a que pspice tenga mejores curvas)
.print TRAN V(1) V(3) ; entrada y salida
.lib EVAL2.lib ; libreria que contiene al uA741
.options nomod nopage ; trtol=0.01
.probe
.end
```

En EJ4_AC.cir y EJ4_TRAN.cir se usa el subcircuito de un amplificador operacional (uA741) que está en la librería EVAL2.lib. Además se ilustra la utilidad de los comandos .STEP y .PARAM. Con lo que se puede hacer un barrido con .STEP, de un parámetro $R_2=\{R_f\}$ y por tanto variar la ganancia de la configuración inversora.

En la gráfica de la derecha, de EJ4_TRAN.cir, se observa cómo al aumentar la ganancia, la señal de salida $V_o=V(3)$ se amplifica, pero finalmente se llega a saturar un poco antes de la alimentación ($\pm 12V$) y se observa que tiende a convertirse en triangular debido al SR bastante bajo del uA741.



SPICE-DOS

1. INSTALACION Y USO

Descomprimir los ficheros, y copiarlos en un directorio. PSPICE para DOS consta de dos programas:

PSPICE1.EXE El pspice propiamente dicho. Su entrada es un fichero acabado en ".CIR" y su salida es un fichero con el mismo nombre, pero acabado en ".OUT". El fichero ????.CIR se crea con cualquier editor de texto, contiene el programa y la descripción del circuito. El fichero ????.OUT contiene los resultados de la simulación, o las explicaciones y posición de los errores cometidos.

PROBE.EXE Permite ver gráficamente los resultados, siempre que en el fichero ????.CIR se pidiera esta opción con la orden ".PROBE". Su fichero de entrada es PROBE.DAT (lo genera PSPICE), y su fichero de configuración es PROBE.DEV.

PSPICE.BAT Es una macro que corre primero PSPICE1 y luego PROBE.

Probe.dev o Pspice.dev Define las características del monitor ("display"), impresora física o grabación en fichero ("hardcopy"). En las versiones iniciales se usaba probe.dev, y se editaba a mano. En las más modernas, se puede seguir editando a mano, pero es más cómodo usar setupdev.exe.

PS.exe Es un "IDE", es decir un entorno para editar circuitos, realizar la simulación, ver los resultados con probe, obtener ayuda sobre los comandos en un solo entorno o programa. Teniendo la posibilidad de ventanas ("windows"), no parece tan útil. Quizás sí resulte útil, la ayuda breve y concisa sobre comandos, tipos de fuentes, tipos de análisis, etc.

StmEd.exe y Parts.exe StmEd.exe "Stimulus-Editor" crea fuentes independientes de forma asistida. Parts.exe es capaz de generar, de forma simple y guiada, modelos de pspice con sus parámetros simplemente suministrando algunos datos clave que los fabricantes dan en las especificaciones de un dispositivo.

"pspice prueba" o "pspice prueba.cir" simula el circuito definido en prueba.cir. El fichero de salida con los resultados sería prueba.out. Tras la simulación se arranca probe.exe inmediatamente. Para no llamar a probe.exe, y mandar los resultados a un fichero salida.out, ejecutar "pspice1 prueba.cir salida.out". En las versiones más antiguas, los resultados que usa probe, se almacenaban en probe.dat; en las modernas en prueba.dat (referenciado en probe.nam).

2. CONFIGURACION

En el fichero PROBE.DEV está la configuración de pantalla e impresora para PROBE.EXE. El fichero PROBE.DEV se puede editar "a mano". En versiones más modernas, se llama PSPICE.DEV y, además, se puede usar el programa setupdev.exe para generar un fichero adaptado a las necesidades personales. Sólo puede haber una línea que defina el tipo de pantalla, pero puede haber varias para definir las impresoras disponibles. PROBE permite elegir en qué impresora se hace la impresión "hardcopy". La impresión puede hacerse a un puerto de impresora, o mejor aún, a un fichero (FILE). Ejemplo de Probe.dev:

```
Display = IBMVGA
Hard-Copy = FILE,HP8_GL2-PCL
Hard-Copy = FILE,TABLE
Hard-Copy = FILE,PS
Hard-Copy = LPT1,HPLJ
```

En el anterior ejemplo la pantalla se configura a IBMVGA y se tienen varias posibilidades de impresión. Se puede imprimir en una impresora HP-Laserjet en LPT1:. Las opciones más útiles son la impresión "en fichero" ("FILE"). "TABLE", graba como tabla numérica de datos los valores de las curvas. "HP8_GL2-PCL" graba como un fichero de plotter en color (vectorial). "PS" graba en formato postscript, para hacer imágenes .eps o generar un pdf.

Otra opción, es correr el programa Probe en una ventana DOS dentro de Windows, y hacer un volcado de pantalla hacia un fichero. Resultará un gráfico binario, cuya resolución dependerá de la que tenga la pantalla del ordenador. Además será conveniente invertir los colores para no imprimir el fondo negro.

Las siguientes tablas dan valores aptos para versiones más antiguas de Spice. Las más modernas, disponen de SetupDev.exe, que muestra más y mejores opciones. Además debe tenerse en cuenta que muchas familias de impresoras comparten el lenguaje (PCL, ESC, PS, HP-GL, etc), y por tanto un modelo puede servir para otra similar.

Valores posibles de pantalla (dependen de la versión de Spice):

TEXT	IBM	IBMClr	IBMEGA	GenericEGA	IBMEGAMono	AT&T	Hercules
Tecmar8	Tecmar4	Tecmar3	Tecmar1	FutureNet			

Recomendación (de monitor más antiguo a más moderno):

TEXT	IBMEGA	IBM	IBMVGA	VGA800	VGA1024
------	--------	-----	--------	--------	---------

Valores posibles de puerto de impresora:

PRN	LPT1	LPT2	LPT3	COM1	COM2	FILE
-----	------	------	------	------	------	------

Valores posibles de impresoras:

TEXT	TEXT132	Okidata	Okidata132	Epson	Epson132	EpsonMX
EpsonMX132	IBMClr	IBMClr132	IBMClrSlw	IBMClrSlw132	Citoh	Citoh132
CitohSlw	CitohSlw132	Citoh-300	Citoh-300-132	Printronix	Printronix132	DECLA50
DECLA100	DECLA100-8	TOSHIBA	TOSHIBA132	HI	HP	HP6
HP8	HPLJ	HPLJ100	HPLJ150	HPLJ300		

3. Entrega de Resultados

Se debe entregar el listado del programa (fichero *.cir) y los resultados que se piden (sólo los que se piden). Estos resultados pueden ser numéricos, que se obtienen del fichero *.out, o bien las gráficas generadas por Probe.exe (o por cualquier otro programa graficador que use los datos numéricos del fichero *.out). Los listados, los resultados, todo, se debe entregar en papel, nada en disquete.

SPICE-WINDOWS

1. INSTALACION

Si se quiere instalar desde el disco duro copiar todos los ficheros a un único directorio. El proceso de instalación permite escoger el directorio de destino, y si se quiere, un tutorial. Una vez instalado, basta correr sólo el Schematic-Designer (PscheD.exe), pues cuando haga falta, éste llamará a Pspice y a Probe.

Este guión se ha hecho para la versión 6.2 de Spice. La versión 8 es muy parecida. Es muy recomendable no utilizar nombres largos, restringirse al convenio 8+3 (de DOS), y evitar espacios y caracteres extraños (p.ej. que tengan número de código mayor de 127). En unidades de red o removibles, puede ser imposible, para la versión 6, grabar en el directorio raíz, pero sí es posible grabar en un directorio de esa unidad.

Spice da la posibilidad de instalar un tutorial, con ejemplos, y métodos de trabajo. Los puntos más interesantes (Versión 6.2) son: 1 2 3 4 y 5 del menú SCHEMATICS, y 1 4 5 7 8 y 9 de SIMULATION.

2. CREACION DEL CIRCUITO: "SCHEMATICS"

Las explicaciones siguientes se refieren al Schematics-Designer (PschedD.exe):

Del menú **File**:

New	crea nuevo circuito
Open	abre circuito existente
Close	cierra circuito actual
Save . . .	graba circuito
SaveAs . .	graba con otro nombre

Del menú **Draw**:

Text	añade una linea de texto al circuito
Wire	cablea el circuito
GetNewPart . . .	añade al circuito un elemento

Del menú **Markers**: (añade markers de distinto tipo)

MarkVoltageLevel	ver la tensión en un nodo
MarkVoltageDiferencial	ver la diferencia de potencial entre dos nodos
MarkCurrentIntoPin . .	ver la intensidad que entra a un elemento
MarkAdvanced	markers especiales (de menor uso)

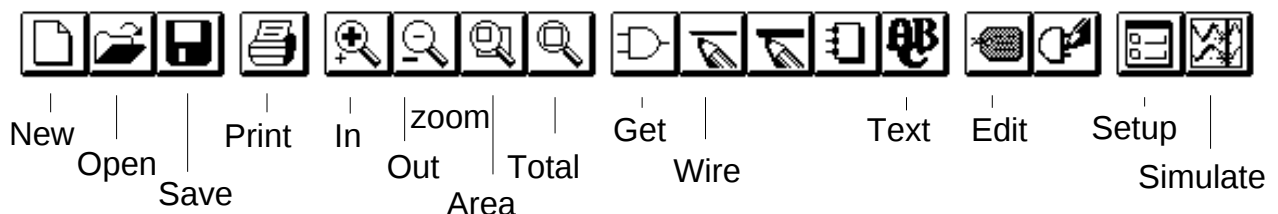
Del menú **Analysis**:

ElectricalRuleCheck .	comprueba que el circuito no tiene cables sueltos
CreateNetlist	crea una descripción del circuito
Setup	especifica el tipo de análisis
Simulate	corre Pspice : simula el funcionamiento del circuito
RunProbe	corre Probe : para ver los resultados de la simulación
ExamineOutput	ver el fichero con los errores, o con los resultados, si todo ha ido bien

Atajos de teclado útiles:

Ctrl-R	Rota (gira) el elemento seleccionado (rotar)
Ctrl-F	Reflexión especular del elemento (flip)
Ctrl-M	Pone Marker de Voltaje (marker)
Ctrl-L	Redibuja el circuito (redraw)

La Barra de Botones:



Clave Lo que hace Atajo de Teclado

New	Crea un circuito nuevo	
Open	Abre un circuito existente	
Save	Graba el circuito actual	Ctrl-S (save)
Print	Imprime circuito	
Zoom In	Amplía (respecto al centro)	Ctrl-I (in)
Zoom Out . . .	Reduce (respecto al centro)	Ctrl-O (out)
Zoom Area . .	Amplía al área deseada	Ctrl-A (area)
Zoom Total . .	Amplía a todo el circuito	Ctrl-N
Get	Añade elementos al circuito	Ctrl-G (get)
Wire	Cablea el circuito	Ctrl-W (wire)
Text	Añade etiquetas de texto	Ctrl-T (text)
Edit	Edita propiedades del objeto seleccionado	
Setup	Especifica el tipo de análisis en la simulación	
Simulate	Simula el funcionamiento del circuito	F11 (y F12 para Probe)

Elementos disponibles:

Cuando se piden elementos (p.ej. con Ctrl-G) aparece un cuadro donde se debe dar la denominación del elemento que se quiere añadir (más abajo doy los elementos más comunes). Si se quieren otros distintos, se pica el botón Browse y aparece a la derecha, una columna de librerías. Para cada librería picada, aparecen a la izquierda los elementos contenidos en ella.

Denominaciones de algunos elementos básicos:

R resistencia
C condensador
Agnd tierra (analog-ground)

Fuentes de tensión:

Vdc de continua (batería)
Vsrc arbitraria (V source). Fijar el valor de continua en el casillero DC.
Vsin senoidal. Fijar los valores de amplitud (VAMP), frecuencia (VFREQ) y offset (VOFF=0)
Vpulse cuadrada. Fijar los valores de mínimo (V1) máximo (V2) ancho del pulso (PW) periodo (PER) tiempo de caída (TF=0) tiempo de subida (TR=0) tiempo de retardo (TD=0).

D1N4002 diodo
Q2N2222 transistor bipolar NPN de propósito general
IRF150 mosfet de canal n (enriquecimiento)
IRF9140 mosfet de canal p (enriquecimiento)
uA741 amplificador operacional 741.
Viewpoint medidor de tensión continua en un punto.

Se pueden mover los elementos si se seleccionan (se pican) y se arrastra el ratón. En los elementos que forman parte del circuito, las propiedades se pueden modificar picando dos veces. En los dispositivos (diodos, transistores, etc.) los parámetros del modelo Spice se pueden modificar con "Edit→Model→EditInstanceModelText" (Util para fijar la tensión zener, o la β de un BJT ¡guardar *.lib!)

Pasos a seguir :

- 1º Construir el circuito (básicamente con Ctrl-G, Ctrl-W o los botones equivalentes)  
- 2º En Analysis_Setup se debe especificar el tipo de análisis - Siempre debe estar marcado (no tocar) el Bias Point Detail.
- Activar DC-Sweep si se quiere hacer un barrido en continua (una fuente cambia su valor de tensión continua en un rango de valores definido por el usuario).
- Activar Transient, si se quiere ver la evolución del circuito en función del tiempo.
- 3º Correr Pspice (y Probe): En la barra de menus Analysis_Simulate (F11) - Si hay errores, se pueden ver con "Analysis" y luego "ExamineNetlist" o "ExamineOutput" (Debajo del error debiera aparecer un "---\$").
- Si va bien empezará Probe (programa para ver gráficamente los resultados del circuito)

3. GRAFICADOR PROBE

Probe sólo aparece si se ha hecho algún análisis distinto del básico (Bias Point Detail), p.ej. un análisis temporal (Transient), o un barrido de continua (DC-Sweep). Si se hace sólo el básico, y se quieren ver las tensiones en los nodos, proceder así: Ctrl-G, escribir Viewpoint, y ponerlo en el nodo en el que se quiera ver la tensión. En la versión 8, es más fácil, pues basta pulsar el botón "V" para ver las tensiones en los nodos, y el botón "I" para ver las corrientes.

Para añadir una curva nueva sobre la gráfica actual se usa "Trace_Add". Se admiten variables y expresiones matemáticas de variables eléctricas. Se dispone de funciones como seno (sin), coseno (cos), tangente (tan), raíz cuadrada (sqrt), derivada (d), integral (s), potencia ($p(x,y)=|x|^y$), mínimo (min), máximo (max), valor absoluto (abs), exponencial (exp), logaritmo natural y decimal

(log y log10) y algunas otras más. Otra forma de añadir curvas a un gráfico, es añadir un Marker en el diagrama del circuito, automáticamente aparece la curva en Probe. Cada curva de un gráfico tiene un símbolo y una denominación, y si se desean borrar, se pica en la denominación y luego se pulsa "supr".

Las especificaciones de los ejes X e Y se cambian con "Plot_Xsettings" y "Plot_Ysettings". Si se desea añadir un nuevo gráfico, usar "Plot_AddPlot". Cuando a un gráfico se le quiere añadir una línea de texto, debe seleccionarse, y luego "Tools_Label_Text". Para ver los valores numéricos de una curva de la gráfica se usa "Tools_Cursor_Display" y luego se pica en el símbolo de la curva en la que se quiera ver el cursor (mover con el ratón y con las teclas de dirección) .

4. Otras Cuestiones

¿Qué se debe entregar?

Se debe entregar una copia en papel del dibujo del circuito, y los resultados que se piden (sólo los que se piden). Estos resultados pueden ser numéricos (texto) o gráficos. Las gráficas, los resultados, todo, se debe entregar en papel, nada en disquete.

¿Cómo imprimir? (En Probe y en Schematics-Designer)

Para sacar los circuitos y resultados de la simulación por impresora, no debería haber problema, si se tiene configurada una impresora en Windows. Al imprimir los circuitos, saldrá una hoja grande con recuadro, y (probablemente) el circuito en una esquina y de tamaño pequeño. Para evitar este inconveniente se pueden sacar imágenes (del circuito o de Probe) como se explica a continuación:

Imágenes vectoriales. Para sacar los gráficos como fichero vectorial, para luego integrarlos con el texto en un documento, se debe añadir en windows la impresora HP-ColorPro, modificando sus propiedades para que el puerto sea un fichero. Este fichero será de texto, con órdenes de ploter, que casi todos los procesadores de texto admiten (con extensiones plt, hpg o hgl, según el programa). Si el gráfico sale girado, cambiar en las propiedades del HP-ColorPro la orientación de papel. Luego desde Schematics o desde Probe, imprimir hacia la impresora (plotter) HP-ColorPro.

Imágenes binarias (o bitmapa).

Para capturar como bitmapa una imagen, puede recurrirse a cualquier método de captura de imágenes de pantalla. Por ejemplo "ImpPnt" imprime toda la pantalla, y "Alt-ImpPnt" sólo la ventana activa. También se pueden capturar con opciones de Pspice. Desde Schematics, se selecciona con una ventana la parte a imprimir. Luego se hace Alt_Edit_Copy to Clipboard. Desde Probe: Alt_Tools_Copy to Clipboard.

Otra opción sería utilizar las posibilidades de captura en Paint-Shop-Pro u otros programas similares.

En cualquier caso el bitmapa conseguido se almacena en el portapapeles "clipboard", y puede ser recuperado en un procesador de texto o programa de dibujo con Ctrl-V.

Para no imprimir el fondo negro, y que los colores (o grises) de impresión sean más visibles, se puede hacer lo siguiente en Paint-Shop-Pro: "Alt_Colors_Decrease-Color-Depth_16-Colors", luego "Alt_Colors_Negative-Image", y finalmente "Alt_Colors_Edit-Palette". De esta manera, el fondo queda blanco, y los colores finales del gráfico se pueden editar a gusto (con "Edit-Palette")

¿Qué ficheros guardar?

De todos los ficheros que salen tras una simulación (*.cir *.sch *.als *.plb ...) sólo son interesantes los que contienen el circuito (el schematics), es decir los acabados en .sch. Si se ha modificado algún dispositivo, se deben guardar los ficheros *.lib. Las imágenes generadas (bmp, gif, hpg) también deben ser guardadas. Cuando se quieran generar el resto de ficheros, simplemente volver a simular el circuito.