

# **Introducción a PSPICE**

## **Historia**

Durante la década de los 60's, IBM desarrolló el software ECAP que luego fue ocupado como referencia por la Universidad de Berkeley para desarrollar el software CANCER. Este fue la antesala de SPICE (*Simulation Program with Integrated Circuits Emphasis*) también desarrollado por la Universidad de Berkeley a principios de los 70's. SPICE era un simulador que, entre otras cosas, permitía analizar grandes ordenadores, estaciones de trabajo y circuitos analógicos sin necesidad de implementarlos en forma física. El éxito logrado por SPICE junto a su uso generalizado, permitió la creación de nuevas versiones del software.

Basado en SPICE nace en 1984 PSPICE, como un simulador de circuitos electrónicos adaptado a computadores personales. Existen versiones de PSPICE tanto para DOS como para Windows. Este último tiene la ventaja de poseer un editor esquemático (Schematics), permitiendo una entrada gráfica para el circuito.

La empresa a cargo de estas implementaciones es MicroSim Corporation que desde 1998 se unió a OrCad. Una versión gratuita para estudiantes de PSPICE versión 9.1 se puede descargar desde el sitio Web:

<http://www.cadencepcb.com/products/downloads/PSpicestudent/default.asp>

## **PSPICE 9.1**

Para diseñar y simular un circuito es necesario tener en cuenta las siguientes etapas:

1. Definir sus partes y de qué manera están conectadas (esquema).
2. Determinar los parámetros de sus componentes y el tipo de análisis a realizar.
3. Establecer los puntos de prueba desde donde se van a efectuar las medidas y/o comprobar las formas de onda.

Dentro de los diferentes análisis que ofrece PSPICE se encuentran:

1. análisis DC (corriente continua),
2. punto de operación,
3. función de transferencia para señal pequeña,
4. respuesta en el tiempo,
5. respuesta en frecuencia,
6. análisis de ruido, y
7. componentes de Fourier.

La mayor funcionalidad que tiene PSPICE es que permite de manera grafica poder representar circuitos eléctricos de distintos grados de complejidad. Esto se debe al subprograma Schematics. Debido que en versiones anteriores se debía realizar dichas simulaciones mediante "código", es decir especificando cada elemento del circuito mediante comandos (lo cual aun se puede realizar en la v9.1).

## Un primer ejemplo

Se desea analizar un circuito RLC paralelo (ver Figura 1). La idea es observar la forma de onda del voltaje dadas ciertas condiciones iniciales en el condensador e inductancia. Se necesita una resistencia, un condensador, una bobina y cable. En el menú *Draw* se activa la función *Get New Part*, mediante la cual se tiene acceso a los diferentes elementos que poseen las bibliotecas. Tanto la Resistencia como el condensador y la bobina se encuentran en la biblioteca *Analog* (R, L y C). Otro elemento muy importante es la Tierra con respecto a la cual se medirá la caída de tensión. Ella se consigue en la biblioteca *Port* bajo el nombre *GND EARTH*.

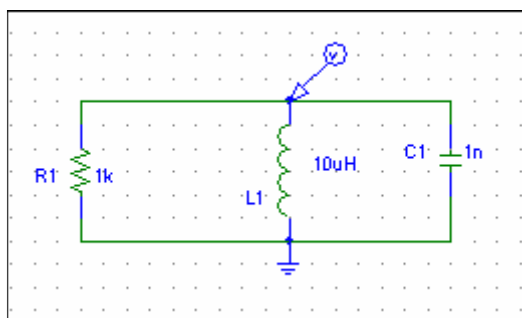


Figura 1: Circuito RLC paralelo en PSPICE.

Los valores de los elementos pueden modificarse haciendo doble clic directamente sobre el valor de ellos. En el caso del condensador y la bobina, es posible incluir condiciones iniciales, llamadas *IC* por Initial Condition (Figura 2). Para el primero, se especifica el voltaje inicial y para el segundo, la corriente inicial (por defecto son nulos).

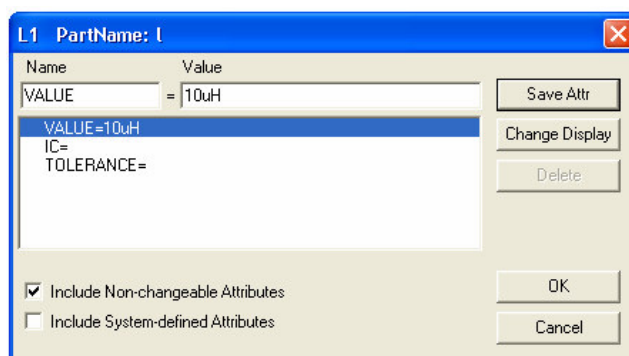
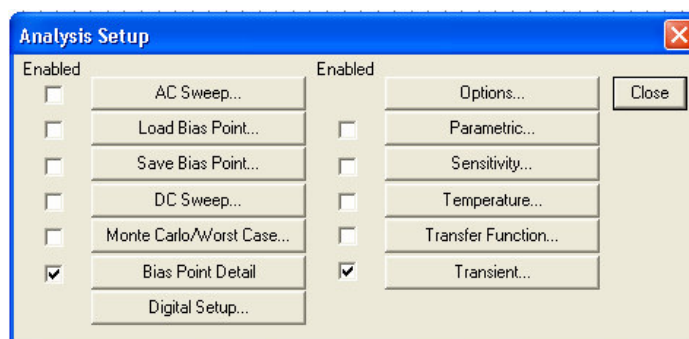


Figura 2: Cuadro de dialogo de la Inductancia L1.

Todas las unidades de las partes están en el sistema MKS (así que no es necesario colocar la palabra Henry, por ejemplo) y pueden utilizarse los sufijos m (mili), u (micro), k (kilo), etc. Otro parámetro que está disponible para los elementos es su tolerancia, la cual se refiere al rango en que varía su valor (ej. 20%) y es útil en el análisis de Montecarlo.

Por otro lado, se puede modificar el nombre de los elementos salvo su primera letra, por ejemplo, para la inductancia, L, y el resto del nombre puede tener 7 caracteres (PSPICE no es sensible a las mayúsculas). En el ejemplo se usó  $R1 = 1[k\Omega]$ ,  $C1 = 1[nF]$ ,  $IC=1[V]$ ,  $L1 = 10[\mu H]$ ,  $IC = 0[A]$ .

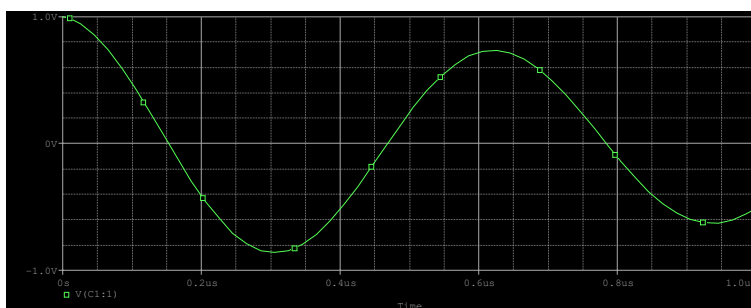
El paso siguiente consiste en unir las partes, para lo cual hay que ingresar al menú *Draw* y activar la función *Wire*. Una vez que hemos definido las componentes del circuito, su conexión y sus valores, especificamos qué tipo de simulación se desea realizar (Figura 3). Para ello, hay que seleccionar la función *Setup* dentro del menú *Analysis*. En este caso, se escoge el análisis transitorio (*Transient*) y se especifica el tiempo final y el paso de simulación (por ejemplo, 10seg y 20nseg respectivamente).



**Figura 3:** Especificación del Tipo de Simulación.

Finalmente, se le indica la medición de la caída de tensión. Una forma gráfica de especificar las variables que se desean observar es mediante el menú *Markers* en el cual se encuentran “puntas de prueba”. En este caso escogemos una punta para medir el voltaje con respecto a tierra, seleccionando *Mark Voltage*.

Para observar los resultados en forma gráfica se debe ir al menú *Analysis* → *Simulate*, con lo cual aparecerá una ventana gráfica con las variables de interés. Dicha ventana posee varias funciones, como observar los valores de la señal en el mismo gráfico y otras que pueden ser exploradas fácilmente. La señal obtenida luce como en la Figura 4. Cabe mencionar, que existe un archivo de salida (en formato de texto), donde pueden apreciarse las especificaciones del circuito mediante “línea de comandos” y los resultados de la simulación. Este archivo puede observarse activando la opción *Analysis* → *Examine Output*.



**Figura 4:** Resultado de la Simulación.

Otro ejemplo interesante resulta de agregar una fuente de corriente sinusoidal al circuito (ISIN). Tal fuente se define mediante los siguientes parámetros:

1.  $I_{off}$  , corriente de offset (componente continua) [A]
2.  $I_{ampl}$ , amplitud de la señal sinusoidal [A]
3. Freq, frecuencia [Hz]
4.  $T_d$ , tiempo de retardo [seg.]
5.  $D_f$  , factor de amortiguamiento [Hz]
6. Fase, desfase de la señal [Grados]

$$I_{SIN} = \begin{cases} I_{off} + I_{ampl} \sin\left(\frac{2\pi}{360} Fase\right) & 0 < t < T_d \\ I_{off} + I_{ampl} \sin\left(2\pi\left\{Freq(t - T_d) + \frac{Fase}{360}\right\}\right) \exp\{-(t - T_d)D_f\} & T_d < t \end{cases}$$

## Variación de elementos pasivos

Suponiendo que se desea analizar el circuito anterior para distintos valores de la resistencia  $R1$ . Basta con reemplazar el valor de la resistencia por “{Rvar}” o cualquier otro nombre entre llaves. Se hace doble click al elemento *PARAM*, que se obtiene desde el cuadro de diálogo *Get New Part*, obteniendo un esquemático como el de la Figura 5.

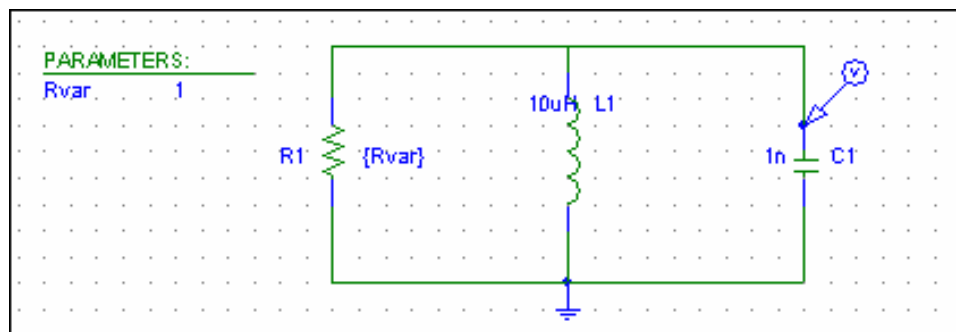


Figura 5: Ejemplo de Variación de la Resistencia.

Luego, ir al menú principal y activar *Analisis* → *Setup*, presionando el botón *Parametrics*. En *Name* se escribe el nombre dado a la resistencia, *Rvar*. En *Sweep Var, Type* se activa la opción “Global Parameter”. En *Sweep Type* se elige el modo de variación deseado. En el ejemplo se ve que la resistencia toma los valores 0.1, 1 y 10[kΩ] (ver Figura 6). La Figura 7 muestra los resultados gráficos del voltaje en el condensador para los distintos casos.

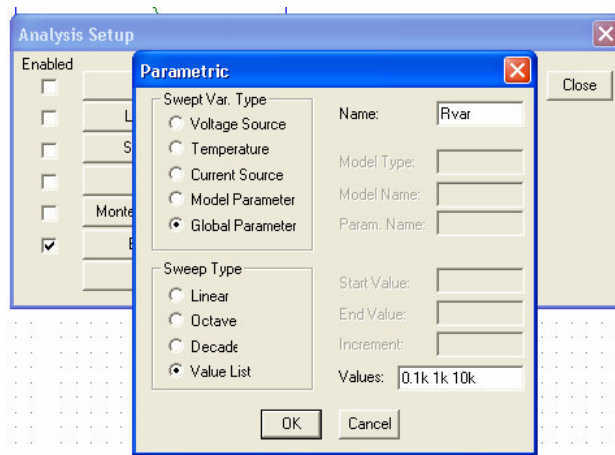


Figura 6: Selección de valores para R1.

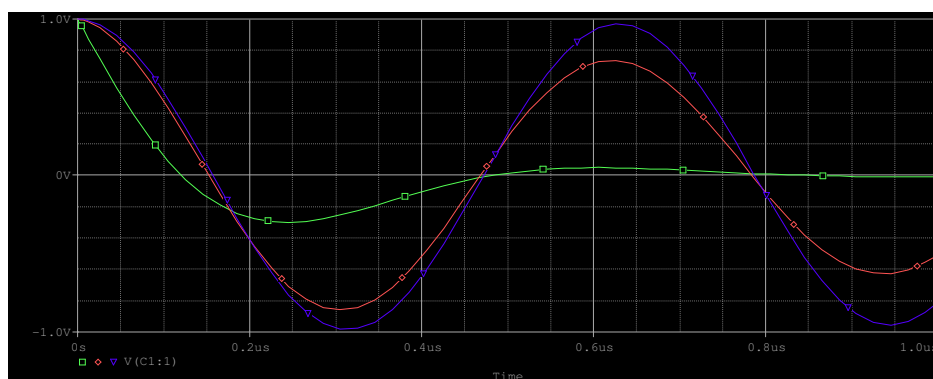


Figura 7: Voltaje en el condensador para distintos valores de R1.

## Análisis en Régimen Permanente Sinusoidal (RPS)

Para realizar un análisis en RPS es necesario que las fuentes independientes estén representadas en forma de fasores. Para ello se usan los elementos VAC e IAC encontrados en *Get New Part*. Al agregar una fuente de corriente al circuito del ejemplo, el circuito queda como el de la Figura 8.

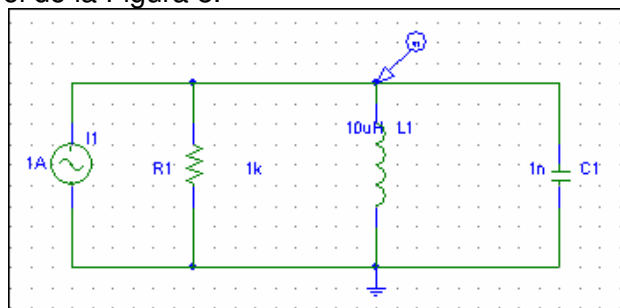
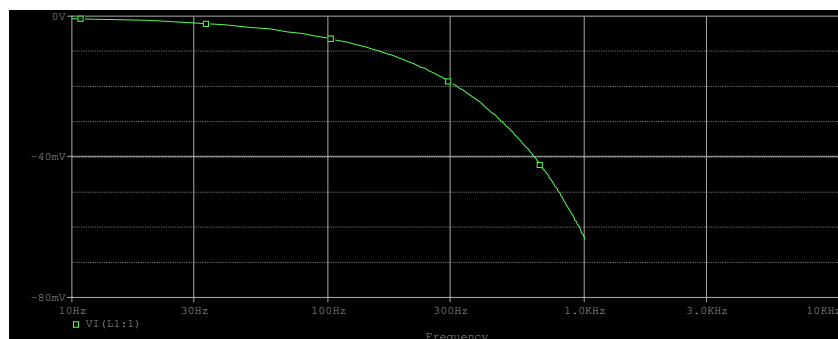


Figura 8: Circuito para Análisis en Frecuencia.

A continuación deben establecerse los parámetros del análisis. Para esto se debe activar la opción **Analysis** → **Setup**. Se debe marcar la opción **AC Sweep**. Se puede analizar el comportamiento del circuito en un intervalo de frecuencias establecido (**Sweep Parameters**) variando con el modo deseado (**AC Sweep Type**).

Los valores de frecuencias deben estar en Hertz. De un fasor se puede obtener distintas representaciones. Los marcadores del menú **Markers** → **Mark Advanced** permiten elegir la opción que se necesite, destacando **vphase**, **iphase**, **vreal**, **ireal**, **vimaginary** e **iimaginary** para las fases y partes real e imaginaria de voltajes y corrientes, respectivamente.



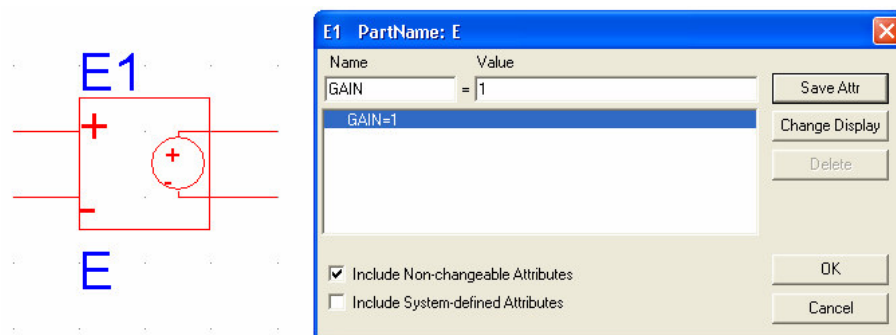
**Figura 9:** Parte imaginaria del voltaje en el condensador en función de la frecuencia.

## Fuentes dependientes

Se obtienen cargando la biblioteca *Analog*, las cuales son:

1. Fuente de voltaje controlada por voltaje (E),
2. Fuente de corriente controlada por corriente (F),
3. Fuente de corriente controlada por voltaje (G), y
4. Fuente de voltaje controlada por corriente (H).

Cada una de estas fuentes hay que conectarla e indicar su ganancia.



**Figura 10:** Fuente de voltaje controlada por voltaje de ganancia unitaria.