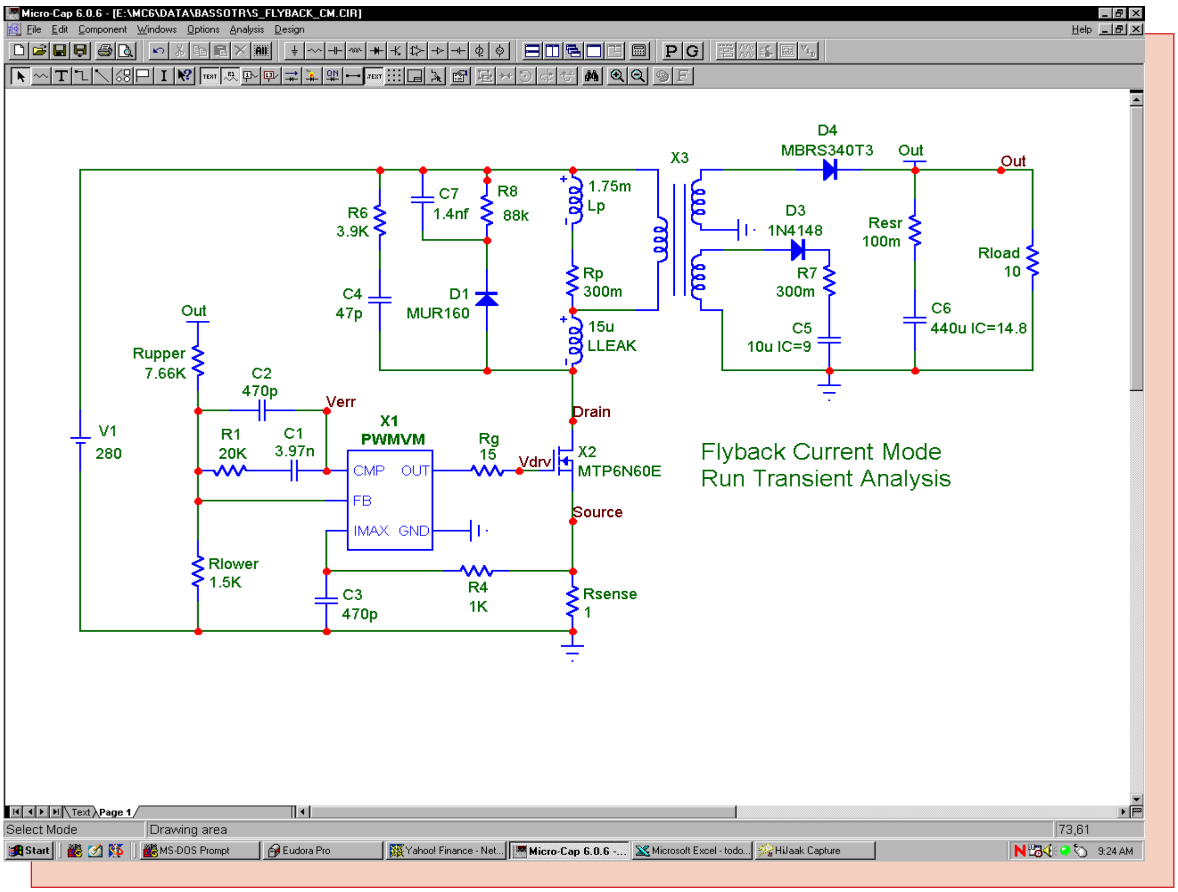
spectrum news

Applications for Micro-Cap™ Users

Winter 2000

Flyback Current Mode Converter



Featuring:

- Changing Digital Power Supplies
- Adding New Parts to the MC6 Libraries
- New Christophe Basso SMPS models

News In Preview

The first article describes how to adapt the power supplies which drive the AD and DA interfaces between analog and digital circuits. It shows you how to change the supply DC value and how to hook it up to your own regulated supply. The second article is a perennial favorite: How to add new parts to Micro-Cap. It describes the procedure for adding new parts based on primitives, subcircuits, and macros. The third article describes the new Christophe Basso designed SMPS models and demonstration circuits. These components let you simulate various switched-mode power supply topologies using both average and switched (time-domain) models. The models employed are included in an upcoming book to be published by Christophe Basso tentatively titled, "Switched-Mode Power Supply SPICE Simulation Cookbook".

Contents

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 6 Questions and Answers

Question: I want to add a new component to the MC6 libraries. How do I do it?

Answer: The long answer is the subject of a full article later on in this newsletter. Here is the short answer:

1) Copy the file containing the subckt or model statement to the data directory.

2) Add the model library file name to the nom.lib text file located in the data directory.
Add a new entry listing your file name:

.LIB "MYFILE.LIB"

3) If the new part is a primitive such as an NMOS device, nothing more is necessary. You simply place a generic NMOS in the schematic and select the new model name from the Model list in the Attribute dialog box.

If the part is a subcircuit or a macro, or it is a primitive like NMOS that you want to simply place without specifying its model name, then you must make an entry in the Component library. See the article for more details.

Question: I just installed MC6 on an NT system and I get the message "Can't find Hasp Server". What did I do wrong?

Answer: The Hasp server is the software that monitors the MC6 security key. The problem may stem from a preexisting Hasp server on the system that someone else placed there and then forgot to remove, or it may also be due to an older obsolete server on the system. To fix the problem you must uninstall the server and then reinstall it. Here's how:

If you go to the Win NT Control Panel and double click on Devices, is there a Hasp or a HaspNT in the list that and is it started?

1) If there is a server present, place disk 4 (or the CD) in the appropriate drive and run:

drive:hinstall -r

then

drive:hinstall -i

Drive is the drive letter of the floppy or CD drive as appropriate, typically A: for a floppy and something higher like D: or E: for a CD drive.

This will remove and then reinstall the security key driver. You would then need to reboot.

2) If there is no server present, place disk 4 (or the CD) in the appropriate drive and run:

drive:hinstall -i

This will install the security key driver. You would then need to reboot.

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.

Mouse Panning

There are multiple methods for navigating schematics and plots. Some of these were discussed in the last newsletter. The most powerful of all is the right-click pan. Here's how it works:

This works like sliding a sheet of paper to better see the area you want. Click the right mouse button in the schematic or plot. While holding the button down, drag the mouse in the direction you want to "slide the paper". This moves the "paper" in the direction of the mouse move. It is omni-directional, works in any mode, is always available, and works on schematics, analysis plots, and in the Shape Editor.

Rotating Parts:

Parts can be rotated while being placed or afterwards. To rotate a part during placement, click the right mouse button. During placement the left mouse button is held down. This means you are holding the left button and clicking the right button once for each rotation. There are eight rotations/reflections, so eight clicks will cycle through the entire range of orientations. To rotate a part after placement, click and hold it with the left button and click the right button until you get the orientation you want.

Selecting different part values depending upon the analysis type:

Sometimes you may want to use different values for resistors, capacitors, and inductors, depending upon whether you are doing an AC or a transient analysis. This typically occurs when you want to do an open loop measurement. Inserting an inductor into the loop lets you open the loop by changing its value to a value that is so high it effectively "opens the loop". Typically you have a capacitor that allows an AC signal to be injected and an inductor that opens the loop. For AC analysis, you might want the capacitor to have a value of 1000F, and for transient analysis, where you want the cap removed, a value of 1p. For the inductor you might want 1000H for transient and 1p for AC. Here is how it is done:

Assign LVALUE to the inductor VALUE attribute and CVALUE to the capacitor VALUE attribute. Then add these commands to either the schematic or the text area.

.define LVALUE IF(analysis==_transient,1p,1000)

.define CVALUE IF(analysis==_transient,1p,1000)

The variable analysis has different values, _transient, _ac, and _dc, _tf, and _sens, _dynamicdc, depending upon the type of analysis chosen.

The IF statements return 1P if the analysis is transient or 1000 otherwise. Now when you run AC or transient analysis the program automatically provides the desired value for the components. Of course you could do this manually, but the automated system is easier and less error prone, especially if you are planning to do many analysis runs.

Changing the Power Supplies of Digital Parts
....Adapted from an earlier Bill Steele article and updated for MC6

The default power supplies of all digital parts in MC6 is a battery whose value is set to 5V. In some applications, the supply must be changed. There are a few methods for changing the power supplies which are dependent on whether it is local or global, on what type of digital family that is to be changed, and on whether it is a digital primitive or a subcircuit. Currently, there are two power supply subcircuits contained in the DIGIO.LIB (digital input/output library): one for the 74/TTL families and one for the CD4000 family. One thing to keep in mind when changing the power supplies is that all of the components in the digital library have their propagation delays and constraint times defined in accordance with 5V operation. Editing the digital library may be a necessity if the delays or constraints are a factor in the simulation.

Changing the Default Power Supply

The two default power supply subcircuits are shown below as they appear in the DIGIO.LIB:

```
*****74/TTL POWER SUPPLY SUBCIRCUIT*****
.subckt DIGIFPWR AGND
+ optional: DPWR=$G_DPWR DGND=$G_DGND
+ params: VOLTAGE=5 REFERENCE=0
V1 DPWR AGND {VOLTAGE}
R1 DPWR AGND 1E9
V2 DGND AGND {REFERENCE}
R2 DGND AGND 1E9
R3 AGND 0 1m
.ends
```

```
*****CD4000 POWER SUPPLY SUBCIRCUIT*****
.param CD4000_VDD=5V
.param CD4000_VSS=0V
.subckt CD4000_PWR AGND
+ optional: VDD=$G_CD4000_VDD VSS=$G_CD4000_VSS
+ params: VOLTAGE={CD4000_VDD} REFERENCE={CD4000_VSS}
V1 VDD AGND {VOLTAGE}
R1 VDD AGND 1E9
V2 VSS AGND {REFERENCE}
R2 VSS AGND 1E9
R3 AGND 0 1m
.ends
```

Editing these supplies is simple. Open the DIGIO.LIB file in the DATA subdirectory. Find the power supply subcircuits in the file. For the 74/TTL power supply subcircuit, you only need to edit the line that begins with '+ params:' and place in the appropriate value for the VOLTAGE parameter. The REFERENCE parameter is by default 0V and may also be changed. The new line should appear similar to:

```
+ params: VOLTAGE=3.3 REFERENCE=0
```

For the CD4000 power supply subcircuit, edit the lines that begin with '.param'. These two .param statements control the supply rails. CD4000_VDD controls the positive power supply and CD4000_VSS controls the ground or negative supply. The edit should appear similar to:

.param CD4000_VDD=15V

There are a couple of drawbacks to this method. The first one is that this changes the value used by all circuits that reference the default value. Any circuits that have been created previously would now use the new value which may not be desired. Using one of the older circuits would require changing the DIGIO.LIB back to its original values or to edit the circuit using one of the other methods mentioned in this article. The second drawback only occurs with the 74/TTL power supply subcircuit. Some of the families that reference this subcircuit, such as the LS family, are only designed to be run at 5V. Changing the power supply would cause these parts to run incorrectly when placed in a mixed mode configuration.

Changing the Power Supply for all CD4000 Parts in a Circuit

Instead of changing the default value, the CD4000 family offers a simple way to change the power supply of all CD4000 components in a single circuit. Notice how in the CD4000 power supply subcircuit there were the two '.param' statements. These statements set the default value in the DIGIO.LIB file. However, if a '.param' statement is present in the schematic, it will override the values present in the DIGIO.LIB file. Therefore, to change the power supply for the circuit, all one has to do is add the following statement either on a schematic page or in the text area.

.param CD4000_VDD=10V

All of the CD4000 components in this circuit would then use a single 10V supply.

Using Separate Power Supplies in a Circuit

Sometimes you may need separate power supplies for digital parts in one circuit. In this case, a new power supply subcircuit and a new I/O model statement will need to be created. The best way to do this is to copy the appropriate power supply subcircuit and I/O model to a new file, so that it won't be overwritten in the case of an upgrade. You can use these as shells to easily create new ones. This example creates a 3.3V power supply for the 74HC family.

```
.subckt DIGPWR3V AGND
+ optional: DPWR=$G_DPWR DGND=$G_DGND
+ params: VOLTAGE=3.3 REFERENCE=0
V1 DPWR AGND {VOLTAGE}
R1 DPWR AGND 1E9
V2 DGND AGND {REFERENCE}
R2 DGND AGND 1E9
R3 AGND 0 1m
.ends
```

The only changes made from the 74/TTL power supply subcircuit is the name of the subcircuit and the VOLTAGE parameter value. The new I/O model for the 3.3V HC family appears below:

```
.model IO_HC_3V uio (
+ DRVH=50 DRVL=50
+ INLD=3p
+ ATOD1="ATOD_HC" ATOD2="ATOD_HC_NX"
+ ATOD3="ATOD_HC" ATOD4="ATOD_HC_NX"
+ DTOA1="DTOA_HC" DTOA2="DTOA_HC"
+ DTOA3="DTOA_HC" DTOA4="DTOA_HC"
+ TSWHL1=3.472ns TSWHL2=3.472ns
```

+ TSWHL3=3.472ns TSWHL4=3.472ns
+ TSWLH1=3.209ns TSWLH2=3.209ns
+ TSWLH3=3.209ns TSWLH4=3.209ns
+ DIGPOWER="DIGPWR3V")

The only changes made from the IO_HC model were the name of the model and the value of the parameter DIGPOWER. Note that the DIGPOWER parameter now contains the name of the power supply that was just created. Minor adjustments may need to be made in the impedances and the switching times, but the changes above will give you a solid model to work with.

Place the name of the new library in the NOM.LIB file, found in the DATA subdirectory, in the same format that all of the other libraries appear. You need to do this so that MC6 can access the library through its internal .lib statement. To use this new model, you would place IO_HC_3V into the I/O MODEL attribute of a digital primitive or if using one of the subcircuits from the digital library, you would replace all instances of IO_HC with IO_HC_3V throughout the subcircuit. Any other HC parts can still access the IO_HC model that would give them a 5V power supply.

Using an On Schematic Power Supply with a Digital Primitive

Having a digital primitive access a power supply that exists on the schematic is as simple as changing one of its attributes. Each primitive has two attributes that control the power supply nodes. The two attributes are POWER NODE and GROUND NODE. By default, these are set to $G_DPWR and $G_DGND which are the nodes that are produced by the 74/TTL power supply subcircuit. These attributes may be set to any nodes that exist in the schematic.

The node number itself may also be used in the attribute, but naming the node is a better procedure since the node numbers may change if the circuit is edited. The schematic in Figure 1 shows three

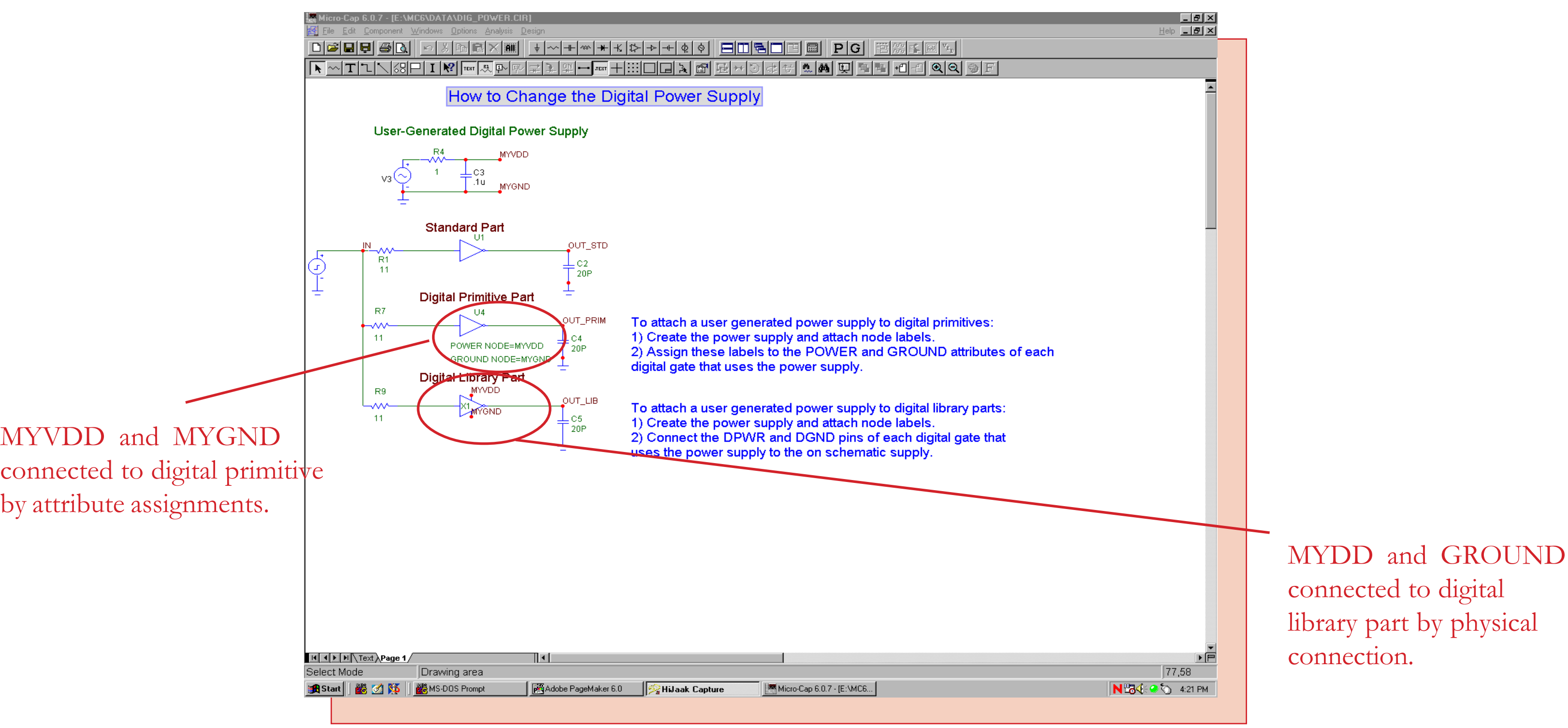


Fig. 1 - Connecting an on-schematic power supply to a digital primitive

similar digital inverters. The top one is a digital primitive that references the standard 5V supply. The middle one is a digital primitive that references the user-generated supply at MYVDD. The bottom inverter is a library part that references the user-generated supply at MYVDD. Figure 1 also shows how the POWER NODE and GROUND NODE attributes have been set to the schematic nodes MYVDD and MYGND.

Using an On Schematic Power Supply with a Digital Library Part

The power and ground nodes of all components in the digital library have been defined as optional nodes as exemplified in the following subcircuit.

```
.SUBCKT 7404_P  1A  1Y
+    optional: DPWR=$G_DPWR DGND=$G_DGND
+     params: MNTYMXDLY=0 IO_LEVEL=0
U1 inv DPWR DGND
+     1A 1Y
+     DLY_04 IO_STD MNTYMXDLY={MNTYMXDLY} IO_LEVEL={IO_LEVEL}
.model DLY_04 ugate (tplhTY=12ns tplhMX=22ns tphlTY=8ns tphlMX=15ns)
.ENDS  7404
```

The DPWR node is the power node and the DGND node is the ground node. The OPTIONAL keyword lets you add one or more nodes to the subcircuit call. If the nodes are included in the circuit, they override the default node values which in this case are derived the standard power supply subcircuit.

In order to include these nodes in a schematic, the pins DPWR and DGND must be added to the component in the Component Editor. Figure 2 displays the modified 7404_P part within the

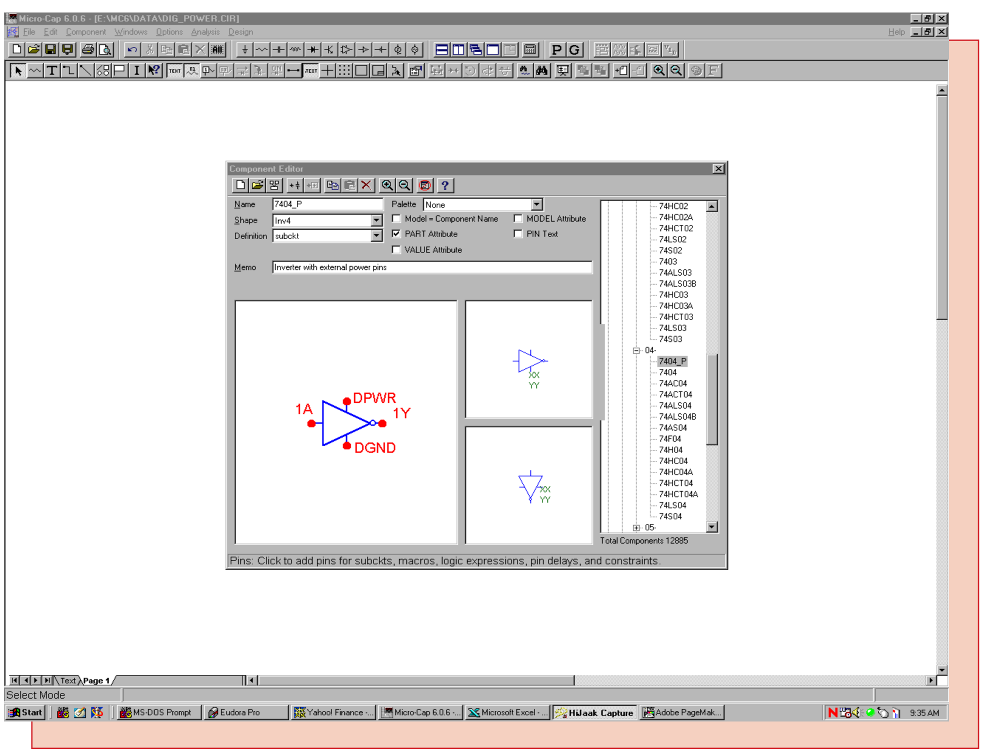


Fig. 2- Specifying optional power pins for a digital library part

Component Editor. The shape for the 7404_P is the INV4 which explicitly provides the two power pins needed. Clicking in the Shape area of the Component Editor lets you add pins to the shape. The DPWR and DGND pins were added and specified as analog pins since the power supply subcircuit consists only of analog components. The power and ground nodes can now be connected in a schematic.

Figure 3 shows the transient analysis and displays the user power supply with a simulated ripple and plots the voltage waveform at the output of a standard digital gate, a digital primitive, and a digital library part. The waveforms from the circuits that use the MYVDD power supply show evidence of the its ripple.

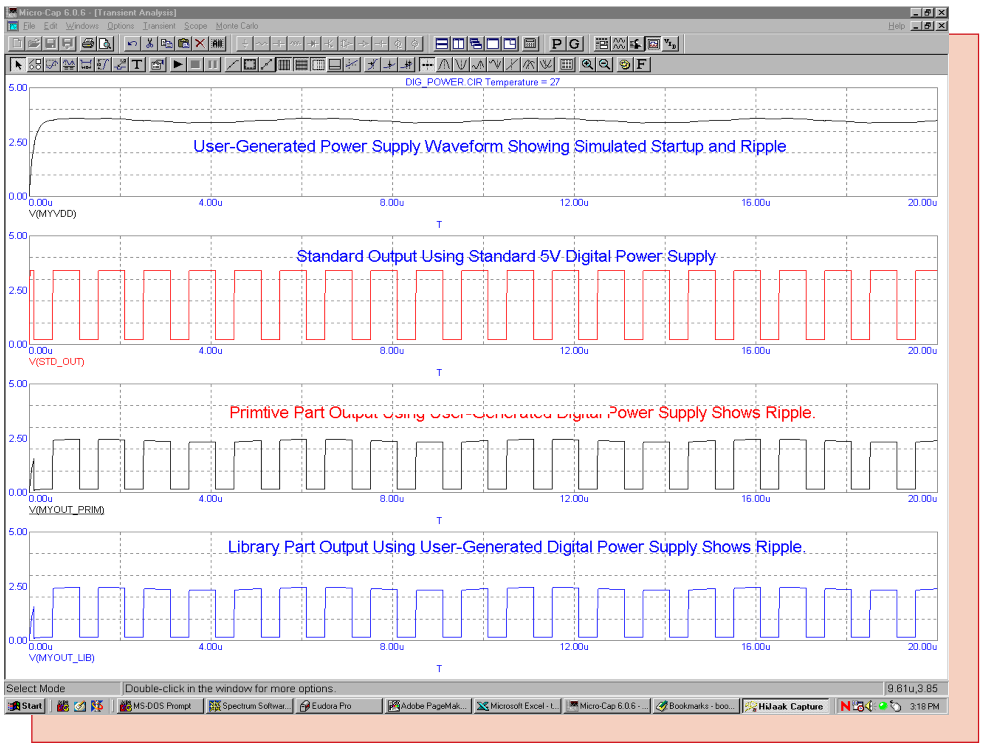


Fig. 3 - Transient analysis of the power node and the three circuit outputs

Adding New Parts to the MC6 Libraries

There are three broad classes of parts that are typically added to the libraries:

- Primitives: These use model statements and include such parts as diodes, NMOS devices, JFETs, and NPNs.

- Subcircuits: These are text descriptions written in SPICE syntax, usually by part manufacturers, that describe a circuit that models the part.

- Macros: These are Micro-Cap schematics, that describe a circuit that models the part.

To add one of these to the MC6 libraries there are three basic steps:

1) Copy the file containing model information to the data directory:

If the part is a subcircuit or a primitive like an NMOS, copy the file containing the subckt or model statement to the data directory. If the part is a macro, create the macro, label its pins with grid text, and save it to the data directory.

If the model information is a model or subckt statement, the file name should be something like MYFILE.LIB. If the model information is a macro, the file name should be something like MYMACRO.MAC. The macro file name extension must be either MAC or CIR.

2) If the part is modeled with a subckt or model statement, add the model library file name to the NOM.LIB text file located in the data directory. This step is unnecessary if the part is a macro. Using the File menu Open command, open the file NOM.LIB. Add a line in the file that lists your file name .LIB "MYFILE.LIB" like this:

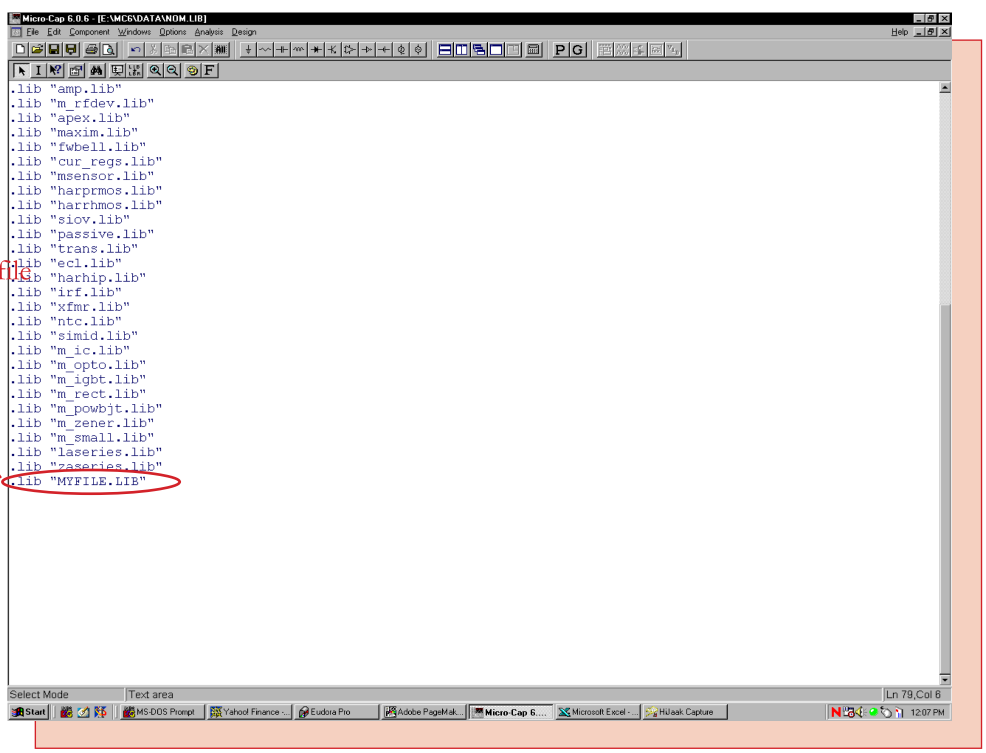


Fig. 4 - Adding the new file name to the NOM.LIB file

Close and save the NOM.LIB file. NOM.LIB is a master index file listing all of the files which contain modeling information in the form of model statements, subcircuits, and MC6 binary model libraries (*.LBR).

3) Make an entry in the Component library:

If the new part is a primitive such as an NMOS device, nothing more is necessary. You simply place a generic NMOS in the schematic and select the new model name from the Model list in the Attribute dialog box.

If the part is a subcircuit or a macro, or it is a primitive like NMOS that you want to simply place without specifying its model name, then you must make an entry in the Component library. Here is how you do that:

• Invoke the Component Editor from the Windows menu.

• Pick a location for the new part by clicking on a file name and a group name in the list at right. You can place the part in any component library file. This includes the original standard.cmp file or any you've have created along the way using the New command. Placing the new part in a small private library file has distinct advantages. When you install a new MC6 software update, the installer renames the standard.cmp file to standard.0 and copies in the new standard.cmp, making your old parts inaccessible. Of course you can always merge the old and new standard files using the Merge command, but it is simpler and cleaner to keep added parts in separate files. All component library files use the extension CMP.

• Click on the Add Component button in the Tool bar below the Component Editor name. This will add a generic part in the chosen file and group.

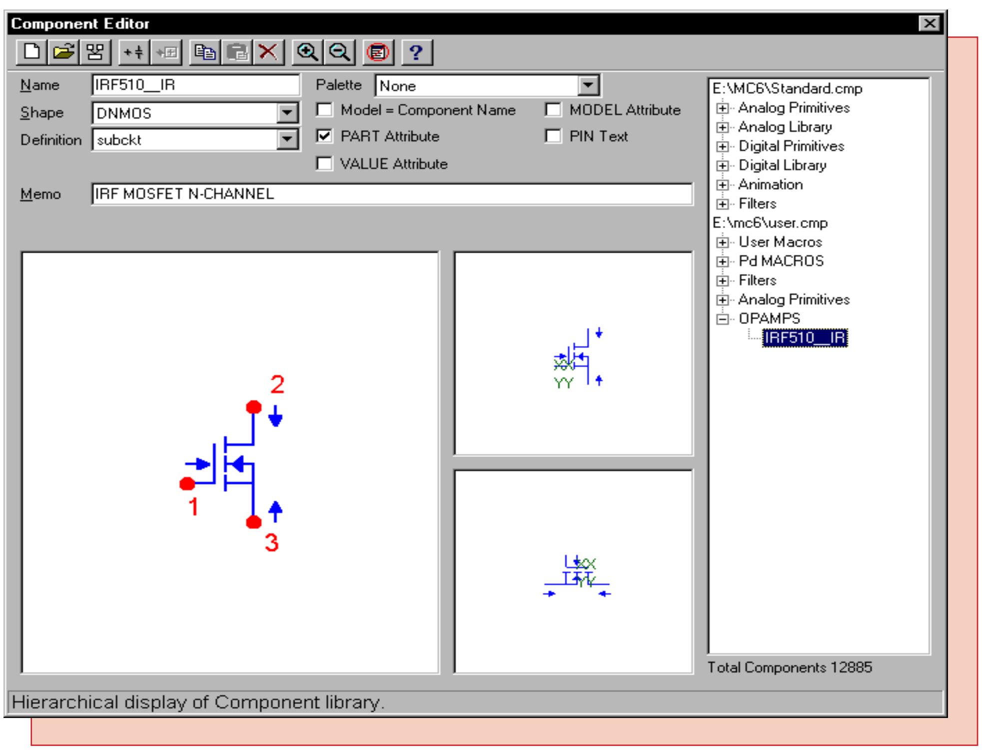


Fig. 5 - Adding the new part to the User.cmp library

• Name field: If the part is a subckt, use the same name used in the .SUBCKT statement. If the part is a macro, use the same name you saved the macro file under.

• Shape field: Pick an existing shape or one you've created with the Shape Editor.

• Definition field: Specify SUBCKT if the part is a subcircuit, MACRO if the part is a macro, or select the primitive name (such as NMOS) if the part is a primitive.

If the part is a primitive like an NMOS device, the pin names will automatically be entered. You need only drag them into position on the shape. If the part is a macro or a subcircuit, you click in the shape display to enter pins which you name and drag into position on the shape. Subcircuit pin names are usually numbers and must be the same as listed in the .SUBCKT statement. Macro pin names are usually node names that have been labeled with grid text to identify the pins.

If the part is a primitive that you want to simply place by name without having to select its model attribute, enable the Model=Component Name option. When you place a primitive part with this option disabled, the Attribute dialog box appears and you must specify the model name. If the option is enabled, then the model name is assumed to be the part name and no dialog box appears. This simplifies part placement.

Quit the Component Editor and save the file. The new part will now appear on the Component menu in the same hierarchial group where it was placed. If you can't remember where you placed it, get some ginseng root extract. They say it's good for your memory. Seriously, if you do forget, the Find Component search command on the Component menu will let you locate it even if you can only remember the first few characters of its name. Once located, you can place it in a schematic, or open the Component Editor and you'll see its location in the hierarchy.

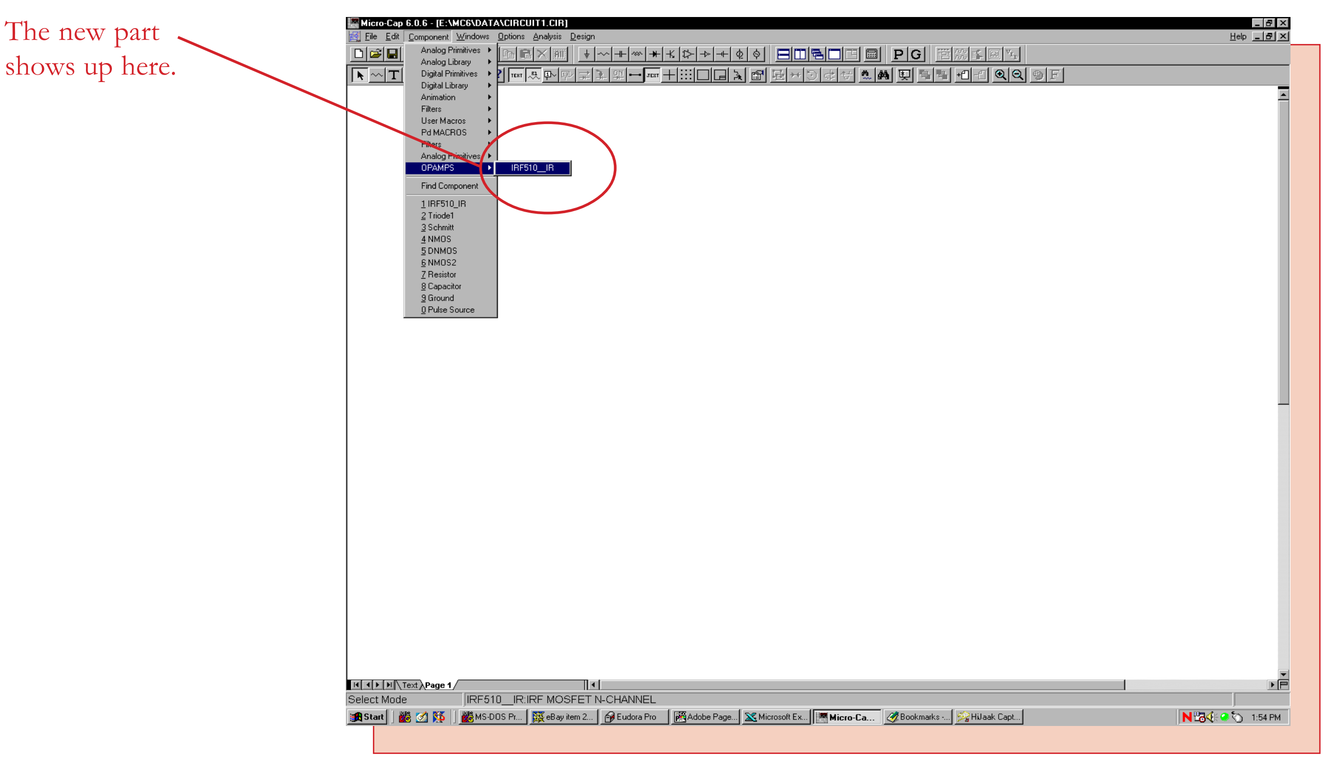


Fig. 6 - Selecting the new part from the Component menu

New Christophe Basso SMPS Models

Christophe Basso is writing a new book preliminarily entitled, "Switched-Mode Power Supply SPICE Simulation Cookbook", in which he describes many interesting and useful models, formulas, and circuits that deal with switched-mode power supplies. All of these were crafted by Christophe or translated from earlier models.

The models and their demonstration circuits have been ported to MC6 and will soon be a part of the MC6 package. Both the models and some thirty circuits which demonstrate their use are included.

Switched Models: Suitable for transient analysis:

Model	Function
PWMCM	PWM Current Mode Generic Controller
PWMVM	PWM Voltage Mode Generic Controller
PUSH_CM	PWM Current Mode Push-Pull Controller
PUSH_VM	PWM Voltage Mode Push-Pull Controller
HALF_CM	PWM Current Mode Half-Bridge Controller
HALF_VM	PWM Voltage Mode Half-Bridge Controller
FULL_CM	PWM Current Mode Full-Bridge Controller
FULL_VM	PWM Voltage Mode Full-Bridge Controller
2SWITCHCM	PWM Current Mode Two-Switch Controller
DEADTIME	Basso Deadtime Generator
DEADSYNC	Basso Deadtime Synchronizer
DEADDRV	Basso Deadtime

Averaged Models: Suitable for AC analysis (and some transient analysis)

Model	Function
PWMCCM	Generic Converter Operating in CCM
PWMDCM	Generic Converter Operating in DCM
BOOSTDCM	Generic Boost Converter Operating in CCM
BOOSTCCM	Generic Boost Converter Operating in DCM
BUCKCCM	Generic Buck Converter Operating in CCM
BUCKDCM	Generic Buck Converter Operating in DCM
FLYBACKCCM	Ridley Flyback Converter Operating in CCM
FLYBACKDCM	Ridley Flyback Converter Operating in DCM
FWDCCM	Ridley Forward Converter Operating in CCM
FWDDCM	Ridley Forward Converter Operating in DCM
FLYBACKVM	Ben-Yaakov GSIM Flyback Voltage Mode Converter
FLYBACKCM	Ben-Yaakov GSIM Flyback Current Mode Converter
BUCKVM	Ben-Yaakov GSIM Buck Voltage Mode Converter
BUCKCM	Ben-Yaakov GSIM Buck Current Mode Converter
FORWARDVM	Ben-Yaakov GSIM Forward Voltage Mode Converter
FORWARDCM	Ben-Yaakov GSIM Forward Current Mode Converter
BOOSTVM	Ben-Yaakov GSIM Boost Voltage Mode Converter
BOOSTCM	Ben-Yaakov GSIM Boost Current Mode Converter
SEPICVM	Basso Single Ended Primary Inductance Converter (Voltage Mode)
SEPICCM	Basso Single Ended Primary Inductance Converter (Current Mode)
SERPA	Basso Series Parallel Model
SERIPA	Basso Series Parallel Model

Here is the list of sample circuit files that demonstrate averaged (AC) and switched (Transient Analysis) simulations.

Demonstrating Averaged Models:

```
A_BUCK_CM.CIR
A_BUCK_VM.CIR
A_SEPIC.CIR
A_BOOST_CM_OL.CIR
A_BOOST_CM_ZOUT.CIR
A_BOOST_VM.CIR
A_BUCKBOOST.CIR
A_FLYBACK.CIR
A_FORWARD.CIR
A_RESO_DC.CIR
A_RESO_OL.CIR
```

Demonstrating Switched Models:

```
S_FULL_VM.CIR
S_FULL_CM.CIR
S_HALF_VM.CIR
S_HALF_CM.CIR
S_BUCK_CM.CIR
S_PUSH_VM.CIR
S_FLYBACK_CM.CIR
S_BUCKBOOST_CM.CIR
S_FLYBACK_VM.CIR
S_PUSH_CM.CIR
S_BUCK_VM.CIR
S_BOOST_VM.CIR
S_BUCKBOOST_VM.CIR
S_FORWARD_VM.CIR
S_2FLY_CM.CIR
S_BOOST_CM.CIR
S_BUCK_SYN2.CIR
S_BUCK_SYN.CIR
S_FORWARD_CM.CIR
S_FULL_XFMR.CIR
S_HALF_XFMR.CIR
S_2FOR_CM.CIR
```

As an example consider the circuit of Figure 7. It is a Ben-Yaakov GSIM buck voltage mode averaged model that is set up so that you can run either AC or transient analysis.

This circuit is set up to test the response of the output voltage to step change in load current in transient analysis and the small-signal open-loop gain in AC analysis. The conditional definition of the CL value opens the loop for AC analysis and closes it for the transient step response.

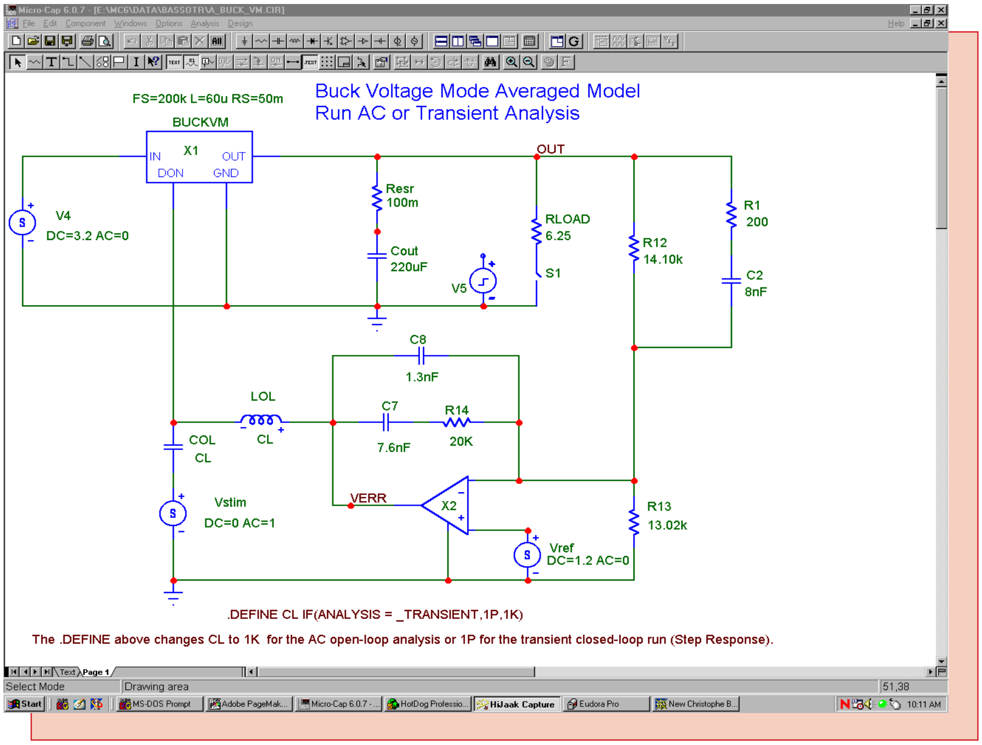


Fig. 7 - GSIM buck voltage mode averaged model circuit

Here is the open-loop AC analysis showing the output voltage gain and phase:

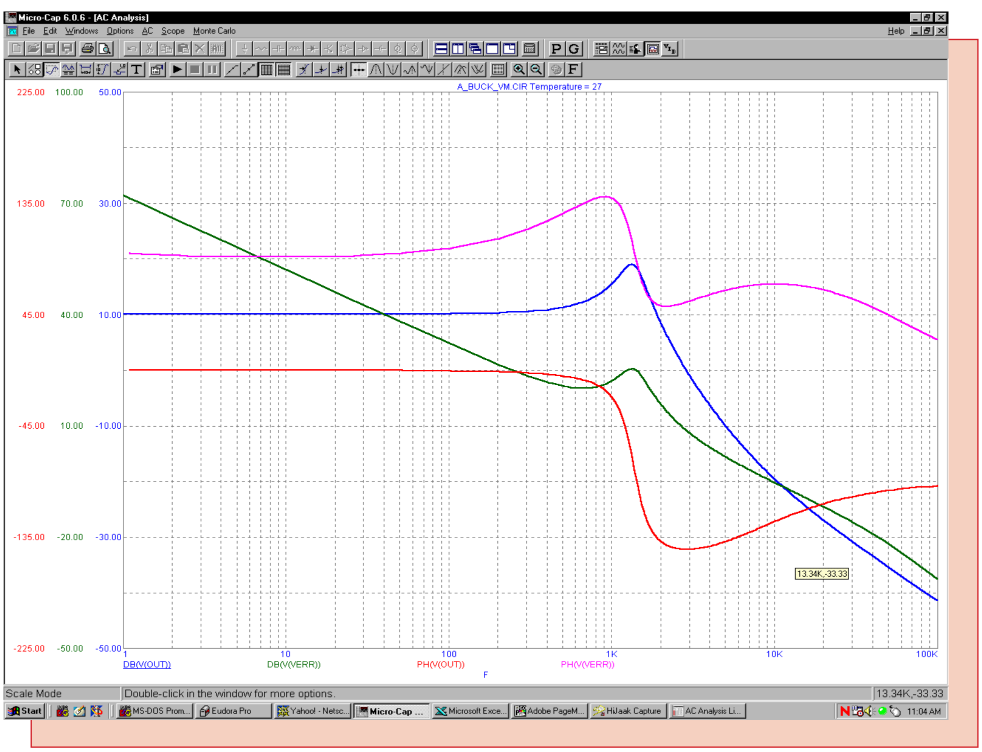


Fig. 8- GSIM buck voltage mode open loop gain plots

Here is the transient analysis step response. The plot shows the response of the output voltage to a step change in the output resistance. The source V10 controls the switch S1, which has an off resistance of 25 ohms, producing a step change in the output current from about 80ma to about 400ma. The output voltage returns to the regulated value within about 1ms.

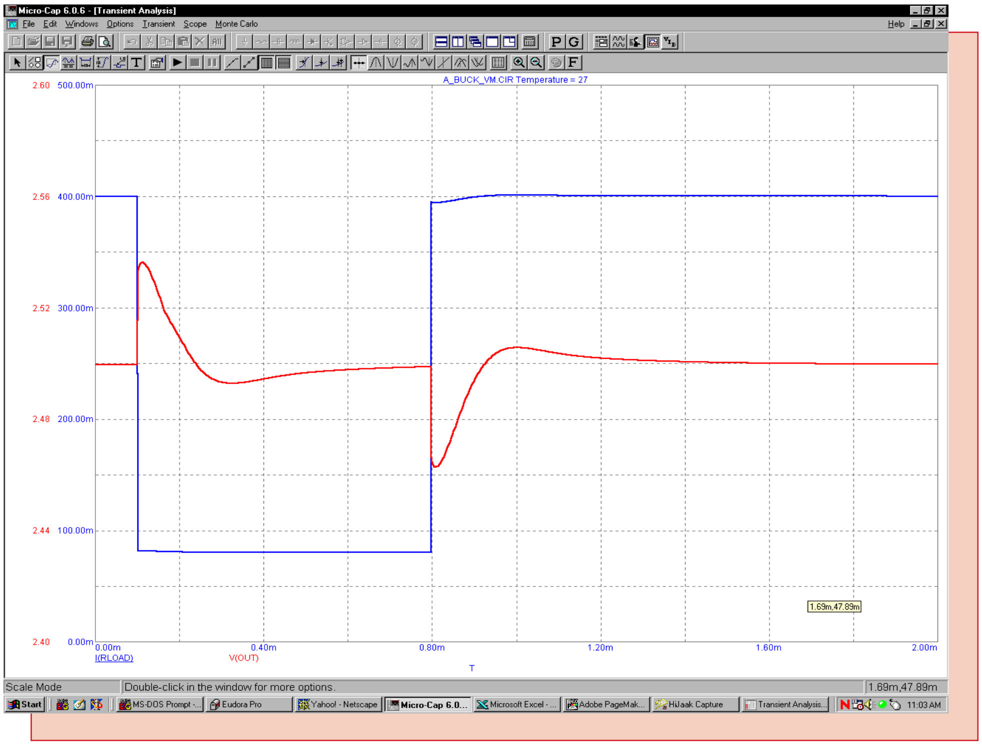


Fig. 9- Buck voltage mode averaged model step response

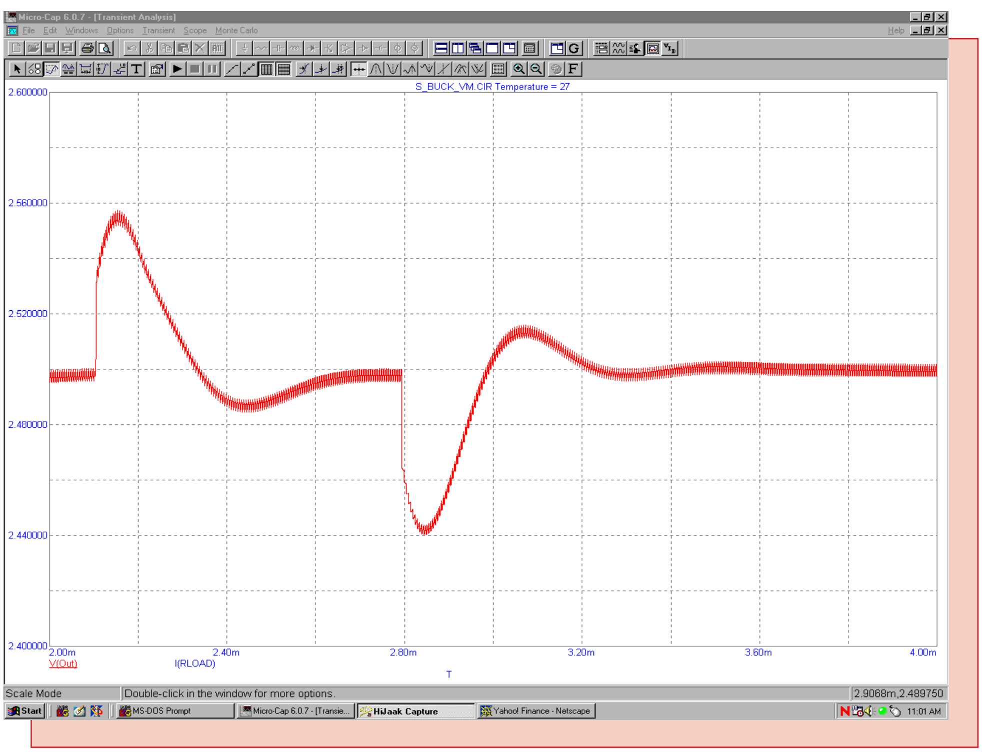


Fig. 10 - Buck voltage mode switched model step response

The interesting thing here is that you can do a transient analysis on a model that is designed to be only an averaged or AC model. Of course you do not see the moment by moment inductor current switching or the output voltage ripple, because these behaviors have been averaged out in the construction of the model.

The model is accurate enough however, to reproduce the large averaged time constants as demonstrated by the damped response to the output load change.

In Figure 10 we show the output of the switched model responding to a similar step change in the output resistance and current. It shows more switching detail, but the overall damped response is about the same.

The switched model transient analysis required about 400 seconds to complete while the averaged model required about 4 seconds. Shorter run time is a major feature virtue of averaged models.

To complete the review, here is another switched model. This is a simulation of a flyback current mode converter. Here is the schematic. At its heart is the PWMVM voltage mode controller.

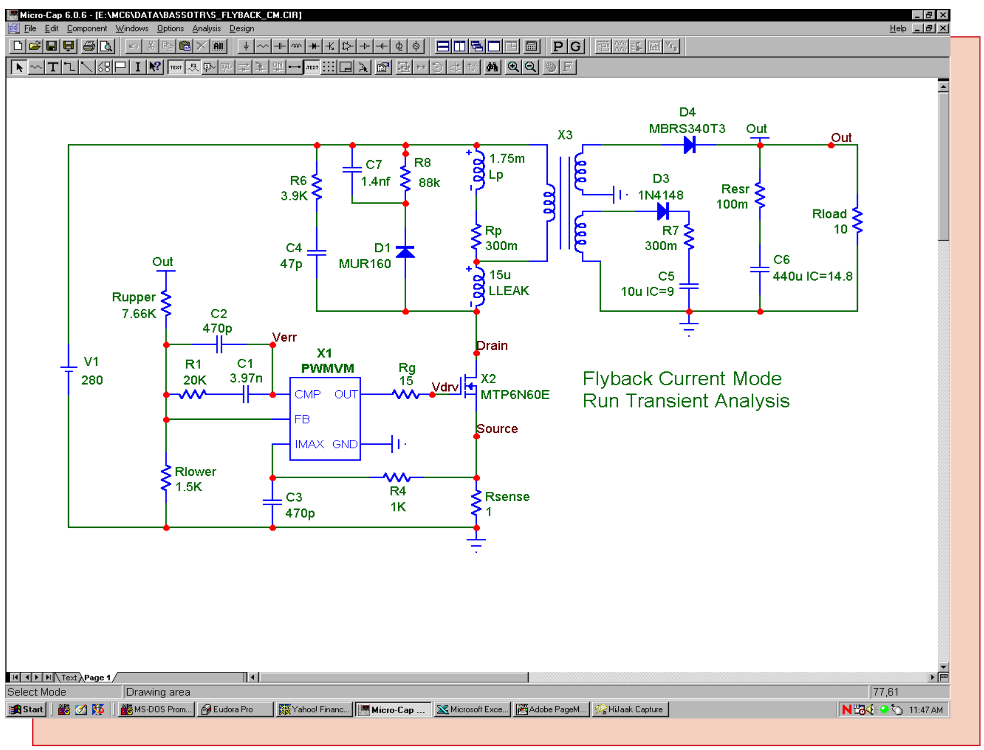


Fig. 11- Flyback current mode converter switched model sample circuit

Here are plots of the output voltage, the clipping current, the leakage current, and the drain-source voltage across the MOSFET switch.

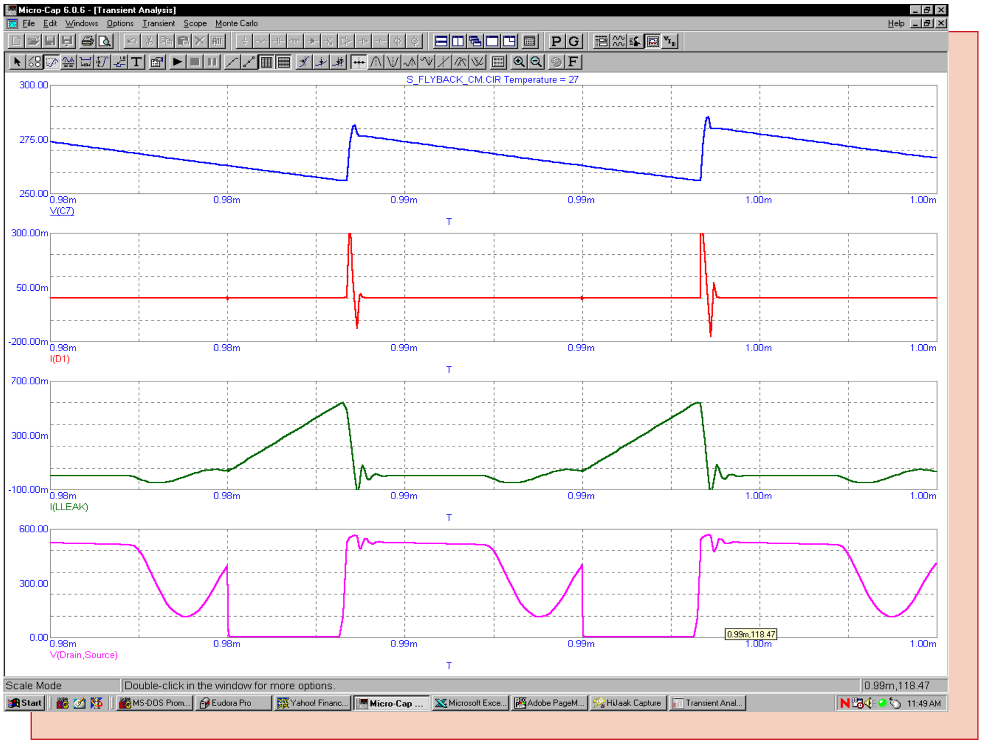


Fig. 12- Flyback current mode converter transient simulation

Product Sheet

Latest Version numbers

Micro-Cap 6 .... Version 0.7.0
Micro-Cap V .... Version 2.1.2
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 6 (LAN or standalone - 1 license) $3595.00
- Upgrade from MC5 Ver 2 to MC6 .... $500.00
- Upgrade from MC5 Ver 1 to MC6 .... $750.00
- Upgrade from MC4 to MC6 .... $1100.00

Prices are subject to change. 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.