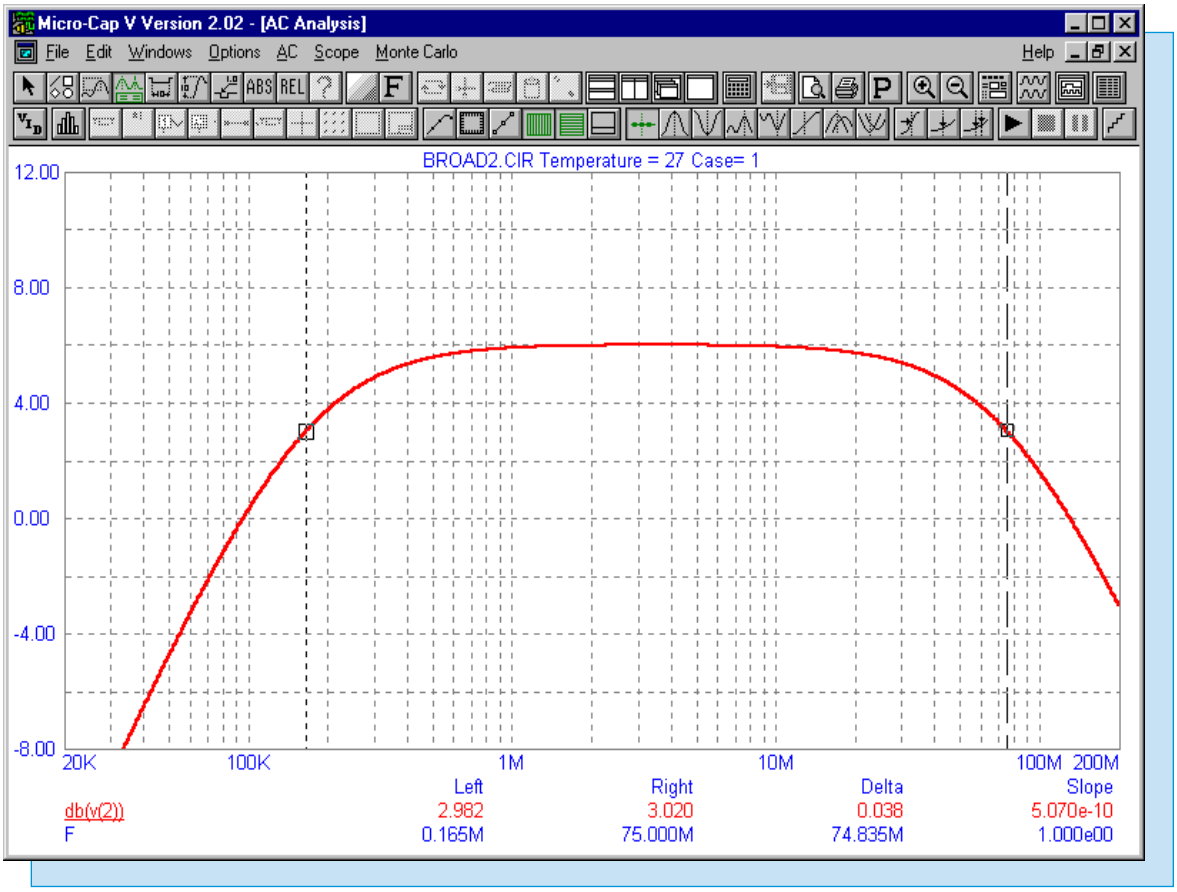
spectrum news

Applications for Micro-Cap™ Users

Winter 1998

Wideband Transformer Macro



Featuring:

- Modeling a Commercially Available Digital Component
- Wideband Transformer Macro
- Importing Graphical Templates
- Sweeping a Resistor in DC Analysis

News In Preview

This issue features an article that describes the process of creating a commercially available digital component from data book information. It then describes how to add this new component into the Component Library. A second article describes creating a macro for a wideband transformer. This macro can convert the frequency data from a data book into an equivalent Micro-Cap transformer model. The third article describes the process of plotting a graphical template into a Micro-Cap simulation. The template is useful when designing a circuit that must be within certain specifications. The final article shows how to sweep a resistor in DC analysis rather than the usual voltage or current source.

Contents

Book Recommendations

- *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

Micro-Cap V Question and Answer

Question: I would like to compare the outputs of different schematics on the same analysis plot. Is there a way to do this in Micro-Cap?

Answer: The Import operator is designed for this purpose. The Import operator can import a waveform from an output text file that has been created by Micro-Cap or by another SPICE program. The following example imports the output waveform, V(Out), of the circuit, WAVEOUT.CIR, into the analysis of the circuit, WAVEIN.CIR. The example will be done in transient analysis, but this procedure can be used in AC and DC analysis as well.

Load the file WAVEOUT.CIR. Go to the Analysis menu and click on Transient Analysis. The Transient Analysis Limits dialog box will appear. For each waveform, there is a set of icons for the waveform options. Click on the icon that looks like  for the waveform V(Out). When enabled, this will write the results of the waveform into the numeric output file. Set the value in the Number of Points field. This value determines the number of data points that are to be written in the text file. Note that in AC analysis, the Frequency Step must be changed from Auto for this to have an effect. The number of data points in the text file controls the accuracy of the imported waveform. Setting this field to 1000 should cover most waveforms. Run the analysis, and then click on the Transient menu and choose Numeric Output. This will load up the file, WAVEOUT.TNO, which is the numeric output file that was just created. In this file will be operating point information, and two columns of tabular data that represent time and V(Out). Go to the File menu and choose Save As. Save this file to a different name such as WAVEOUT.OUT. This doesn't have to be done, but it prevents the file from being overwritten if you run the transient analysis again. Load the file WAVEIN.CIR. Go to the Analysis menu and click on Transient Analysis. On a new waveform line, place the following into the Y Expression field:

Import(Waveform.out,V(Out))

The syntax of the Import operator is:

Import(f,y)

where f is the file name, and y is the waveform that is to be imported from the file f. The X expression must be T. Run the analysis, and the waveform V(Out) will be plotted along with any other waveforms that have been defined.

There are two other things that the user needs to be aware of in using the Import operator. First of all, the X expression for the Import operator is limited. For transient analysis, the X expression must be T. For AC analysis, the X expression must be F. For DC analysis, the X expression must be the voltage or current of the Input 1 source being swept. Also, when creating the numeric output file in Micro-Cap, the waveform name is truncated to 9 characters. For example, if you are saving the waveform dB(V(Out)) to the numeric output file, it will actually place the waveform in the file as dB(V(Out). When you try to import this, it will complain due to the mismatched parentheses. The numeric output file must be manually edited so that the waveform name will be valid. Two ways to fix the above waveform would be to change the header in the numeric output file to either of the following: dB(V(Out)) or dB(VOut), and then use the new name in the Import operator. The waveform name in the Import operator must match exactly with the waveform header, and the parentheses must match.

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.

Pop Up Analysis Variable List

An easy way to view circuit variables and operators that are available for an analysis is through the Pop Up Analysis Variable List. This pop up list appears when the right mouse button is clicked in a Y Expression field in the Analysis Limits dialog box. This list shows available variables, constants, functions, and operators. It also lets you expand the Y Expression field which allows you to view and edit longer expressions with greater ease.

The variables section includes node voltages, digital states, device voltages, device currents, resistances, capacitances, inductances, charges, fluxes, H fields, and B fields. The constants include the fixed variables, J, PI, TRUE, and FALSE. The functions include complex, DSP, calculus, miscellaneous, and transcendental functions. The operators include relational, boolean, and arithmetic operators.

Choosing one of the variables, constants, functions, or operators from the list will place it in the Y Expression field from which the mouse was clicked in. The variables, constants, and operators will be placed where the cursor is at. To set this location, you would need to place the cursor with the left mouse button prior to invoking the Variable List. The functions will wrap around everything that was previously in the expression field and use all of it as a single parameter. For functions that need multiple parameters, the other parameters would need to be specified after the function has been placed in the field.

Clicking the right mouse button in the P, X Expression, Range, and FMT fields invokes a simpler menu showing suitable choices. This feature is described in the Transient Analysis chapters of both manuals.

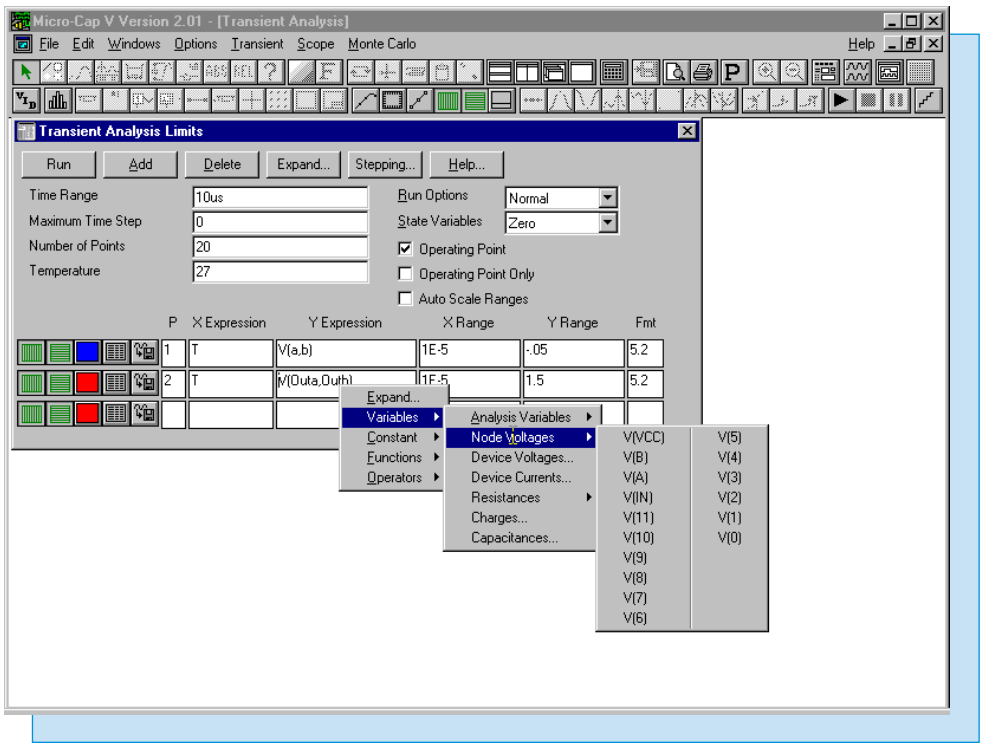


Fig. 1 - Pop Up Analysis Variable List

Modeling a Commercially Available Digital Component

Although Micro-Cap has many digital models, a user may need one that does not exist in the library. In that case, a subcircuit or macro model of the digital component would need to be created. The following article covers the process of creating a digital subcircuit model from data book information. The model that is to be created is the 74195, which is a 4-Bit Parallel-Access Shift Register. Data book information about this component can be viewed in the "TTL Logic Data Book" from Texas Instruments, 1988, p. 2-655 to 2-662. The logic diagram appears in Figure 2. This register features parallel inputs, parallel outputs, J-K serial inputs, shift/load control, and an overriding clear.

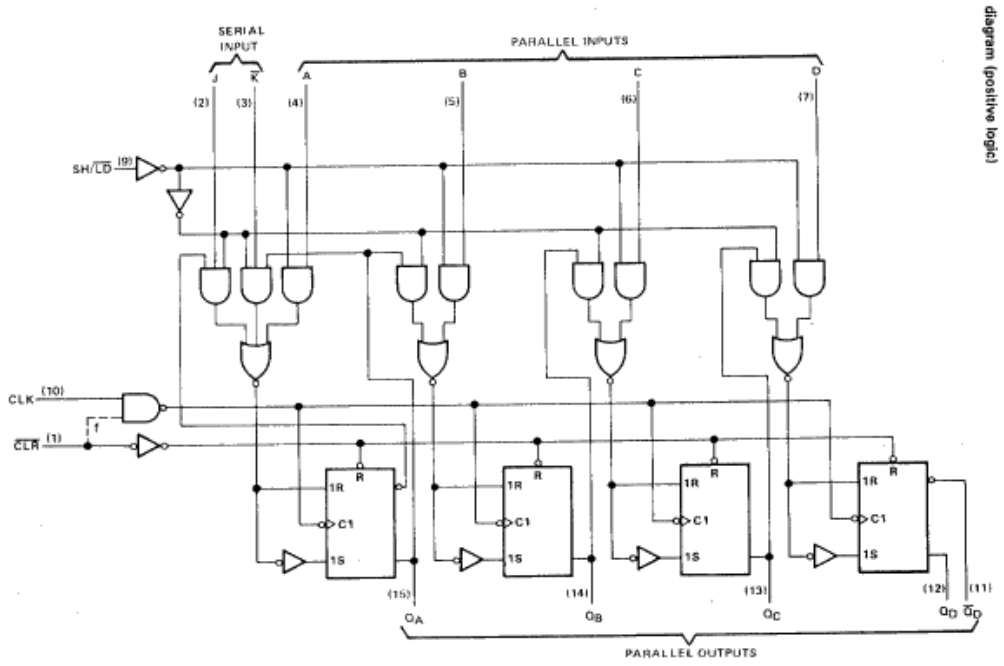


Fig. 2 - 74195 Logic Diagram

The subcircuit listing for the 74195 component appears in Listing 1. This subcircuit is also available in the DIG195.LIB library file that comes with Micro-Cap. The model consists of a subcircuit header, a .Ends statement, and four different devices: a logic expression device, a J-K flip-flop array, a pin-to-pin delay device, and a constraint device. The description of each of these is as follows:

The Subcircuit Header (.SUBCKT)

The subcircuit header denotes the beginning of the subcircuit definition. It defines the node names that are used in calling the subcircuit, along with optional nodes and parameters. All but two of the 74195 's pins have been listed in the subcircuit call. The missing pins are the power and ground pins. These two pins have been placed as optional pins through the "optional:" keyword. Their default values are the globally defined values of $G_DPWR and $G_DGND. Having the power and ground pins as optional pins lets you place the 74195 on a schematic without having to externally connect it to power supplies. The two parameters, MNTYMXDLY and IO_LEVEL, control the digital delay and the digital I/O value, respectively, for the subcircuit and are defined through the "params:" keyword. The default value of 0 specifies that these parameters will take the values of DIGMNTYMX and DIGIOLVL, which are set in the Global Settings.

```
.SUBCKT 74195  CLK SH/LDBAR J KBAR CLRBAR A B C D QA QB QC QD QDBAR
+  optional: DPWR=$G_DPWR DGND=$G_DGND
+   params: MNTYMXDLY=0 IO_LEVEL=0

U74195 LOGICEXP(13,9) DPWR DGND
+      CLK SH/LDBAR J KBAR CLRBAR A B C D QA_O QB_O QC_O qabar
+      j0 j1 j2 j3 k0 k1 k2 k3 clock
+      D0_GATE IO_STD MNTYMXDLY={MNTYMXDLY} IO_LEVEL={IO_LEVEL}
+
+    LOGIC:
+      clock = {(~(CLK & CLRBAR))}
+      k0 = {~((SH/LDBAR & J & qabar) | (SH/LDBAR & KBAR & QA_O) |
+             (~SH/LDBAR & A))}
+      k1 = {~((QA_O & SH/LDBAR) | (~SH/LDBAR & B))}
+      k2 = {~((QB_O & SH/LDBAR) | (~SH/LDBAR & C))}
+      k3 = {~((QC_O & SH/LDBAR) | (~SH/LDBAR & D))}
+      j0 = {(~k0)}
+      j1 = {(~k1)}
+      j2 = {(~k2)}
+      j3 = {(~k3)}

Uf0  JKff(4)  DPWR DGND
+    $D_HI CLRBAR clock
+    j0 j1 j2 j3 k0 k1 k2 k3
+    QA_O QB_O QC_O QD_O  qabar qbbar qcbar QDBAR_O
+    D0_EFF IO_STD MNTYMXDLY={MNTYMXDLY} IO_LEVEL={IO_LEVEL}

Udly PINDLY (5,0,2) DPWR DGND
+    QA_O QB_O QC_O QD_O QDBAR_O
+    CLK CLRBAR
+    QA QB QC QD QDBAR
+    IO_STD MNTYMXDLY={MNTYMXDLY} IO_LEVEL={IO_LEVEL}
+
+    BOOLEAN:
+     CLOCK = {CHANGED_LH(CLK,0)}
+     CLEAR = {CHANGED_HL(CLRBAR,0)}
+
+    PINDLY:
+      QA QB QC QD QDBAR = {
+          CASE(
+               CLEAR, DELAY(-1,19ns,30ns),
+               CLOCK & TRN_LH, DELAY(-1,14ns,22ns),
+               CLOCK & TRN_HL, DELAY(-1,17ns,26ns),
+               DELAY(-1,20ns,31ns)
+             )
+          }
```

Listing 1 - 74195 Subcircuit

```
Ucnstr CONSTRAINT(9) DPWR DGND
+    CLRBAR CLK A B C D J KBAR SH/LDBAR
+    IO_STD IO_LEVEL={IO_LEVEL}
+
+     FREQ:
+       NODE = CLK
+       MAXFREQ = 30MEG
+     WIDTH:
+       NODE = CLK
+       MIN_HI = 16ns
+     WIDTH:
+       NODE = CLRBAR
+       MIN_LO = 12ns
+     SETUP_HOLD:
+      CLOCK LH = CLK
+       DATA(6) = A B C D J KBAR
+       SETUPTIME = 20ns
+     SETUP_HOLD:
+      CLOCK LH = CLK
+       DATA(1) = CLRBAR
+       SETUPTIME_LO = 25ns
+     SETUP_HOLD:
+      CLOCK LH = CLK
+       DATA(1) = SH/LDBAR
+       SETUPTIME_LO = 25ns
+       SETUPTIME_HI = 10ns

.ENDS  74195
```

Listing 1 - 74195 Subcircuit (continued)

The Logic Expression Device (U74195)

The logic expression device defines the combinatorial logic that feeds into the flip-flops. This device uses 13 inputs to produce the four J, four K, and one clock input to the J-K flip-flop array. Note that five of the inputs to this device are from the outputs of the flip-flop array. Feedback external to the logic expression is allowed, but feedback in the internal logic expressions is forbidden. The logic expression device is used as a simpler means of defining the logic for a group of gates. The operators available are:

~	Unary negation
&	AND
^	Exclusive OR
|	OR

By examination, it can be seen that the logic expressions in this device match exactly with the logic in the logic diagram. The k0-k3 outputs correspond to the 1R inputs of the flip-flops, the j0-j3 outputs correspond to the 1S inputs of the flip-flops, and the clock output corresponds to the C1 inputs of the flip-flops. This device has been defined with the D0_GATE timing model, which is a zero delay timing model, and the IO_STD I/O model, which represents the standard TTL family of digital components.

The J-K Flip-Flop Array Device (Uf0)
The JKFF device defines an array of four J-K flip-flops. Though the logic diagram displays S-R flip-flops, the J-K flip-flops can be a direct substitute since the indeterminate state of S=1 and R=1 is logically impossible. The Presetbar pin has been defined as $D_HI, which forces that pin to a high state during simulation so it will have no effect on the flip-flops. The timing model for the array has been defined as D0_EFF which is a zero delay and zero constraint timing model.

The Pin-to-Pin Delay Device (Udly)
The PINDLY device provides a way of modeling conditional pin delays. This device is extremely useful in modeling digital components as all delays may be concentrated in one device, no matter what the trigger of the delay is, instead of having to allocate delays into individual gates in order to produce the correct propagation delays. In this case, the delays for five output pins are being defined with only two reference pins, CLK and CLRBAR. All five outputs share the same propagation delays and therefore, can all be defined in the same case. When an output changes, the pin-to-pin delay device will go through the output's delay conditions and choose the first delay that produces a true result. This could produce an incorrect result if a delay condition after the first true one has the desired propagation delay. Therefore, delay conditions that require a higher precedence should be placed at the top of the list. For example, the 74195 has an overriding clearbar input, so the delay condition that is dependent on the CLRBAR input should be at the top of the list. The -1 value in some of the delay values defines an unspecified setting and the actual delay time will be calculated from the Unspecified Propagation Delay rules. For the 74195, a change in the CLRBAR input from high to low produces a delay of (-1,19ns,30ns). A change in the CLK input from low to high with a corresponding change of low to high in one of the outputs produces a delay of (-1,14ns,22ns). A change in the CLK input from low to high with a corresponding change of high to low in one of the outputs produces a delay of (-1,17ns,26ns). If all of these conditions fail, then the delay of (-1,20ns,31ns) will be used. The delays are specified as (min,typ,max). The specified delays for the 74195 can be found in the Switching Characteristics table in the TI data book.

The Constraint Device (Ucnstr)
The CONSTRAINT device provides a means to check complex conditional timing parameters. It is capable of checking setup time, hold time, pulse width, and frequency specifications. No timing model is required for the device, only an I/O model is needed. The CONSTRAINT device for the 74195 checks the maximum frequency of CLK, the high level pulse width of CLK, the low level pulse width of CLRBAR, and the setup times of all of the inputs in relation to the low to high transition of the CLK input. Any number of FREQ:, WIDTH:, and SETUP_HOLD: keywords may be used. The constraint timing specifications can be found in the Recommended Operating Conditions table in the TI data book. Any violation of these constraints produces a warning, but does not affect the simulation.

The .ENDS Statement
The .ENDS statement signifies the end of the subcircuit. All devices used to model the component must appear between the .SUBCKT and .ENDS statements. The component name that appears on the .ENDS line is optional.

Once the model is finished, the name of the file that contains the model should be placed in the file NOM.LIB if it is not already there. The NOM.LIB file contains a list of all libraries that Micro-Cap accesses automatically. The NOM.LIB file can be found in the DATA subdirectory. If the 74195 was created in a file called 74195.LIB which resides in the DATA directory, then place the following new line in the NOM.LIB file:

.lib "74195.lib"

Adding the 74195 to the Component Menu

To access the 74195 through the Component menu, the device must be linked to a shape and given pins in the Component Editor. Figure 3 displays the settings in the Component Editor that were used for the 74195. To add the device, highlight the group that you would like to place it in, and then click on the Add Component command button. This places a new entry into the high-lighted group.

The name should be the name that appears on the .SUBCKT line. The shape chosen was the 74195, but it may be any shape that has been created through the Shape Editor. The definition must be 'subckt' to define it as a subcircuit model.

Next, the pins from the subcircuit header need to be entered in the Shape/pin display. Click in the Shape/pin display window at the end of the first lead, and type in the first pin name 'SH/LDBAR'. Click OK. Repeat this process with all of the other pins in the subcircuit header. The pin names must match exactly with the names in the subcircuit header. The pin connection dots and the pin names may each be dragged to their desired location. For easier connections on the schematic, the pin connection dots should be at the end of each lead. The optional power and ground pins could be added in at this point if the part needed to use power supplies available only in the schematic. If the pins are not added, the 74195 will use the power supplies defined in the DIGIO.LIB file. We will leave out the optional pins because their default values are suitable.

Set the checkbox options to how they appear in Figure 3. Close and save the component settings. The 74195 component can now be accessed through the Component menu.

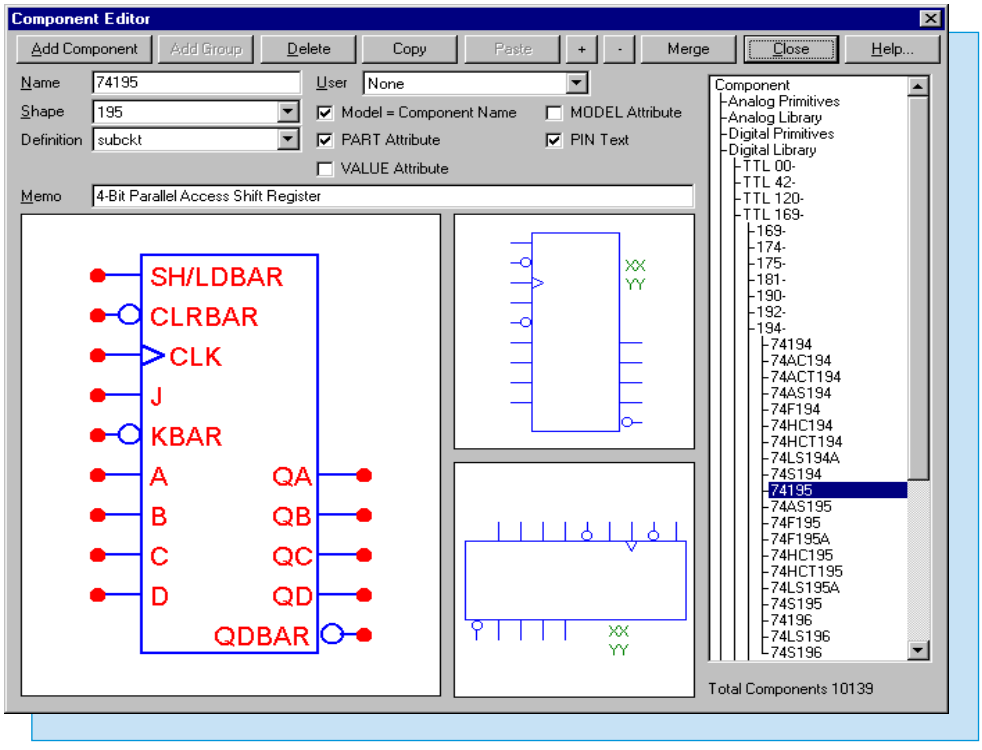


Fig. 3 - Component Editor Settings for the 74195

A sample circuit for the 74195 appears in Figure 4. The resulting analysis appears in Figure 5. The shift, load, and clear features are all demonstrated in this analysis. As can be seen, the 74195 works as specified according to the function table in the TI data book.

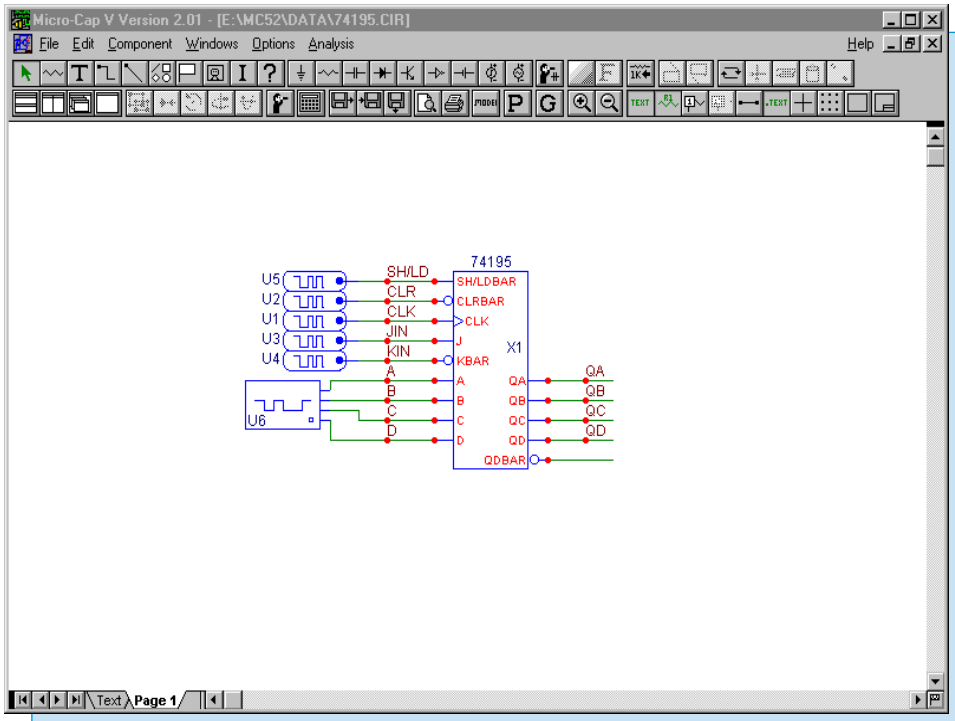


Fig. 4 - 74195 Test Circuit

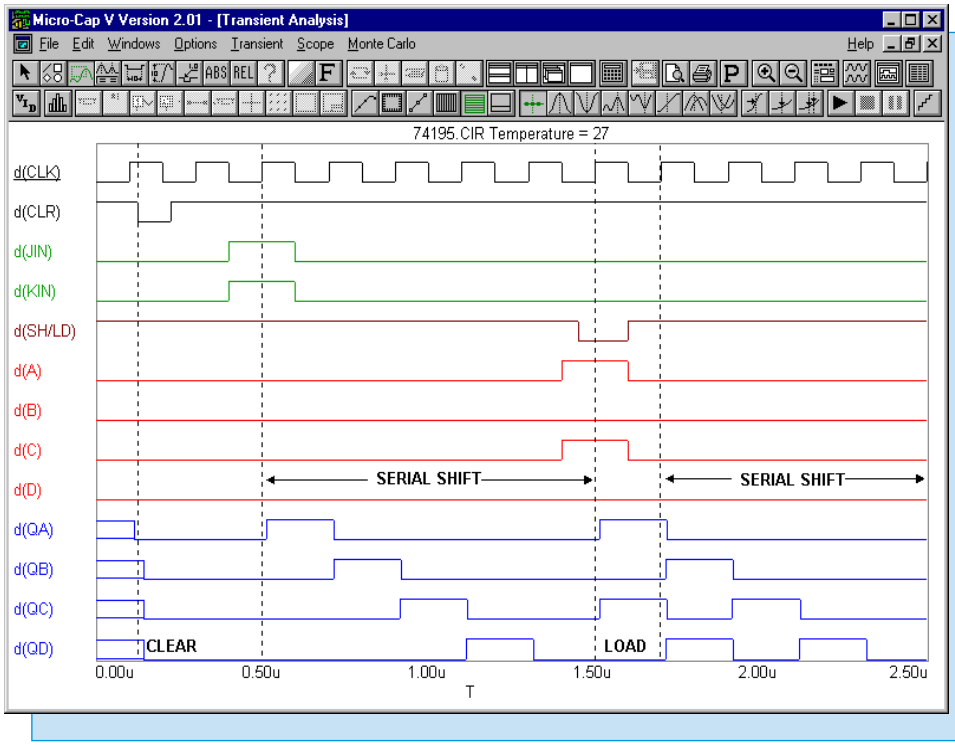


Fig. 5 - 74195 Test Analysis

Wideband Transformer Macro

A wideband transformer is a transformer that is designed to pass a frequency band of several decades. These transformers are usually used to handle complex waveforms rather than simple sinusoidal waveforms or for impedance matching. The macro model for the wideband transformer appears in Figure 6. This model was derived from the article "Spice model simulates broadband transformer" by Michael Steffes which appears in the February 15, 1996 Design Ideas Supplement for EDN.

The macro is a simple circuit consisting of two inductors, two resistors, and a coupling K device. However, data sheets for wideband transformers typically specify four parameters: the impedance ratio, the two -3dB frequencies of the bandpass response, and the primary impedance that the frequencies were measured at. The macro imports these four parameters. RS is the primary impedance, n is the square root of the impedance ratio, fl is the low -3dB frequency, and fh is the high -3dB frequency. Therefore, the two inductances, the coupling coefficient, and the secondary impedance will all need to be calculated from these parameters with the following equations:

$LP = (RS/2)/(2*PI*fl)$
$LS = n^2*LP$
$k = (1/(1+4*(fl/fh)))^{1/2}$
$RL=n^2*RS$

The LP, LS, and RL VALUE attributes all have the appropriate equation above defined directly into the attribute. The K1 COUPLING attribute is defined as K1, and then a .define statement is used to input the equation. Either method will produce the same results. As this is a macro circuit, the desired input and output pins have all been labelled with text. In this case, the nodes have been named as Inp, Inm, Outp, and Outm.

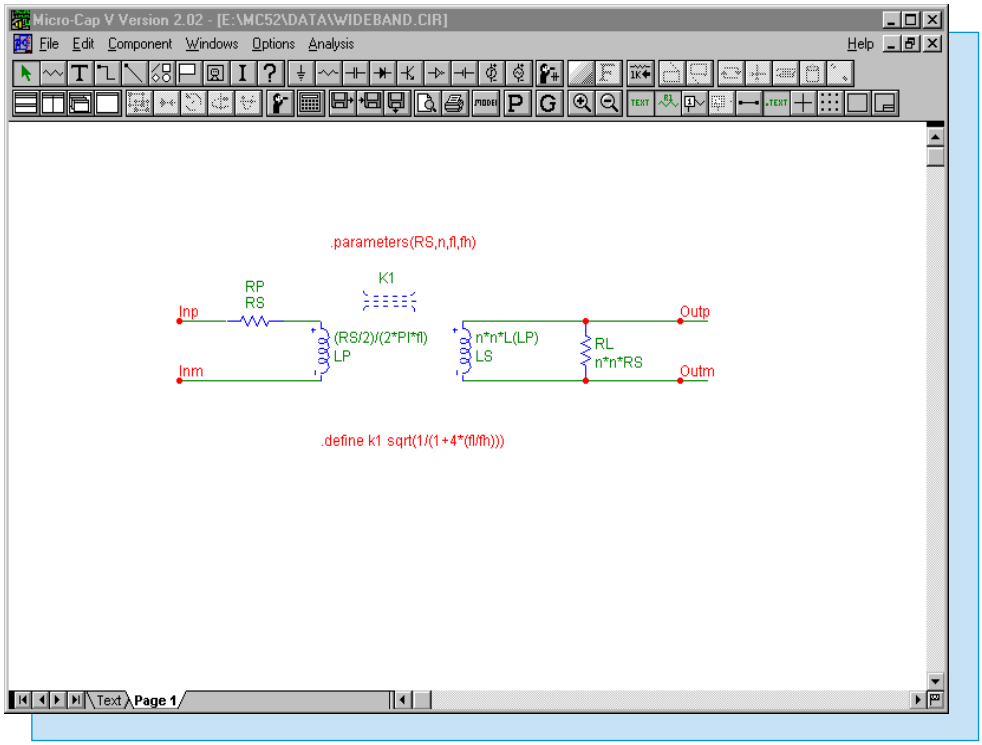


Fig. 6 - Wideband Transformer Macro

The Component Editor settings for the wideband transformer macro appear in Figure 7. The name of the macro is the same as the macro circuit that was created, Wideband. It has been given the transformer shape and macro definition. Four pins have been added to the shape. Note that the pin names match exactly with the node names that were defined in the macro circuit.

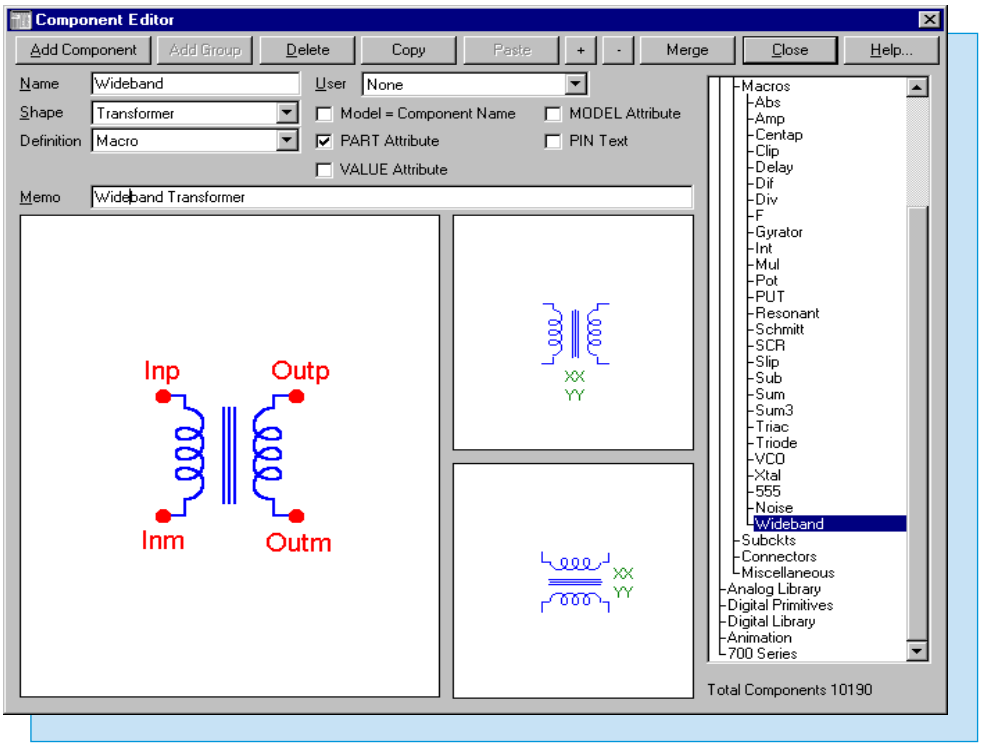


Fig. 7 - Component Editor Settings for the Wideband Macro

The test circuit for the macro appears in Figure 8. This is a simple test circuit designed to analyze the frequency characteristics of a RFTN-16 transformer. The V1 source has its VALUE attribute defined as AC 1 for a 1V AC small signal source. The Wideband macro has its VALUE attribute defined as:

Wideband(50,4,165k,75Meg)

This describes a transformer with a primary impedance of 50 ohms, an impedance ratio of 16 (n^2), a low -3dB frequency at 165KHz, and a high -3dB frequency at 75MHz.

The AC analysis results appear in Figure 9 and display the bandpass response of the transformer. The circuit has been swept from 20KHz to 200MHz, and the output voltage of the macro has been plotted in dB. The two -3dB frequencies have been denoted with cursors, and as can be seen, they are at the points in which the gain is -3dB down from its 6dB peak. The 6dB peak is correct as the midband gain is calculated as:

$Vo/Vi = n*(RL/(RL+n^2*RS)) = n/2$
$Vo/Vi = 2 = 6dB$

This macro is only applicable when the ratio of fh/fl is greater than 100.

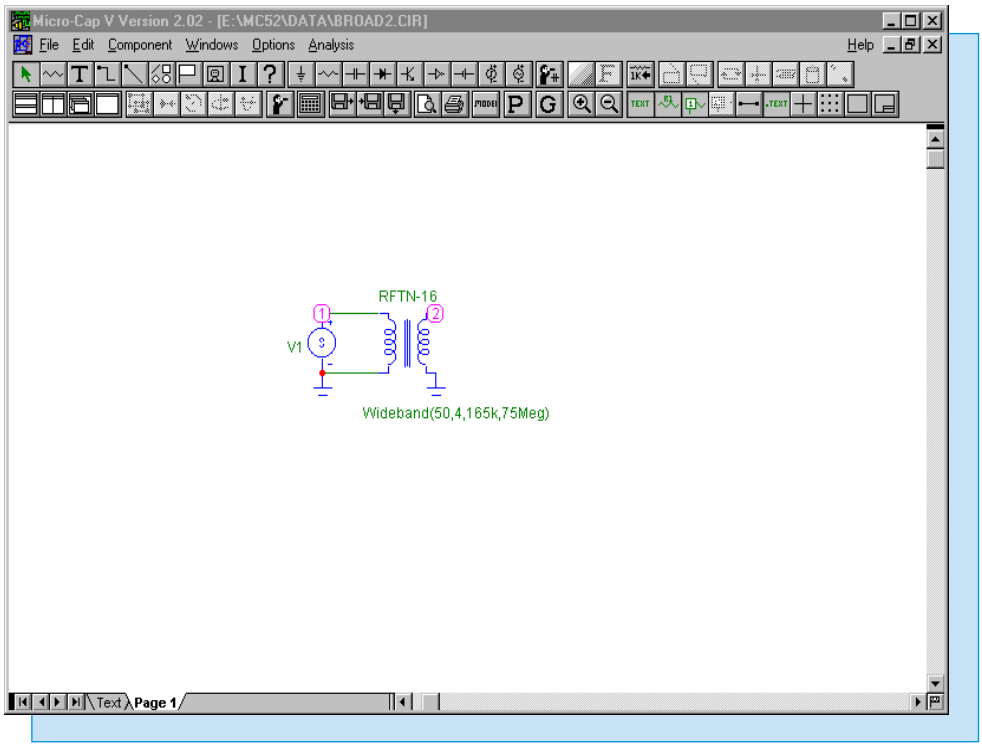


Fig. 8 - Wideband Transformer Test Circuit

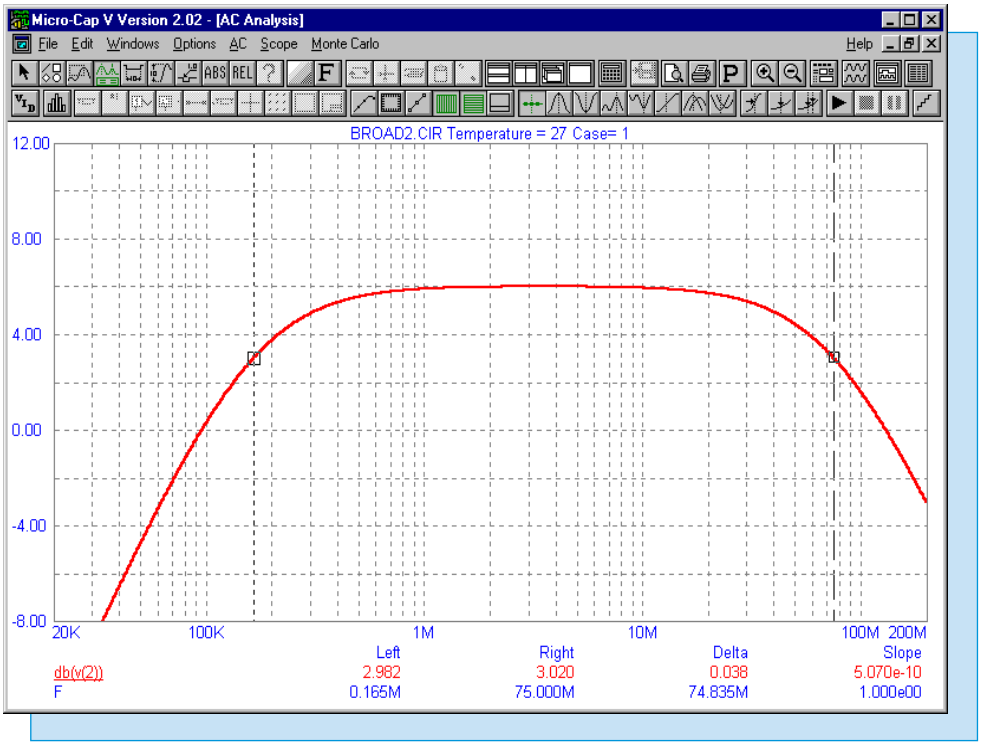


Fig. 9 - Wideband Transformer AC Analysis

Importing Graphical Templates

In designing a circuit, the output response may need to fall within a certain range. For example, a filter may be designed in order to meet certain frequency response specifications. It is possible in Micro-Cap to create graphical templates that let you plot the maximum and minimum characteristics of a waveform to ensure that the output response falls within the correct range. There are two methods for plotting templates in AC analysis: using the Import function or using a Laplace table source.

Importing a Text File

The simplest method is to create a text file that can be used with the Import function to plot the template. The text format that the Import function can handle is the format that is created by the numeric output option in Micro-Cap or the SPICE text output option from another program. The following is an example of this format.

```
Temperature = 27 Case= 1

      F          bppass
   (KHz)
 0.325000      10
 0.420000      38
 1.150000      38
 1.600000      10
```

The above text has been placed in a file called Bpmax.out. The first line indicates the specific run that this data is to be imported with. If this data does not match the type of run then an error will occur. In this instance, the first line specifies that the analysis is run at 27C and that it covers the first run performed (Case=1). The data is then listed in tabular columns. There may be more than two columns of data, but they will all share the same X axis data.

The first column will always be the X axis data. For transient analysis, it must be time. For AC analysis, it must be frequency. For DC analysis, it must be the current or voltage of the source that is being swept. Note that the units for the frequency column are placed on the line below the waveform header, but they may also be included on each line of data instead. If the units are specified in engineering notation, MHz must actually be specified as MegHz.

The second column specifies the Y axis data. In this case, the waveform has been called 'bppass'. The waveform may be given any unique name as long as it has 9 or fewer characters and matching sets of parentheses. There is no need to specify the type of waveform that the data represents such as voltage or current. It will plot according to the data in the column, so the above data could be used with voltage in magnitude or dB, current, phase, or any type of waveform that is plotted. During simulation, the Y axis variable will remain constant for any X axis input that is less than the lowest specified X value or greater than the highest specified X value.

The above file will be plotted as the maximum template for a bandpass filter. The template is designed to make sure that the magnitude of the filter output voltage, when the input is a 1V AC small signal source, does not exceed the limits defined in the template. A second template will also be plotted using the same technique. This template will define a minimum voltage level for the filter. The information for the minimum template is contained in a separate file called Bpmin.out. This file appears as follows:

Temperature = 27 Case= 1

F (KHz)	bppass
0.325000	-5
0.500000	33
1.000000	33
1.505000	-5

The maximum and minimum template waveforms have both been referred to as 'bppass' which is fine because they are contained in different files. In the User Definitions file, which can be accessed through the Options menu, the following two .define statements are entered.

```
.define bpmax import(bpmax.out,bppass)
.define bpmin import(bpmin.out,bppass)
```

These two statements let the templates be imported easily into an AC simulation by just specifying 'bpmax' or 'bpmin' in the Y Expression field of the AC Analysis Limits dialog box. Since the .defines in the User Definitions file are global, any circuit can use the 'bpmin' or bpmax' specifications to plot the template waveforms in AC analysis.

Figure 10 displays the results of a single AC analysis run of the bandpass filter. The V(S3) waveform is the output of the filter, and the bpmax and bpmin waveforms provide the template for the filter analysis.

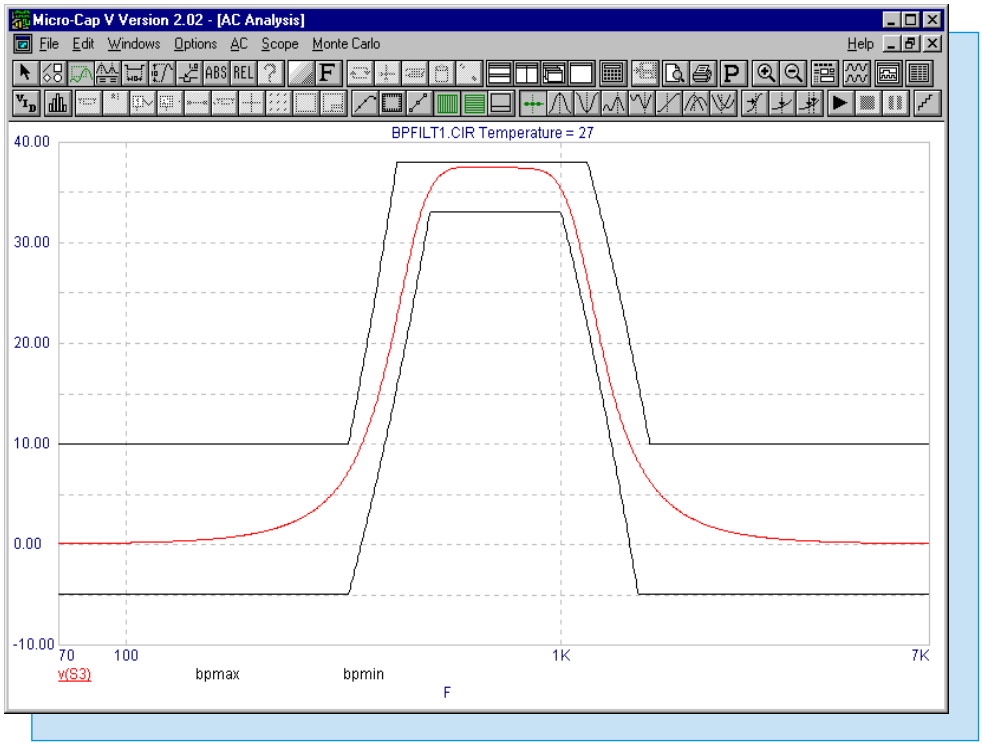


Fig. 10 - Import Bandpass Template Simulation

The Import method is the easiest method to use. No components need to be placed in the schematic, and once the text files are created, the template can be plotted by just specifying the import function or .define variable in the Y expression field. The drawback in this method occurs when multiple runs are simulated as in a Monte Carlo or stepping run. Each run needs to have a separate data listing within the imported file. For example, if a 10 run Monte Carlo analysis is to be simulated, the Bpmax.out file would have to contain ten copies of the data. The only difference in the data would be that the Case value would have to be incremented for each copy. A simulation involving stepping would also need to have as many copies of the data as there are runs. Instead of the Case value, the name and value of the component being stepped would need to be specified in the header. For this reason, using Laplace table sources to import AC templates is a simpler method in the case of multiple run simulations.

Using a Laplace Table Source

The Laplace table sources can handle triplets of data in the form of (frequency,magnitude,phase). The sources will interpolate between specified data points and will be constant for any frequencies that are below the lowest frequency and above the highest frequency. The Laplace table sources would have to be added to a schematic in the configuration that appears in Figure 11 in order to produce the two template waveforms from the first method.

This configuration consists of two Laplace table sources and a SPICE independent source. The number of Laplace table sources is dependent on the number of template waveforms being plotted on a 1:1 basis. The SPICE independent source can be found in the Waveform Sources section of the Analog Primitives and is called 'V'. This source has its VALUE attribute defined as 'AC 1' to supply a 1 volt small signal waveform to the Laplace sources. Either a SPICE independent source, a sine source, a pulse source, or a user source must be present at the input of the table sources to provide the correct AC stimulus. Otherwise, the template plot will be incorrect.

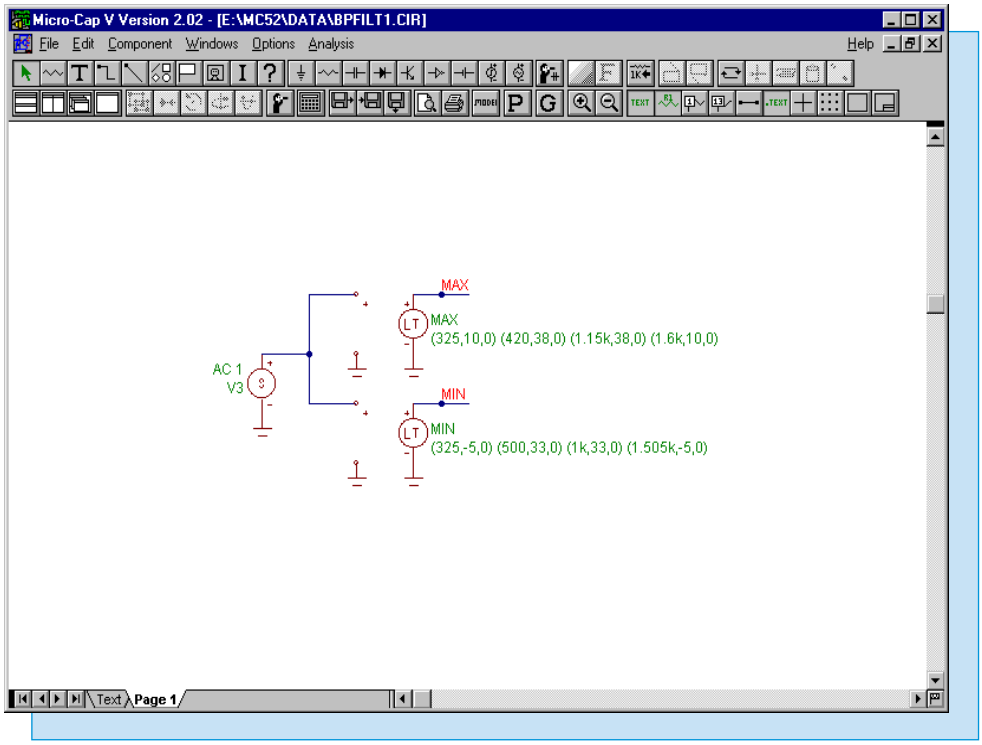


Fig. 11 - Laplace Sources Configuration

The FREQ attributes in the table sources have been defined with the same data that was contained in the Bpmax.out and Bpmin.out files. The phase data has been specified as 0 for each data point as it is not being used in this case. The FREQ attribute for the Max source (maximum template) is defined as:

(325,10,0) (420,38,0) (1.15k,38,0) (1.6k,10,0)

The FREQ attribute for the Min source (minimum template) is defined as:

(325,-5,0) (500,33,0) (1k,33,0) (1.505k,-5,0)

The KEYWORD attribute for both sources has been specified as MAG to indicate that the magnitude value is expressed in true magnitude rather than in dB. The table sources will produce a voltage output that is equivalent to the desired template. Figure 12 displays the AC analysis results of a 10 run Monte Carlo analysis. The waveforms V(Max) and V(Min) are the template waveforms calculated from the table sources.

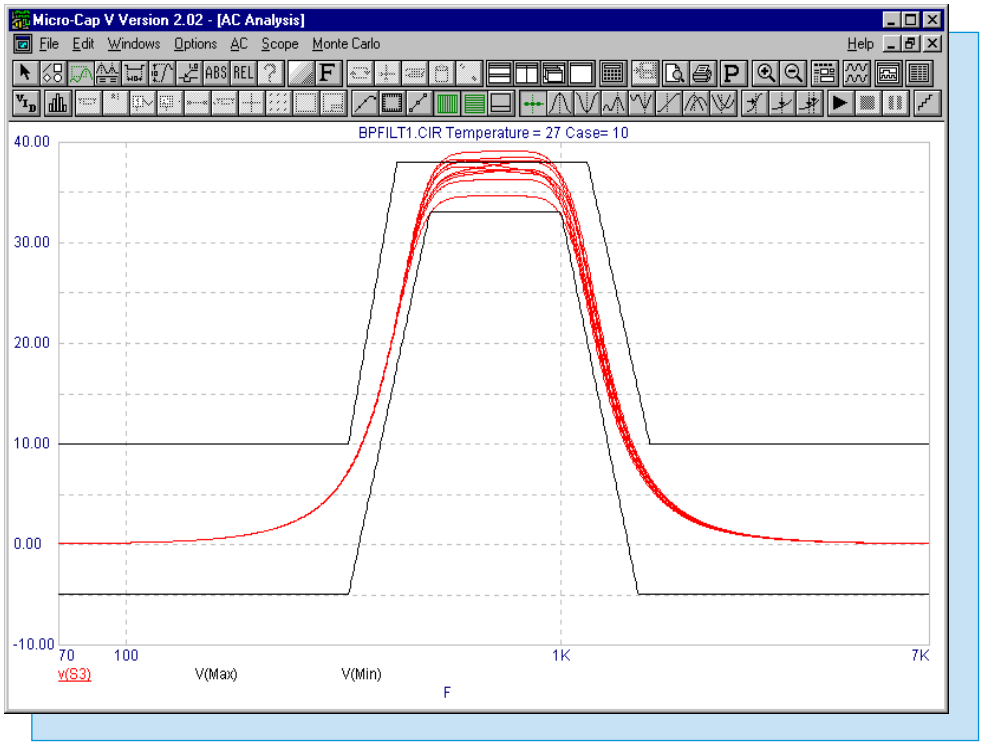


Fig. 12 - Laplace Table Source Template Simulation

Sweeping a Resistor in DC Analysis

The premise of DC analysis is to sweep either a voltage or current source to view the DC transfer functions of a schematic. However, in some types of circuits, the component that may need to be swept for meaningful results is a resistor. Sweeping a resistor can be accomplished by linking its value to a voltage source and then sweeping the voltage source in DC analysis.

The circuit in Figure 13 is a basic current mirror configuration. The components V1, R1, and Q1 define a reference current. An equivalent current is subsequently produced in the collector of Q2. There are two things to note in this circuit. The first is the battery floating to the right of the circuit. This battery has been given the PART attribute of Vsweep and is the source that will be swept in DC analysis. Since it is floating, it will have no direct effect on the current mirror operation. The second thing to note is the definition of the R1 resistor. The VALUE attribute for the resistor has been defined as V(Vsweep). Therefore, the resistance of R1 will be swept with the same values that are defined for Vsweep in the DC analysis limits.

In the DC analysis limits dialog box, Input 1 is defined as Vsweep and Input 1 Range is defined as 60K,10K,5. Two waveforms will be plotted. The Y expressions for these waveforms are I(R1) and I(R2). The X expressions have both been defined as R(R1). These settings will sweep the Vsweep battery from 10K to 60K which will in turn sweep the R1 resistor with the same values. The plot produced will compare the reference current with the output current to confirm that the output current tracks the reference current throughout the resistance range. Figure 14 displays the results of the DC analysis.

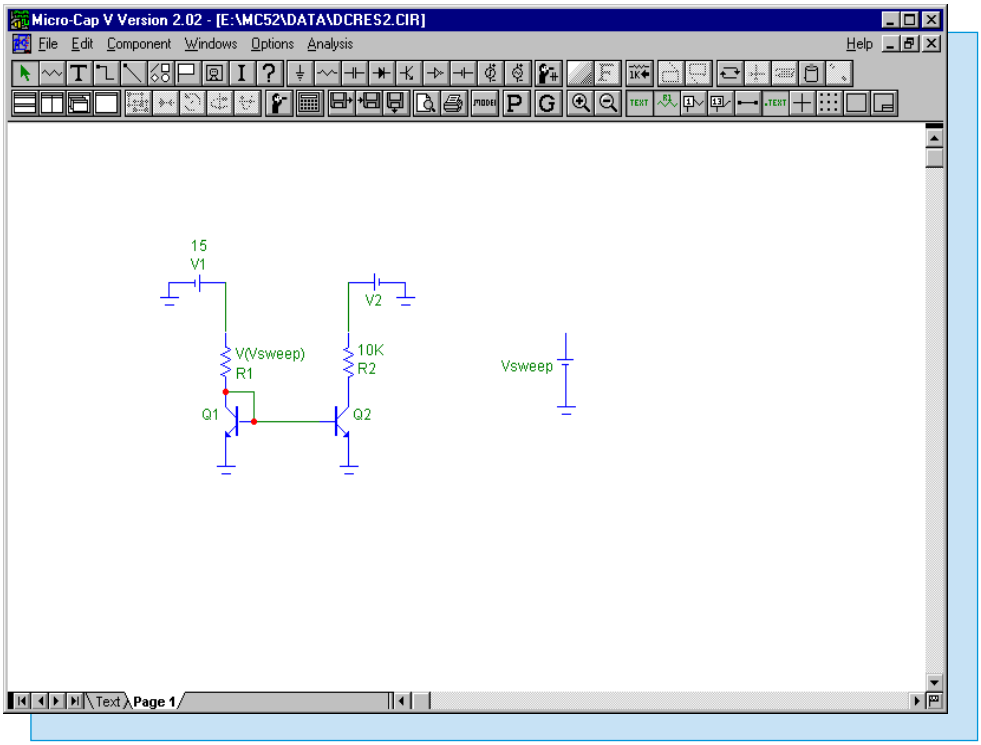


Fig. 13 - Current Mirror Circuit

As can be seen in Figure 14, the output current closely tracks the reference current for the specified values of the R1 resistor. The slight difference occurs due to the effect of the finite beta of the transistors and the effect that the output voltage has on the output current.

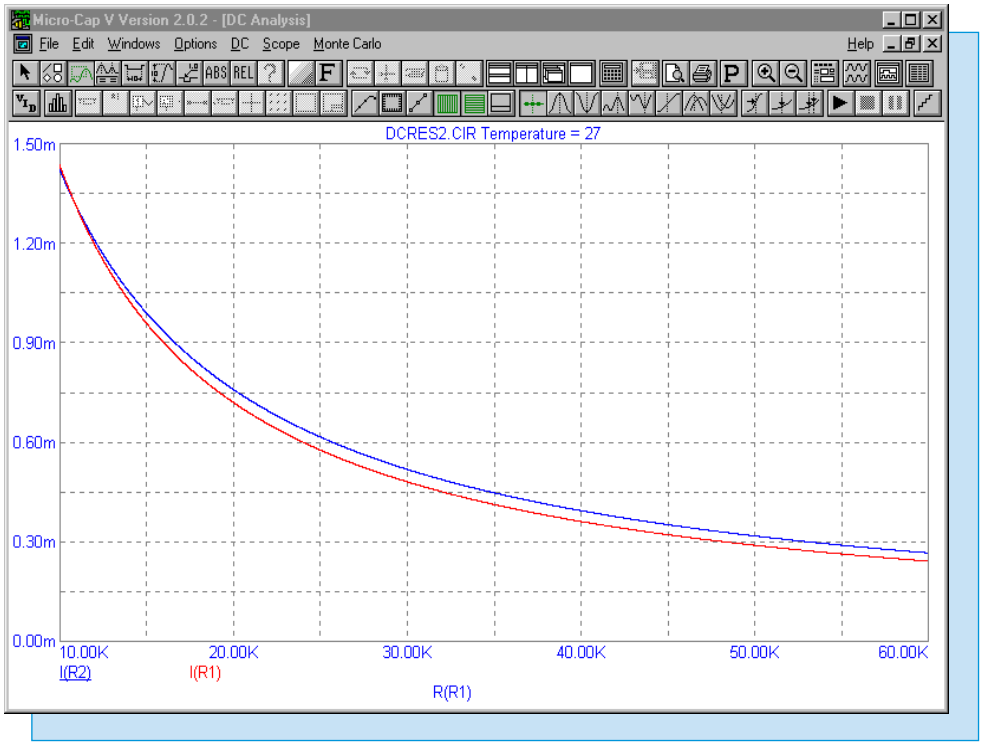


Fig. 14 - DC Analysis of Resistor Sweep

Product Sheet

Latest Version numbers

Micro-Cap V ..... Version 2.02
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.