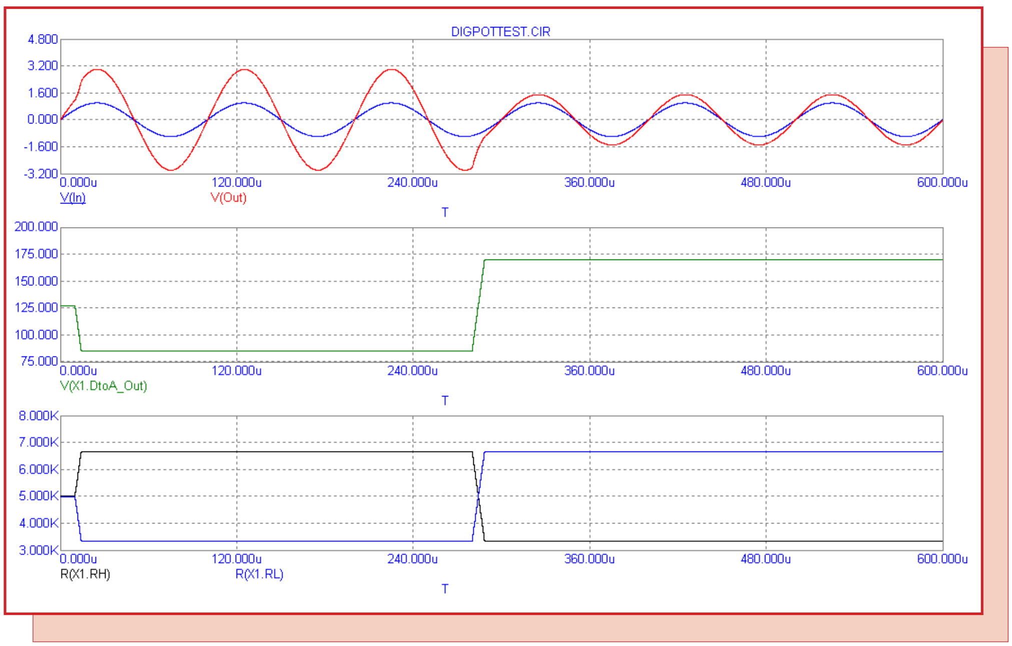
spectrum news

Applications for Micro-Cap™ Users

Fall 2002

News



Digital Potentiometer Macro

Featuring:

- Digital Potentiometer Macro
- Converting an Excel File to a User Source File
- Creating Eye Diagrams

News In Preview

This newsletter's Q and A section describes the mathematics of the VCO macro and why the need for an integrator exists. The Easily Overlooked Features section describes a couple of methods that can be used to change the values of multiple components at once.

The first article describes a model for a digital potentiometer.

The second article describes the process of converting an Excel file into a User Source file that can then be imported and ran in a simulation.

The third article describes a method for creating eye patterns in transient analysis.

Contents

News In Preview 2
Book Recommendations 3
Micro-Cap Questions and Answers 4
Easily Overlooked Features 5
Digital Potentiometer Macro 6
Converting an Excel File to a User Source File 11
Creating Eye Diagrams 14
Product Sheet 16

Book Recommendations

General 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

- *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

MOSFET Modeling

- *MOSFET Models for SPICE Simulation, William Liu, Including BSIM3v3 and BSIM4*, Wiley-Interscience, First Edition, ISBN# 0-471-39697-4

VLSI Design

- *Introduction to VLSI Circuits and Systems,* John P. Uyemura, John Wiley & Sons Inc, First Edition, 2002 ISBN# 0-471-12704-3

Micro-Cap - German

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

Micro-Cap - Finnish

- *Elektroniikkasimulaattori,* Timo Haiko, Werner Soderstrom Osakeyhtio, 2002. ISBN# ISBN 951-0-25672-2

Design

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

- *Microelectronic Circuits.,* Adel Sedra, Kenneth Smith, Fourth Edition, Oxford, 1998

High Power Electronics

- *Power Electronics,* Mohan, Undeland, Robbins, Second Edition, 1995. ISBN# 0-471-58408-8

- *Modern Power Electronics,* Trzynadlowski, 1998. ISBN# 0-471-15303-6

Switched-Mode Power Supply Simulation

- *SMPS Simulation with SPICE 3,* Steven M. Sandler, McGraw Hill, First Edition, 1997. ISBN# 0-07-913227-8

- *Switch-Mode Power Supply SPICE Simulation Cookbook*, Christophe Basso, McGraw-Hill 2001. This book describes many of the SMPS models supplied with Micro-Cap.

Micro-Cap Questions and Answers

Question: I'm trying to create a VCO using an NFV source. I've defined the NFV source as:

sin(2*PI*FSweep*t)

and I then use a .define statement to define the FSweep variable as:

.define FSweep (1Meg * t + 1000)

When I run transient analysis for .01s, the frequency of the sine source seems to be erratic and high when it should steadily increase from 1kHz to 11kHz, but when I set FSweep to a constant number, it works as I expect. How do I get this to work correctly?

Answer: The VCO you created was actually our first draft for a VCO macro also and with the results we experienced we had to really look into the mathematics of it. The reason for the discrepancy is that the frequency of a sine wave is actually the derivative of the sine's argument (hence the integrator we have in the VCO macro) so if the expression is as follows:

sin(arg) = sin(2*PI*v*t)
f=d(arg)/dt = 2*PI*(v+t*dv/dt)

where v is also a time based equation.

f=2*PI*(v+t*dv/dt) is the actual frequency being produced but
f=2*PI*v is the theoretical frequency we want.

That is why, when the input is a constant, it will work fine since t*dv/dt goes to 0.

The best method for solving this is to use the VCO macro that comes with Micro-Cap. If you are intent on creating your own VCO, the derivative factor must be taken into account to produce a working model.

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 a toolbar button.

Changing Multiple Component Values

Changing the values of multiple components at once has obvious benefits over clicking on and editing each of the components individually. There are two main methods for performing this operation.

.Define Statement

When placing a component in the schematic, you can define the appropriate attribute with a symbolic name and then use a .define statement to actually define the value. Multiple components would be defined with the same .define statement. For example, you may want to define the Model attribute of multiple resistors with the symbolic name TOLS. In the text area of the schematic, you would have statements such as:

.define TOLS Resmod5
.model Resmod5 Res (R=1 Lot=5%)
.model Resmod10 Res (R=1 Lot=10%)

To easily change the model of all of the resistors that use the define statement you would just need to change the .define statement so that TOLS references the Resmod10 model. The drawback of this is that if you change the .define statement you change the value of all components that reference it so you can't pick and choose which components to change.

Change button in the Attribute dialog box

To pick and choose the components you want to edit, use the Change button that is in the upper right hand corner of the Attribute dialog box. The Change button invokes the Change Attribute To dialog box which provides a list of all of the components that are of the same type that reside in the schematic. Select the components from this list, and the edit for that specific attribute will be applied to all of them. For example, if you want to change the model of multiple resistors, you would double click on one of the resistors to be changed. Edit the model attribute for that resistor and then hit the Change button. A list of all of the resistors in the schematic will appear. Choose the other resistors that you want the new model to be applied to and hit OK. The model for all of the selected resistors will be set to the new value.

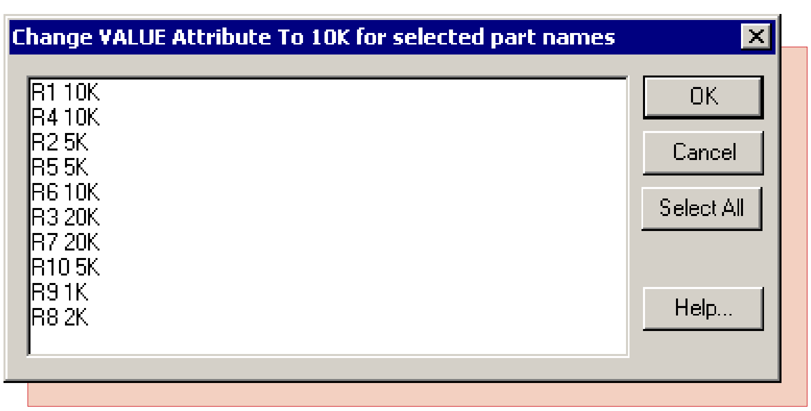


Fig. 1- Change Attribute To dialog box

Digital Potentiometer Macro

A digital potentiometer performs the same function as a mechanical potentiometer or a variable resistor. Instead of a mechanical wiper, the wiper position in a digital potentiometer is determined by the events on logical inputs. Digital potentiometers have a few advantages over their electromechanical counterparts. They can be used to handle a wide variety of specialty tasks that are not possible with the standard potentiometer. Among these are remote process control, remote calibration, analog to digital conversions, digital to analog conversions, variable gain amplification, and variable oscillation.

The digital potentiometer macro in this article was loosely based on the MAX5450-5455 series of devices from Maxim. These devices are digital potentiometers with an up/down interface consisting of two fixed resistors with 256 digitally controlled wiper contacts (taps). There are three digital inputs that determine the tap point that will be accessible to the wiper. These digital inputs are INC, CS, and U/D. The INC input is the wiper increment control input. A high to low transition on this input increments or decrements the wiper position depending on the states of CS and U/D. The CS input is the chip select input. When this input is low, the wiper position may be changed through INC and U/D. The U/D input is the up/down control input. When this input is high, a high-to-low transition on INC increments the wiper position, and when the input is low, a high-to-low transition on INC decrements the wiper position. If the wiper is at either the high or low end of the resistor array, any increment or decrement in the direction of the endpoint will not change the wiper position, so in other words, there is no wrap around effect. Finally, upon power-up the initial setting for the wiper is at midscale (tap 127).

Macro Circuit

The basic modeling of the digital potentiometer macro consists of a 256 bit counter whose outputs are sent to a DtoA converter. The output of the DtoA converter is then used to calculate the value of a pair of resistors.

The 256 bit counter keeps track of the tap position the wiper is at. The counter used in the macro appears in Figure 2. The heart of the counter consists of 8 J-K flip-flops that store the tap position on their non-inverting outputs. The INC input is fed directly into the clock inputs on each of the flip-flops. The clear and preset pins of each flip-flop are connected to either the Hi node or the Set node. The Hi node is at a constant one state that is produced by a Fixed Digital component. The Set node waveform is defined by a digital stimulus which produces a short 10ns zero state that then returns to the one state for the rest of the simulation. This waveform initializes the flip-flops. The U1 through U7 flip-flops are preset and the U8 flip-flop is cleared at the beginning of a simulation. This sets the initial counter value to 127 to simulate the wiper being at midscale upon power-up.

The counter will be disabled under three instances. The first one is when the CSBAR input is set high. The second one occurs when the U/DBAR input is set high and the non-inverting outputs of the flip-flops are all high. This only occurs when the counter is trying to increment and the output of the counter is already at 255 and thus prevents it from rolling over to 0. The third instance occurs when the U/DBAR is set low and the inverting outputs of the flip-flops are all high. This occurs when the counter output is at 0, and the counter is trying to decrement. It prevents the rollover from 0 to 255.

The J-K inputs of each flip-flop are tied together. The logic that feeds into the J-K inputs controls whether the flip-flop will toggle or remain in its current state at the next negative transition of the clock. When the counter is incrementing, if all of the less significant non-inverting outputs are high then the flip-flop will be set to toggle upon the next clock transition. When the counter is decrementing, if all

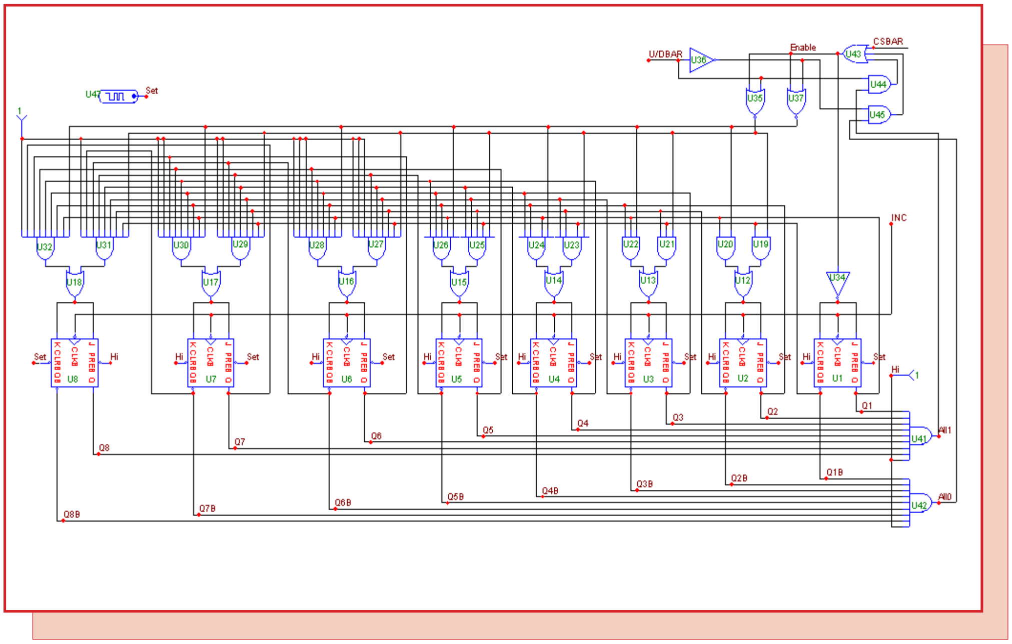


Fig. 2 - 256 Bit Digital Counter

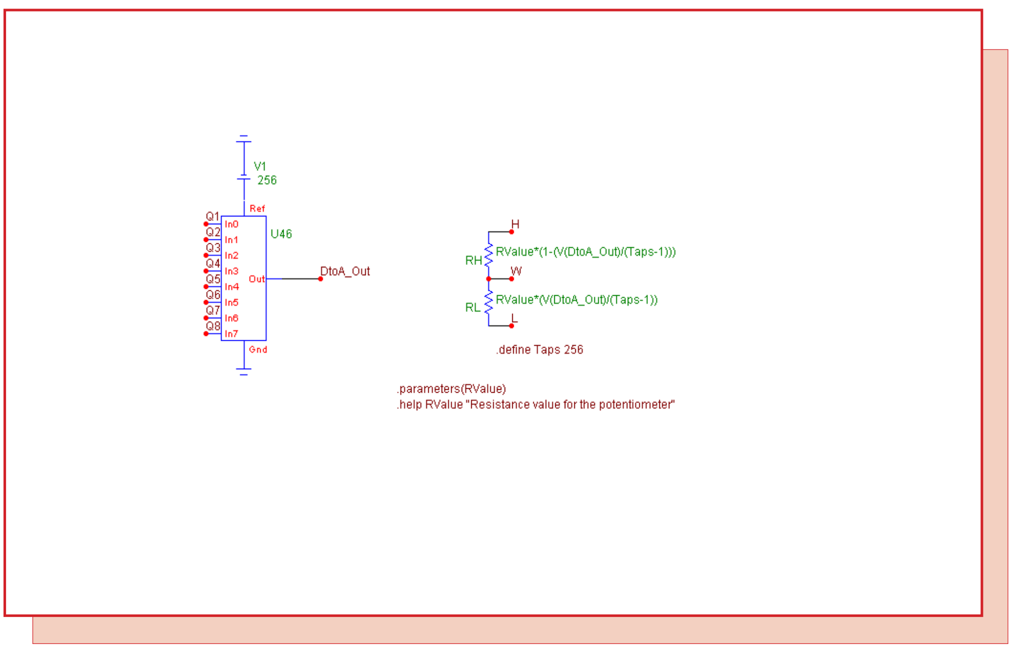


Fig. 3 - DtoA and Potentiometer Resistors

of the less significant inverting outputs are high then the flip-flop will be set to toggle upon the next clock transition. Otherwise, the flip-flop will stay in its current state. In both cases, the counter must be enabled for the flip-flop to toggle.

The non-inverting outputs of the flip-flops are fed into a DtoA component. The purpose of the DtoA component is to convert the digital value of the counter into an analog voltage that the resistors can reference to adjust their resistance values. The Reference node for the DtoA is a 256 volt battery. The 256 volt reference causes the analog output to be equivalent to the value of the actual tap position that is being assigned. For example, if the counter is pointing to the 87th tap position, the voltage at the output of the node will be 87V. The advantage in this method is that it is easy to view which tap position is being used when running an analysis. Simply plot the voltage at the node DtoA_Out within the macro.

The two resistors of the potentiometer then use the output voltage of the DtoA component in determining their own resistance values. When the counter is incremented, the resistance value of the RL resistor is increased, and subsequently, the RH resistor value is decreased. The VALUE attribute of the RL resistor is defined as:

RValue*(V(DtoA_Out)/(Taps-1))

and the VALUE attribute of the RH resistor is defined as:

RValue*(1-(V(DtoA_Out)/(Taps-1)))

RValue is the macro parameter that is passed through to the circuit. It represents the resistance value for the potentiometer. The variable Taps is the number of taps in the potentiometer and is assigned the value 256 through a .define statement. The three outputs of the potentiometer are the outputs H, W, and L. These correspond to the high terminal, wiper terminal, and low terminal of the potentiometer. The DtoA and resistor portion of the macro can be seen in Figure 3.

Example Circuit

The example circuit for the digital potentiometer macro appears in Figure 4. The opamp, a MAX402, is set up in a basic non-inverting gain configuration. The opamp is powered by 5V and -5V power supplies and has a 1V, 10kHz sine wave going into the non-inverting input. The presence of the digital potentiometer in the feedback provides a variable gain amplification for the circuit. The RValue parameter for the macro is set at 10k. Three digital stimuli control the wiper position of the potentiometer. The .define statements for each of the stimuli are as follows:

```
.define INC
+label=start
+0ns 0
+50ns 1
+100ns goto start -1 times

.define CS
+0 1
+10u 0
+14.2u 1
+280u 0
+288.5u 1
```

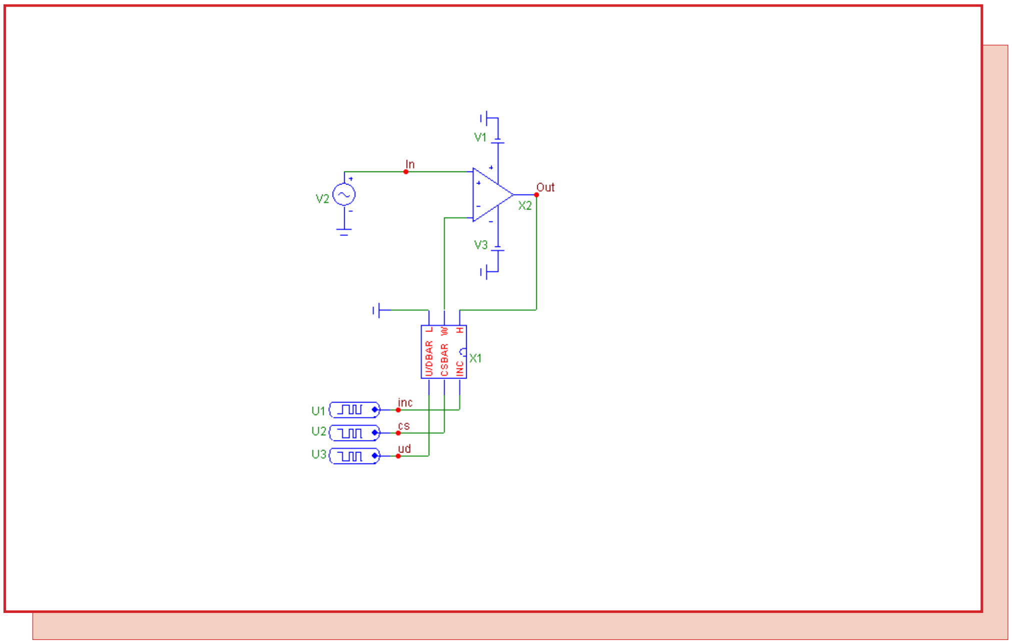


Fig. 4 - Digital Potentiometer Example Circuit

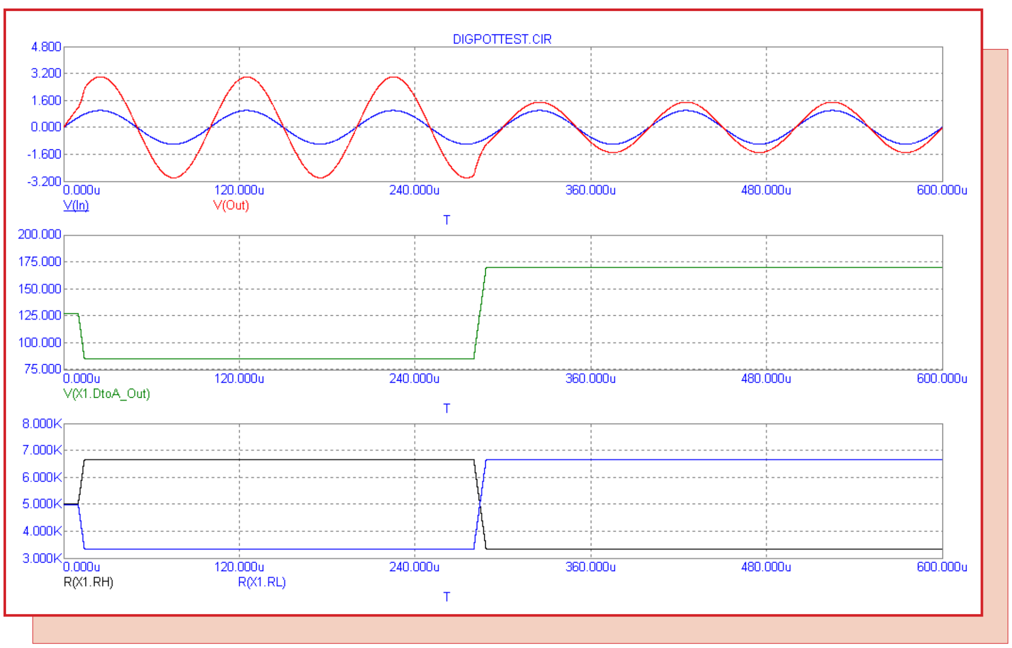


Fig. 5 - Digital Potentiometer Example Analysis

```
.define UD
+0 0
+14.2u 1
+48u 1
```

The transient analysis results for the circuit appear in Figure 5. The top plot group displays the voltages at nodes In and Out. The middle plot displays the voltage at the output of the DtoA component inside the potentiometer macro, and the bottom plot displays the values of the two potentiometer resistors. The simulation starts with the potentiometer at midscale (tap 127). From 10us to 14.2us, the wiper position is decremented until it reaches tap 85. At this point, the RH resistor is at 6.667kOhms and the RL resistor is at 3.333kOhms, and the gain of the opamp is set at 3. From 280us to 288.5us, the wiper position is incremented until it reaches tap 170. The resistance of RH is now 3.333kOhms, the resistance of RL is 6.667kOhms, and the gain of the opamp is set at 1.5.

Modifying the Macro Circuit

Not all digital potentiometers will use 256 taps for the wiper. If a potentiometer uses less taps, this macro can be easily modified to simulate that device with three steps.

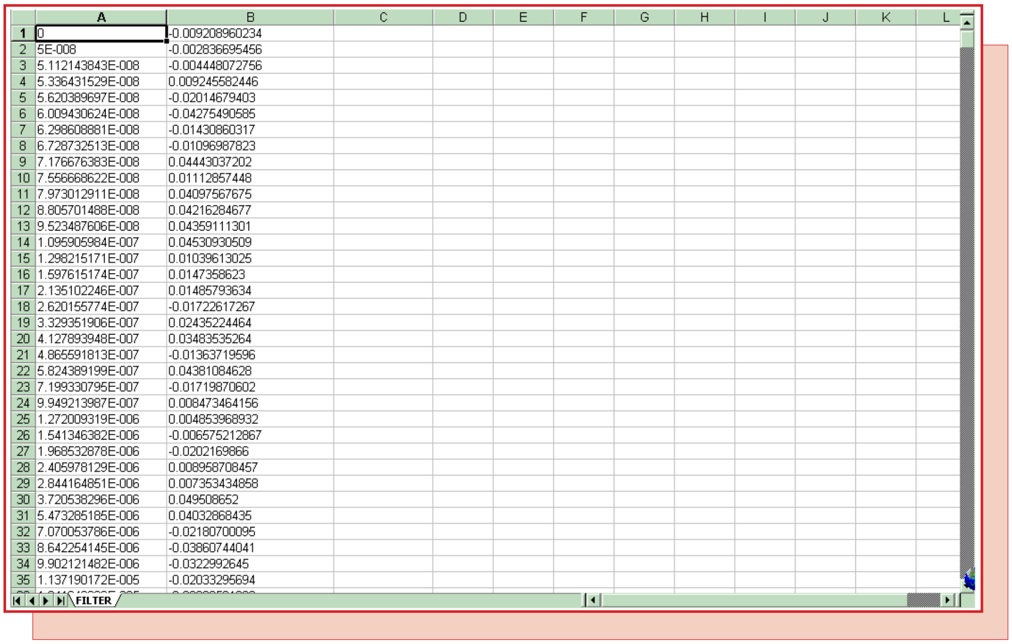
1) Change the inputs to the And gate in the counter whose output is at node All1. This And gate determines the maximum value that the counter will go to using the outputs of the flip-flops. For example, if you want to simulate 128 taps, you would replace the Q8 input with Q8B. The counter would then go no higher than 127.

2) Swap the necessary Hi and Set texts on the prebar and clear nodes of the flip-flops to create the initial power-up value. For example, if you want the 128 bit counter to power-up at midscale, you would need to place the Set text on the clear pin and the Hi text on the preset pin of the U7 flip-flop. The power-up wiper position would then start at tap 63.

3) Change the .define Taps statement so that Taps is set to the number of taps that will be used by the device. For the 128 tap potentiometer, the statement should read '.define Taps 128'.

Converting an Excel File to a User Source File

Some third party engineering software programs have the capability to output waveform data points into an Excel Workbook format. In some cases, it can be very useful to use this waveform information to define an input source in a Micro-Cap schematic. This article explains the process of how to convert a voltage waveform that is contained in an Excel Workbook file into the file format that can be imported by a User Source in a Micro-Cap schematic file for use in transient analysis.



	A	B
1	0	-0.009208960234
2	5E-008	-0.002836695456
3	5.112143843E-008	-0.004448072756
4	5.336431529E-008	0.009245582446
5	5.620389697E-008	-0.02014679403
6	6.009430624E-008	-0.04275490585
7	6.298608881E-008	-0.01430860317
8	6.728732513E-008	-0.01096987823
9	7.176676383E-008	0.04443037202
10	7.556668622E-008	0.01112857448
11	7.973012911E-008	0.04097567675
12	8.805701488E-008	0.04216284677
13	9.523487606E-008	0.04359111301
14	1.095905984E-007	0.04530930509
15	1.298215171E-007	0.01039613025
16	1.597615174E-007	0.0147358623
17	2.135102246E-007	0.01485793634
18	2.620155774E-007	-0.01722617267
19	3.329351906E-007	0.02435224464
20	4.127893948E-007	0.03483535264
21	4.865591813E-007	-0.01363719596
22	5.824389199E-007	0.04381084628
23	7.199330795E-007	-0.01719870602
24	9.949213987E-007	0.008473464156
25	1.272009319E-006	0.004853968932
26	1.541346382E-006	-0.006575212867
27	1.968532878E-006	-0.0202169866
28	2.405978129E-006	0.008958708457
29	2.844164851E-006	0.007353434858
30	3.720538296E-006	0.049508652
31	5.473285185E-006	0.04032868435
32	7.070053786E-006	-0.02180700095
33	8.642254145E-006	-0.03860744041
34	9.902121482E-006	-0.0322992645
35	1.137190172E-005	-0.02033295694

Fig. 6 - Excel Workbook Example File

Figure 6 displays the data points of a voltage waveform within an Excel Workbook file. The first column defines the time in seconds of each data point. When defining a User Source for transient analysis, it is mandatory that the first column be time. The second column defines the value of the voltage at each time point. Neither of these columns can be scaled. For example, the voltage column must be in volts rather than millivolts or kilovolts, and the time column must be in seconds. The only information that should be in both columns are the actual data points. Any headers at the top of the column should be deleted.

The first step is to save the Excel file as a comma delimited text file. This is easily done by clicking on File and Save As. In the Save As Type list box, there is a CSV (comma delimited) option. Select this option and click Save. This converts the Excel Workbook format into a simple ASCII text file that has the column data comma delimited. The file extension for the new file will be .CSV. It needs to be changed to .USR. This change can be made in many ways such as through Wordpad or Windows Explorer. The .USR file should then be placed in the directory that your Data path refers to.

The second step is to add the header information into the .USR file. The header information tells Micro-Cap what type of information is in the file and the name of the waveform so that it can be processed when an analysis is run. The header will appear as:

[Main]
FileType=USR
Version=2.00
Program=Micro-Cap

[Menu]
WaveformMenu=V(In) vs T

[Waveform]
Label=V(In) vs T
MainX=T
LabelX=T
LabelY=V(In)
Format=SimpleNoX
Data Point Count=1041

The [Main] section defines the format version that is being used. This section should appear exactly as it is shown here for all User Source files.

The [Menu] section defines the names of the waveforms that are present in the file. Since there is only one waveform in this file, there is only one WaveformMenu tag defined. In this case, the waveform is called 'V(In) vs T'. Micro-Cap uses these tags to display the names of the waveforms in the file in both the Attribute dialog box and in the Analysis Limits dialog box.

Each waveform in the file must have a [Waveform] section defined for it. This section defines the names of the data columns and declares the format that the data is expressed in. The Label tag must match one of the WaveformMenu tags in the [Menu] section. This tag contains the string that is to be defined in the Expression attribute of the User Source when the source is placed in a schematic. The Format tag defines the format that the data is in. In this case, SimpleNoX states that there will be only two columns of data with the left column being time and the right column voltage (T,Y). Since the format is SimpleNoX, both the MainX and LabelX tags will be defined as T for time, and the LabelY tag is defined as V(In) for this waveform. The Data Point Count tag declares the number of data points the waveform consists of. This number can be easily determined in Excel by the row number the file ends on. The actual data points for the waveform should immediately follow the [Waveform] section.

An example circuit that uses this new User Source file is displayed in Figure 7. The user file that was created for this example, FALL2002.USR, simulates a pulse with an injected noise source on top. The User Source, U1, that references this file is fed into an active Chebyshev filter. The U1 User Source has its FILE attribute defined as 'FALL2002.USR', and its EXPRESSION attribute defined as 'V(In) vs T'.

The resulting transient analysis of this circuit appears in Figure 8. V(In) is the User Source input, and V(Out) is the output of the filter. As can be seen in the analysis, the User Source is particularly adapted to importing waveforms that can't be defined easily with standard sources or through equations.

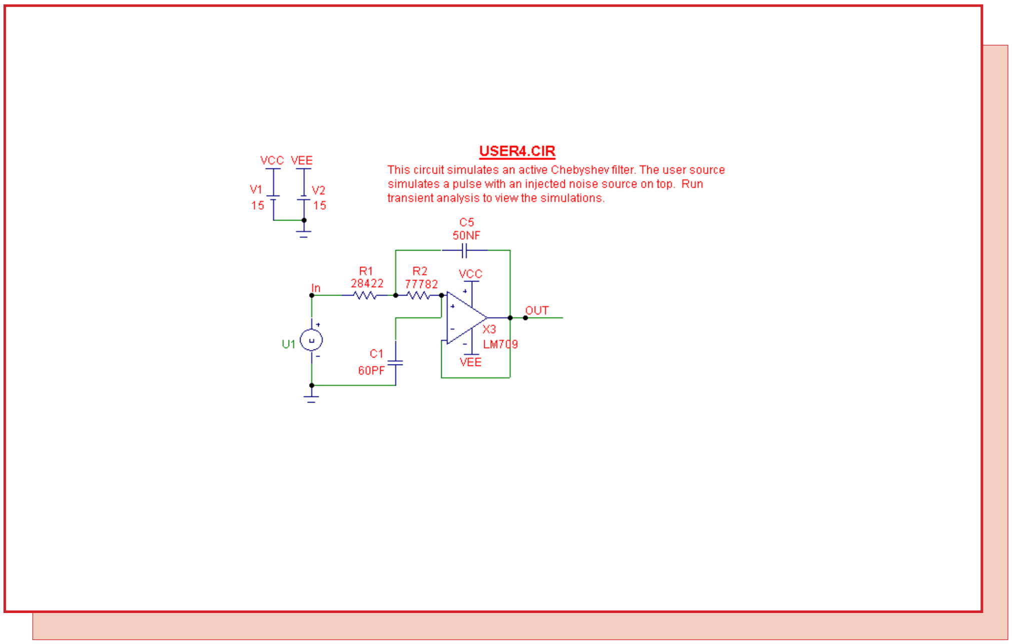


Fig. 7 - User Source Example Circuit

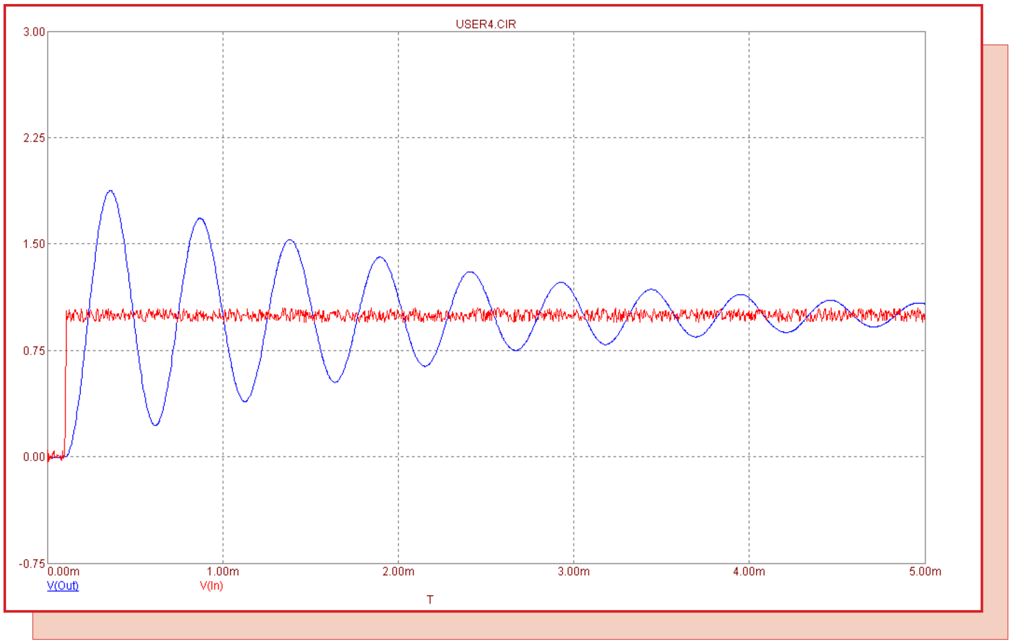


Fig. 8 - User Source Example Analysis

Creating Eye Diagrams

The purpose of an eye diagram is to analyze the transition time deviations of a digital communications signal. The deviations, also known as jitter, are a measure of the signal quality and represent the variance in the actual transition time from the ideal transition time. In addition to jitter, an eye diagram also produces information on the voltage swing, the rise time, and the fall time of the signal. Jitter is apparent when a repetitive waveform is displayed versus a reference waveform. To create an eye diagram with an oscilloscope, one uses the horizontal sawtooth oscillator to display the waveform versus the sawtooth. This sawtooth is achieved in Micro-Cap through the use of the Mod function.

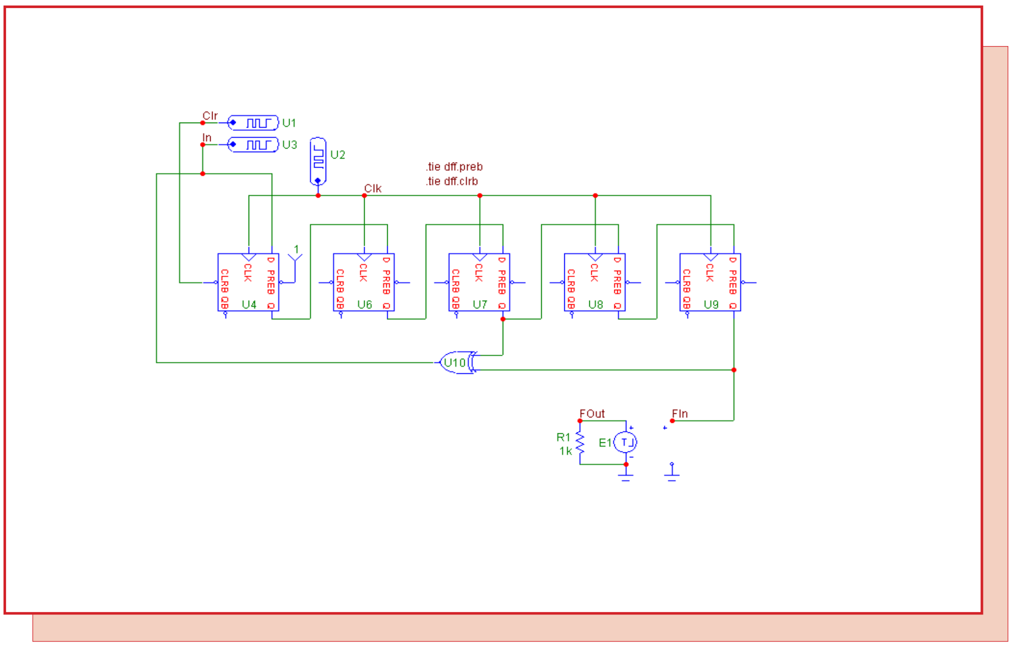


Fig. 9 - Eye Diagram Example Circuit

The circuit in Figure 9 creates a bit stream that can be analyzed through an eye diagram. The bit stream is produced by a basic five bit shift register, consisting of five D flip-flops, that has the outputs of the third and fifth flip-flop as inputs to an exclusive-or gate. The output of the exclusive-or gate is then fed back into the D input of the first flip-flop. Note that the PREB and CLRB pins for all of the flip-flops are connected together using the .TIE statement such as:

.TIE DFF.PREB
.TIE DFF.CLRB

The first statement connects the PREB pin of all D flip-flops in the circuit to one node, and the second statement does the same for the CLRB pin of all D flip-flops. Three digital stimulus sources control the input to the register. The U1 digital stimulus clears the flip-flops with a short zero pulse at the beginning of the simulation. The U2 stimulus creates a clock signal with a 5ns period. The U3 stimulus ensures that a one state is present at the In node during the first clock transition. After this clock transition, the stimulus produces a high impedance state for the rest of the run in order to not interfere with the feedback from the exclusive-or gate.

The output of the fifth flip-flop is passed into a Laplace Table source which has been defined to act as an ideal bandpass filter. The FREQ attribute for the LTVofV source has been defined as:

(1,-40,0) (1k,-40,0) (1Meg,0,0) (100Meg,0,0) (300Meg,-40,0)

Between 1MHz and 100MHz, the signal will pass through unhindered. At any other frequency, the signal will be attenuated.

The eye diagram is created in transient analysis. The Mod function is used to create the sawtooth waveform needed in the X Expression field. The Mod function is the remainder after integer division. Mod produces a sawtooth when used in a transient expression in the form of '(T - delay) Mod Per'. The delay value specifies the amount of analysis time that will be excluded from the eye diagram. The Per value in the Mod expression is the period of the sawtooth which is usually a multiple of the data period.

The resulting eye diagram appears in Figure 10. The top waveform is the actual output of the bandpass filter, v(FOut), over the entire 2us analysis run. The bottom waveform also plots V(FOut), but it does so versus the sawtooth waveform created by the following X expression:

(t-502.5n) mod 10n

This will create the desired eye diagram. The delay in the expression has been set to 502.5ns. The delay was set to such a large number in order to eliminate much of the initial transient of the v(FOut) waveform. The period has been set to 10ns to cover two of the clock cycles.

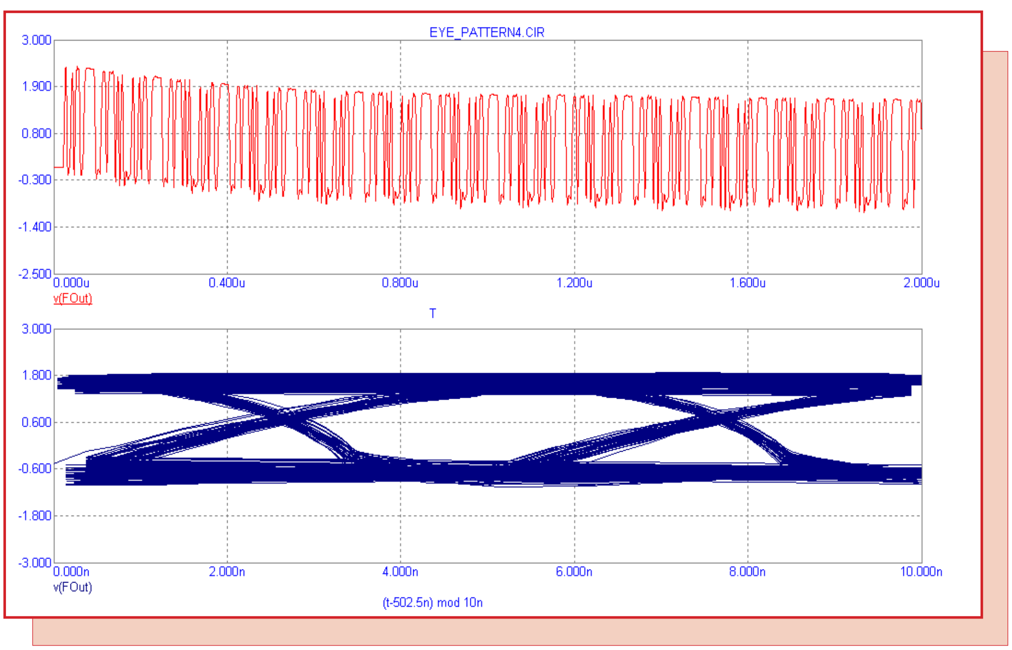


Fig. 10 - Eye Diagram Analysis

Product Sheet

Latest Version numbers

Micro-Cap 7 .... Version 7.1.7
Micro-Cap 6 .... Version 6.3.3
Micro-Cap V .... Version 2.1.2

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
User Group .... micro-cap-subscribe@yahoogroups.com