

Introducción a MicroSim PSpice® : DESCRIPCIÓN DEL ENTORNO

1. Introducción

El origen del simulador SPICE puede remontarse a hace más de treinta años. A mediados de la década de los 60 IBM desarrolló el programa ECAP, que más tarde serviría como punto de partida para que la Universidad de Berkeley desarrollara el programa CANCER. Hasta principios los años 70 los circuitos electrónicos se analizaban casi exclusivamente de forma manual, mientras que de forma progresiva, la complejidad de éstos iba en aumento.

En ese momento un grupo de la Universidad de Berkeley, tomando como base el programa CANCER, desarrolló la primera versión de **SPICE** (*'Simulation Program with Integrated Circuits Emphasis'*): un simulador de propósito general que permitía analizar sobre grandes ordenadores y estaciones de trabajo, circuitos analógicos sin necesidad de montarlos físicamente.

En SPICE el circuito y análisis requerido es descrito en un fichero de texto mediante un serie de comandos y declaraciones. A este fichero se le llama fichero del circuito. La sintaxis de las declaraciones permite dejar parámetros sin especificar con lo que tomarán valores por defecto. El fichero del circuito es leído por SPICE, éste comprueba que no hay errores en las conexiones y sintaxis declaradas, y finalmente hace la simulación.

El éxito alcanzado por el programa, así como su utilización generalizada, ha originado la aparición de numerosas versiones de SPICE.



En 1984 nace de manos de MicroSim Corporation la primera adaptación para ordenadores personales: **PSpice**. Desde entonces, PSpice ha ido renovándose continuamente. Versiones ampliamente extendidas de PSpice son la 8.0 (última desarrollada por MicroSim) y la 9.1 (primera tras la fusión de OrCAD y MicroSim), hasta llegar a la 16.0, la última actualmente en el mercado.

SPICE está considerado como el estándar en análisis electrónico. Es referencia y base de numerosos simuladores del mercado.

2. Descripción del Entorno MicroSim PSpice 8.0 (evaluación)

2.1. Design Manager

El conjunto de aplicaciones que componen el simulador está agrupado en un **centro de diseño**: *Design Manager* (ver Fig.1). De esta forma se proporciona un entorno integrado para simulación y análisis de circuitos analógicos, digitales y mixtos.

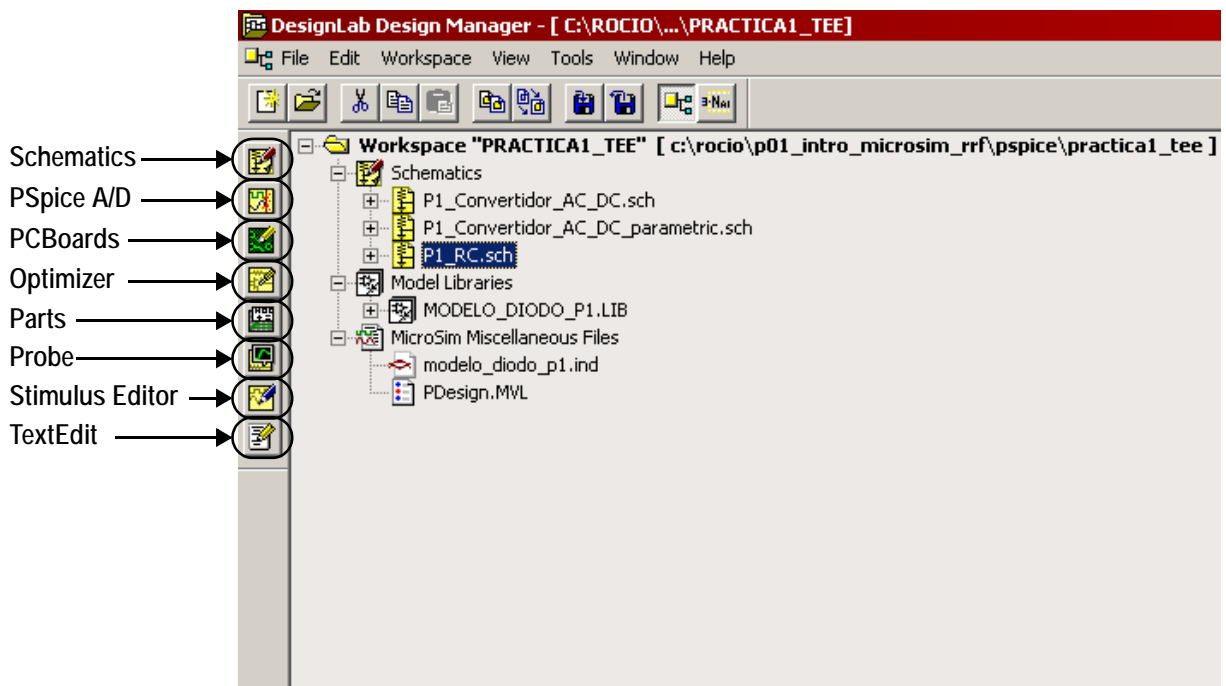


Fig. 1 Ventana de *Design Manager*.

En cierto modo, es realmente un conjunto de programas especializados, donde todos trabajan juntos para crear un entorno de desarrollo EDA (*'Electronics Design Automation'*) potente. Los programas que contiene son:

- ♦ Schematics
- ♦ PSpice A/D
- ♦ PCBoards ^{†1}
- ♦ PSpice Optimizer
- ♦ Parts
- ♦ Probe
- ♦ Stimulus Editor
- ♦ TextEdit

Con *Design Manager* resulta fácil realizar operaciones de gestión de los ficheros utilizados por las distintas aplicaciones, ya que detecta las relaciones de dependencia entre ellos. Así no corremos el riesgo de perder un fichero necesario para un determinado diseño, al cambiarlo de carpeta. Los ficheros están agrupados creando diferentes entornos de trabajo.

1. Esta aplicación permite la edición de placas circuitos impresos o PCBs (*'Printed Circuit Board'*). Su estudio está fuera de los propósitos de la asignatura, por lo que no se describirá a continuación.

2.2. Schematics

Esta aplicación permite la **captura de esquemáticos** (CE). Con la CE, se introduce y edita el esquemático del circuito de forma sencilla usando la técnica de arrastrar y soltar con el ratón. Así la posibilidad de conexiones erróneas o nudos flotantes es menor que si introdujéramos el esquemático en modo texto (en entornos sin CE).

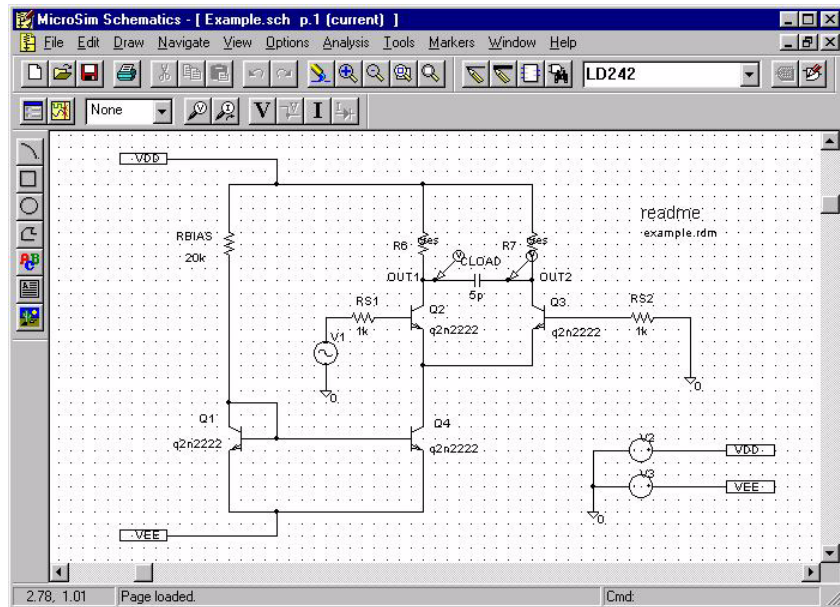


Fig. 2 Ventana de Schematics.

En *Schematics*, el esquemático del circuito y los análisis asociados se guardan en único fichero con extensión **.sch**. Este fichero se utilizará posteriormente (al ejecutar el simulador) para generar automáticamente el fichero del circuito en modo texto, de extensión **.cir**, que resulta legible para el simulador.

Cada uno de los dispositivos que podemos seleccionar, arrastrar y soltar en *Schematics* no son más que **símbolos gráficos**, que nos son útiles para representar un esquema eléctrico. Estos símbolos tienen asociados una serie de **atributos**, que son los que realmente definen sus características. *Schematics* dispone de una herramienta capaz de editar símbolos, de forma que podemos añadir, quitar o modificar los atributos de un símbolo o definir nuestra propia librería de símbolos (extensión **.sbl**).

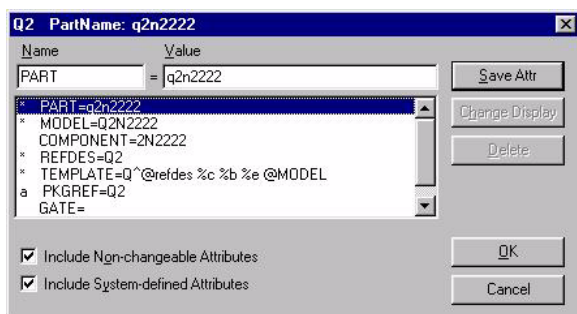


Fig. 3 Ventana de atributos de un símbolo.

Algunos atributos son modificables desde *Schematics* haciendo doble “click” sobre el símbolo, por ejemplo el valor óhmico de un resistor. En la Fig.3 podemos ver algunos atributos asociados a un transistor bipolar (BJT). Los que están señalados con asterisco sólo podrán ser modificados desde el editor de símbolos, el resto pueden modificarse desde *Schematics*.

Uno de los atributos más importantes es **model**, ya que determina el modelo eléctrico asociado al símbolo. Puede ser tanto un modelo específico de PSpice (diodo,

transistor bipolar, MOSFET, etc.), como un subcircuito. El valor de *model* debe coincidir exactamente con el nombre de la definición **.model** o **.subckt** que está en una librería de modelos. Las librerías de modelos son ficheros con extensión **.lib**.

Otro atributo de importancia es *template*, ya que define la correspondencia entre los pines del símbolo y los del modelo eléctrico; un error en esta correspondencia dará resultados erróneos. En *Schematics User's Guide*, que se proporciona en formato PDF con el simulador, puede encontrarse una lista completa de atributos.

Tipos de análisis y configuración

Schematics permite seleccionar el tipo de análisis que deseemos del circuito, así como las opciones para éste desde la ventana **Analysis Setup**.

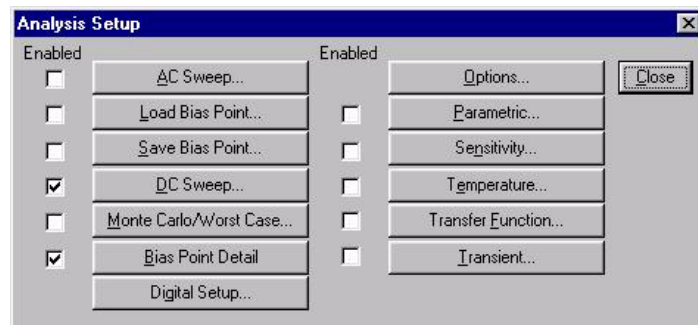


Fig. 4 Ventana de Analysis Setup.

2.2.1 AC Sweep: Barrido en AC y análisis de ruido

El barrido en AC es un análisis en frecuencia que calcula la respuesta en pequeña señal del circuito, linealizado entorno al punto de polarización, en el rango de frecuencias especificado.

El análisis de ruido (**Noise Analysis** dentro del menú de *AC Sweep*) calcula las contribuciones de ruido de cada dispositivo.

2.2.2 Load/Save Bias Point: Cargar y salvar el punto de operación

Permiten restablecer y salvar en un fichero los cálculos del punto de operación para ser utilizados en sucesivas simulaciones.

2.2.3 DC Sweep: Barrido en DC

Hace un análisis en continua realizando un barrido de la variable especificada, que puede ser una fuente de tensión o de corriente, el parámetro de un modelo o la temperatura, en el rango de valores especificado para dicha variable.

El análisis puede ser anidado (**Nested Sweep** dentro del menú de *DC Sweep*), de forma que se barran dos variables o parámetros a la vez.

2.2.4 Monte Carlo/Worst Case: Análisis de Monte Carlo y de peor caso

Realizan análisis de tipo estadístico que actúan sobre las variaciones de los valores de los componentes conforme a su tolerancia.

2.2.5 Bias Point Detail: Punto de operación detallado

Al seleccionarlo se indica al programa que calcule el punto de operación del

circuito y que los resultados en fuentes y dispositivos sean enviados al fichero de salida tipo texto, de extensión **.out**.

2.2.6 Digital Setup: Configuración digital

En este apartado se especifican los estados iniciales de biestables, el tipo de retraso temporal utilizado en los dispositivos digitales (mínimo, típico o máximo), y los modelos de adaptación de variables analógicas a digitales. Es decir, podemos seleccionar el tipo de circuito de interfaz utilizado y así indicar si queremos evitar los estados intermedios *R* (*rise*, en subida), *F* (*fall*, en bajada) y *X* (indeterminado) de los dispositivos digitales.

2.2.7 Options: Opciones

Permiten establecer las opciones, límites y parámetros de control para diferentes análisis. Hay dos tipos de opciones: aquellas que especifican un valor numérico y aquellas que indican la activación o no de una opción.

2.2.8 Parametric: Análisis paramétrico

Será útil cuando nos interese conocer cómo varía la respuesta de un circuito en función de los valores que, en forma de incrementos, va tomando algún parámetro de un dispositivo, el valor de un componente, la temperatura, etc.

2.2.9 Sensitivity: Análisis de sensibilidad en continua

Calcula la sensibilidad en continua de cada una de las variables de salida que definamos respecto a los dispositivos del circuito, así como a sus parámetros. La información resultante de este análisis se recoge en el fichero **.out** de salida.

2.2.10 Temperature: Temperatura

Permite ajustar la temperatura a la que se analiza el circuito. El valor por defecto es de 27°C.

2.2.11 Transfer Function: Función de transferencia en pequeña señal

Calcula y envía al fichero de salida, la ganancia en pequeña señal, la resistencia de entrada y la resistencia de salida. Si el circuito es no-lineal, entonces primero halla su modelo equivalente linealizando el circuito entorno al punto de operación.

2.2.12 Transient: Análisis transitorio

Proporciona la respuesta del circuito en función del tiempo, en el intervalo especificado.

Limitaciones de Schematics

En la versión de evaluación 8.0, *Schematics* presenta las siguientes restricciones:

- El tamaño máximo de la página de diseño es A4.
- El número máximo de símbolos en un esquemático es de 25.
- Sólo pueden utilizarse a la vez nueve librerías de símbolos (**.sbl**).
- El número máximo de símbolos en la librería creada por el usuario es 20.
- Las librerías de ejemplo constan de 22 componentes analógicos y 140 digitales.

2.3. PSpice A/D

Constituye el auténtico **simulador** y se puede decir que el resto de aplicaciones gira entorno a él. Permite la simulación de circuitos analógicos, digitales y mixtos. A partir del esquemático del circuito (**.sch**), genera automáticamente el fichero del circuito tipo texto (**.cir**) y ejecuta los análisis que se hayan indicado. Como resultado, el circuito genera el fichero de salida tipo texto (**.out**) y el fichero de datos (**.dat**), que será utilizado por *Probe* para la visualización gráfica de resultados.

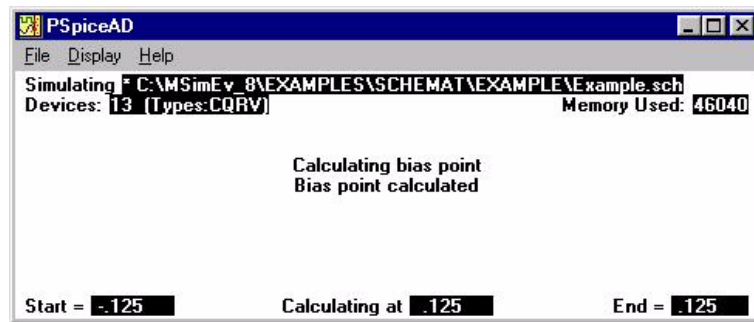


Fig. 5 Ventana de PSpice A/D.

Limitaciones de PSpice A/D

La versión de evaluación 8.0 presenta las siguientes restricciones:

- 64 nudos como máximo
- 10 transistores
- 3 amplificadores operacionales
- 65 puertas básicas o primitivas o una combinación de éstas en dispositivos digitales
- 10 líneas de transmisión ideales con no más de 4 no ideales (líneas con pérdidas, *RLGC*) y 4 líneas acopladas

2.4. PSpice Optimizer

Esta aplicación calcula el valor de unos parámetros del circuito para que éste cumpla una serie de condiciones. Es decir, *PSpice Optimizer*, mediante llamadas a *PSpice A/D*, realiza simulaciones de forma iterativa ajustando los valores de los parámetros hasta que se cumpla la **función objetivo**, definida bajo una serie de **restricciones**. Se puede prescindir de restricciones si no es necesario imponerlas.

Por ejemplo (ver Fig.6), si partimos de un circuito serie constituido por un pila de 5V, una resistencia y un diodo, podemos definir como función objetivo que la corriente sea 1mA, con un rango variación de 5 μ A. El parámetro tomado será el valor de la resistencia y la restricción, que la potencia disipada en el resistor sea mayor o igual que 4mW. Dependiendo de la restricción impuesta se puede o no llegar a un valor del parámetro que satisfaga las condiciones impuestas (puede haber o no solución).

Limitaciones de PSpice Optimizer

En la versión de evaluación 8.0 sólo podemos definir una función objetivo, un parámetro y una restricción.

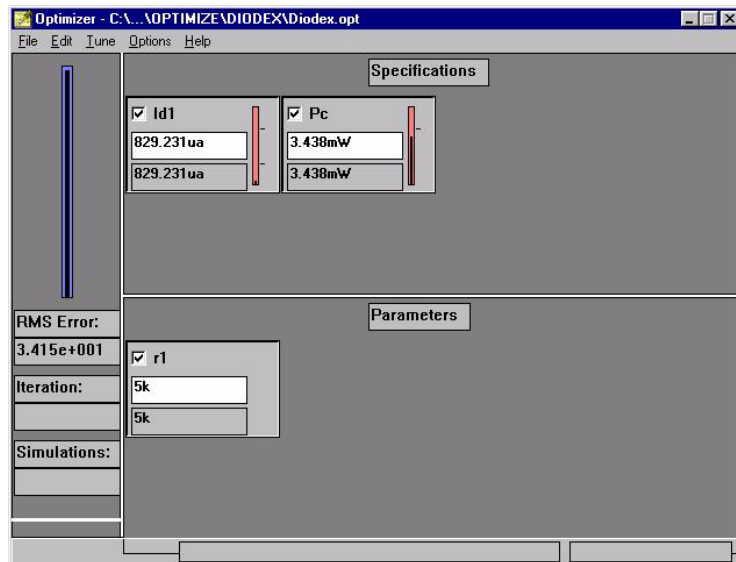


Fig. 6 Ventana de PSpice Optimizer.

2.5. Parts

Con la aplicación *Parts* podemos ampliar nuestra librería de componentes analógicos sin más que introducir la información que los fabricantes proporcionan en las **data sheets** u hojas de datos. Es decir, *Parts* convierte automáticamente estos datos en un modelo básico de PSpice, utilizando la declaración `.model`, o en un subcircuito, usando `.subckt`. *Parts* puede proporcionar gráficas de las características del dispositivo a modelar.

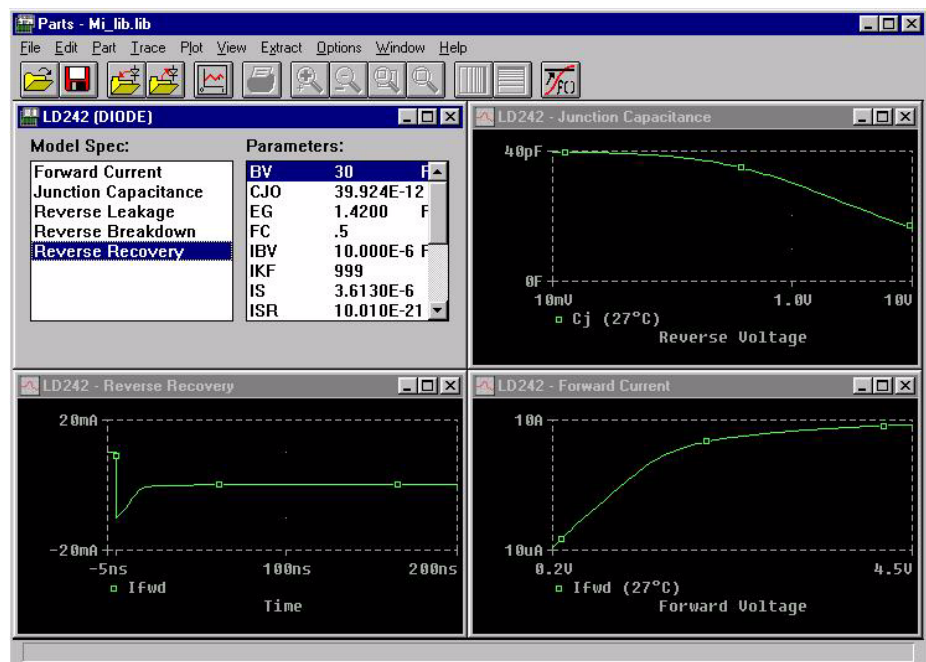


Fig. 7 Ventana de Parts.

Limitaciones de Parts

En la versión de evaluación 8.0 la herramienta de modelado de dispositivos está limitada a la extracción de parámetros de diodos.

2.6. Probe

Es el analizador de formas de onda de PSpice que permite **visualizar** de forma interactiva los resultados de la simulación, empleando **gráficos** de alta resolución. *Probe* representa no sólo simples formas de onda, como tensiones y corrientes, sino también complejas expresiones aritméticas, implicando a dichas señales, e incluso la transformada de Fourier. *Probe* lee los datos de las señales del fichero **.dat** que genera *PSpice A/D*.

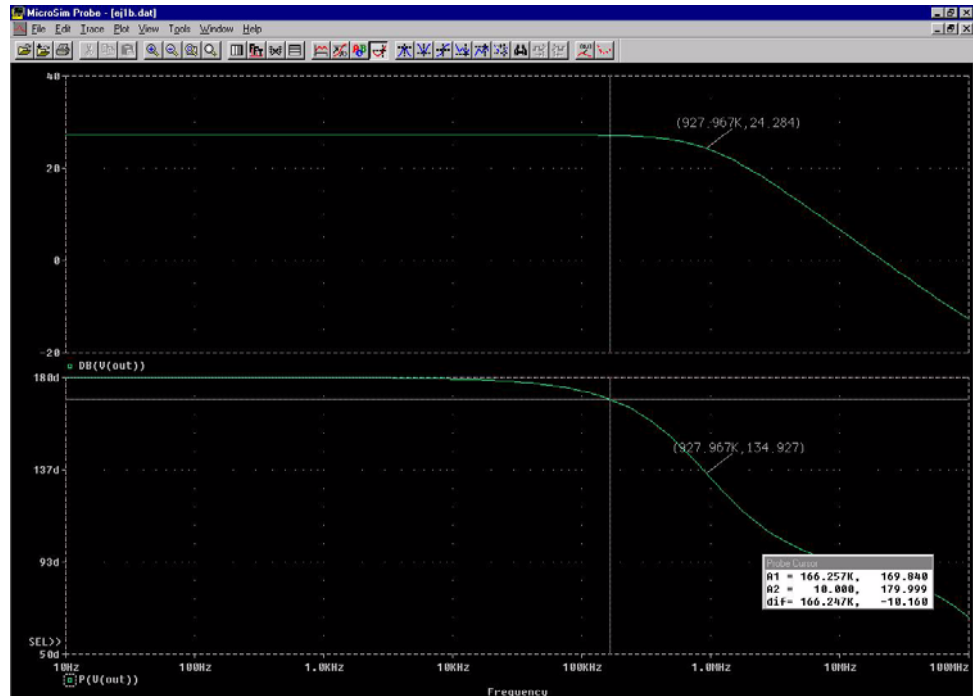


Fig. 8 Ventana de *Probe*.

En simulaciones mixtas *Probe* puede visualizar **formas de onda analógicas y digitales** simultáneamente con una base de tiempos común.

Probe también permite la definición de **macros y funciones objetivo (goal functions)**. Las funciones objetivo calculan un valor simple de una forma de onda de *Probe*, tal como el ancho de un pulso, el tiempo de subida de una señal, el pico máximo, el ancho de banda de un filtro, etc. Las funciones objetivo de *Probe* pueden ser utilizadas por *Optimizer*.

2.7. Stimulus Editor

El editor de estímulos es una herramienta que nos permite de forma fácil y sencilla **crear estímulos o formas de onda de entrada** para nuestros circuitos. Las formas de onda analógica pueden ser sinusoidales, trenes de pulsos, exponenciales, lineales a tramos y señales de FM. Las digitales pueden ser señales de reloj o cualquier secuencia digital.

Limitaciones de *Stimulus Editor*

En la versión de evaluación 8.0 sólo se pueden editar señales analógicas sinusoidales y digitales de reloj.

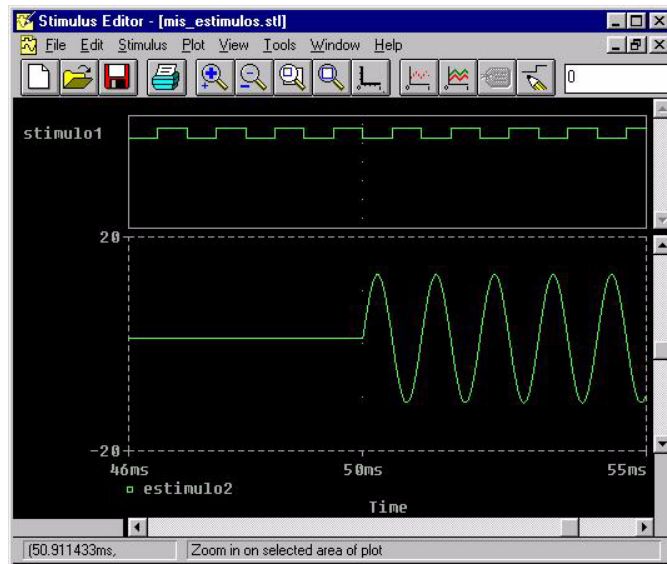


Fig. 9 Ventana de *Stimulus Editor*.

2.8. *TextEdit*

Es un **editor de texto** con el se pueden introducir **modelos** y **subcircuitos**. También permite visualizar los ficheros tipo texto **.cir** y **.out**, de entrada y salida de PSpice AD.

```

**+      Ikf=33.8m Xtb=1.5 Br=4.851m Nc=2 Isc=0 Ikr=0 Rc=1.5 Cj
**+      Mjc=99.99m Vjc=.75 Fc=.5 Cje=3.184p Mje=.3171 Vje=.75
**+      Tf=385.5p Itf=0 Vtf=0 Xtf=0 Rb=10)
*****
*TTL schottky mia
*****
*      74S04_mio inverter Schottky

*$
.subckt 74S04_mio A Y
  optional: DPWR=$G_DPWR DGND=$G_DGND
  +      params: MNTYMXDLY=0 IO_LEVEL=0
U1 inv DPWR DGND
  +      A Y
  +      D_S04 IO_S MNTYMXDLY={MNTYMXDLY} IO_LEVEL={IO_LEVEL}
.ends

*$
.model D_S04 ugate (
  +      tplhty=3ns tplhmx=4.5ns
  +      tphlty=3ns tphlmx=5ns
  \
)

```

Fig. 10 Ventana de *TextEdit*.

Es útil para modelar los dispositivos que *Parts*, por sus limitaciones, no permite y también para editar nuestras propias librerías.

Con suficientes conocimientos es posible crear el fichero del circuito **.cir** sin necesidad de utilizar *Schematics*.