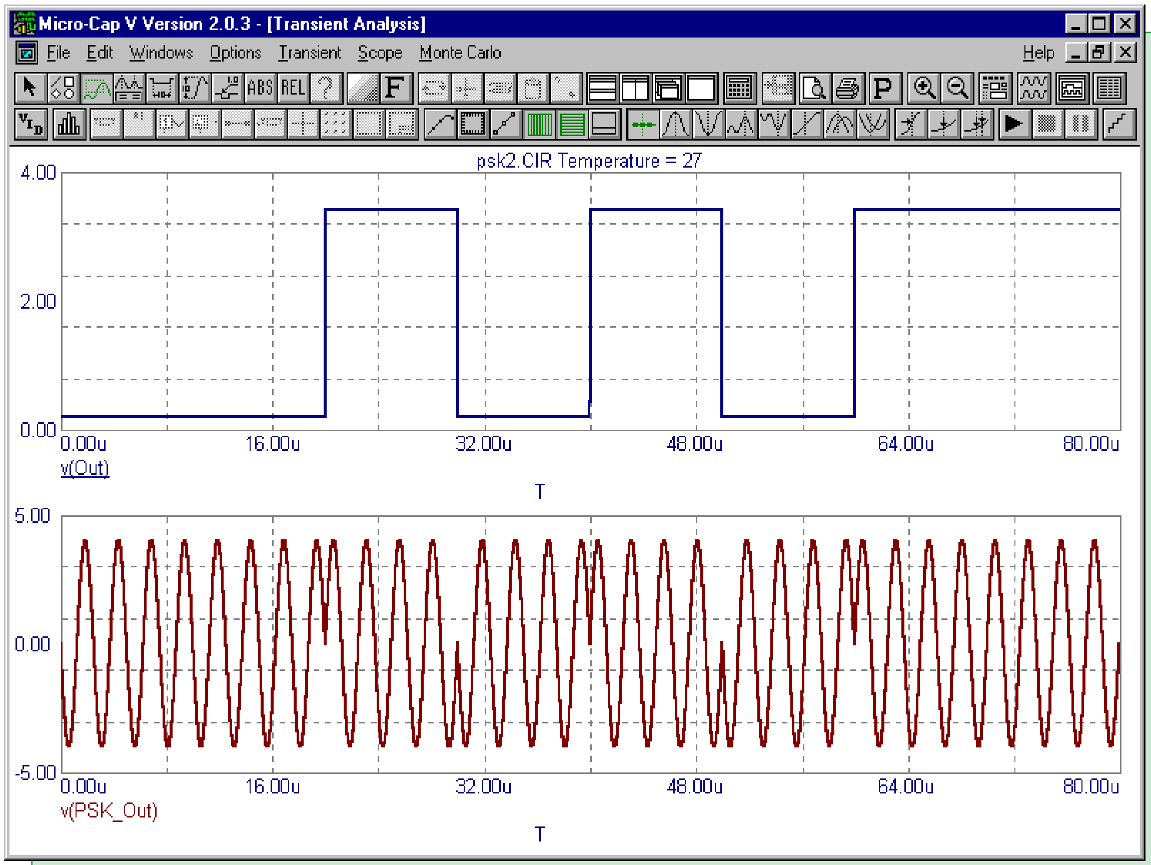
spectrum news

Applications for Micro-Cap™ Users

Spring 1998

PSK Modulation


Micro-Cap V Version 2.0.3 - [Transient Analysis]
File Edit Windows Options Transient Scope Monte Carlo Help
psk2.CIR Temperature = 27
4.00
2.00
0.00
0.00u 16.00u 32.00u 48.00u 64.00u 80.00u
v(Out)
T
5.00
0.00
-5.00
0.00u 16.00u 32.00u 48.00u 64.00u 80.00u
v(PSK_Out)
T

Featuring:
- Using a Vendor Supplied Subcircuit
- Chebyshev Filter Macro
- PSK Modulator Macro
- Storing the Time of an Event

News In Preview

This issue features an article that describes the process of taking a SPICE subcircuit model from a vendor and implementing it in Micro-Cap. This method is useful in accessing the thousands of parts that are available through the Internet. Two macros are displayed in this issue. The first macro is of a Chebyshev filter. The macro will accept basic filter parameters and produce the desired second order filter from them. The second macro is of a PSK modulator. This macro converts a binary input into a PSK modulated waveform which is used in communication circuits. Finally, this issue contains a procedure for storing the time that an event occurs in a simulation. This can be used to define a time based equation that does not start at t=0.

Contents

News In Preview 2
Book Recommendations 3
Micro-Cap V Question and Answer 4
Easily Overlooked Features 5
Using a Vendor Supplied Subcircuit 6
Chebyshev Filter Macro 11
PSK Modulator Macro 15
Storing the Time of an Event 18
Product Sheet 20

Book Recommendations

Micro-Cap / SPICE

- *Computer-Aided Circuit Analysis Using SPICE*, Walter Banzhaf, Prentice Hall 1989. ISBN# 0-13-162579-9
- *Macromodeling with SPICE*, Connelly and Choi, Prentice Hall 1992. ISBN# 0-13-544941-3
- *Semiconductor Device Modeling with SPICE*, Paolo Antognetti and Giuseppe Massobrio McGraw-Hill, Second Edition, 1993. ISBN# 0-07-002107-4
- *Inside SPICE-Overcoming the Obstacles of Circuit Simulation*, Ron Kielkowski, McGraw-Hill, First Edition, 1993. ISBN# 0-07-911525-X
- *The SPICE Book,* Andrei Vladimirescu, John Wiley & Sons, Inc., First Edition, 1994. ISBN# 0-471-60926-9
- *SMPS Simulation with SPICE 3,* Steven M. Sandler, McGraw Hill, First Edition, 1997. ISBN# 0-07-913227-8
- *MOSFET Modeling with SPICE Principles and Practice,* Daniel Foty, Prentice Hall, First Edition, 1997. ISBN# 0-13-227935-5

German

- *Schaltungen erfolgreich simulieren mit Micro-Cap V,* Walter Gunther, Franzis', First Edition, 1997. ISBN# 3-7723-4662-6

Design

- *High Performance Audio Power Amplifiers,* Ben Duncan, Newnes, First Edition, 1996. ISBN# 0-7506-2629-1

Micro-Cap V Question and Answer

Question: I am building a diode bridge, but when I rotate a diode, it will only go through horizontal and vertical positions. I would like to have the diode in a diagonal orientation. Is there a way to do this?

Answer: There is a part specifically set up for this orientation. If you go to the Component/Analog Primitives/Passive Components menu, there is a component called D45. This component uses the same model as the standard Diode model. However, its shape has been defined to rotate through the four diagonal orientations.

Question: I have a sine source in my schematic with an amplitude of 50mV. When I go to AC analysis, the magnitude is 1V. No matter what value the A parameter is set to in the model statement, it always returns 1V. Why?

Answer: The sine source, pulse source, and user source will always have a 1V magnitude in AC analysis. The model statements or text files for these components will only have an effect in transient analysis or during the DC operating point operation. In order to change the magnitude of an AC small signal source, either the V or the I component must be used. These components are available under the Waveform Sources section of the Analog Primitives. To define their magnitude, the AC keyword must be used in the VALUE attribute of the source. The following text would define the VALUE attribute of the V source as a 50mV AC small signal source:

```
AC 50m
```

Question: I have used a .IC statement to set a voltage across an inductor, but when I run transient analysis, the results are different than what I set.

Answer: In the Transient Analysis Limits, the Operating Point must be disabled. A .IC statement will usually keep the voltage fixed through an operating point calculation. However, when it computes an operating point, it short circuits all inductors. Thus an initial condition across an inductor is a mathematical impossibility and may end up producing extremely large currents within the circuit as the voltage iterates to 0. Disabling the operating point will fix this. Similarly, a voltage source also can not have a .IC statement across it.

Question: When I step a parameter, how can I see which waveform goes with which step?

Answer: After a stepping operation, multiple waveforms will be available on the screen. To view which waveform belongs to which step, you need to invoke Cursor mode. This can be done by choosing Cursor, under the Mode selection of the Options menu or by pressing the F8 hotkey. The title of the plot will appear with the stepped parameter value of the waveform that the cursor is currently on. The title, when only stepping one parameter, will look similar to this:

```
PRLC.CIR  Temperature=27  R1.Value=50
```

where the cursor is currently on the waveform produced when R1 was 50 ohms. To shift the cursor to the other steps, press the Up or Down Arrow cursor key.

Easily Overlooked Features

This section is designed to highlight one or two features per issue that may be overlooked because they are not made visually obvious with an icon or a menu item.

Documenting with WMF Files
Micro-Cap has the capability of producing two types of graphic files: bitmaps (.BMP) and metafiles (.WMF). Both can be easily copied to the clipboard and pasted into other programs through the **Copy to Clipboard** section of the **Edit** menu. Metafiles have the additional benefit that they may be copied to a file on disk through the **Copy the Entire Window to WMF File...** option under the **Edit** menu. The .WMF file can contain the schematic or any of the plot windows that are created. These files can be used as documentation in such programs as Microsoft Word, Adobe Pagemaker, and even Micro-Cap itself. Micro-Cap lets you import a .WMF file into a schematic while in Metafile mode. The file may be placed anywhere in the schematic, and the size edited as desired.

In Figure 1, the sample circuit PRLC.CIR has had two metafiles imported into its schematic. In this example, transient analysis has been run twice, plotting the input one time and then the output the other time. After each of these analyses, a .WMF file was created with a unique name. In the schematic, Metafile mode was enabled and the two graphic files were placed by their respective nodes.

To import a .WMF file in MC5, the file does not have to be created from Micro-Cap. For example, if a company logo is available in .WMF format, this graphic can also be imported into a schematic.

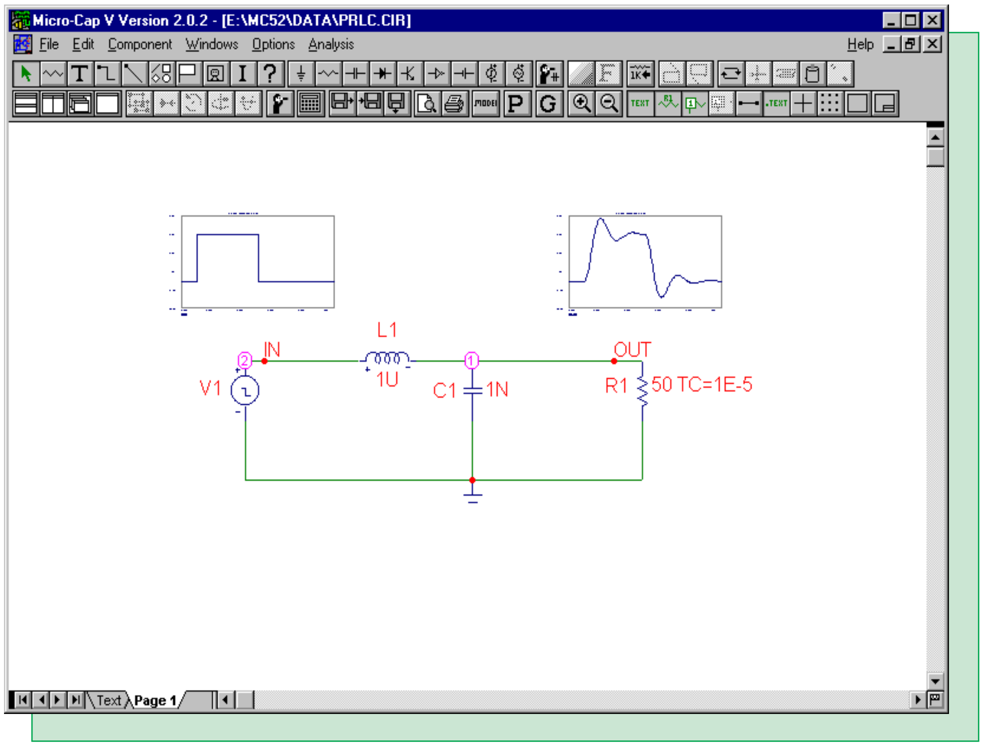

Micro-Cap V Version 2.0.2 - [E:\MC52\DATA\PRLC.CIR]
File Edit Component Windows Options Analysis Help
IN
L1
1U
OUT
V1
C1
1N
R1
50 TC=1E-5
Text Page 1

Fig. 1 - Documenting with WMF Files

Using a Vendor Supplied Subcircuit

Many vendors now produce SPICE models of their components. Thousands of these SPICE models are available for downloading from the Internet from companies such as National, Motorola, TI, Siemens, and many more. A list of some of these companies may be found on our web site at:

http://www.spectrum-soft.com/library.html

While most of these models are included in the latest version of Micro-Cap, there will be a few that will need to be added by the user if they are to be used in a simulation. SPICE models can come in two different formats: .SUBCKT and .MODEL. Almost all vendor supplied models are in the .SUBCKT form. This article describes the process of adding one of these subcircuit models into Micro-Cap's component library.

The Subcircuit Model
For this example, the component to be added will be the AD8011 opamp from Analog Devices. This opamp is a 300MHz, 1mA current feedback amplifier. Analog Device's SPICE models may be found on their web site at:

http://www.analog.com/support/Design_Support.html

The AD8011 model was downloaded from their web site and saved in a file called AD8011.MOD. This file contains just the AD8011 subcircuit model, but any number of models may be saved in a single file. The AD8011.MOD file should be copied into the DATA subdirectory. This file may be opened in any text editor or in Micro-Cap itself. There are four items that need to be noted in the file before it can be added to Micro-Cap. All of these items appear in the block of text below.

```
* Node assignments
*                 non-inverting input
*                 | inverting input
*                 | | positive supply
*                 | | |  negative supply
*                 | | |  |  output
*                 | | |  |  |
.SUBCKT AD8011an 1 2 99 50 28
```

The last line describes three important aspects of the model. The keyword .SUBCKT defines the type of model. In this case, it states that this is a subcircuit model rather than a simple .MODEL. .MODEL statements may appear within a .SUBCKT model, but those statements aren't relevant for importing the model into Micro-Cap. The name of the subcircuit comes next which in this case is 'AD8011an'. Finally, the text string '1 2 99 50 28' defines the pin numbers of the input and output nodes of the model. The first seven lines are comments which are denoted by having the asterisk as the first character of the line. These lines describe the node assignments for the subcircuit pins, which appear on the .SUBCKT line, as follows:

1 - non-inverting input
2 - inverting input
99 - positive supply
50 - negative supply
28 - output

Both the pin number (or pin name) and its assignment need to be known before adding the part to the component library. The file may be closed once the type of model, the name of the model, the pin numbers (or pin names), and the pin assignments are known.

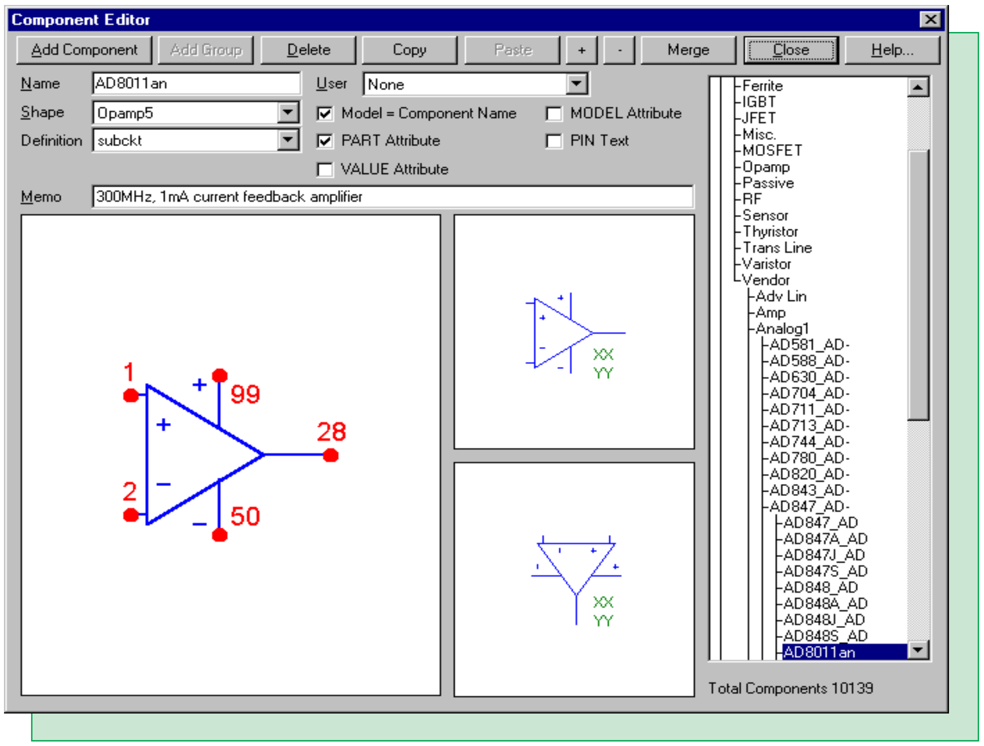
The Shape Editor

The next step in adding a subcircuit model to Micro-Cap is to create a new shape or choose an existing shape for the component. The Shape Editor can be found under the Windows menu. This editor controls the shape library for Micro-Cap. There are currently about 800 shapes available within the shape library. Any of these existing shapes may be used when a model is added to the component library.

In this case, the AD8100 model is a five pin operational amplifier device. Upon scrolling through the list of shapes available in the Shape Editor, the Opamp5 shape is found to be a good match for such a device. Once a shape is found that fits the part, the Shape Editor can be closed without performing any actions in it. However, if an appropriate shape is not found, then one will need to be created in the Shape Editor. This would be done by clicking on the Add command button, defining a shape name, and then using the shape elements to create the shape.

The Component Editor

After closing the Shape Editor, the next step is to enter the component information into the Component Editor. The Component Editor can also be found under the Windows menu, and it controls the component library for Micro-Cap. The settings for the AD8011 model appear in Figure 2.



Name: AD8011an
User: None
Shape: Opamp5
Definition: subckt
[✓] Model = Component Name
[ ] MODEL Attribute
[✓] PART Attribute
[ ] PIN Text
[ ] VALUE Attribute
Memo: 300MHz, 1mA current feedback amplifier

```
├Ferrite
├IGBT
├JFET
├Misc.
├MOSFET
├Opamp
├Passive
├RF
├Sensor
├Thyristor
├Trans Line
├Varistor
└Vendor
  ├Adv Lin
  ├Amp
  ├Analog1
    ├AD581_AD-
    ├AD588_AD-
    ├AD630_AD-
    ├AD704_AD-
    ├AD711_AD-
    ├AD713_AD-
    ├AD744_AD-
    ├AD780_AD-
    ├AD820_AD-
    ├AD843_AD-
    ├AD847_AD-
      ├AD847_AD
      ├AD847A_AD
      ├AD847J_AD
      ├AD847S_AD
      ├AD848_AD
      ├AD848A_AD
      ├AD848J_AD
      ├AD848S_AD
      ├AD8011an
```

Total Components 10139

Fig. 2 - AD8011 Component Settings

On the right hand side of the Component Editor is the Component selector. This hierarchical list controls the order of the components in the Component menu of the Schematic Editor. The model may be added into any of the available groups or into a group created by the user. To navigate through the list, double click on the group names to toggle them open or closed. Highlight the desired group name and then click on the Add Component command button. This will place a new entry into the list that will be edited to create the AD8011. In this case, the group chosen for the AD8011 was the Analog Library / Vendor / Analog1 / AD847_AD- group.

The Name field should be defined as 'AD8011an'. This is the name of the subcircuit as it appears on the .SUBCKT line in the AD8011.MOD file. For ease of use, the Name field and the subcircuit name should match exactly, but that is not a requirement for the Component Editor. Click in the field for Shape and choose the shape 'Opamp5' from the list. This list of shapes is managed by the Shape Editor. Click in the field for Definition and choose the definition 'subckt' from the list. This defines the component as one that uses a subcircuit model to obtain its electrical characteristics. Since the Name field matches the name of the subcircuit exactly, enable the Model=Component Name checkbox. This stops the Attribute dialog box from opening when the component is placed in a schematic and makes it a one step process for adding the part to a schematic. The Memo field contains a text description of the component and is optional.

The pin names are defined in the lower left hand window. Click in this window to invoke the Pin Name dialog box which defines the pin name and its type. For example, click on the non-inverting input lead, define the pin name as '1', and make sure it is an analog type. Hit OK. Then click on the inverting input lead, define the pin name as '2', and hit OK. Repeat this procedure for the other three pins. After placement, the pin may be moved by dragging on the pin dot, and the pin name may be moved by dragging on the name.

The two windows in the middle of the Component Editor control the initial placement of the attribute text in the schematic. The text location is specified by dragging the generic text block to the desired location relative to the shape in the two basic orientations. Hit the Close command button and then save the changes.

The NOM.LIB File

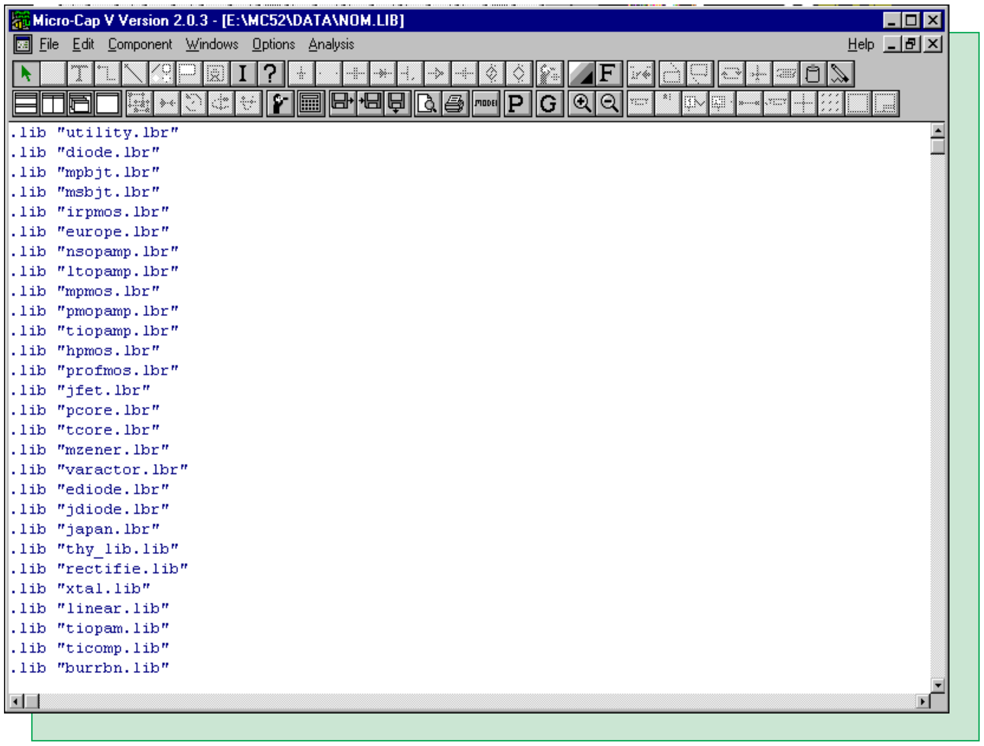
The NOM.LIB file is stored in the DATA subdirectory. This file is a text file that stores a list of all of the libraries that Micro-Cap will automatically access. Open this file in Micro-Cap, and it will appear as in Figure 3. On a new line in this file, add in the statement:

```
.lib  "AD8011.MOD"
```

If you choose not to keep the AD8011.MOD file in the DATA subdirectory, then a path would also have to be specified, such as:

```
.lib  "C:\MYLIBS\AD8011.MOD"
```

This statement would reference the file in the MYLIBS directory on the C: drive. Micro-Cap will now be able to find the AD8011 subcircuit model for any schematic.


Micro-Cap V Version 2.0.3 - [E:\MC52\DATA\NOM.LIB]
File Edit Component Windows Options Analysis Help

```
.lib "utility.lbr"
.lib "diode.lbr"
.lib "mpbjt.lbr"
.lib "msbjt.lbr"
.lib "irpmos.lbr"
.lib "europe.lbr"
.lib "nsopamp.lbr"
.lib "ltopamp.lbr"
.lib "mpmos.lbr"
.lib "pmopamp.lbr"
.lib "tiopamp.lbr"
.lib "hpmos.lbr"
.lib "profmos.lbr"
.lib "jfet.lbr"
.lib "pcore.lbr"
.lib "tcore.lbr"
.lib "mzener.lbr"
.lib "varactor.lbr"
.lib "ediode.lbr"
.lib "jdiode.lbr"
.lib "japan.lbr"
.lib "thy_lib.lib"
.lib "rectifie.lib"
.lib "xtal.lib"
.lib "linear.lib"
.lib "tiopam.lib"
.lib "ticomp.lib"
.lib "burrbn.lib"
```

Fig. 3 - The NOM.LIB File

The Schematic Editor

The final step is using the AD8011 component in a schematic. The AD8011 will appear in the Component menu in the same group that it was added into in the Component Editor. Click on the Component menu and then on the following sequence within the menu: Analog Library, Vendor, Analog1, AD847_AD-, AD8011an. Once the component is selected from the Component menu, it can be placed in the schematic and is available for immediate simulation. The only requirement for this part is that the NAME attribute must match exactly with the subcircuit name. In this case, that would be 'AD8011an' which is its default NAME attribute due to the settings in the Component Editor.

Figure 4 displays a test circuit for the AD8011 opamp. This test circuit uses the AD8011 opamp in a simple gain of two configuration. The opamp is running at +/- 5V.

Figure 5 displays the transient analysis results of the test circuit. The input to the opamp is a 100mV, 15ns pulse with a 50% duty cycle. Both the input and the output waveforms are being plotted over a time range of 50ns.

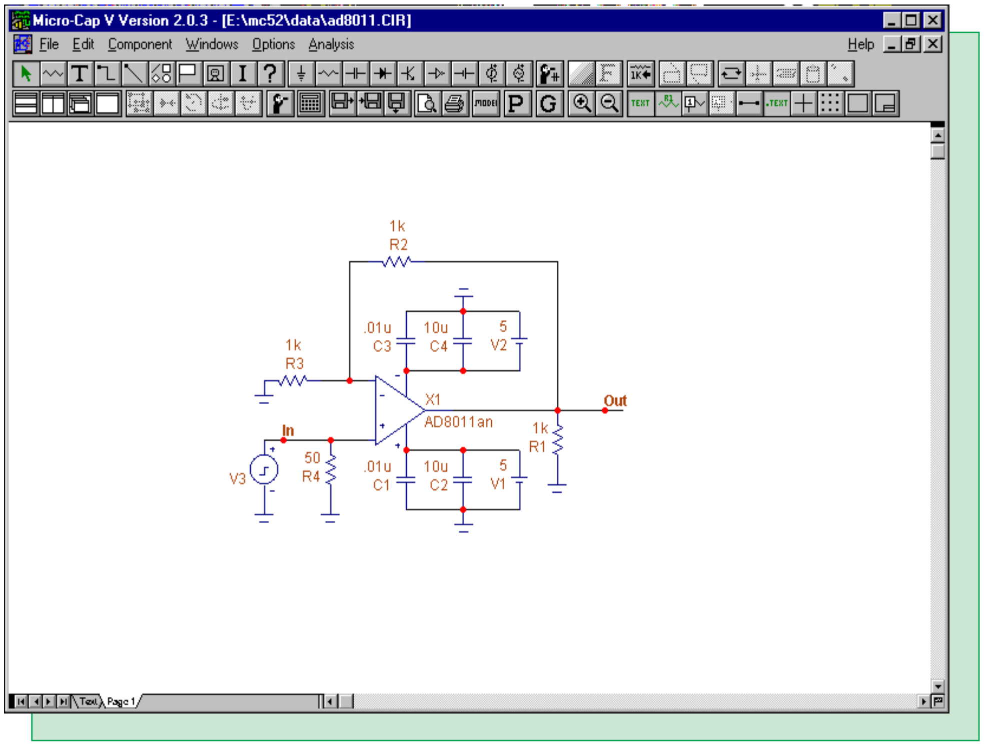

Micro-Cap V Version 2.0.3 - [E:\mc52\data\ad8011.CIR]
File Edit Component Windows Options Analysis Help
1k
R2
.01u
C3
10u
C4
5
V2
1k
R3
X1
AD8011an
Out
In
50
R4
V3
1k
R1
.01u
C1
10u
C2
5
V1
Text Page 1

Fig. 4 - AD8011 Test Circuit

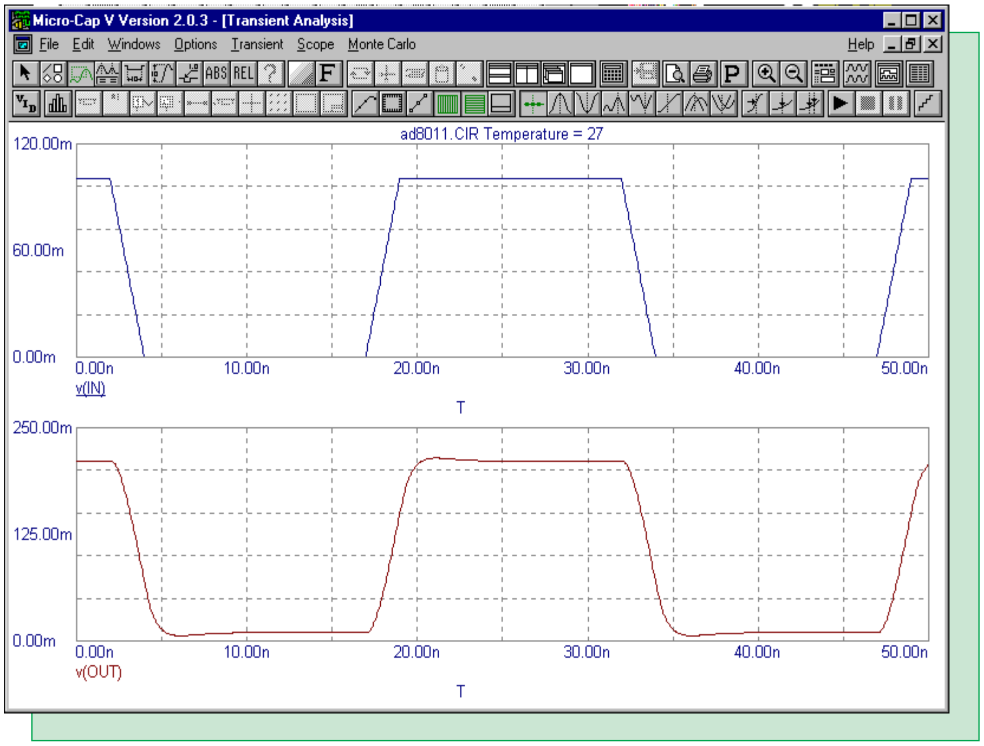
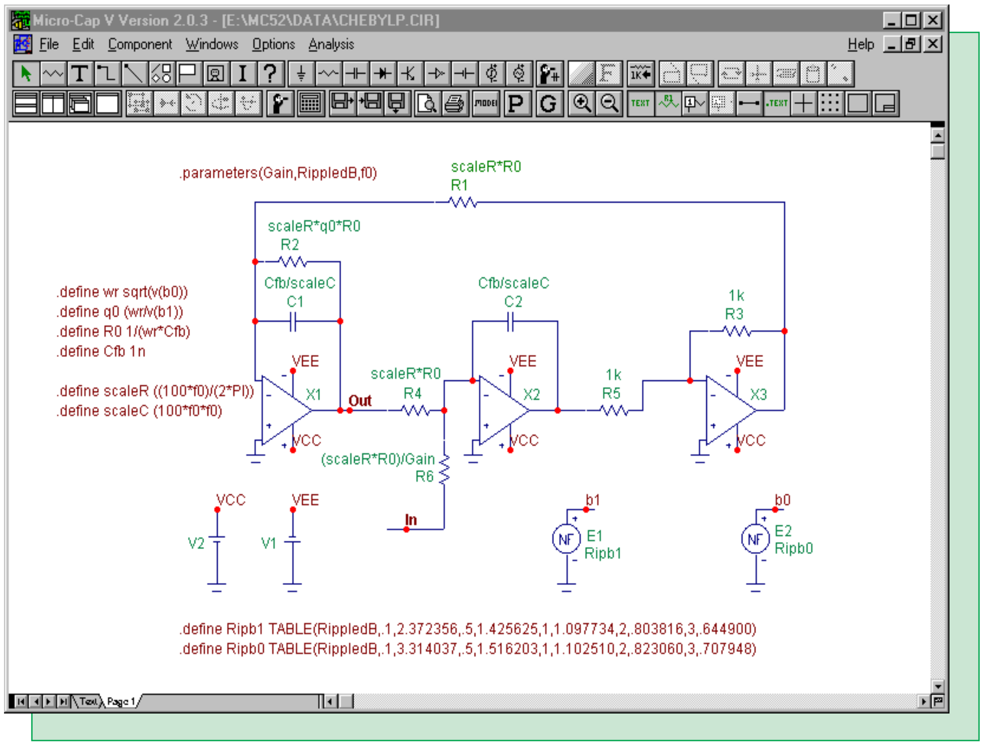


Fig. 5 - AD8011 Test Analysis

Chebyshev Filter Macro

Filters are a circuit element that seem to mesh perfectly with the macro capability of Micro-Cap. The macro capability is designed to produce components that can be varied through the use of parameters. Most filters consist of a basic structure whose component values can be modified through the use of well known equations. A macro component can be created that represents a specific filter's type, order, response, and implementation. The circuit in Figure 6 is the macro circuit for a low pass, 2nd order, Chebyshev filter with Tow-Thomas implementation.



```
.parameters(Gain,RippledB,f0)

.define wr sqrt(v(b0))
.define q0 (wr/v(b1))
.define R0 1/(wr*Cfb)
.define Cfb 1n

.define scaleR ((100*f0)/(2*PI))
.define scaleC (100*f0*f0)

.define Ripb1 TABLE(RippledB,.1,2.372356,.5,1.425625,1,1.097734,2,.803816,3,.644900)
.define Ripb0 TABLE(RippledB,.1,3.314037,.5,1.516203,1,1.102510,2,.823060,3,.707948)
```

Fig. 6 - Low Pass Chebyshev Filter Macro

The macro circuit uses three passed parameters to define the filter. These parameters are Gain, RippledB, and f0. The Gain parameter defines the magnitude of the gain of the filter in the pass-band region. The RippledB parameter defines the magnitude of the ripple in dB units in the passband region. The f0 parameter defines the cutoff frequency where the filter transitions between the passband and the stopband.

The equations for this macro were taken from Sedra and Smith's "Microelectronic Circuits" book. The Cfb value along with the resistance values for the R3 and R5 resistors are arbitrary. The equations taken from the book are as follows:

wr = sqrt(b0)
q0 = wr / b1
R0 = 1 / (wr * Cfb)
R1 = R0
R2 = q0 * R0
R4 = R0
R6 = R0 / Gain
C1 = C2 = Cfb

The scaleR and scaleC equations are used to scale the resistors and capacitors so that the filter's response occurs at the correct cutoff frequency. The scaleR factor multiplies the value of the R1, R2, R4, and R6 resistors, and the scaleC factor divides the value of the C1 and C2 capacitors.

The three opamps are defined as level 1 opamps and use all default model parameters. These opamps are ideal and will not have an affect on the expected response of the filter. The opamps are powered by +/- 15V batteries although with a level 1 opamp, the battery values will not have an effect.

In the equations for wr and q0, b1 and b0 parameters are specified. The values for b1 and b0 can be found in tables that specify the denominator polynomials for Chebyshev filters. The basic transfer function of the filter is:

```
1/(s*s + b1 * s + b0)
```

For this macro, the b1 and b0 values have been defined for when the ripple magnitude is .1dB, .5dB, 1dB, 2dB, and 3dB. These values have been implemented in the following two .define statements.

```
.define  Ripb1  TABLE(RippledB,.1,2.372356,.5,1.425625,1,1.097734,2,.803816,3,.644900)
.define  Ripb0  TABLE(RippledB,.1,3.314037,.5,1.516203,1,1.102510,2,.823060,3,.707948)
```

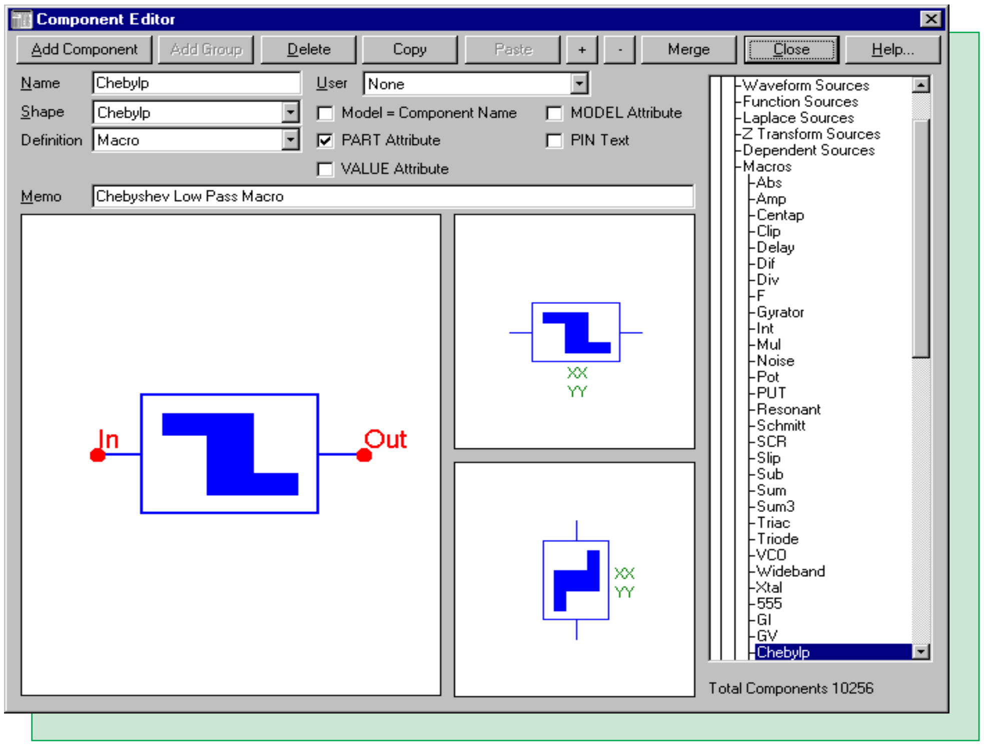
These two .define statements use the Table operator. The Table operator uses the format:

```
Table(x,x1,y1,...xn,yn)
```

where x is the input value and x1,y1,...xn,yn are the data pairs. In this case, the RippledB parameter will be passed into these .define statements. The output of the operator will be the b1 and b0 values that correspond to the RippledB input value. Values entered for RippledB that are not one of the specified data values above will have their output value interpolated from the existing data points. The interpolated values may not have the accuracy expected so it is recommended that the RippledB parameter be limited to only the ripple magnitudes that are stated within the Table operator. The outputs of the .define statements, Ripb1 and Ripb0, are then used to define the magnitude of the E1 and E2 components which are NFV sources. The E1 component has both its VALUE and FREQ attributes defined as Ripb1, and the E2 component has both its VALUE and FREQ attributes defined as Ripb0. The VALUE attribute will control the magnitude of the sources during transient analysis and any DC operating point calculations. The FREQ attribute overrides the VALUE attribute for AC analysis and will control the magnitude during an AC simulation. If the FREQ attributes are not defined, the NFV sources, since they are a constant voltage, would be treated as short circuits in AC, and the AC analysis results would be incorrect. Therefore, the voltages of the E1 and E2 components will always be the Ripb1 and the Ripb0 values for transient, AC, or DC analysis. The subsequent voltages at node b1 and b0 are then used in the .define statements for wr and q0.

The use of the Table operators and the E1 and E2 components lets us define the b1 and b0 parameters through the ripple magnitude parameter. Otherwise, the b1 and b0 parameters would have to be added into the .parameters statement, replacing the RippledB parameter. It is obviously much easier to state the value of the magnitude of the ripple rather than knowing the polynomial coefficients of the Chebyshev transfer function.

The Component Editor settings for the Chebyshev filter macro appear in Figure 7. The name of the macro is the same as the macro circuit that was created, Chebylp. The Chebylp shape was created in the Shape Editor for this macro and signifies the basic low pass function of the filter. The definition for the component is stated as 'macro'. Two pins have been added to the shape. These are the In and Out pins which will link to the nodes labelled In and Out in the macro circuit.


Component Editor

Name: Chebylp
User: None
Shape: Chebylp
Definition: Macro
☐ Model = Component Name
☐ MODEL Attribute
☑ PART Attribute
☐ PIN Text
☐ VALUE Attribute
Memo: Chebyshev Low Pass Macro

Total Components 10256

Fig. 7 - Component Editor Settings for the Chebyshev Macro

The filter macro is tested with the circuit in Figure 8. This circuit is simply a pulse source that feeds into the input of the filter. The pulse source defaults to a 1V AC small signal source for AC analysis. The macro has its VALUE attribute defined as:

Chebylp(1,ripple,1k)

This defines the filter with a gain of 1 and a cutoff frequency of 1KHz. The ripple magnitude has been defined with the parameter 'ripple', and a corresponding .define has been placed in the text area of the circuit that states:

.define ripple 2

Chebyshev filters are characterized by their ripple response in the passband and a monotonically decreasing transmission in the stopband. The analysis in Figure 9 is the AC response of this filter. The symbolic parameter ripple has been stepped at the values of .1, .5, 1, 2, and 3 with the List option. This will step the magnitude of the ripple in the passband response. The top waveform shows the response from 1Hz to 10KHz of db(v(OUT)), and the bottom waveform zooms in on the ripple of the db(v(OUT)) waveform. As can be seen from the top plot, the gain in the pass-band is at 0dB and the cutoff frequency is at 1KHz.

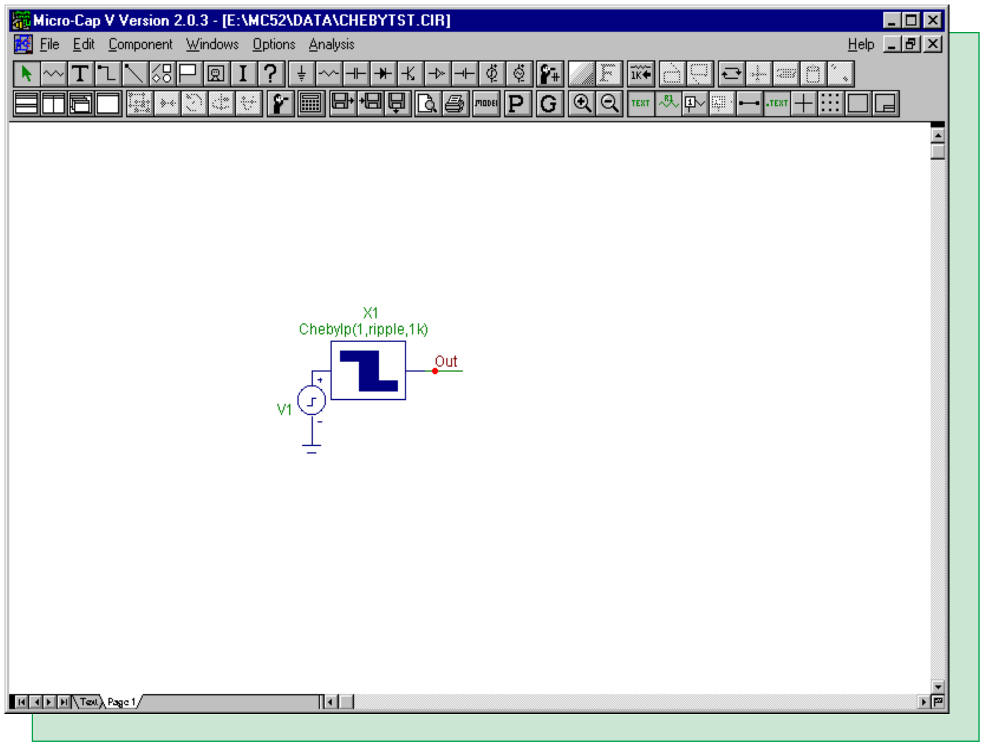

Micro-Cap V Version 2.0.3 - [E:\MC52\DATA\CHEBYTST.CIR]
File Edit Component Windows Options Analysis Help
X1
Chebylp(1,ripple,1k)
Out
V1

Fig. 8 - Chebyshev Filter Test Circuit

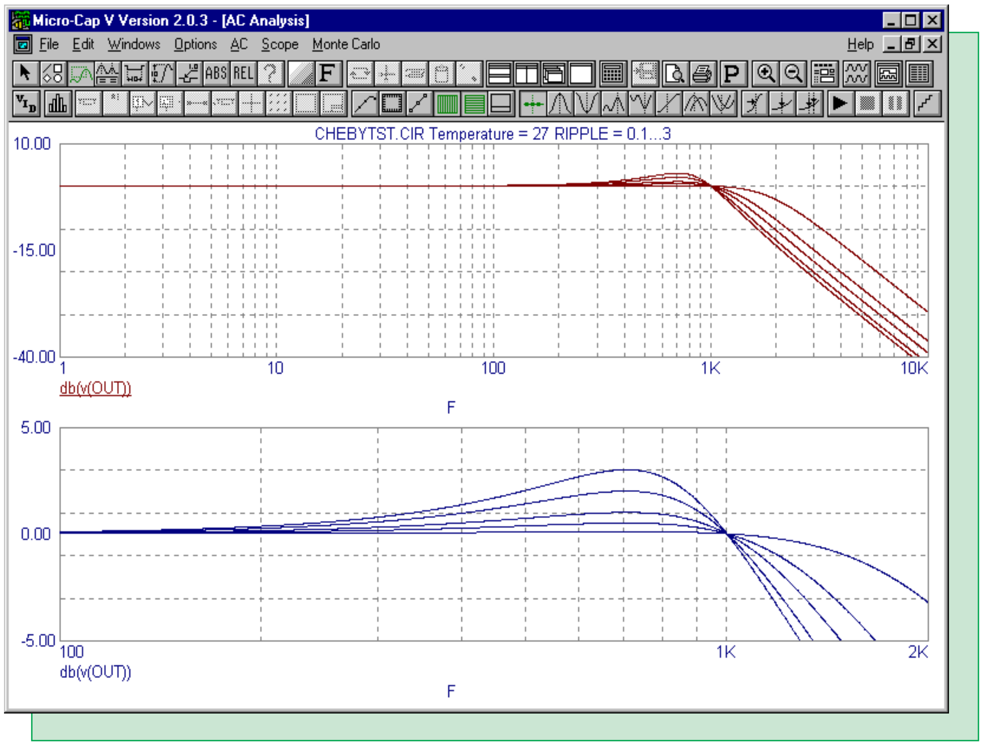


Fig. 9 - Chebyshev Filter AC Analysis

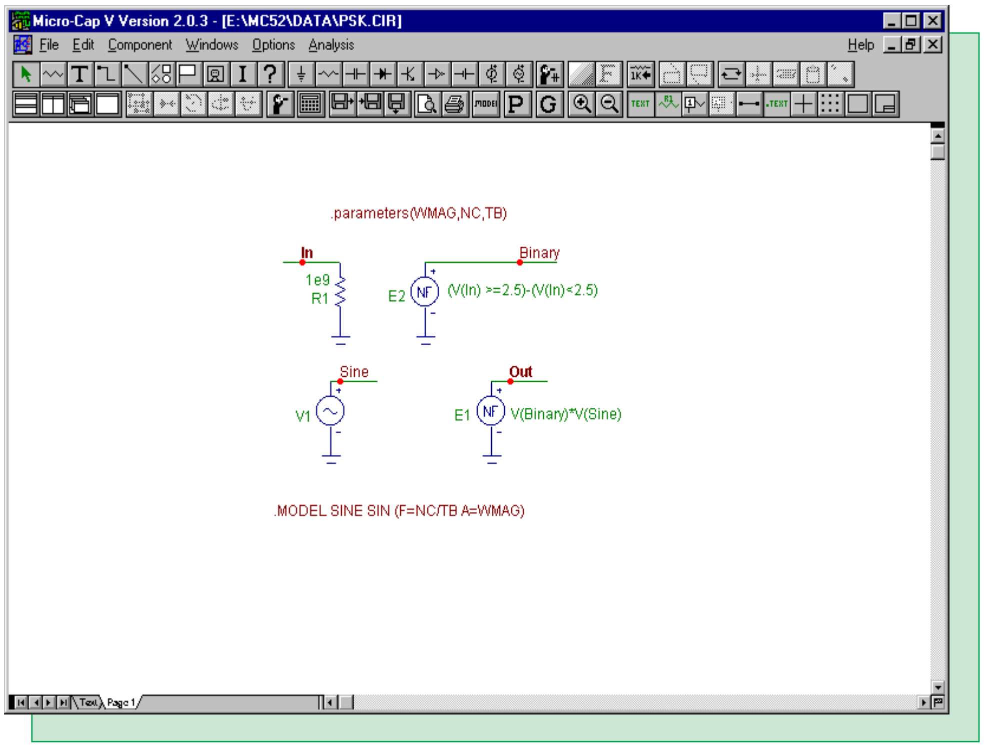
PSK Modulator Macro

In communication systems, a binary PCM (Pulse-Code Modulation) signal is often superimposed onto a carrier signal. There are many modulation techniques for this procedure among which are amplitude, frequency, and phase modulation. Phase modulation, also known as phase-shift keying (PSK) modulation, produces a waveform with the following characteristics:

$V_{PSK}(t) = A*\cos(w_o*t + \text{phase}(t))$

where A is a fixed amplitude and phase will vary according to the binary waveform. A common PSK waveform will have its phase at 0 degrees when the binary signal is at a one state and 180 degrees when the binary signal is at a zero state.

The macro circuit for a PSK modulator appears in Figure 10. This macro can accept either a digital or an analog waveform as its input and will produce the PSK modulated waveform at its output. The PSK macro accepts three parameters: WMAG, NC, and TB. WMAG defines the magnitude of the output waveform. NC defines the number of cycles of the output waveform that will occur in the duration of one bit of the input waveform. TB defines the duration of a single bit in seconds.


Micro-Cap V Version 2.0.3 - [E:\MC52\DATA\PSK.CIR]
File Edit Component Windows Options Analysis Help
In
Binary
1e9
R1
E2
Sine
Out
V1
E1
Text Page 1

.parameters(WMAG,NC,TB)

(V(In) >=2.5)-(V(In)<2.5)

V(Binary)*V(Sine)

.MODEL SINE SIN (F=NC/TB A=WMAG)

Fig. 10 - PSK Modulator Macro Circuit

The macro consists of a resistor, a sine source, and two nonlinear function voltage sources. The resistor is present at the input node for two reasons. The first reason is to provide a DC path to ground for the In node so that any element may be connected to it. The second reason is that the NFV sources are only able to work with analog references. If the input waveform happens to be a digital waveform, the resistor will force Micro-Cap to convert it to its equivalent analog voltage for use with the nonlinear function voltage source, E2. The value of the resistor is set to 1e9 ohms so that it will not have a loading effect. The NFV source, E2, has its VALUE attribute defined as:

(V(In) >= 2.5) - (V(In) < 2.5)

This equation will convert the input waveform into a signal that has a value of either 1V or -1V. If the voltage at node In is greater than or equal to 2.5V, than the source will produce an output of 1V. If the voltage at node In is less than 2.5V, then the source will produce an output of -1V. The 2.5V threshold is arbitrary. In this case, it was setup to handle a standard TTL digital waveform. The threshold should be approximately in the middle of the one and zero state voltages of the input waveform. The sine source, V1, has been defined using the following model statement:

.MODEL SINE SIN (F=NC/TB A=WMAG)

where the amplitude of the source is defined by the parameter WMAG, the frequency of the source is determined from the ratio of the NC and TB parameters, and the phase of the source is at its default value of 0. Finally, the NFV source, E1, multiplies the voltages of the E2 and V1 sources. This source produces the PSK output. When the input waveform is at a one state, then the sine source waveform is multiplied by the value 1 which produces a sine output with a 0 degrees phase shift. When the input waveform is at a zero state, then the sine source waveform is multiplied by the value -1 which produces a sine output with a 180 degrees phase shift.

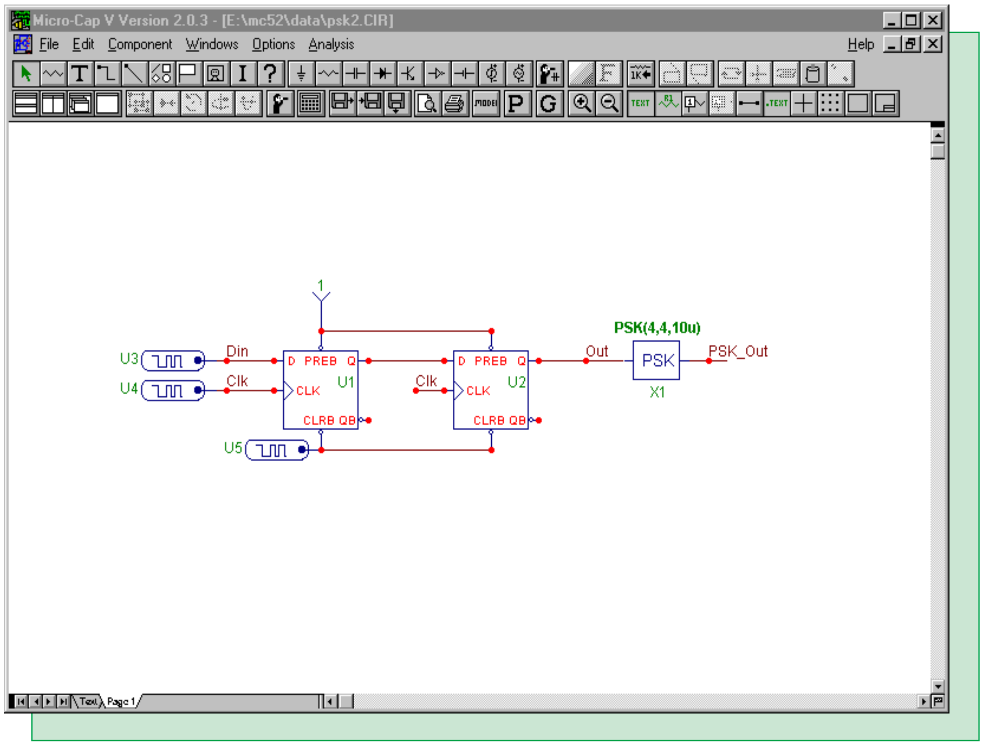

Micro-Cap V Version 2.0.3 - [E:\mc52\data\psk2.CIR]
File Edit Component Windows Options Analysis Help
1
PSK(4,4,10u)
U3
U4
U5
Din
Clk
D PREB Q
CLK
CLRB QB
U1
U2
Out
PSK
PSK_Out
X1
Text Page 1

Fig. 11 - PSK Macro Test Circuit

The test circuit for the PSK macro is displayed in Figure 11. The circuit consists of a simple two bit shift register that feeds the PSK macro. The two D flip-flops are defined with zero gate delays. The clock input to the flip-flops has a period of 10us. The preset inputs have been wired to a Fixed Digital component which produces a one state on those pins. The clear inputs have a short zero pulse of 100ns to initialize the outputs of the flip-flops to zero at the beginning of the simulation. The PSK macro has its VALUE attribute defined as:

PSK(4,4,10u)

which will produce an output waveform with a 4V magnitude at a frequency of 400KHz.

Figure 12 displays the results of an 80us transient analysis simulation. The top waveform, V(Out), is the binary input to the PSK macro. The bottom waveform, V(PSK_Out), is the modulated waveform. Note the 180 degree phase shifts of the PSK_Out waveform whenever the V(Out) waveform transitions states.

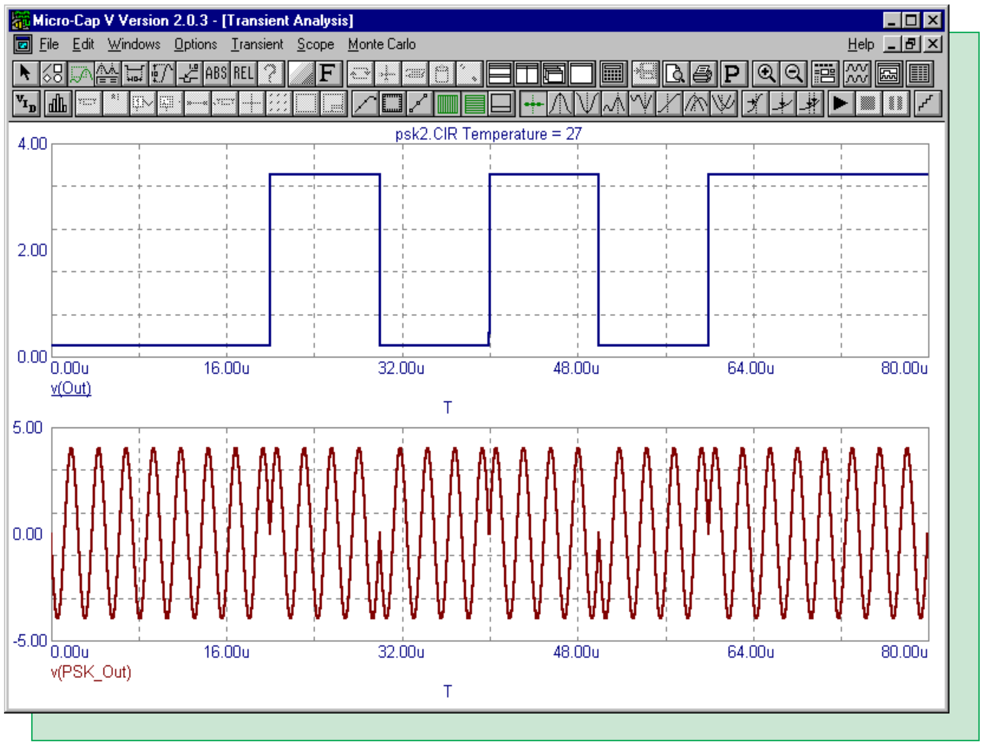

Micro-Cap V Version 2.0.3 - [Transient Analysis]
File Edit Windows Options Transient Scope Monte Carlo Help
psk2.CIR Temperature = 27
4.00
2.00
0.00
0.00u
16.00u
32.00u
48.00u
64.00u
80.00u
v(Out)
T
5.00
0.00
-5.00
v(PSK_Out)

Fig. 12 - PSK Macro Test Output

Storing the Time of an Event

In some cases, a circuit may need a time based element that is triggered when an event elsewhere in the circuit occurs. Micro-Cap has the t variable, which is the value of the simulation time in transient analysis, in order to include time in circuit expressions. However, practical use of this variable assumes the expression starts at t=0 or that the time of the triggering event is known. If the time of the triggering event is unknown, then that event time must be dynamically stored in the circuit for use in an expression. The circuit in Figure 13 displays a technique for storing the time that an event occurs.

In this circuit, the time that the switch opens is used as a reference in a time based exponential waveform. The trigger portion of the circuit consists of a battery, a switch, and a resistor. The nodes swp and swn will essentially be shorted when the switch is closed and will have approximately 10V across them when the switch is open. The sources within the dashed box monitor the switch voltage and will store the time that the switch opens. The S1 source is a sample and hold source. Its EXPR attribute is defined as:

V(tevent)

which is the voltage across E1, and it is sampling the voltage every 1ns. The NFV source, E1, has its VALUE attribute defined as:

if(v(swp,swn)<2,t,v(samp))

The output of the E1 source will be the time of the simulation when V(swp,swn) is less than 2 and will be the value of the S1 source when V(swp,swn) is greater than or equal to 2. The S1 source mirrors the E1 source which is a ramp of voltage equal to the simulation time until the switch opens, when both voltage sources lock on to the value of the last time stored in the sample and hold when the switch was closed. If the sample period of the sample and hold source is small relative to the simulation time, this stored value will essentially be the time that the event occurs. This value will be stored in both sources until the switch closes upon which the sources will again have the value of the current simulation time. Note that the sampling period of the sample and hold source will limit the maximum timestep of the simulation and may effect the speed of the simulation if set very small. The voltage of either the E1 or S1 sources may subsequently be used in other equations as a time reference. The NFV source, E2, has its VALUE attribute defined as:

(10*exp(-(t-v(tevent))/.3u))

When the switch is closed, the source will produce a constant 10V since t and v(tevent) are equivalent. When the switch opens, the source produces an exponentially decaying source using the v(tevent) voltage as a reference.

The results of the simulation are displayed in Figure 14. The top waveform is the voltage across the switch, v(swp,swn), with the switch opening at 500ns. The middle waveform is the voltage of the E1 source, v(tevent). As can be seen in the plot, when the switch opens, the voltage at this node locks in at the time of the event which in this case is 500nV. The bottom waveform is the voltage of the E2 source, v(out), and begins its exponential decay upon the opening of the switch.

This storage technique will work with any other dynamic circuit expression such as voltage, current, instantaneous power, etc. In the If statement, simply replace the variable t with the desired expression that is to be stored.

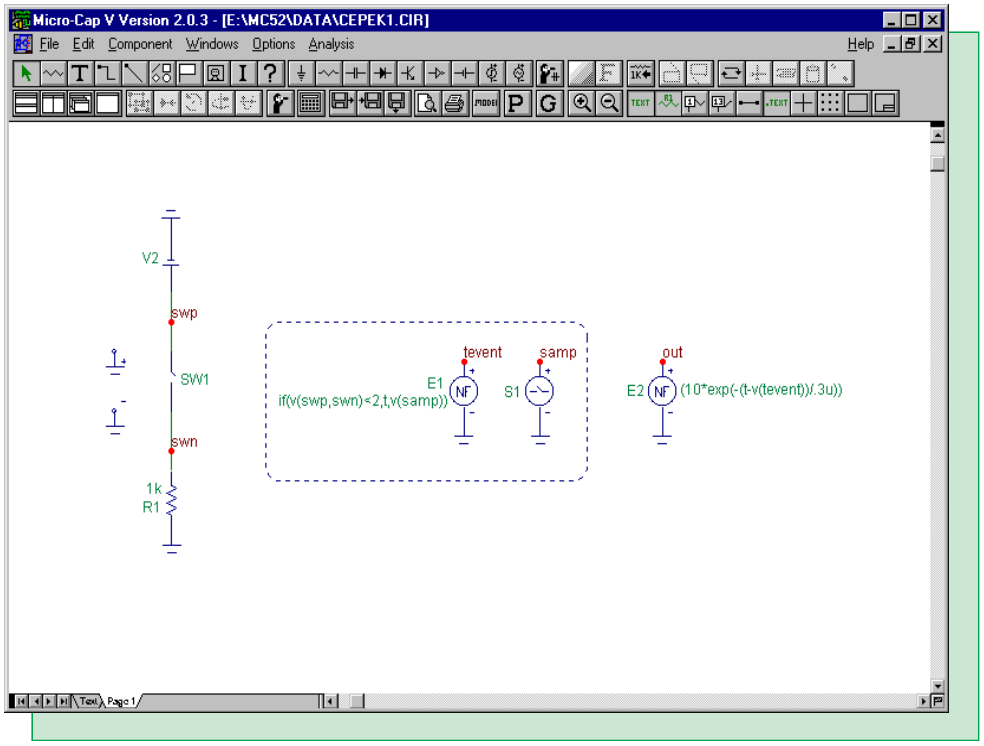

Micro-Cap V Version 2.0.3 - [E:\MC52\DATA\CEPEK1.CIR]
File Edit Component Windows Options Analysis Help
V2
swp
SW1
swn
1k
R1
tevent
samp
E1
if(v(swp,swn)<2,t,v(samp))
S1
out
E2
(10*exp(-(t-v(tevent))/.3u))
Text Page 1

Fig. 13 - Event Time Storage Circuit

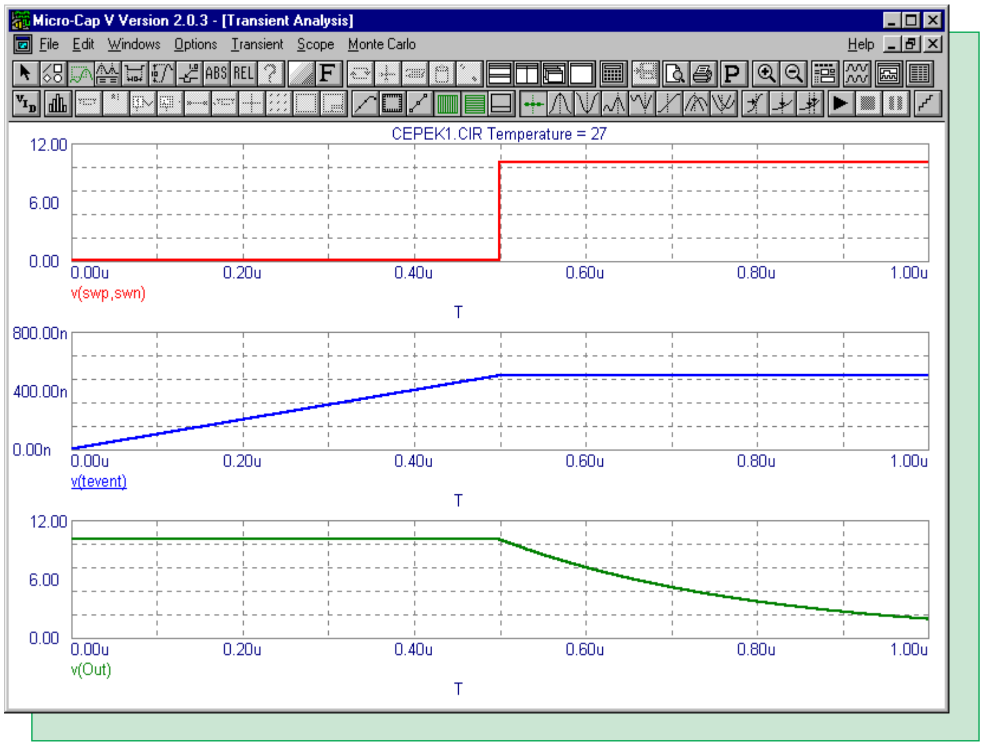


Fig. 14 - Event Time Storage Analysis

Product Sheet

Latest Version numbers

Micro-Cap V	Version 2.03
Micro-Cap IV IBM/NEC/MAC	Version 3.04

Spectrum's numbers

Sales	(408) 738-4387
Technical Support	(408) 738-4389
FAX	(408) 738-4702
Email sales	sales@spectrum-soft.com
Email support	support@spectrum-soft.com
Web Site	http://www.spectrum-soft.com

Spectrum's Products

• Micro-Cap V	$3495.00
• Upgrade from MC5 Version 1 to MC5 Version 2	$500.00
• Upgrade from MC4 to MC5	$1000.00
• Upgrade from MC3 to MC5	$2000.00

You may order by phone or mail using VISA, MASTERCARD, or American Express. Purchase orders accepted from recognized companies in the U.S. and Canada. California residents please add sales tax.