



CENTRO DE FORMACIÓN PROFESIONAL
REVILLAGIGEDO – Jesuitas (Gijón)

Mariano Pola, 46 - Tel. 985 32 13 15 - Fax 985 32 22 30
33212 Gijón - <http://www.revillagigedo-sj.org>

LTspice IV

José Manuel Fernández García

Software.

El programa se puede descargar, gratuitamente, de la página oficial de Linear Thechnology:

www.linear.com

Directorios y Archivos de LTspice.

Por omisión, el LTspice se instala en:

C:\Archivos de programa\LTC\LTspiceIV

En el raíz de LtspiceIV aparecen diversos archivos, entre los que se encuentra el ejecutable de la aplicación: `scad3.exe`, así como dos carpetas: "examples" y "lib"

La carpeta "examples" se subdivide en otras subcarpetas que contienen diversos ejemplos de circuitos para simular.

La carpeta "lib" contiene tres subcarpetas:

- `cmp` Contiene diversas librerías de componentes que corresponden a semiconductores de Spice, como diodos, bjt, mosfets, etc.
- `sub` Contiene librerías de componentes subcircuitales.
- `sym` Contiene símbolos de componentes

Todas las librerías y archivos de símbolos son archivos de texto, incluidos los archivos que contienen los circuitos de simulación (extensión `.asc`)

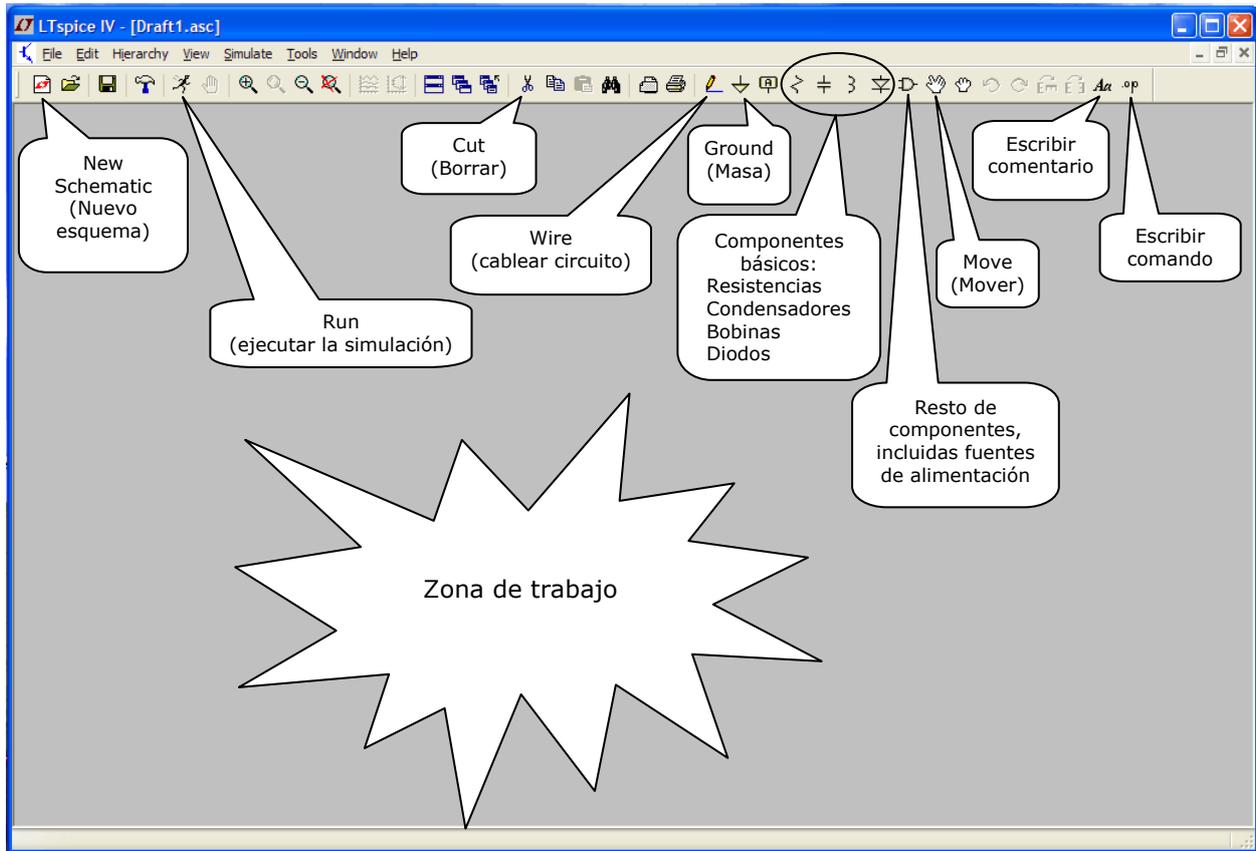
Archivos de Texto de LTspice.

Todos los archivos de texto han de comenzar por una línea de comentario, que por ser la primera no es necesario que se preceda del asterisco. Si hubiese más líneas de comentario, sí deberán de ser precedidas por un asterisco, o un punto y coma.

Esto es así, porque sino LTspice interpretaría la primera línea como un comando.

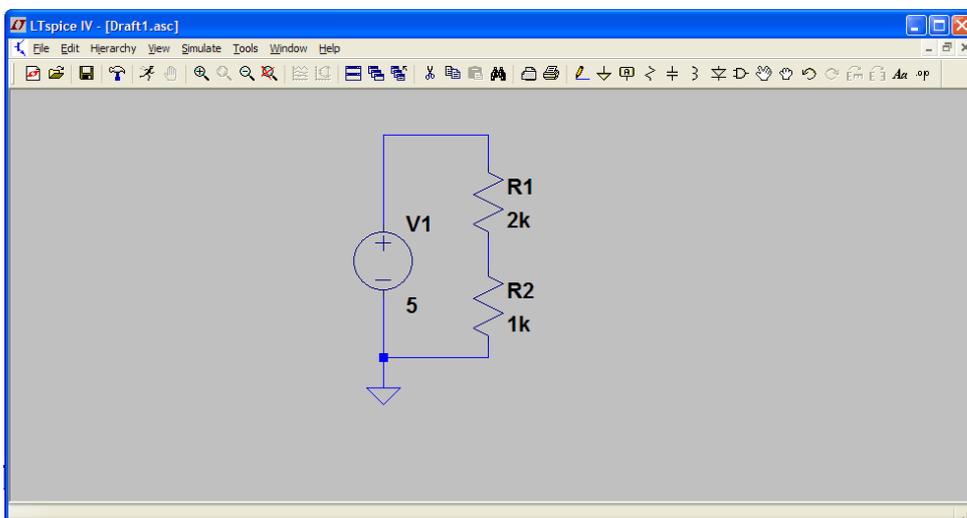
Entorno Gráfico.

Cuando entramos en el LTspice nos encontramos con un agradable entorno gráfico como el que se muestra en la siguiente imagen, en la que se ha indicado las herramientas básicas.



Nada más entrar en el entorno LTspice, lo primero que hacemos es abrir la zona de trabajo con "Nuevo esquema". Seguidamente, situamos los componentes en la zona de trabajo y procedemos a cablear. Finalmente, ejecutamos la simulación (Run).

Supongamos que vamos a simular un circuito tan simple como un divisor de tensión resistivo en vacío, formado por una resistencia de 2K y otra de 1K, alimentado con una tensión de 5V (corresponde al componente "voltage")



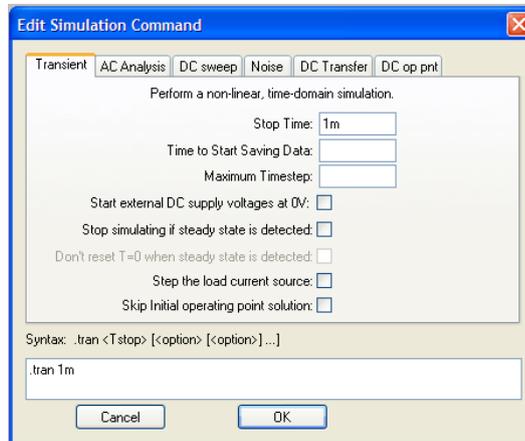
Tras situar los componentes, hacemos clic con el botón derecho sobre ellos para proceder a dimensionarlos.

Seguidamente, procedemos a ejecutar la simulación (Run).

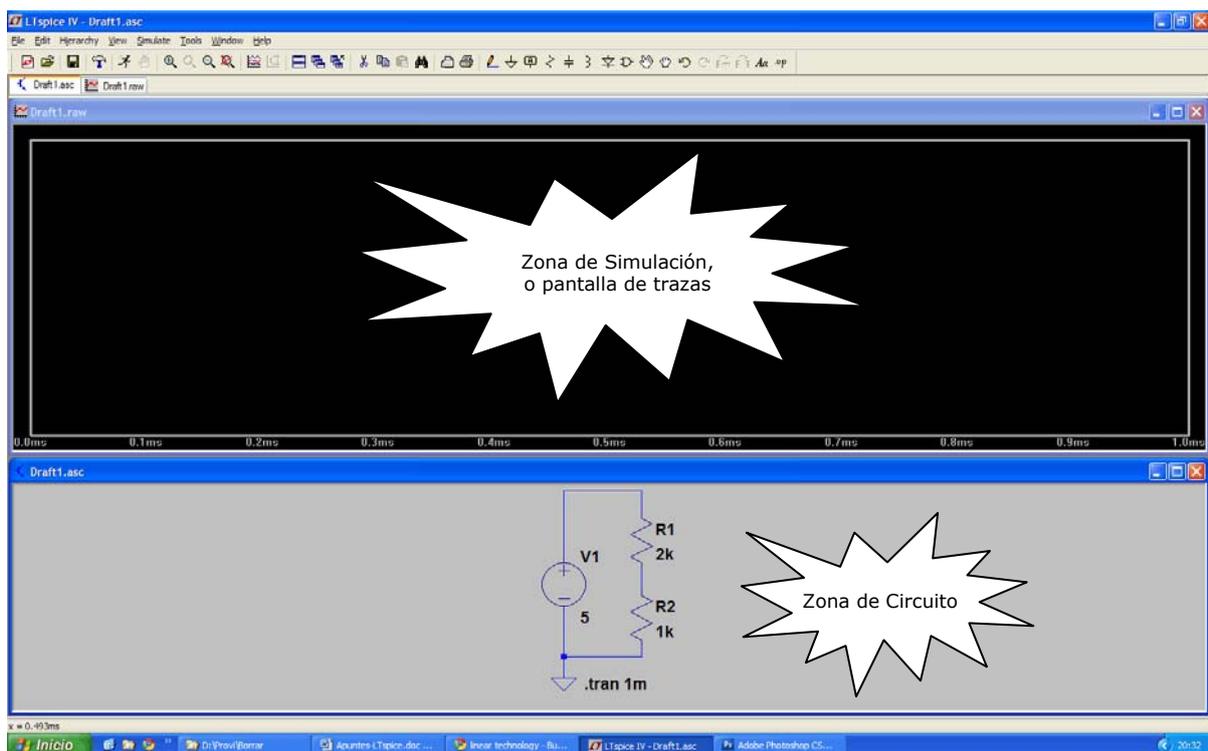
Lo primero que nos aparece es la ventana del editor de simulación, en el que hemos puesto 1m (significa 1ms) en Stop Time, lo que indica que la simulación se hará a lo largo de 1ms. En este caso, como se trata de un circuito de continua, no tendría importancia haber utilizado cualquier otro tiempo. Hemos escogido una pequeña para mayor rapidez de cálculos en la simulación.

Vemos cómo en la parte inferior de la ventana aparece, automáticamente, el comando: `.tran 1m`, que a su vez queda escrito en el propio circuito.

Naturalmente, esta ventana no hubiese aparecido si hubiésemos escrito directamente este comando mediante la opción o herramienta `.op`



Una vez aceptada esta ventana, se nos abre la ventana de simulación junto con la de circuito:



Sin más que pasar el cursor por los elementos del circuito vemos que éste puede adoptar la forma de una pica de prueba o de una pinza amperimétrica.

Cuando adopta la forma de pica de prueba, si hacemos clic en ese lugar, aparecerá en la zona de simulación la curva (en este caso una recta) correspondiente a la tensión en ese punto respecto de masa. Cuando adopta la forma de pinza amperimétrica, nos aparecerá la curva (de nuevo, en este caso, se tratará de una recta) correspondiente a la corriente eléctrica.

En cualquier momento podemos conocer el nombre de un nodo sin más que pasar por encima de él el cursor, apareciéndonos dicho nombre en la parte inferior izquierda de la pantalla.

Nodo de Masa.

El nodo de masa siempre es el **0**

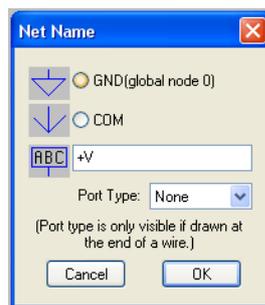
Los demás nodos se van numerando, automáticamente, como: N001, N002, etc.

Utilizando Etiquetas para Nodos (Label Net).

Puede ser útil denominar a ciertos nodos con un nombre que nos diga algo, como, por ejemplo: Salida, Entrada, +V, etc.

Para ello bastará con elegir la herramienta Label Net, que aparece simbolizada por el icono 

Con esto se nos abre una ventana, en la que en el campo marcado como  escribiremos el nombre que deseamos dar a la etiqueta (en nuestro ejemplo: +V)



Tras aceptar con OK, situamos la etiqueta (nos queda pinchada en el cursor) en cuantos puntos queramos asignar al mismo nodo +V. Con esto, no solo nos queda mejor identificado el circuito, si no que evitamos tener que cablear desde unos puntos a otros, a veces distantes, ya que todos aquellos a los que se les ha asignado la misma etiqueta pertenecerán al mismo nodo.

Netlist.

El archivo de texto del circuito se puede ver en: View / SPICE Netlist

Por ejemplo, el divisor de tensión que hemos visto al tratar el entorno gráfico, tendría la siguiente Netlist:

```
* C:\Archivos de programa\LTC\LTspiceIV\Draft1.asc
V1 N001 0 5
R1 N001 N002 2k
R2 N002 0 1k
.tran 1m
.backanno
.end
```

La primera línea, precedida de un asterisco, es sólo un comentario, donde por omisión aparece la ruta. En la segunda línea vemos que la fuente de tensión V1 está conectada entre los nodos N001 y masa, y tiene una tensión de 5V

La tercera línea especifica que la resistencia R1 de 1K está conectada entre los nodos N001 y N002

La quinta línea indica que se hace un análisis durante 1ms

Sentido Positivo de las Corrientes.

LTspice supone un sentido de la corriente, que se visualiza al pasar el cursor, en forma de pinza amperimétrica, por encima del terminal del componente. El que la medida de la corriente aparezca como positiva o negativa en la zona de simulación, dependerá de si su sentido real coincide o no, respectivamente, con el sentido supuesto.

Análisis en Continua o Régimen Permanente.

Si no se dice lo contrario, LTspice presenta los resultados de un análisis a partir del régimen permanente o régimen de continua.

`.op` → punto de funcionamiento. Indica que se hace un análisis en continua o régimen permanente.

Análisis Transitorio.

Supongamos que queremos simular un circuito serie RC, con 1K y 22 μ F, alimentado con una tensión CC de 10V, con el fin de ver la curva de crecimiento exponencial de la tensión en el condensador.

En condiciones normales, LTspice comienza el análisis a partir del régimen permanente (punto estático de funcionamiento), por lo que veríamos en la tensión del condensador la misma tensión de alimentación.

Para hacer el análisis transitorio que deseamos, es preciso indicar otras condiciones iniciales mediante la directiva UIC (Use Initial Conditions). En concreto indicaríamos que el condensador parta de una tensión cero.

La forma de proceder sería la siguiente: Dibujamos el circuito y asignamos valores.

Seguidamente, escribimos el texto-directiva o comando: `.tran 0 2 0 UIC` y procedemos a simular.

Añadir nuevas Trazas.

Por omisión LTspice visualiza trazas de V e I. Si se desea visualizar otras trazas habrá que utilizar la directiva Add.Trace

Por ejemplo, si deseamos ver la traza de la potencia en una resistencia R1, conectada entre los nodos N001 y N002, procederíamos de la siguiente forma: Comenzamos la simulación y sobre la pantalla de trazas hacemos clic con el derecho. En el menú desplegable que nos aparece elegimos Add Trace y en el campo: Expression(s) to Add escribimos:

$$V(N001,N002)*I(R1)$$

Valores Medio y Eficaz.

Si hacemos clic con el izquierdo, a la vez que mantenemos pulsado CTRL, sobre el nombre de una traza, nos aparece información sobre el intervalo de análisis y ciertos valores a lo largo de dicho intervalo.

Por ejemplo, si actuamos sobre una traza de tensión, nos aparecerán los límites del intervalo de tiempo mediante Interval Start e Interval End, así como el valor medio, en Average, y el eficaz en RMS.

Si Actuamos sobre una traza de potencia, nos aparecen los valores medio (Average) y el Integral. El valor integral se refiere a energía, correspondiendo al producto del valor medio (Average) por el tiempo de ensayo (Interval End – Interval Start).

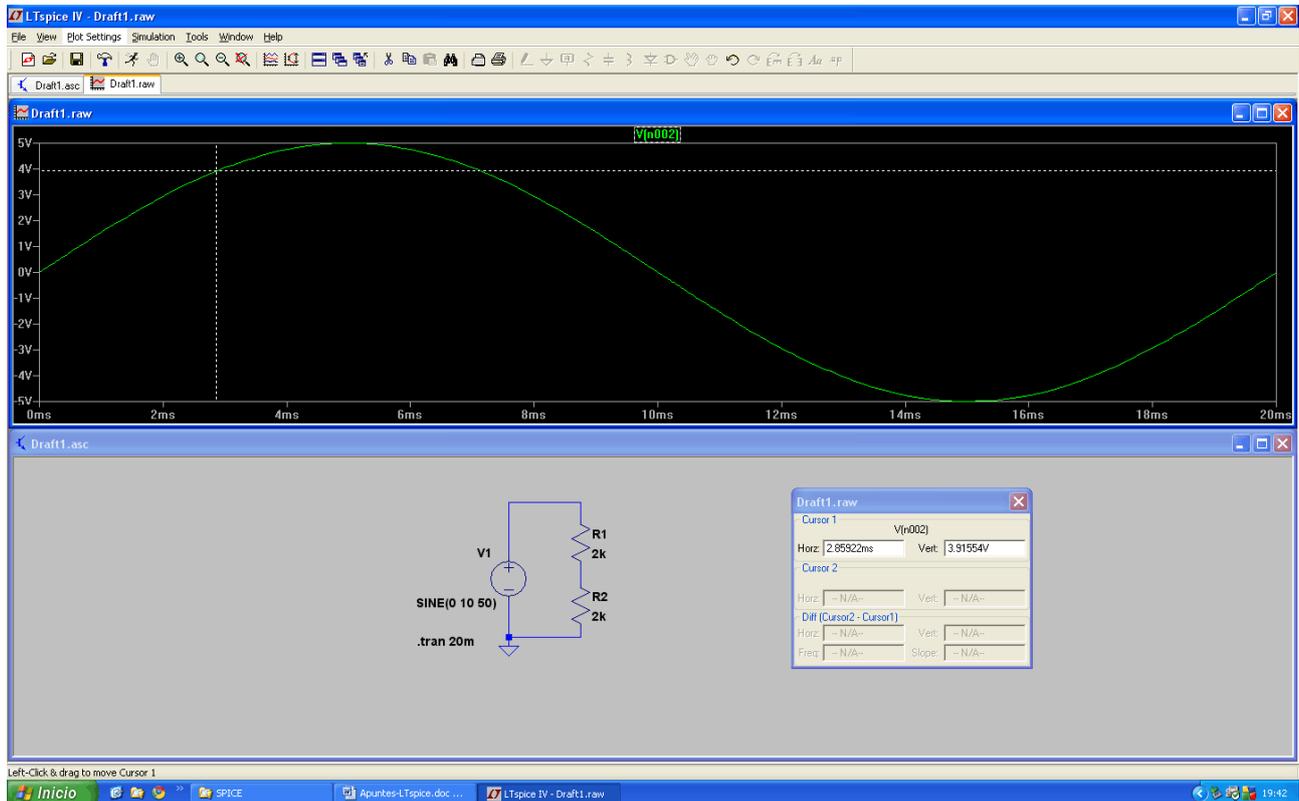
Con las mediciones de potencia hay que tener especial cuidado cuando su traza tenga valores negativos, pues dado que se trata de un valor medio podemos llegar a absurdos si no tomamos la precaución de considerar el valor absoluto.

Establecer Límites Manuales.

En ocasiones podemos necesitar ver, y evaluar sus valores medio y/o eficaz a partir de un cierto instante. Para ello hacemos clic con el derecho sobre la ventana de trazas y elegimos la opción: Manual Limits y en la parte inferior de la ventana en Horizontal Axis en el campo Left podemos indicar a partir de qué instante vamos a visualizar las trazas. Naturalmente, si la onda es periódica esto no influirá en los valores medio y eficaz, pero sí lo hará en el caso, por ejemplo, de que una traza tuviese un pico elevado al principio y luego la señal se estabilizase en una señal periódica.

Efectuar Medidas sobre Trazas.

Una vez que estamos simulando, y sobre la zona de trazas, vemos que en la parte superior aparecen los nombres de las trazas que tenemos en la pantalla. Los nombres, también, aparecen en el mismo color que las trazas. Si hacemos clic sobre uno de los nombres (el cursor está en forma de mano), veremos que nos aparecen dos ejes perpendiculares entre sí, de color blanco y a trazos. Pinchamos sobre el eje vertical (aunque pinchemos sobre el horizontal, engancharíamos automáticamente el vertical) y vemos cómo el punto de cruce de ambos ejes se desplaza a lo largo de la traza, a la vez que en la ventana que se nos ha abierto podemos leer directamente las coordenadas de los puntos por los que nos vamos desplazando, tanto del eje horizontal, que será de tiempos, como del vertical que será, por ejemplo, una tensión, una corriente, etc.



Transformadores.

Para simular un transformador se colocan dos bobinas, por ejemplo, L1 y L2, cuyos valores se eligen de acuerdo con la relación de transformación que se desee para el transformador. Basta recordar que el coeficiente de autoinducción, para una bobina recta, viene dado por la ecuación:

$$L = \frac{4 \cdot \pi \cdot N^2 \cdot S \cdot \mu}{l}$$

Siendo:

L (coeficiente de autoinducción de la bobina); N (número de espiras);
S (Sección del núcleo); μ (permeabilidad del núcleo); l (longitud de la bobina)

Sin más que tener en cuenta que, en un transformador, ambas bobinas comparten el mismo núcleo, se cumple: $L = K \cdot N^2$

Según esto:

$$\frac{L1}{L2} = \frac{N1^2}{N2^2} \rightarrow \frac{N1}{N2} = \sqrt{\frac{L1}{L2}} \rightarrow r_t = \sqrt{\frac{L1}{L2}}$$

En el menú elegimos **.op** y escribimos la directiva: **K1 L1 L2 1** con lo que indicamos que las bobinas L1 y L2 se encuentran acopladas con un coeficiente de acoplamiento igual a 1 (corresponde al acoplamiento máximo o ideal. En la práctica este coeficiente es siempre menor de 1). Si hubiese más grupos de bobinas acopladas, los denominaríamos: K2, K3, etc. Por ejemplo, si un grupo de acoplamiento tuviese tres bobinas, se escribiría una directiva del tipo siguiente: K1 L1 L2 L3 1

Análisis con Barrido de Frecuencias.

En ocasiones es preciso realizar el análisis de un circuito viendo su respuesta a una cierta gama de frecuencias. Por ejemplo, partamos de un simple filtro paso bajo formado por una fuente de tensión de alterna, una resistencia y un condensador.

Vamos a Simulate y en el menú desplegable elegimos Edit Simulation Cmd, seguidamente entramos en la pestaña AC Analysis y elegimos el Type of Sweep, por ejemplo, Octave. Elegimos, también, los puntos de ensayo por octava, por ejemplo 10. El margen de frecuencias para el que deseamos hacer el análisis, se indica mediante Star Frequency y Stop Frequency. Por ejemplo, podemos hacer Star Frequency = 10 y Stop Frequency = 10MEG, para hacer un barrido desde 10 Hz hasta 10 MHz

Debemos indicar la amplitud de la tensión que se va a emplear para el análisis (aparte del que ya tuviese asignado la fuente de tensión) haciendo clic con el derecho sobre el generador de tensión y en la zona de Small signal AC analysis(.AC), en el campo de AC Amplitude, escribimos el valor.

Simulamos y llevamos una sonda sobre el nodo superior del condensador, con lo que nos aparecerá el correspondiente diagrama de Bode: Se nos mostrarán dos trazas: una continua correspondiente al diagrama de ganancia en función de la frecuencia, y otra a trazos que se corresponde con el desfase en función de la frecuencia.

Curva Característica de un Diodo.

Dibujamos un circuito serie formado por una fuente de intensidad I1 (vamos en el menú a **component** y escogemos **current**) y el diodo con su cátodo a masa. Supongamos que vamos a ensayar el 1N4148 hasta un máximo de 100mA. Haciendo clic con el derecho sobre la fuente, le asignamos un valor de 100mA.

Vamos a Simulate y elegimos Edit Simulation Cmd. En la pestaña DC sweep, en 1st Source, escribimos I1 en el campo de Name of 1st Source to Sweep. En Type of Sweep elegimos Linear. En Start Value situamos un 0. En Stop Value escribimos 100mA, y en Increment escribimos 0.5mA. A la vez que vamos escribiendo estos parámetros vemos cómo se va escribiendo, en la parte inferior de la ventana, el comando **.dc I1 0 100mA 0.5mA**

También podríamos haber escrito directamente dicho comando, sin necesidad de haber ido al editor de simulación.

Simulamos visualizando la tensión en el ánodo del diodo. Veremos que nos sale la curva característica del diodo con el eje de intensidades en abscisas.

Curvas Características de Salida de un Transistor.

Dibujamos un circuito con un transistor, una fuente de intensidad I1 (vamos en el menú a **component** y escogemos **current**) entre base y emisor, y una fuente de tensión continua V1 de, por ejemplo, 10V entre el colector y el emisor.

Vamos a Simulate y elegimos Edit Simulation Cmd. En la pestaña DC sweep, en 1st Source, escribimos V1 (la fuente que pongamos como primera, corresponderá al parámetro del eje de abscisas) y cumplimentamos los siguientes campos:

Name of 1st Source to Sweep	V1
Type of Sweep	Linear
Start Value	0
Stop Value	10
Increment	0.1

Seguidamente, definimos como segunda fuente al generador de intensidad I1. Para ello, pasamos a la pestaña 2nd Source y cumplimentamos los siguientes campos:

Name of 2nd Source to Sweep	I1
Type of Sweep	Linear
Start Value	0
Stop Value	5m
Increment	1m

Una vez introducidos estos parámetros, vemos cómo aparece, en la parte inferior de la ventana, el comando **.dc V1 0 10 0.1 I1 0 5m 1m**

También podríamos haber escrito directamente dicho comando, sin necesidad de haber ido al editor de simulación. Simulamos visualizando la corriente de colector. Veremos que nos salen las curvas características de salida del transistor.

Recta de Carga.

Podemos dibujar la recta de carga sobre las curvas características de salida, antes determinadas. Para ello supongamos que polarizamos el circuito de colector con una tensión de 10V y que colocamos en el colector una resistencia de 50Ω

Sobre la ventana de simulación en la que tenemos las curvas de salida, hacemos clic con el derecho para añadir una nueva traza (Add Trace). En la parte inferior de la ventana (campo de Expression(s) to add:) escribimos la expresión:

(10- V(n001))/50

En nudo n001 corresponde al colector.

Listado de Trazas.

En ocasiones es útil tener una lista de las trazas y poder visualizar unas concretas.

Hacemos clic con el derecho sobre la pantalla de trazas y elegimos la opción Select Steps. Nos aparece la ventana Select Displayed Steps, con una lista de las trazas. Para visualizar una concreta bastará con seleccionarla. Para visualizar varias iremos seleccionándolas mediante clic izquierdo a la vez que mantenemos pulsado CTRL. También tenemos la opción de seleccionar todas las trazas (botón Select All).

Análisis en Temperatura.

Supongamos que queremos ver las variaciones de la curva característica del diodo ante variaciones de temperatura; por ejemplo, para temperaturas de 20°C , 100°C y 125°C

Para ello, bastará incluir, en la misma simulación que hemos realizado para la curva de un diodo, el comando siguiente:

.step temp list 20 100 125

Seguidamente vamos a RUN y colocamos una sonda de tensión en el ánodo del diodo.

El mismo análisis en temperatura lo podríamos realizar con las curvas de salida de un BJT. Para ello, tras aplicar RUN, colocaríamos una sonda de corriente en el colector del transistor.

Si se observan problemas en la simulación, pruébese a volver a aplicar RUN con la pantalla de trazas abierta.

Simular un Potenciómetro.

Para simular un potenciómetro de 5K dibujaremos en serie dos resistencias R1 y R2 . A una de ellas le asignamos el valor: {5K-mando} y la otra: {mando}

El término "mando" puede ser sustituido por otro cualquiera.

Escribimos el comando: **.Step Param mando List 1K 2K 3K 4K**

Para simular hemos de configurar el editor de simulación. Para ello, podemos cubrir los correspondientes campos con los siguientes valores:

Stop Time:	1
Time Start Saving Data:	0
Maximun Timestep:	0.1

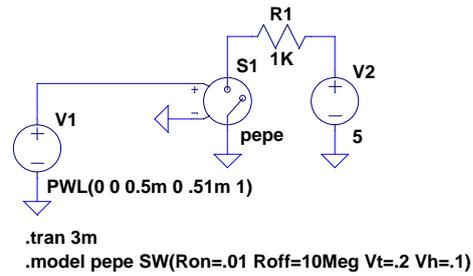
Al introducir estos valores veremos cómo en la parte inferior de la ventana del editor de simulación se genera el comando: **.tran 0 1 0 0.1** que, también, podríamos haber escrito directamente como un texto de comando (icono .op)

El resultado de la simulación son cuatro líneas horizontales correspondientes a los cuatro valores de la tensión en el punto medio del divisor (donde colocamos la sonda de tensión) correspondientes a los cuatro valores especificados en List

Otra forma de presentar el resultado de la simulación del potenciómetro podría haber sido escribiendo el comando **.op** (o configurando el editor de simulación en modo: DC op pnt) , en vez del **.tran 0 1 0 0.1**

En este caso aparece una traza que nos muestra, de forma continua, la tensión en función de la variación del parámetro, en nuestro caso, mando.

Simular el Instante de Cierre de un Interruptor.



Utilizamos un interruptor controlado por tensión (S1) y lo controlamos mediante una fuente de tensión PWL definida para producir una señal que hemos definido por los siguientes puntos: En el instante $t=0$ el valor es de 0V, en el instante $t=0.5ms$ el valor es de 0V, en el instante $t=0.51ms$ el valor es de 1V (se sobrentiende que a partir a aquí se mantendrá en 1V). El interruptor lo definimos para que cerrado tenga una resistencia de 0.01Ω y abierto de $10M\Omega$, a la vez que indicamos que cierre cuando la tensión de mando alcance el valor de 0.3V (correspondiente a V_t+V_h) y que se abra cuando la tensión de mando sea de 0.1V (correspondiente a V_t-V_h)

Trabajando con Archivos WAV.

Podemos introducir un archivo WAV como señal de entrada a un circuito y generar otro archivo WAV como salida.

Por ejemplo, supongamos que tenemos un circuito de audio al que vamos a introducir como señal de entrada el archivo **entrada.wav** y la respuesta la vamos a obtener como la tensión en el nodo N005, respuesta que salvaremos en un archivo al que denominaremos **salida.wav**. Se ha de especificar el número de bits de cada muestra (bits de sampleo), por ejemplo: 16 , así como la frecuencia de muestreo, por ejemplo, 44.1 KHz

Editamos el circuito y sacamos una fuente de tensión, por ejemplo V1, de value: V ; hacemos clic con el derecho sobre V y escribimos el texto:

```
WAVEFILE=entrada.wav
```

Seguidamente escribimos el comando:

```
.WAVE salida.wav 16 44.1k V(N005)
```

Como no hemos especificado ruta alguna para el archivo de salida, éste se salvará en el mismo directorio de trabajo en el que tenemos el circuito y el archivo de entrada. No obstante, se podría especificar una ruta cualquiera (incluyendo unidad).

Varistor.

En el bloque de componentes [SpecialFunctions] nos encontramos con el Varistor. Se trata de un componente de cuatro patillas.

Una vez sacado el componente se hace clic sobre él y en la ventana de "Componente Attribute Editor", en el campo Value, se le asigna el parámetro "Rclamp" con un valor, por ejemplo: Rclamp=1

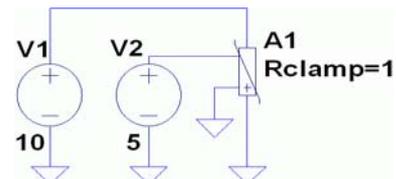
El circuito siguiente muestra el funcionamiento básico de este componente.

Si $V1 \leq V2$, entonces el Varistor actúa como un circuito abierto:

$$I(V1) = 0$$

Si $V1 > V2$, entonces el Varistor se hace conductor, actuando como una fuente de tensión V2 con una resistencia interna Rclamp:

$$I(V1) = \frac{V1 - V2}{Rclamp}$$



En todos los casos, la corriente de control es nula: $I(V2) = 0$

Funciones Lógicas.

Por omisión, las puertas proporcionan una tensión de salida entre 0 y 1V. Como quiera que esto no es lo usual con que vamos a trabajar (lo más frecuente será trabajar con niveles TTL, entre 0 y 5V), bastará con indicar explícitamente el parámetro de tensión. Para ello, haremos clic con el derecho sobre el símbolo de la puerta lógica, con lo que nos aparece la ventana "Component Attribute Editor" y sin más que hacer doble clic en el campo vacío de Value, escribiremos el valor que deseamos para el nivel alto, asignándolo al parámetro Vhigh

Las entradas que no utilicemos las dejamos sin conectar, al igual que la masa (inferior, central).

PULSE(0 5 1m 5n 2n 1m 2m)

V1

V2

PULSE(0 5 1.5m 5n 5n 1.5m)

.tran 3m

A1

Component Attribute Editor

This is the second attribute to appear on the netlist line.

Attribute	Value	Vis.
Prefix	A	
InstName	A1	X
SpiceModel	OR	
Value	Vhigh=4.5	
Value2		
SpiceLine		
SpiceLine2		

En este caso hemos asignado una tensión de 4,5V

Siguiendo este mismo procedimiento podemos asignar valores a otros parámetros, tales como la tensión correspondiente al nivel bajo (Vlow), el tiempo de subida (Trise), etc. Para esto conviene ir a la ayuda

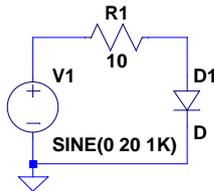
allows one device to act as 2-, 3-, 4- or 5- input gates with true, inverted, or complementary output with no simulation speed penalty for unused terminals. That is, the AND device acts as 12 different types of AND gates. The gates default to 0V/1V logic with a logic threshold of .5V, no propagation delay, and a 10hm output impedance. Output characteristics are set with these instance parameters:

Name	Default	Description
Vhigh	1	Logic high level
Vlow	0	Logic low level
Trise	0	Rise time
Tfall	Trise	Fall time
Tau	0	Output RC time constant
Cout	0	Output capacitance
Rout	1	Output impedance
Rhigh	Rout	Logic high level impedance
Rlow	Rout	Logic low level impedance

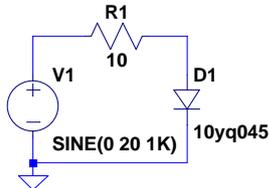
Trabajando Con Modelos Diferentes a los Estándar del LTspice.

Supongamos que tenemos que trabajar con un componente que no figura en las librerías estándar del LTspice, como puede ser el diodo 10yq045. Seguimos los siguientes pasos:

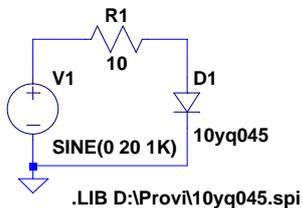
1. Buscamos en Internet el modelo del diodo 10yq045. Supongamos que lo encontramos en el archivo: 10yq045.spi
2. Guardamos el archivo, por ejemplo en: D:\Provi\10yq045.spi
3. Entramos en el LTspice y dibujamos el circuito con el diodo 10yq045



4. Hacemos clic con el derecho sobre la D del diodo, y en la ventana que se nos abre escribimos el nombre del nuevo diodo en sustitución de la D



5. Finalmente, ya sólo resta escribir el comando .LIB que nos dirija a la librería en cuestión



NOTA: No es necesario escribir toda la ruta de la librería, si la guardamos en la misma carpeta en la que tenemos el circuito. En este caso, y siguiendo con el ejemplo anterior, hubiese bastado con el comando: `.LIB 10yq045.spi`

Subcircuitos.

A un circuito se le puede asignar un símbolo (sólo necesario si se va a trabajar con entorno gráfico) y utilizarlo como si de un componente se tratase. Otra situación en la que se utilizan subcircuitos es cuando no disponemos de un componente determinado en nuestras librerías y al buscarlo en Internet nos lo encontramos en forma de subcircuito.

Vamos a ver algunos ejemplos de esta situación. Los siguientes pasos pueden modificarse, y sólo representan un ejemplo de cómo se procedió en estos casos.

Ejemplo1.

Supongamos que queremos simular un simple circuito con el MOST de canal N: IRF740, que no lo tenemos en las librerías de nuestro LTspice.

En primer lugar buscamos en la red este dispositivo. Para ello es cómodo utilizar el Google. Escribimos, por ejemplo, en la búsqueda: **IRF740 spice model**

Nos descargamos, en este caso, un archivo de texto denominado: **nemos3t.mod**

Abrimos el archivo, por ejemplo: con el Bloc de Notas, y nos encontramos con una lista de numerosos dispositivos. Borrarnos lo que no nos interesa del archivo, o simplemente copiamos en el portapapeles lo que queremos y formamos con ello otro archivo.

Podemos salvar la cabecera y la parte en la que se alude al IRF740. De esta forma nos queda el archivo de texto siguiente, donde hemos realizado un pequeño cambio: hemos añadido el terminal Sus (sustrato) en la cabecera, pues aunque ésta no es procesable (sólo es comentario) sirve de orientación para conocer el orden en que están asignadas las patillas del transistor. De esta forma, creamos el subcircuito para cuatro terminales y nos servirá para otros casos en los que la fuente no venga conectada al sustrato (cosa que sí sucede con el IRF740)

En definitiva, sustituimos la línea: ***N-EMOS 3T pinout: D,G,S** por la que podemos ver en nuestro archivo ya modificado:

```
*=====
*N-EMOS 4T pinout: D,G,S,Sus
*Copyright (c) 1996 MicroCode Engineering, Inc.
*All Rights Reserved
*=====
```

```
*IRF740 MCE 4-2-96
*400V 10A .55 ohm HEXFET pkg:TO-220 2,1,3
.SUBCKT XIRF740 10 20 40 40
M1 1 2 3 3 DMOS L=1U W=1U
RD 10 1 .26
RS 30 3 14.7M
RG 20 2 15
CGS 2 3 1.28N
EGD 12 0 2 1 1
VFB 14 0 0
FFB 2 1 VFB 1
CGD 13 14 1.54N
R1 13 0 1
D1 12 13 DLIM
DDG 15 14 DCGD
R2 12 15 1
D2 15 0 DLIM
DSD 3 10 DSUB
LS 30 40 7.5N
.MODEL DMOS NMOS (LEVEL=3 THETA=60M VMAX=833K ETA=2M VTO=3 KP=4.95)
.MODEL DCGD D (CJO=1.54N VJ=.6 M=.68)
.MODEL DSUB D (IS=41.5N N=1.5 RS=.125 BV=400 CJO=903P VJ=.8 M=.42 TT=370N)
.MODEL DLIM D (IS=100U)
.ENDS XIRF740
```

Podemos ver que se trata de un subcircuito por la línea: **.SUBCKT XIRF740 10 20 40 40**

Son indicativos de subcircuito, la directiva: **.SUBCKT** y la **X** precediendo al nombre del dispositivo.

Este archivo lo salvamos, por ejemplo, como **IRF740.lib**, y lo hacemos en una carpeta a la que denominamos, por ejemplo, **mosfet1**. De esta forma todos los archivos estarán en la misma carpeta y evitamos el escribir rutas.

El siguiente paso (sólo porque vamos a utilizar el entorno gráfico) será asignar un símbolo a este subcircuito. Para ello, lo que podemos hacer es utilizar uno de los símbolos para un nMOST de los que ya tengamos. Para ello podemos a donde tengamos instalado el LTspice, e ir a la carpeta Sym y buscamos un archivo de símbolo que corresponda con un transistor nMOST. Nos encontramos el **nmos.asy**, hacemos una copia de él, a la que denominamos Xnmos.asy y sobre la que vamos a trabajar. Hacemos doble clic sobre Xnmos.asy y se nos abre el Editor de Símbolos del LTspice.

Borramos la línea que une el sustrato con la fuente y añadimos el terminal o nodo de sustrato: Vamos a Edit y elegimos la opción: Add Pin/Port. En el cuadro de propiedades vemos que le asigna el 4 al Netlist Order, cosa que está bien; añadimos en Label la denominación del sustrato: **Sus** y el resto lo dejamos tal cual está (no visible), por lo tanto aceptamos con OK y procedemos a situar el terminal de sustrato. Seguidamente, dibujamos las líneas que nos unan los nodos de sustrato y de fuente (nos había quedado desconectado al borrar la conexión de fuente-sustrato) con el resto del símbolo, usando para ello Draw / Line

Conviene, también revisar el orden de los pines. Para ello hacemos clic con el derecho sobre los nodos y comprobamos que esté todo correcto. Observamos la siguiente correspondencia entre Label y el número de Netlist Order:

D	1
G	2
S	3
Sus	4

Vemos que está todo correcto, pues coincide con el orden que figura en el archivo, antes creado, de acuerdo con la línea:

***N-EMOS 4T pinout: D,G,S,Sus**

Seguidamente vamos a Edit / Attributes / Edit Attributes, con lo que se nos abre la ventana del Symbol Attribute Editor.

En el campo Symbol Type podemos elegir entre Cell y Block, elegimos Cell (Block corresponde a un símbolo de bloque y Cell a un símbolo convencional). En Prefix asignamos una X (ya que se trata de un subcircuito). En Value lo dejamos en blanco (borramos el término NMOS que tiene), de esta forma lo modificamos, en su momento, en el esquema; así nos queda válido para otros subcircuitos. No obstante, se podría escribir, por ejemplo, el nombre del dispositivo (IRF740).

En Description (no es más que un comentario) podemos escribir: Subcircuito de NMOS

Lo demás lo dejamos en blanco y aceptamos con OK

En el símbolo que estamos editando vemos que aparece el texto: Unnn. Este es un asunto que no está claro, pues luego en la Netlist nos sale como referencia del subcircuito: XU1, cuando en realidad debería ser más bien X1. De todas formas, funciona bien así.

Con todo esto ya tenemos el símbolo acabado. Cerramos y aceptamos salvar. Ya podemos ir al simulador y buscar nuestro componente-subcircuito: Xnmos.asy

Dibujamos el circuito con el IFR740. Hacemos clic derecho sobre el símbolo del MOST y en Value escribimos: XIRF740 (el resto lo dejamos como está).

En el menú elegimos el icono **.op** y escribimos la directiva: **.LIB IRF740.LIB**, con lo que estamos indicando que el modelo se encuentra en la librería IRF740.lib

Antes de simular hemos de salvar el circuito, y lo hacemos en el mismo directorio en el que estamos guardando todo (carpeta **mosfet1**), y con el nombre que deseemos.

Ahora ya podemos ir a simular (RUN).

Ejemplo 2.

Supongamos que necesitamos trabajar con el IGBT de Signetics Thomson: IGBT-STGF10NB60SD Partimos de que no tenemos ni símbolo ni librería.

En la Web del fabricante encontramos la librería Spice para este componente en modo de subcircuito. Nos la descargamos y le damos el nombre, por ejemplo, de: STGF10NB60SD.lib

Antes de salvarlo, le hacemos un pequeño arreglo (recuérdese que tan sólo es un archivo de texto), consistente en añadir una segunda línea de comentario, al principio del archivo, con la asignación de nodos a patillas. Observando las tres líneas siguientes, deducimos que hay una relación del nodo 2 con G, del nodo 3 con E, y del nodo 1 con C (obsérvese lo resaltado en negrita).

```
LG 10 2 7.5E-09
LE 3 15 7.5E-09
LC 13 1 4.5E-09
```

Sin embargo, en estas librerías de subcircuitos, es usual que se nos de explícitamente la asignación de patillas a nodos, aunque éste no ha sido el caso.

En definitiva, nos quedará el siguiente archivo de librería:

```
* MODELLING FOR STGF10NB60SD
*           C G E
.SUBCKT STGF10NB60SD 1 2 3
LG 10 2 7.5E-09
LE 3 15 7.5E-09
LC 13 1 4.5E-09
RG 10 5 31
RC 14 13 0.158E-02
RE 15 12 0.399E-02
RF 7 10 1MEG
RX 4 14 925.724
CGE 12 5 0.649E-09
CCG 17 4 0.105E-08
CK 16 4 0.321E-11
DCG 12 11 DG
Q1 12 11 14 Q
M1 11 5 12 12 MOS
E1 16 5 8 7 1
E2 17 5 6 7 1
G1 7 9 11 10 1u
D1 8 9 DI
D2 9 6 DI
D3 16 4 DO
R1 7 8 1MEG
R2 7 6 1MEG
.ENDS STGF10NB60SD

.MODEL MOS NMOS
+ LEVEL = 3
+ VTO = 3.418
+ KP = 2.512
+ THETA = 0.986E-01

.MODEL Q PNP
+ IS = 0.245E-13
+ VAF = 894.872
+ BF = 0.864
+ NE = 1.974
+ ISE = 0.141E-15
+ IKF = 8.034
+ BR = 0.389E-02
+ NC = 2.714
+ NK = 0.887
+ TF = 1E-06
+ ITF = 1
+ VTF = 10
+ XTF = 0.1
.MODEL DI D
+ IS = 0.01E-12
+ RS = 0

.MODEL DO D
+ IS = 0.01E-12
+ CJO = 50E-12
+ VJ = 0.75
+ M = 0.35

.MODEL DG D
+ IS = 0.01E-12
```

+ BV = 740
+ VJ = 0.75
+ M = 0.35

* END OF MODELLING

El siguiente paso será crear un símbolo para este subcircuito. Dibujaremos el símbolo de un IGBT.

Para tener una referencia del tamaño, podemos abrir el editor de símbolos con otro que ya tengamos, por ejemplo, de un transistor NPN. Seguidamente, podríamos borrar y dibujar las líneas necesarias sobre el símbolo del NPN para convertirlo en el IGBT ; teniendo cuidado, al guardar, de ir a Guardar Como... , no vayamos a pisar el símbolo que hemos sacado como referencia de tamaño.

No obstante, lo que vamos a hacer será crear uno nuevo desde el principio. Sí será conveniente abrir un símbolo, como el del NPN, para tomar unas notas sobre su tamaño y seguidamente cerrarlo.

Vamos al editor de símbolos: Abrimos el LTspice y elegimos File / New Symbol

En primer lugar dibujamos el símbolo, con Draw / Line

En segundo lugar vamos a Add Pin Port y vamos situando los tres pines de forma que se vaya cumpliendo la correspondencia entre los valores: 1, 2 y 3 de Netlist Order con los respectivos C, G y E de Label.

En tercer lugar vamos a Edit / Attributes / Edit Attribute y en el campo de Type elegimos Cell, en el campo de Prefix escribimos X y en Value escribimos STGF10NB60SD

En cuarto vamos a Edit / Attributes / Attribute Window y en la ventana que nos sale pinchamos en Inst Name y OK. Con esto vemos que nos sale un texto (Unnn) que situamos, por ejemplo, en el lado derecho del símbolo. Este texto se corresponderá con la referencia del componente.

Repetimos el proceso eligiendo Value, con lo que nos sale un texto (STGF10NB60SD) que podemos situar debajo del anterior.

Finalmente, guardamos, por ejemplo, con el nombre de: X-IGBT.asy

Ejemplo3.

El siguiente proceso corresponde al amplificador operacional LM741

En primer lugar, se descargó un modelo de este amplificador operacional como subcircuito, al que denominamos como: **LM741.lib**

```
*/
*/
*LM741 OPERATIONAL AMPLIFIER MACRO-MODEL
*/
*
* connections:  non-inverting input
*               | inverting input
*               || positive power supply
*               ||| negative power supply
*               ||| | output
*               ||| | |
*               ||| | | |
.SUBCKT LM741 1 2 99 50 28
*
*Features:
*Improved performance over industry standards
*Plug-in replacement for LM709,LM201,MC1439,748
*Input and output overload protection
*
*****INPUT STAGE*****
*
IOS 2 1 20N
*^Input offset current
R1 1 3 250K
R2 3 2 250K
I1 4 50 100U
R3 5 99 517
R4 6 99 517
Q1 5 2 4 QX
Q2 6 7 4 QX
*Fp2=2.55 MHz
C4 5 6 60.3614P
*
*****COMMON MODE EFFECT*****
*
I2 99 50 1.6MA
*^Quiescent supply current
EOS 7 1 POLY(1) 16 49 1E-3 1
*Input offset voltage.^
R8 99 49 40K
R9 49 50 40K
*
*****OUTPUT VOLTAGE LIMITING*****
V2 99 8 1.63
D1 9 8 DX
D2 10 9 DX
V3 10 50 1.63
*
*****SECOND STAGE*****
*
EH 99 98 99 49 1
G1 98 9 5 6 2.1E-3
*Fp1=5 Hz
R5 98 9 95.493MEG
C3 98 9 333.33P
*
*****POLE STAGE*****
*
*Fp=30 MHz
G3 98 15 9 49 1E-6
R12 98 15 1MEG
C5 98 15 5.3052E-15
*
*****COMMON-MODE ZERO STAGE*****
*
*Fpcm=300 Hz
G4 98 16 3 49 3.1623E-8
L2 98 17 530.5M
R13 17 16 1K
*
*****OUTPUT STAGE*****
*
F6 50 99 POLY(1) V6 450U 1
E1 99 23 99 15 1
R16 24 23 25
D5 26 24 DX
```


Podemos crear un subdirectorio de trabajo al que llamaremos, por ejemplo, **Operacional**.
Supongamos que su ruta es la siguiente: D:\Trabajos\Simulación\Operacional

Copiamos el modelo **LM741.lib** , quedándonos D:\Trabajos\Simulación\Operacional\LM741.lib

Entramos en el LTspice y procedemos a dibujar un circuito.

Agregamos el comando necesario para que el programa pueda localizar la librería o modelo:

.LIB D:\Trabajos\Simulación\Operacional\LM741.lib

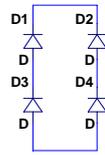
Podemos simplificar este comando, siempre que tengamos la librería cargada en el mismo directorio de trabajo, como es en el caso que estamos describiendo. En definitiva, bastará con utilizar el comando:

.LIB LM741.lib

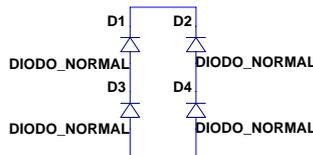
Ejemplo 4.

Vamos a crear un subcircuito a partir de un circuito determinado.

Por ejemplo, crearemos un subcircuito correspondiente a un puente rectificador genérico. Para ello, entramos en el LTspice y dibujamos un puente rectificador con diodos.



Seguidamente, puesto que vamos a dejar los diodos por omisión (genéricos) y así los definiremos en el modelo, hacemos clic con el derecho sobre las D y escribimos, por ejemplo, DIODO_NORMAL. En definitiva, nos queda el circuito siguiente:



Opcionalmente podemos salvar el circuito, por ejemplo: puente.asc , aunque no es necesario.

Vamos a View / SPICE Netlist , con lo que se nos abre una ventana de la que podemos copiar en el portapapeles y utilizar para crear un archivo de texto (por ejemplo, con el Bloc de Notas). La Netlist es la siguiente:

```
* D:\PROFESIONAL\MATERIAS\E-Analógica\Última Actualización\Prácticas-
LTspice\P25\puente.asc
D1 N002 N001 DIODO_NORMAL
D2 N003 N001 DIODO_NORMAL
D3 N004 N002 DIODO_NORMAL
D4 N004 N003 DIODO_NORMAL
.model D D
.lib C:\ARCHIV~1\LTC\LTSPIC~1\lib\cmp\standard.dio
.backanno
.end
```

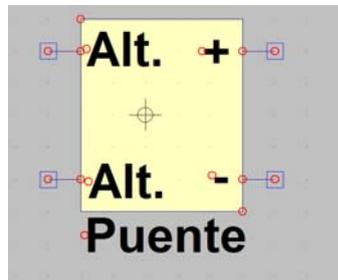
De este archivo nos quedamos sólo con las líneas que especifican los nodos de los componentes. Modificamos la línea donde se especifica el modelo de los diodos: Obsérvese que hemos sustituido la D por DIODO_NORMAL y hemos añadido () a continuación de la segunda D con ánimo de especificar que los parámetros son por omisión (no hemos escrito nada dentro del paréntesis).

Ya sólo falta añadir tres líneas más: Una primera línea de comentario, en la que aprovechamos para dejar indicado el orden de los terminales (ALT ALT + -) que se corresponderán con números de orden de los pines o puertos cuando dibujemos el símbolo para este subcircuito. Una segunda línea en la que especificamos el comando .SUBCKT, con el nombre del subcircuito (PUENTE) y los nodos correspondientes (y en el mismo orden) a los dos terminales de alterna y a los de salida. Finalmente, escribimos una última línea correspondiente al comando .END

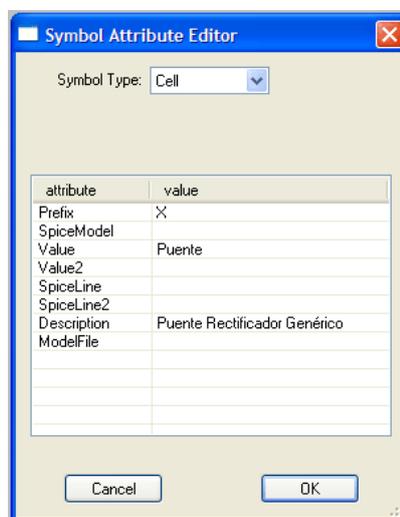
En definitiva, nos queda el siguiente archivo que corresponde al modelo del puente. Lo salvamos, por ejemplo, como Puente.sub

```
*          ALT  ALT  +  -
.SUBCKT PUENTE N002 N003 N001 N004
D1 N002 N001 DIODO_NORMAL
D2 N004 N002 DIODO_NORMAL
D3 N003 N001 DIODO_NORMAL
D4 N004 N003 DIODO_NORMAL
.MODEL DIODO_NORMAL D()
.ENDS PUENTE
```

El siguiente paso es dibujar el símbolo. El cuerpo del símbolo lo dibujamos con Draw / Rect
Con Draw / Line dibujamos los terminales, y situamos en sus extremos los nodos con Edit / Add Pin/Port
Vamos situando los cuatro pines de forma que se vaya cumpliendo la correspondencia entre los valores:
1, 2, 3 y 4 de Netlist Order con los respectivos "ALT.", "ALT.", "+" y "-" de Label.



Seguidamente vamos a Edit / Atributos / Edit Attribute y cumplimentamos la ventana de la siguiente forma:



Finalmente, salvamos el símbolo, por ejemplo, como Puente.asy

LA AYUDA DEL LPSPICE.

Para interpretar el formato de los modelos de componentes: Ayuda / Contenido / LTspice / Circuits Elements / elegimos un dispositivo o componente.

ELEMENTOS DE CIRCUITO Y COMANDOS.

Elementos de Circuitos y Fuentes	1ª Letra	Nombre del Tipo de Modelo
Bipolar junction transistor	Q	NPN/PNP
Capacitor	C	CAP
Current-controlled current source	F	
Current-controlled switch	W	ISWITCH
Current-controlled voltage source	H	
Diode	D	D
Exponential source	EXP	
GaAs MES field-effect transistor	B	GASFET
Independent current source	I	
Independent voltage source	V	
Inductor	L	IND/CORE
Junction field-effect transistor	J	NJF/PJF
MOS field-effect transistor	M	NMOS/PMOS
Mutual inductors (transformer)	K	
Piecewise linear source		PWL
Polynomial source		POLY(n)
Pulse source		PULSE
Resistor	R	RES
Single-frequency frequency modulation source		SFFM
Sinusoidal source		SIN
Transmission line	T	
Voltage-controlled current source	G	
Voltage-controlled switch	S	VSWITCH
Voltage-controlled voltage source	E	

Análisis y Funciones	Comandos
AC or frequency analysis	.AC
DC operating analysis	.OP
DC sweep	.DC
End of subcircuit	.ENDS
Fourier analysis	.FOUR
Frequency response transfer function	.FREQ
Function definition	.FUNC
Global nodes	.GLOBAL
Graphical postprocessor	.PROBE
Include file	.INC
Initial conditions	.IC
Library file	.LIB
Model definition	.MODEL
Node setting	.NODESET
Noise analysis	.NOISE
Options	.OPTIONS
Parameter definition	PARAM
Parameter variation	.PARAM
Parametric analysis	.STEP
Plot output	.PLOT
Print output	.PRINT
Sensitivity analysis	.SENS
Subcircuit definition	.SUBCKT
Table	TABLE
Temperature	.TEMP
Transfer function	.TF
Transient analysis	.TRAN
Value	VALUE
Width	.WIDTH

SITIOS WEB CON MODELOS SPICE.

Analog Devices

http://products.analog.com/products_html/list_gen_spice.html

Apex Microtechnology

<http://eportal.apexmicrotech.com/mainsite/index.asp>

Coilcraft

<http://www.coilcraft.com/models.cfm>

Comlinear

<http://www.national.com/models>

Elantec

<http://www.elantec.com/pages/products.html>

Epcos Electronic Parts and Components

http://www.epcos.de/web/home/html/home_d.html

Fairchild Semiconductor Models and Simulation Tools

<http://www.fairchildsemi.com/models/>

Infineon Technologies AG

<http://www.infineon.com/>

Intersil Simulation Models

<http://www.intersil.com/design/simulationModels.asp>

International Rectifier

<http://www.irf.com/product-info/models/>

Johanson Technology

<http://www.johansontechnology.com/>

Linear Technology

<http://www.linear-tech.com/software/>

Maxim

<http://www.maxim-ic.com/>

Microchip

<http://www.microchip.com/index.asp>

Motorola Semiconductor Products

<http://www1.motorola.com/>

National Semiconductor

<http://www.national.com/models>

Philips Semiconductors

<http://www.semiconductors.philips.com/>

Polyfet

<http://www.polyfet.com/>

Teccor

<http://www.teccor.com/asp/sitemap.asp?group=downloads>

Texas Instruments

<http://www.ti.com/sc/docs/msp/tools/macromod.htm#comps>

Zetex

<http://www.zetex.com/>

Analog & RF Models

<http://www.home.earthlink.net/~wksands/>

Analog Innovations

<http://www.analog-innovations.com/>

Duncan's Amp Pages

<http://www.duncanamps.com/>

EDN Magazine

<http://www.e-insite.net/ednmag/>

Intusoft Free SPICE Models

<http://www.intusoft.com/models.htm>

MOSIS IC Design Models

<http://www.mosis.org/>

Planet EE

<http://www.planetee.com/>

PSpice.com

<http://www.pspice.com/>

SPICE Models from Symmetry

<http://www.symmetry.com/>

SPICE Model Index

<http://homepages.which.net/~paul.hills/Circuits/Spice/ModelIndex.html>

OTROS SITIOS INTERESANTES CON MODELOS SPICE.

<http://www.diodes.com/products/spicemodels/index.php>

<http://www.onsemi.com/PowerSolutions/supportDoc.do?type=models&category=823>

<http://www.emwonder.com/spicemodels/>