Personal Computer
Circuit Design
Tools

intusoft

Power Specialist's App Note Book

P.O.Box 710
San Pedro, Ca. 90733-0710
Tel. (310) 833-0710
Fax (310) 833-9658
web - www.intusoft.com
email - info@intusoft.com

POWER SPECIALIST’S APP NOTE BOOK
Edited by Charles E. Hymowitz

Intusoft
P.O. Box 710
San Pedro, CA 90733-0710 USA

While the editor and the publishers believe that the information and the guidance given in this work is correct, all parties must rely upon their own skill and judgement when making use of it. Neither the editor nor the publishers assume any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

Printed in the U.S.A.

rev. 98/11

Acknowledgments

Authors
Christophe BASSO, consultant, Sinard, France

Charles Hymowitz, Intusoft, San Pedro, CA USA, charles@intusoft.com

Lawrence Meares, Intusoft, San Pedro, CA USA

Harry H. Dill, Deep Creek Technologies, Inc. Annapolis, MD USA, Testdesigner@compuserve.com

A. F. Petrie, Independent Consultant, 7 W. Lillian Ave., Arlington Heights, IL USA

Mike Penberth, Technology Sources, NewMarket Suffolk, U.K.

Martin O’Hara, Motorola Automotive and Industrial Electronics Group, Stotfold Hitchin Herts. U.K.

Kyle Bratton, Naval Aviation Depot, Cherry Point NC USA

Chris Sparr, Naval Aviation Depot, Cherry Point NC USA

Lloyd Pitzen, CCI Incorporated Arlington, VA USA

Editors
Charles Hymowitz
Bill Halal

Table Of Contents

Switched Mode Power Supply Design

Magnetics Design and Modeling

Signal Generators

Modeling For Power Electronics

Motor and Relay Modeling

Test Program Development and Failure Analysis for SMPS

Switched Mode Power Supply Design

Authors
Christophe BASSO, consultant, Sinard, France
Charles Hymowitz, Intusoft
Larry Meares, Intusoft
Scott Frankel, Analytical Engineering Services
Steve Sandler, Analytical Engineering Services

Average simulations of FLYBACK converters with SPICE3

Christophe BASSO

May 1996

Within the wide family of Switch Mode Power Supplies (SMPS), the Flyback converter represents the preferred structure for use in small and medium power applications such as wall adapters, off-line battery chargers, fax machines, etc. The calculations involved in the design of a Flyback converter, especially one which operates in discontinuous mode, are not overly complex. However, the analysis of the impact of the environment upon the system may require a lengthy period of time: ESR variations due to temperature cycles, capacitor aging, load conditions, load and line transients, the effects of the filter stage, etc. must be considered.

A SPICE simulator can help the designer to quickly implement his designs and show how they react to real world constraints. The simulation market constantly releases SMPS models, and the designer can rapidly lose himself in the eclecticism of the offer. This article will show how you can benefit from these new investigation tools.

Simulating SMPS with SPICE is not a new topic

In 1976, R. D. Middlebrook settled the mathematical basis for modeling switching regulators [1]. Middlebrook showed how any boost, buck, or buck-boost converter may be described by a canonical model whose element values can be easily derived. In 1978, R. Keller was the first to apply the Middlebrook theory to a SPICE simulator [2]. At that time, the models developed by R. Keller required manual parameter computation in order to provide the simulator with key information such as the DC operating point. Also, the simulation was only valid for small signal variations and continuous conduction mode.

Two years later, Dr. Vincent Bello published a series of papers in which he introduced his SPICE models [3]. These models had the capacity to automatically calculate DC operating points, and allowed the simulated circuit to operate in both conduction modes, regardless of the analysis type (AC, DC or TRAN). Although these models are 15 years old, other models have been introduced since then, our example circuits which have been based upon them will demonstrate how well they still behave.

Switching or average models ?

Switching models will exhibit the behavior of an electrical circuit exactly as if it were built on a breadboard with all of its nonlinearities. The semiconductor models, the transformer and its associated leakage elements, and the peripheral elements are normally included. In this case, the time variable **t** is of utmost importance since it controls the overall circuit operation and performance, including semiconductor losses and ringing spikes which are due to parasitic elements. Because SMPS circuits usually operate at high frequencies and have response times on the order of milliseconds, analysis times may be very long. Furthermore, it is practically impossible to evaluate the AC transfer function of the simulated circuit due to the switch.

Average models do not contain the switching components. They contain a unique state equation which describes the average behavior of the system: in a switching system, a set of equations describe the circuit's electrical characteristics for the two stable positions of the switch/ (es), ON or OFF. The "state-space-averaging" technique consists of smoothing the discontinuity associated with the transitions of the switch/ (es) between these two states. The result is a set of continuous non-linear equations in which the state equation coefficients now depend upon the duty cycles **D** and **D′** (1-D). A linearization process will finally lead to a set of continuous linear equations. An in-depth description of these methods is contained in D. M. Mitchell's book, "DC-DC Switching Regulators Analysis", distributed by e/j BLOOM Associates.

The general simulation architecture

The key to understanding the simulation of SMPS with a SPICE simulator is to first experiment with very simple structures. Figure 1 shows the basic way to simulate an average voltage-mode Flyback converter with its associated components. As a starting point, simply draw a minimum part count schematic: simple resistive load, output capacitor with its ESR, perfect transformer (XFMR symbol), no input filter, no error amplifier etc.

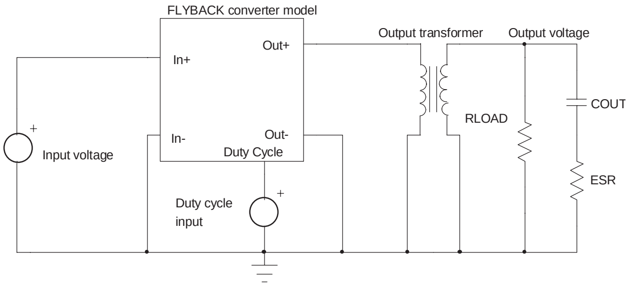


Figure 1

By clicking on the average Flyback model symbol or simply filling in the netlist file, the working parameters will be entered, i.e. the operating switching frequency, the value of the primary power coil, etc. Some recent models require the loop propagation delays or overall efficiency. The parameters for the remaining components are obvious, except for the duty cycle input source. This source will directly pilot the duty cycle of the selected model. By varying the source from 0 to 1V, the corresponding duty cycle will sweep between 0 and 100%. For the first simulation, without an error amplifier, you will have to adjust this source such that the output matches the desired value. This value corresponds to the DC operating point that SPICE needs for its calculations. The correct value can be determined incrementally or via the features in Intusoft's (San-Pedro, CA) IsSpice software. The Interactive Command Language (ICL), is a tremendously powerful language which has been primarily derived from the SPICE3 syntax and allows the designer to dynamically run SPICE commands without going back and forth from the schematic to the simulator. Below is a brief example of how the previous iteration process could be written:

```
while V(OUT)<=15   ;while the voltage at node OUT is
                    less than or equal to 15V
tran 1u 100u    ;run a TRANSIENT analysis lasting 100us
alter @Vduty[dc]=@Vduty[dc]+1mV  ;increment the duty
                                  source by 1mV steps
print mean(V(OUT)) mean(@Vduty[dc])  ;print the output
                    and the duty source average values end
```

The SPICE simulator will compute the different values and refresh the output windows until the specified conditions are met. At this time, Vduty for the desired output value is known and can be reflected back to the schematic. In order to reduce execution time, and yield a more precise result, you could also run a DC sweep, although this method is less flexible.

Simulation trick: temporarily replace your large output capacitor with a small value in order to shorten the necessary transient time at every iteration. Small values require fewer switching cycles in order to reach the output target level.

The Pulse Width Modulator gain

In a voltage-controlled Flyback SMPS, the conduction time of the primary switch depends upon the DC voltage that is compared with the oscillator sawtooth, as shown in Figure 2:

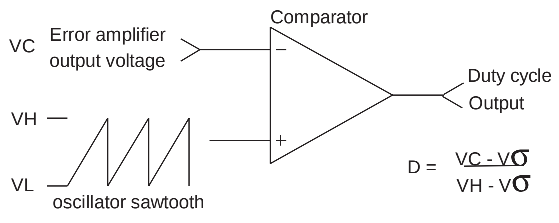


Figure 2

This circuit can be seen as a box which converts a DC voltage (the error amplifier voltage) into a duty cycle (D). The average models accept a 1 volt maximum duty cycle control voltage (D=100%). Generally, the IC's oscillator sawtooth can swing up to 3 or 4 volts, thus forcing the internal PWM stage to deliver the maximum duty cycle when the error amplifier reaches this value. To account for the 1 volt maximum input of our average models, the insertion of an attenuator with 1/(VH-Vσ) ratio after the error amplifier output is mandatory. For example, if the sawtooth amplitude of the integrated circuit we use is 2.5Vp-p, then the ratio will be: 1/2.5=0.4. In our simulation schematic, to account for the previous sawtooth peak-to-peak value, we would have to restrict the maximum output value of the error amplifier to 2.5 volts and limit the lower value to greater than VL. Figure 3 updates the schematic of Figure 1.

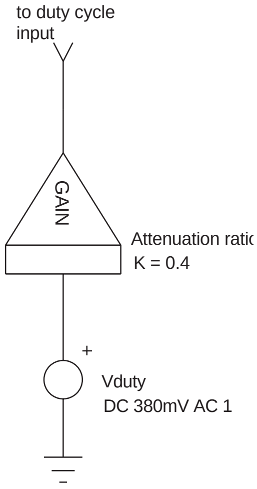


Figure 3

Performing AC simulations

We now have a functional open-loop system with the correct DC output value. The purpose of the next stage will be to sweep the duty cycle source around its DC steady-state level. This will give us the open-loop AC response of the circuit. The Vduty source keeps its DC statement to provide SPICE with a DC point, but the AC 1 command is added. Monitoring the AC output voltage yields the graph of Figure 4, which shows the control to output transfer function for a discontinuous Flyback
converter.

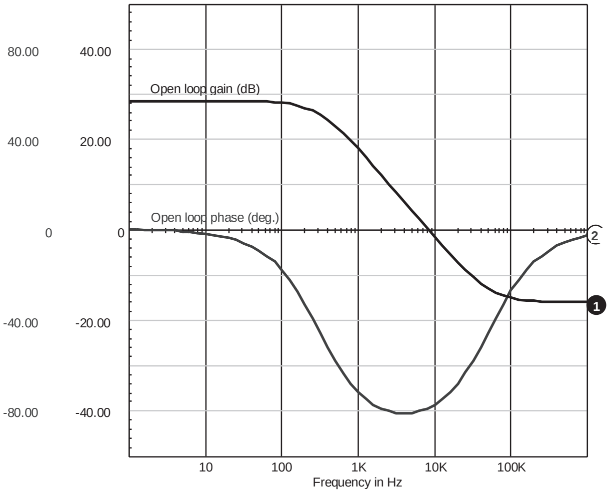


Figure 4

Adding the error amplifier

The error amplifier can be selected in function of various criteria: bandwidth, open-loop gain, etc. From a SPICE point of view, the simpler the model, the faster the simulation runs. The easiest method is to use a perfect amplifier like the one depicted in Figure 5a. This model is a simple voltage controlled source which amplifies the input voltage by the open loop gain. To overcome the problems associated with perfect sources, i.e. unrestrained output voltage, the action of a limiting element will confine the output voltage swing within a convenient range. This model is the simplest error amplifier model you can create. Figure 5b represents a transconductance type with its associated clipping network.

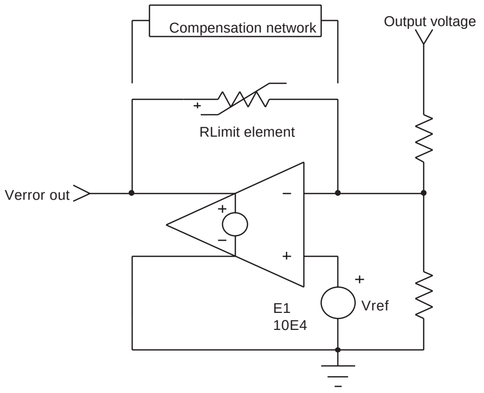


Figure 5a

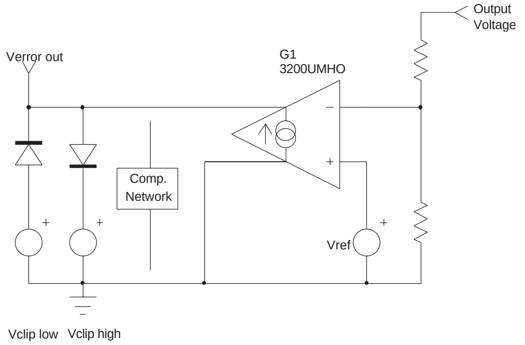


Figure 5b

To account for the characteristics of the error amplifier integrated in your real PWM controller, some components have to be added in order to tailor the response curve. The previous models do not really lend themselves to the addition of various internal pole and zero transfer functions.

Figure 5c shows another type of amplifier that will facilitate this task.

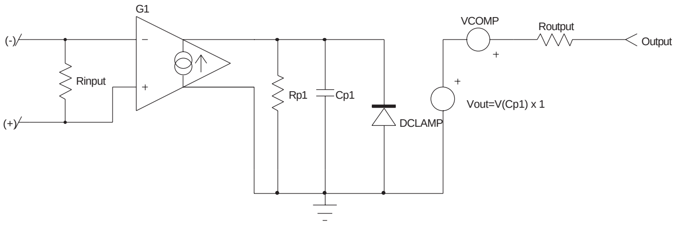


Figure 5c

This complete model associates a voltage controlled current source and a unity gain buffer. The first pole is modeled across Rp1 and Cp1, while other passive filter structures may be added between Cp1 and the unity gain buffer. The first stage output clipping is made via the diode DCLAMP and forces the output voltage of G1 to remain within the desired boundaries. Thus, the negative limit is the diode threshold voltage and the upper limit corresponds to the breakdown voltage of the diode. It can be adjusted by the IsSpice BV parameter in the diode model. In order to deliver an output voltage which matches the amplifier specifications, the VCOMP source will compensate the negative threshold of the diode, but also has to be reflected back to its BV value. In our example, the amplifier swings between 200mV and 5V with the following values: .MODEL DCLAMP D (BV=4.2V IBV=10mA) and VCOMP=680mV, as Figure 5d shows:

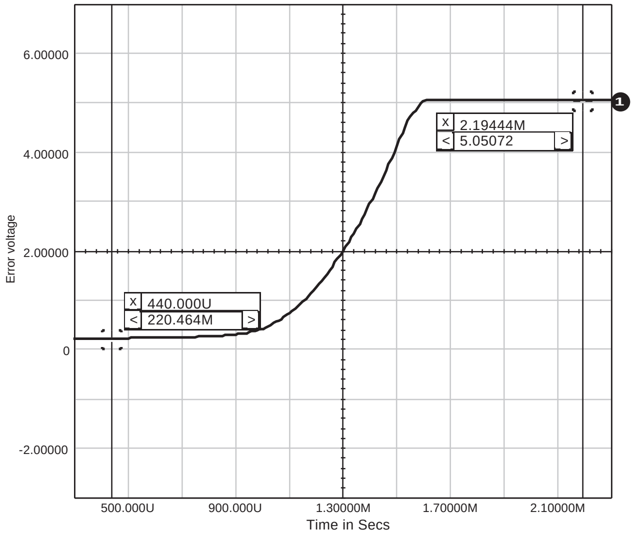


Figure 5d

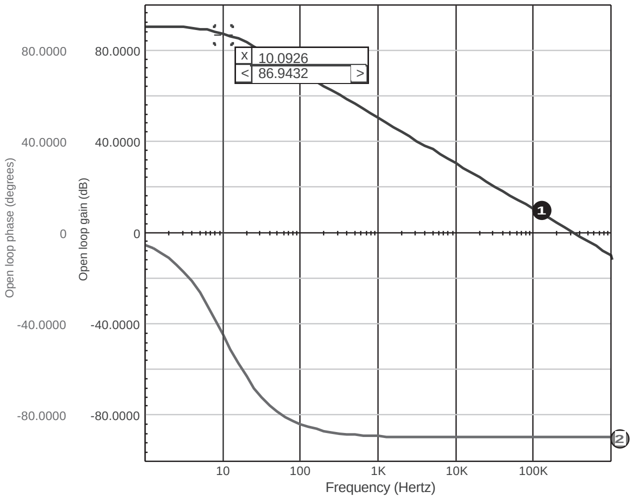


Figure 5e

The open loop gain is given by: $AV_{OL} = G1*Rp1$. With G1=100µMHOS and Rp1=316MΩ, we have an open loop gain of 90dB. The first pole is at 1/2πRp1∗Cp1, which is 10Hz if Cp1=50.36pF. Figure 5e confirms these results.

Simulation trick: Figure 5f provides an alternative for connecting the reference voltage. It offers better transient behavior and simplifies the feedback network as a first approximation. Note that Vref now becomes -Vout, with $R_{ref} = R_i$.

The R_{ref} resistor in series with the VREF source does not play a role in the loop gain, as long as AOP is closed by R_f. This condition maintains a virtual ground at the negative input of AOP, and R_{ref} is in the loop gain calculation. But if you now remove R_f, or add a capacitor in series, the Vout/Output DC gain is no longer the open loop gain of AOP alone, but is multiplied by 0.5 ($R_{ref} = R_i$) because the negative pin is ½ Vout instead of zero.

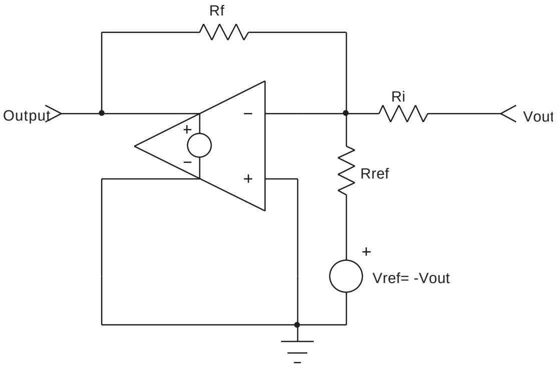


Figure 5f

Some SPICE editors not only propose the switching models of many PWM controllers, but also their standalone internal error amplifiers that can easily be incorporated in place of the previous simplified structures.

Opening a closed loop system

When the complete SMPS structure is drawn, it might be interesting to temporarily open the loop and perform AC simulations. The error amplifier can thus be isolated, and the designer has the ability to adjust the compensation network until the specifications are met. The fastest way to open the loop is to include an LC network as depicted in Figure 6. The inductive element maintains the DC error level such that the output stays at the required value, but stops any AC error signal that would close the loop. The C element permits an AC signal injection, thus allowing a normal AC sweep.

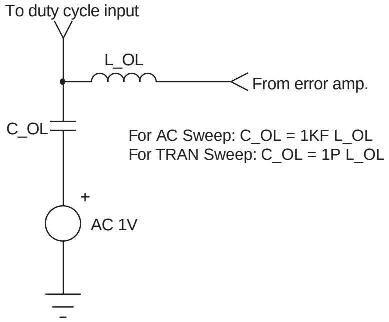


Figure 6

This method has the advantage of an automatic DC duty cycle adjustment, and allows you to quickly modify the output parameter without having to adjust the duty source.

Average simulation of the Flyback converter in discontinuous mode

Figure 7a shows a complete average Flyback converter made with Dr. Vincent Bello's models. These models use SPICE2 syntax and can therefore be run on any SPICE compatible engine.

In [3], Dr. Bello described the basic structure models (Buck, Boost ...) and showed how to create topologies such as Flyback and Forward converters. At this time, one model correspondeds to a particular Conduction Mode: Continuous (CCM) or Discontinuous (DCM). The Flyback converter operating in DCM is built with the PWMBCK (Buck) and the PWMBBSD (Discontinuous boost) as depicted in Figure 7a. The primary coil is simply shortened since it does not affect discontinuous operation in the average model. However, Dr. Bello states that despite state-space average technique results, keeping the inductor at its nominal value produces a second high-frequency pole and a RHP zero, as Vorperian demonstrated [4].

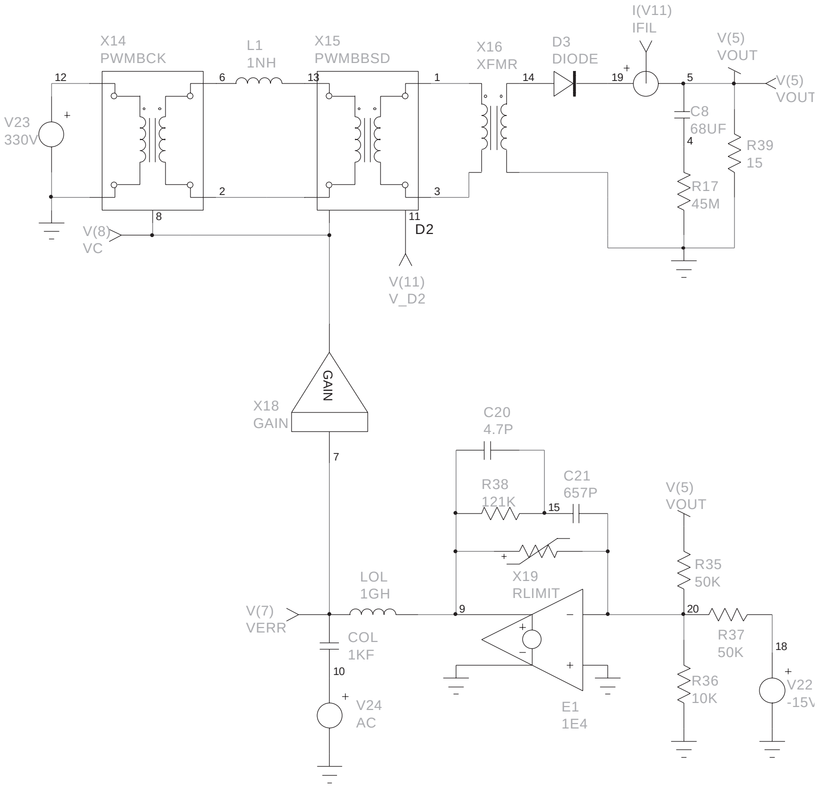


Figure 7a

In the models we used here, PWMBBSD has to be modified such that it accounts for operating parameters. In our application (see parameters below) E2 will take the following value:

E2 25 0 1 2 1.25M ; E2 = 1/2∗L∗F_{SW}=T_{SW}/2∗L

The SMPS drawn represents an off-line wall-adapter delivering 15V@1A. Its nominal characteristics and the corresponding pole-zero calculation are described below:

<u>Operating parameters:</u>

V_{out}=15V	I_{nom}=1A	R_{load}=15Ω
V_{in}=330V	L_p=4mH	N_s/N_p = 0.05
C_{out}=68µF	ESR=45mΩ	F_{sw}=100kHz
V_{ramp}=1.7Vpp		

The iteration process gave a Vduty source of 581mV, which corresponds to a duty cycle of 34.2% (0.581/1.7)

G_{PWM} = 1/1.7 = -4.6dB

$K = 2L_pF_{sw} / (R_{load} * (N_s/N_p)^2) = 0.1333$

$G_1 = (V_{in} / \sqrt{K}) * N_s/N_p = 45.19 = 33.1\text{dB}$

$G_{Vout/Vduty}$= G_1(dB) + G_{PWM}(dB) = 28.5dB (26.6)

$G_{Vout/Vin} = (D / \sqrt{K}) * N_s/N_p = 0.0468$ (-26.6dB for the open-loop DC audio susceptibility)

$F_{P1} = 2 / 2\pi C_{out}R_{load} = 312\text{Hz}$

$F_{z1} = 1 / 2\pi C_{out}ESR = 52\text{kHz}$

To verify the various gains, we open the loop and insert a DC source of 581mV, as previously shown in Figure 3. Then we ask SPICE to perform a .TF (Transfert Function) analysis:

.TF V(5) Vduty ; dV_{out}/dV_{duty} open loop gain
.TF V(5) Vin ; dV_{out}/dV_{in} open loop gain,
audio susceptibility

Once computed, the results are placed in the output file. In our application, the .TF statements gave 26.49051 and 0.04544, respectively. The open-loop characteristics of this Flyback operating in DCM were already depicted in Figure 4.

If the amplifier error exhibits a gain A_{ErrAmp} , and in the absence of a divider network (Figure 5f), the new closed loop parameters can be expressed as:

$dV_{out}/dV_{in} = (G_{Vout/Vin}) / (1 + A_{ErrAmp}G_{Vout/Vduty})$
closed loop audio susceptibility

$\varepsilon = \text{Vref} * [1 / (1 + A_{ErrAmp}G_{Vout/Vduty})]$
static error

For instance, suppose that an error amplifier with a gain of 100 and a 15V reference voltage is used to close the previous SMPS. With the simulated parameters, the static error is evaluated at: 15∗1 / (1 + 100∗26.49051) = 5.66mV. The output voltage is then 14.99434V. If we now step the input voltage by 10V, the corresponding rise in output voltage will be:
10∗0.04544 / (1 + 100∗26.49051) = 171.5µV, which is 14.99451V. Figure 7b shows how SPICE reacts to this test.

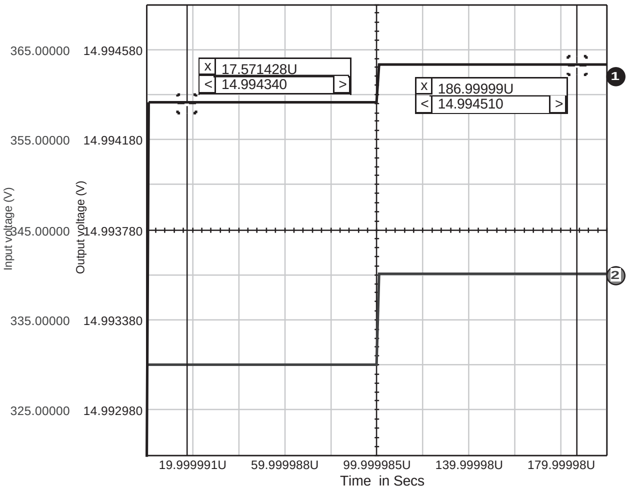


Figure 7b

Once all of the modifications are done, the designer can easily tailor the error amplifier so that the SMPS fulfills his target criteria. In our example, the compensation network gives a 21kHz bandwidth (Figure 7c).

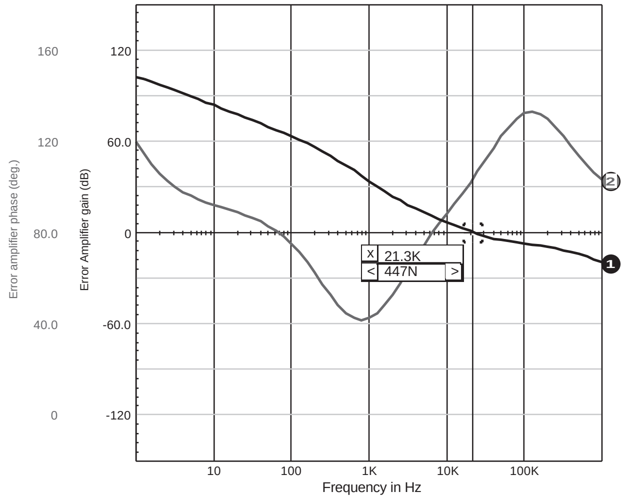


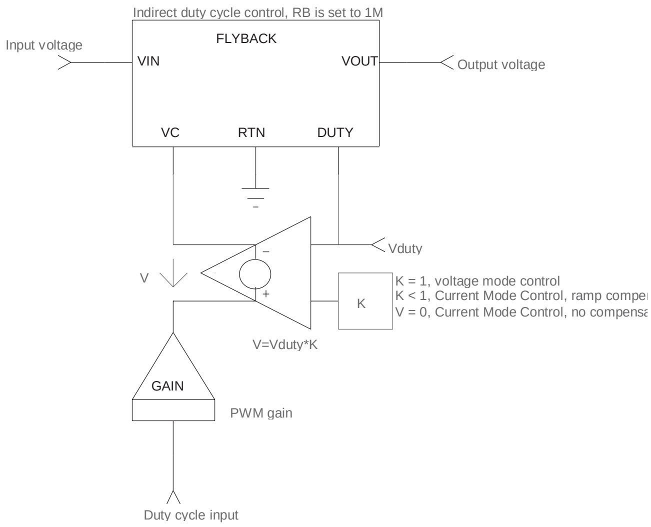
Figure 7c

New generation models

Intusoft has recently released a SPICE model library for Power Supply designers that includes new models that work in both DCM and CCM and can also be configured in current mode. The library also includes models for various PWM and PFC ICs. These SPICE3 compatible models have been developed by Steven Sandler of Analytical Engineering (Chandler, AZ) and are fully described in [5].

The models are represented by a single box in which the user enters the typical working parameters: Lp, F_{sw} etc. but also new parameters such as overall efficiency, propagation delay, and load resistance. By adding an external voltage source, you can configure the model the way you want to: voltage mode control, voltage mode control with feedforward, or current mode control with/without compensation ramp.

The Flyback model is the one of our main interests, and its symbol appears in Figure 8a with its associated parameters. The various operating modes are obtained by inserting an external voltage source in series with the VC input with the proper polarity, as shown in Figure 8a.



Indirect duty cycle control, RB is set to 1M
FLYBACK
Input voltage
VIN
VOUT
Output voltage
VC
RTN
DUTY
Vduty
V
−
+
K
K = 1, voltage mode control
K < 1, Current Mode Control, ramp compe
V = 0, Current Mode Control, no compensa
V=Vduty*K
GAIN
PWM gain
Duty cycle input

Flyback parameters:

L = primary coil, NC = current sense transformer turns ratio
NP = output transformer turns ratio; F = operating frequency,
EFF = efficiency
RB = current sense resistor; TS = current loop propagation delay

Figure 8a

Figure 8b shows the same SMPS as the one we previously studied. The circuit is built with the Flyback model in direct duty cycle control, and the simplified error amplifier structure is replaced with an amplifier model like that which is shown in Figure 5c.

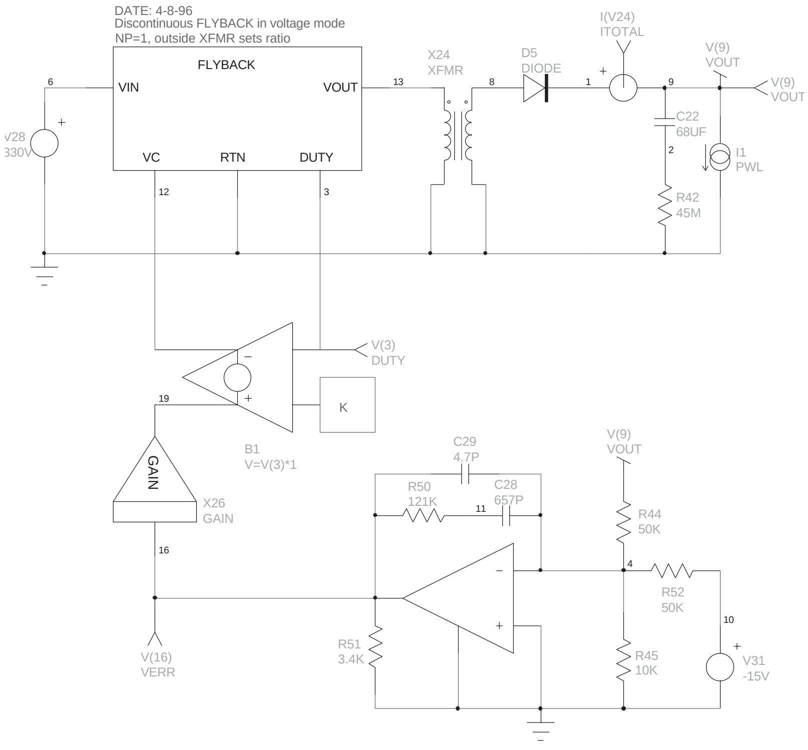


Figure 8b

The output is loaded by a current source whose purpose is to make the supply react to a sudden load increase, as shown in Figure 8c.

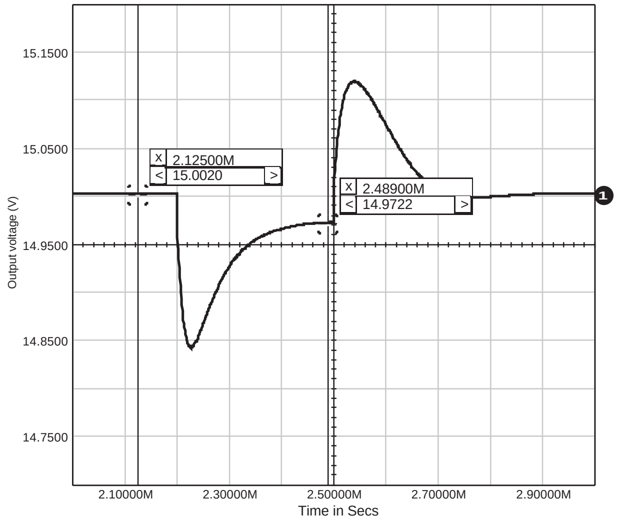


Figure 8c

At this stage, you can modify the error amplifier structure and modify the configuration until the SMPS behaves as required.

Current Mode Control SMPS

The Flyback model can be easily modified to toggle from one structure to another. Figure 9a shows the new arrangement of the current mode control without implementing any ramp compensation.

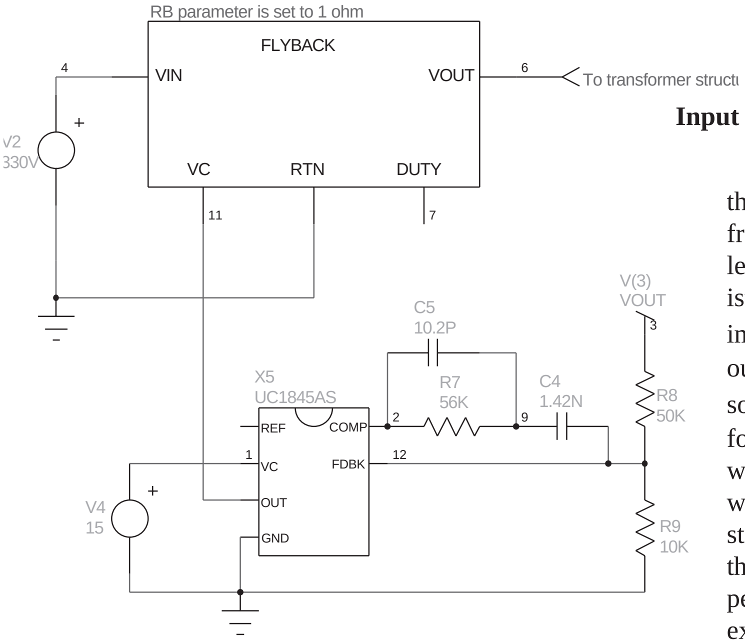


Figure 9a

In our application, the internal RB, the current sense element, is set to 1Ω. The compensation network has been slightly modified in order to account for the different gain values. The $G_{Vout/Vduty}$ gain now depends upon the current control factor (K) which is set by the sense resistor and, if present, by the internal current gain and the current sense transformer ratio. The error amplifier is replaced with the real UC3845 amplifier which is available as a single SPICE model. Its output is clipped to 1V and the output of the internal error amplifier goes through a diode before it is divided by 3, just as it is built in the real part. You may easily add a compensation ramp, as Figure 8a describes, with a simple voltage-controlled source and the appropriate coefficient.

One of the features of the Current Control Mode is its inherent feedforward capability. Figure 9b compares the response to an input step of the previous direct duty cycle SMPS and its Current Mode Control version. The **.**TF statement gives a $G_{Vout/Vin}$ of 0.0067, which is 17dB better than the equivalent direct duty cycle version.

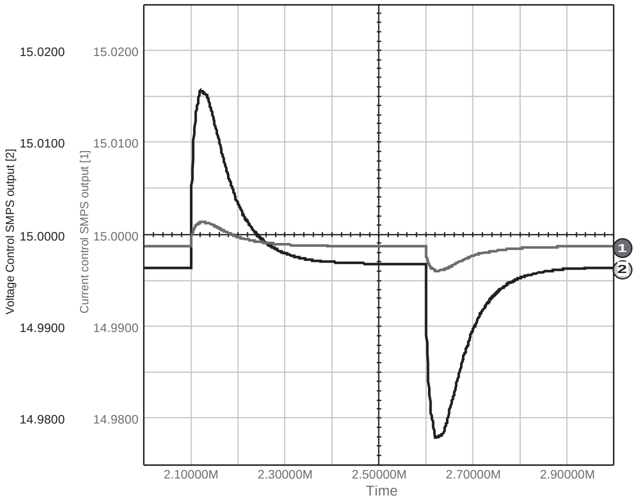


Figure 9b

Input impedance of the power supply

The input impedance has a direct impact on the overall stability when an EMI filter is connected in front of the supply. SPICE will help the designer to select an EMI structure without degrading the characteristics of the power supply. The DC value of the input impedance is easily calculated by: $Z_{in} = V^2_{in}\eta/P_{out}$. (For our 85% efficiency, 15W SMPS operating on a 330V source, the impedance is 6.17kΩ or 75.8dBΩ). Unfortunately, the input impedance is complex. It varies with the frequency and exhibits a negative peaking which is somewhat damped, depending on the SMPS structure. The EMI filter is primarily an LC network. If this network is loaded by a negative resistor whose value perfectly compensates the ohmic losses of the coil, any excitation of the LC network will make it oscillate. Because of the closed loop system, the dynamic impedance, dV_{in}/dI_{in} is negative. To avoid the previous situation, the designer should keep the input impedance, Z_{in}, well above the filter's output impedance Z_{out}. Current Control Mode Power Supplies are less sensitive to the input impedance peaking. When used in conjunction with an EMI filter, these SMPS will be less sensitive to the negative resistance effect than their Voltage Mode counterparts. Figure 9c clearly shows the differences in the Z_{in} variations depending on the SMPS topology: Voltage Mode or Current Mode Control.

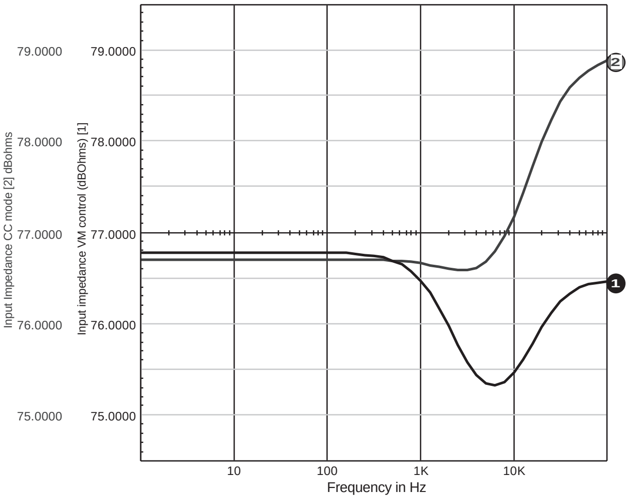


Figure 9c

Once the EMI filter is selected, an excitation stimulus on the input or the load will immediately reveal any parasitic resonance, thus inviting the designer to modify its calculations.

The Flyback converter in continuous conduction mode

In [6], D. Caldwell showed in practical terms how to implement the PWM switch theory as described by V. Vorperian [4], but, unfortunately, did not give any application examples. The Unicon model is written in SPICE2 syntax and thus permits its use on various compatible platforms. It operates in both DCM and CCM. Figure 10 shows a continuous power supply which uses the Caldwell model. The SMPS operates in continuous mode and delivers 12V to a 2.4Ω resistive load. As the author stated in his article, you need to edit the UNICON netlist and modify the EDIS generator in order to account for the operating parameters: for a 66µH coil associated with a 80kHz operating frequency, the last EDIS parameter equals 0.0947∗(T/2L).

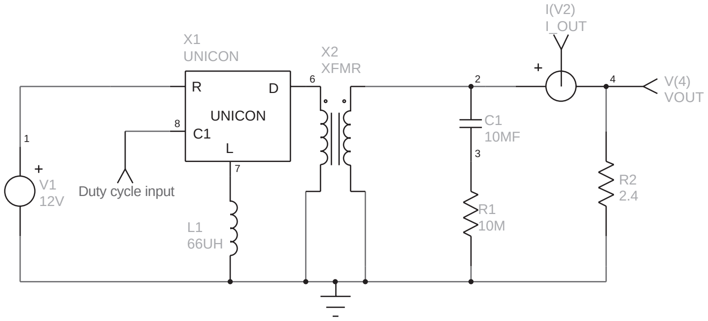


Figure 10

The Flyback operating in the continuous voltage mode is always harder to stabilize because of its second order behavior, and also because of the presence of a Right Half-Plane (RHP) zero. The RHP zero moves with the operating parameters and the designer is forced to roll-off the gain so that the SMPS stays stable within its operating range. Once again, SPICE will ease the designer's task by providing all of the necessary investigation tools to cover the numerous situations encountered by the design in its future life.

Figure 11 depicts another Flyback converter structure, using a model introduced by Lloyd Dixon in [7]. The model was originally written in PSpice syntax (Microsim, Irvine, CA), but the use of arbitrary SPICE3 B sources can easily accomplish the same functions, as we'll describe later. The model works in both operating modes, CCM and DCM. In DCM, the model naturally accounts for the high-frequency Vorperian's pole and RHP zero. The compensation of the error amplifier takes into account the presence of the low frequency pole and zeros, as described below:

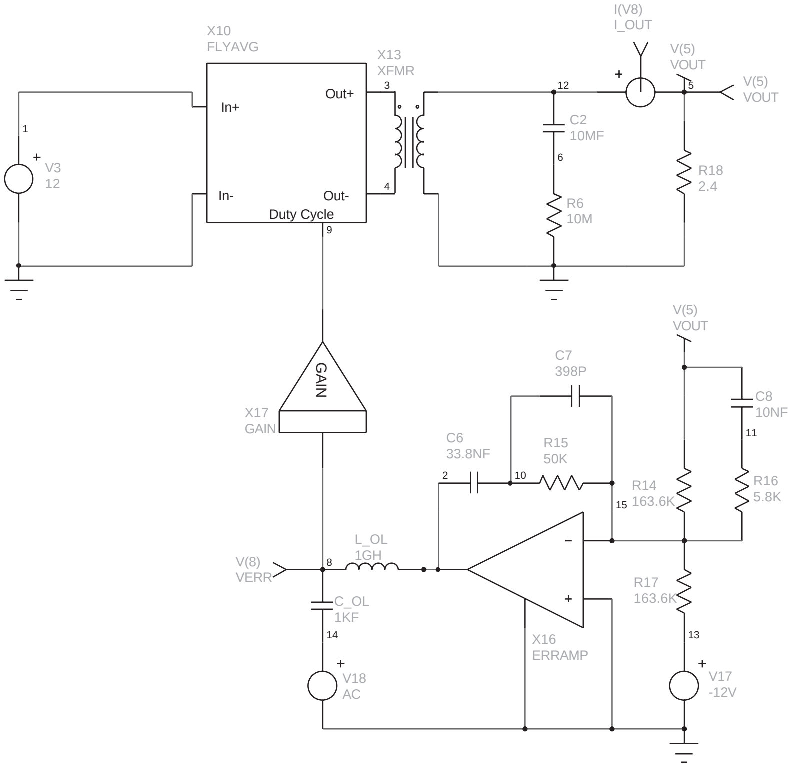


Figure 11

Operating parameters:

V_{out}=12V I_{nom}=5A R_{load}=2.4Ω
V_{in}=12V L_p=66µH N_s/N_p = 1.1
C_{out}=10mF ESR=10mΩ F_{sw}=80kHz
V_{ramp}=2.5V_{pp}

Reflected output voltage at the primary:
$V_r = V_{out} / (N_s/N_p)$ = 10.91V, neglecting the diode's voltage drop

$D_{nom} = V_r / (V_r + V_{in}) = 0.476$
$L_e = L_p / (1\text{-}D_{nom})^2 = 240.4\mu H$
$G_{PWM} = 1 / 2.5 = -7.96dB$
$G_1(dB) = V_{in} / (1\text{-}D_{nom})^2 = 32.8dB$
$G_2(dB) = 20LOG\ (N_s/N_p) = 0.83dB$
$G_{Vout/Vduty} = G_1(dB) + G_{PWM}(dB) + G_2(dB) = 25.7dB$
$G_{Vout/Vin} = D_{nom} / (1 - D_{nom}) * N_s/N_p = V_o/V_{in} = 1 = 0dB$
$F_{P1} = 1 / 2\pi N_s/N_p \sqrt{C_{out}L_e} = 93.31Hz$
$F_{z1} = 1 / 2\pi ESR * C_{out} = 1.591kHz$
$F_{z2} = R_{load} / [(N_s/N_p)^2 * D_{nom} L_e 2\pi] = 2.758kHz$ (Right Half-Plane zero)

You could also use the V. Bello models, as already described in Figure 7a. To make the converter operate in continuous conduction mode, simply replace the PWMBBSD model with the PWMBST model, without modifying its internal node list. This block is also fully documented in the reference papers [3]. The coil is set to its nominal value (66µH) and you can immediately run an AC analysis to obtain the plot of Figure 12. Nevertheless, the error amplifier compensation network which is shown in Figure 7a is no longer valid to stabilize the continuous SMPS. Figure 11 represents a possible solution.

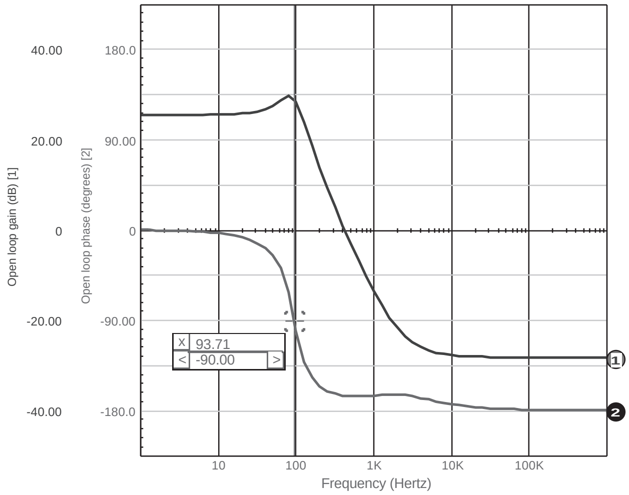


Figure 12

The -90° point corresponds to the -3dB cutoff of the second order system. Note that this value, and also the open-loop gain, are very close to the ones theoretically calculated. At F_{z1}, the slope becomes -1 with a boost in the phase plot. F_{z2} starts to act and because of its position in the right half plane, it induces a phase lag. The slope is now 0. This graphic immediately shows you where the various poles and zeros are located, and, by varying some key parameters, you can follow their respective displacements. The compensation network which has been calculated using the worst case conditions then becomes more straightforward.

The closed loop audio susceptibility is easily evaluated by decreasing C_OL and L_OL to 1pF and by adding the AC 1 statement to the input source. Figure 13 shows how the supply behaves.

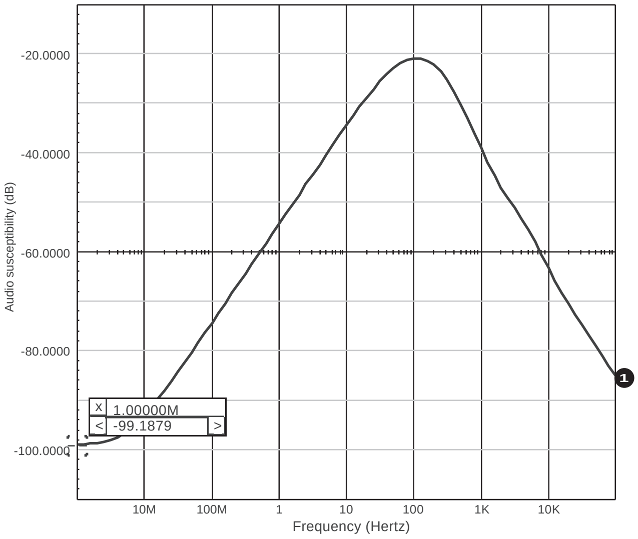


Figure 13

The open loop gain obtained by the .TF statement, $G_{Vin/Vout}$ and $G_{Vout/Vduty}$, respectively, is 0.9983 and 18.503. The AOP of Figure 11 exhibits an open loop gain of 10k. But, as we previously stated, in the absence of a feedback resistor, the Vout/Verr gain becomes 5k. The DC audio closed loop susceptibility is then: 20LOG[0.9983 / (1 + 18.503*5000)] = -99.3dB

Limitations inherent to the continuous voltage mode

The error amplifier structure depicted in Figure 11 severely impairs the time response of the power supply in the presence of large output variations. The elements responsible for this behavior are the C7 and C8 capacitors which charge to large transient values when the error amplifier's output is pushed to its maximum. This phenomenon is described in details in [9]. To highlight this problem and eventually compensate it, simply replace the load by a PWL current source that simulates a large load step. Figure 14 represents the simulation result of the error amplifier's output and indicates the amount of time which is required in order to properly recover the output transient.

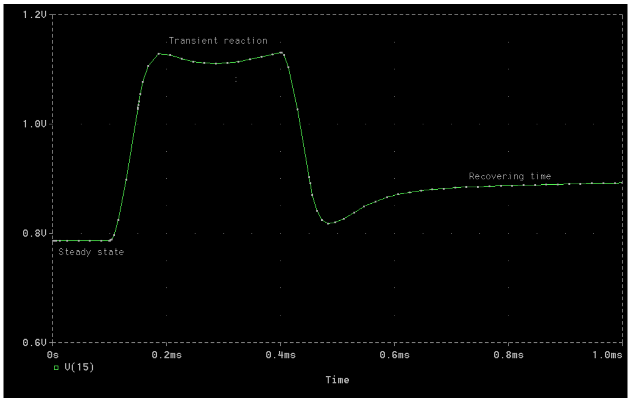


Figure 14

Using the models with different platforms

The use of a model can be extended to various SPICE compatible platforms as long as its syntax conforms to the Berkeley definition. For instance, the simple G, E or complex polynomial sources (POLY) allow the model to be used with different simulators. But if the designer adopts a proprietary syntax, he naturally reduces the implementation of his models among the remaining systems. Lloyd Dixon's model in [7] uses PSpice syntax to clamp the output voltage of some internal sources. This syntax is not SPICE3 compatible. If you would like to run the model with Intusoft's IsSpice or Cadence's Analog WorkBench (San-Jose, CA), you'll need to modify the syntax. One easy solution lies in working with arbitrary sources or B elements, since they accept in-line equations or implement If-Then-Else structures. As a first step, let's look at the following PSpice lines as they are written in Dixon's Flyavg model. The first is intended to limit the output variations of a generator within users-defined boundaries, and the second sets the value of the current generator:

```
ED2       12 0      TABLE {V(11A)-V(11)}  100M,100M 1,1
          ; PSpice voltage source
G0         4 3      VALUE =
+ {V(9,8)*1000*V(12)/(V(11)+V(12))}
          ; PSpice current source
```

If you try to run these lines using the previously referenced simulators, the internal parser will generate an error. The ED2 generator produces a voltage equal to the difference between nodes 11A and 11, but its output is clipped between 100MV and 1V. For IsSpice and AWB, these lines will look like this:

```
B_ED2 12 0 V = V(11A,11) < 100MV ? 100M :
+ V(11A,11) > 1 ? 1 : V(11A,11)          ; IsSpice
B_ED2 12 0 V = IF ( V(11A,11) < 1, IF (V(11A,11) < 100M,
+ 100M, V(11A,11)),1)          ; AWB
```

The Boolean style helps you to understand these lines: If the first condition V(11A,11) < 100MV is true, Then ED2=100MV, If the second condition V(11A,11) > 1V is true Then ED2 = 1V, Else (if any of the two previous conditions is met) ED2 = V(11A,11).

The G0 current generator can also be simply written as:

```
B_G0 4 3 I = V(9,8)*1000*V(12)/(V(11)+V(12))
          ; IsSpice and AWB
```

or, for a voltage source:

```
B_ED 4 3 V= V(9,8)*500*V(13)
          ; IsSpice and AWB
```

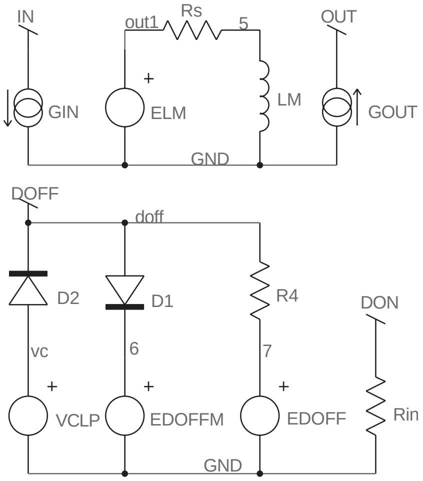
PSpice also has If-Then-Else conditions, and thus allows you to write logical expressions. For instance, suppose you want to build a perfect comparator whose output is the B1 voltage source, then the following lines illustrate how to make it with through the different syntax:

```
B1        OUT GND V = V(PLUS) > V(MINUS) ?
+ 5V : 10MV                    ; IsSpice (C syntax)

B1        OUT GND V= IF ( V(PLUS) > V(MINUS),
+ 5, 10M )                     ; AWB

E_B1      OUT GND VALUE = { IF ( V(PLUS)
+ > V(MINUS), 5V, 10MV ) }         ; PSpice
```

Because of their perfect behavior, the B elements used in comparison functions often require a small RC circuit as an output interface to slow down their transitions. Fixed resistors between the inputs and ground may also be necessary in order to provide the simulator with a DC path in the presence of infinite Pspice input impedances. IsSpice implements the statement .OPTIONS RSHUNT=100MEG, which adds a DC path of 100MEGΩ to ground for all the nodes in the simulated circuit.

Modeling the Flyback converter, other solutions

Professor Sam Ben-Yaakov, from the Ben-Gurion University of the Negev (Israel), has developed a range of models for simulating numerous converter structures [10]. The Flyback model he created is of great interest for the designer since it works for both continuous and discontinuous modes. Another nice feature lies in its simplicity, making the simplifies the conversion from one simulator to another. Figure 15 shows its internal connections and associated sources. The full model is described in Listing 1.



GIN = I(LM)*V(DON)/(V(DON)+V(DOFF))
ELM = V(IN)*V(DON)-V(OUT)*V(DOFF)/N
GOUT = I(LM)*V(DOFF)/N/(V(DON)+V(DOFF))
EDOFFM = 1-V(DON)-9M
EDOFF = 2*I(LM)*FSW*LM/V(DON)/V(IN)-V(DON)

Figure 15

Listing 1 describes a full functional netlist of the discontinuous converter of Figure 7a. The model accounts for the high frequency pole and the RHP zero highlighted by Dr. Vorperian's work. These combined actions can be seen in Figure 16. These points are well above the switching frequency and, most of the time, they can be neglected by the designer. You should always keep in mind that any reference to and/or discussion of poles or zeros above 1/2 the switching frequency is purely fictitious. The poles and zeros in question (other than the first pole and ESR zero) will always shift toward high frequencies at which the average model does not hold. The Nyquist sampling theorem restricts our ability to deal with cases in which the modulation signal is above 1/2 the sampling (switching) frequency. Needless to say, from the engineering point of view, that the frequency response above 1/3 of the switching frequency is irrelevant.

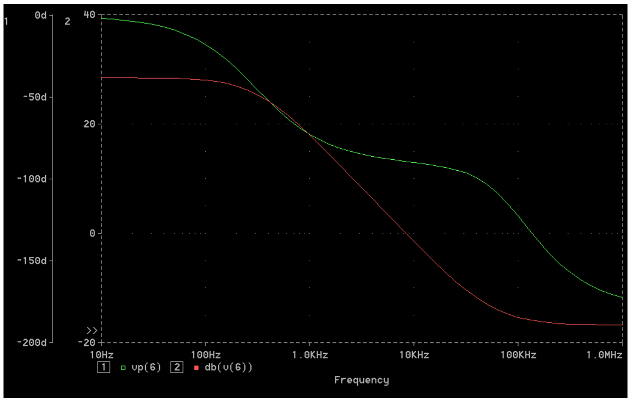


Figure 16

If we take the operating parameters of the first discontinuous Flyback converter example, we can calculate the values of the VORPERIAN's pole and zero, as described in his paper [4]:

$Sz_2 = R_f/M*(1 + M)*L_p$ Right Half-Plane Zero

$Sp_2 = 2F_{sw}*[(1/D) / (1 + 1/M)]^2$ Second High Frequency Pole, D = 0.33

$R_f = R_{load}/(N_s/N_p)^2 = 6k\Omega$ Reflected load to the transformer's primary

$M = V_{out}/(N_s/N_p)/V_{in} = 0.909$ Transfer Ratio

The numerical application leads to: $Fz_2 = 137.6$kHz and $Fp_2 = 66.3$kHz.

If you do not want the model to account for the high-frequency pole and zero, you can, in discontinuous conduction mode, decrease the LM coil between nodes 5 and 8 to 1nH. You will then obtain a phase curve which is similar to the one in Figure 4.

Primary regulated Flyback converters

The primary regulation is a feedback method in which the output level is sensed via an auxiliary winding, thus avoiding all galvanic isolation related problems. If the average models cannot highlight the regulation defaults associated with leakage inductances, they may ease your work when you tackle the stability discussion. That is to say, when multiple outputs are implemented, all of the output networks (capacitors, loads etc.) have to be reflected back to the regulated winding with the corresponding turns ratios. Figure 17 shows how to modify the previous structures in order to implement primary regulation and perform fast and efficient AC analysis.

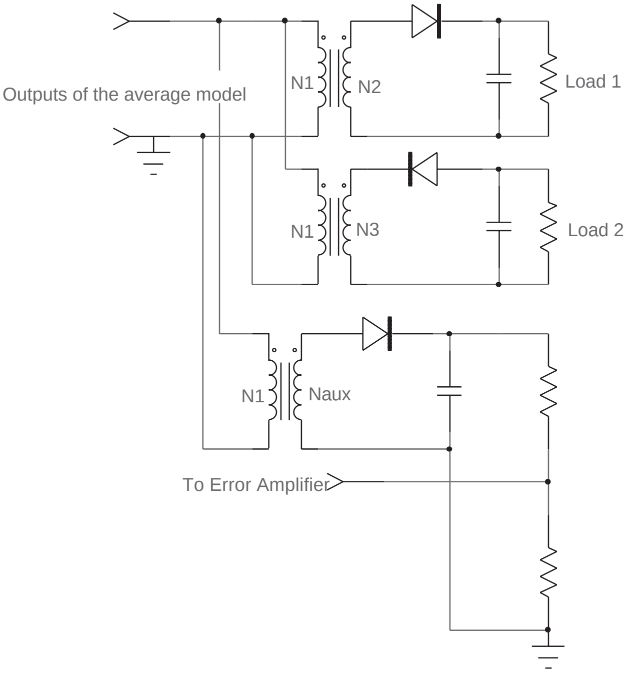


Figure 17

Conclusion

The lack of comprehensive articles upon the subject has made the SPICE approach a difficult stage for SMPS designers who are not used to the simulation philosophy. This article presents a step-by-step method to implement the available models for simulating your own Flyback structures on a SPICE platform. The proprietary libraries and the public domain models will allow you to easily simulate other kinds of topologies such as Buck or Forward converters. Power Factor Correction simulations with Boost structures may also be accomplished, as demonstrated in [3], [5], and [7].

References

1. R. D. MIDDLEBROOK and S. CUK, " A general Unified Approach to Modeling Switching Converter Power Stages ", IEEE PESC, 1976 Record, pp 18-34
2. R. KELLER, "Closed Loop Testing and Computer Analysis Aid Design Of Control Systems ", Electronic Design, November 22, 1978, pp 132-138
3. V. BELLO, "Circuit Simulation of Switching Regulators Using HSPICE ", META-SOFTWARE
   e-mail: DrVGB@AOL.COM
4. V. VORPERIAN, "Simplified Analysis of PWM Converters Using The Model of The PWM Switch, Parts I (CCM) and II (DCM) ", Transactions on Aerospace and Electronics Systems, Vol. 26, N°3, May 1990
5. S. SANDLER, "SMPS Simulations with SPICE3 ", McGraw Hill, e-mail: ssandler@aeng.COM

6. D. CALDWELL, “Techniques Let You Write General Purpose SPICE Models ”, EDN September 6, 1991
7. L. DIXON, “SPICE Simulations of Switching Power Supply Performances ”, UNITRODE Power Supply Design Seminar, SEM-800, e-mail: 71045.1206@COMPUSERVE.COM
8. Power Integration Data-Book, AN-6 and AN-8, “SPICE Modeling for Voltage-Mode Flyback Power Supplies ” (DCM and CCM)
9. L. DIXON, “Closing the Feedback Loop, appendix A, B and C ”, UNITRODE Power Supply Design Seminar, SEM-500
10. Sam BEN-YAAKOV, “Average Simulation of PWM Converters by Direct Implementation of Behavioral Relationships”, IEEE Applied Power Electronics Conference (APEC’93), pp 510-516, e-mail: SBY@BGUEE.BGU.AC.IL

Acknowledgment

I have greatly appreciated the help of Pr. Sam Ben-Yaakov and Daniel Adar (Ben-Gurion University, Israel), Dr Vincent Bello, Lloyd Dixon (Unitrode) and Daniel Mitchell (Collins-Rockwell).

Listing 1: Complete discontinuous Flyback converter with Sam Ben-Yaakov’s model

```
****** SAM BEN-YAAKOV’S FLYBACK MODEL *******
.SUBCKT FLYBACK DON IN OUT GND
.PARAM FS=100K                    ; Switching frequency
.PARAM L=4M                       ; Primary coil
.PARAM N=1                        ; Internal transformer
BGIN IN GND I = I(VLM)*V(DON)/(V(DON)+V(DOFF))
* GIN IN GND VALUE = { I(VLM)*V(DON)/(V(DON)+V(DOFF)) }
BELM OUT1 GND V = V(IN)*V(DON)-V(OUT)*V(DOFF)/{N}
* ELM OUT1 GND VALUE = { V(IN)*V(DON)-V(OUT)*V(DOFF)/{N} }
RM OUT1 5 1M
LM 5 8 {L}
VLM 8 GND
BGOUT GND OUT I = I(VLM)*V(DOFF)/{N}/(V(DON)+V(DOFF))
* GOUT GND OUT VALUE = { I(VLM)*V(DOFF)/{N}/(V(DON)+V(DOFF)) }
VCLP VC 0 9M
D2 VC DOFF DBREAK
D1 DOFF 6 DBREAK
R4 DOFF 7 10
BEDOFFM 6 GND V = 1-V(DON)-9M
* EDOFFM 6 GND VALUE = { 1-V(DON)-9M }
BEDOFF 7 GND V = 2*I(VLM)*{FS}*{L}/V(DON)/V(IN)-V(DON)
* EDOFF 7 GND VALUE = { 2*I(VLM)*{FS}*{L}/V(DON)/V(IN)-V(DON) }
.MODEL DBREAK D (TT=1N CJO=10P N=0.01)
.ENDS FLYBACK
***** Perfect Transformer model ******
.SUBCKT TRANSFORMER 1 2 3 4
RP 1 2 1MEG
E 5 4 1 2 0.05
F 1 2 VM 0.05
RS 6 3 1U
VM 5 6
.ENDS TRANSFORMER
***** Error Amp. model ******
.SUBCKT ERRAMP 20 8  3  21
*                    +  - OUT GND
RIN 20 8 8MEG
CP1 11 21 16.8P
E1 5 21 11 21 1
R9 5 2 5
D14 2 13 DMOD
ISINK 13 21 150U
Q1 21 13 16 QPMOD
ISOURCE 7 3 500U
D12 3 7 DMOD
D15 21 11 DCLAMP
G1 21 11 20 8 100U
V1 7 21 2.5
V4 3 16 80M
RP1 11 21 316MEG
.MODEL QPMOD PNP
.MODEL DCLAMP D (RS=10 BV=2.8 IBV=0.01 TT=1N)
.MODEL DMOD D
.ENDS ERRAMP
**** PWM modulator Gain Model ****
.SUBCKT PWMGAIN 1 2
E1 2 0 1 0 0.5882
.ENDS PWMGAIN
**** Sam BEN-YAAKOV’s model in Discontinuous mode ****
.TRAN 1U 1000US
.AC DEC 10 10HZ 1MEG
.PRINT AC  V(6)  VP(6)  V(13)  VP(13)
.PRINT TRAN  V(6)  V(13)
R1 4 0 45M                      ; Output Capacitor’s ESR
C1 6 4 68U                      ; Output Capacitor
X8 11 1 5 0 FLYBACK             ; Sam BEN-YAAKOV’s model
X9 5 0 6 0 TRANSFORMER          ; Output transformer
R8 6 8 50K
R9 8 0 10K
X11 2 11 PWMGAIN                ; PWM modulator gain
X12 0 8 13 0 ERRAMP             ; Error amplifier
V6 9 0 -15V                     ; Output reference voltage
R10 13 10 121K                  ; Erramp Compensation network
C5 10 8 657P                    ; Erramp Compensation network
C6 8 13 4.7P                    ; Erramp Compensation network
R12 9 8 50K
L2 2 13 1GH                   ; Open loop coil, 1P for .TRAN, 1GH for .AC
C7 2 3 1KF                    ; Open loop capacitor, 1P for .TRAN 1kF for .AC
V7 3 0 AC 1                     ; AC stimulus
* I1 6 0 PWL 0 100M 100U 100M 101U 1A 500U 1A 501U 100M
*                               ; Output step response for .TRAN run
RL 6 0 15                       ; Nominal load for .AC run
V1 1 0 330                      ; Input voltage
.END
```

A Tutorial Introduction to Simulating Current Mode Power Stages

Christophe BASSO, consultant, Sinard, France, e-mail: basso@esrf.fr

February 1997

SPICE simulation of Current Mode Control (CMC) switch Mode Power Supplies (SMPS) is certainly not a new topic. A lot of people have contributed to make this domain affordable to the design engineer and nowadays EDA packages are shipped with comprehensive dedicated libraries. However, in lack of a tutorial documentation, the choice and the implementation of the models is not always obvious to the novice designer. This article will detail the utilization of 90's models developed around the PWM switch model and more recent ones included in the new INTUSOFT's IsSpice4 SMPS library package (San-Pedro, CA).

What kind of model do I need?

Depending on the analysis to be carried on, several choices are offered to the designer. The first one is called a Small-Signal Model (SSM). It assumes that the variations of concern (e.g. output or input voltages) are small around a steady-state DC operating point. In this case, second-order AC cross-products can be neglected and the model is linear around its operating point. The SSM is then usually employed for harmonic simulations where the AC transfer functions are of interest. Do not use a SSM to simulate a 100% transient load span, the result would be wrong since AC cross-products could no longer be negligible. Some SSM models can find their DC point alone, some not and the operating point must then be fed by the designer.

Vincent BELLO has been the first in the 80's to port MIDDLEBROOK's non-linear state-space average models to the SPICE domain [1]. In BELLO's models, the previous AC terms were no longer neglected but rather dynamically multiplied by some *POLY* SPICE2 statements. Large-signal variations could then be simulated and allowed the user to visualize the effects of a 0 to 100% duty-cycle sweep. These large-signal models (LSM) are best suited for TRANSIENT runs.

Modeling SMPS, two distinct approaches

There are two ways to model a SMPS system. The first one is the well known state-space averaging (SSA) method introduced by R. D. MIDDLEBROOK in 1976 [2]. Without describing the process once more, one can state that SSA models the converter in its entire electrical form, as shown in **figure 1a** for a BUCK. In other words, the SSA process is carried over all the elements of the converter, including various in/out passive components. Depending on the converter structure, the process can be very long and complicated.

In 1986, Larry MEARES and Vatché VORPERIAN, from Virginia Polytechnic Institute (VPEC), developed the concept of the PWM switch model [3, 3a]. VORPERIAN wondered why not simply model the power switch alone, and then insert an equivalent model into the converter schematic, exactly the same way it is done when studying the transfer function of a bipolar amplifier (**figure 1b**). With his method, VORPERIAN demonstrated among other results, that the flyback converter operating in discontinuous conduction mode (DCM) was still a second order system, affected by a high-frequency RHP zero.

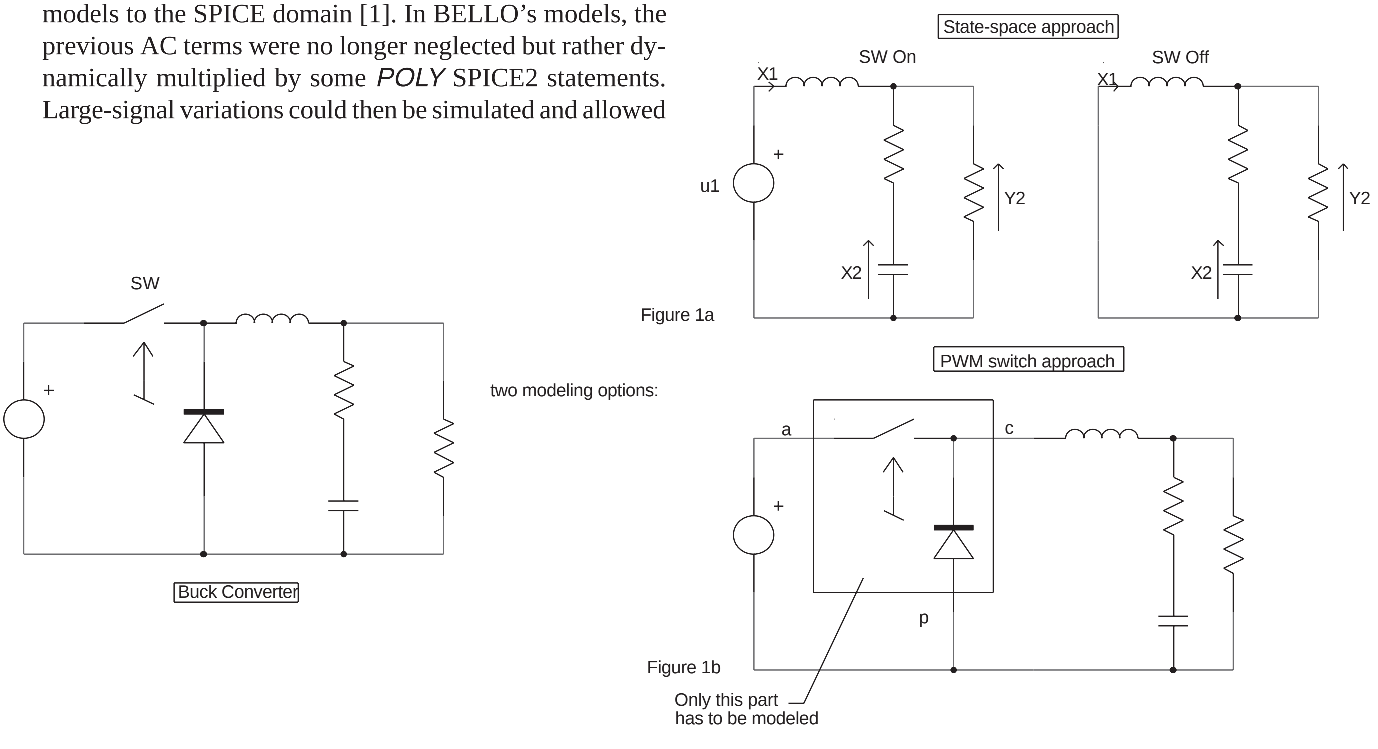


Figure 1a

Figure 1b

Figure 1

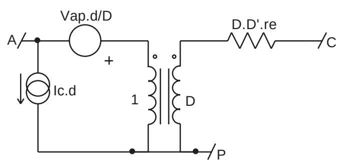


Figure 2a

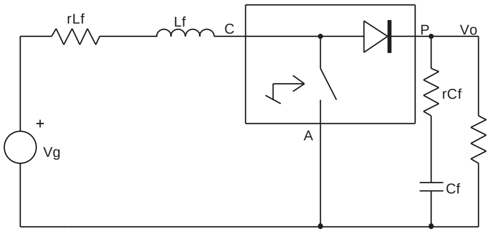


Figure 2b

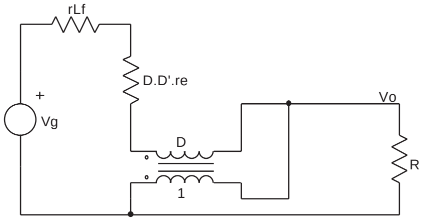


Figure 2c

Since its introduction, the PWM Switch has not been the object of many publications in the specialized press and some designers might think that its use is only reserved to modeling experts. Because its implementation is easy and powerful, we will go through a quick example, but without entering into the details of its electrical origins.

Calculating a transfer function with the PWM switch

The PWM switch model operating in Continuous Conduction Mode (CCM) is presented in **figure 2a** and can be split into an AC and a DC part. Isolating one part and applying it to the converter under study, gives the designer an immediate insight on how the converter performs in the domain of concern. Reference [3]'s BOOST example appears in **figure 2b** that shows how to properly include the PWM model. If we first consider the DC operation, Lf shorted and Cf open, **figure 2c** appears (re = R // rCf and D'=1-D).

There are different approaches to solve this kind of circuit where the transformer is not commonly wired: the classical brute force, using nodal and loop equations or the soft approach that will consist to transform the schematics until a well known structure is found. Generally speaking, the first method usually leads to correct but abstruse results in which the action of a component inside the considered function is not obvious. Inversely, the soft method produces so-called low entropy expressions [4] and yields insight into the circuit under study.

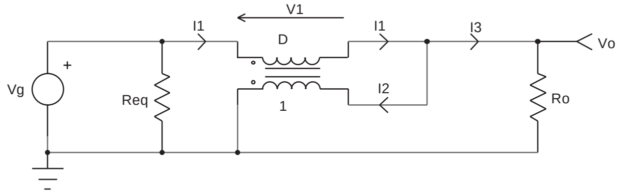


Figure 3a

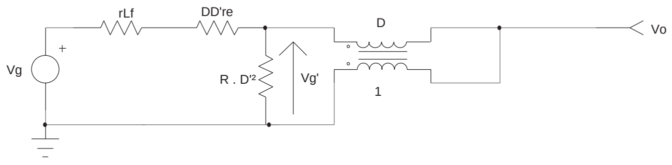


Figure 3b

Let us adopt the second method, thus redrawing a simplified version of figure 2c, as depicted in **figure 3a**. We first mark the currents, keeping in mind that a current entering a winding by a dot leaves the coupled winding by the other dot, in the same direction.

The Vo/Vg transfer function is easily obtained after a few lines:

Vg - V1 = Vo
V1 = -Vo * D
Vg + Vo * D = Vo
Vg = Vo - Vo * D so Vo/Vg = 1 / (1-D) or **Vo/Vg = 1/D'** (1)

The input impedance, or the way Ro is reflected across Vg (Req), is also simple to derive:

N1.I1 = N2.I2. Since I1 = Vg/Req, it is possible to write:
N1 * Vg/Req = N2 . I2.
From KIRCHHOFF's law, I2 = I1 - I3, with I3 = Vo/Ro
By definition, N1=D et N2=1
D * Vg/Req = Vg/Req - Vo/Ro
D * Vg/Req = (Vg * Ro - Req * Vo) / (Req . Ro).
Simplifying by Req :
D * Vg = Vg - (Req * Vo)/Ro
Vg / Vo . (1 - D) = Req/Ro. From [1], Vg/Vo = 1-D,
so: **Req = Ro * D'²** (2)

The I3/I1 ratio is important to feed the model with its DC operating points, as we will see later on. If Pin=Po, one can write: Vg. I1 = Vo . I3, so I1/I3 = Vo / Vg = 1 / D'. Back to VORPERIAN's model of figure 2b,
Ic = -I1 = - Io / D'.

With these simple formulas, **figure 3b** represents our DC BOOST where Ro has been reflected according to equation (2). We are in presence of a simple resistive divider whose output Vg' undergoes a 1/D' multiplier ratio (1). Thus, the DC Vo/Vg transfer function is really straightforward. So, after

$$\frac{Vo}{Vg} = \frac{1}{D'} \cdot \frac{1}{1 + \frac{rLf}{D'^2 \cdot R} + \frac{re \cdot D}{R \cdot D'}} = M$$

factoring the R*D'² term: (3)

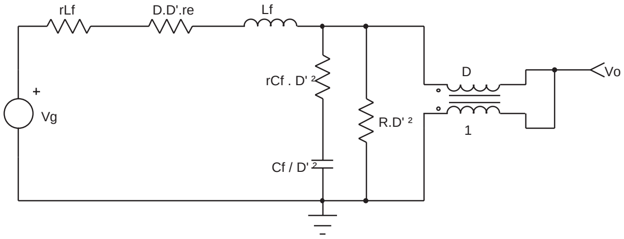


Figure 4a

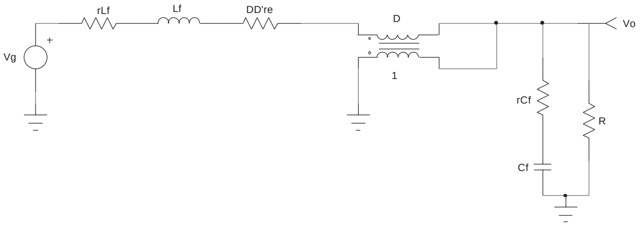


Figure 4b

The AC open-loop line to output transfer function will use the previous results to transform **figure 4a** to **figure 4b**'s drawing, where all the output elements (rCf and Cf) have been reflected to the other side of the transformer.

Figure 4b unveils a classical LC filter affected by its parasitic elements, once again followed by a 1/D' multiplier. Since we want to obtain the Vo/Vg transfer function but also the Zin Zout parameters of this circuit, a good method is to use matrix algebra. Matrix algebra is well suited for numerical computations on a computer and SPICE makes an extensive use of it. It is true that the symbolic answer given by a transfer matrix does not give the designer much insight of the circuit's operation. But one remarkable point is that once you found the matrix coefficients, the resulting transfer matrix contains, in one shot, all the parameters of interest (**figure 5b**). Further, if matrixes require a constant attention when do you manipulate them by hand, it becomes a child play when you use some mathematics programs such as Mathsoft's MathCAD (Cambridge, MA).

To solve figure 4b's problem, we draw a simplified schematic of the LC filter (**figure 5a**) where we put state and output variables.

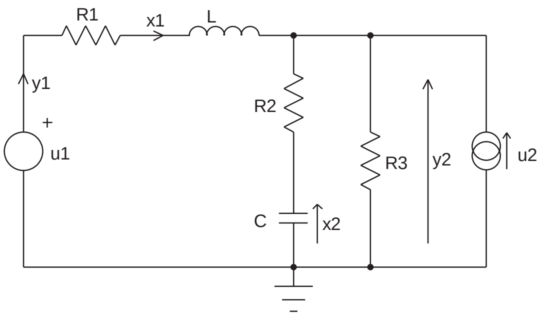


Figure 5a

$$T(s) = \begin{bmatrix} \frac{Y1(s)}{U1(s)} & \frac{Y1(s)}{U2(s)} \\ \frac{Y2(s)}{U1(s)} & \frac{Y2(s)}{U2(s)} \end{bmatrix} \quad T_{1,1} = \text{Zin}\ T_{2,1} = \text{Vo/Vg}\ T_{2,2} = \text{Zout}$$

Figure 5b

The generalized transfer function of a n^{th} order linear passive system is: $M(s) = [M (sI-A)^{-1} B + N]$, where A and M are the state coefficient matrixes, B and N the source coefficient matrixes [5]. The steps will be to write the state and output equations, ordered as follows:

<u>State equations</u>

$$x1' = -\frac{1}{L}\cdot\left[R1 + \frac{R2\cdot R3}{(R2+R3)}\right]\cdot x1 + \frac{1}{L}\cdot\left(\frac{R2}{R2+R3} - 1\right)\cdot x2 + \frac{1}{L}\cdot u1 - \frac{1}{L}\cdot\frac{R2\cdot R3}{R2+R3}\cdot u2$$

$$x2' = \frac{R3}{(R2+R3)\cdot C}\cdot x1 - \frac{1}{C\cdot(R2+R3)}\cdot x2 - \frac{R3}{C\cdot(R2+R3)}\cdot u2$$

<u>Output equations</u>

$$Y1 = x1$$

$$Y2 = x1\,R3\cdot\left(1 - \frac{R3}{R2+R3}\right) + \frac{R3}{R2+R3}\cdot x2 + u2\cdot R3\cdot\left(1 - \frac{R3}{R2+R3}\right)$$

The final results delivered by MathCAD are in a <u>clear ordered form</u>. Vo/Vg ratio is extracted from figure 5b's matrix transfer, T_{21}:

$$\frac{Vo}{Vg} = \frac{1}{D'}\cdot\frac{R3}{R1+R3}\cdot\frac{1 + s\cdot C\cdot R2}{s^2\cdot L\cdot C\cdot\left(\frac{R3+R2}{R1+R3}\right) + \left[s\cdot\frac{L + C\cdot(R2\cdot R3 + R3\cdot R1 + R1\cdot R2)}{R1+R3}\right] + 1}$$

After replacing by the elements by figure 4b's values and putting the equation into a second order form, we extract the first zero s_{z1} and the tuning frequency ω_o:

$$\frac{1}{\sqrt{L\cdot C}}\cdot\sqrt{\frac{R1+R3}{R2+R3}} = \frac{1}{\sqrt{L\cdot C}}\cdot\sqrt{\frac{rLf + re\cdot D\cdot D' + D'^2\cdot R}{rCf + R}}$$

$\omega_{z1} = 1 / R2 * C = 1 / rCf * Cf$

$s^2*L*C*(R2+R3)/R1+R3 = s^2/\omega_o{}^2$ —> $\omega_o =$

Zin and Zo can be deducted the same way. To obtain the Vo/d AC transfer function, you can replace the PWM switch model by its small-signal equivalent and re-arrange the schematic until a known structure is found, exactly as we previously did. Discontinuous Conduction Mode (DCM) study would have required the use of the appropriate PWM switch model, but the principle remains the same.

SPICE simulations with the PWM switch model

Despite the fact that the PWM switch model was intended to be an alternative tool for teaching SMPS theory, the equivalent SPICE model lends itself very well to simulations. In his original form, the PWM switch model is only able to simulate an harmonic behavior. That is to say, you will have to provide the model with its DC operating points. For a BOOST operating in CCM, the DC parameters are shown in the box below. The INTUSOFT's IsSpice4 netlist of the PWM switch model in CCM is as follow:

BOOST DC PARAMETERS:

```
D={(VOUT-VIN)/VOUT}
VAP={-VOUT}
VAC={-VIN}
VCP={-VOUT+VIN}
IA={-((VOUT^2)/RL/VIN)*D}
IP={-VOUT/RL}
IC={-VOUT/RL/(1-D)}
```

```
.SUBCKT SWITCH          1  2 3  4  {D=0.45 VAP=11 IC=0.8}
*                       A P C Control
B1 7 1 V = { V(4)*(VAP/D) }
B2 1 2  I = { V(4)*IC }
B3 7 2  I = { I(Vd)*D }
B4 9 2 V = { V(7,2)*D }
Vd 9 3 0
.ENDS
```

If we now simulate figure 2b's schematic, we obtain the well-known second order response of a BOOST converter operating in CCM.

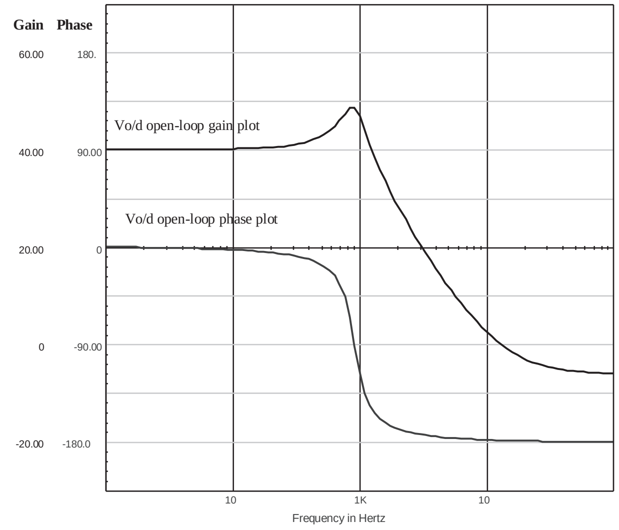


Figure 6

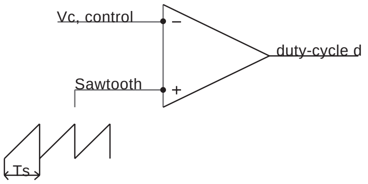
Current mode models

Numerous CMC models have been developed over the past decade. The first models suffered from their inability to predict the instabilities inherent to this kind of control. For instance, they were able to properly model the low frequency response of the CMC power stage, but the current-loop instability had to be addressed as a separate issue. In 1990, Raymond RIDLEY of VPEC, showed that a CMC power stage was best modeled by a third order polynomial form [6]. In his thesis, RIDLEY identified the current sampling action as being culprit of experimentally observed $F_{switching}/2$ sub-harmonic oscillations. Actually, a CMC differs from a voltage mode converter in the way the duty-cycle is generated. In **figure 7a**, the naturally sampled duty-cycle modulator is fed by an error voltage Vc and a reference sawtooth. This is classical voltage mode. **Figure 7b** depicts a current mode modulator where the current sense information is added, resulting in a different transfer function Fm for the Pulse Width Modulator section. Since the power stage was not affected by this change, RIDLEY built his model using the average PWM switch to which he added an internal current sampling loop. The new model is presented in **figure 7c**, for steady-state on/off inductor voltages. He(s) is the second order polynomial form RIDLEY found to represent the sampling process in the continuous time, Ri scales the current information (delivered by a simple resistor or via a transformer) and Fm models the duty-cycle generation, as explained in **figures 7a** (Ri=0) or **7b** (Ri≠0). RIDLEY's model is universal since reducing Ri to 0 (or a very low value in IsSpice4) shadows the internal current loop and turn the model into voltage mode.

Current mode instabilities

A current mode controlled SMPS exhibits one low frequency pole, ω_p, and two poles which are located at Fs/2. These poles move in relation to the duty cycle and the

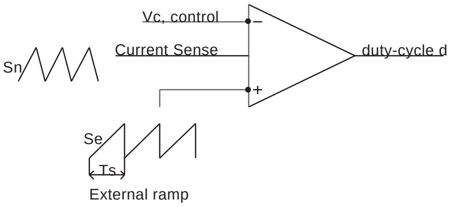
Voltage mode



$$Fm = \frac{d}{Vc} = \frac{1}{Se \cdot Ts}$$

Figure 7a

Current mode



$$Fm = \frac{1}{(Sn + Se) \cdot Ts} = \frac{1}{m_c \cdot Sn \cdot Ts} \quad \text{where} \quad m_c = 1 + \frac{Se}{Sn}$$

Figure 7b

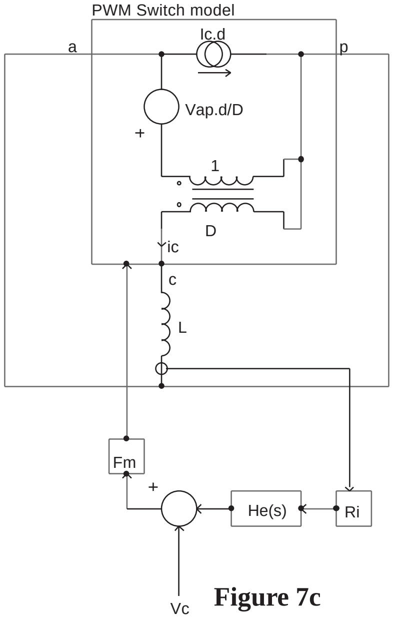


Figure 7c

external compensation ramp, when present. The two high frequency poles present a Q that depends on the compensating ramp and the duty cycle. RIDLEY demonstrated that the Q becomes infinite at D=0.5 with no external ramp, confirming the inherent instability of a current mode SMPS which has a duty cycle greater than 0.5. Q and ω_p, which are part of the V_o/V_c transfer function, are expressed as follows:

$$Q = \frac{1}{\pi \cdot (mc \cdot D' - 0.5)}$$

$$\omega_p = \frac{1}{CR} + \frac{Ts}{LC} \cdot (mc \cdot D' - 0.5)$$

where $m_c = 1 + S_e / S_n$. S_e is the external ramp slope, S_n is the inductor on-time slope. D' = 1 - D

The presence of two high-frequency poles in the V_o/V_c transfer function is due to the sampling process of the inductance current. Actually, this process creates two RHP zeroes in the current loop which are responsible for the boost in gain at Fs/2 but also stress the phase lag at this point. If the gain margin is too low at this frequency, any perturbation in the current will make the system unstable since both voltage and current loops are embedded. You can fight the problem by providing the converter with an external compensation ramp. This will oppose the duty cycle action by lowering the current-loop DC gain and correspondingly increasing the phase margin at Fs/2, which will damp the high Q poles in the V_o/V_c transfer function. As other benefits of ramp compensation, RIDLEY showed that an external ramp whose slope is equal to 50% of the inductor downslope could nullify the audio susceptibility (mc=1.5). As more external ramp is added, the low frequency pole ω_p moves to higher frequencies while the double poles will be split into two distinct poles. The first one will move towards lower frequencies until it joins and combines with the first low frequency pole at ω_p. At this point, the converter behaves as if it is operating in voltage mode (mc=32, **figure 8c**).

CMC models and IsSpice4

In his thesis report, RIDLEY presented his models in a simple SPICE2 format, without any parameter passing capability. The following IsSpice4 netlist provide greater flexibility by providing full SPICE3 parameter passing for the CCM model. The DCM listing will not be printed here but can be obtained via e-mail to the author at basso@esrf.fr.

```
.SUBCKT PWMCCM   1  2  3  4  5 {RI=0.33 L=37.5U FS=50K RL=1
+ D=0.45 VAP=11 VAC=6 IC=0.8 VP=2}
*                              A P  C C' Control
.PARAM TS = {1/FS}                ; Switching time
.PARAM PI = 3.14159               ; PI constant
.PARAM KF = {-(D*TS*RI/L)*(1-D/2)}
.PARAM KR = {((1-D)^2*TS*RI)/(2*L)}
**** PWM Switch model ****
BE2 7 1 V = V(17)*({VAP/D})
BG1 1 2 I = V(17) * {IC}
BGxf 7 2 I = I(Vxf) * {D}
BExf 9 2 V = V(7,2) * {D}
Vxf 9 3 0
**** He(s) Circuit ****
Hi 10 0  Vxf  1
C1 10 12 {TS/PI}
L1 12 13 {TS/PI}
C2 13 14 {TS/PI}
Re 14 15  -1.57
E1 15 0 12 0 -1E6
**** Summing gains ****
BEd 16 0 V = { V(1,4)*KF + V(4,2)*KR + V(15)*RI + V(5) }
**** Modulator Gain ****
BEFm 17 0 V = { V(16)*1/(VP+(VAC*TS*RI/L)) }
.ENDS
```

BUCK DC parameters:

```
D={(VOUT/VIN)*(RL+RS)/RL} ; DC duty cycle for continuous mode
D={SQRT((M^2*8*L)/(((2-M)^2-M^2)*RL*(1/FS)))}
                                        ; Discontinuous mode
VAP={VIN}
VAC={VIN-VOUT}
VCP={VOUT}
IC={VOUT/RL}
IA={(VOUT/RL)*M}
IP={IC-IA}
```

As for the PWM switch, you need to provide the model with DC operating points. BUCK DC parameters are given above. You will notice the presence of the C' connection in the model. This connection is intended to inject current information inside the model, as reference [6] thoroughly explains.

A simple BUCK converter operating at 50kHz will be simulated, according to **figure 8a**.

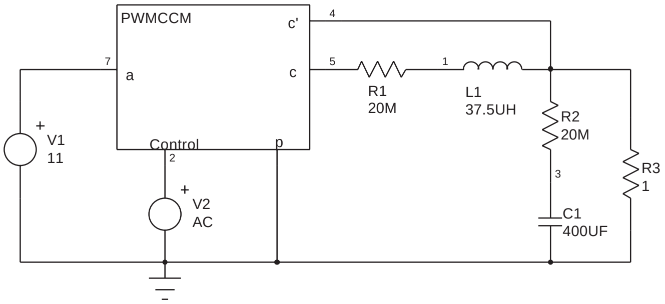


Figure 8a

The simulation results of **figures 8c** and **8d** clearly demonstrate the action of the compensation ramp upon the CMC BUCK converter. These curves have been automatically generated by IsSpice4 using its powerful Interactive Command Language in conjunction with the INTUSOFT's graphical investigation tool, IntuScope. For large mc (32), the converter's transfer function behaves like a classical voltage mode control system. Inversely, in lack of compensation ramp, the high Fs/2 Q will make the SMPS unstable in response to a transient step. The model would also let you investigate different transfer functions (audio susceptibility), Zin/Zout parameters and give an immediate insight of the compensation ramp effects.

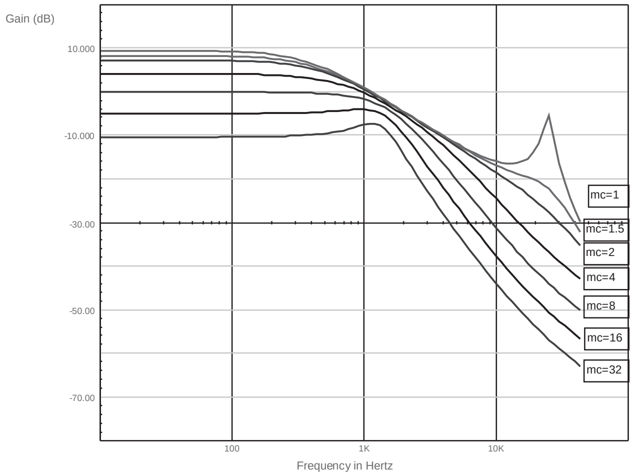


Figure 8b

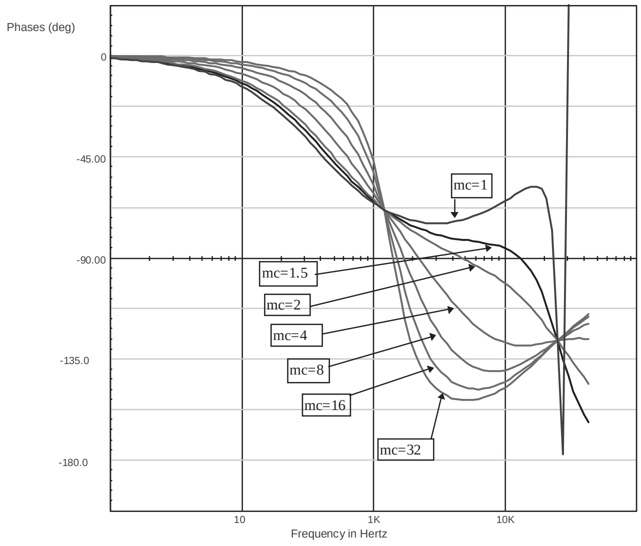


Figure 8c

New large-signal models

INTUSOFT has recently released its new SMPS library that contains numerous IsSpice4 models operating in switched or averaged mode. Among these novelties, the FLYAVG and FORWARD average models let you simulate any voltage/current mode converter and allow the testing to large steps. Reference [7] details the way the models were derived. As a simulation example, we will take figure 8a's BUCK converter to see how you can take profit of these new parts. **Figure 9a** depicts the electrical schematic whose netlist will be given to IsSpice4.

The FLYAVG model does not take into account the previous Fs/2 high frequency poles. But it is well suited to study the impact of the compensation ramp upon the overall characteristics. For instance, RIDLEY showed an nullified audio susceptibility for an external ramp whose slope equals 50% of the inductor downslope. To highlight this phenomenon, let us sweep the K coefficient which sets the amount of external compensation in figure 9a. IsSpice4's optimizer capability will ease the work by providing the adequate automation tool. You simply need to add the following line and run IsSpice4 to see multiple run results gathered upon a single graph (**figure 9b**): *OPT RAMP=0.1 TO 0.15 STEP=5M.

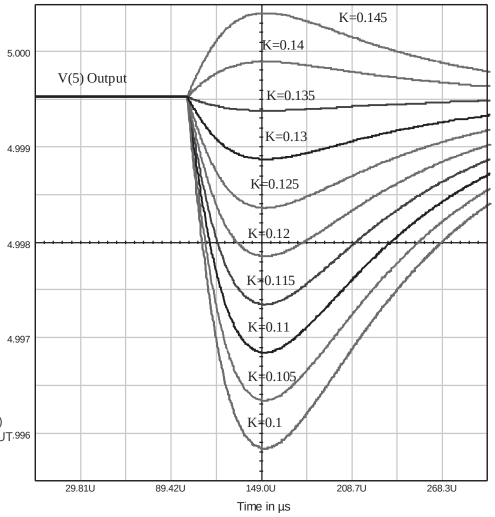


Figure 9b

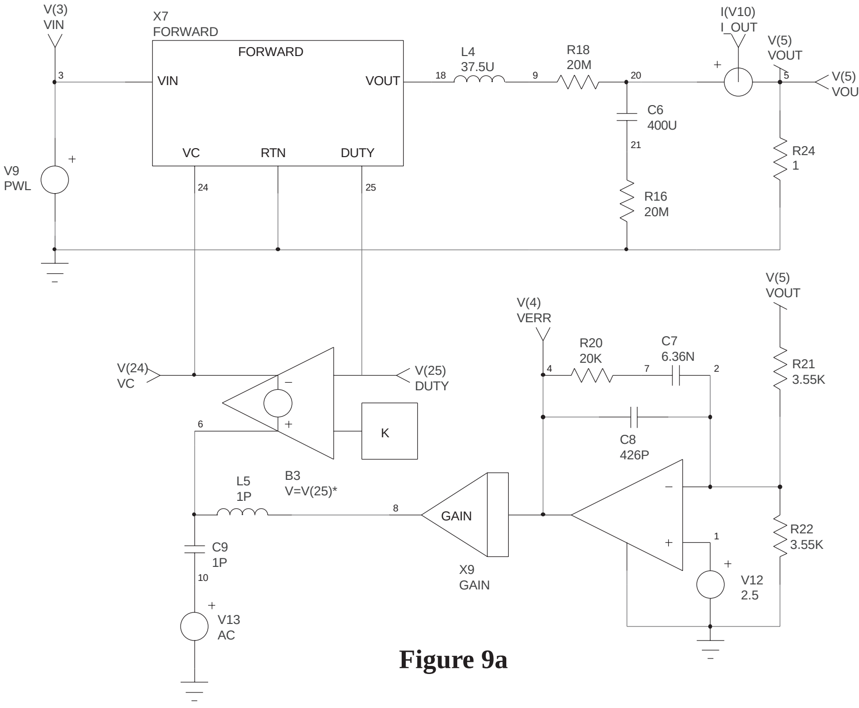


Figure 9a

For low compensation ramps, the phase of the line-to-output transfer function is negative. It is clearly depicted by the lower curves where the sudden stepped-up input voltage engenders a negative output transient. As more ramp is injected, the output step diminishes until null audio susceptibility is obtained, providing the designer with the value of the optimum ramp (K=0.135).

Conclusion

Both small and large-signal models should help the designer to better understand the intricacies of CMC converters stability. The new INTUSOFT's SMPS library includes averaged but also popular switched models (e.g UC384X family), giving the designer all the tools he needs to develop rugged designs from the theoretical study (average models) to realistic simulations (switched models).

References

1. V. BELLO, "Circuit Simulation of Switching Regulators Using HSPICE ", META-SOFTWARE
   (e-mail: DrVGB@AOL.COM)
2. R. D. MIDDLEBROOK and S. CUK, " A general Unified Approach to Modeling Switching Converter Power Stages ", IEEE PESC, 1976 Record, pp 18-34
3. Vatché VORPERIAN, "Simplified Analysis of PWM Converters Using The Model of The PWM Switch, Parts I (CCM) and II (DCM) ", Transactions on Aerospace and Electronics Systems, Vol. 26, N°3, May 1990
4. R. D. MIDDLBROOK, "Low-Entropy expressions: The key to Design-Oriented Analysis", IEEE Frontiers in Education, 21st Annual Conference, Purdue University, September 21-24, 1991;399-403
5. D. M. MITCHELL, "DC-DC Switching Regulators Analysis", distributed by e/j BLOOM Associates (71147.3274@compuserve.com).
6. R. B. RIDLEY, "A new small-signal model for current-mode control", PhD. dissertation, Virginia Polytechnic Institute and State University, 1990 (e-mail : RRIDLEY@AOL.COM)
   This document can be ordered from Ray RIDLEY's homepage : http://members.aol.com/ridleyeng/index.html
7. S. M. SANDLER, "SMPS Simulation With SPICE3", McGraw-Hill, ISBN 0-07-913227-8 (e-mail : SSANDLER@AENG.COM)

Write your own generic SPICE Power Supplies controller models

Christophe BASSO, e-mail: basso@esrf.fr

November 1996

Simulating the switching behavior of a Switch Mode Power Supply (SMPS) is not always an easy task. This is especially true if the designer wants to use an exact SPICE model for the Pulse Width Modulator (PWM) controller which will be used in the design. The PWM model may exist, but its syntax may be incompatible with your simulator. If the model has not been created, the debate over whether or not to do the simulation is closed! The solution that is proposed in this article consists of writing your own generic model of the PWM controller and then adapting its intrinsic parameters to comply with the real one you are using. Fixed frequency Current Control Mode (CCM) and Voltage Control Mode (VCM) models will be thoroughly covered in this article, as well as the model translation between different simulators.

The Berkeley B element, the standard behavior element

An efficient PWM model that is easy to model must include functions that are generic. For instance, it would not be clever to model an internal current comparator with the complete transistor architecture of a LM311 or a LM193. Fortunately, there is a simple in-line equation that can describe the perfect comparison function. By adding some passive elements to incorporate various effects (propagation delay, input offset voltage, etc.) we can achieve the functionality we need without sacrificing the simulation speed.

The non-linear controlled source, or B element, is part of Berkeley SPICE3, which was released to the public domain in 1986. Depending on the compatibility of your SPICE 3 simulator, the corresponding syntax may vary significantly. B elements can be linear or non-linear current or voltage sources. Some vendors have expanded the B element syntax to include BOOLEAN and IF-THEN-ELSE functions. For INTUSOFT's IsSpice4 (San Pedro, CA) and CADENCE's Analog WorkBench Spice Plus (San-Jose, CA), the writing of I or V math equations using B elements is the same because both are SPICE 3 compatible. For example, current/voltage generators whose current depends on various nodes can be expressed as:

```
B1 1 0 I = V(5,8)*100*V(10)/(V(8)+V(12))
         ; IsSpice or AWB current source
B2 2 0 V= V(9,8)*500*V(12)
         ; IsSpice or AWB voltage source
```

MICROSIM's PSpice (Irvine, CA) has departed from the Berkeley standard and uses a different syntax. PSpice modifies the standard calls for dependent voltage controlled sources (E and G elements). The equivalent PSpice examples are as follows:

```
G1 1 0 VALUE = { V(5,8)*100*V(10)/(V(8)+V(12)) }
         ; PSpice current source
E2 2 0 VALUE = { V(9,8)*500*V(12) }
         ; PSpice voltage source
```

Implement your logical operations

As stated in the above "paragraph, BOOLEAN and IF-THEN-ELSE expressions have become a part of most vendors' B elements. Their implementation also depends on the SPICE simulator. INTUSOFT exploits the concept of "binary" voltage, that is to say, a node value which is lower or higher than a user-defined threshold can be associated with 1 or 0. This threshold is driven by the keyword LTHRESH, whose value is set via a .OPTIONS line. Two other options, LONE and LZERO, will define the HI and LO electrical values which are delivered by a B element source when it is performing such BOOLEAN operations. A simple NAND equation between two nodes is simply expressed as:

```
BNAND 3 0 V= ~ ( V(1) & V(2) )  ;IsSpice complemented
         (~) AND operation between V(1) and V(2)
```

Because the other SPICE simulators do not directly support this syntax, it would be much easier to adopt a simpler expression in order to simplify any further translations. If we pass the logical thresholds directly into the equation, we obtain the following IsSpice IF-THEN-ELSE statement:

```
BNAND 3 0 V= (V(1)>800M) & (V(2)>800M)?   0V:   5V
```

In other words, IF V(1) is greater than 800mV AND V(2) is greater than 800mV, THEN V(3,0)=0V; ELSE V(3,0)=5V

Now the translation to AWB and PSpice syntax is more straightforward:

```
E_BNAND 3 0 VALUE = { IF ( (V(1)>800M) &
+ (V(2)>800M), 0V, 5V ) }            ; PSpice NAND gate
BNAND 3 0 V = IF ( (V(1)>800M) &&
+ (V(2)>800M), 0, 5 )                ; AWB NAND gate
```

Note that the AWB parser does NOT accept suffixes for the passed numerical values and the && symbol is doubled as in the C language to distinguish a logical AND from a binary AND. The diversity in implementing the B elements is only bound by the user's imagination. What we have shown above is only a small part of the possibilities offered by Behavioral Modeling via the B element.

If your simulator does not support B element modeling, the situation becomes complex. Some examples on how to model the logical functions with SPICE2 syntax are given at the end of this article.

Here's an application example, a simple voltage limiter which limits the differential voltage of nodes 1 and 2 between 100mV and 1V:

```
E1 3 0 TABLE {V(1)-V(2)}  100M,100M 1,1  ; PSpice
B1 3 0 V = IF ( V(1,2) < 1, IF (V(1,2) < 100M, 100M,
+ V(1,2)),1)                  ; AWB (No suffixes!)
B1 3 0 V = V(1,2) < 100MV ? 100M : V(1,2) > 1 ?
+ 1 : V(1,2)                  ; IsSpice
```

In other words,
IF V(1,2) is less than 100mV, THEN V(3,0)=100mV; ELSE IF V(1,2) is greater than 1V, THEN V(3,0)=1V, ELSE V(3,0)=V(1,2)

B elements switch in essentially a zero time span. This characteristic may create convergence problems on transitions associated with these perfect sources. We recommend that you tailor the output switching times in a more realistic manner. A simple RC network is suitable for this purpose. A perfect comparator which accounts for these conditions is given below. We have included it in a SUBCIRCUIT in order to highlight the philosophy of constructing your own models:

```
.SUBCKT COMP          1     2     3
*                     (+)   (-)   OUT
E1 4 0 VALUE = { IF ( V(1) > V(2), 5V, 0 ) }          ; PSpice syntax;
IsSpice Syntax: B1 4 0 V=V(1) > V(2) ? 5 : 0
RD 4 3 100                                            ; RC network
CD 3 0 100P                                   ; to slow down transitions
.ENDS COMP
```

Now that we have reviewed the basics of generating in-line equations, let's dip into the nitty-gritty of a constant frequency CCM PWM controller.

Current mode controllers, a well known architecture

Figure 1, the internal circuitry of a generic single output CCM PWM controller.

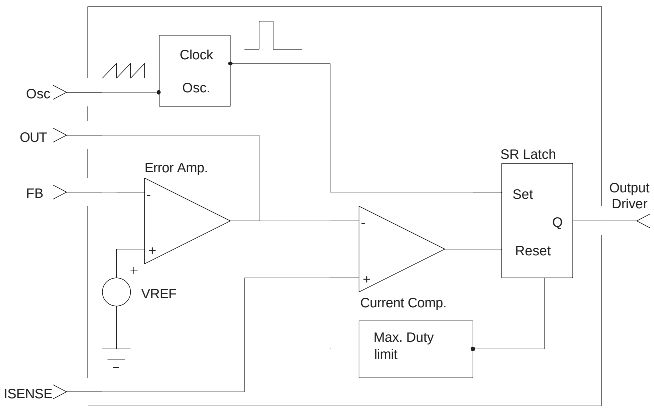


Figure 1

The modelling of such a block consists of: a) defining <u>and</u> testing each subcircuit <u>individually</u>, and
b) assembling all of these domino-like circuits to form the complete generic model. All individual blocks should be tested before they are used within larger models. Below are some recommendations that will ease your task:

- Draw the symbol of your generic model with your favorite schematic capture tool. Once this is done, you won't have to worry about incorrect pin connections (as you would if you were creating a SPICE netlist with a text editor). Internal subcircuit testing is simplified since you may then access the connection pins directly in the schematic model, and the pin passing process (from schematic to netlist) is performed automatically.

- Place comments on every pertinent line, either with a "*" in column one (for a complete line), or a ";" immediately preceding a comment within a line. Also, use different commented header names for each section of code within the listing.

- Use descriptive names for the components you assemble in the subcircuit netlist, i.e. VCLOCK for a clock source, RDUM for a dummy load, etc.

- Use subcircuits whenever a function is called more than once. Even if the function is only called once, you can create a subcircuit and therefore simplify the netlist. This will also facilitate the writing of new models because the .SUBCKT functions are easily pasted into the netlist. Also, if required, the conversion process to another platform will be greatly simplified.

- Use realistic parameter values for primitive SPICE components such as the diode (D). These models may generate convergence problems since some of the default parameters are set to zero. For example,
  .MODEL DMOD D (TT=1N CJO=10P RS=100M)

- Use a main subcircuit pin number of up to 10 or 20 and use incremental decimal digit notation as you change the internal function. This is especially recommended for complex models in which the parent subcircuit may be large. Below is an arbitrary example, where nodes 7 through 19 are preserved in order to output test signals or add additional pins:

```
.SUBCKT    EXAMPLE          1  2  3  4  5  6
**** MAIN CLOCK ****
VCLOCK 20 21 PULSE          ; Main clock
ICHARGE  22 24 10MA         ; Current charge of capacitor C1
**** TRIGGER  ****
RTHRES 30 33 10K            ; Threshold high
CDEL 33 38 10NF             ; Propagation delay
**** COMPARATOR ****
RINP 40 42 10K              ; Input resistor
RFEED 45 49 120K            ; Feedback resistor
.ENDS EXAMPLE
```

Writing the model step by step

Let’s start with the synchronization signals, Clock, Osc. and Max. duty cycle. The first one will initiate the on-time of the external switch by triggering the flip-flop latch. Its frequency sets the functioning period of the entire PWM circuit, and will therefore require user input (PERIOD). Osc. is provided for ramp compensation purposes. It delivers a signal which is equivalent to that which was delivered by the classical linear charge of the external oscillator RC network. Using the oscillator ramp is a possible option for ramp compensation, but there are others such as charging a RC network from the MOSFET driver output. If the capacitor voltage is kept at around 1 volt, it is possible to obtain a very low impedance linear ramp, without adversely affecting the PWM oscillator. Osc. will have the same period as Clock, but the user will select its peak amplitude (RAMP). Once the Clock pulse is issued, the Max. duty cycle must reset the latch after a specific period of time. This time is user-defined (DUTYMAX) and selects the maximum allowable duty cycle.

Parameters for IsSpice and PSpice are quite similar in format: the parameter, or any equations between parameters, is enclosed by curly braces. Our three generators are listed below. Arbitrary node numbers are used to simplify their understanding:

```
VCLK 1 0 PULSE 0 5 0 1N 1N 10N {PERIOD}
VRAMP 2 0 PULSE 0 {RAMP} 0 {PERIOD-2N} 1N 1N
{PERIOD}
VDUTY 3 0 PULSE 0 5 {PERIOD*DUTYMAX} 1N 1N
{(PERIOD-PERIOD*DUTYMAX)-2N} {PERIOD}
```

A quick simulation of this set of equations appears in figure 2, where a maximum duty cycle of 0.5 was selected.

The current comparator requires a simple equation followed by a RC network which slows down its transitions. The model is the same as the one given in the example above.

The SR latch may be defined in many ways. We do not recommend the use of a proprietary flip-flop model. You can draw a classical RS flip-flop and add a couple of inverters in order to generate the required signals. Figure 3 shows the electrical circuit.

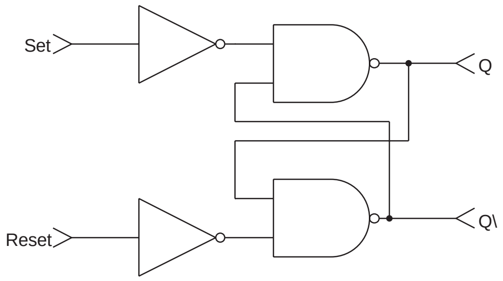


Figure 3

The subcircuit will appear as shown below, according to common AWB and IsSpice syntax rules:

```
.SUBCKT FFLOP 6  8  2  1
*                S  R  Q  Q\
BQB  10 0 V=(V(8)<800M) & (V(2)>800M) ? 0 : 5   ; one input
                                                 ; inverted two input NAND
BQ   20 0 V=(V(6)<800M) & (V(1)>800M) ? 0 : 5
RD1  10 1 100                                    ; delay elements
CD1  1 0 10p IC=5
RD2  20 2 100
CD2  2 0 10p IC=0
.ENDS FFLOP
```

The “IC” statements are mandatory in order to avoid conflicts when SPICE computes the DC operating point. You will then add the keyword “UIC” (Use Initial Conditions) at the end of the .TRAN statement.

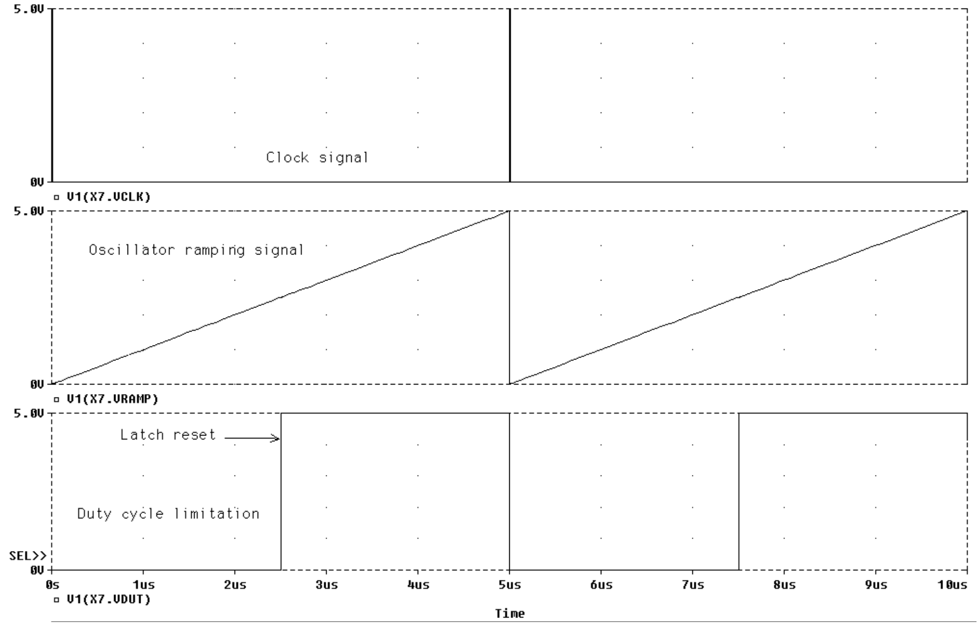


Figure 2

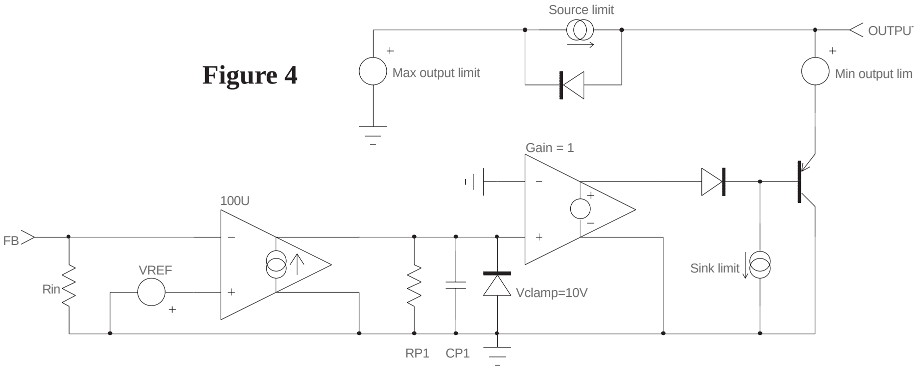
A simplified error amplifier macro-model

There are an infinite number of ways to realize the error amplifier model. However, keep in mind that the simplest model yields the fastest simulation runs. We will use basic building blocks to create a model with the following parameters:

- DC open-loop gain: 90dB, or 31622 {GAIN}
- First pole: 30Hz {POLE}
- Maximum output voltage: {VHIGH}
- Minimum output voltage: {VLOW}
- Maximum sink current: {ISINK}
- Maximum source current: {ISOURCE}

The last two parameters correspond to the output current capability of the op-amp. Modeling its output current capability. instead of using a simple resistor R_{out} in series with the final voltage source, yields more accurate results. This is because once the loop is closed, the dynamic output impedance of the amplifier is close to zero since the open-loop gain T_{OL} is large: $R_{out} = R_{open-loop} / (1+T_{OL})$. However, this expression is only valid within the limit of the current capability of the amplifier. If the amplifier current exceeds this limit, the component is unable to maintain the proper voltage level, and the output plummets to zero (or rises, if the sink current limit is exceeded). A simple resistor cannot properly model this effect. The electrical schematics is given below:

Figure 4



Voltage and current offset effects are not modeled, but can be easily added. Offset currents are important, especially when high value feedback networks are used (in the presence of high voltage regulated output, for instance). Output clipping of the voltage controlled sources is always a problem. It is alleviated by using a voltage controlled current source whose output is clipped by a diode and then buffered by a unity gain voltage controlled source. Sink and source limits are associated with the output transistor; the sink limit is dependent of its static gain (default BF=100). The final netlist is as follows:

```
.SUBCKT ERRAMP 20    8    3      21
*                    +    -    OUT    GND
RIN 20 8 8MEG
CP1 11 21 {1/(6.28*(GAIN/100U)*POLE)} ; pole calculation
E1 5 21 11 21 1
R9 5 2 5
D14 2 13 DMOD
IS 13 21 {ISINK/100}          ; sink limit, with BF=100
Q1 21 13 16 QPMODI
ISRC 7 3 {ISOURCE}            ; source limit
D12 3 7 DMOD
D15 21 11 DCLAMP
G1 21 11 20 8 100U
V1 7 21 {VHIGH-0.6V}          ; max output clipping
V4 3 16 {VLOW-38MV}           ; min output clipping
RP1 11 21 {GAIN/100U}         ; open loop gain calculation
.MODEL QPMOD PNP
.MODEL DCLAMP D (RS=10 BV=10 IBV=0.01)
.MODEL DMOD D (TT=1N CJO=10P)
.ENDS ERRAMP
```

The test of the amplifier confirmed the presence of the pole and the current output limits.

Test of the complete current mode model

Now that all of the individual elements have been defined and tested, it is time to place them within the final subcircuit model, PWMCM. The output driver model is simplified, and converts the latch levels to user-defined voltages which are associated with a resistor:

```
E_BOUT 15 0 VALUE = { IF ( V(10) > 3.5, {VOUTHI}, {VOUTLO} ) }
                                        ; node 10 is the latch output
ROUT 15 1 {ROUT}
```

For editing convenience, the final PWMCM model will not be printed in this article, but may be downloaded from our BBS or our Internet Web site. The complete example .CIR files are available there also. They are available in PSpice, IsSpice and SPICE2 formats.

The test circuit is a buck converter which delivers 5V at 10A. All of the elements have been calculated using the new release of POWER 4-5-6 [2], which was developed by RIDLEY Engineering (http://members.aol.com/ridleyeng/index.html). Figure 5a depicts this switch-mode converter:

The ramp compensation is accomplished by summing a fraction of the oscillator sawtooth and the current sense information. It has several beneficial effects that we will discuss later on. This circuit has been simulated in 25s upon a P120 machine for a 200µs transient run. With this simulation speed, output response to load or input step can be accomplished rather quickly. Figure 5b shows the start-up conditions before the output voltage has reached its final value. A switching frequency of 200kHz has been selected.

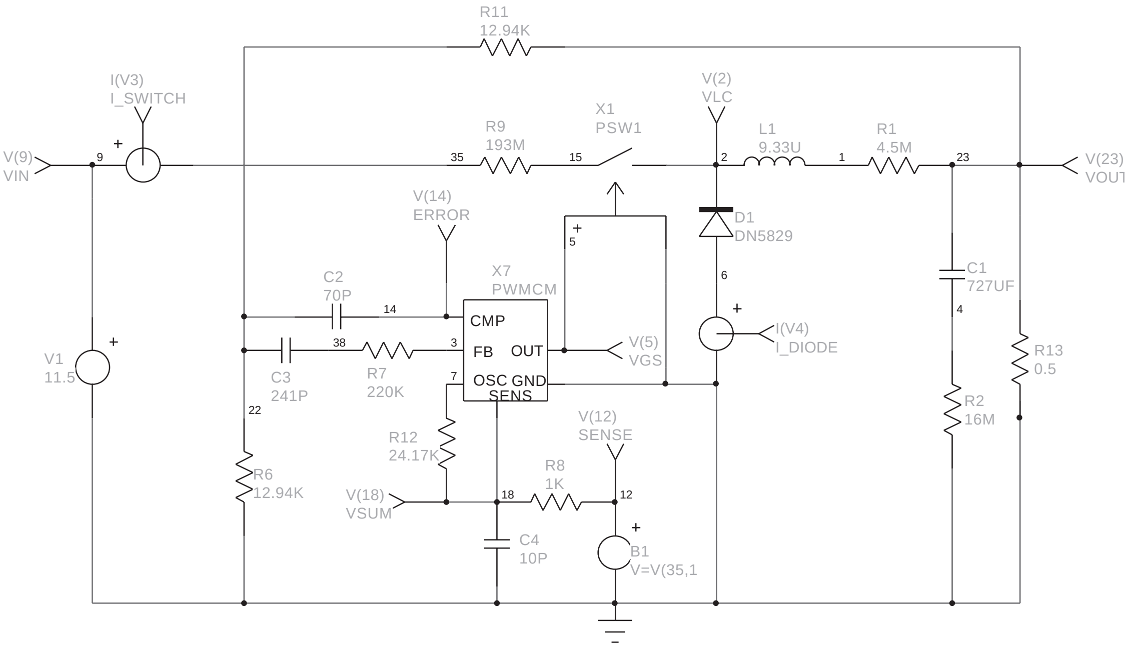


Figure 5a

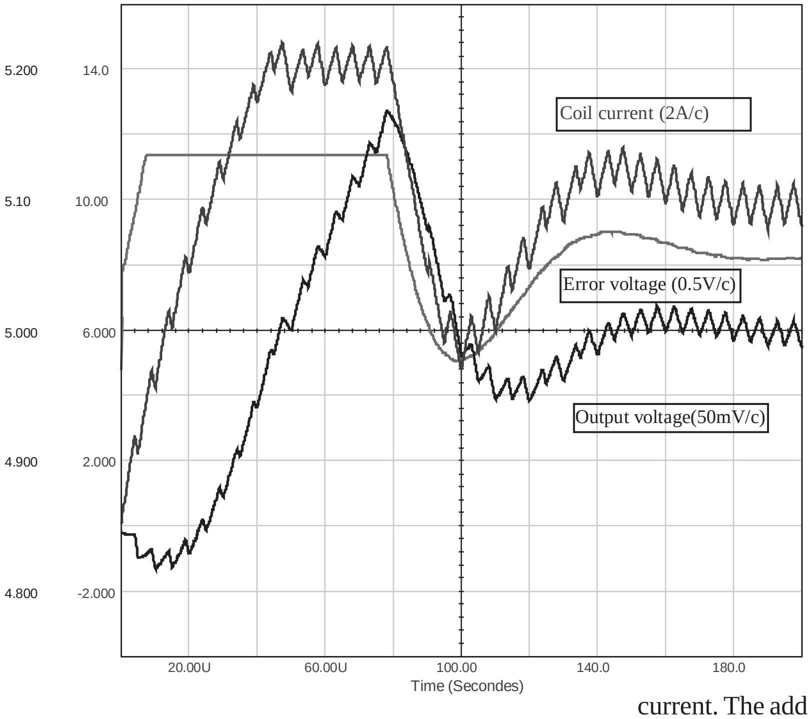


Figure 5b

Current mode instabilities

The control-to-output transfer function (V_o/V_c) of a continuous current mode control converter is a three pole system, as Raymond RIDLEY demonstrated in 1990 [1]: one low frequency pole, ω_p, and two poles which are located at Fs/2. These poles move in relation to the duty cycle and the external ramp. The two high frequency poles present a Q that depends on the compensating ramp and the duty cycle. RIDLEY demonstrated that the Q becomes infinite at D=0.5 with no external ramp, which confirms the inherent instability of a current mode SMPS which has a duty cycle greater than 0.5. Q and ω_p, which are part of the V_o/V_c transfer function, are expressed as follows:

$$Q = \frac{1}{\pi \cdot (mc \cdot D' - 0.5)}$$

$$\omega_p = \frac{1}{CR} + \frac{Ts}{LC} \cdot (mc \cdot D' - 0.5)$$

where $m_c = 1 + S_e / S_n$. S_e is the external ramp slope, S_n is the inductor on-time slope. D' = 1 - D

The low frequency pole, $\omega_{p,}$ moves to higher frequencies as additional compensation ramp is injected. The addition of the external ramp will also damp the Q factor of the double Fs/2 poles, which are the result of the current sampling action, in continuous inductor current. The addition of more ramp will split the double pole into two distinct poles. The first one will move towards lower frequencies until it joins and combines with the first low frequency pole at ω_p. At this point, the converter behaves as if it is operating in voltage mode.

Think of the current loop as an RLC network which is tuned to Fs/2. If we excite this network with a transient current step, it will ring like an RLC response [3], whose damping depends on the duty cycle and on the amount, if any, of ramp compensation. By increasing the duty cycle, we will raise the DC gain of the current loop until the phase margin at Fs/2 vanishes and makes the system unstable. When the duty cycle is greater than 0.5, the current gain curve crosses the 0dB point at Fs/2, and because of the abrupt drop in phase at this point, the converter oscillates. This sharp drop in phase at Fs/2 is created by the double RHP zeros which are engendered by the sampling action in the current information. These double RHP zeros which appear in the current gain are transformed into double poles when one calculates the V_o/V_c transfer function. The Q of these poles is inversely proportional to the phase margin in the current gain at Fs/2. Compensating the system with an external ramp will oppose the duty cycle action by lowering the DC gain and increasing the phase margin at Fs/2, which will damp the high Q poles in the V_o/V_c transfer function.

To highlight this phenomenon, let's open the voltage loop and place a fixed DC source at the right tail of R11 (node 23) in Figure 5a. R12 is elevated to 1MEG in order to suppress any ramp compensation. If we abruptly change the input voltage from 18V to 12.5V, the Fs/2 component appears (100kHz) and is damped after several switching periods, since the duty cycle is less than 0.5. Further stressing of the output would lengthen the damping time or produce a steady-state oscillation. The result is depicted in **Figure 5c**, where a filter has removed the main switching component from the coil current to allow the Fs/2 signal to be properly established.

If this network is now closed within a high gain outer loop, any current perturbation will make the entire system oscillate at Fs/2, even if the loop gain has a good phase margin at the 0dB crossover frequency. This so-called gain peaking is attributed to the action of the high-Q poles, which push the gain above the 0dB line at Fs/2, and produce an abrupt drop in phase at this point. If the duty cycle is smaller than 0.5, the oscillations will naturally cease, but if the duty cycle is greater, the oscillation will remain, as **Figure 5d** demonstrates with the FFT of the error amplifier voltage (Vin=11.5V). In conclusion, providing an external ramp is a wise solution, even if your SMPS duty cycle will be limited to 0.5: the Fs/2 Q will be reduced, thereby preventing oscillations.

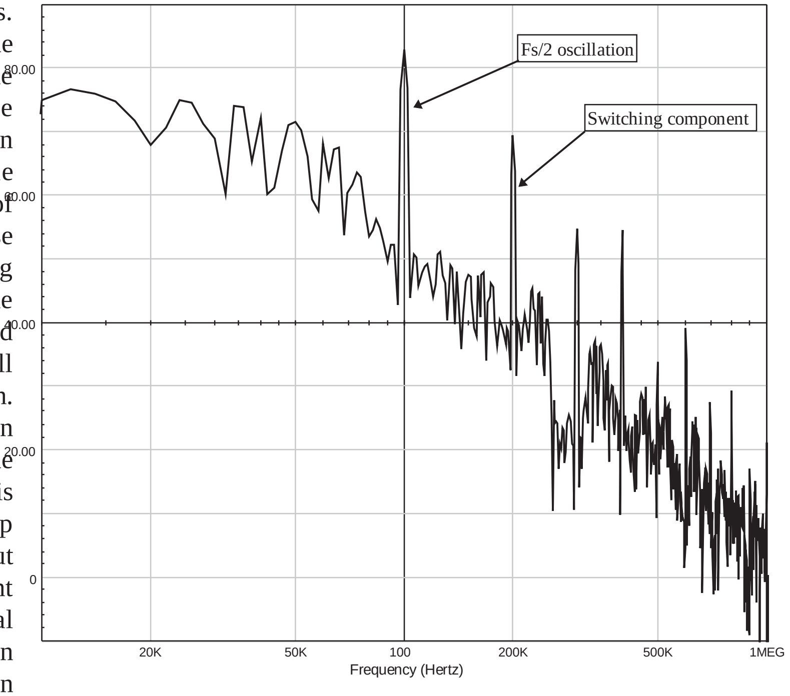


Figure 5d

The audio susceptibility is also affected by slope compensation. RIDLEY showed in his work that an external ramp whose slope is equal to 50% of the inductor downslope could nullify the audio susceptibility. As previously stated, excessive ramp compensation makes the converter behave as if it is in voltage mode, which degrades the audio susceptibility. Also, if minimal compensation or no ramp is provided, good input voltage rejection is achieved, the phase of the resulting audio susceptibility is <u>negative</u>; and an increase in input voltage will cause the output voltage to decrease. **Figure 5d** illustrates these behaviors when the input of the buck converter is stressed by a 6V variation. The upper curve depicts the output voltage for a critical ramp compensation. The voltage difference in the output envelope is only 10mV for a 6V input step, which leads to a (theoretical) ΔVout/ΔVin of -55dB. The middle curve shows how the response starts to degrade as additional ramp is

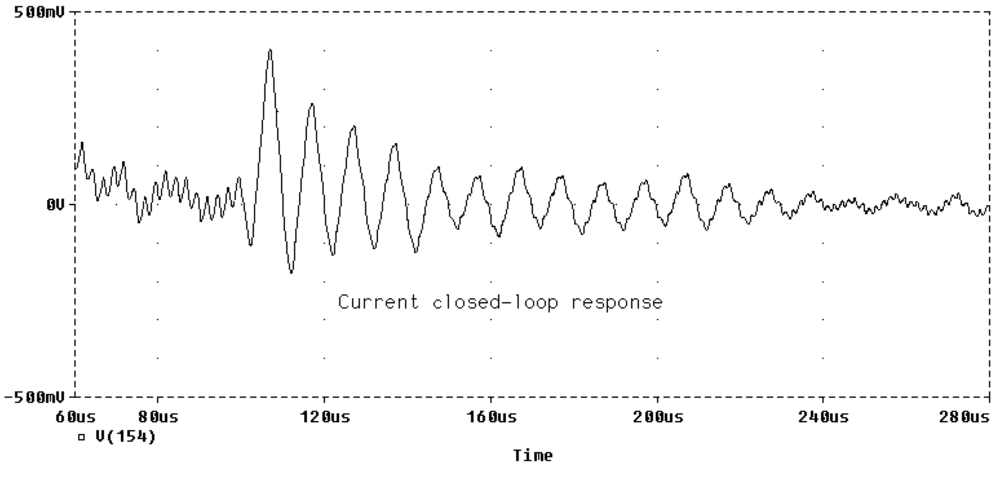
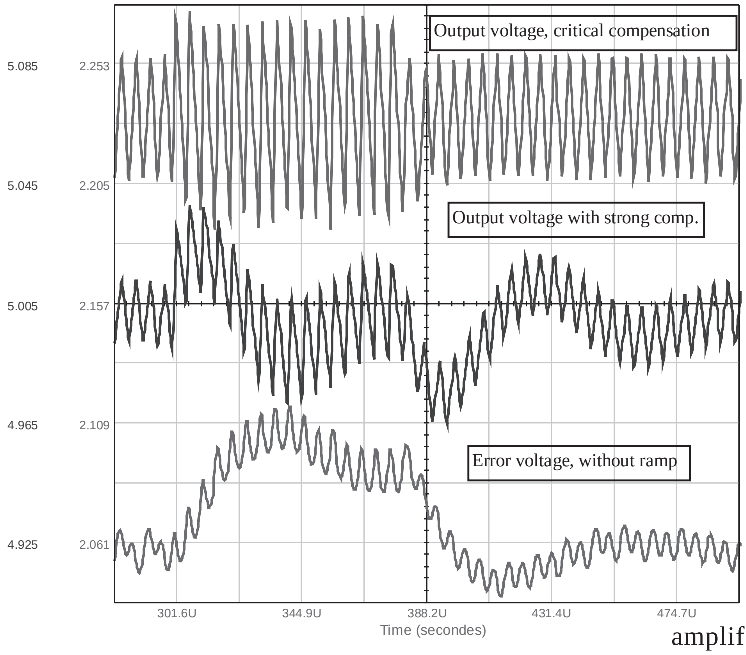


Figure 5c

added. The lower curve represents the error amplifier response when a slight ramp compensation is added. The decrease in the output voltage is clearly revealed by the rise in the error voltage.



A close look at the error voltage response time leads to the closed-loop bandwidth of the SMPS. The measured rise time, t_r, is roughly 22µs, which gives a bandwidth of: BW=1/π*t_r = 14.5kHz, which corroborates our initial design goal value which was passed to POWER 4-5-6.

The voltage mode model, PWMVM

The voltage mode generic controller will follow the description given in **Figure 6**.

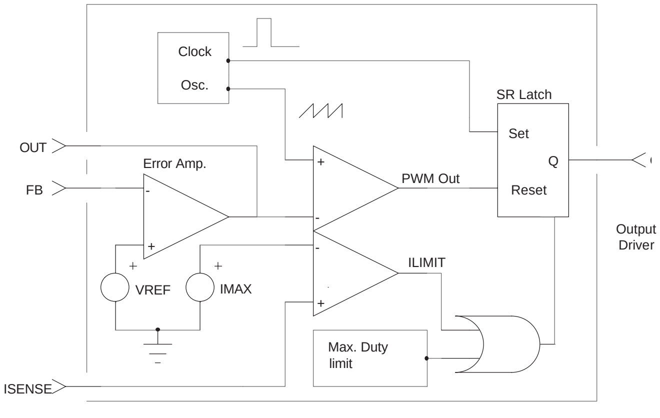


Figure 6

The architecture allows the inclusion of a current limitation circuit to reduce the on-time of the external power switch when its peak current exceeds a user-defined limit. This option is strongly recommended to make a rugged and reliable SMPS design that can safely handle input or output overloads. By simply connecting the ISENSE input to ground, you disable this option.

In this model, the duty cycle is no longer controlled by the current information (except in limitation mode). It is controlled by the PWM modulator, which compares the error voltage with the reference sawtooth. The error amplifier output swing will then define the duty cycle limits. Since this output swing is user dependent, the model will calculate the peak and valley voltages of the reference sawtooth such that the chosen duty cycle boundaries are not violated. **Figure 7** depicts the well-known naturally sampled PWM modulator.

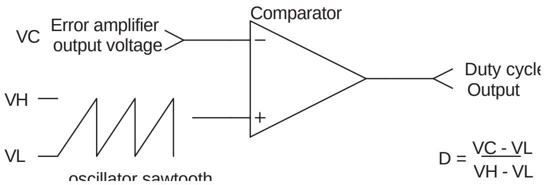


Figure 7

Since you will provide the main subcircuit with the duty cycle limits and the error amplifier output swing, it is possible to calculate the corresponding sawtooth peak values, V_{valley} and V_{peak}. In MICROSIM's Pspice or INTUSOFT's IsSpice, it is easy to define some particular variables with a .PARAM statement. The reading of the remaining lines in the netlist is then considerably simplified:

```
.PARAM VP = {(VLOW * DUTYMAX - VHIGH *
+   DUTYMIN + VHIGH - VLOW) / (DUTYMAX -
+   DUTYMIN)}
.PARAM VV = {(VLOW - DUTYMIN * VP) / (1 -
+                    DUTYMIN)}
```

The sawtooth source then becomes:

```
VRAMP 1 0  PULSE {VV} {VP} 0 {PERIOD-2N} 1N
+ 1N {PERIOD}
```

The OR gate which routes the reset signal to the latch from the PWM or the limitation comparator requires a simple in-line equation, followed by the classical delay network:

```
.SUBCKT OR2 1 2 3
E_B1 4 0 VALUE = { IF ( (V(1)>800M) | (V(2)>800M), 5V, 0 ) }
RD 4 3 100
CD 3 0 10P
.ENDS OR2
```

Since the remaining elements have already been defined (comparators, error amplifier etc.), we are all set. As previously stated, the final PWMVM model will not be printed in this article, but can be downloaded from our BBS or our Internet Web site. The test circuit of **Figure 8a** is a

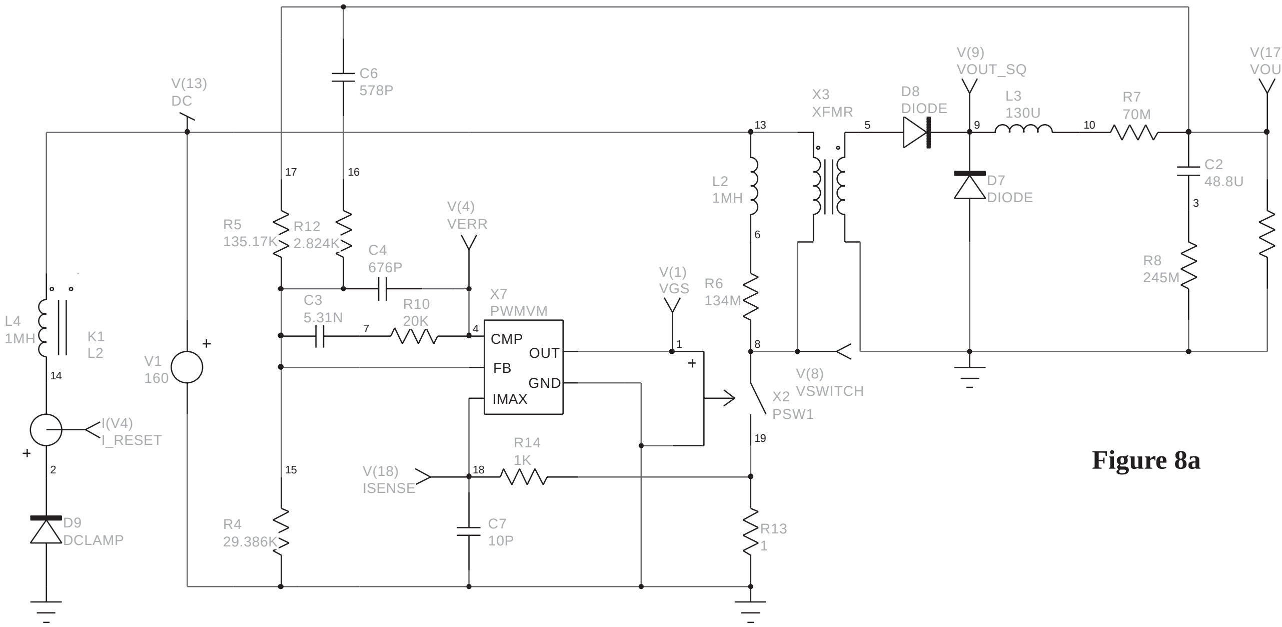


Figure 8a

forward converter which delivers 28V@4A from a 160V input source.

The switching frequency is set at 200kHz, with a maximum duty cycle of 0.45 because of the forward structure. **Figure 8b** depicts the curves which are obtained at start-up. The power switch is modeled with a smooth transition element, as provided by MICROSIM and INTUSOFT. The error amplifier is pushed to its upper limit, and needs some time to recover this transient situation. This behavior is typical of the adopted compensation scheme for a voltage mode converter which is operating in continuous mode.

Modelling with SPICE2

If you own a SPICE2 compatible simulator, you simply cannot use the B element syntax. To overcome this limitation, some equivalent (but more time consuming) circuits can be constructed. The first generic function which is called in our models is the perfect comparator. **Figure 9a** shows one solution. The unlabeled resistors provide a DC path to ground (10MEG).

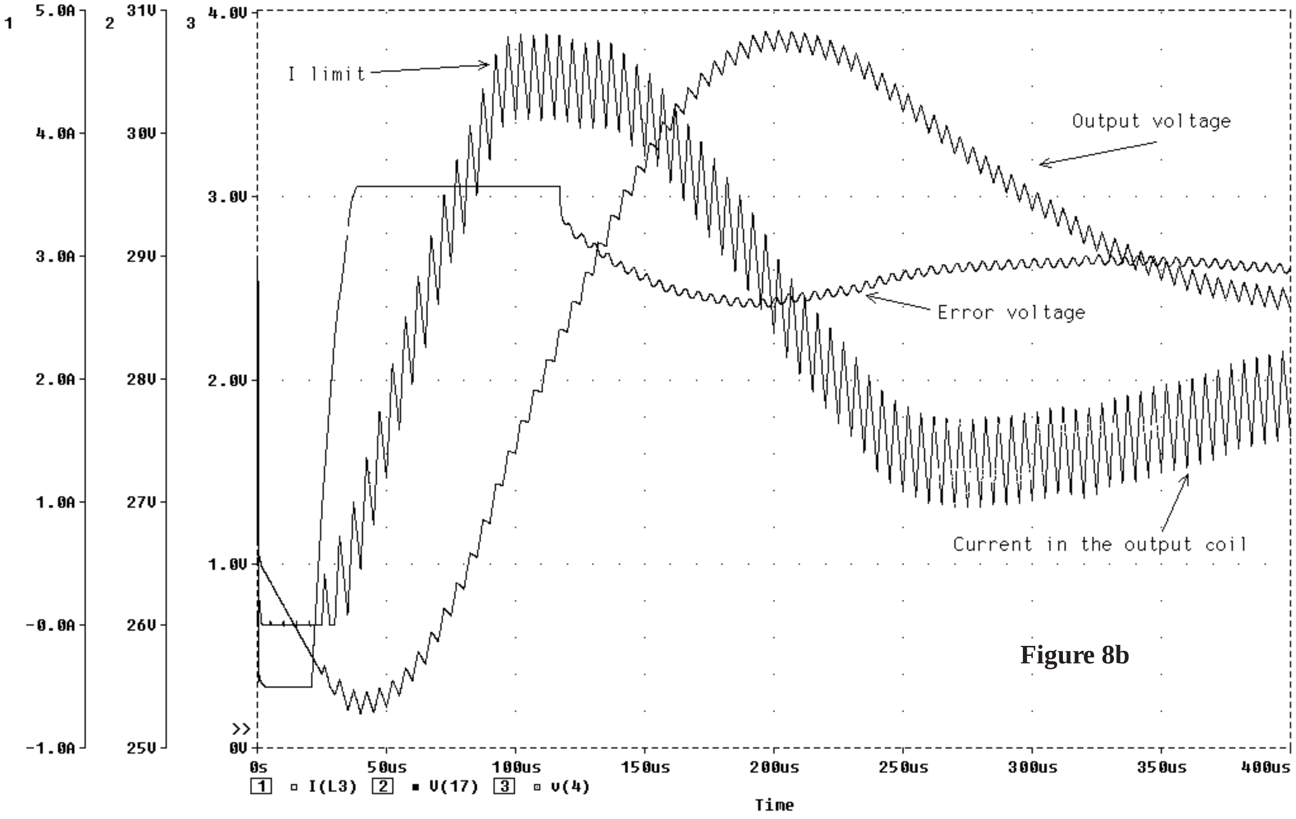


Figure 8b

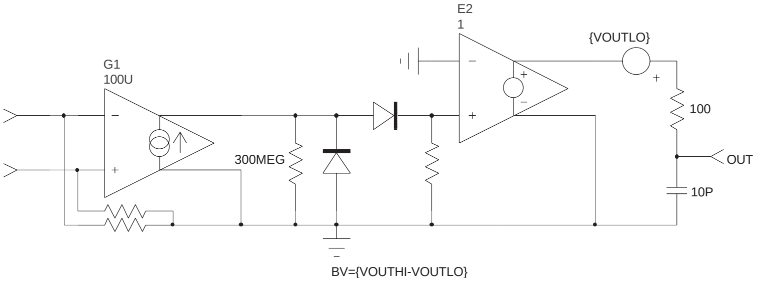


Figure 9a

The logical functions are less obvious, at least if you want to build something easily. The ideal gates in **Figure 9b** simulate quickly and converge well. They use the ideal SPICE2 voltage controlled switch or the PSpice/IsSpice smooth transition switch.

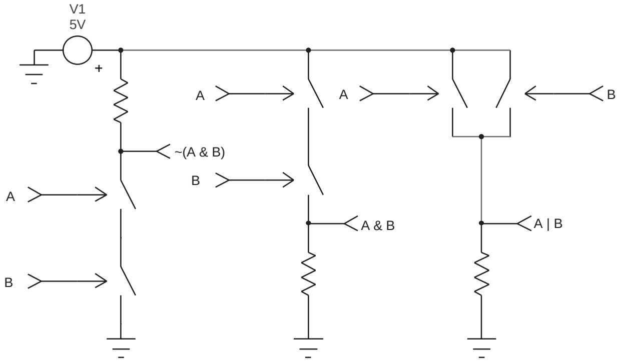


Figure 9b

The Flip-Flop is also translated to SPICE2 syntax, as the following lines explain.

```
**** FFLOP ****                  **** 2 INPUT NAND ****
.SUBCKT FFLOP 6 8 2 1            .SUBCKT NAND 1 2 3
*          S R Q Q\              RDUM1 1 0 10MEG
RDUM1 6 0 10MEG                  RDUM1 2 0 10MEG
RDUM2 8 0 10MEG                  S1 3 5 1 0 SMOD
XINVS 6 6 10 NAND                S2 5 0 2 0 SMOD
XINVR 8 8 11 NAND                RL 3 4 100
XNAQB 11 2 10 NAND               CD 3 0 10P
XNAQ 10 1 20 NAND                VCC 4 0 5V
RD1   10 1 100                   .MODEL SMOD VSWITCH (RON=1
CD1   1 0 10P IC=5               +ROFF=1Meg VON=3 VOFF=100M)
RD2   20 2 100                   .ENDS NAND
CD2   2 0 10P IC=0
.ENDS FFLOP
****
```

The simulation of the buck converter circuit of figure 5a using SPICE2 syntax took 71seconds on our P120 machine, giving an increase in simulation time of 184%.

If you are interested in the SPICE2/SPICE3 macro modeling technique and the SPICE engine in general, consult "The SPICE Book", written by Andrei VLADIMIRESCU [4] or "SMPS Simulation with SPICE3", written by Steve SANDLER [5].

Conclusion

This article describes some guidelines which will help you in the process of writing your own generic models for the platform of your choice. The two models, PWMCM and PWMVM, simulate quickly and converge very well, therefore allowing the designer to readily implement the average model results to see if they are verified by the "real world" switched elements.

References

1. R. B. RIDLEY, "A new small-signal model for current-mode control", PhD. dissertation, Virginia Polytechnic Institute and State University, 1990
   This document can be ordered from the VPEC Web site: http://www.vpec.vt.edu/cgi-bin/home.html
2. POWER 4-5-6 Technical manual, RIDLEY ENGINEERING (Battle-Creek, MI)
3. B. HOLLAND, "Modelling, Analysis and Compensation of the Current Mode Converter", Powercon 11, 1984 Record, Paper H-2
4. A. VLADIMIRESCU, "The SPICE Book", John Wiley & Sons, ISBN 0-471-60926-9
5. S. SANDLER, "SMPS Simulation With SPICE3", McGraw-Hill, ISBN 0-07-913227-8

Keep your Switch Mode Supply stable with a Critical-Mode Controller

MOTOROLA Semiconductors, Christophe BASSO, Toulouse, France, e-mail: basso@esrf.fr

November 1997

Switch Mode Power Supplies (SMPS) can operate in two different conduction modes, each one depicting the level of the current circulating in the power choke when the power switch is turned on. As will be shown, the properties of two black boxes delivering the same power levels but working in different conduction modes, will change dramatically in DC and AC conditions. The stress upon the power elements they are made of will also be affected. This article explains why the vast majority of low-power FLYBACK SMPS (off-line cellular battery chargers, VCRs etc.) operate in the discontinuous area and present a new integrated solution especially dedicated to these particular converters.

Defining the mode

Figure 1a and **1b** show the general shape of a current flowing through the converter's coil during a few cycles. In the picture, the current ramps up when the switch is closed (ON time) building magnetic field in the inductor's core. When the switch opens (OFF time), the magnetic field collapses and, according to LENZ's law, the voltage across the inductance reverses. In that case, the current has to find some way to continue its flow and start its decrease (in the output network for a FLYBACK, through the freewheel diode in a BUCK etc.).

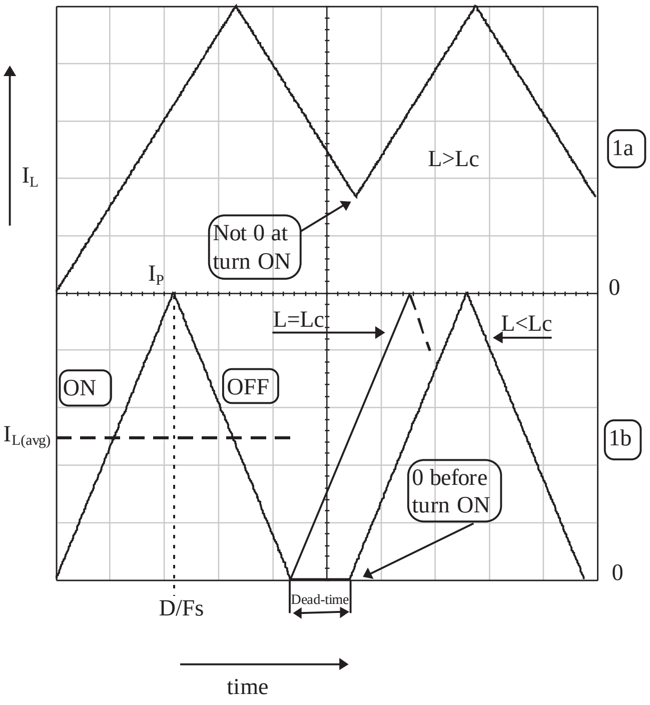
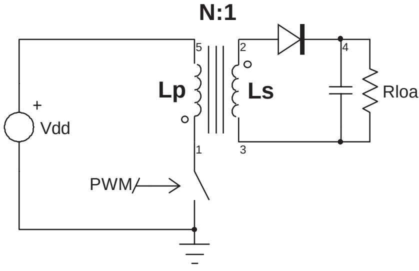


Figure 1

If the switch is switched ON again during the ramp down cycle, before the current reaches zero (figure 1a), we talk about Continuous Conduction Mode (CCM). Now, if the energy storage capability of the coil is such that its current dries out to zero during OFF time, the supply is said to operate in Discontinuous Conduction Mode (DCM). The amount of dead-time where the current stays at a null level defines how strongly the supply operates in DCM. If the current through the coil reaches zero and the switch turns ON immediately (no dead-time), the converter operates in Critical Conduction Mode.

Where is the boundary?

There are three ways you can think of the boundary between the modes. One is about the critical value of the inductance, L$_C$, for which the supply will work in either CCM or DCM given a fixed nominal load. The second deals with a known inductance L. What level of load, R$_C$, will push my supply into CCM? Or what minimum load my SMPS should see before entering DCM? The third one uses fixed values of the above elements but adjusts the operating frequency, F$_C$, to stay in critical conduction. These questions can be answered after a few lines of algebra corresponding to **figure 2**'s example, a FLYBACK converter:



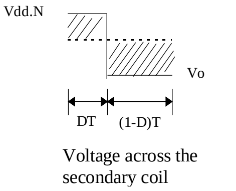


Figure 2a **Figure 2b**

To help determine some key characteristics of this converter, we will refer to the following statements:

- The avg. inductor voltage per cycle should be null (1)
- From figure 1b, when L=L$_C$, I$_{L(avg)}$ = 2xIp (2)
- An 100% efficiency leads to Pin=Pout (3)

The DC voltage transfer ratio in CCM is first determined using statement (1), thus equating **figure 2b**'s areas: $Vdd \times N \times D = Vo \times (1 - D)$.

After factorization, it comes $\frac{Vout}{Vin} = \frac{D}{(1-D)} \times N$ (4).

As we can see from **figure 1b**, the flux stored in the coil during the ON time is down to zero right at the beginning of the next cycle when the inductance equals its critical value (L=L$_C$). Mathematically this can be expressed by integrating the formula:

- $V_L \times dt = L \times dI_L$ thus, $\int_0^{\frac{D}{Fs}} V_L.dt = L_C.\int_0^{I_P} dI_L$.

$\Rightarrow \dfrac{Vin \times D}{Fs} = Lc \times Ip = 2 \times I_{L(avg)} \times L_C$, from (2).

- From (3), $Vin \times I_{L(avg)} = Io \times (Vin \times N + Vout)$,

or $I_{L(avg)} = Io \times (N + \dfrac{Vout}{Vin})$.

- By definition, $Io = \dfrac{Vout}{R}$ and

$Vout = Vin \times N \times \dfrac{D}{1-D}$ from (4).

If we introduce these elements in the above equations, we can solve for the critical values of R_C, L_C and F_C:

$$R_C = \frac{2 \times L_C \times F_S \times N^2}{(1-D)^2} \qquad L_C = \frac{R_C \times (1-D)^2}{2 \times F_S \times N^2}$$

$$F_C = \frac{R_C \times (1-D)^2}{2 \times L_C \times N^2}$$

Filling-in the bucket

The FLYBACK converter, as with the BOOST and BUCK-BOOST structures, has an operating mode comparable to someone filling a bucket (coil) with water and flushing it into a water tank (capacitor). The bucket is first presented to the spring (ON time) until its inner level reaches a defined limit. Then the bucket is removed from the spring (OFF time) and flushed into a water tank that supplies a fire engine (load). The bucket can be totally emptied before refilling (DCM) or some water can remain before the user presents it back to the spring (CCM). Let's suppose that the man is experimented and he ensures that the recurrence period (ON+OFF time) is constant. The end-user is a fireman who closes the feedback loop via his voice, shouting for more or less flow for the tank. Now, if the flames suddenly get bigger, the fireman will require more power from its engine and will ask the bucket man to provide the tank with a higher flow. In other words, the bucket operator will fill his container longer (ON time increases). BUT, since by experience he keeps his working period constant, the time he will spend in flushing into the tank will naturally diminish (OFF time decreases), so will the amount of water poured. The fire engine will run out of power, making the fireman shout louder for more water, extending the filling time etc. The loop oscillates! This behavior is typical for converters in which the energy transfer is not direct (unlike the BUCK derived families) and severely affects the overall dynamic performances. In time domain, a large step load increase requires a corresponding percentage rise of the inductor current. This necessitates a temporary duty-cycle augmentation which (with only two operational states) causes the diode conduction time to diminish. Therefore, it implies a decrease in the average diode current at first, rather than an increase as desired. When heavily into the continuous mode and if the inductor current rate is small compared to the current level, it can take many cycles for the inductor current to reach the new value. During this time, the output current is actually reduced because the diode conduction time (TOFF) has been decreased, even if the peak diode current is rising. In DCM, by definition, a third state is present whether neither the diode or the switch conduct and the inductor current is null. This "idle time" allows the switch duty cycle to lengthen in presence of a step load increase without lowering the diode conduction time. In fact, it is possible for the DCM circuit to adapt perfectly to a step load change of any magnitude in the very first switching period, with the switch conduction time, the peak current, and the diode conduction time all increasing at once to the values that will be maintained forevermore at the new load current.

The extra delay is mathematically described by a Right Half-Plane Zero (RHPZ) in the transfer function ($Av = \dfrac{(1 - s_{z1})\times\ldots}{\ldots}$) and forces the designer to roll-off the loop gain at a point where the phase margin is still secure. Actually, a classical zero in the Left Half-Plane ($Av = \dfrac{(1 + s_{z1})\times\ldots}{\ldots}$) provides a boost in gain AND phase at the point it is inserted. Unfortunately, the RHPZ gives a boost in gain, but lags the phase. More viciously, its position moves as a function of the load which makes its compensation an almost impossible exercise. Rolling-off the gain well under the worse RHPZ position is the usual solution. Let's also point out that the low-frequency RHPZ is only present in FLYBACK type converters (BOOST, BUCK-BOOST) operating in CCM and moves to higher-frequencies (then becoming negligible) when the power supply enters DCM. The loop compensation becomes easier. For additional information, reference [1] gives an interesting *experimental* solution to cure the BOOST from its low-frequency RHPZ.

How can I model my converter?

Two main solutions exist to carry AC and DC studies upon a converter. The first one is the well known State-Space Averaging (SSA) method introduced by R. D. MIDDLEBROOK and S. CÙK in 1976 [2] that leads to average models. In the modeling process, a set of equations describes the electrical characteristics of a switching system for the two stable positions of the switches, as **figures 3a** and **b** portrait for a BOOST type converter.

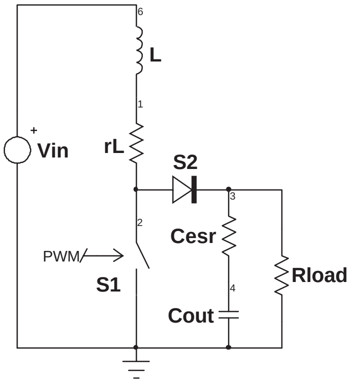


Figure 3a

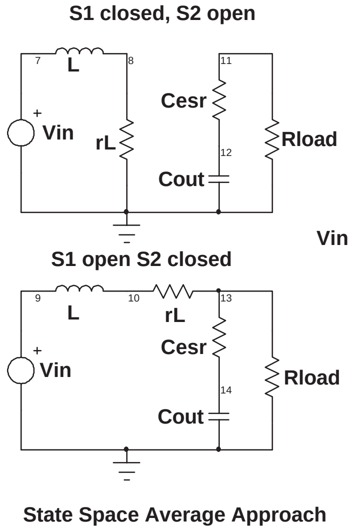


Figure 3b

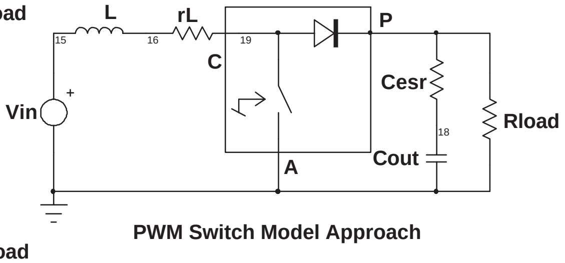


Figure 3c

The SSA technique consists in smoothing the discontinuity associated with the transition between these two states, then deriving a model where the switching component has disappeared in favor of a unique state equation describing the average behavior of the converter. The result is a set of continuous non-linear equations in which the state coefficients now depend upon the duty cycles **D** and **D'** (1-D). A linearization process will finally lead to a set of continuous linear equations. The reader interested by an in-depth and pedagogical description of these methods will find all the necessary information in MITCHELL's book [3].

As one can see from **figure 3**, the SSA models the converter in its entire electrical form. In other words, the process should be carried over all the elements of the converter, including various in/out passive components. Depending on the converter structure, the process can be very long and complicated.

In 1988, Vatché VORPERIAN, from Virginia Polytechnic Institute (VPEC), developed the concept of the Pulse Width Modulation (PWM) switch model [4]. VORPERIAN considered simply modeling the power switch alone, and then inserting an equivalent model into the converter schematic, in exactly the same way as it is done when studying the transfer function of a bipolar amplifier (**figure 3c**). With his method, VORPERIAN demonstrated among other results, that the flyback converter operating in DCM was still a second order system, affected by high-frequency second pole and RHPZ. An introduction to simulating with VORPERIAN's models is detailed in reference [5].

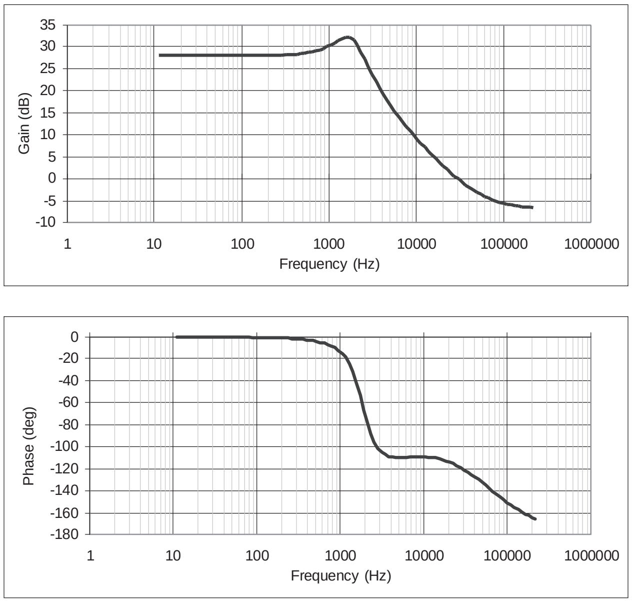
The Bode plot of the FLYBACK converter

From the previous works, the poles and zeroes of converters operating in DCM and CCM have been extracted, giving the designer the necessary insight to make a power

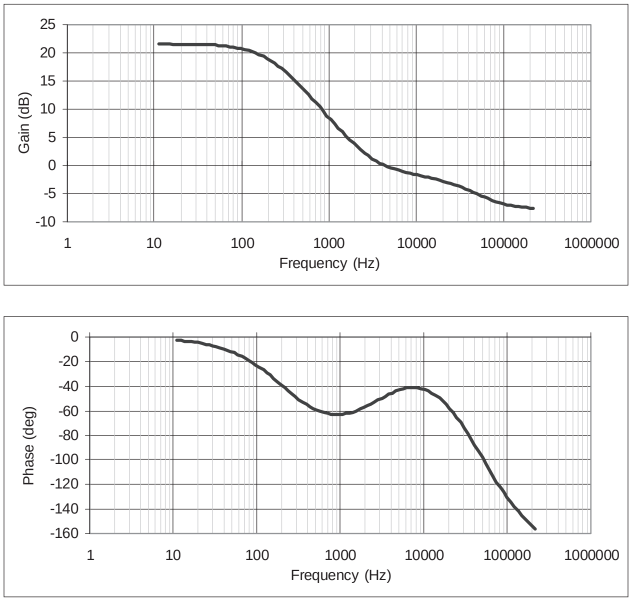
Table 1	DCM	CCM
1st order pole	$\frac{2}{2\times\pi\times R_{load}\times C_{out}}$	
2nd order pole	High frequency pole, see reference [4]	$\frac{(1-D)}{2\times\pi\times\sqrt{L_P\times C_{out}}}$
Left Half-Plane Zero	$\frac{1}{2\times\pi\times R_{ESR}\times C_{out}}$	$\frac{1}{2\times\pi\times R_{ESR}\times C_{out}}$
Right Half-Plane Zero	High frequency RHPZ,see reference [4]	$\frac{R_{load}\times(1-D)^2}{2\times\pi\times L_P\times D}$
Voutput/Vinput DC Gain	$N\times D\times\sqrt{\frac{R_{load}}{2\times L_P\times F_{SW}}}$	$\frac{D}{(1-D)}\times N$
Voutput/Verror DC Gain	$\frac{V_{input}}{V_{SAW}}\times\sqrt{\frac{R_{load}}{2\times L_P\times F_{SW}}}$	$\frac{V_{input}}{V_{SAW}}\times\left(1+\frac{V_{output}}{V_{input}}\right)^2$

supply stable and reliable. The summary in Table 1, gives their positions in function of the operating mode, and also specifies the various gain definitions for a FLYBACK converter. For Table 1, F_{SW} = switching frequency, V_{SAW} = sawtooth amplitude of the oscillator's ramp, and L_p = primary inductance

The Bode plots can be generated in a multitude of manual methods or in a more automated way by using a powerful dedicated software such as POWER 4-5-6 [6]. We have asked the program to design two 100kHz voltage-mode SMPS with equivalent output power levels, but operating in different modes. The results are given below (**figure 4**), including the high-frequencies pole and RHPZ in DCM, as described in [4].



Continuous Conduction Mode (**figure 4a**)



Discontinuous Conduction Mode (**figure 4b**)

From the above pictures, it is clear that the DCM converter will require a simple double-pole single zero compensation network (type 2 amplifier), while a two-pole two-zero type 3 amplifier appears to be mandatory to stabilize the CCM converter. Furthermore, the CCM's second order pole moves in relationship to the duty cycle while the poles/ zeroes are fixed in DCM.

SPICE simulations of the converter

One can distinguish between two big families of converter SPICE models, average and switching. The average models implement either the SSA technique or the VORPERIAN's solution. Since no switching component is associated with these models, they require a short computational time and can work in AC or TRANSIENT analysis. Some support large transient sweeps, while some only accept small-signal conditions. On the other side, switching models are the SPICE reproduction of the breadboard world and simulate the supply using the PWM controller you selected or the MOSFET model given by its manufacturer. Both models have their own advantages: average models simulate fast, but by definition, they cannot include leakage energy spikes or parasitic noise effects. Switching models take longer time to run because the simulator has to perform a thin analysis (internal step reduction) during each commutation cycles but since parasitic elements can be included, they allow the designer to dive into the nitty-gritty of the converter under study. Reference [7] will guide you in case you would like to write a switching model yourself.

SPICE models are available from several sources, but INTUSOFT (San-Pedro, CA), the IsSpice4 editor, has recently released his new SMPS library which gathers numerous average and switching models. Among these models, we will describe a very simple and accurate model which has been developed by Sam-BEN-YAAKOV from Ben-Gurion university (ISAREL). This model converges well and finds its DC point alone. Finally, it allows AC simulations as well as large signal sweeps. The netlist is given below:

```
**** Sam BEN-YAAKOV FLYBACK's Model ****
.SUBCKT FLYBACK DON IN OUT GND {FS=??? L=???
+ N=???}
BGIN IN GND I=I(VLM)*V(DON)/(V(DON)+V(DOFF))
BELM OUT1 GND V=V(IN)*V(DON)-V(OUT)*V(DOFF)/{N}
RM OUT1 5 1M
LM 5 8 {L}
VLM 8 GND
BGOUT GND OUT I=I(VLM)*V(DOFF)/{N}/
+ (V(DON)+V(DOFF))
VCLP VC 0 9M
D2 VC DOFF DBREAK
D1 DOFF 6 DBREAK
R4 DOFF 7 10
BDOFFM 6 GND V=1-V(DON)-9M
BDOFF 7 GND V=2*I(VLM)*{FS}*{L}/V(DON)/
+ V(IN)-V(DON)
.MODEL DBREAK D (TT=1N CJO=10P N=0.01)
.ENDS
```

The implementation of the model is really straightforward, as demonstrated by **figure 5a** schematic which shows the converter we already dimensioned with the help of POWER 4-5-6.

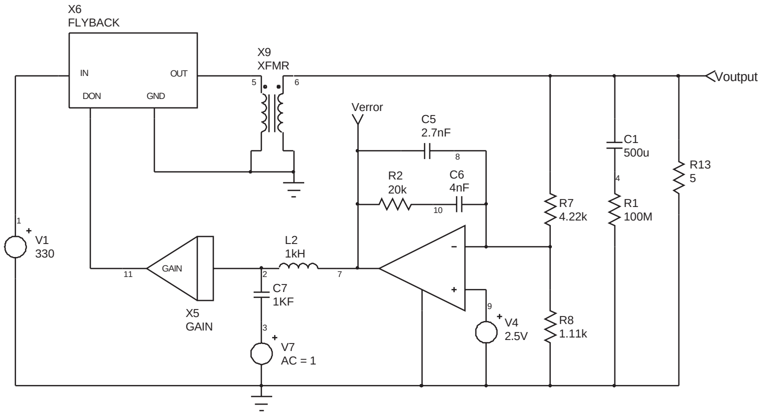


Figure 5a

It is interesting to temporarily open the loop and conduct AC simulations in order to isolate the error amplifier in AC and let you adjust the compensation network until the specifications are met. The fastest way to open the loop is to include an LC network as depicted in the above schematic (L2-C7). The inductive element maintains the DC error level such that the output stays at the required value. But it stops any AC error signal that would close the loop. The C element gives an AC signal injection thus allowing a normal AC sweep. To do so, let L2 1kH and C7 1kF. In the opposite sense, run a TRANSIENT by decreasing L2 to 1nH and C7 to 1pF. This method presents the advantage of an automatic DC duty cycle adjustment and allows you to quickly modify the output parameter without tweaking the duty source at every change.

The error amplifier model is directly derived from the specifications given by the controller's data-sheets you selected. A simplified macro-model can be built and simulated as refererence [7] details. You can also directly include a full detailed component to highlight the impact of its key parameters upon the supply under study (slewrate etc.). X5 subcircuit simulates the gain introduced by the PWM modulator. You can see it as a box converting a DC voltage (the error amplifier voltage) into a duty cycle D. The average models accept up to 1 volt as a duty cycle control voltage (D=100%). Generally, the IC's oscillator sawtooth can swing up to 3 or 4 volts, thus forcing the internal PWM stage to deliver the maximum duty cycle when the error amplifier reaches this value. To account for the 1 volt maximum input of our average models, the insertion of an attenuator with $1/V_{SAW}$ ratio after the error amplifier output is mandatory. For example, if the sawtooth amplitude of the integrated circuit we use is 2.5VPP, then the ratio will be: 1/2.5=0.4.

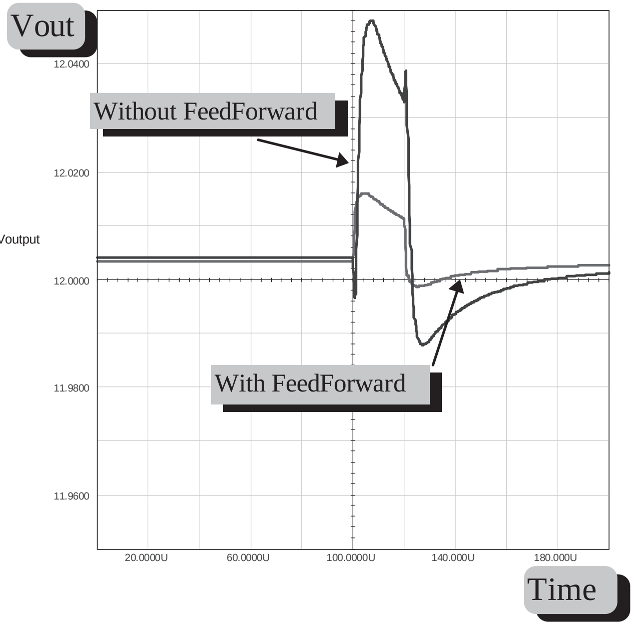


Figure 5b

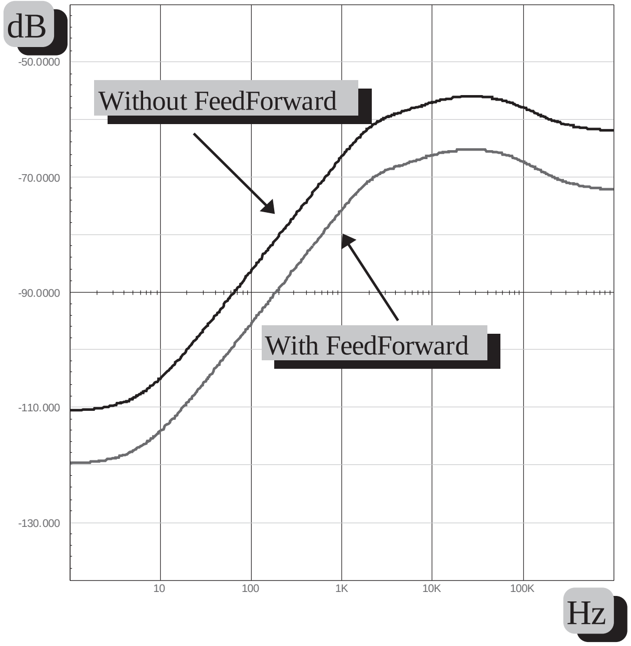


Figure 5c

These kind of SPICE circuits let you immediately check the parameters of interest without sacrificing your time in watching the machine computing! The audio susceptibility is delivered in a snap shot, as **figure 5b** portraits. Adding a bit of feedforward with a simple source in series with X5's output (500U*V(1)) gives you, as expected, a better behavior . The transient response to an input step does not take longer, as **figure 5c** depicts (10V input step) for both of the previous conditions.

A critical mode controller

As we saw, keeping your SMPS in the discontinuous mode will let you to design the compensation network in a more easy way. It will also ensure a stable and reliable behavior as long as you stay in the discontinuous area. How can you be sure to stay in DCM, regardless the load span you apply at the output?? Two solutions: a) you calculate L_p in order to always stay in DCM, but you assume to know all load conditions. b) you permanently adjust the switching frequency to stay DCM, whatever the load is. This last solution has been adopted in the MC33364 from MOTOROLA (Tempe, AZ). The critical conduction controller ensures a switch turn on immediately after the primary current has dropped to zero. In this case, you do no longer worry about the values your load will take since the controller tunes its frequency to keep the SMPS in DCM. The stability is then guaranted over the full load range.

Am I critical?

To answer this question, the controller needs to know the level of the primary current. The most economical solution exploits the signal delivered by the auxiliary winding. When this one has fallen to zero, an internal set is delivered to the latch, initiating a new cycle. In case the synchronisation signal would be lost, or when using the IC in a standalone application, an internal watchdog timer restarts the converter if the driver's output stays off more than 400µs after the inductor current has reached zero.

Output switching frequency clamp

As we already said, the system adjusts its frequency to maintain the DCM. However, in absence of load, the operational frequency can shift to high values, engendering unacceptable switching losses and making the design of the EMI filter a difficult task. To circumvent this intrinsic problem, the designers of the IC have added an internal frequency clamp whose function is to limit the maximum excursion. The MC33364 is thus declined in two versions including or not the clamping capability: 33364D and D1 limit to 126kHz the upper value of the switching frequency, while the 33364D2 does not host this feature.

Good riddance startup resistor!

The majority of offline SMPS are self supplied. A startup resistor charges a bulk capacitor until the IC's undervoltage limit is reached. While the bulk capacitor voltage begins to decrease, the circuit starts to actuate the switching transistor and the auxiliary supply feeds the controller back through the rectifier. But once the steady-state level is reached, the startup resistor is still there and wastes some substantial energy in heat. In low power SMPS, you hunt down any source of wasted power to raise the overall efficiency at an acceptable value. **Figure 6** shows the method MOTOROLA has implemented in the 33364 to quash the startup element.

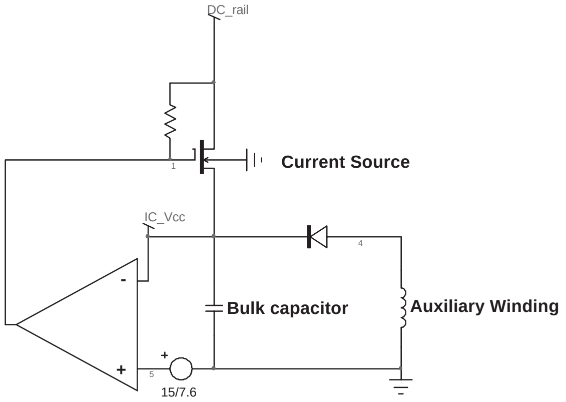


Figure 6

It works as follow: when the mains is first applied to the converter, the MOSFET charges up the bulk capacitor until the voltage on its pins reaches the startup threshold of 15V. At this time, the MOSFET opens and the circuit operates by itself.

A low part-count converter

The 33364 has been specifically designed to save a maximum of parts. **Figure 7** illustrates this will for an economical 12W AC/DC wall adapter.

The leakage energy spike is clipped by the R5-C5 network whose second function is to smooth the rising drain voltage, correspondingly limiting the radiated noise. This last feature is unfortunately no longer valid when you use a clipping circuit made of a fast rectifier and a zener diode. The circuit's sensitivity to the noise present on the sense resistor is largely diminished by the implementation of a leading edge blanking network. This system blanks the starting portion of the primary ramp-up current which can be the seat of spurious spikes: a resonance with the parasitic inter-winding capacitors and the ON gate-source current.

Since every current mode converter are inherently unstable over a certain duty-cycle value, it can be wise to add some current ramp compensation even in DCM, as Ray RIDLEY demonstrated in the 90's [8]. How can you provide the 33364 sense input with some ramp to since no oscillator pinout is available? **Figure 8** shows a possible solution by integrating the driver's output. The resulting linear ramp will add to the sense information, thus stabilizing the converter. You can also adopt this method in other cases, even when the oscillator's ramp is available. The integrator solution prevents the internal oscillator to be externally loaded which in certain circumstances can lead to erratic behaviors.

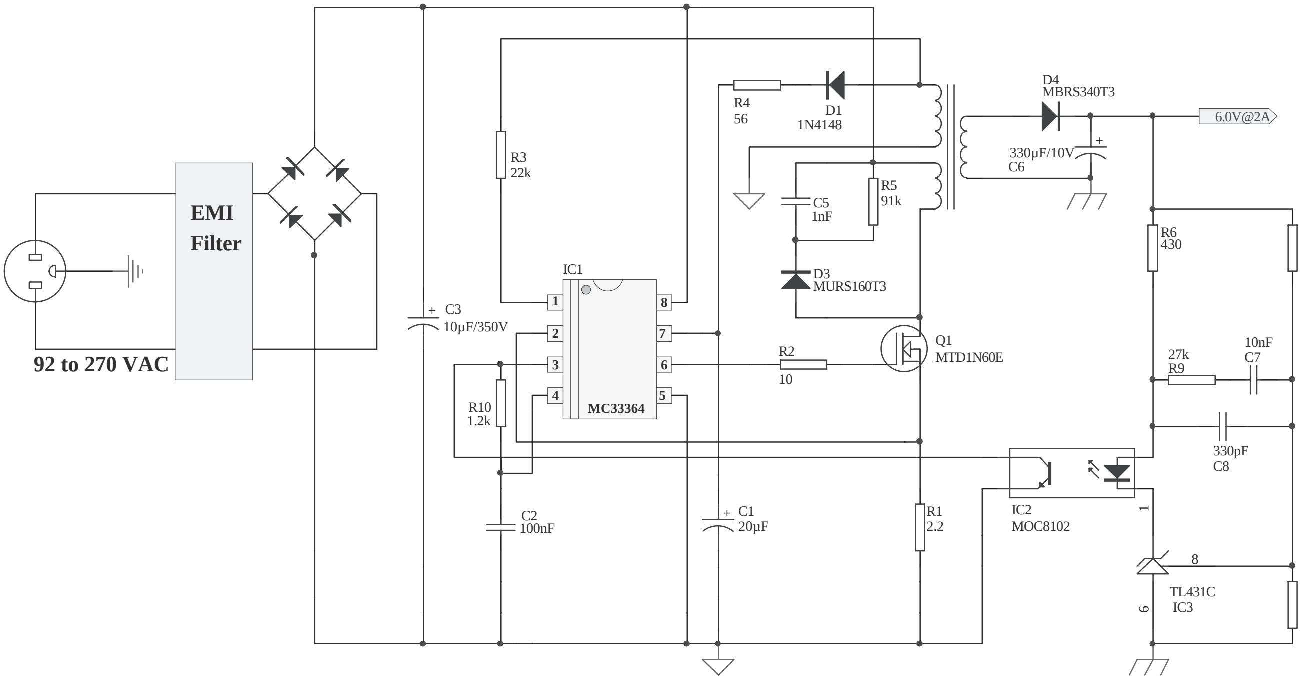


Figure 7

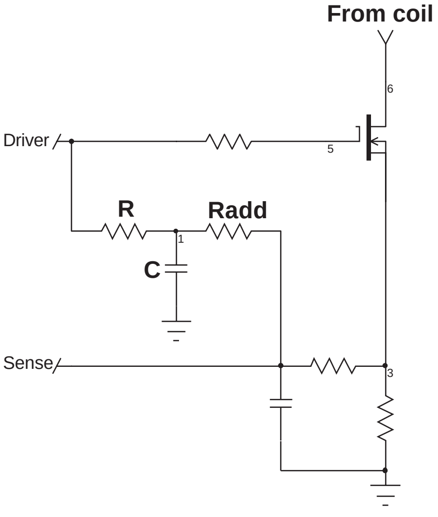


Figure 8

References

1. Elimination of the Positive Zero in Fixed Frequency Boost and Flyback Converters, D. M. SABLE, B. H. CHO, R. B. RIDLEY, APEC, March 1990
2. R. D. MIDDLEBROOK and S. CUK, " A general Unified Approach to Modeling Switching Converter Power Stages ", IEEE PESC, 1976 Record, pp 18-34
3. D. M. MITCHELL, "DC-DC Switching Regulators Analysis", distributed by e/j BLOOM Associates (http://www.ejbloom.com).
4. Vatché VORPERIAN, "Simplified Analysis of PWM Converters Using The Model of The PWM Switch, Parts I (CCM) and II (DCM) ", Transactions on Aerospace and Electronics Systems, Vol. 26, N°3, May 1990
5. C.BASSO, "A tutorial introduction to simulating current mode power stages", PCIM October 97
6. Ridley Engineering home page: http://members.aol.com/ridleyeng/index.html
7. C. BASSO, "Write your own generic SPICE Power Supplies controller models, part I and II", PCIM April/May 97
8. R. B. RIDLEY, "A new small-signal model for current-mode control", PhD. dissertation, Virginia Polytechnic Institute and State University, 1990 (e-mail : RRIDLEY@AOL.COM)

Intusoft Newsletter

Personal Computer Circuit Design Tools

March 1993 Issue

intusoft

(310) 833-0710 Fax (310) 833-9658

Simulating SMPS Designs

The ubiquitous personal computer has made a mess of office power mains. Its characteristic cosine shaped power pulse requires 75% more current carrying capacity than is necessary. Moreover, the turn-on surge is so large that it can cause operational glitches in other equipment. Government regulations will soon force power supply designers to correct this problem. In this article, we will explore some of the design concepts for power factor friendly switched mode power supplies, SMPS. Our motivation here is to show how IsSpice can be used as a high level design tool to explore concepts and develop design requirements before beginning the detailed design phase.

First, let's look at the problem. To begin with, the power line impedance must be modeled. With the topology shown in Figure 1, we don't expect the component values to affect current very much, however, the voltage distortion is totally dependent on this configuration. The short circuit impedance was estimated to be equally divided between resistive and inductive components. The high frequency impedance

Exploring SMPS Designs Using IsSpice
Average Models For Switching Converter Design
Larry Meares, Charles Hymowitz
Excerpts from the March 1993 Intusoft Newsletter #29

Exploring SMPS Designs Using IsSpice

was approximated to be that of free space, hence the choice for the capacitance and the inductor damping. The model could be made more accurate by using test data for a specific case; but the current waveform will not change much and hence the power factor calculation will not be seriously affected by the approximation used here.

The circuit topology of most off-line SMPS uses a full wave capacitor-rectifier with some soft start provisions, mainly to prevent component damage at start-up. The detailed circuit is shown in Figure 1. The graph illustrates the effect of this circuitry on input power along with the capacitor-rectifier voltage. Power factor is defined as the product of the RMS current and voltage divided by the average input power delivered to a circuit. For this circuit; the following measurements were made using IntuScope, a SPICE

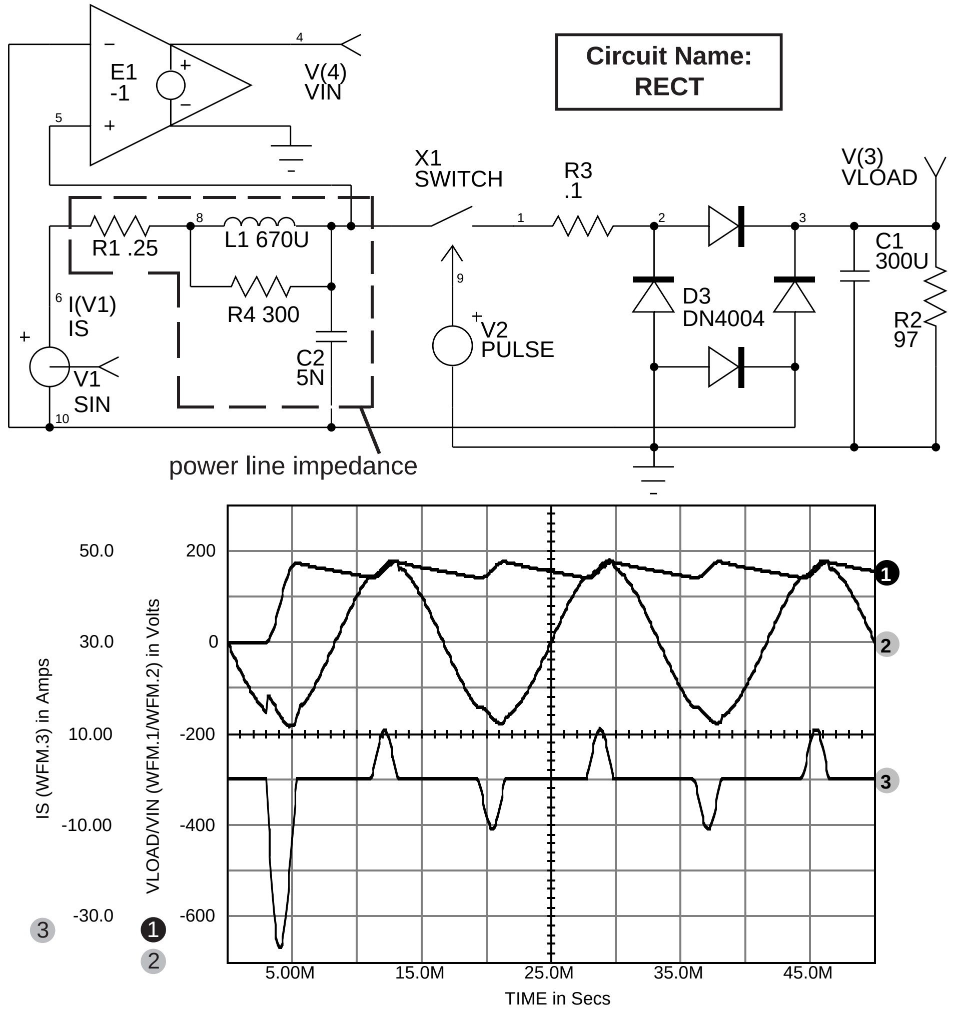


Figure 1, Rectifier circuit showing typical SMPS start-up transient. Current is 3.7A RMS, 37A Peak with the peak output ripple at about 5% of the average output voltage.

data post processing program:

Iin=3.68 ARMS	Vin=120 VRMS
Pavg=263 Watts	Power Factor = .596

The voltage, VLOAD, in Figure 1, is the unregulated input to the switched mode power regulator. Notice that the peak ripple is about 5% of the average voltage. The regulator must remove this ripple along with other line and load variations.

The object of a power factor correction circuit is to force the input current to be sinusoidally varying and in phase with the input voltage. The input power will then be pulsating at a frequency of twice the input voltage ranging from zero to twice the average input power. The output voltage and current, on the other hand, must be constant. Therefore, it is necessary to provide reactive elements to store power which can later be used when the input power is less than required.

To accomplish this, the circuit shown in Figure 2 uses a buck-boost regulator to control the input current at a desired level. In this

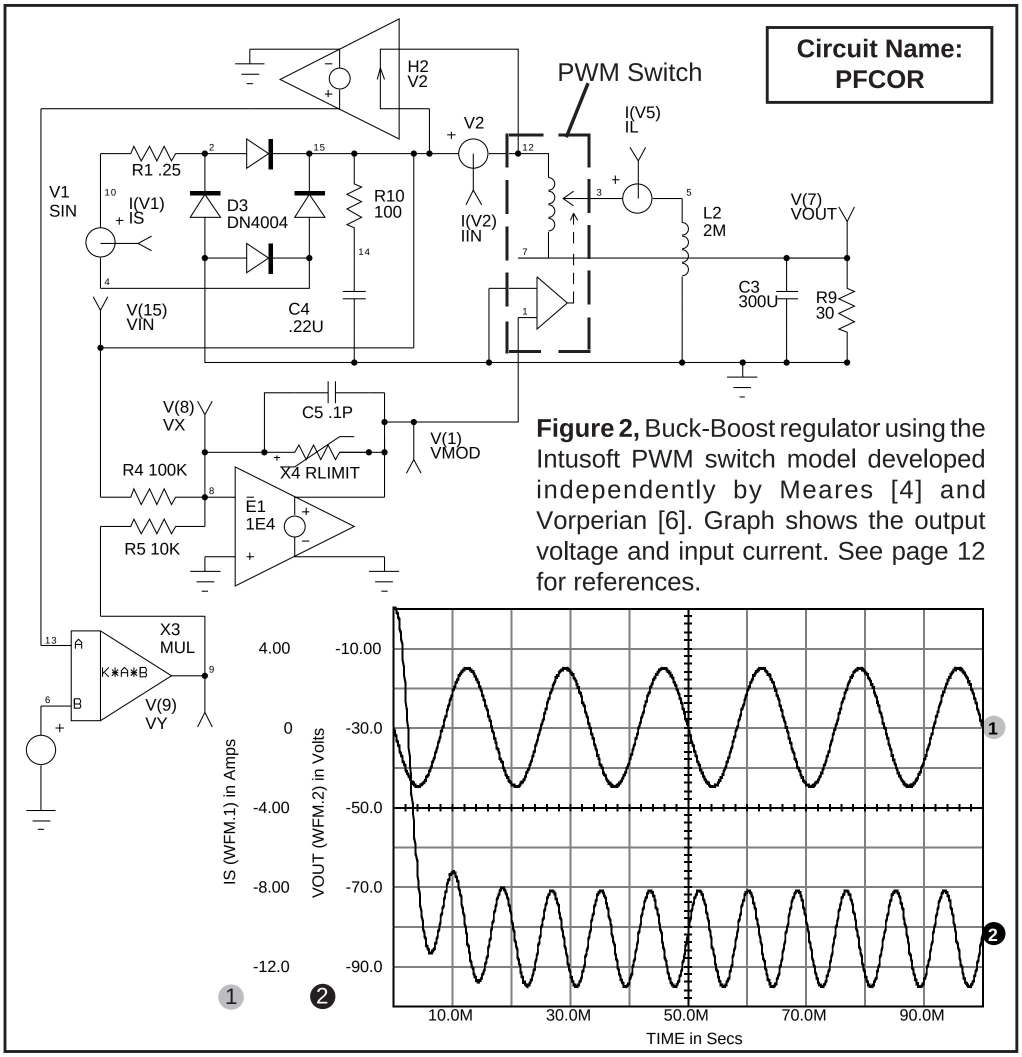


Figure 2, Buck-Boost regulator using the Intusoft PWM switch model developed independently by Meares [4] and Vorperian [6]. Graph shows the output voltage and input current. See page 12 for references.

circuit, the pulse width modulator model from the Intusoft library, called PWM, is connected in the buck-boost configuration. To learn more about using this model and review the various PWM topologies, see the article entitled "Average Models For Switching Converter Design" in this newsletter. Here, we will show why the buck-boost configuration has excellent control characteristics for input current.

In Figure 2, the input current is compared with a signal proportional to the rectified input (V(15)) and the resulting error signal (VMOD) is used to control the PWM switch. Notice the use of the limiting resistor (Rlimit) in the error amplifier. This was previously discussed in depth in the September 1992 newsletter. This type of limiter converges and performs better than hard limiters created with Table-type functions. The resulting waveforms show excellent control of the power factor and start-up current. The power factor, as measured using IntuScope, is 1.0. The output voltage magnitude is lower than that of the capacitor-rectifier circuit and the percent ripple has increased from 5% to 15%.

So far we haven't addressed the output regulator. Ideally, the output regulator is a constant power device as is our input regulator. With this configuration we will be using a constant current source to drive a constant current load; therefore, the voltage at the output of our first stage will be difficult to control. Also, the transfer function from input to output will contain those dreaded right half plane zeros, causing problems in designing a responsive controller.

If we select a buck regulator for the second stage, then feedback can be used to control the input current such that the second PWM controller has a 50% duty cycle. If this control loop is very fast, then, the ripple will distort the input current. On the other hand, if it is too slow, the second stage may be presented with too high or too low an input voltage. To overcome these conflicts, the input current requirement can be estimated based on the output power and input voltage. For example, $IS \ast Vs= k \ast Vset \ast Ireq$; where:

IS	=	Input current	Vs	=	Input voltage
Vset	=	Desired output voltage	Ireq	=	Required output current
Vout	=	Output voltage	Iout	=	Output current
k	=	Efficiency			

then, IS $= k \ast Vset \ast Ireq/ Vin$
however, Ireq $= Vset / Rload$
and Rload $= Vout / Iout$
finally giving IS $= k \ast Vset \ast Vset \ast Iout / (Vout \ast Vin)$

Several simplifications can be made. First, Vin is assumed to be a constant since it won't vary by more then 15%. Next, since the division by VLOAD may be costly, we should investigate the consequence of making it a constant. The schematic in Figure 3 shows the complete two regulator circuit along with the control mechanism we just discussed. Figure 4 shows the performance comparison when the Vout division is removed. The start-up is somewhat slower, however, the slight performance degradation is not objectionable. Power factor, as measured using IntuScope, was .991. The duty cycle control provided by amplifier E3 makes up for variation in efficiency with load and changes in input voltage. Its authority is limited to 20% of the total control range. This limit will be adjusted as the design progresses as will the compensation component values.

Overall, we have produced a solid framework for an initial design. The next step is to add the input and output noise filters and finalize the compensation circuitry. Then, the power components can be sized based on the currents and voltages found in this simulation. It is important to extract the maximum information from the continuous time model because the actual switched model simulation will run hundreds of times slower. The complete expanded two regulator circuit (PCOR5) is contained on the newsletter floppy disk. Due to space limitations, the final design is discussed in sections. The results are substantially the same as those shown in Figure 4. The power factor is still above 99% although some waveform distortion is beginning to creep in because capacitors were added at the input for noise filtering.

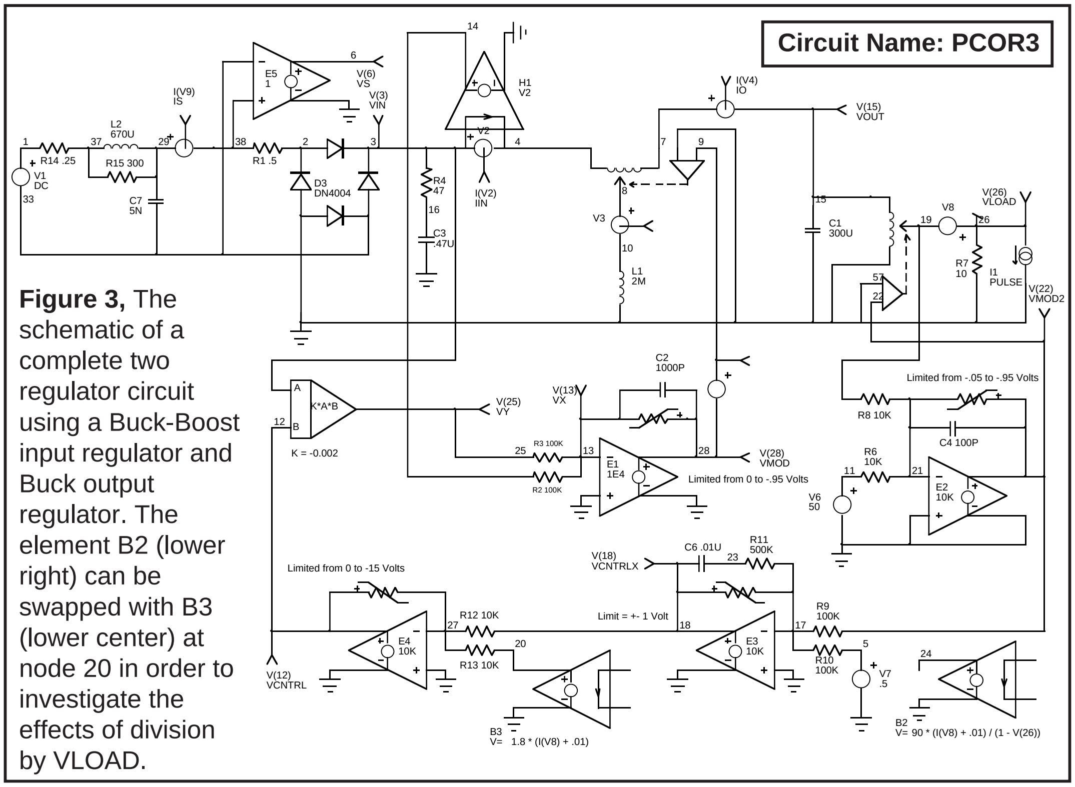


Figure 3, The schematic of a complete two regulator circuit using a Buck-Boost input regulator and Buck output regulator. The element B2 (lower right) can be swapped with B3 (lower center) at node 20 in order to investigate the effects of division by VLOAD.

The compensation scheme is interesting, especially the outer loop that is used to adjust the output regulator head room. The loop gain for this section is shown in Figure 5 and the schematic

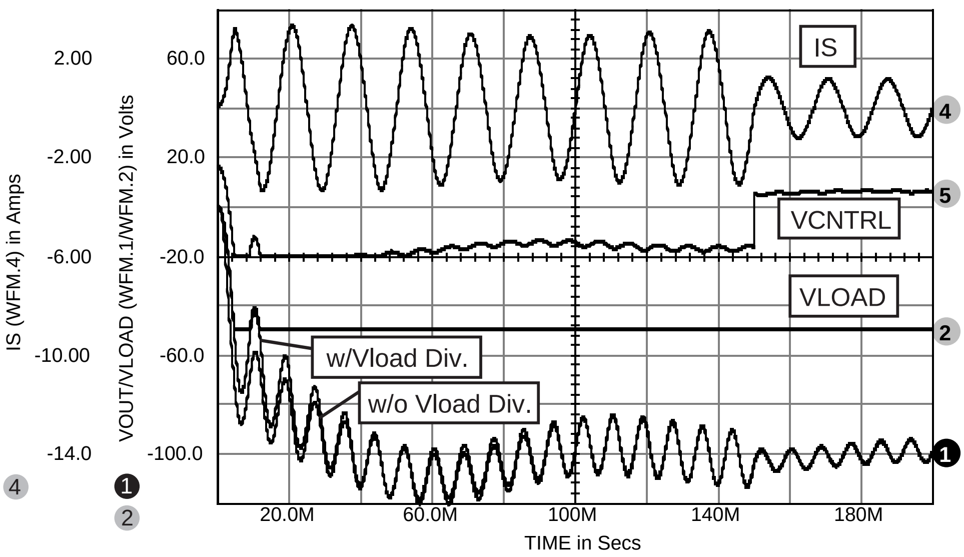


Figure 4, Comparison between controllers with and without VLOAD division reveals minimal differences.

of the compensation circuitry for this loop is shown in Figure 6. Notice the technique for making closed loop frequency response measurements. A voltage source is inserted at the point we wish to "cut" the loop, in this case V2. The circuit is excited using only this voltage source. The gain is the difference in the log of the magnitudes and the phase is the difference in phase at each side of the voltage source, VINOL and VOUTOL. The exciting signal must only come from the source we inserted, otherwise the voltages on either side will contain an additional component, making the gain and phase calculations wrong. Capacitor C3 is used to shape the start-up current. If it were to be removed, start-up would be fasted; however, the start-up surge current would be larger. If your final input regulator has current limiting, then this capacitor would be unnecessary.

The output regulator, shown in Figure 7, uses a "poor mans" form of current feedback in order to extend loop bandwidth past the L1-

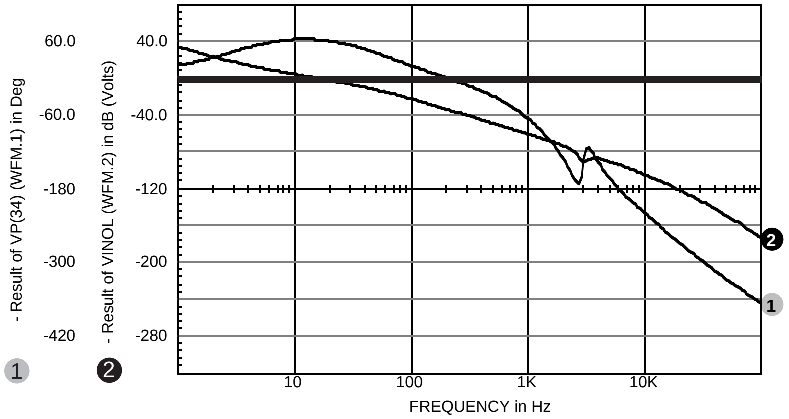


Figure 5, Frequency response of the outer loop head room regulator (VINOL/VOUTOL).

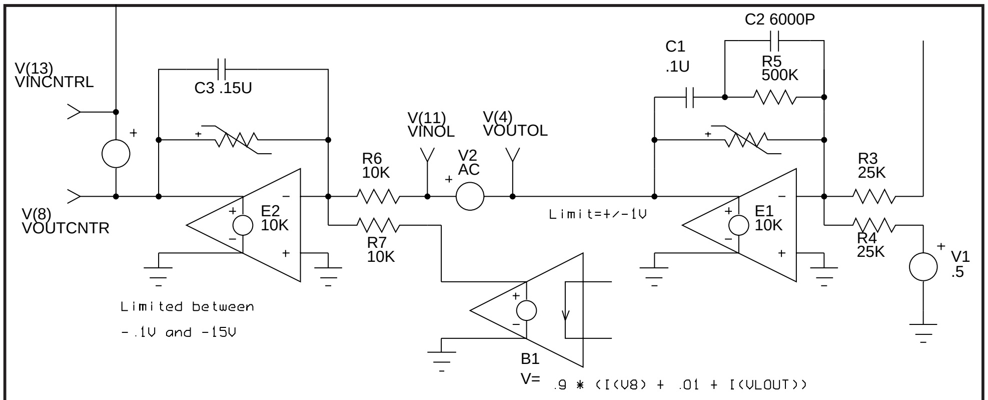


Figure 6, Head room regulator section showing the compensation circuitry and technique for calculating a closed loop response.

C2 resonance. Current feedback is established by integrating the voltage across L1 using amplifier E1. The voltage at test point VLOAD should also be summed, however, it is approximately a constant and was omitted to reduce complexity. The VLOAD-VLOADIN loop response is shown in Figure 8. In order to help the AC analyses converge rapidly, a .NODESET statement was inserted in order to set the output to 5mV for the DC analysis. The limiters using RLIMIT elements force reasonable operating ranges and also act to aid convergence. Rlimit is made with the new "If-Then-Else" behavioral feature in IsSpice3. Generic parameters include Vmax, Rval, Vmin, and Rmin.

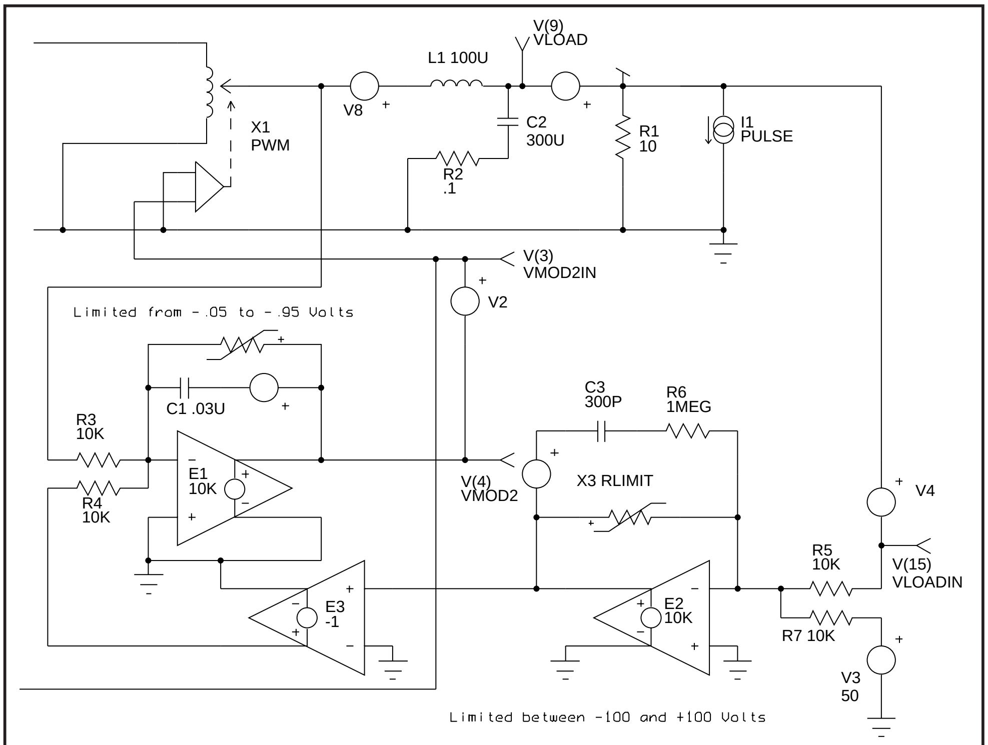


Figure 7, The output stage regulator section. The loop is broken at VLOADIN and V4 is inserted as the AC stimulus source.

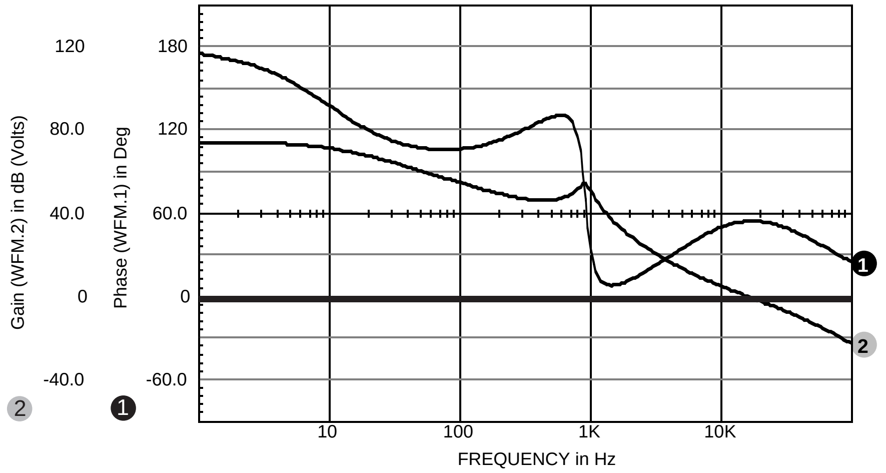


Figure 8, The VLOAD-VLOADIN loop response for the buck output regulator section.

The final stabilization and filtering is shown for the input regulator section in Figure 9, with Figure 10 showing the results. In this loop (VMOD-VMODIN), we see the major resonant peaks from the input regulator at 100 Hz and the input filter at 2KHz.

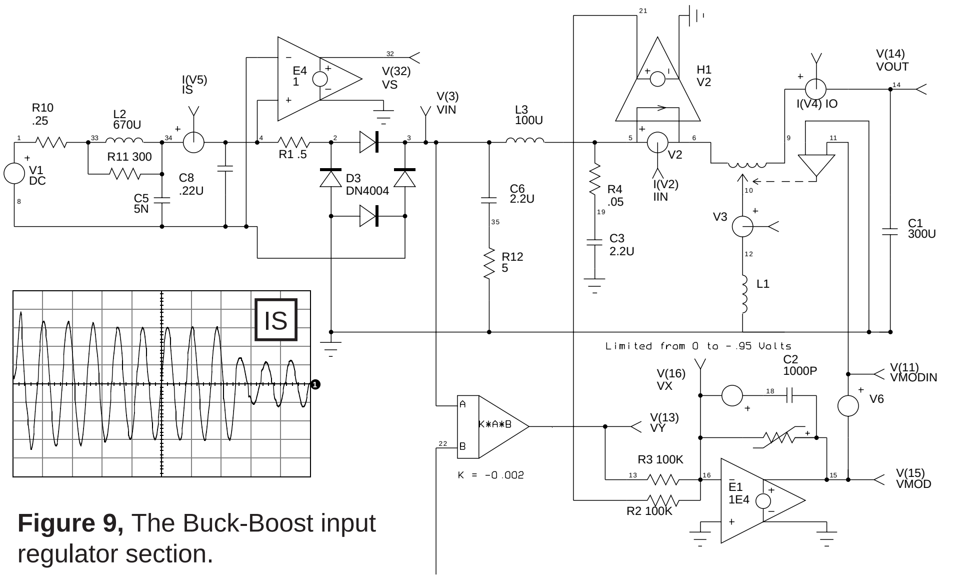


Figure 9, The Buck-Boost input regulator section.

While the design is by no means complete, we are confident that the topologies, filters, and control loops are all realizable. It's clear that initialization of the regulators, particularly in relation to housekeeping power turn on must be kept in mind as the design progresses. Selection of the "best" pulse width modulator technology will further refine the control loops. Schematic and SPICE netlists for the circuits shown here, and for switching models of various regulators that can be utilized along with the PWM switch, are included on the newsletter floppy disk. The designs presented here are geared to a 50KHz chopper frequency for the input regulator and 100-200 KHz for the output regulator. Changing these frequencies will affect performance and require revisiting the control loop design.

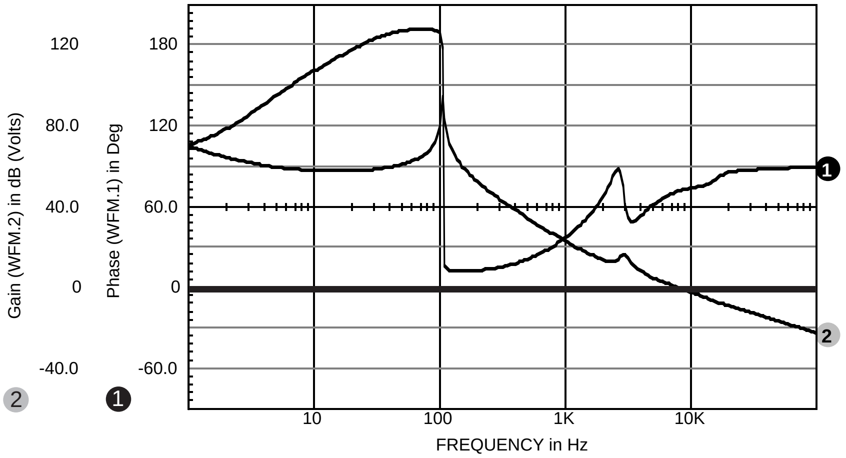


Figure 10, The VLOAD-VLOADIN loop response for the input regulator section.

Average Models For Switching Converter Design

The mathematical basis for modeling switching regulators was established by Middlebrook using state-space average techniques [1]. The implementation using SPICE elements was initially done by Keller [2] and Bello [3] with further enhancements by Meares [4] and Vorperian [6]. The resulting "PWM switch" model can be used for DC, AC small signal, and large signal time domain simulations. The basic PWM model is simply an electrically variable turns ratio DC transformer, hence the pictorial representation. The duty cycle control has two inputs allowing a differential control voltage. It has been shown to provide excellent experimental correlation, even in the neighborhood of the Nyquist frequency (usually 1/2 the switching frequency). In fact, it is possible to predict response past the Nyquist frequency by adding a zero order hold element. But we all know enough to keep our control system bandwidths well below the Nyquist frequency in order to avoid modal nonlinearities. The Buck-Boost and Cuk topologies offer the capability to convert the input voltage to an output voltage ranging from nearly zero to substantially greater than the input voltage. Both topologies, in the transformerless configuration, produce the opposite polarity output voltage from

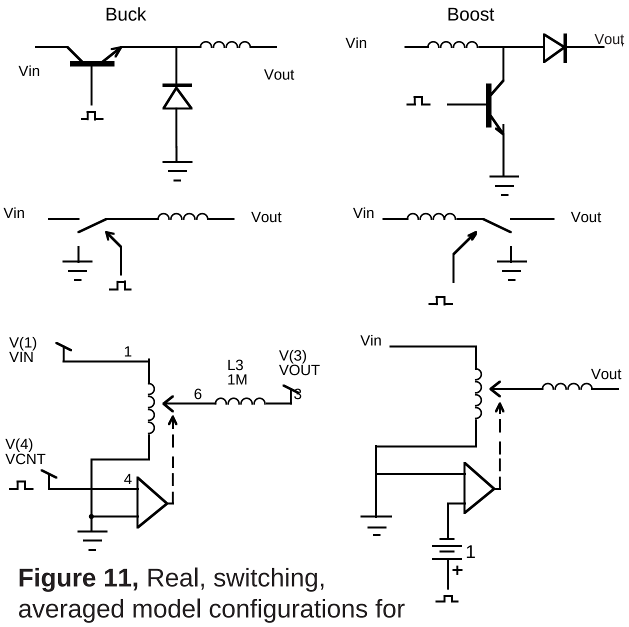


Figure 11, Real, switching, averaged model configurations for the Buck and Boost power stages.

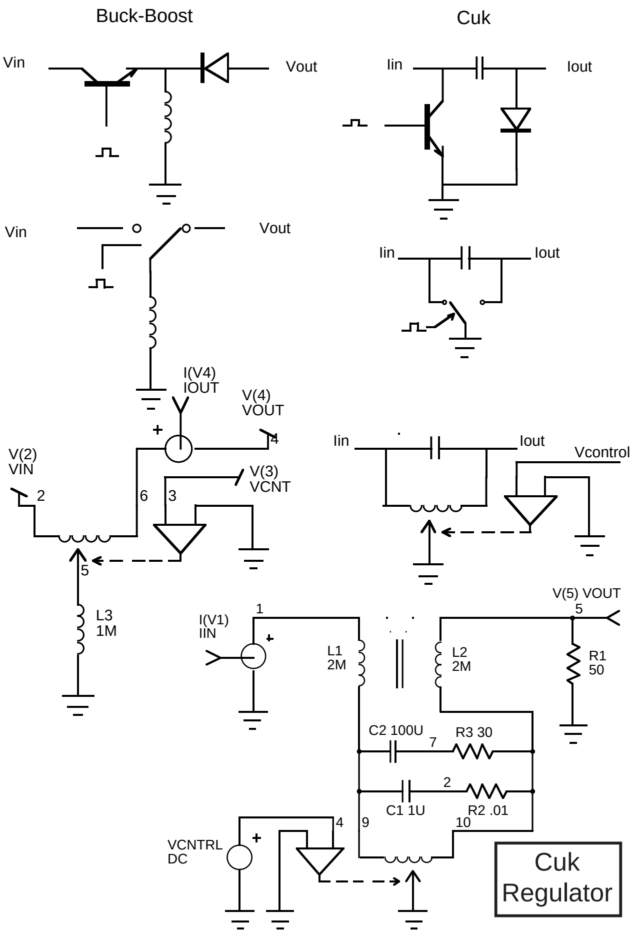


Figure 12, Real, switching and averaged configurations for the Buck-Boost and Cuk power stages. Actual Cuk regulator with external parts shown below right.

the input voltage. Since most converters will incorporate transformer isolation within their topology, the polarity inversion is usually not significant.

The duty ratio is defined as the period the PWM switch is turned on divided by the switching period. When forcing switching in both directions, the distinction gets a bit blurred. It is frequently necessary to replace the duty ratio, D, with its inverse, that is Di=1 - D (Fig 11 right bottom), which is the time the opposite switch is on divided by the switching period. For the AC analysis, this will shift phase by 180°. In the real world, of course you will have to determine the control range and polarity of your PWM in order to account for the scaling and polarity differences between your circuit and the simulation.

The Basic Power Supply Topologies

There are 4 basic power supply topologies; Buck, Boost, Buck-Boost and Cuk. The most familiar is the Buck regulator. It takes an input voltage from a power source and chops it into a series of pulses. The pulsating voltage is then smoothed using an inductor to produce an average output voltage which is the product of the duty ratio and the input voltage. A typical Buck regulator is shown in Figure 11 using bipolar components. Notice that we always associate one reactive, averaging element with each topology. In the first 3 topologies we are concerned with voltage transformation and allow ripple currents; while the dual CUK topology (Figure 12), transforms current and allows ripple voltage. All 4 topologies require additional filtering at the input and output to make a practical power supply. The Boost configuration shown in Figure 11 is quite different in appearance when viewed as a set of bipolar components; however, when replaced with a "forced" PWM switch representation, it is simply a Buck regulator with the input and the output reversed. Representing each of these configurations using "forced" switching is the same as restricting them to continuous conduction of current in the reactive element. It turns out that this assumption is valid over most operating points of the power electronics we design. This is a very important consideration for simulation because the computational complexity is dramatically increased when discontinuous conduction must be simulated. For the Buck regulator, this occurs at light load; in fact with no load at all the Buck regulator with free wheeling switch commutation fails to provide any regulation, making output and input voltages equal for duty ratios approaching zero. Clearly this condition must be solved in your circuit design, usually with some kind of bleed load; even a status light would work. The point is that the discontinuous conduction region is generally avoided and the continuous conduction region is frequently extended using non-linear inductors. While Intusoft and the enclosed newsletter floppy provide models capable of transitioning this region[1], we strongly recommend using the continuous model, at least for initial design tradeoff's.

The PWM switch is a versatile element that allows simulation of the majority of features of switching regulators. Its use is essential for performing effective simulations. The floppy disk, included for newsletter subscribers (available to non-subscribers for $20), contains all of the schematics and SPICE circuit netlists in this newsletter, plus models for switching regulators (SG1524, 25, 26, and UA1846 taken from reference [10]). Models for the basic PWM topologies, the PWM switch subcircuit, and an assortment of over 50 models for high power components (Mosfets, BJTs, diodes, SCRs, and IGBTs) are also included.

[1] R. Middlebrook and S. Cuk, "A General Unified Approach to Modeling Switching Converter Power Stages", IEEE PESC, 1976, pp. 18-34

[2] R. Keller, "Closed-loop Testing and Computer Analysis Aid Design of Control Systems", Electronic Design, Nov. 22 1978, pp. 132-138

[3] V. Bello, "Computer Aided Analysis of Switching Regulators Using SPICE2", IEEE PESC, 1980, pp. 3-11

[4] L. Meares, "New Simulation Techniques Using SPICE", IEEE APEC April 1986, pp. 198-205

[5] L. Meares, "Modeling Pulse Width Modulators", Intusoft Newsletter, August 1990

[6] V. Vorperian, "Nonlinear Modeling of the PWM Switch", IEEE Transactions on Power Electronics, Vol. 4, #2, April 1989

[7] V. Vorperian, "Simplify Your PWM Converter Analysis Using The Model of The PWM Switch", VPEC Current, Fall 1988

[8] V. Vorperian, "Simplify PWM Converter Analysis Using a PWM Switch Model", PCIM March 1990 pp. 8-16

[9] L. Dixon, "Spice Simulation of Switching Power Supply Performance", Unitrode Corp., 1991

[10] V. Bello, "Simulation of Switching Regulators Using SPICE2; A Collection of Papers and Subcircuit Models", Meta-Software, 1991

References [9, 10] describe many of the PWM models included on the newsletter floppy disk and add many excellent regulator design examples.

Intusoft Newsletter

Personal Computer Circuit Design Tools

#43 August 1995 Issue

intusoft

(310) 833-0710 Fax (310) 833-9658

Simulating SMPS Designs

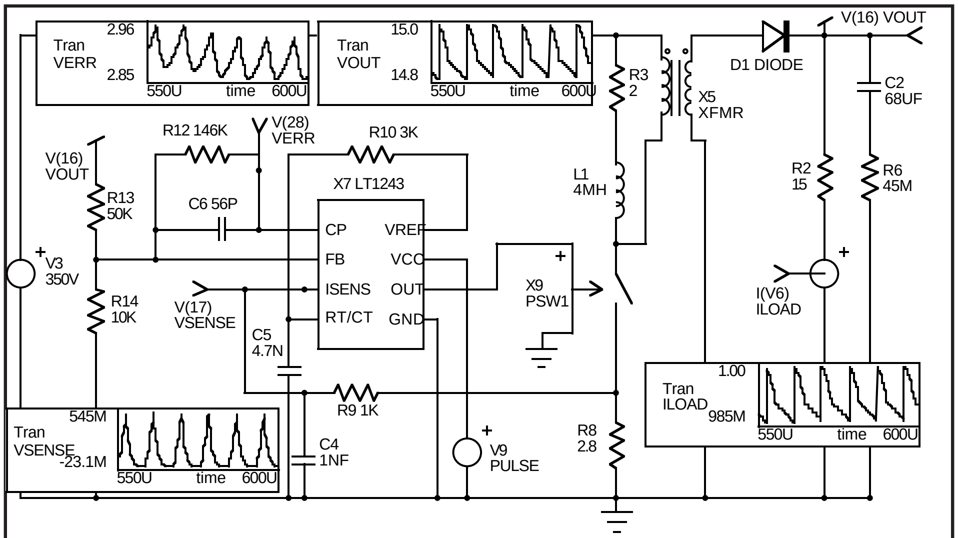


Figure 1, Intusoft's New Power Supply Designer's Library allows you to simulate forward converters like this one using the LT1243. See page 2 for more details.

Simulating SMPS Designs

SSDI Diode Rectifier Models

High Efficiency Step-Down Converter

Steve Sandler, Charles Hymowitz

Excerpts from the August 1995 Intusoft Newsletter #43

Simulating SMPS Designs

In the June 1995 newsletter we introduced a new SPICE model library for power supply designers. The library contains a comprehensive set of nonlinear models for popular Pulse Width Modulation (PWM) ICs. Here we continue our review with a few examples of what the models allow you to accomplish.

Applications today are much more demanding, requiring increases in switching frequency, higher efficiency and lower standby current. State space models simply do not reveal many important factors which influence these performance characteristics. As shown in Table 1, the models accurately account for many characteristics including prop. delay, switching speed, drive capability, and operating current.

The features included in the models allow you to directly compare the performance of components from different vendors. You can also analyze the effects of different implementations such as peak current mode control, hysteretic current control, low voltage and low operating current to name a few. The new library has over 400 models:

- **Models for various PWM ICs including those from Unitrode, Linear Technology, Siliconix, and Cherry Semiconductor**
- **'Unified' state space model for forward, and flyback converters topologies plus state space models for several specific ICs**
- **Nonlinear Magnetic Cores and Transformers**
- **Power Mosfet Drivers**
- **Motor controller IC (UC1637)**
- **Power factor correction IC (UC1854)**

The models are compatible with ICAP/4 systems on DOS, NEC, Windows, Macintosh and the Power Macintosh. ICAP/4Lite and ICAP/4Lite Xtra systems must have the IsSpice4 upgrade in order to run the new models.

Hysteretic CMC and Buck Regulators

Figure 2 shows a current mode converter example using the Cherry CS-322. The circuit uses hysteretic control, which is different from constant frequency or constant off time control. In constant frequency control, the instant of turn-on is always independent of the closed loop dynamics of the supply thereby limiting its regulation performance. However, because hysteretic control can change both edges of the switching waveform to keep a constant inductor current hysteresis, it has no mode on instabilities; no slope compensation is required. In Figure 2, the load resistance was varied from 5Ω to 2.5Ω. The

UC1842/3/4/5 Model Performance Comparison

Parameter	Condition	Type Spec	IsSpice4
Reference			
Output Voltage	1mA	5V	5V
Load Regulation	1-20mA	6mV	6.2mV
Oscillator Section			
Initial Accuracy		52kHz	52.1kHz
Amplitude		1.7Vp-p	1.72Vp-p
Discharge Current Standard		10mA	9.8mA
Error Amplifier			
Input Voltage		2.5V	2.5V
AVOL		90dB	89.1dB
Unity Gain Bandwidth		1mHz	1mHz
Output Sink Current		6mA	6.07mA
Output Source Current		-0.8mA	-0.8mA
Vout High		6V	5.64V
Vout Low		0.7V	0.72V
Current Sense			
Gain		3V/V	3V/V
Maximum Input Signal		1V	1V
Delay To Output		150nS	151nS
Output Section			
Output Low Level	20mA	0.1V	0.13V
	200mA	1.5V	1.43V
Output High Level	20mA	13.5V	13.56V
	200mA	13.5V	13.43V
Rise Time		50nS	35nS
Fall Time		50nS	50nS
Undervoltage Lockout			
Start Threshold	UC1842/4	16V	16V
	UC1843/5	8.4V	8.4V
Min Operating after Turn-On	UC1842/4	10V	10V
	UC1843/5	7.6V	7.6V
PWM Section			
Maximum Duty Cycle	UC1842/4	97%	97%
	UC1843/5	48%	49%
Minimum Duty Cycle		0%	0%
Total Standby Current			
Start-up Current	UC1842/4	0.5mA	0.45mA
	UC1843/5	0.5mA	0.51mA
Operating Current	UC1842/4	11mA	11.1mA
	UC1843/5	11mA	11.5mA

Table 1, The PWM models use a mixed mode modeling approach. They combine switches, behavioral models, and logic gates in order to provide a full set of features, simulation efficiency and excellent accuracy as shown above.

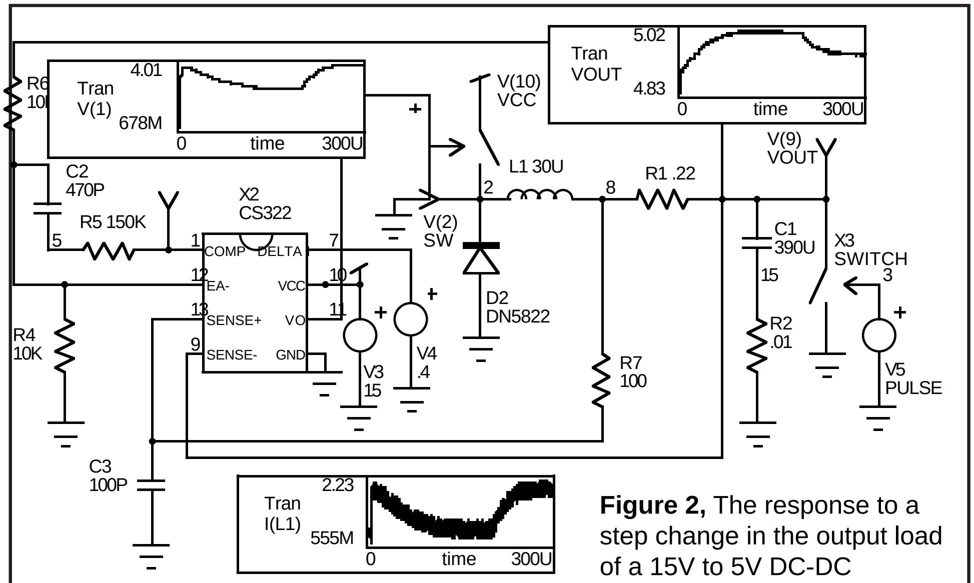


Figure 2, The response to a step change in the output load of a 15V to 5V DC-DC converter using the Cherry CS-322 Hysteretic Current mode controller. The switching frequency is 300kHz.

CS-322 simulates very quickly; the entire 300μs simulation took only 181.9s on a Pentium/90 even with TMAX (maximum timestep) limited to 200ns. Gear integration was used to speed the simulation while Tmax was set to 200n in order to maintain accuracy. In Figure 3 we see the result when the load is short-circuited by changing the switch resistance from 5Ω to 1mΩ in 1μs.

Figure 4 shows a 5V-to-3V, 300mA, 1MHz buck regulator example as described in the Siliconix Low Voltage DC-to-DC Converter design guide. The design uses the high speed Si9145 switchmode controller and low on-resistance (P-channel 20mΩ Si4435, N-channel 100mΩ Si9952) Little Foot® power Mosfets. Traditionally, SMPS simulations, especially with this level of detail, can take a long time to simulate. New modeling techniques, however, have brought the simulation

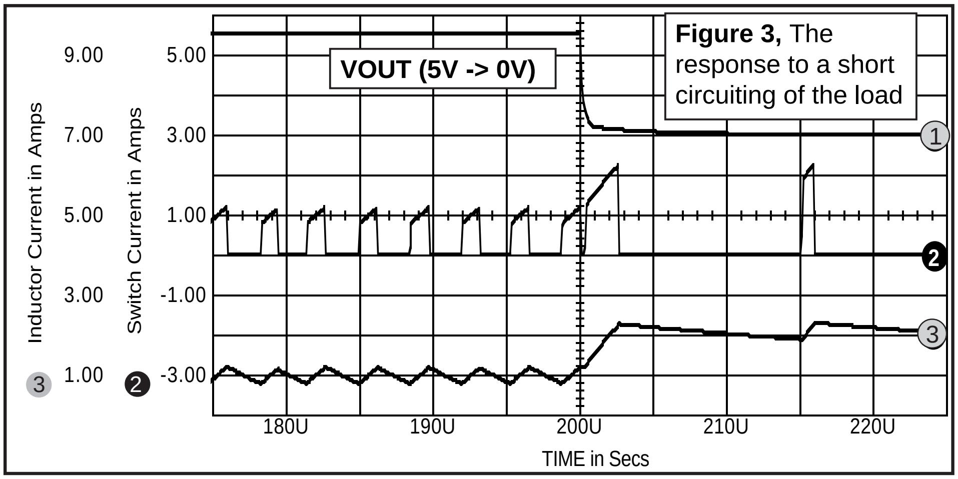


Figure 3, The response to a short circuiting of the load

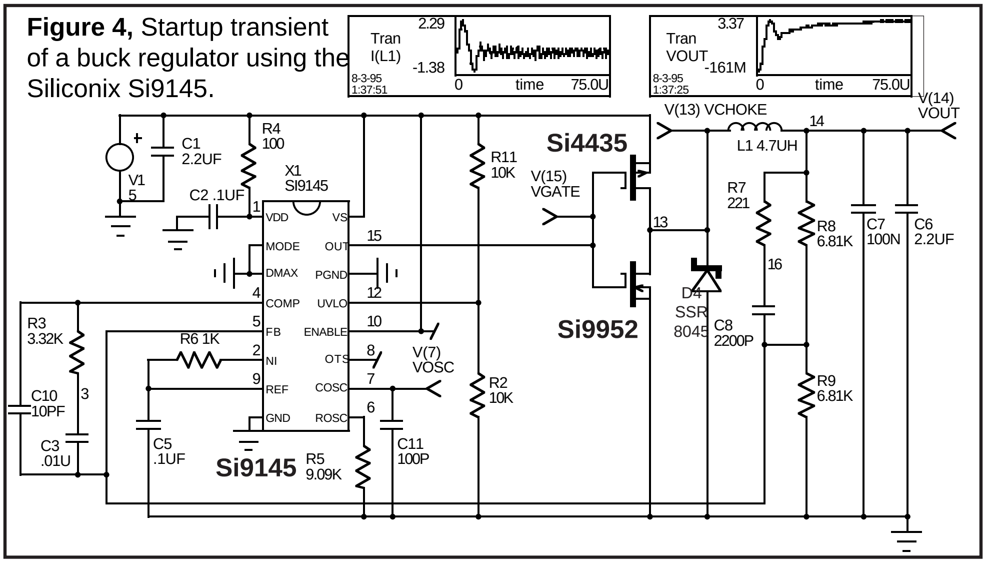


Figure 4, Startup transient of a buck regulator using the Siliconix Si9145.

time down. This simulation (.tran 50n 75n 0 10n) takes only 450s on a 486/25. Starting the simulation presented some problems. The .tran UIC keyword is used to initialize the internal logic in the 9145. Unless the VDD capacitors (C1/C2) are initialized with ICs (IC=5V) the VDD pin will start low and ramp to 5V and the switching action will not begin.

New Simulation Tutorial Book for SMPS Designers

For those interested in pursing more in-depth SMPS simulation techniques and modeling, Intusoft will be releasing a book on the subject entitled "**SMPS Simulating with SPICE 3**" in the near future. Stay tuned to the newsletter for more information.

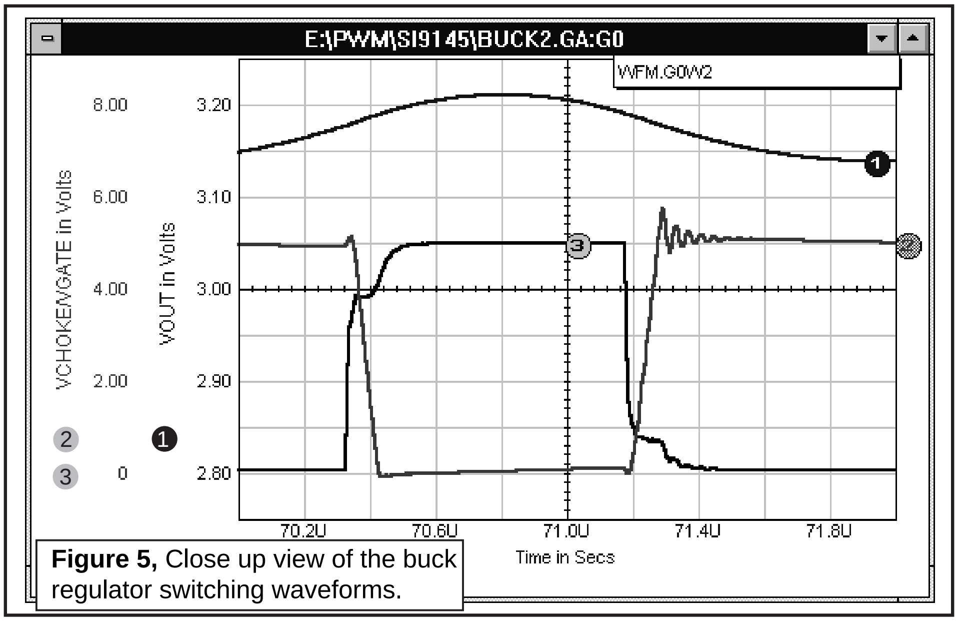


Figure 5, Close up view of the buck regulator switching waveforms.

Thanks to Steve Sandler, Analytical Engineering (602) 890-7191 for his contributions to this article and the Power Supply Library. Steve is a design consultant specializing in SMPS simulation, modeling, and design.

The Intusoft Modeling Corner

In this edition of the modeling corner we present models from **SSDI** and **Intusoft Tech support**. The *Intusoft Newsletter* subscription disk contains new models from Analog Devices, plus the models applicable to the switching converter example in Figure 10; a TL431 voltage reference, over 100 TIP Power BJT models, and a number of schottky diodes. Over 40 Asian-Pacific BJT and FET models are also included.

SSDI Provides IsSpice4 Rectifier Models

Solid State Devices Inc. (Irvine, CA, (714) 670-7734) has created models of their SRMx, SDR937, and SSR8045 (0.4V drop @ 20A) power diodes. SSDI specializes in Power semiconductors. The SPICE model netlists are shown in Figure 9. The diode models were made using average measurements from 10 randomly selected parts of each type. More models are forthcoming and will be available on future *Intusoft Newsletter* floppies for subscribers.

How Accurate Are The Models?

The accuracy of the models is controlled by the SPICE parameters. Forward voltage, including temperature effects, is characterized by the IS, N, RS, XTI, and EG parameters. The following table provides the actual measured and simulated data along with the error. All of the simulations were performed using the ICAP/4Windows software.

IF (A)	Simulated VF	Measured VF	%Error
5	0.739	0.738	-0.042
10	0.787	0.786	-0.116
20	0.838	0.838	-0.044
50	0.92	0.92	0.040
100	1.002	1.002	-0.079

```
.MODEL SRM1UF D (IS=4.9E-5 RS=.77M N=2.45 TT=65.6N BV=303.5
+IBV=100U M=.252 CJO=669.7P VJ=.75 XTI=4) ; 100V 20A

.MODEL SRM3UF D (IS=9.7E-5 RS=.74M N=2.78 TT=65.6N BV=486
+IBV=100U M=.102 CJO=211.1P VJ=.75 XTI=4) ; 300V 20A

.MODEL SRM5 D (IS=.0210 N=7.5 RS=.71M TT=1.344E-7 BV=1111
+IBV=100U M=.348 CJO=467.4P VJ=.75 XTI=28) ; 500V 20A

.MODEL SRM6UF D (IS=59.6M RS=.63M N=11.5  TT=129.2N BV=1237
+IBV=100U M=.410 CJO=709.1P VJ=.75 XTI=58) ; 600V 20A

.MODEL SRM5SOFT D (IS=4.95E-7 RS=1.1M N=1.96 TT=11.52U BV=1128
+IBV=100U M=.39 CJO=678.3P VJ=.75 XTI=4) ; 500V 20A

.MODEL SSR8045 D (IS=26M RS=3.57M N=2 BV=50 IBV=5M
+CJO=2.86N VJ=.75 M=.333 TT=14.4P ; 45V 40A

.MODEL SDR937 D(IS=8.359E-6 N=2.458 RS=1.93M CJO=600P M=0.6 VJ=0.34
+ IBV=100U BV=674 TT=30N EG=1.15) ; 700V 100A
```

Figure 9, SSDI has created several new IsSpice4 diode models for their line of power devices.

High Efficiency Step-Down Converter

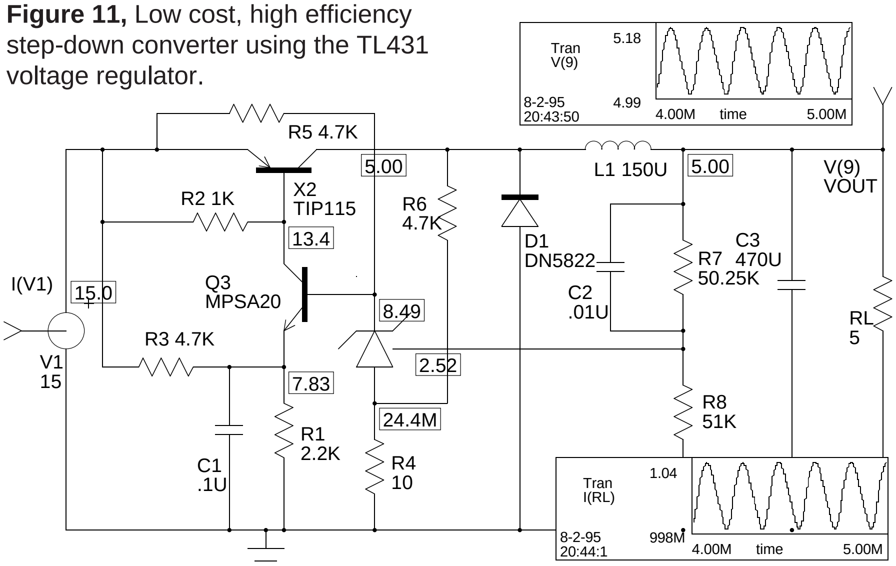
```
.SUBCKT TL431     7  6   11
*                 K  A  FDBK
.MODEL DCLAMP D (IS=13.5N RS=25M N=1.59
+ CJO=45P VJ=.75 M=.302 TT=50.4N BV=34 IBV=1M)
V1 1 6 2.495
R1 6 2 15.6
C1 2 6 .5U
R2 2 3 100
C2 3 4 .08U
R3 4 6 10
G2 6 8 3 6 1.73
D1 5 8 DCLAMP
D2 7 8 DCLAMP
V4 5 6 2
G1 6 2 1 11 0.11
.ENDS
```

Figure 10, The SPICE 2 model for the TL431 regulator.

The TL431 programmable precision reference (Figure 10) may be used to implement a low cost stepdown switching converter. The TL431 performs the function of both a voltage reference and a voltage comparator, all contained within a three pin package. The programmable output voltage feature permits a wide range of output voltages by changing one resistor value. The test circuit (Figure 11) achieved a simulated efficiency of over 70% for a 5 volt output. Other simulated results are shown in the figure, along with measured results from the Motorola Linear IC data book. Initial conditions were set on the output LC filter elements. The circuit did not require these initial conditions to start the circuit switching, but did require that the maximum transient time step be specified. The IsSpice4 simulation revealed that the converter operation is very sensitive to parasitics which explain some of the differences between simulated and measured results.

Test	**Conditions**	**Simulated Results**	**Measured Results**
Line Regulation	Vin=10-20V Io=1A	200.2mV	53mV
Output Ripple	Vin=10V Io=1A	41.83mVp-p	50mVp-p
Output Ripple	Vin=20V Io=1A	125.58mVp-p	100mVp-p
Efficiency	Vin=15V Io=1A	70%	82%

Figure 11, Low cost, high efficiency step-down converter using the TL431 voltage regulator.



Intusoft Newsletter

Personal Computer Circuit & System Design Tools

intusoft

Issue #47 Aug. 1996
Tel. (310) 833-0710
Fax (310) 833-9658

IsSpice4 Scripting Gives You More Power

By Scott Frankel, Analytical Engineering

One of the most powerful features of Intusoft's SPICE is the Interactive Command Language (ICL). Unfortunately, it is also one of the most unused features. This article will attempt to illustrate the potential capabilities of ICL by demonstrating its ability to make complex simulations easy. ICL is basically a macro or scripting language. You can use ICL to perform complex tasks, or test procedures, using a series of commands rather than by pulling down a menu or by double clicking a part and changing its value. The ICL contains over 60 commands that allow you to:

- Sweep any circuit variable (resistance, voltage, temperature, model parameters, etc.)
- Watch for any operating condition or combination of circuit variables and make decisions on what test path to choose based on the results
- Set Simulation Breakpoints on any circuit condition
- Run any SPICE analysis, in any order, as many times as desired
- Record data from anywhere in the circuit (curve families, power dissip., complex measurements, etc.)

IsSpice4 Scripting Gives You More Power
Scott Frankel
Excerpts from the August 1996 Intusoft Newsletter #47

ICL scripts can be run in one of two ways; either batch style or interactively. If the netlist is placed in the input .CIR file, IsSpice4 will auto-run the entire script when the simulator is started. If the script is placed in the Simulation Control Window inside IsSpice4, then the script can be run interactively, line by line. The latter method is useful for script debugging. In both cases, the simulation is not limited to only 1 AC, DC, or transient analysis. Any number of analyses can be run. Two example scripts are shown below.

Parameter Sweeping - ICL Script Example #1

```
view tran v(3)              Display waveform in real time
foreach param -2  -4  -6    Parameter values for dummy variable “param”
alter @V7[DC]=$param        Change the DC value of V7 to equal param
tran 1n 200n                Run a transient analysis
print v(3)                  Save the data for V(3)
end                         Do next parameter (if any)
```

Repeating Loop W/Breakpoints - ICL Script Example #2

```
save all allcur allpow                     Save data
view tran v(3)                             Real time display
function rms(vec) sqrt(mean(vec*vec))      Create a function
dowhile rms(v(3)) > 400m                   Do script while V(3) > .4
stop when @q5[p]<.25 when V(8)>1.6         Break if OR condition is true
tran 1n 100n                               Run a transient analysis
alter @q5[temp]=@q5[temp]+10               Increment the temperature
print @q5[temp] rms(v(3))                  Save data
end                                        End loop
```

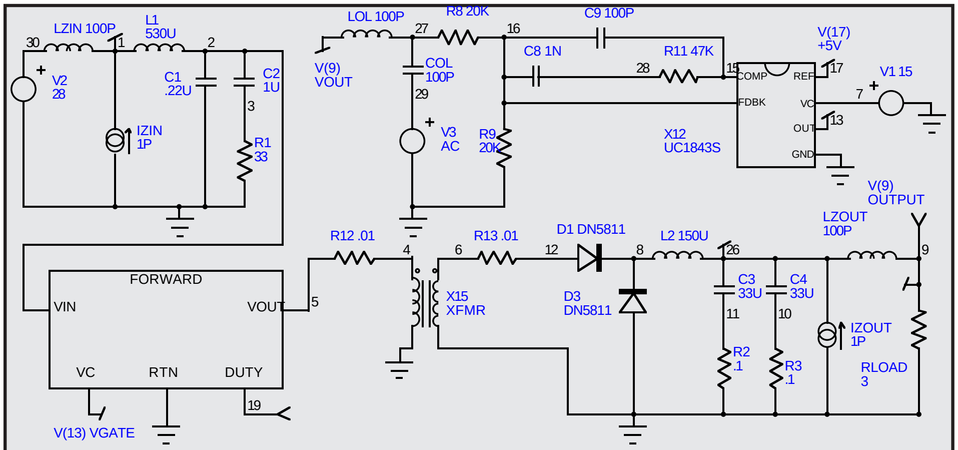


Figure 10, Schematic of a buck converter. The simulation is driven by the ICL script in Figure 11.

The circuit in Figure 10 is a forward buck derived converter. You will notice that there are several inductors, capacitors, and AC current and voltage sources in the schematic that do not affect the operation of the circuit. We will use these components to measure the frequency response, and input and output impedance with a single simulation. The ICL script is shown in Figure

```
*INCLUDE ICL.TST ; Get script (.Control -> *.PRINT lines) out of the ICL.TST file
.CONTROL ; Start of the Included ICL script
SAVE ALL ; Save all node voltages, necessary to print out waveforms later
*The next two lines effectively open the control loop so we can check the loop response
ALTER @LOL[INDUCTANCE]=100 ; Set LOL = 100 Henries
ALTER @COL[CAPACITANCE]=100 ; Set COL = 100 Farads
ALIAS GAIN VDB(9) ; Measure the gain in DB at node 9
ALIAS PHASE VP(9) ; Measure the phase at node 9
AC DEC 20 10 1000K ; Perform an AC analysis
PRINT GAIN PHASE ; Save the Gain and Phase waveforms to the output file
* The next section of code allows the circuit to measure input impedance
ALTER @LOL[INDUCTANCE]=10P ; Set LOL = 10pH in order to close the loop
ALTER @COL[CAPACITANCE]=10P ; Set COL = 10pF in order to close the loop
ALTER @LZIN[INDUCTANCE]=100  ; Set LZIN = 100H in order to inject an AC signal
* into the input in order to measure input impedance
ALTER @IZIN[ACMAG]=1 ; Inject an AC current of 1 (magnitude)
ALTER @Izout[ACMAG]=0 ; Turn off AC current source on the output of the converter
ALIAS ZIN VDB(1) ; Measure the gain in DB at node 1
AC DEC 20 10 100K ; Perform another AC analysis
PRINT ZIN ; Saves the input impedance waveform to the output file
*The next section of code allow the circuit to measure output impedance
ALTER @LZIN=1P ; Set LZIN = 1PH
ALTER @IZIN[ACMAG]=0  ; Change input AC magnitude to zero
ALTER @Izout[ACMAG]=1  ; Change output AC magnitude to 1A
ALIAS ZOUT VDB(26) ; Measure the gain in DB at node 26
AC DEC 20 10 100K ; Perform another AC analysis
PRINT ZOUT ; Save the output impedance waveform to the output file
.ENDC ; End the ICL control block
*.PRINT AC GAIN PHASE ; Optional, Generate IntuScope Waveform List
*.PRINT AC ZOUT
*.PRINT AC ZIN
```

Figure 11, An ICL script (with explanations) used by IsSpice4 to run a series of unattended AC simulations on the forward converter.

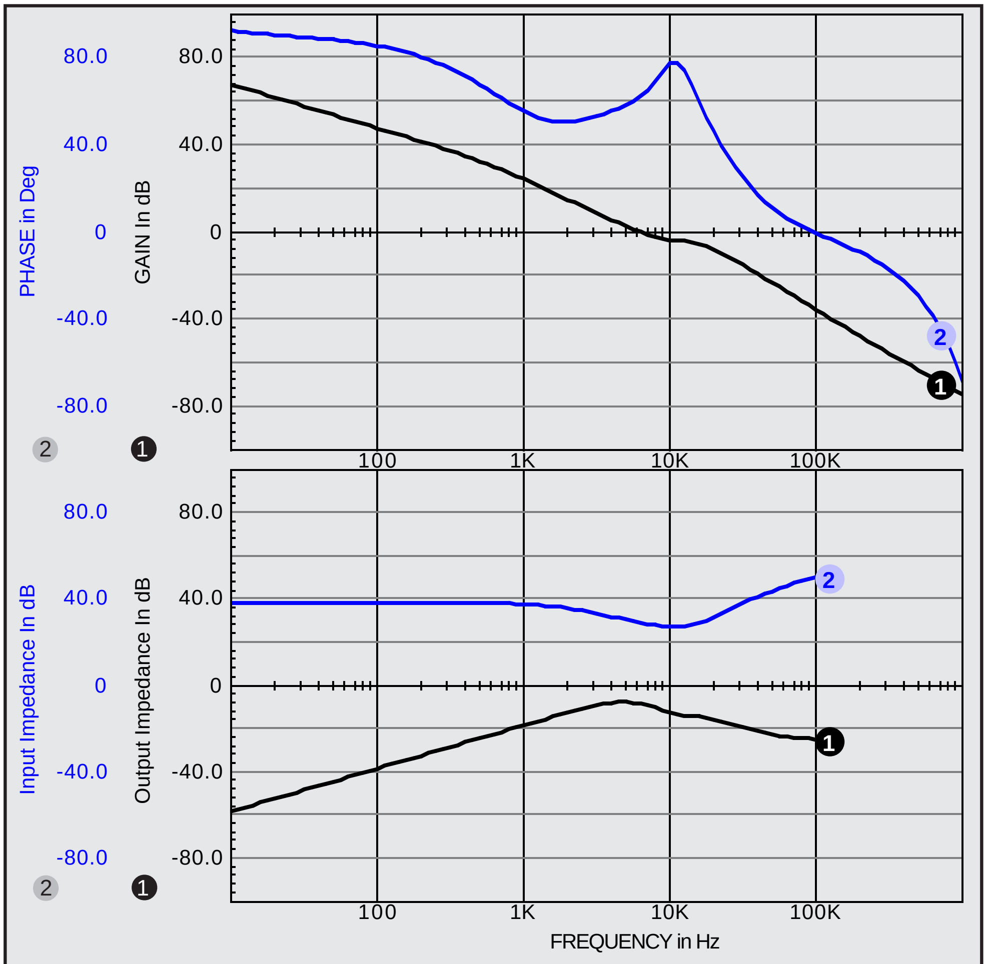


Figure 12, Gain Phase plot (top) and Input/Output Impedance plot (bottom).

11. The script, called by the ∗INCLUDE statement, is appended to the SPICE netlist and run when IsSpice4 is started. In this batch style mode, the .Control and .Endc statements are required. The results of the ICL script are shown in Figure 12.

Remember, without ICL, these results would have taken three separate SPICE runs with three separate schematics and three separate netlists. The simple demonstration presented here is only a hint of the abilities of ICL. This capability could be used to run a Monte Carlo analysis or an Optimizer routine and return numerous parameters using only one IsSpice4 simulation file! Clearly the potential benefits of ICL suggest more investigation into this powerful tool. Read the chapter on ICL in your IsSpice4 manual and start saving time today.

Scott Frankel graduated from Arizona State University with a BSEE degree. He currently works at Analytical Engineering Incorporated (602) 890-7197. AEI offers engineering design, simulation, and circuit modeling services.

Magnetics Design and Modeling

Authors
Robert Martinelli, consultant, Analytic Artistry
Charles Hymowitz, Intusoft

Designing a 12.5W 50kHz Flyback Transformer

Robert Martinelli, Charles Hymowitz, Intusoft, USA, e-mail: charles@intusoft.com

October 1998

Magnetics Designer is a versatile tool that designs all types of layer wound transformers and inductors. It is unique in the marketplace. The frequency range of the magnetics that can be designed extends from DC to over 5MHz. Unlike finite element analysis (FEA) programs Magnetics Designer is a synthesis tool as well as an analysis tool. With FEA based tools you must enter far more information about the design than you usually have, especially if you are in the initial design stages. In addition, the FEA analysis times can be quite lengthy, even hours. This negates the interactive nature of the design process. Magnetics Designer, on the other hand, operates virtually instantly allowing far more optimizations and "what-if" scenarios to be explored.

With Magnetics Designer all you need to design a transformer is the operating frequency, temperature rise, core type, and the desired voltage and current for each winding. Magnetics Designer does the rest through it sophisticated synthesis algorithms. The result is a fully designed and characterized magnetic meeting all of your specifications.

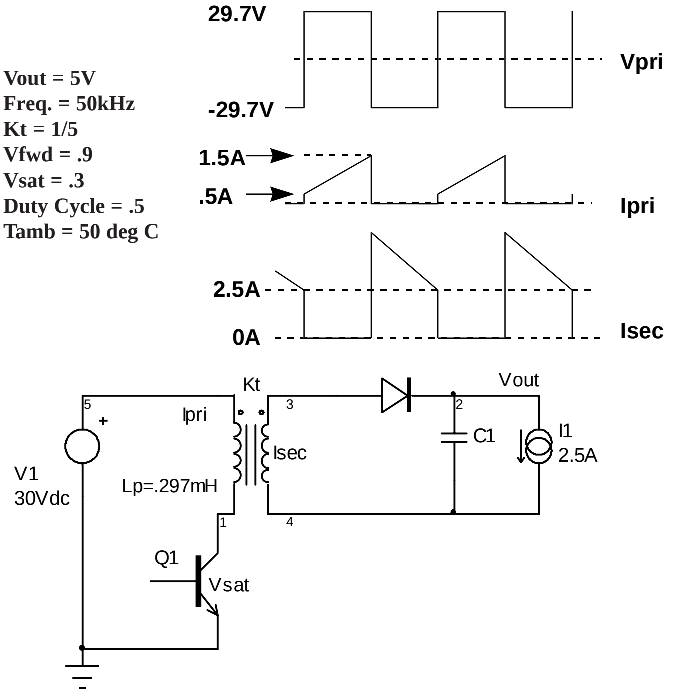
Here we will take a look at how Magnetics Designer can be used to design a flyback transformer. Although it's called a transformer, the principal magnetic component in a flyback regulator is actually an inductor. It is best described as an inductor because energy is intentionally stored in the core or core gap. Therefore, we will use the inductor design features of Magnetics Designer to create a flyback transformer.

The first step is to determine the specifications. A flyback regulator is designed to operate in the discontinuous mode. That is, the flux in the flyback transformer (inductor) returns to 0 on every cycle, as evidenced by the fact that the current on both the primary and secondary are equal to zero during a part of the switching cycle. During the "on" time of the transistor, energy is stored in the inductor. During the "off" time of the transistor, energy is released to the output.

The required parameters for the (inductor) flyback design are: core type and material, operating frequency, Edt (volt-seconds), primary and secondary (AC and DC) currents, required inductance, and the peak primary current.

The type of core we will use is the EI Ferrite from TDK. Magnetics Designer includes a database of over 7000 cores and dozens of materials including newly developed exotic materials.

The Edt across the primary winding is found from the equation:



Vout = 5V
Freq. = 50kHz
Kt = 1/5
Vfwd = .9
Vsat = .3
Duty Cycle = .5
Tamb = 50 deg C

Figure 1, The basic voltage and current waveforms and structure of a flyback regulator.

$$Edt = \frac{DVpk}{fsw} = \frac{(.5)(29.7)}{50kHz} = 2.97x10^{-4}V - \sec$$

Although, the primary winding peak current (see Figure 1) is 1.5 Apk, the switching regulator usually has a current limit which is 10 to 20% greater than the maximum load condition. Therefore, we choose Ipk = 1.8 Apk. Since the turns ratio is 1/5, the peak current reflected to the secondary is 9 Apk. The average (DC component of current) on the primary is:

$$Idc(pri) = \frac{D(Ipk + Imin)}{2} = \frac{.5(.5 + 1.5)}{2} = .5Adc$$

On the secondary, the average current is

$$Idc(\sec) = \frac{(1 - D)(Ipk + Imin)}{2} = \frac{.5(7.5 + 2.5)}{2} = 2.5Adc$$

It can be shown that the RMS value of a trapezoidal waveform is found from the following equation:

$$Irms = \sqrt{D\left[(Ipk \cdot Imin) + \frac{1}{3}(Ipk - Imin)^2\right]}$$

Likewise, using principles derived from Fourier analysis,

$$I^2 rms = I^2 dc + I^2 ac$$

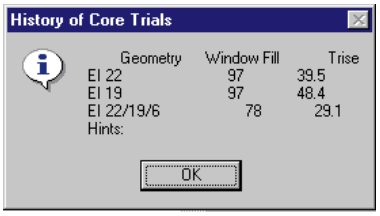
where Iac is the RMS value of the AC component of current (i.e. the RMS value of I(t)-Idc).

Using the Irms equations above, the RMS current for the primary and secondary windings is calculated to be .746 and 3.68 Arms respectively. Solving for Iac, the AC currents were calculated to be .54 and 2.7 Arms respectively.

The inductance on the primary was originally specified as .297 mH. Also the turns ratio was defined to be 1/5. Therefore, the inductance on the secondary is 11.9 uH.

The requirements were entered on the Inductor screen (Figure 2). The ambient temperature was set to 50C in the Options tab. The maximum allowed temperature rise is set 30 degrees C. Litz wire was chosen due to the high operating frequency.

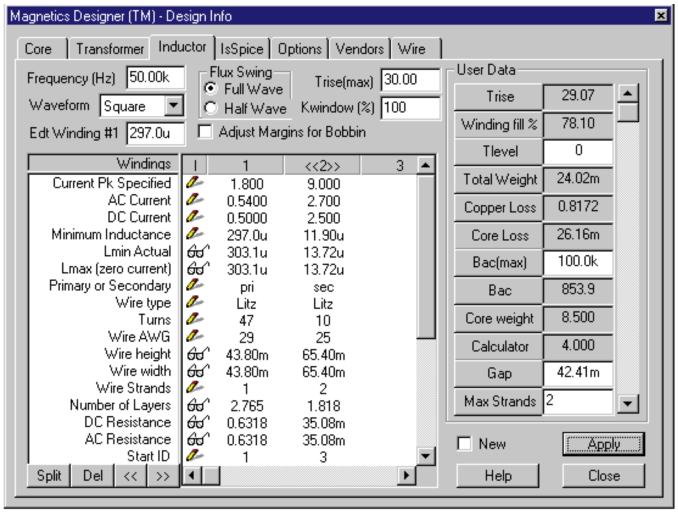
After the Apply button is clicked the program performs thousands of calculations. It tries different stranding arrangements, different wire sizes, and so forth, in an effort to generate a design with the minimum power losses in the smallest core possible, that still meets the fill and temperature rise specifications. Magnetic Designer provides a list of the cores it tried along with their respective fills and temperature rises. The last geometry is the one the program finally settled on. The total time to calculate the full design specifications on a Pentium 133 is under 1 second! The initial design, shown in Figure 3, generally met the design requirements with the exception that the turns ratio was not exactly 1/5.



History of Core Trials

Geometry	Window Fill	Trise
EI 22	97	39.5
EI 19	97	48.4
EI 22/19/6	78	29.1

Hints:



Magnetics Designer (TM) - Design Info

Core | Transformer | Inductor | IsSpice | Options | Vendors | Wire

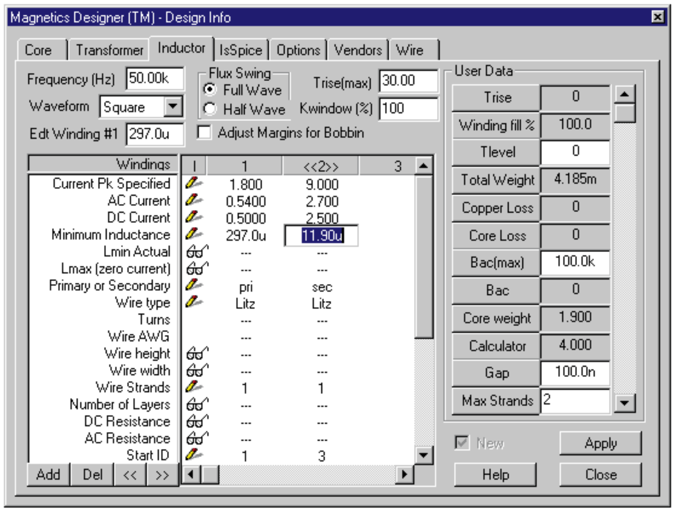
Frequency (Hz) 50.00k
Waveform Square
Edt Winding #1 297.0u
Flux Swing ● Full Wave ○ Half Wave
Trise(max) 30.00
Kwindow (%) 100
□ Adjust Margins for Bobbin

Windings	1	<<2>>	3
Current Pk Specified	1.800	9.000	
AC Current	0.5400	2.700	
DC Current	0.5000	2.500	
Minimum Inductance	297.0u	11.90u	
Lmin Actual	303.1u	13.72u	
Lmax (zero current)	303.1u	13.72u	
Primary or Secondary	pri	sec	
Wire type	Litz	Litz	
Turns	47	10	
Wire AWG	29	25	
Wire height	43.80m	65.40m	
Wire width	43.80m	65.40m	
Wire Strands	1	2	
Number of Layers	2.765	1.818	
DC Resistance	0.6318	35.08m	
AC Resistance	0.6318	35.08m	
Start ID	1	3	

User Data	
Trise	29.07
Winding fill %	78.10
Tlevel	0
Total Weight	24.02m
Copper Loss	0.8172
Core Loss	26.16m
Bac(max)	100.0k
Bac	853.9
Core weight	8.500
Calculator	4.000
Gap	42.41m
Max Strands	2

Figure 3, The initial design using Litz wire. The information for each winding is shown in the center of the screen under "Windings 1 and 2". The performance of the whole design is shown on the right side.

The window fill algorithm calculates the build assuming that two windings will not share space on a single layer. That is, if a winding has 1.1 layers, the build algorithm calculates the height of that winding as if it consumed two full layers. Since the primary winding occupies 2.765 layers and the secondary occupies 1.818 layers the window is not utilized to its fullest efficiency.



Magnetics Designer (TM) - Design Info

Core | Transformer | Inductor | IsSpice | Options | Vendors | Wire

Frequency (Hz) 50.00k
Waveform Square
Edt Winding #1 297.0u
Flux Swing ● Full Wave ○ Half Wave
Trise(max) 30.00
Kwindow (%) 100
□ Adjust Margins for Bobbin

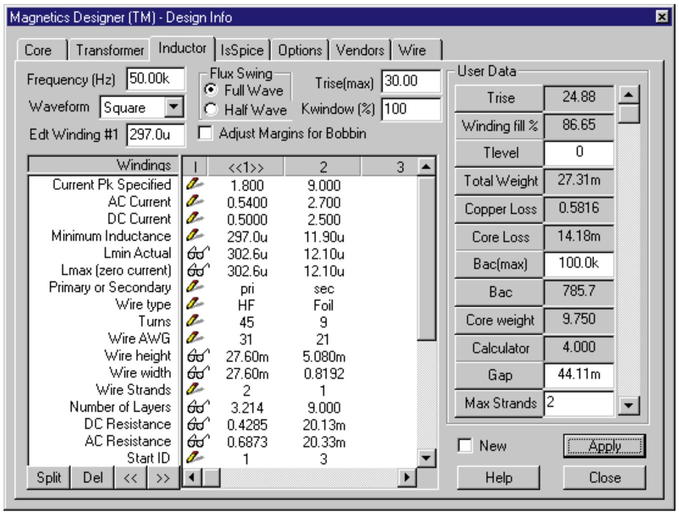
Windings	1	<<2>>	3
Current Pk Specified	1.800	9.000	
AC Current	0.5400	2.700	
DC Current	0.5000	2.500	
Minimum Inductance	297.0u	11.90u	
Lmin Actual	---	---	
Lmax (zero current)	---	---	
Primary or Secondary	pri	sec	
Wire type	Litz	Litz	
Turns	---	---	
Wire AWG	---	---	
Wire height	---	---	
Wire width	---	---	
Wire Strands	1	1	
Number of Layers	---	---	
DC Resistance	---	---	
AC Resistance	---	---	
Start ID	1	3	

User Data	
Trise	0
Winding fill %	100.0
Tlevel	0
Total Weight	4.185m
Copper Loss	0
Core Loss	0
Bac(max)	100.0k
Bac	0
Core weight	1.900
Calculator	4.000
Gap	100.0n
Max Strands	2

Figure 2, The inductor screen with frequency, Edt, maximum allowable temperature rise, current peak, AC/DC current, and minimum inductance entered. The Apply button generates the rest of the flyback design.

Turns are always rounded to the nearest integer by the program. Therefore, some slight adjustments are required to achieve exact turns ratios. For example, if the program calculates that 46.5 turns are required on the primary and 9.6 turns are required on the secondary, the primary will have 47 turns and the secondary will have 10 turns. The designer would then have to adjust the turns on the primary to achieve and exact turns ratio of 5/1.

Several design changes were tried to improve the inductor design. It was originally assumed that Litz wire was necessary to avoid excessive losses due to potential high frequency AC resistance problems (skin and proximity effects). To test this assumption, heavy formvar (HF) wire was substituted for the Litz wire. This resulted in a significant temperature rise increase. After closely reviewing the results, it was determined that this problem was primarily due to increased AC resistance on the secondary side. Therefore, foil windings were substituted for the magnet wire on the secondary. Foil windings typically have very low AC resistance since the thickness required to achieve the desired cross-sectional area is usually much thinner than the skin depth of corresponding wire. Since a foil winding was used, the program adds layer insulation, wrappers, and end margins to prevent shorts between turns and other windings.



Frequency (Hz) 50.00k
Waveform Square
Edt Winding #1 297.0u
Flux Swing ● Full Wave ○ Half Wave
Trise(max) 30.00
Kwindow (%) 100
□ Adjust Margins for Bobbin

Windings	<<1>>	2	3
Current Pk Specified	1.800	9.000	
AC Current	0.5400	2.700	
DC Current	0.5000	2.500	
Minimum Inductance	297.0u	11.90u	
Lmin Actual	302.6u	12.10u	
Lmax (zero current)	302.6u	12.10u	
Primary or Secondary	pri	sec	
Wire type	HF	Foil	
Turns	45	9	
Wire AWG	31	21	
Wire height	27.60m	5.080m	
Wire width	27.60m	0.8192	
Wire Strands	2	1	
Number of Layers	3.214	9.000	
DC Resistance	0.4285	20.13m	
AC Resistance	0.6873	20.33m	
Start ID	1	3	

User Data

Trise	24.88
Winding fill %	86.65
Tlevel	0
Total Weight	27.31m
Copper Loss	0.5816
Core Loss	14.18m
Bac(max)	100.0k
Bac	785.7
Core weight	9.750
Calculator	4.000
Gap	44.11m
Max Strands	2

□ New

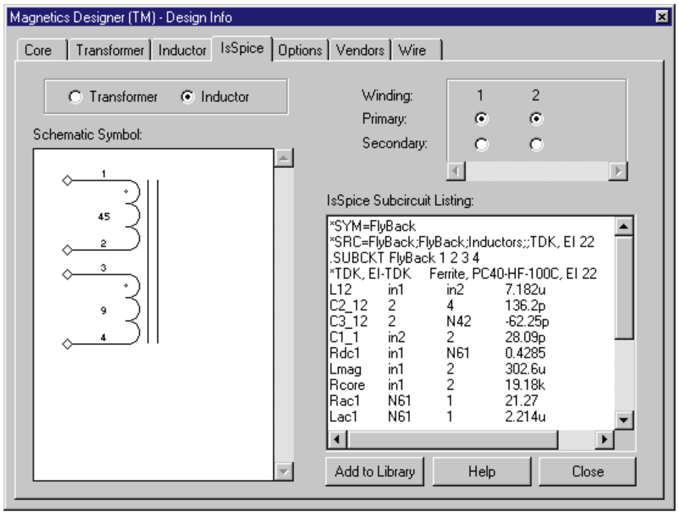
Figure 4, The flyback design using heavy formvar and Foil windings. Here, the Windings window show the required turns, wire size, strands, layers, and AC/DC resistance. The User Data fields show the temperature rise, fill, weight, losses, and flux density.

Next, the number of turns on the primary was set to 45 so that the turns ratio from primary to secondary would be exactly 5/1. To maintain the inductance at approximately .3 mh, the Gap was increased to .044 cm. The result of these changes are shown in Figure 4.

The final design meeting all specification is shown in Figures 4 and 5. It has a slightly lower peak flux, lower temperature rise, lower AC and DC resistance, and uses a smaller geometry than the initial design.

At this point you explore the design further by splitting windings or trying different materials. A winding sheet and design summary report can be printed out or copied to Microsoft Excel. As shown in Figure 6, Magnetics Designer can also produce a SPICE model and schematic symbol for the magnetic. This allows you to simulate the device while it is being built. The SPICE model can be used with ANY SPICE program.

Magnetics Designer provides a simple and fast way to design custom transformers and inductors, a key part of virtually all electronic systems and many EMI filtering applications. The software even designs planar magnetics, an important part of many new high frequency applications. Magnetic design is an art that is slowly being lost as experienced designers retire. Products like Magnetics Designer help fill the void by providing intelligence and design expertise in a software tool.



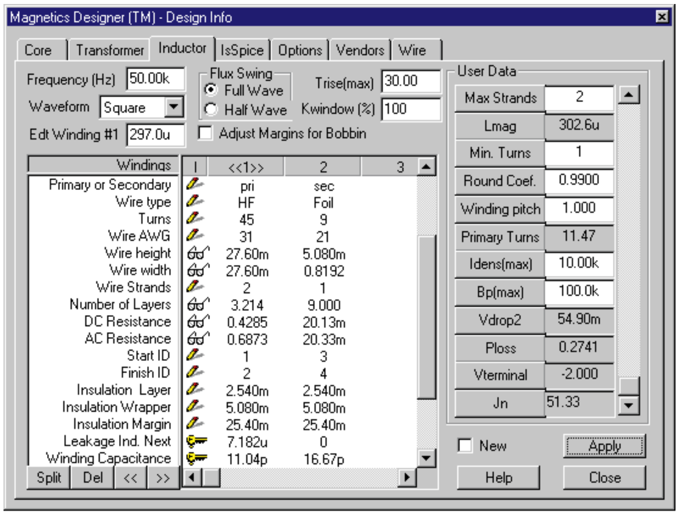
○ Transformer ● Inductor

Winding:	1	2
Primary:	●	●
Secondary:	○	○

IsSpice Subcircuit Listing:

```
*SYM=FlyBack
*SRC=FlyBack;FlyBack;Inductors;;TDK, EI 22
.SUBCKT FlyBack 1 2 3 4
*TDK, EI-TDK    Ferrite, PC40-HF-100C, EI 22
L12     in1     in2     7.182u
C2_12   2       4       136.2p
C3_12   2       N42     -62.25p
C1_1    in2     2       28.09p
Rdc1    in1     N61     0.4285
Lmag    in1     2       302.6u
Rcore   in1     2       19.18k
Rac1    N61     1       21.27
Lac1    N61     1       2.214u
```

Figure 6, A SPICE model (SPICE 2G syntax) can be produced along with a reconfigurable schematic symbol.



Frequency (Hz) 50.00k
Waveform Square
Edt Winding #1 297.0u
Flux Swing ● Full Wave ○ Half Wave
Trise(max) 30.00
Kwindow (%) 100
□ Adjust Margins for Bobbin

Windings	<<1>>	2	3
Primary or Secondary	pri	sec	
Wire type	HF	Foil	
Turns	45	9	
Wire AWG	31	21	
Wire height	27.60m	5.080m	
Wire width	27.60m	0.8192	
Wire Strands	2	1	
Number of Layers	3.214	9.000	
DC Resistance	0.4285	20.13m	
AC Resistance	0.6873	20.33m	
Start ID	1	3	
Finish ID	2	4	
Insulation Layer	2.540m	2.540m	
Insulation Wrapper	5.080m	5.080m	
Insulation Margin	25.40m	25.40m	
Leakage Ind. Next	7.182u	0	
Winding Capacitance	11.04p	16.67p	

User Data

Max Strands	2
Lmag	302.6u
Min. Turns	1
Round Coef.	0.9900
Winding pitch	1.000
Primary Turns	11.47
Idens(max)	10.00k
Bp(max)	100.0k
Vdrop2	54.90m
Ploss	0.2741
Vterminal	-2.000
Jn	51.33

□ New

Figure 5, Both the Winding and User Data fields scroll revealing more information like leakage inductance, winding capacitance, weight per winding, magnetizing inductance and more.

Designing a 50W Forward Converter Transformer With Magnetics Designer

Robert Martinelli, Charles Hymowitz, Intusoft, USA, e-mail: charles@intusoft.com

October 1998

In order to introduce you to the power of Magnetics Designer, we will synthesize a transformer for a forward converter which is similar to that shown in Figure 1.

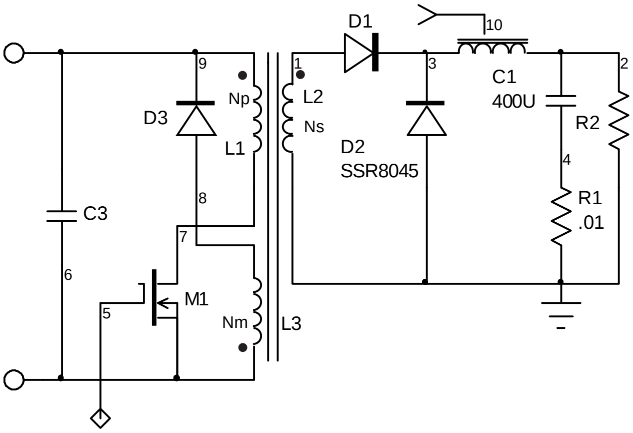


Figure 1, Magnetics Designer can synthesize transformers for designs like this forward converter.

When the transistor turns on, the voltages on the starts (dotted ends) of the transformer are driven positive, forward biasing D1. While the voltage is positive, the inductor current increases towards its maximum value while magnetizing current builds up in the transformer. When the transistor turns off, transformer magnetizing must continue to flow. Therefore, the only path for magnetizing current to flow is through D2, thus reversing the polarity across the transformer and providing a voltage to reset the flux. However, when the current in the transformer winding reaches zero, the voltage across the winding reduces to zero, indicating that the transformer flux has returned to the residual flux of the core material.

Throughout the off-time of the transistor, inductor current decays. However, if the inductance is large enough, the inductor current will not return to zero and the converter is described as operating in the continuous conduction mode. If the inductance is small, the inductor current returns to zero during the off time. The second case is described as operating in the discontinuous mode. In this example, we will assume that the converter is operating in the continuous mode. The forward converter waveforms are shown in Figure 2.

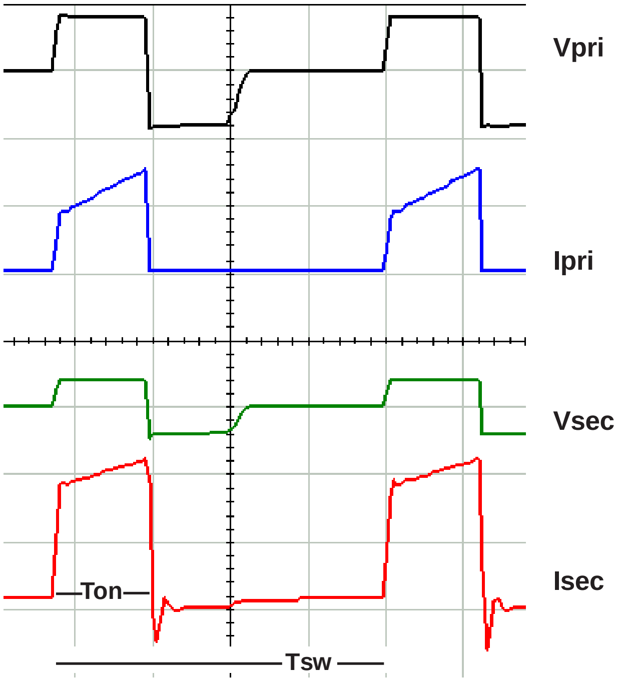


Figure 2, Key waveforms for the forward converter.

The following equations describe the converter behavior:

Output voltage: $Vo \quad = \quad Dk_t \eta\, Vin$

where Vo is the output voltage, D is the duty cycle of the switch, η is the efficiency of the converter, Vin is the input voltage, and Kt is the turns ratio transformer.

P-P Inductor Current: $\Delta I_l \quad = \quad \frac{(1-D)T_{sw}}{L}(Vo+Vfwd)$

where I_L is the peak-to-peak inductor current, Vfwd is the rectifier forward voltage, Tsw is the switching time and L is the inductance.

Ave. Inductor Current: $I_{l\,(ave)} = I_o = \frac{V_o}{R_o}$

where Io is the output current, and Ro is the load resistance.

Generally, a converter must operate over a wide dynamic range of input voltage and load current. However, Magnetics Designer only needs to consider the case which results in the maximum transformer ratings. For the forward converter, this is low line where the RMS current in the windings is maximum and duty cycle is .5 (50%). Since the average output voltage from the regulator is maintained at a constant, the core losses, on a first order basis, are unaffected by line and load changes.

The maximum required steady state output for our design is +5Vdc at 10Adc with a minimum input voltage of 40V. The transformer output voltage must, on the average, be equal to the output voltage plus the rectifier drop. Assuming that the 5 volt output uses conventional rectifiers, the transformer output voltages should be somewhat greater than 5.7V. Using the previous equations, with a frequency of 100kHz, and an 8μH inductor, the peak-to-peak inductor current is 3.56A. Ipk (Ip) is therefore 11.78A (Ip=Idc + Ip-p/2, Im=Idc - Ip-p/2). The average voltage across the secondary is (Vo+Vfwd)/D = 5V+.7V/.5 = 11.4V.

For a unipolar trapezoidal waveform, the DC current is:

$$I_{dc} = \frac{D(Ip + \mathrm{Im})}{2} = .5(11.78 + 8.22)/2 = 5\mathrm{A}$$

The RMS current in the winding is given by:

$$I_{rms} = \sqrt{D\left(I_p \times I_m + \frac{1}{3}(I_p - I_m)^2\right)} = 7.11\mathrm{A}$$

Knowing the DC and RMS currents, the AC current is:

$$I_{ac} = \sqrt{I^2_{rms} - I^2_{dc}} = 5.05\mathrm{A}$$

The required turns ratio is then Vpk-pri/Vpk-sec = 3.508. (The average voltage on the primary, which is the same as the peak voltage for a square wave, is 40 V while Vpk-sec is 11.4 V). The DC (Idc-pri) current on the primary is 1.43 Adc (=5/3.508) and the AC component of current is 1.44 Arms (=5.05/3.508).

A forward converter has a flux swing which begins at Br and achieves a maximum value, Bmax. Therefore, the flux type is half wave. The output power is 57 watts (5.7 Vdc x 10 Adc). We also assume that the converter operates at 100 kHz, that the ambient air temperature is 25 degrees C, and that the maximum desired surface temperature is 75 degrees C (50 degree rise).

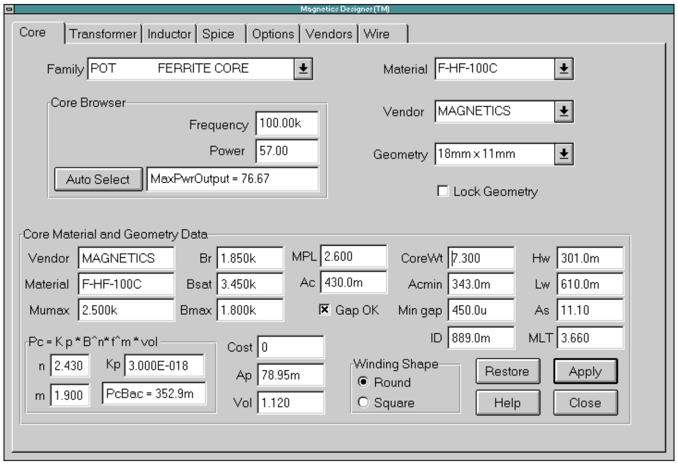
Finally, it is assumed that the inductor ripple current and transformer magnetizing current are small relative to the various winding currents and that they do not appreciably affect the RMS current in any of the transformer windings. Without this assumption, the waveforms for each winding would be more complicated, and additional effort would be required in order to calculate the transformer requirements.

The transformer's design assumptions are summarized below:

Core Type	= Pot Core, Ferrite, Magnetics
Material type	= F (High Frequency, 100 deg. C data)
Max Temp. Rise	= 50 deg C
Max. Amb. Temp.	= 25 deg C
Max. Window File	= 90%
Flux Type	= Half Wave
Output power	= 57 Watts
Frequency	= 100 kHz
Waveform Type	= Square Wave
Vpri	= 40 V ave
Idc-pri	= 1.43 Adc
Iac-pri	= 1.44 Aac (rms)
Vsec(5v)	= 11.4 V average
Idc(5v)	= 5 Adc
Iac(5v)	= 5.06 Aac (rms)
Vflyback	= 40.0 V average
Iac(40v)	= 200m Aac (rms)

Initial Computer Generated Design

The core material and family are first selected in the Core Selection screen (Figure 3). After entering the power and frequency, the Core Browser is used to make an initial core selection. The Core Browser will select the smallest core that can handle the frequency and power specified.



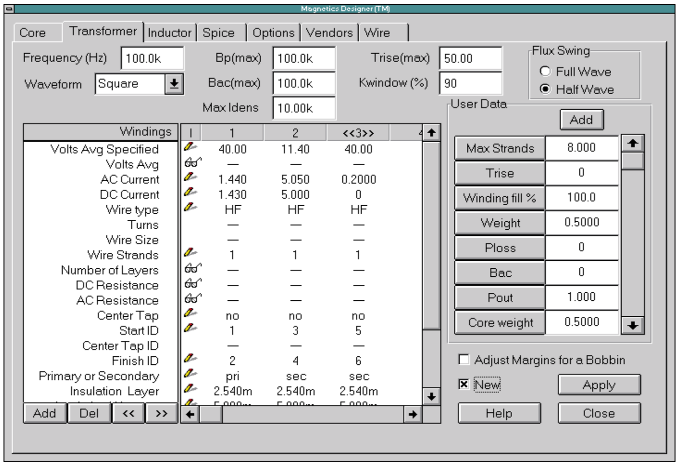
Family: POT FERRITE CORE
Material: F-HF-100C
Core Browser
Frequency: 100.00k
Power: 57.00
MaxPwrOutput = 76.67
Vendor: MAGNETICS
Geometry: 18mm x 11mm
☐ Lock Geometry

Core Material and Geometry Data
Vendor: MAGNETICS
Material: F-HF-100C
Mumax: 2.500k
Br: 1.850k
Bsat: 3.450k
Bmax: 1.800k
MPL: 2.600
Ac: 430.0m
☒ Gap OK
CoreWt: 7.300
Acmin: 343.0m
Min gap: 450.0u
ID: 889.0m
Hw: 301.0m
Lw: 610.0m
As: 11.10
MLT: 3.660

Pc = K p * B^n * f^m * vol
n: 2.430
Kp: 3.000E-018
m: 1.900
PcBac = 352.9m
Cost: 0
Ap: 78.95m
Vol: 1.120
Winding Shape
● Round
○ Square

Figure 3, The Core selection screen, after using the Core Browser to make an initial core selection.

The electrical requirements for the three windings are then entered on the Transformer design screen (Figure 4). The pencil icons indicate fields where data can be entered. The eye glasses indicate calculated "per winding" results that can be viewed. The User Data section on the right contains input and output parameters associated with the entire design. Design constraints (temperature rise, window fill, etc.) are entered at the top.



Magnetics Designer (TM)

Frequency (Hz) 100.0k
Waveform Square
Bp(max) 100.0k
Bac(max) 100.0k
Max Idens 10.00k
Trise(max) 50.00
Kwindow (%) 90
Flux Swing: ○ Full Wave ● Half Wave

Windings	1	2	<<3>>
Volts Avg Specified	40.00	11.40	40.00
Volts Avg	—	—	—
AC Current	1.440	5.050	0.2000
DC Current	1.430	5.000	0
Wire type	HF	HF	HF
Turns	—	—	—
Wire Size	—	—	—
Wire Strands	1	1	1
Number of Layers	—	—	—
DC Resistance	—	—	—
AC Resistance	—	—	—
Center Tap	no	no	no
Start ID	1	3	5
Center Tap ID	—	—	—
Finish ID	2	4	6
Primary or Secondary	pri	sec	sec
Insulation Layer	2.540m	2.540m	2.540m

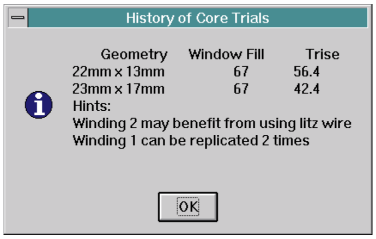
User Data

Max Strands	8.000
Trise	0
Winding fill %	100.0
Weight	0.5000
Ploss	0
Bac	0
Pout	1.000
Core weight	0.5000

□ Adjust Margins for a Bobbin
☒ New

Figure 4, The Transformer design screen after entering the basic electrical requirements.

To start the initial calculation and optimization of the transformer design, we first check the New box (bottom right, Figure 4) and then select the Apply button. The History of Core Trials dialog, in Figure 5, displays the results of the optimization process. This includes the window fill and temperature rise for each core that has been tried in the selected family.



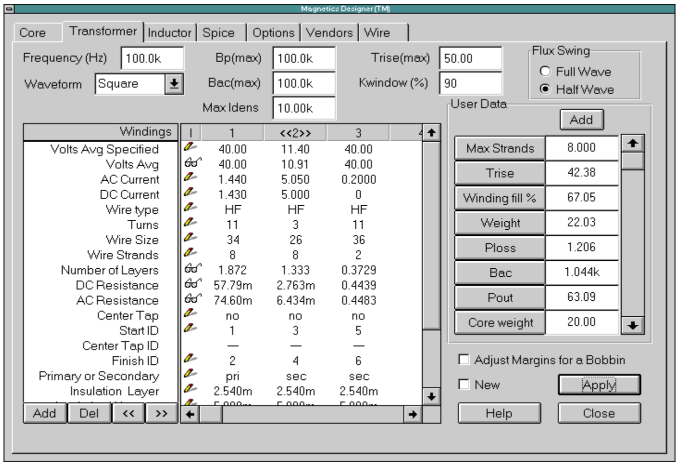
History of Core Trials

Geometry	Window Fill	Trise
22mm x 13mm	67	56.4
23mm x 17mm	67	42.4

Hints:
Winding 2 may benefit from using litz wire
Winding 1 can be replicated 2 times

Figure 5, The History of Core Trials screen shows which cores were tried by Magnetics Designer.

The resulting transformer performance for the selected core (23mm x 17mm) is then displayed in the Transformer screen.



Magnetics Designer (TM)

Frequency (Hz) 100.0k
Waveform Square
Bp(max) 100.0k
Bac(max) 100.0k
Max Idens 10.00k
Trise(max) 50.00
Kwindow (%) 90
Flux Swing: ○ Full Wave ● Half Wave

Windings	1	<<2>>	3
Volts Avg Specified	40.00	11.40	40.00
Volts Avg	40.00	10.91	40.00
AC Current	1.440	5.050	0.2000
DC Current	1.430	5.000	0
Wire type	HF	HF	HF
Turns	11	3	11
Wire Size	34	26	36
Wire Strands	8	8	2
Number of Layers	1.872	1.333	0.3729
DC Resistance	57.79m	2.763m	0.4439
AC Resistance	74.60m	6.434m	0.4483
Center Tap	no	no	no
Start ID	1	3	5
Center Tap ID	—	—	—
Finish ID	2	4	6
Primary or Secondary	pri	sec	sec
Insulation Layer	2.540m	2.540m	2.540m

User Data

Max Strands	8.000
Trise	42.38
Winding fill %	67.05
Weight	22.03
Ploss	1.206
Bac	1.044k
Pout	63.09
Core weight	20.00

□ Adjust Margins for a Bobbin
□ New

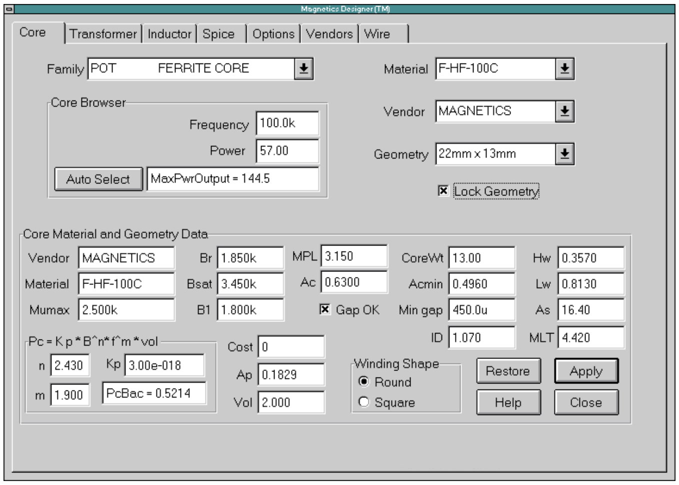
Figure 6, The Transformer screen and resulting design values after Magnetics Designer's optimization.

When the New option is selected, Magnetics Designer will iterate the design for successively larger cores until the temperature rise constraint is met or a new core can't be selected. For each core, Magnetics Designer will try to find the wire size, turns, and number of strands required to achieve the specified voltages, low AC/DC resistance, and optimal layer utilization, all while fulfilling the stated constraints of window fill and temperature rise.

As we have demonstrated in this case, the powerful algorithms in Magnetics Designer will normally produce a design that meets all stated requirements **without any further user input required**! Figure 6 shows that the layers are utilized quite well, and the Trise and Window fill values are approximately 42.38 degrees C and 67%, respectively.

While the initial design is satisfactory, it could be improved. The History of Core Trials dialog provides some recommendations on possibly improving the design. These may include changes to the wire type, number of parallel windings, and hints on smaller cores that may, with some adjustment, be able to handle the design parameters. For instance, Figure 5 shows that the smaller 22mm x 13mm core almost made the temperature rise. It may be a good candidate for further optimization.

Magnetics Designer may select a different core geometry, depending on the data changed in the Transformer screen. This geometry change may dramatically affect the overall transformer performance, and make it difficult to optimize a particular design. Therefore, it is best to "lock down" the core geometry (Figure 7).



Magnetics Designer (TM)

Family POT FERRITE CORE
Material F-HF-100C
Vendor MAGNETICS
Geometry 22mm x 13mm
☒ Lock Geometry

Core Browser
Frequency 100.0k
Power 57.00
Auto Select MaxPwrOutput = 144.5

Core Material and Geometry Data
Vendor MAGNETICS
Material F-HF-100C
Mumax 2.500k
Br 1.850k
Bsat 3.450k
B1 1.800k
MPL 3.150
Ac 0.6300
☒ Gap OK
CoreWt 13.00
Acmin 0.4960
Min gap 450.0u
ID 1.070
Hw 0.3570
Lw 0.8130
As 16.40
MLT 4.420

Pc = K p * B^n* f^m * vol
n 2.430
m 1.900
Kp 3.00e-018
PcBac = 0.5214
Cost 0
Ap 0.1829
Vol 2.000

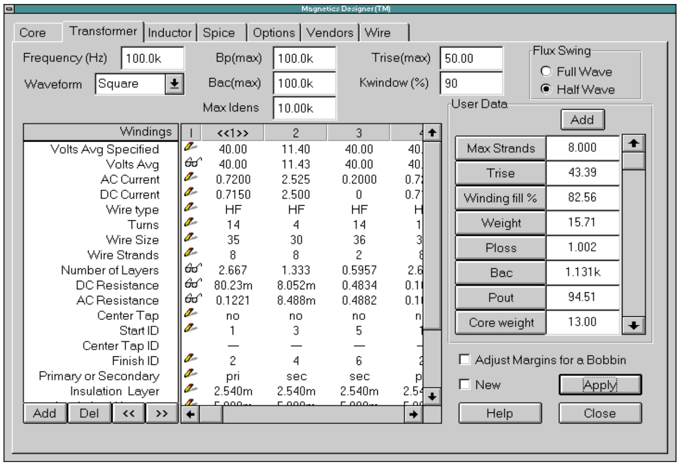
Winding Shape: ● Round ○ Square

Figure 7, You can lock a particular core geometry in order to experiment with a different design improvements.

As shown in Figure 7, the 22mm x 13mm core is selected and the "Lock Geometry" option is checked. Since we have changed the core geometry, we must first check

New and then select Apply so that Magnetics Designer can update the windings characteristics for the newly selected core. We find that the temperature rise is now 56.41 degrees, which is above our set constraint.

Let's try one of the previous suggestions and split winding 1 and 2 (once each) and then perform the New and Apply operations. Splitting the highest power winding can provide several benefits (i.e. better temperature rise for the same core area). The resulting transformer screen then looks like Figure 8.



Frequency (Hz): 100.0k
Waveform: Square
Bp(max): 100.0k
Bac(max): 100.0k
Max Idens: 10.00k
Trise(max): 50.00
Kwindow (%): 90
Flux Swing: ○ Full Wave ● Half Wave

Windings	I	<<1>>	2	3	4
Volts Avg Specified	✎	40.00	11.40	40.00	40.
Volts Avg	👓	40.00	11.43	40.00	40.
AC Current	✎	0.7200	2.525	0.2000	0.7
DC Current	✎	0.7150	2.500	0	0.7
Wire type	✎	HF	HF	HF	H
Turns	✎	14	4	14	1
Wire Size	✎	35	30	36	3
Wire Strands	✎	8	8	2	8
Number of Layers	👓	2.667	1.333	0.5957	2.6
DC Resistance	👓	80.23m	8.052m	0.4834	0.1
AC Resistance	👓	0.1221	8.488m	0.4882	0.1
Center Tap	✎	no	no	no	n
Start ID	✎	1	3	5	1
Center Tap ID		—	—	—	—
Finish ID	✎	2	4	6	2
Primary or Secondary	✎	pri	sec	sec	p
Insulation Layer	✎	2.540m	2.540m	2.540m	2.54

User Data	
Max Strands	8.000
Trise	43.39
Winding fill %	82.56
Weight	15.71
Ploss	1.002
Bac	1.131k
Pout	94.51
Core weight	13.00

□ Adjust Margins for a Bobbin
□ New
Apply

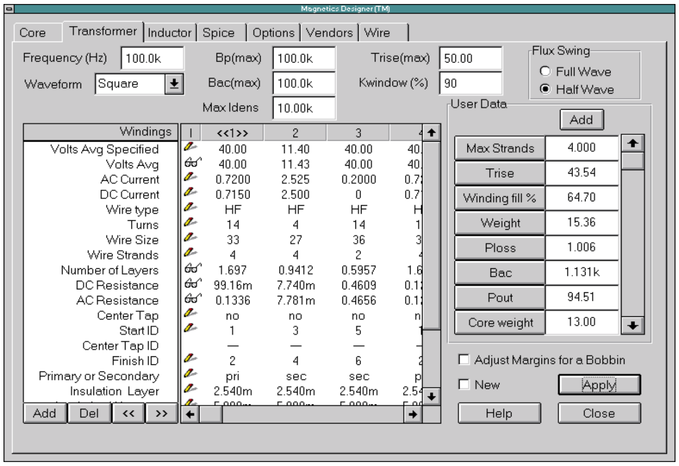
Figure 8, The transformer design after splitting the highest power winding.

Magnetics Designer knows that we are splitting a winding, rather than adding a new winding, and automatically splits the current between the two windings. **It should be noted that the AC resistance calculation includes both skin and proximity effects.** You can move the windings around and change the layer configuration by using the << and >> buttons at the bottom of transformer dialog. Magnetics Designer will recalculate the AC resistance based on the new configuration.

The simple operation of splitting the windings achieves the design goals for this smaller core. However, four of the windings now require 8 strands, which could be costly to manufacture.

We can set the maximum number of strands per winding via the Max Strands field. Located in the User data section of the Transformer screen, this series of buttons and fields provides access to both input variables and output results. Later in this article, we will explore how you can derive your own customized output results. In this case, we will change Max Strands from 8 to a more manageable 4. Magnetics Designer also allows you to change the type of wire for each windings. You have the choice of heavy formvar, small formvar, foil, square, double square, litz, or pcb traces (for planar magnetics).

Changes that affect the transformer geometrically should be accompanied by a recycling of the design using the New option. The Apply button alone (without New checked) is primarily used for minor design changes. With the New and Lock Geometry options checked, Magnetics Designer will "stir up" the windings using the selected geometry in order to arrive at the best set of characteristics.



Frequency (Hz): 100.0k
Waveform: Square
Bp(max): 100.0k
Bac(max): 100.0k
Max Idens: 10.00k
Trise(max): 50.00
Kwindow (%): 90
Flux Swing: ○ Full Wave ● Half Wave

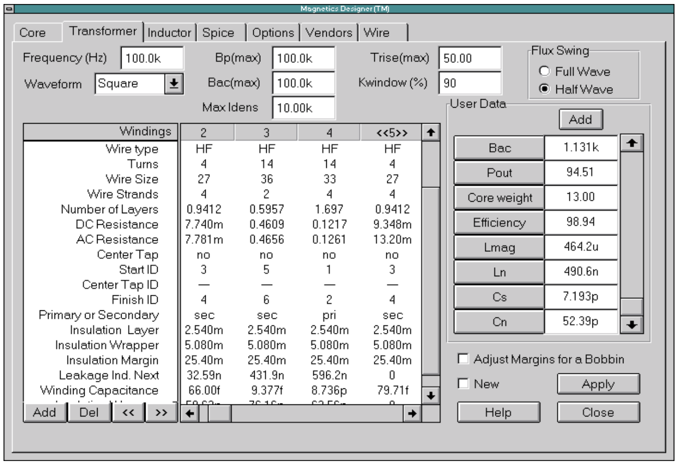
Windings	I	<<1>>	2	3	4
Volts Avg Specified	✎	40.00	11.40	40.00	40.
Volts Avg	👓	40.00	11.43	40.00	40.
AC Current	✎	0.7200	2.525	0.2000	0.7
DC Current	✎	0.7150	2.500	0	0.7
Wire type	✎	HF	HF	HF	H
Turns	✎	14	4	14	1
Wire Size	✎	33	27	36	3
Wire Strands	✎	4	4	2	4
Number of Layers	👓	1.697	0.9412	0.5957	1.6
DC Resistance	👓	99.16m	7.740m	0.4609	0.1
AC Resistance	👓	0.1336	7.781m	0.4656	0.1
Center Tap	✎	no	no	no	n
Start ID	✎	1	3	5	1
Center Tap ID		—	—	—	—
Finish ID	✎	2	4	6	2
Primary or Secondary	✎	pri	sec	sec	p
Insulation Layer	✎	2.540m	2.540m	2.540m	2.54

User Data	
Max Strands	4.000
Trise	43.54
Winding fill %	64.70
Weight	15.36
Ploss	1.006
Bac	1.131k
Pout	94.51
Core weight	13.00

□ Adjust Margins for a Bobbin
□ New
Apply

Figure 9, The transformer design after setting the maximum number of strands per winding to 4.

Figure 9 shows the resulting transformer screen which now meets all of the design goals again. The split winding information can be shown by scrolling the spreadsheet window. The ID numbers account for the paralleling of the various windings. By scrolling down the main spreadsheet window and the User Data button column on the right, we are able to see other calculated design data such as the efficiency, leakage inductance and parasitic capacitance (Figure 10).



Frequency (Hz): 100.0k
Waveform: Square
Bp(max): 100.0k
Bac(max): 100.0k
Max Idens: 10.00k
Trise(max): 50.00
Kwindow (%): 90
Flux Swing: ○ Full Wave ● Half Wave

Windings	2	3	4	<<5>>
Wire type	HF	HF	HF	HF
Turns	4	14	14	4
Wire Size	27	36	33	27
Wire Strands	4	2	4	4
Number of Layers	0.9412	0.5957	1.697	0.9412
DC Resistance	7.740m	0.4609	0.1217	9.348m
AC Resistance	7.781m	0.4656	0.1261	13.20m
Center Tap	no	no	no	no
Start ID	3	5	1	3
Center Tap ID	—	—	—	—
Finish ID	4	6	2	4
Primary or Secondary	sec	sec	pri	sec
Insulation Layer	2.540m	2.540m	2.540m	2.540m
Insulation Wrapper	5.080m	5.080m	5.080m	5.080m
Insulation Margin	25.40m	25.40m	25.40m	25.40m
Leakage Ind. Next	32.59n	431.9n	596.2n	0
Winding Capacitance	66.00f	9.377f	8.736p	79.71f

User Data	
Bac	1.131k
Pout	94.51
Core weight	13.00
Efficiency	98.94
Lmag	464.2u
Ln	490.6n
Cs	7.193p
Cn	52.39p

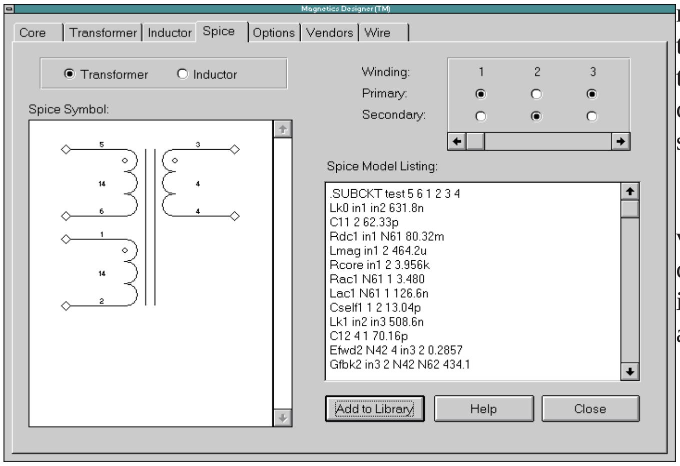
□ Adjust Margins for a Bobbin
□ New
Apply

Figure 10, Magnetics Designer displays a wealth of calculated data in the spreadsheet (left) and User Data (right) areas.

Magnetics Designer Has Unique Features

Magnetics Designer has two very unique features. The first is the ability to produce a **SPICE Model** of your transformer or inductor design. Figure 11 shows the SPICE screen which allows you to configure a schematic symbol and save the resulting SPICE 2 compatible subcircuit netlist. The model includes all the core and copper losses, AC/DC resistance, leakage and magnetizing inductance and winding capacitances. The leakage inductance is calculated based on a reluctance model while the capacitance values are based on a charge conserving representation.

SpiceNet, OrCAD, and Protel compatible schematic symbols are produced. This allows you to immediately use your new design in a schematic capture program and perform circuit simulations of your entire power system.



● Transformer ○ Inductor

Winding:	1	2	3
Primary:	●	○	●
Secondary:	○	●	○

Spice Model Listing:

```
.SUBCKT test 5 6 1 2 3 4
Lk0 in1 in2 631.8n
C11 2 62.33p
Rdc1 in1 N61 80.32m
Lmag in1 2 464.2u
Rcore in1 2 3.956k
Rac1 N61 1 3.480
Lac1 N61 1 126.6n
Cself1 1 2 13.04p
Lk1 in2 in3 508.6n
C12 4 1 70.16p
Efwd2 N42 4 in3 2 0.2857
Gfbk2 in3 2 N42 N62 434.1
```

Figure 11, Magnetics Designer produces a SPICE model of your transformer or inductor design.

As an example of this capability, Figure 12 shows a SpiceNet schematic and IsSpice4 simulation results of a 50W forward converter using our transformer design.

IsSpice4 includes models for many PWM ICs, power semiconductors, and power electronics devices. When coupled with Magnetics Designer, the two make a complete circuit design and analysis system that no other software vendor can match.

The second unique feature is the exposure of virtually all of the design variables used in the program. Magnetics Designer allows the user to freely create new output measurements and even affect the optimization criteria of the program. Parameters such as core area and flux density, thermal conductivity, power losses, leakage inductance and capacitive parasitics, resistance values, and mechanical specifications are all available.

Figure 13 shows an example of a new temperature variable which was created using the copper loss, core loss, core area, and ambient temperature parameters. Both user input and calculated output parameters are available, and additional equations may be added.

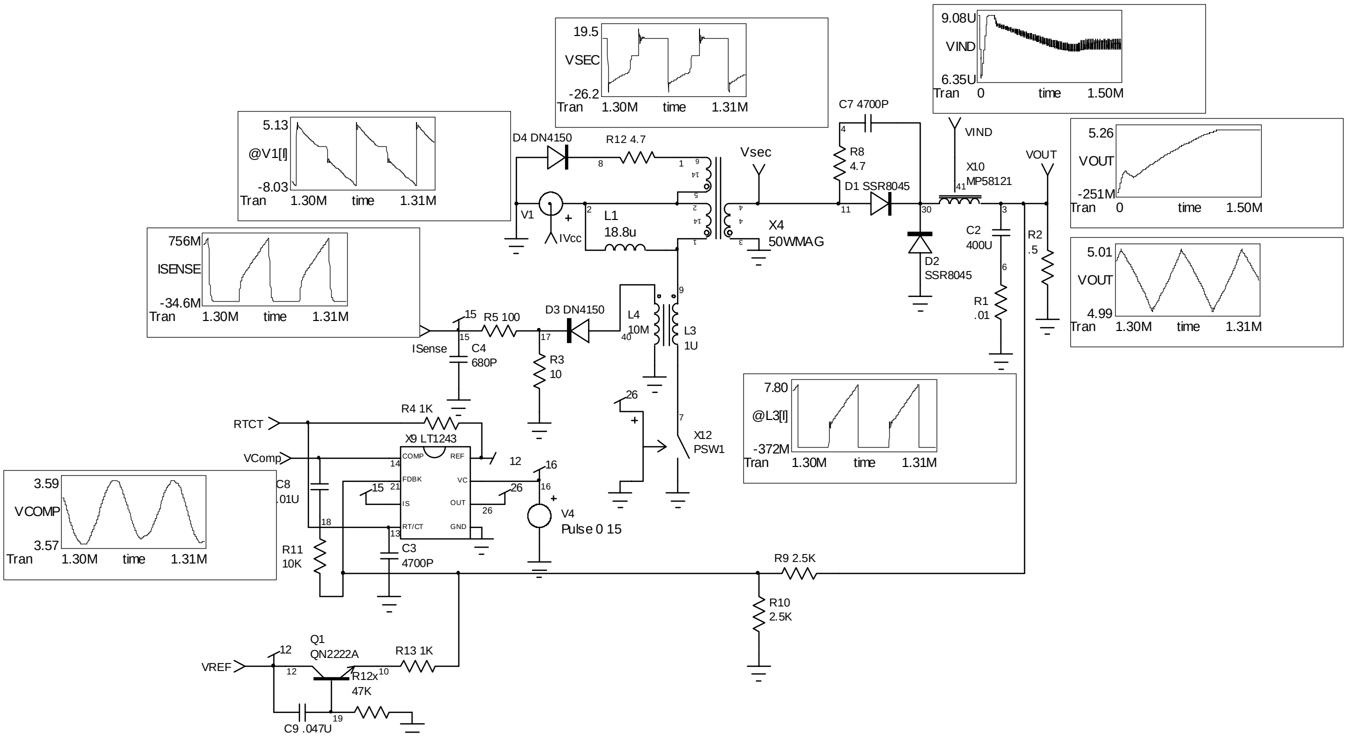
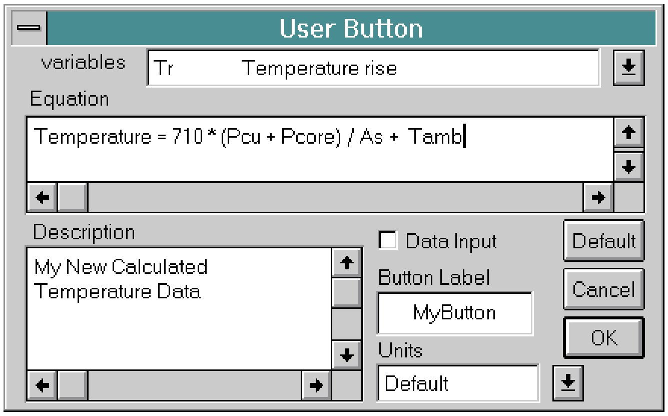


Figure 12, IsSpice4 simulation results of a 50W forward converter. The transformer design and its SPICE model were generated by Magnetics Designer. In this particular simulation, the transformer turns ratio has been changed from .285 to .5, thus changing the duty cycle to approximately 25%.



variables: Tr Temperature rise

Equation: Temperature = 710 * (Pcu + Pcore) / As + Tamb

Description: My New Calculated Temperature Data

□ Data Input

Button Label: MyButton

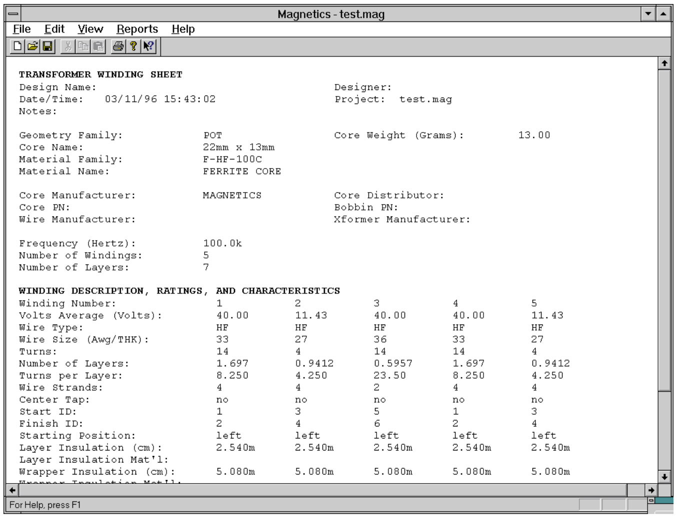
Units: Default

Figure 13, An example of a user generated equation that was added to the User Data area of the Transformer screen. The result of the equation will be shown in the User Data field next to the button.

This feature gives Magnetics Designer extreme flexibility and opens up many design boundaries for exploration.

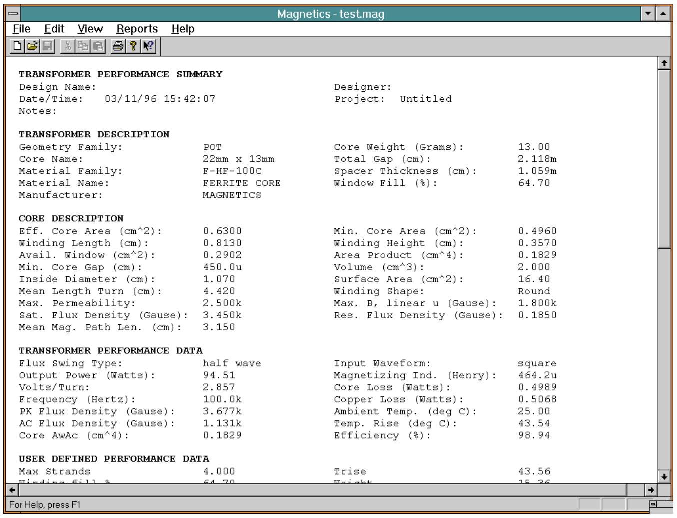
Magnetics Designer Reports

Magnetics Designer produces a complete report of the characteristics of your design in the form of an electrical performance summary and a winding sheet. Output from a typical report will appear as follows:



```
TRANSFORMER WINDING SHEET
Design Name:                                    Designer:
Date/Time:    03/11/96 15:43:02                 Project:  test.mag
Notes:

Geometry Family:            POT                 Core Weight (Grams):         13.00
Core Name:                  22mm x 13mm
Material Family:            F-HF-100C
Material Name:              FERRITE CORE

Core Manufacturer:          MAGNETICS           Core Distributor:
Core PN:                                        Bobbin PN:
Wire Manufacturer:                              Xformer Manufacturer:

Frequency (Hertz):          100.0k
Number of Windings:         5
Number of Layers:           7

WINDING DESCRIPTION, RATINGS, AND CHARACTERISTICS
Winding Number:               1          2          3          4          5
Volts Average (Volts):        40.00      11.43      40.00      40.00      11.43
Wire Type:                    HF         HF         HF         HF         HF
Wire Size (Awg/THK):          33         27         36         33         27
Turns:                        14         4          14         14         4
Number of Layers:             1.697      0.9412     0.5957     1.697      0.9412
Turns per Layer:              8.250      4.250      23.50      8.250      4.250
Wire Strands:                 4          4          2          4          4
Center Tap:                   no         no         no         no         no
Start ID:                     1          3          5          1          3
Finish ID:                    2          4          6          2          4
Starting Position:            left       left       left       left       left
Layer Insulation (cm):        2.540m     2.540m     2.540m     2.540m     2.540m
Layer Insulation Mat'l:
Wrapper Insulation (cm):      5.080m     5.080m     5.080m     5.080m     5.080m
```

Figure 14, The Winding Sheet contains the manufacturing information for your transformer.



```
TRANSFORMER PERFORMANCE SUMMARY
Design Name:                                    Designer:
Date/Time:    03/11/96 15:42:07                 Project:  Untitled
Notes:

TRANSFORMER DESCRIPTION
Geometry Family:            POT                 Core Weight (Grams):         13.00
Core Name:                  22mm x 13mm         Total Gap (cm):              2.118m
Material Family:            F-HF-100C           Spacer Thickness (cm):       1.059m
Material Name:              FERRITE CORE        Window Fill (%):             64.70
Manufacturer:               MAGNETICS

CORE DESCRIPTION
Eff. Core Area (cm^2):      0.6300              Min. Core Area (cm^2):       0.4960
Winding Length (cm):        0.8130              Winding Height (cm):         0.3570
Avail. Window (cm^2):       0.2902              Area Product (cm^4):         0.1829
Min. Core Gap (cm):         450.0u              Volume (cm^3):               2.000
Inside Diameter (cm):       1.070               Surface Area (cm^2):         16.40
Mean Length Turn (cm):      4.420               Winding Shape:               Round
Max. Permeability:          2.500k              Max. B, linear u (Gause):    1.800k
Sat. Flux Density (Gause):  3.450k              Res. Flux Density (Gause):   0.1850
Mean Mag. Path Len. (cm):   3.150

TRANSFORMER PERFORMANCE DATA
Flux Swing Type:            half wave           Input Waveform:              square
Output Power (Watts):       94.51               Magnetizing Ind. (Henry):    464.2u
Volts/Turn:                 2.857               Core Loss (Watts):           0.4989
Frequency (Hertz):          100.0k              Copper Loss (Watts):         0.5068
PK Flux Density (Gause):    3.677k              Ambient Temp. (deg C):       25.00
AC Flux Density (Gause):    1.131k              Temp. Rise (deg C):          43.54
Core AwAc (cm^4):           0.1829              Efficiency (%):              98.94

USER DEFINED PERFORMANCE DATA
Max Strands                 4.000               Trise                        43.56
```

Figure 15, A summary of the electrical performance of your magnetic design includes all the information found in the core screen, transformer or inductor screens, and User Data fields.

Signal Generators

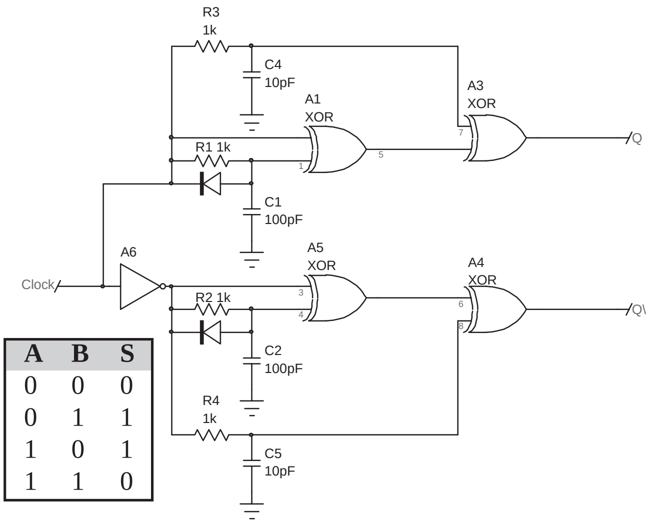
Authors
Christophe BASSO, consultant, Sinard, France
Charles Hymowitz, Intusoft
Larry Meares, Intusoft
Steve Sandler, Analytical Engineering Services

IsSpice4 introduces a dead-time in your bridge simulations

Christophe BASSO, MOTOROLA Semiconductor, Toulouse, France

February 1998

Bridge or half-bridge designs using MOSFETs or IGBTs need some dead-time between the commutations to avoid any cross-conduction current spikes. This statement is also valid in Switch Mode Power Supplies (SMPS) implementing synchronous rectification. In a simulation environment, it is not always an easy task to write the stimuli so as to define a dead-time between commutations. Classical PULSE or PWL commands are impractical, especially when either frequency or pulse width are changed during the simulation run.



A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Figure 1

The discrete solution to implement a dead-time generator

Figure 1 shows the solution built around a few logical XOR gates. The principle uses the truth table of an XOR or XNOR gate which states that its output is at a high or low level when both inputs are at different logical states.

This difference between the levels is made through the RC networks R1-C1 and R4-C5. The output of the A1-A5 gates is then a short pulse whose width is dependent upon the RC constant of its input network. This pulse will blank the signal delivered to the output and thus generates the required dead-time.

These logical functions can easily be implemented using INTUSOFT's IsSpice4 (San-Pedro, CA) Analog Behavioral Modeling features as demonstrated by the netlist given below:

```
.SUBCKT DEADTIME 1 50 51 {DT=500N VHIGH=10V
+  VLOW=100M RS=10}
* Clock_In Q Qbar
* Developed by Christophe BASSO (FRANCE)
RIN 1 0 1MEG
B6 17 0 V=V(1)>2V ? 10 : 0
R3 17 18 1k
C3 18 0 {DT/(1000*4.14)}
B4 21 0 V=V(25,19)<100MV ? {VLOW} : {VHIGH}
RCQ 21 60 100
CCQ 60 0 10P
BQ 61 0 V=V(60)
RSQ 61 50 {RS}
R4 22 23 1k
C4 23 0 {DT/(1000*4.14)}
B5 24 0 V=V(26,20)<100MV ? {VLOW} : {VHIGH}
RCQB 24 70 100
CCQB 70 0 10P
BQB 71 0 V=V(70)
RSQB 71 51 {RS}
R5 17 25 1k
C5 25 0 {DT/(1000*41.4)}
R6 22 26 1k
C6 26 0 {DT/(1000*41.4)}
D3 23 22 DISCH
D4 18 17 DISCH
B1 22 0 V=V(1)>2V ? 0 : 10
B2 19 0 V=V(17,18)<100MV ? 0 : 10
B3 20 0 V=V(22,23)<100MV ? 0 : 10
.MODEL DISCH D BV=100V CJO=4PF IS=7E-09 M=.45 N=2
+ RS=.8 TT=6E-09 VJ=.6V
.ENDS
```

The subcircuit needs to be fed with the dead-time value as well as the output high and low levels. The input clock is TTL-CMOS compatible. By changing B5 line to V=V(26,20)<100MV ? {VHIGH} : {VLOW}, the generator becomes suitable to drive a synchronous rectifier, as demonstrated by **figure 2**.

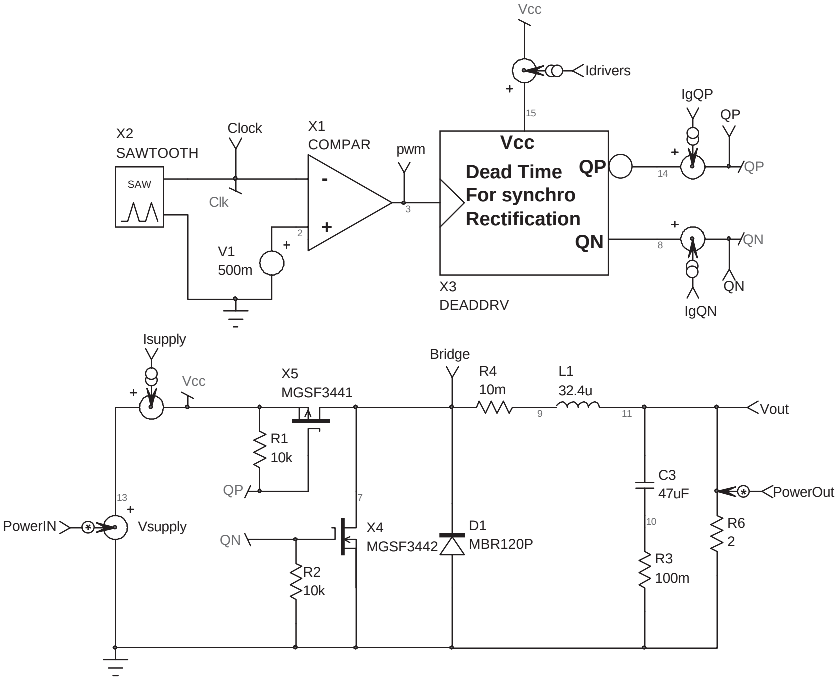


Figure 2
Application schematic in synchronous rectification

Figure 3 details the output signals delivered by the generator that clearly save the MOSFETs from any conducting overlap.

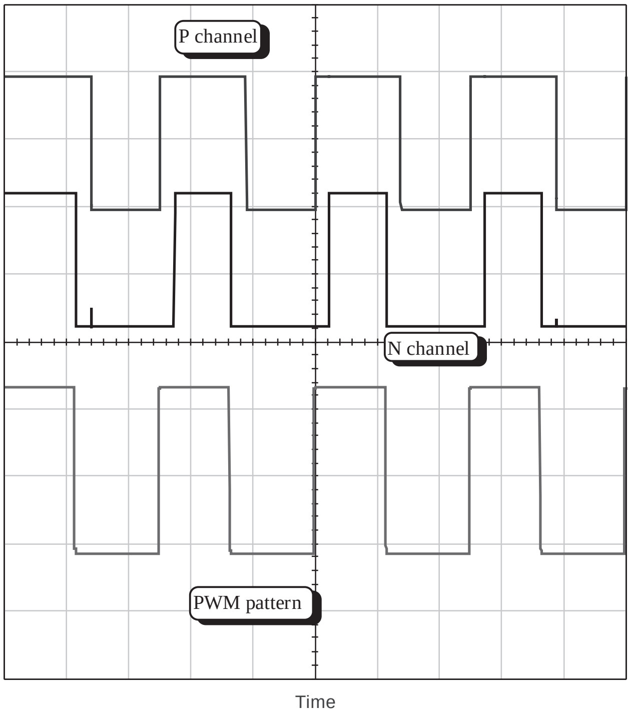


Figure 3
This picture clearly shows the absence of overlap between commutations

Three Phase Generator

Larry Meares, Intusoft, USA

January 1988

SPICE sine wave sources can be delayed in order to get different phase relationships. The delayed sources, however, remain at the initial starting value until the simulation reaches the specified delay. The following circuit can be used to get a source that starts three separate phases immediately at the beginning of the simulation. In this circuit, a sinusoidal source starting with zero delay is integrated using a capacitor fed from a current source. This 90 degree lagging signal is then summed with various weighting constants to give the desired waveforms.

Several things must be done to make this a useful element. First, the capacitor initial condition must be computed automatically and second, the generator needs to be controlled as though it were a piece of laboratory test equipment.

Controlling the phase and magnitude error signals makes it possible to evaluate circuit performance parameters that may not be seen in the laboratory. Test equipment that performs these operations is not commonly available. In instances like this, the power of the SPICE simulator can be applied to help make correct design decisions early. Decisions that may otherwise have been made without adequate technical information.

Algebraic expressions using the selected design parameters must be set up to define the various subcircuit element values. VGEN controls the amplitude of the sine wave generator, Vs = VGEN∗SIN(wt). FREQ is used directly in the sine wave generator and algebraically to define the phase shifting capacitor value. The phase shifting capacitor is used to form a generator, Vc = VGEN∗COS(wt). With both sine and cosine components, the following equations are used to combine the 2 signals for each of the 3 phases.

$$V(\phi = 0 \text{ Deg.}) = Vs$$

$$V(\phi = 120 \text{ Deg}) = COS(\phi)Vs + SIN(\phi)Vc = -.866Vs - .5Vc$$

$$V(\phi = -120 \text{ Deg}) = -.866Vs + .5Vc$$

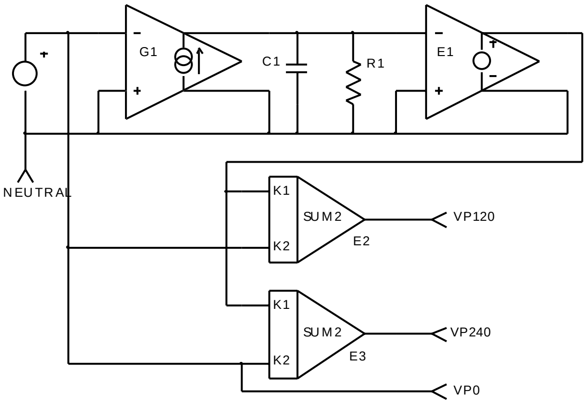
When a small phase error is introduced, then ϕ is replaced by ϕ + δ, and the small angle trigonometric series approximation for δ is substituted as follows:

$$\begin{aligned} SIN(\phi+\delta) &= SIN(\phi)COS(\delta) + COS(\phi)SIN(\delta) \\ &= SIN(\phi)(1 - \delta^2/2) + COS(\phi)(\delta - \delta^3/6) \end{aligned}$$

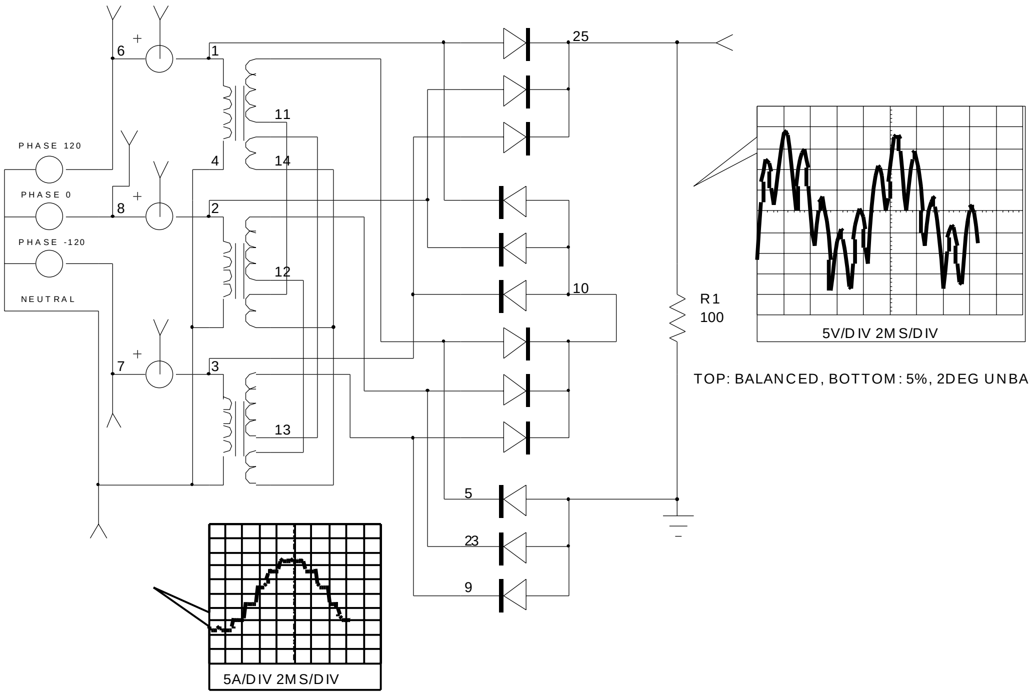
The sign of the cosine term must be adjusted to account for the lag in the generator. Notice that the cosine generator must start initially at VGEN. The pulse generator, I1, initializes the capacitor at time t = 0- and turns off at t = 0+. Constants are used in the algebraic expressions for conversion from degrees to radians, percent to fractions and for the series expansion coefficients.

```
IsSpice Subcircuit Netlist:
.SUBCKT GEN3 3 7 1 20
C1 2 20 {1/(6.28319K*FREQ)} IC={VGEN}
R1 2 20 1E6
I1 20 2 PULSE {VGEN*1U} 0
* MAKES UIC UNNECESSARY
E1 5 20 20 2 1
V1 3 20 SIN 0 {VGEN} {FREQ}
E2 7 20 POLY(2) 5 20 3 20 0 -866.00M -500.00M
E3 1 20 POLY(2) 5 20 3 20 0
+ {(1 + .01 * MAGERR) * (.866 *(1-.5 * (.0174533
+ * PHASE)^2)-.5 * .0174533 * PHASE *
+ (1 + .166667 * (.0174533 * PHASE)^2))}
+ {(1+.01*MAGERR)*(-.5*(1-.5 * (.0174533 *
+ PHASE)^2)-.866 * .0174533 * PHASE *
+ (1 + .166667 * (.0174533 * PHASE)^2))}
G1 20 2 20 3 1M
R2 7 0 100MEG
R3 1 0 100MEG
R4 3 0 100MEG
R5 5 0 100MEG
.ENDS
```

Three Phase Generator Source



The three phase generator source was tested in an application that used auxiliary transformers to make a 12 phase rectifier. This type of rectifier has advantages in improved power factor and reduced output ripple. The simulation can be used to evaluate the effectiveness of these improvements under the conditions of unbalanced input power. The IsSpice simulation results, shown next, illustrates a significant second harmonic ripple in the output.



TOP: BALANCED, BOTTOM: 5%, 2DEG UNBALANCE

Simulating Pulse Code Modulation

There are applications, such as uninterruptible power supplies (UPS) that convert a DC input voltage to a sinusoidal AC output voltage. The basis of the conversion is very similar to the conversion of a DC input to a DC output voltage. One of the more difficult aspects of DC to sinewave conversion is obtaining a regulated, low distortion sinewave reference. The goal is to provide a variable frequency and amplitude with low harmonics and a zero DC term.

While pulse width modulation is a possibility, another method is to utilize a string of ones and zeros. When repeated this string will possess a Fourier series consisting of a fundamental and some harmonics. By picking all of your ones and zeros correctly, you can force most of the lower harmonics to zero and still provide a variable amplitude output that is both microcontroller friendly and free of a high frequency carrier.

The following demonstrates an unusual task for IsSpice4. The circuit in Figure 6 simulates a single bit pulse code representation of a sinewave. The same circuit is easily extended to any number of phases. The fundamental problem was generating a bit pattern for the sinewave reference. The pulse generator, V1, is used as the clock. Since we will generate 256 bit values this clock is 256 times greater than the output frequency. Flip-flop X1 latches the data between clock pulses. V2 is a sinewave used for a reference. R1, R5, C1 and C2 filter the pulse coded waveform and reconstruct the sinewave. B1 is a behavioral If-Then-Else comparator that sets the output bit high if the output is lower than the sinewave reference value, or sets the output bit low if the output is higher than the sinewave reference value. B2 level shifts the bit values to a zero-one format.

Looking at the cross-probed waveforms, you can see the bit patterns and the sinewave output at each filter stage. Note that more sophisticated filters could produce lower distortion, as could more values in the data table. The output listing from Bits_Out, node V(9) in the IsSpice4 output file, is a series of ones and zeros and is the bit pattern representing the sine wave.

Using A State Machine To Generate A Sine ROM

We could use the circuitry in Figure 6 to generate the pulse codes for other parts of a simulation, however, it is much more efficient to code the digital output into a ROM. Fortunately, IsSpice4 includes a state machine model that can store and play back the pulse code data. Table 2 shows a partial listing of the state machine input file. The data in the Output column is from

Simulating Pulse Code Modulation

Steve Sandler, Excerpts from the August 1995 Intusoft Newsletter #43

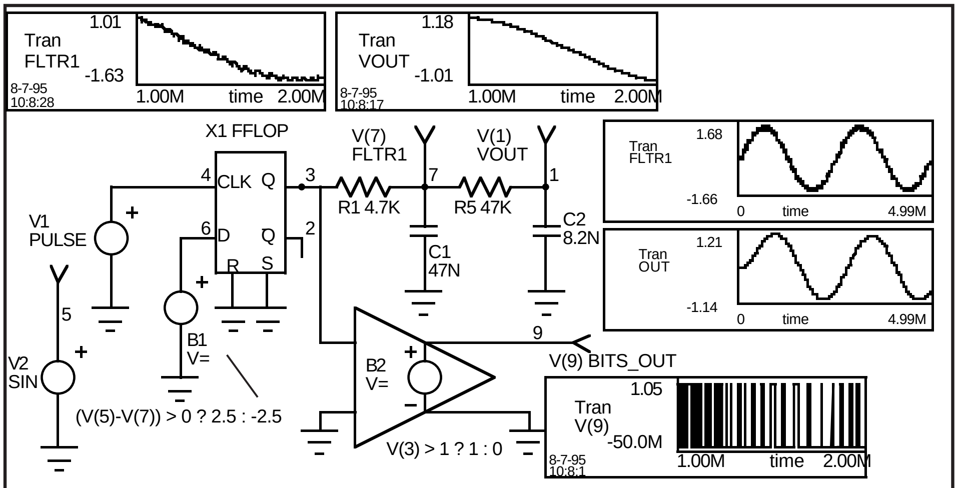


Figure 6, Circuit used to generate a sine wave and the 1 bit code representing the sine wave. A portion of the sine wave and bit stream has been cross-probed.

Figure 6 node 9. Figure 7 uses the state machine to create a model for an 8 bit counter with the 256 bit ROM. The clock frequency must be 256 times the desired sinewave frequency. The Tstep value of 9.766u in the simulation's .Tran statement (.Tran 9.766u 4.99M) is important since this is the period of the sample. Selecting a different number requires a Tmax value which slows down the simulation. Note that by changing the clock frequency we can change the sine wave frequency.

The runtime performance of the state machine is far superior to the Figure 6 method (Figure 6: 79.15 seconds, State Machine circuit: 14.50 seconds, Pentium/75) and its improved performance will become significant over the course of many runs. In addition, the IsSpice4 state machine is a C code model that is separate from the simulator. The model's source code is available to those who might wish to expand its functionality and input formats. For example, it could be possible for the state machine to accept ABEL or JEDEC descriptions.

Three Phase Sine Reference

As an extension to the single phase case, Figure 8 demonstrates a three phase sine wave reference circuit. A 6 stage shift

State	Output	Transition		
0	0s	0	->	1
1	1s	0	->	2
2	1s	0	->	3
3	0s	0	->	4
4	1s	0	->	5
5	1s	0	->	6
...	...	...	...	...
253	0s	0	->	254
254	1s	0	->	255
255	1s	0	->	0

Table 2, The input to the state machine generated by Figure 6 and used in Figure 7. The Output is the state machine output at the particular state. The Transition determines which state the machine will move to depending on the input. In this case, the input is held at 0 and the machine simply progresses state by state until state 255 where it then repeats.

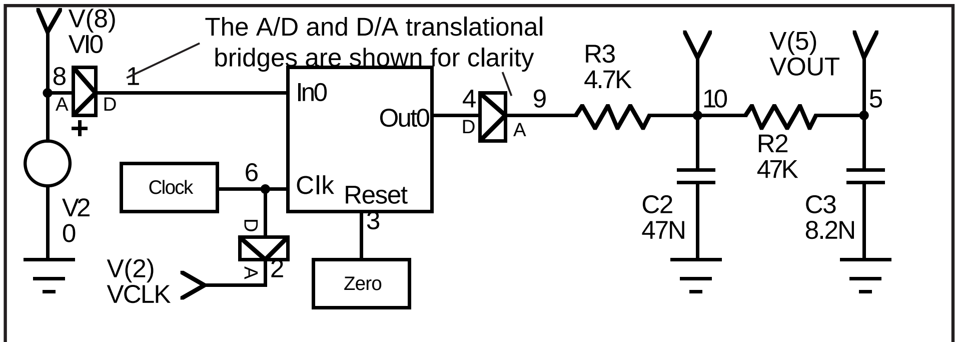


Figure 7, This circuit generates the same sine waves shown in Figure 6, however, here a ROM (state machine) is used to driver the filter.

register is used to generate 3 quasi-square waves which are exactly 120 degrees apart. Each waveform has a conduction angle of 120 degrees. The 120 degree quasi-square waveform has the advantage of having no third harmonic content; the first significant harmonic is the fifth. Each quasi-square wave is filtered by a second order active low pass filter. The quasi-square waves are created by averaging 2 square waves which are phase shifted by 60 degrees. The sinewave output distortion could be further reduced by using a higher order active filter, reducing the corner frequency of the existing filters, or replacing the quasi-square waveform with a more sophisticated waveform to eliminate several more harmonics. Care must be taken in the placement of the filters, since component tolerances could easily alter the phase angles between phases; something that could be investigated with the IsSpice4 Monte Carlo analysis. As in the previous case, a state machine model could be created to hold the waveform's pulse codes.

References

[1] "The Quest for magic Sine Waves", Don Lancaster, Circuit Cellar Ink, #59, 6/95
[2] "SMPS Simulation With SPICE 3", Steve Sandler, forthcoming from Intusoft

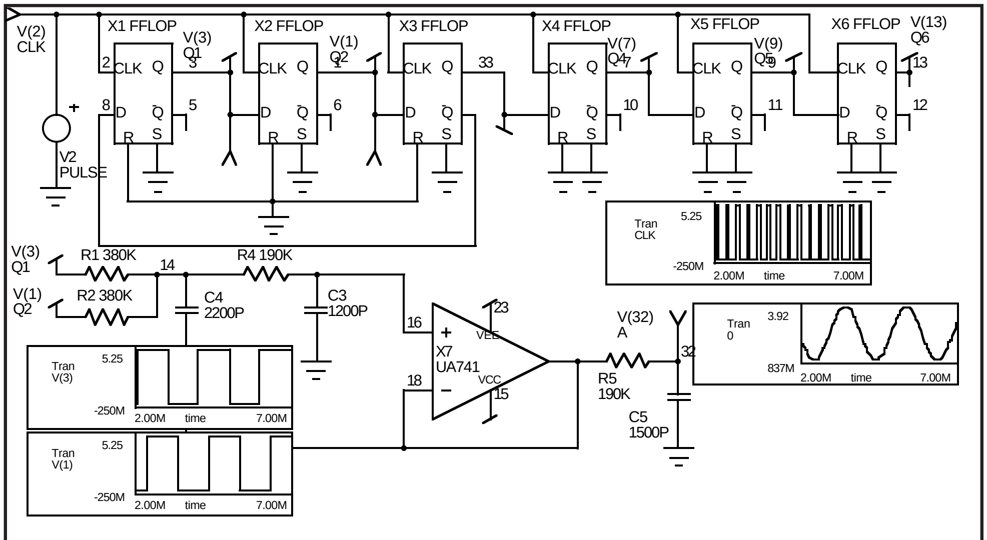


Figure 8, Portion of the circuit used to generate a 3 phase sine wave. The taps at the flip-flops are used to drive the active filters. Only one of the 3 phases is shown.

Modeling For Power Electronics

Authors
Christophe BASSO, consultant, Sinard, France
Charles Hymowitz, Intusoft
Larry Meares, Intusoft
Slim Petrie, consultant, Arlington Heights, IL
Steve Sandler, Analytical Engineering Services

A Spice Model For TRIACs

A. F. Petrie, Independent Consultant, 7 W. Lillian Ave., Arlington Heights, IL 60004
Charles Hymowitz, Intusoft

SPICE is the most popular program for simulating the behavior of electronic circuits. The biggest stumbling block that engineers run into is turning vendor data sheet specifications into SPICE models that emulate real devices and run without convergence problems. This is especially true for power devices, like Triacs, where the cost of testing and possibly destroying devices is prohibitive. The following paper describes the FIRST known SPICE subcircuit macro model for a Triac[1].

Introduction

Berkeley SPICE is the most popular program for simulating the behavior of electronic circuits. The biggest stumbling block most engineers run into is turning company specifications or vendor data sheets into SPICE device models that emulate real devices and run without problems. Here is an approach which works well for an TRIAC and derives directly from data book parameters.

A triac is a bilateral switch that can be triggered into conduction regardless of its polarity. It is modeled by using two NPN/PNP transistor pairs connected back-to-back as shown on the left of Figure 4. The base of each transistor is connected to the collector of the other. This produces positive feedback, resulting in the required switching action. At MT2, the emitter metallization overlaps the base of the NPN transistor. This forms a lateral resistor in the base (P) region and is shown as RH in the model. RH determines the holding current of the triac. A similar resistor exists at MT1. These resistors hold the triac off unless triggered by the gate or the holding current from a previous "on" condition.

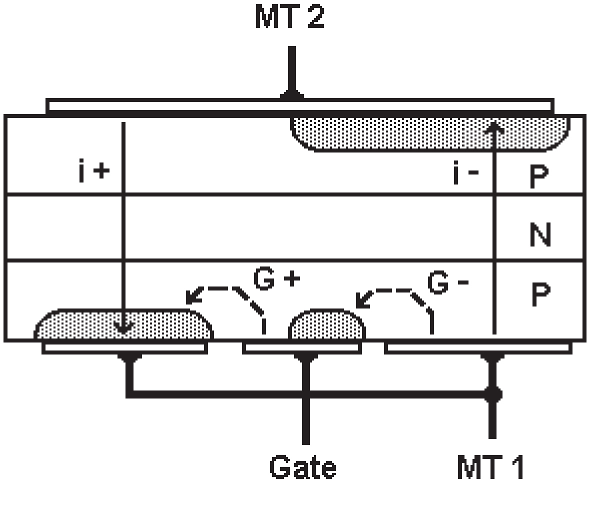


Figure 1, Basic Triac structure

Although this configuration has the basic triggering function of the triac, it must be enhanced to emulate other important parameters such as off state leakage, breakover voltage and current, and voltage and current characteristics in all four modes of operation. Figure 4 shows the full triac subcircuit. Resistors and zener diodes are used to simulate the breakdown voltages and leakage currents. Dependent sources are used to emulate the various triggering modes and to allow a wider range of trigger data to be entered.

If adjusting the model parameters for the triac subcircuit seems a little daunting, you can use SpiceMod. SpiceMod is a software program that quickly converts data book parameters into SPICE model parameters. Entering only the device type and maximum voltage and current ratings will produce a realistic model as all other parameters are scaled from them. Naturally, the more data you enter, the more exact the model will be. Figure 5 shows the triac entry screen from SpiceMod.

This model and method are used for the TRIAC in the SpiceMod program. This program allows direct conversion of data book information to SPICE models for diodes, zeners, bipolar transistors, JFETs, MOSFETs, SCRs and now Triacs. Models can be generated as fast as data can be entered. This

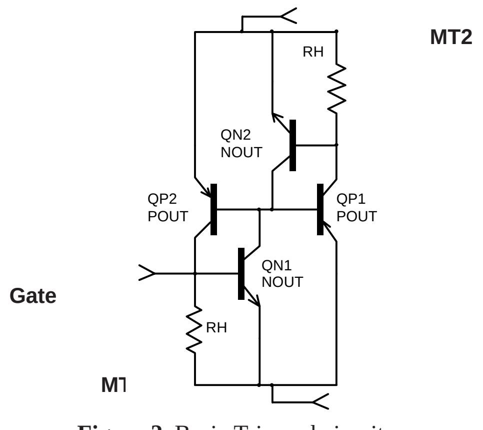


Figure 2, Basic Triac subcircuit

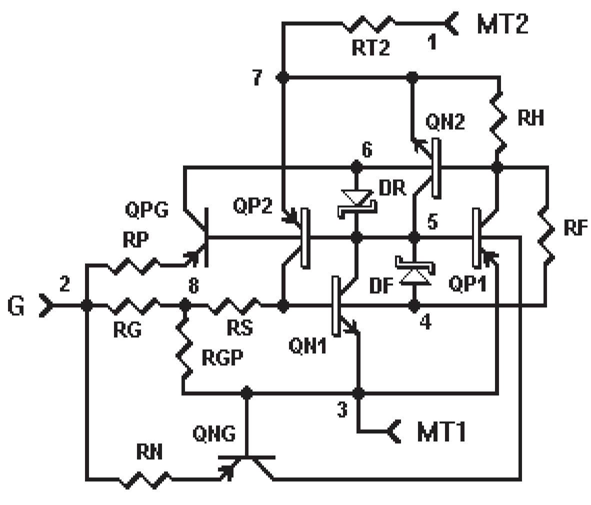


Figure 3

makes it easy to generate models for custom or "house" devices. Limit devices can be modeled by entering limit device data. Because these models use standard SPICE syntax, they can be used with any program that adheres to SPICE rules.

Data books may show the same values for all four gate trigger current and voltage modes. However, the values are different in all four modes. In a real triac, when the gate is driven with the opposite polarity from the MT2 terminal, the gate must supply enough current to drive the shunting resistance to twice the gate voltage, thus the trigger current cannot be the same for both polarities, regardless of what the data book limits say. G+ with MT2- always requires more gate drive than other modes. In this case, it is only necessary to enter the first mode gate voltage and current and let the program estimate the others for the most realistic model. SpiceMod guards against entering grossly unrealistic data and will attempt to produce the closest realistic model in these situations.

The SPICE 2G compatible netlist for the triac is shown in Table 1 (next page). Along with the "Affects" column in Figure 5, you can get an idea of the relationship between which data sheet parameters correspond to which SPICE model parameters.

Figure 4 shows the SPICE subcircuit that was developed to solve these problems. Replacing the trigger transistors with diodes and voltage controlled current sources reduces the number of SPICE parameters needed and provides some isolation between the four modes of operation. The gate connections still emulate the feedback to the gate.

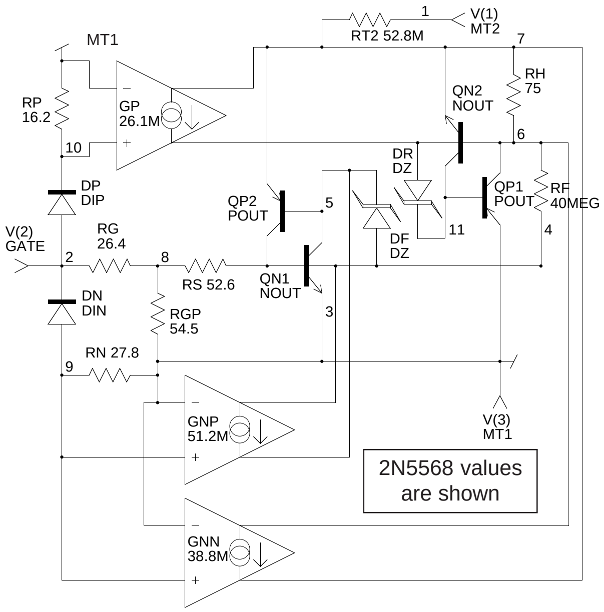
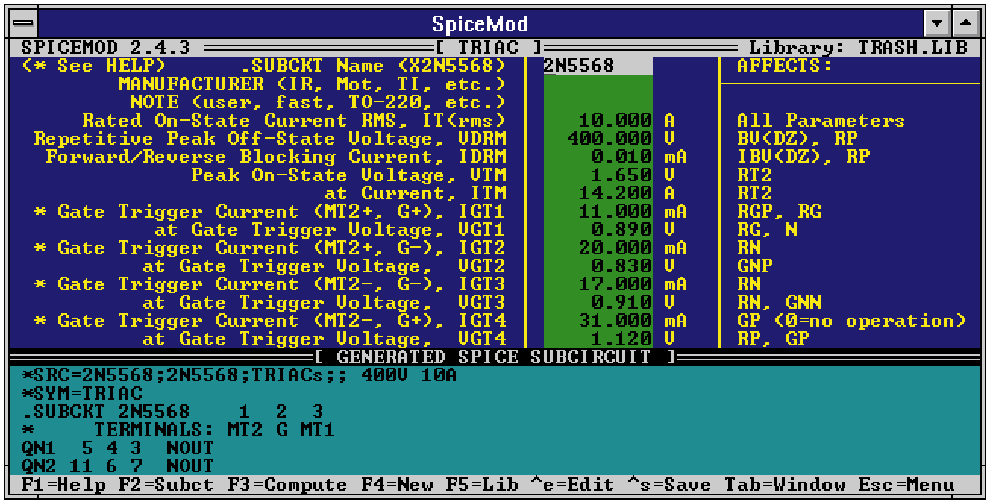


Figure 4, SPICE 2G.6 compatible Triac subcircuit.

Table 1 lists the .SUBCKT for this same model. QN1, an NPN transistor, has a forward beta of 20 and is connected to the gate at its base. QP2, a PNP transistor, is connected to MT2 and QN1 in a positive feedback mode. Due to the fact that there are two stable states (on and off) for the TRIAC, it is necessary to add the "OFF" statement after the transistors in the .SUBCKT model or SPICE may never evaluate the off condition or may hang due to the uncertainty. The OFF command causes SPICE to start the calculations assuming the off condition. If the device is really "on", there is no stable "off" state, so the OFF command has no effect on the

```
.SUBCKT 2N5568    1  2  3
*     TERMINALS: MT2 G MT1 Mot. 400V 10A
QN1  5 4 3  NOUT         ; Output Transistors
QN2 11 6 7  NOUT         ; "OFF" keyword for Q devices
QP1 6 11 3  POUT
QP2  4 5 7  POUT
DF  4  5  DZ             ; Forward breakdown diode
DR  6 11  DZ             ; Reverse breakdown diode
RF  4 6  40MEG           ; Forward leakage current (controls IDRM).
RT2 1 7  52.8M           ; Controls "on" resistance
RH  7 6  75              ; Controls reverse holding current
RGP 8 3  54.5            ; Controls fwd holding current and trigger current
RG  2 8  26.4            ; with RGP controls VGT
RS  8 4  52.6            ; with RGP controls forward holding current
DN  9 2  DIN             ; Diode to isolate G- triggering modes
RN  9 3  27.8            ; Controls current in G- trigger modes
GNN  6 7 9 3 38.8M       ; Controls G-, MT2- trigger voltages
GNP  4 5 9 3 51.2M       ; Controls G-, MT2+ trigger voltages
DP 2 10  DIP             ; Diode to isolate G+, MT2- trigger mode
RP 10 3  16.2            ; Controls current in G+, MT2- trigger mode
GP  7 6 10 3 26.1M       ; Controls G+, MT2- trigger voltage
.MODEL DIN  D (IS=53.5F)       ; Conducts in G- modes
.MODEL DIP  D (IS=53.5F N=1.19) ; Higher drop diode conducts only in
                                          ; G+, MT2- mode
.MODEL DZ   D (IS=53.5F N=1.5 IBV=10U BV=400)
.MODEL POUT PNP (IS=53.5F BF=5 CJE=235P TF=25.5U)
.MODEL NOUT NPN (IS=53.5F BF=20 CJE=235P CJC=46.9P TF=1.7U)
.ENDS
```

Table 1, SPICE 2G Triac model generated by the SpiceMod modeling program from manufacturer's data sheet parameters.



```
SPICEMOD 2.4.3 ==========[ TRIAC ]========== Library: TRASH.LIB
(* See HELP)       .SUBCKT Name (X2N5568)   2N5568         AFFECTS:
        MANUFACTURER (IR, Mot, TI, etc.)
         NOTE (user, fast, TO-220, etc.)
      Rated On-State Current RMS, IT(rms)    10.000 A      All Parameters
 Repetitive Peak Off-State Voltage, VDRM   400.000 V      BV(DZ), RP
  Forward/Reverse Blocking Current, IDRM     0.010 mA     IBV(DZ), RP
               Peak On-State Voltage, VTM     1.650 V      RT2
                          at Current, ITM    14.200 A      RT2
 * Gate Trigger Current (MT2+, G+), IGT1    11.000 mA     RGP, RG
           at Gate Trigger Voltage, VGT1     0.890 V      RG, N
 * Gate Trigger Current (MT2+, G-), IGT2    20.000 mA     RN
          at Gate Trigger Voltage,  VGT2     0.830 V      GNP
 * Gate Trigger Current (MT2-, G-), IGT3    17.000 mA     RN
          at Gate Trigger Voltage,  VGT3     0.910 V      RN, GNN
 * Gate Trigger Current (MT2-, G+), IGT4    31.000 mA     GP (0=no operation)
          at Gate Trigger Voltage,  VGT4     1.120 V      RP, GP
==========[ GENERATED SPICE SUBCIRCUIT ]==========
*SRC=2N5568;2N5568;TRIACs;; 400V 10A
*SYM=TRIAC
.SUBCKT 2N5568    1  2  3
*     TERMINALS: MT2 G MT1
QN1   5 4 3   NOUT
QN2  11 6 7   NOUT
F1=Help F2=Subct F3=Compute F4=New F5=Lib ^e=Edit ^s=Save Tab=Window Esc=Menu
```

Figure 5, Data sheet parameters (above left) used to create the SPICE Triac subcircuit (Table 1). To make a new model, data sheet values are entered into the SpiceMod entry screen. As they are entered the subcircuit values are calculated. The more data that is entered, the more accurate the final model will be. The subcircuit parameters affected by each entered parameter are shown to right.

accuracy of the result. (Hint: any time SPICE is used to evaluate a bi-stable device, the OFF or ON command must be used to obtain the correct state.)

Triacs have two stable states. Therefore, it may be necessary to tell SPICE which state to use, especially when you want to start a simulation with the triac in the off state. To do this you can issue the “OFF” keyword on the subcircuit transistor lines.

Figure 6 shows the low current region of a Motorola 2N5568 triac with no input applied to the gate. This curve was generated by holding the current to the gate at zero and sweeping the MT2 current from -50mA to +50mA. The slope of the curve at the zero axis is determined by the IDRM specification. The maximum voltage swings in the off state are determined by the VDRM specification. The trigger points (+6 and -10mA) are determined by the holding current.

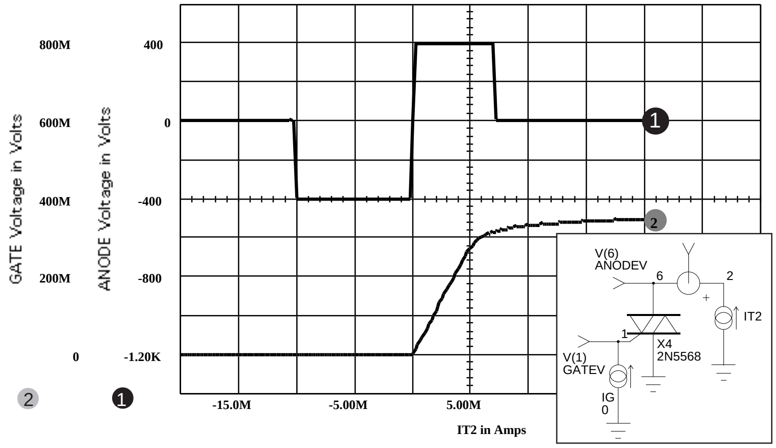


Figure 6, DC analysis of the Motorola 2N5568 triac. Several FREE models are posted on the Intusoft Web Site.

Simplified behavioral triac models are included with IsSpice4. Their superior simulation speed and idealized response can make them useful for investigating triac control circuitry. However, with power devices such as the triac, you will normally need models that exhibit 2nd order effects as well. Without them you can’t run realistic simulations. Most other SPICE vendors do not offer such sophisticated power semiconductor models. Fortunately, Intusoft does.

There are four gate trigger modes that must be modeled. The first mode is similar to an SCR with MT2 positive and the gate also positive. For convenience, this is combined with the fourth mode where the gate is also positive and MT2 is negative. Figure 7 is a plot of MT2 voltage verses gate current using a 12 volt supply and a 100 ohm load resistor. Note that the device turns on at the specified 11 mA when MT2 is positive and at 31 mA of gate current when MT2 is negative. The first mode gate current and voltage are controlled by RG and RGP. The fourth mode gate current and voltage are controlled by resistor RE and the forward beta (BF) of QPG. Table 2 lists the SPICE file used to generate this curve.

The gate negative mode makes use of diode DN, resistor RN, and transconductance generator GN in the subcircuit to turn on transistors QP1 or QP2. GN is used to adjust the gate sensitivity when MT2 is negative and BF of QP2 is used to adjust the gate sensitivity when MT2 is positive. This method allows adjustment of gate sensitivity with minimum interaction. Note that an NPN transistor could have been used in place of DN, RN and GN in the model, but a transistor model is more complex and the control parameters more difficult to calculate.

SPICE Problems:

When running a SPICE analysis using a bi-stable device like a TRIAC, the device usually switches on or off, causing a discontinuity in the output. Since SPICE makes new calculations starting with the previous values, it doesn't like discontinuities. To help SPICE converge, the following OPTIONS statement is recommended:

.OPTIONS RELTOL=.01 ITL1=500 ITL2=200

The increased iterations only occur when needed and the reduced accuracy is adequate for a switching circuit. This is far more desirable than receiving a "no convergence" message.

I made mention earlier of the unrealistic data found in manufacturer's data books. In a real Triac, when the gate is driven with the opposite polarity from the MT2 terminal, the gate must supply enough current to drive the shunting resistance to twice the gate voltage, thus the trigger current cannot be the same for both polarities, regardless of what the data book limits say. The values are just made high enough to cover all devices and polarities. For this reason, the SpiceMod program will not allow grossly unrealistic values to be entered.

References:

1. R. L. Avant, F. C. Lee, and D. Y. Chen, A Practical SCR Model for Computer Aided Analysis of AC Resonant Charging Circuits, IEEE, July, 1981.

2. Paolo Antognetti and Giuseppe Massobrio, Semiconductor Device Modeling with SPICE, McGraw-Hill, 1988.

3. "Thyristor Device Data," Motorola Inc., DL131, REV3, 1991.

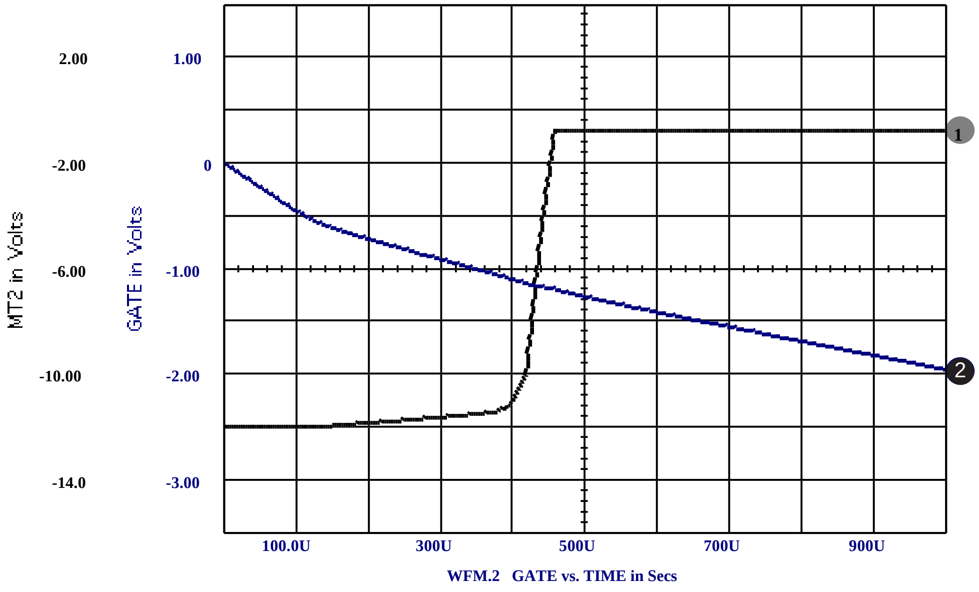
```
TRIAC-GT.CIR - TRIAC GATE TRIGGER CURVES
.OPTIONS RELTOL=.01 ITL1=500 ITL2=1500
*.DC IG  0 -.05 -.0005 VCC -12 12 24
* ^ Gate Trigger Curve
* Note: for some triac models the DC analysis will
* not converge. The Transient analysis may be substituted
.TRAN 1u 1m UIC
.PRINT DC V(1) V(2)
.PRINT TRAN V(1) V(2)
*ALIAS V(1)=MT2 VOLTAGE
*ALIAS V(2)=GATE VOLTAGE
*ALIAS IT2=MT2 CURRENT
*ALIAS IG=GATE CURRENT
*INCLUDE TRIAC.LIB
*         ^ Use your library path and name.
VCC 3 0 -12V ; or 12V
*       ^Set for the proper biasing if using transient
RL  3 1 100
*       ^Adjust for device.
IG  0 2 0 pulse 0 -.05 0 1M
X1  1 2 0 T435-800 ; 2N5568
*         ^ Change to desired device.
.END
```

Table 2, The switching circuit TRIAC-GT.CIR used to test the DC and transient Triac performance.

SpiceMod models diodes, Bjts, Jfets, Mosfets, SCRs, IGBTs, Power Mos, Power BJTs, Sidac, Sidactors, and Darlington BJTs and now models zeners and triacs. SpiceMod is compatible with all SPICE simulators and is available from Intusoft

A. F. "Slim" Petrie is retired from Motorola Inc., where he was a Principal Staff Engineer and Dan Noble Fellow. He develops both hardware and software and holds 24 U S Patents. His "Circuit Analysis" program was sold through Apple Computer in the early 1980s. With a keen appreciation for the problems of the working engineer, he continues to develop tools to make that job easier. He can be reached at 7 W. Lillian Ave., Arlington Heights, IL 60004, where he welcomes your comments.

Figure 7, a plot of MT2 voltage versus gate current using a 12 volt supply and a 100 ohm load resistor.



A Spice Model For IGBTs

A. F. Petrie, Independent Consultant, 7 W. Lillian Ave., Arlington Heights, IL 60004
Charles Hymowitz, Intusoft

SPICE is the most popular program for simulating the behavior of electronic circuits. The biggest stumbling block that engineers run into is turning vendor data sheet specifications into SPICE models that emulate real devices and run without convergence problems. This is especially true for power devices, like IGBTs, where the cost of testing and possibly destroying devices is prohibitive. The following paper describes the FIRST known SPICE subcircuit macro model for IGBTs[1].

Introduction

You've finally tested a version of your design that seems to work well, but you would feel a lot better if you KNEW the circuit would work well with all the devices that the vendor will supply in production. You found a model in a library, but you are not sure what specifications from the data book apply to that model. The following paragraphs will try to clarify the relationship between data book specifications and a new Insulated Gate Bipolar Transistor (IGBT) subcircuit SPICE model.

Modeling An IGBT

An IGBT is really just a power MOSFET with an added junction in series with the drain. This creates a parasitic transistor driven by the MOSFET and permits increased current flow in the same die area. The sacrifice is an additional diode drop due to the extra junction and turn-off delays while carriers are swept out of this junction.

Figure 1 shows a simplified schematic of an IGBT. Note that what is called the "collector" is really the emitter of the parasitic PNP. What we have is a MOSFET driving an emitter follower. Although this model is capable of producing the basic function of an IGBT, refinements are required for more accurate modeling and to emulate the nonlinear capacitance and breakdown effects.

Expanded IGBT Model

Figure 2 shows the complete subcircuit. Table 1 shows the corresponding SPICE 2G.6 compatible subcircuit netlist for an International Rectifier IRGBC40U device [2]. The subcircuit is generic in nature, meaning, that component values in the subcircuit can be easily recalculated to emulate different IGBT devices. The model accurately simulates, switching loses, nonlinear capacitance effects, on-voltage, forward/reverse breakdown, turn-on/turn-off delay, rise time and fall tail, active output impedance, collector curves including mobility modulation.

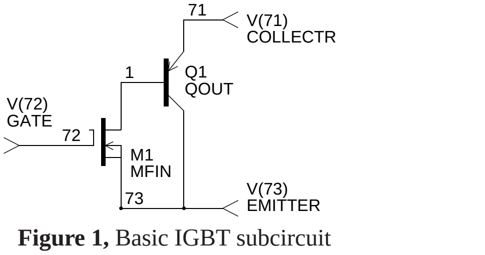


Figure 1, Basic IGBT subcircuit

Let's discuss the subcircuit one component at a time:

Q1 is a PNP transistor which functions as an emitter-follower to increase the current handling ability of the IGBT. BF (Forward Beta) is determined by the step in the turn-off tail which indicates the portion of the current handled by the PNP. TF (Forward Transit Time) controls the turn-off tail time. The OFF control parameter can be added to aid DC convergence by starting DC calculations with Q1 turned off.

MOSFET M1 emulates the input MOSFET [3, 4]. The Berkeley SPICE Level=3 model is used in the .MODEL MFIN statement in order to better model modern device characteristics. VMAX (Maximum Drift Velocity) controls the collector (drain) curves in the saturation region, and hence the VCE(on) voltage. THETA (Mobility Modulation Parameter) is used to reduce the gain at high gate voltages which is normally exponential. ETA (Static Feedback) is similar to the "Early effect" in bipolar transistors and is used to

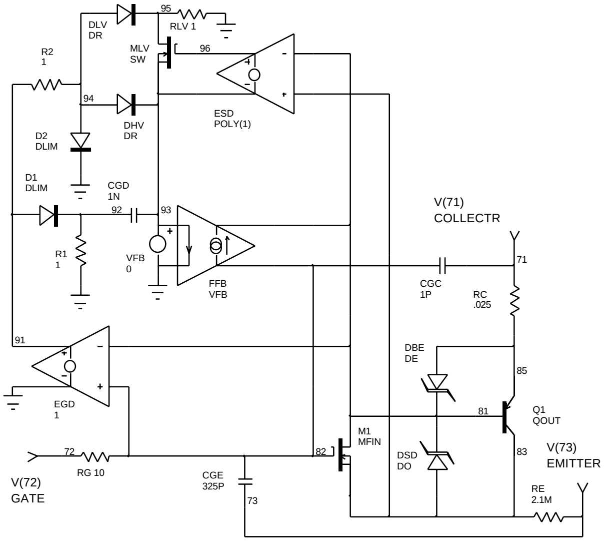


Figure 2, SPICE 2G.6 compatible IGBT subcircuit

Table 1, IRGBC40U IGBT Subcircuit

```
.SUBCKT IRGBC40U 71 72 74
*     TERMINALS:  C  G  E
* 600 Volt  40 Amp  6.04NS  N-Channel IGBT
Q1  83 81 85     QOUT
M1  81 82 83 83  MFIN L=1U W=1U
DSD 83 81  DO
DBE 85 81  DE
RC  85 71  21.1M
RE  83 73  2.11M
RG  72 82  25.6
CGE 82 83  1.42N
CGC 82 71  1P
EGD 91  0 82 81  1
VFB 93  0  0
FFB 82 81  VFB  1
CGD 92 93  1.41N
R1  92  0  1
D1  91 92  DLIM
DHV 94 93  DR
R2  91 94  1
D2  94  0  DLIM
DLV 94 95  DR 13
RLV 95  0  1
ESD 96 93  POLY(1) 83 81  19  1
MLV 95 96 93 93  SW
LE  73 74  7.5N
.MODEL SW NMOS (LEVEL=3 VTO=0 KP=5)
.MODEL QOUT PNP (IS=377F NF=1.2 BF=5.1 CJE=3.48N
+ TF=24.3N XTB=1.3)
.MODEL MFIN NMOS (LEVEL=3 VMAX=400K THETA=36.1M ETA=2M
+ VTO=5.2 KP=2.12)
.MODEL DR D (IS=37.7F CJO=100P VJ=1 M=.82)
.MODEL DO D (IS=37.7F BV=600 CJO=2.07N VJ=1 M=.7)
.MODEL DE D (IS=37.7F BV=14.3 N=2)
.MODEL DLIM D (IS=100N)
.ENDS
```

control the slope of the collector curves in the active region and hence the output impedance. VTO (Threshold Voltage) is directly proportional to Gate Threshold Voltage VGE(th). KP (Intrinsic Transconductance) is related to the test parameter gfe (Forward Transconductance) but must be adjusted for VTO, VMAX, THETA, and ETA.

DSD emulates the source-drain (substrate) diode, its capacitance, and forward breakdown voltage. VJ and M have been adjusted to better emulate the (Coes) capacitance curve. This diode includes breakdown voltage and capacitor CBD as CJO. DBE, the B-E diode of the output transistor, emulates the reverse breakdown of the PNP base-emitter junction (and of the IGBT). IS is made small and N large to avoid shunting the junction in the forward direction.

RC, the collector resistance, represents the resistive part of VCE(on). With the B-E diode, RC controls the VCE(on) voltage. RE, the emitter ohmic resistance, provides the feedback between emitter current and gate voltage. RG, the gate resistance, combines with the gate capacities in the subcircuit to help emulate the turn-on and turn-off delays, and the rise and fall times.

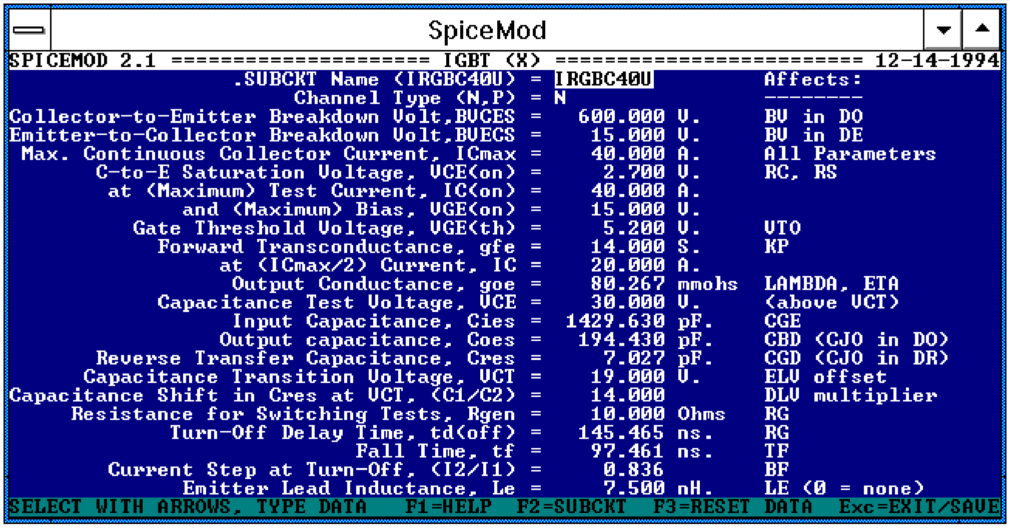
CGE, the Gate-to-Emitter capacitor, equals Cies minus Cres. CGC, the Gate-to-Collector capacitor, is a fixed capacitor representing package capacitances which are important at high voltages where Cres is small.

There are nine parts that replace the CGDO capacitor to more accurately model the change in capacitance with gate and drain voltage [5]. EGD is a voltage generator equal to M1's gate-to-drain voltage which is used to supply voltage to the feedback capacitance emulating subcircuit. VFB is a voltage generator used to monitor the current in the feedback capacitance emulation subcircuit for FFB. FFB is a current controlled current source used to inject the feedback current back into M1. EGD, VFB, and FFB provide the necessary power to drive the feedback components in parallel without loading M1. They also permit ground connections in the subcircuit, improving convergence and accuracy. CGD is the fixed part of the gate-to-drain capacitor. R1 and D1 limit its operation to the region where the gate voltage exceeds the drain voltage. DHV is a diode which emulates the gate-to-drain capacitor at high voltages. R2 and D2 limit its operation to the region where the drain voltage exceeds the gate voltage. DLV is a diode which emulates the gate-to-drain capacitance variation with drain voltages (variable part of Cres) below the transition voltage. The multiplier (=C1/C2 - 1) used is determined by the size of the capacitance step needed. RLV shunts its current to ground at higher voltages. ESD is a voltage controlled voltage source that senses source-to-drain voltage and drives MLV. The POLY form is used so that the proper offset voltage can be inserted without an additional element. MLV is used as a switch to disconnect DLV from the feedback at higher voltages, emulating the drastic reduction in feedback capacitance with voltage found in most modern IGBTs.

LE emulates the emitter lead inductance. 7.5 nanohenries represents the lead inductance of a TO-220 plastic package. The total lead inductance Le is an important high speed limit parameter and should include all external lead inductance through which output current flows before it reaches the common ground with the drive circuit. The inductance of the drain and gate leads have little effect on simulations but could be easily added to the subcircuit. You may, however, want to add in 7 nH per cm. or 18 nH per inch for any PCB traces or wires. Typical internal inductances are: TO-220 (plastic): 7.5 nH , TO-218 (plastic): 8 nH (1 bond wire), 4 nH (2 wires), TO-204 (TO-3) (metal): 12.5 nH [6].

Software Solution To Modeling Headaches

If entering and adjusting all of these parameters seems a little too complex and time-consuming, you can take the easy way out and generate your IGBT subcircuit using SpiceMod, a general purpose SPICE modeling program that supports IGBT model development. SpiceMod derives SPICE parameters from generally available data book information. The most unique feature of SpiceMod is its estimation capability. If some of the data sheet parameters are not available, SpiceMod will provide estimates for data not entered based on the data that is entered. Thus, SpiceMod will never leave a key SPICE parameter at its default value.



```
SPICEMOD 2.1 ===================== IGBT (X) ======================== 12-14-1994
                       .SUBCKT Name (IRGBC40U) = IRGBC40U        Affects:
                            Channel Type (N,P) = N               --------
Collector-to-Emitter Breakdown Volt,BVCES =   600.000 V.     BV in DO
Emitter-to-Collector Breakdown Volt,BVECS =    15.000 V.     BV in DE
 Max. Continuous Collector Current, ICmax =    40.000 A.     All Parameters
       C-to-E Saturation Voltage, VCE(on) =     2.700 V.     RC, RS
        at (Maximum) Test Current, IC(on) =    40.000 A.
              and (Maximum) Bias, VGE(on) =    15.000 V.
          Gate Threshold Voltage, VGE(th) =     5.200 V.     VTO
            Forward Transconductance, gfe =    14.000 S.     KP
                at (ICmax/2) Current, IC  =    20.000 A.
                 Output Conductance, goe =    80.267 mmohs  LAMBDA, ETA
            Capacitance Test Voltage, VCE =    30.000 V.     (above VCT)
                 Input Capacitance, Cies =  1429.630 pF.     CGE
                Output capacitance, Coes =   194.430 pF.     CBD (CJO in DO)
       Reverse Transfer Capacitance, Cres =     7.027 pF.     CGD (CJO in DR)
      Capacitance Transition Voltage, VCT =    19.000 V.     ELV offset
Capacitance Shift in Cres at VCT, (C1/C2) =    14.000        DLV multiplier
     Resistance for Switching Tests, Rgen =    10.000 Ohms   RG
             Turn-Off Delay Time, td(off) =   145.465 ns.    RG
                            Fall Time, tf =    97.461 ns.    TF
        Current Step at Turn-Off, (I2/I1) =     0.836        BF
              Emitter Lead Inductance, Le =     7.500 nH.    LE (0 = none)
SELECT WITH ARROWS, TYPE DATA   F1=HELP  F2=SUBCKT  F3=RESET DATA  Exc=EXIT/SAVE
```

Figure 3, Data sheet parameters (above left) used to create the SPICE IGBT subcircuit (Table 1). To make a new model, data sheet values are entered into the SpiceMod entry screen. As they are entered the subcircuit values are calculated. The more data that is entered, the more accurate the final model will be. The subcircuit parameters affected by each entered parameter.

This is the downfall of many modeling programs and can cause the resulting SPICE model to be invalid. Figure 3 shows the input parameters from the data book and the SPICE parameters that are primarily affected by each input for the device in Table 1.

SpiceMod is so intelligent that a reasonable first order device model can be obtained by simply entering the voltage and current ratings of the device. Of course, the more data entered, the more accurate the final model. In addition to IGBTs, SpiceMod also produces models for diodes, zeners, BJTs, JFETs and MOSFETs, and subcircuit macromodels for power transistors, Darlington transistors, power MOSFETs, and SCRs [7]. All of the models are Berkeley SPICE 2G compatible and can be used with any SPICE program on any computer platform. Detailed next are the DC and Transient performance characteristics of the outlined IGBT model.

IGBT Testing

Figure 4 shows the output characteristics of the IRGBC40U as simulated by IsSpice4, a native mixed mode SPICE 3F based simulator. Note the offset from zero caused by the base-emitter diode of the PNP. The slight slope of the curves, controlled by ETA, represents the output impedance. The values are well within the data sheet tolerances without any need for optimization. This is not surprising given the possible variation in the device's gfe. However, it is easy to see that with the simple circuits provided here, it is quite easy to tweak the model performance for a given situation.

The model exhibits forward and reverse breakdown effects. Although not normally operated in these modes, inductive flyback effects can easily drive an IGBT into one or both of these regions. Because IGBTs are frequently used in switching power supplies, this is not an unusual occurrence. Excess energy in reverse breakdown was a frequent killer of early IGBTs.

Figure 5 shows the capacitance variations verses gate and collector voltages for the model. The X-axis is collector-to-gate voltage, so the left part with negative voltages actually represents positive gate voltage while the right part represents positive collector voltages.

Note that all capacitance tests are made with the IGBT in a non-conducting mode. In normal operation the capacitive feedback current is multiplied by (BF+1) at the output, so BF is an important parameter.

The circuit in Figure 7 (TSWITCH.CIR) is used to simulate various switching effects. The current generator available in IsSpice4 replaces the inductor and two other switching devices normally used for this test. Note the two-input voltage controlled current source that is added to multiply the IGBT voltage and current to compute power (measured across the one-ohm resistor). This power (current) is then integrated by the capacitor CE to get energy (as voltage). The multiplication and integration could have just as easily been done in a SPICE post-processing program. However, when the waveforms are calculated by IsSpice4 the simulated waveforms can be cross-probed directly on the schematic as shown in Figure 7. It should be noted that the data sheet values for switching characteristics can be greatly affected by the test circuit and test load used. Care should be given to properly constructing the test circuit based

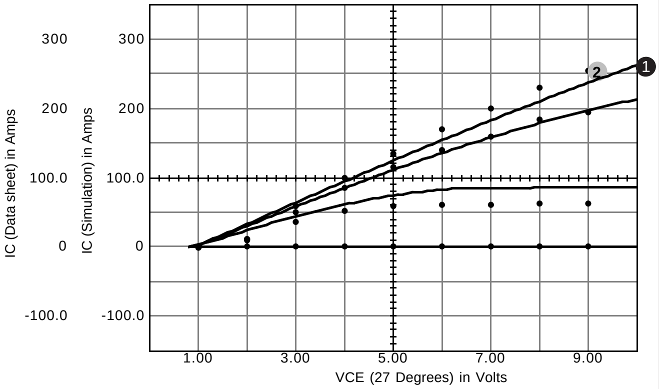


Figure 4, Data sheet (waveform 2, dots) and simulated (waveform 1, solid) output characteristics for the IRGBC40U.

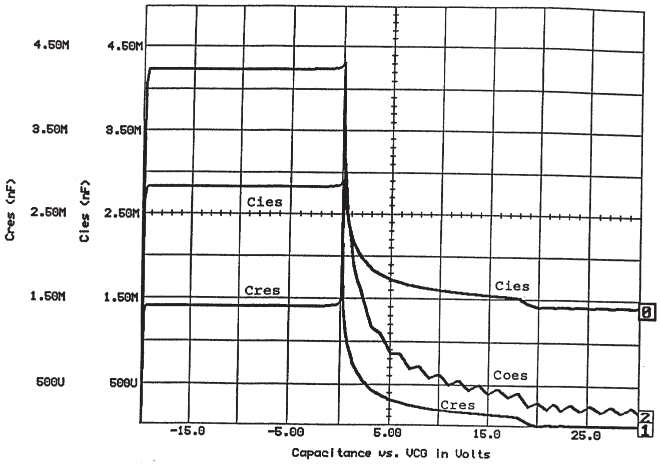


Figure 5, Simulated capacitance characteristics for the IRGBC40U. All waveforms are scaled the same.

```
TSWITCH.CIR - Device Switching Characteristics
.PRINT TRAN V(3) V(4,3) V(5,6) V(6) I(VC)
.IC V(6)=0
.TRAN 2N   1000N
*ALIAS  V(6)=ESW
*ALIAS  V(3)=VOUT
RIN 1 2 10  ;SET TEST R(GEN)
X1 30 2 0 IRGBC40U  ;REPLACE WITH YOUR DEVICE NAME
VC 3 30
IL 0 4 20  ;SET TEST CURRENT
RL 3 4 .01
GPWR 0 5  POLY(2) 3 0 4 3 0 0 0 0 100
* MULTIPLIES VOLTAGE AND CURRENT TO YIELD POWER AS V(5,6)
RPWR 5 6 1
CEN 6 0 1  ;INTEGRATES POWER TO GIVE ENERGY/PULSE AS V(6)
D2 0 4 DZEN
.MODEL DZEN D(BV=480 IBV=.001)
*              ^ SET TEST VOLTAGE
REN 6 0 1E6 ;PROVIDES DC PATH TO GROUND
VIN 1 0 PULSE  0    15    0    1N   1N   200N  1000N
.END
```

Figure 6, The switching circuit TSWITCH.CIR (below) used to test the transient IGBT performance.

on the data sheet information, otherwise the simulation results may not be comparable with the actual performance.

Switching losses are calculated by multiplying the IGBT current and voltage waveforms during the switching period (figure 8). Note that the voltage does not begin to fall until the current reaches maximum and that the current does not begin to fall until the voltage reaches maximum. Note the long tail on the current waveform due to the PNP (controlled by TF).

Figure 9 shows the instantaneous power and cumulative energy curves which match the curves in Figure 8. Note that the scale is millijoules, so the final value is 1.5 millijoules.

Temperature Effects

Diode voltage shifts due to temperature are properly modeled by SPICE, but others are not well emulated. Resistive shifts with temperature can be approximated by adding a temperature coefficient to RC (**`RC 85 71 21.1M TC=.01`** for SPICE 2, or **`RC 85 71 21.1M RMOD & .MODEL RMOD R TC1=.01`** for SPICE 3). This was not included in the subcircuit because it can cause error messages due to differences in SPICE implementations from some vendors. Temperature effects can best be handled by entering data book parameters at temperature into the subcircuit for an accurate high temperature model.

Example Usage: 3 Phase IGBT Inverter

As a practical example, a 3 phase inverter with simplified motor load was simulated (Figure 10). The IGBT model allows examination of both circuit and IGBT related design issues. For the inverter circuit, Figure 10 shows the line-line and line-neutral quantities, as well as the IGBT switching waveforms. In Figure 11, the effect of varying the load inductances (LA, LB, and LC) is displayed. The control circuitry has been simplified so as not to unneces-

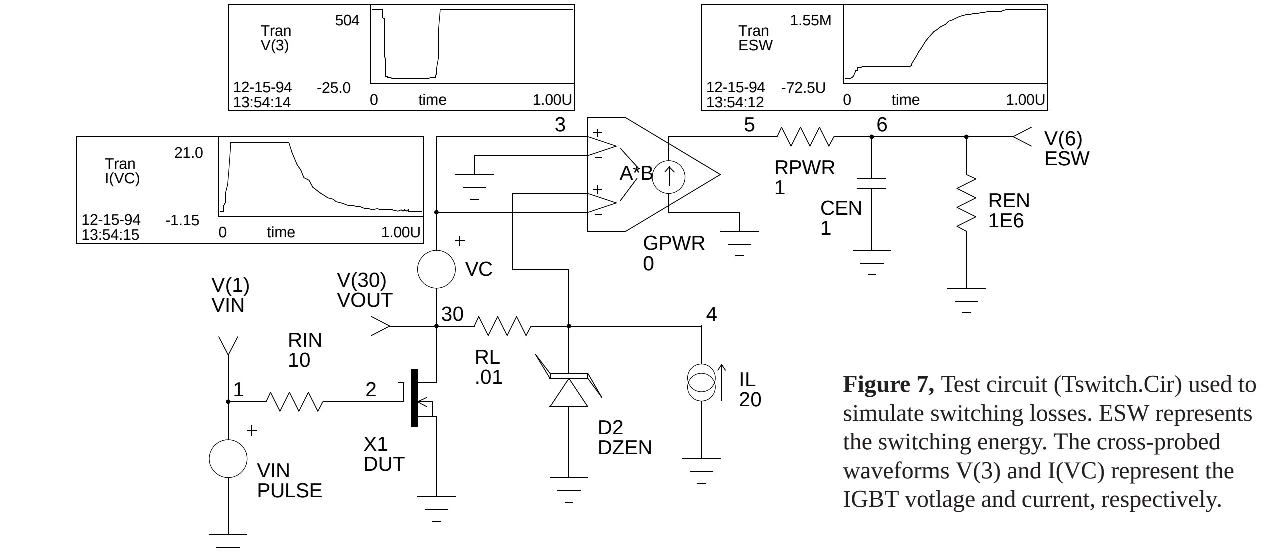


Figure 7, Test circuit (Tswitch.Cir) used to simulate switching losses. ESW represents the switching energy. The cross-probed waveforms V(3) and I(VC) represent the IGBT votlage and current, respectively.

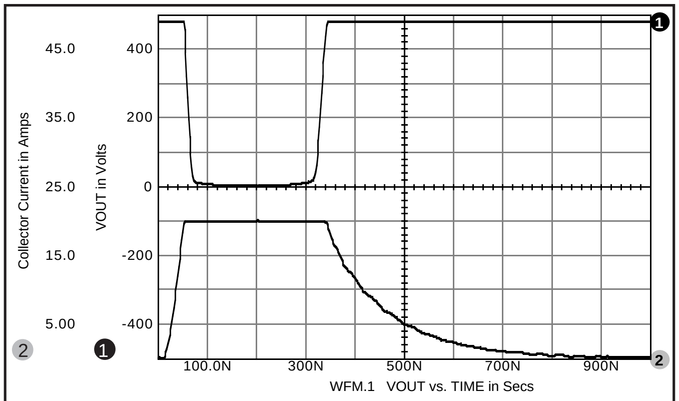


Figure 8, Switching losses are calculated by multiplying the current and voltage waveforms during the switching period.

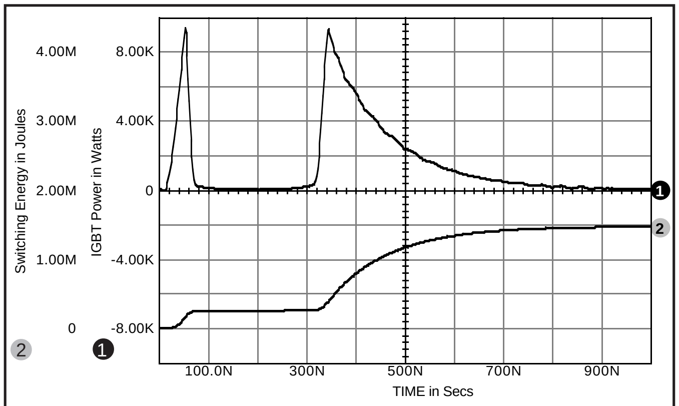


Figure 9, The instantaneous power (waveform 1) and cumulative energy (waveform 2) curves which match the curves in figure 8.

sarily complicate the simulation. An anti-parallel diode has been included in the IGBT subcircuit used in this simulation by adding a diode from nodes 74 to 71. For those of you who think that such simulation are beyond the capability of PC, on a 90MHz Pentium the 166 element inverter circuit runs in 28.05 seconds. On a 275MHz Digital Alpha AXP PC) it runs in under **<u>6 seconds</u>**!!

Issues such as parallel IGBT operation, overcurrent/short circuit protection circuitry, and various snubber configurations can also be explored with the model.

Conclusions and Future Work

There are a number of ways to better model the nonlinear gate-drain capacitance. An enhanced method using the SPICE 3 B element is described in [8]. It uses half the number of elements and allows alternate capacitance responses, such as a sigmoidal response, to be constructed. More importantly, [9] describes a new AHDL (Analog Hardware Description Language) based on 'C' that will allow much more accurate and efficient IGBT models to be developed.

A SPICE IGBT subcircuit has been developed that relates well to data book information. It models the DC collector family and on- voltages, non-linear capacitance effects, and switching characteristics. Forward and reverse breakdown characteristics are also included.

The model finally gives power engineers the ability to simulate all types of IGBT based circuits [9]. An intelligent modeling program has been introduced that quickly generates custom SPICE subcircuits from data supplied by the user and estimates reasonable values for any missing data by scaling from the supplied data.

References

[1] Charles E. Hymowitz, Intusoft Newsletter, "Intusoft Modeling Corner", Intusoft, June 1992, San Pedro, CA 90731

[2] "Insulated Gate Bipolar Transistor Designer's Manual", International Rectifier, El Segundo, CA 90245

[3] Andrei Vladimirescu and Sally Liu, "The Simulation of MOS Integrated Circuits Using SPICE", UCB/ERL M80/7, University of California, Berkeley, CA 94720

[4] Paolo Antognetti and Giuseppe Massobrio, "Semiconductor Device Modelling with SPICE", McGraw-Hill, 1988

[5] Charles-Edouard Cordonnier, Application Note AN-1043, "Spice Model for TMOS Power MOSFETs", Motorola Inc. 1989

[6] Lawrence G. Meares and Charles E. Hymowitz, "Simulating with SPICE", Intusoft, San Pedro, CA 90731

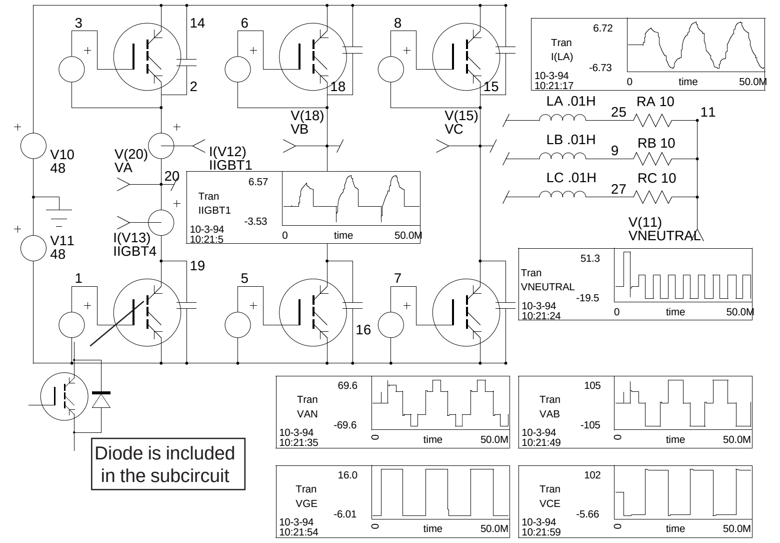


Figure 10, A 3 phase inverter circuit. Cross-probed waveforms show the line-line and line-phase voltages, the phase A current, and the IGBT switching waveforms.

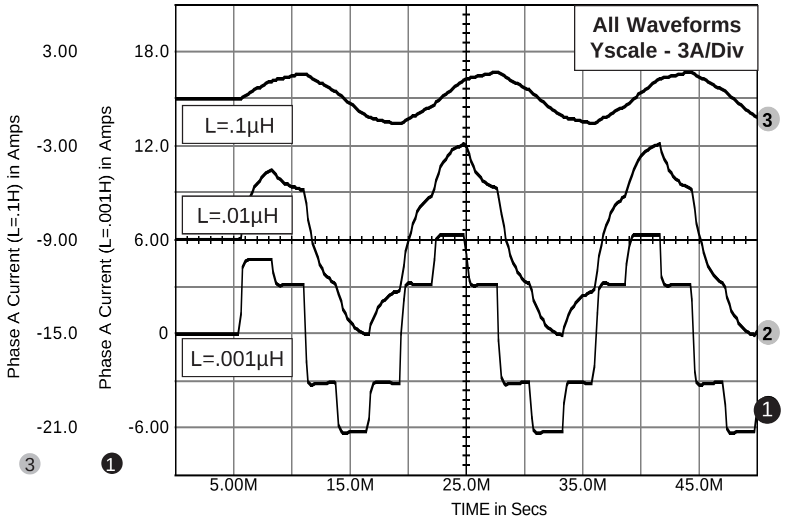


Figure 11, The waveforms show the phase A current for three different simulations where the load inductances LA, LB, and LC were varied (.1µH, .01µH, and .001µH).

[7] SpiceMod User's Guide, Intusoft, June 1990, San Pedro, CA 90731

[8] Charles E. Hymowitz, Intusoft Newsletter, "New Technique Improves Power Models", Intusoft, June 1992, San Pedro, CA 90731

[9] Charles E. Hymowitz, Intusoft Newsletter, "3 Phase IGBT Inverter" & "New AHDL Based On 'C'", October 1994, San Pedro, CA 90731

Sample models for several IGBT devices are available free of charge on the Compuserve CADD/CAM/CAE Vendor forum, Library 21 (Go CADDVEN at any ! prompt) for Compuserve users and an ftp site (ftp.iee.ufrgs.br.) for Internet users. The SpiceMod program is available from Intusoft.

SPICE 3 Models Constant Power Loads

Charles Hymowitz, Intusoft, USA, e-mail: charles@intusoft.com

October 1996

In power supply design and testing a load that draws constant power is often needed. For example, for battery-energy measurements using a simple resistor as a load is unacceptable for applications that require you to accurately monitor generated or consumed power.

Battery energy density (watts/hour) is normally found by measuring the total output power (W=V×I) over time. This task is relatively simple if you use an automated system to take frequent and periodic measurements of the discharge current and battery voltage while keeping track of time. Monitoring discharge time using a simple resistor or a constant-current sink as a load yields an inaccurate result, particularly when you want to compare the performance of different battery chemistries. The various battery types have many discharge-voltage characteristics, and power output varies as the output voltage changes. To maintain constant power, you must adjust the discharge current to compensate for battery-voltage variation during discharge. As the battery voltage drops, load current must increase to maintain a constant V×I.

Constant-power loads can exhibit a negative incremental input resistance within the regulation bandwidth of the power converter. Distributed power systems that include a large percentage of constant-power loads and contain energy storage devices may be susceptible to potentially destabilizing interactions of these elements.

To model such a load for simulation purposes, we can use the Berkeley SPICE arbitrary dependent source B element. For example:

```
B1 N1 N2 I={Power_Load}/V(N1,N2)
```

For this expression, the power is equal to V*I which is equal to

V(N1,N2)*(Power_Load/V(N1,N2))
= Power_Load as desired.

The problem with this approximation is that it undefined near V(N1,N2) = 0. When calculating the small signal bias point or initial transient solution for a circuit, IsSpice4 scales back the GMIN and independent source values. IsSpice4 relies on the assumption that, when the power supplies are close to 0, all devices in the circuit are turned off. The B1 statement violates this assumption. In any case, it is also not a good model of a real constant power load for low voltages.

A realistic constant power load can only consume power over a limited range of applied voltage. When the voltage drops below this range, the load's impedance stops falling. For certain types of loads, a series connection of two resistances can be used with one fixed and one variable. This can be written as:

$$R_{total} = Rmin + Rvar = Rmin + V^2/P \text{ -->}$$
$$I = V/R_{total} = V/(Rmin+V^2/P) = 1/(Rmin/V+V/P).$$

For low voltages, I=V/Rmin, for high voltages, I=P/V.

The crossover occurs at:

$$Rmin/V = V/P \text{ -->} V^2 = RminP \text{ -->} V^2/Rmin = P$$

when the power dissipated in Rmin equals the desired power, P. This is the point of maximum power dissipation into Rmin. For higher voltages the current falls and most of the power is dissipated by Rvar.

The corresponding IsSpice4 statement is:

```
B1 N1 N2 I=1 / (RMIN/V(N1,V2) +
+ V(N1,N2)/PLOAD)
```

where RMIN and PLOAD must be defined before simulation. The RMIN and PLOAD values can be passed to B1 by using the following syntax:

```
.PARAM RMIN=val1 PLOAD=val2

B1 N1 N2 I=1 / ({RMIN}/V(N1,V2) +
+ V(N1,N2)/{PLOAD})
```

where val1 and val2 are numbers or expressions.

References

[1] "Load simulator maintains constant power", Khy Vijeh, Analog Devices, Santa Clara, CA, EDN Design Idea April 11, 1996

[2] Microsim, The Design Center, Application Notes, 1996

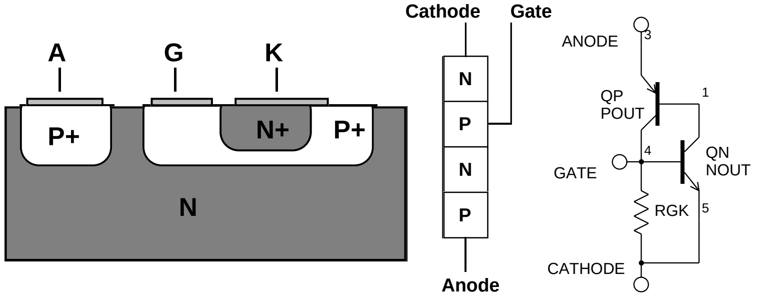
Simulating Circuits With SCRs

When modeling devices for simulation, the best model is one that accurately emulates the real part, simulates efficiently, converges well, and is easy to create. A new SCR model, implemented in the forth coming version of the SpiceMod, SPICE modeling spreadsheet program, is just such a model. This article explains the model, which is based on the two transistor [1,2] and four transistor Intusoft model [3], and provides an example of its use.

The Basic SCR Model

Figure 1 shows the design of a typical planar "shorted emitter" SCR and the classical two transistor combination used to model the positive feedback effects of the four semiconductor layers. The NPN transistor models the three layers at the cathode end and the PNP transistor models the three layers at the anode end. The two center layers are common to both transistors.

Figure 1, Cross section of a planar shorted emitter SCR and its basic equivalent representation using NPN and PNP transistors.



The cathode (emitter) metallization overlaps the gate (base) of the NPN transistor, forming a leakage path to carry leakage currents around the gate until the gate is triggered from an external source. This is shown as RGK in the model.

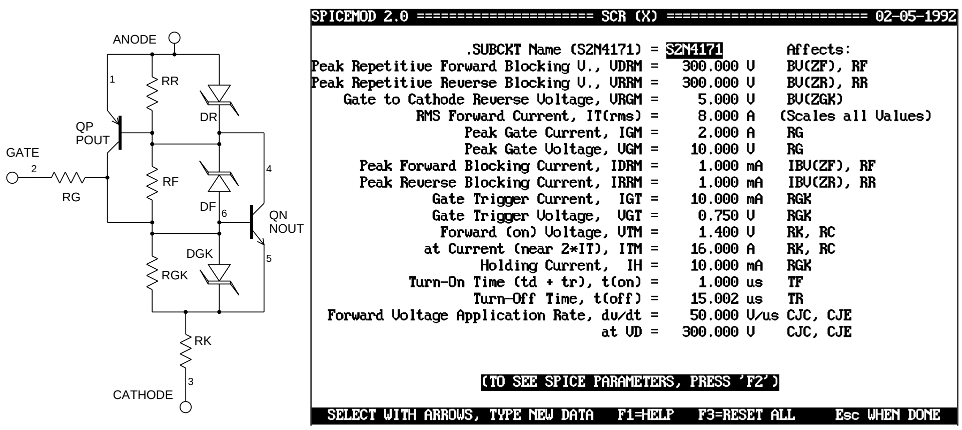
The base of each transistor is connected to the opposite collector forming positive feedback as soon as both base-emitter junctions become forward biased. Although this model has the triggering function of an SCR, it must be enhanced to emulate other parameters such as off state leakage, break-over voltage and current, reverse breakdown characteristics and dv/dt turn-on.

SpiceMod Improves The SCR Model

Figure 2 shows the additions to the basic model. Table 1 lists a sample .SUBCKT netlist from the PreSpice libraries. For *Intusoft Newsletter* subscribers, several additional new SCR models have been included on the enclosed floppy disk. The availability of the new SpiceMod program, containing new SCR, IGBT, and Zener models will be announced in the next newsletter.

Simulating Circuits With SCRs

Slim Petrie, Charles Hymowitz, Excerpts from the February 1992 Intusoft Newsletter #24



```
SPICEMOD 2.0 ===================== SCR (X) ======================== 02-05-1992

                        .SUBCKT Name (S2N4171) = S2N4171         Affects:
Peak Repetitive Forward Blocking V., VDRM =    300.000 V    BV(ZF), RF
Peak Repetitive Reverse Blocking V., VRRM =    300.000 V    BV(ZR), RR
    Gate to Cathode Reverse Voltage, VRGM =      5.000 V    BV(ZGK)
              RMS Forward Current, IT(rms) =      8.000 A   (Scales all Values)
                     Peak Gate Current, IGM =      2.000 A    RG
                     Peak Gate Voltage, VGM =     10.000 V    RG
         Peak Forward Blocking Current, IDRM =      1.000 mA   IBV(ZF), RF
         Peak Reverse Blocking Current, IRRM =      1.000 mA   IBV(ZR), RR
                  Gate Trigger Current,  IGT =     10.000 mA   RGK
                  Gate Trigger Voltage,  VGT =      0.750 V    RGK
                 Forward (on) Voltage, VTM =      1.400 V    RK, RC
               at Current (near 2*IT), ITM =     16.000 A    RK, RC
                     Holding Current,  IH =     10.000 mA   RGK
              Turn-On Time (td + tr), t(on) =      1.000 us   TF
                      Turn-Off Time, t(off) =     15.002 us   TR
  Forward Voltage Application Rate, dv/dt =     50.000 V/us CJC, CJE
                                     at VD =    300.000 V    CJC, CJE


                    (TO SEE SPICE PARAMETERS, PRESS 'F2')

 SELECT WITH ARROWS, TYPE NEW DATA   F1=HELP   F3=RESET ALL     Esc WHEN DONE
```

Figure 2, SpiceMod takes the data sheet parameters shown above and generates an IsSpice .SUBCKT model (schematic on left) for the SCR. Parameters not available will be estimated by SpiceMod based on the data that is entered insuring a realistic model.

Forward Break-over/Reverse Characteristics

The following parameters are used to calculate various forward and reverse characteristics. Only parameters that are generally available in data sheets are used because this is normally the only data that engineers have access to.

Parameters For Forward Break-Over and Reverse Characteristics

VDRM = Max. Forward Voltage
IDRM = Leakage Current @ VDRM
VRGM = Max. Reverse Gate Voltage
VGT = Gate Trigger Voltage
IH = Holding Current

VRRM = Max. Reverse Voltage
IRRM = Leakage Current at VRRM
IT = RMS Forward Current
IGT = Gate Trigger Current

QN, the NPN transistor, has a forward beta of 100 and contains the gate at its base. QP, the PNP transistor, has a forward beta of one to emulate the (lower gain) high voltage junction near the anode. The resistor RF emulates the forward leakage current when the SCR is “off” (QN is off, but QP is on). Zener diode DF sets the breakdown voltage in the forward direction. Internal series resistance, RS, is added to this diode to produce a rounded peak.

$$RF = (BFp + 1) * VDRM / IDRM \qquad (1)$$

where BFp = BF of QP (set at 1)

for DF:

$$BV = VDRM \qquad (2)$$

$$IBV = IDRM / 10 \qquad (3)$$

$$RS = RF / 10 \qquad (4)$$

Resistor RR emulates the leakage when the SCR is reverse biased. The base-emitter junction of QP sees most of the voltage in this mode. The zener diode, DR, sets the reverse breakdown point. Diode DGK serves a similar function for QN, but its breakdown voltage is set between five to seven volts (VGR).

Forward /Reverse Characteristics *con't*

$$RR = VRRM / IRRM \tag{5}$$

for DR:

$$IS = .001 \ast ISp \tag{6}$$
$$BV = VRRM \tag{7}$$
$$IBV = IRRM / 10 \tag{8}$$

for DGK:

$$IS = .001 \ast ISn \tag{9}$$
$$BV = VRGM \tag{10}$$
$$IBV = IDRM / 10 \tag{11}$$

Because these diodes shunt the base-emitter junctions, their IS (leakage current) is made 1000 times smaller than the IS of the transistors to minimize their effect on these junctions. It is also important not to use any base or emitter resistance (RB, RE) in these transistors as they would cause voltage drops that would forward bias the diodes, robbing current from the transistors.

Gate Characteristics

Parameters Used For Gate Characteristics

IGM = Peak Gate Current

VTM = Forward (on) Voltage

VGM = Peak Gate Voltage

ITM = Forward (on) Current

It is necessary to add a cathode resistance to emulate the increase in gate voltage with cathode current. RK serves this function. The rest of the "on" voltage is simulated with model parameter RC in QN, eliminating the need for an additional external part. A gate resistor, RG, has been added to properly simulate the resistance found when over-driving the gate.

$$R(sat) = (VTM - VD) / ITM \tag{12}$$
$$RK = .2 \ast R(sat) \tag{13}$$
$$RC = .8 \ast R(sat) \tag{14}$$
$$RG = ((VGM - VD) / IGM) - RK \tag{15}$$

where VD = B-E diode drop (approx. VGT)

Because this model has separate resistors shunting each junction, convergence is improved over other models.

Dynamic Characteristics

Switching time limits are modeled using NPN/PNP charge storage model parameters. Note that TF is the forward transit time, not the fall time, and TR is the reverse transit time, not the rise time.

$$TF = .179 \ast t(on) \tag{16}$$
$$TR = 1.7 \ast t(off) \tag{17}$$

Parameters Used For Dynamic Characteristics

t(on), td+tr = turn-on Time
t(off), tq = turn-off Time
dv/dt = Forward Voltage Application Rate

The dv/dt limit is reached in an SCR when a change in the anode voltage flowing through the output capacitance (COE) causes sufficient current to flow in the gate circuit to turn the SCR on. The effective collector capacitance (CJC) can be calculated using the dv/dt specifications:

$$COE = IH\ /dv/dt \tag{18}$$

$$CJC = COE\ (\text{adjusted for test voltages}) \tag{19}$$

$$= COE * (1+VD/VJE)\hat{}MJE \tag{20}$$

where VJE=.75 and MJE=.33 (IsSpice default). The input capacitance, which is seldom given in data books, is estimated at five times the collector capacitance:

$$CJE = 5 * CJC \tag{21}$$

```
.SUBCKT C180    1 2 3
*    TERMINALS:  A G K
*  Powerex 100 Volt  150 Amp  SCR
QP  6 4 1  POUT OFF
QN  4 6 3  NOUT OFF
RF  6 4    10K
RR  1 4    6.66K
RG  6 3    10
RGS 2 6    4.66M
DF  6 4    ZF
DR  1 4    ZR
DG  6 3    ZG
.MODEL ZF   D (IS=94F IBV=2M BV=100 RS=1.5K)
.MODEL ZR   D (IS=94F IBV=2M BV=133)
.MODEL ZG   D (IS=94F IBV=2M BV=5)
.MODEL POUT PNP (IS=94P BF=1 CJE=37.8N TF=337N TR=170U)
.MODEL NOUT NPN (IS=94P BF=100 RE=233U RC=233U
+ CJE=37.8N CJC=7.56N TF=337N TR=170U)
.ENDS
```

Table 1, IsSpice subcircuit listing for the Powerex C180 phase control SCR with VDRM=100V and IT(avg)=150A.

Simulating A Resonant Regulator

Using a model for an SCR, (Philips BTW38) a simple resonant buck regulator (Figure 3) was constructed. The L1/C2 combination has a resonant frequency of 4kHz. The SCR is controlled by the pulsed current source, I2. Of note is the fact that the damping network across diode D1 is needed to reduce ringing at the cathode which might turn the SCR on. Without the R/C network, the SPICE simulation may abort unless the .OPTIONS parameters RELTOL is relaxed to .01. The constant current source I1, models a constant load. In order to achieve a steady state response at the beginning of the simulation, its value must be adjusted to the average value of the current flow in the inductor.

Conclusions

A SPICE SCR subcircuit has been developed that relates well to data book parameters. It emulates both the forward and reverse characteristics while avoiding convergence problems. However, the breakdown behavior, emulated in this and other models, may cause convergence problems due to various cascading

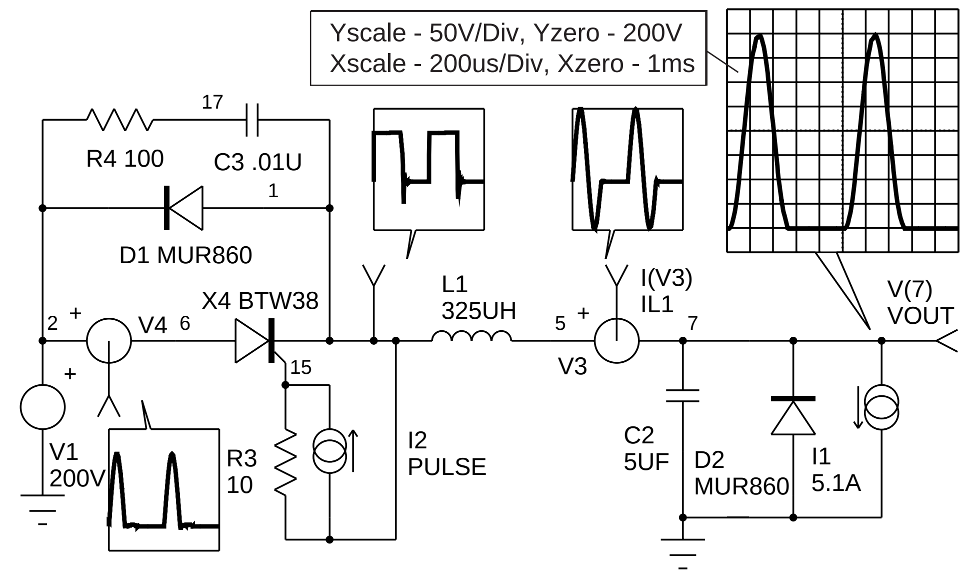


Figure 3, Using the new SCR model, a resonant buck regulator circuit can be simulated.

nonlinearities. Therefore, aborted simulation results should be checked to see if the device ratings were exceeded, especially near the end of the simulation. Sensitive gate SCR devices can be modeled because the internal RGK will generally have a large value, thus allowing the addition of an external resistor. GTOs, which are similar to SCRs except for a smaller gate resistor, can also be modeled by putting in the proper Peak gate current and voltage.

References and Biography

[1] "A Practical SCR Model for Computer Aided Analysis of AC Resonant Charging Circuits", R.L. Avant, F.C. Lee, and D.Y. Chen, IEEE, July, 1981.

[2] "A SPICE Model for the SCR", Waiman, F. Ki, Master Thesis, Department of E & CS, UC Berkeley, Dec., 1979.

[3] "PreSpice User's Guide", Intusoft, 1991.

[4] "Semiconductor Device Modelling with SPICE", Paolo Antognetti and Giuseppe Massobrio, McGraw-Hill, 1988.

[5] "Bipolar Power Transistor and Thyristor Data", Motorola Inc., DL111, REV3, 1984.

A great deal of thanks goes to the author of this article and developer of SpiceMod, the SPICE modeling Spreadsheet. A. F. "Slim" Petrie. Mr. Petrie is retired from Motorola Inc., where he was a Principal Staff Engineer and Dan Noble Fellow. He develops both hardware and software and holds 24 U.S. Patents. His "Circuit Analysis" program was sold through Apple computer in the early 1980s. With a keen appreciation for the problems of the working engineer, he continues to develop tools to make that job easier. He can be reached at 7 W. Lillian Ave., Arlington Heights, IL 60004, where he welcomes your comments.

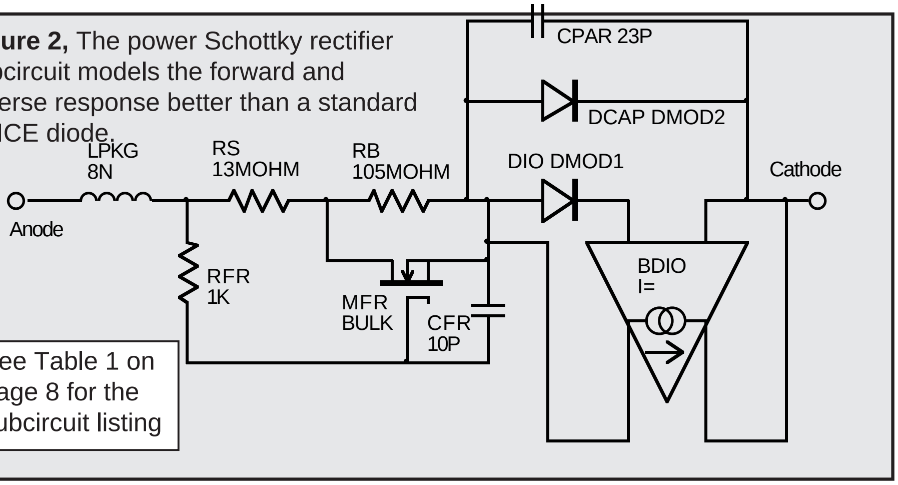
Power Schottky/Soft Recovery Diodes

Schottky rectifiers are widely used in switched mode converters due to their inherently lower forward voltage characteristics and superior recovery time. Schottky diodes can normally be modeled using the built-in SPICE diode. However, in the case of power Schottky rectifiers, two effects must be accounted for that are not part of the built-in SPICE model. The first is the conductivity modulation in the forward conductance and the second is the reverse leakage current which is an exponential function of both temperature and reverse voltage. IsSpice3 contains the functions necessary to construct a reasonable power Schottky model that adds these enhancements.

Figure 2 shows the subcircuit schematic of the new power Schottky rectifier. The diode, DIO, accounts for most of the traditional Schottky behavior. A Schottky diode is a majority carrier device whose reverse recovery time is zero. However, its larger junction capacitance, CJO, will produce a similar effect. Therefore, the minority carrier storage time parameter, TT, is left at its default value of 0. The emission coefficient, N, is normally set close to 1. The saturation current value, IS, is typically much higher (two orders of magnitude) than for p-n diodes. The values for saturation current temperature exponent, XTI =2, and energy gap, EG=0.69, are set based on the values expected for Schottky barrier diodes. In comparison to the manufacturer's reverse current data, the high temperature leakage appears to be slightly overstated by these values, but within likely product variations.

The voltage sensitive effects are produced by BDIO, an exponential nonlinear current source. Such a source would have been considerably more difficult to construct with SPICE 2G.6 based programs. With the in-line equation feature of IsSpice3,

Figure 2, The power Schottky rectifier subcircuit models the forward and reverse response better than a standard SPICE diode.



See Table 1 on page 8 for the subcircuit listing

Power Schottky and Soft Recovery Diodes
Charles Hymowitz
Excerpts from the September 1993 Intusoft Newsletter #32

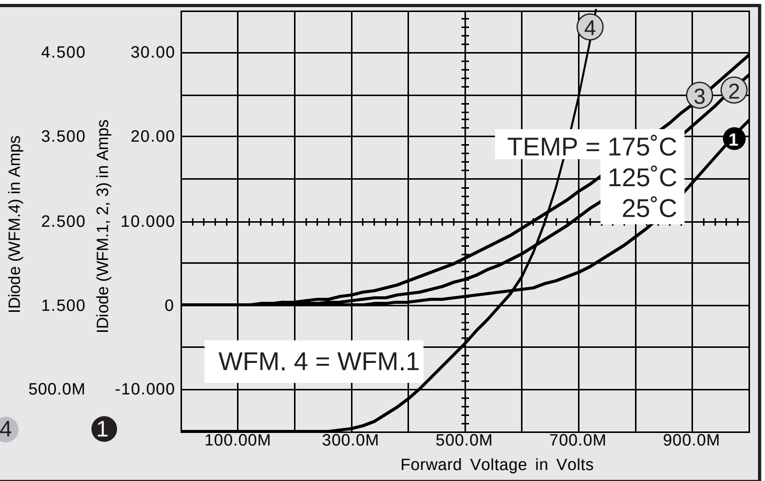


Figure 3, Graph of IF vs. VF at different temperatures. Alternate scaling for waveform 4 reveals the nonlinear response of the new model.

this function becomes trivial. The forward recovery of the diode, not readily verifiable because of the lack of data, is modeled by the time constant of RFR and CFR. Deleting CFR and shorting RFR makes the forward recovery instantaneous.

Figures 3 and 4 show the VF vs. IF response in the forward and reverse directions over temperature. All waveforms from BOTH graphs were obtained in one simulation pass using the simulation scripting features of IsSpice3. The netlist for the IR10CTQ150 (10A 150V) part used here is listed in Table 1 on page 8. Several other models are enclosed on the subscription floppy disk.

Modeling Soft Recovery

Modeling of the transient behavior of power diodes is extremely important for power electronics. By properly choosing the optimum diode recovery characteristic, the designer can significantly reduce switching losses, voltage spikes, RFI and EMI. As

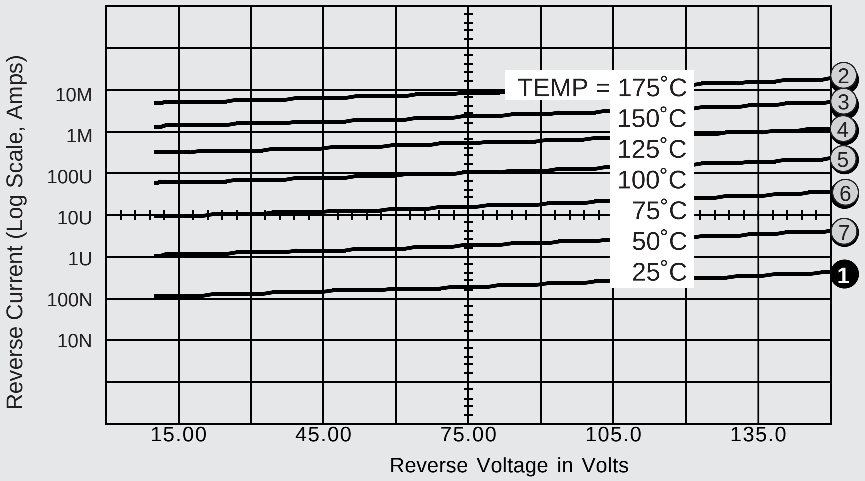


Figure 4, Graph of the reverse leakage current vs. temperature shows that the current is an exponential function of both temperature and voltage.

discussed earlier, the built-in diode model in SPICE does not exhibit forward recovery. In addition, the default diode model exhibits an abrupt “snappy” reverse recovery. With the help of IsSpice3 we will look at how to rectify the situation by enhancing the built-in model. The subcircuit presented here is based on a simplified physical model for a high voltage p-i-n structure operating in high level injection as is typical for most power diodes.

A variety of approximating models are possible for improving the accuracy of simulation without resorting to the use of C-code subroutines or other proprietary modeling languages. Both of these approaches suffer from a myriad of syntax, compatibility and logistical hurdles. One such model has been proposed in [1]. This model can easily be implemented with IsSpice3 or even SPICE 2, though somewhat less efficiently.

Reverse Recovery

SPICE’s internal recovery model of a diode only models charge storage as a function of the diode current. When the diode is commutated, the diode will remain on until all the charge is removed, after which, it will suddenly snap off. The reverse recovery time, Trr, consists of two components Ta and Tb. Ta is caused by the charge storage in the depletion region of the junction and represents the time between the zero current crossing and the peak reverse current, Irrm. Tb is caused by the charge storage in the bulk semiconductor material. Turn off proceeds with an ultimate reversal of the biasing current and the junction remaining forward biased (charged). This continues until the charge is removed from one of the junctions adjacent to the -i layer. The -i layer is still charged as the reverse voltage rapidly increases.

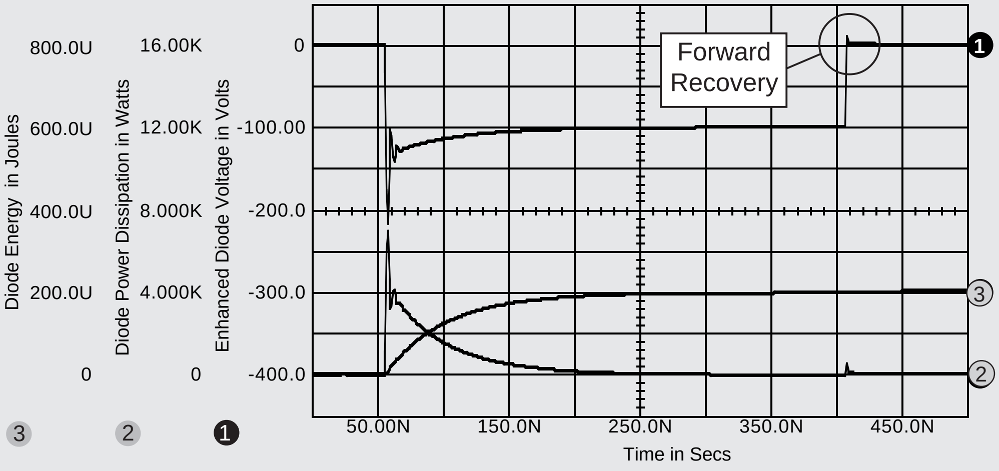


Figure 5, Curves showing the diode voltage, instantaneous power (Dpower), and energy (integral of Dpower) curves as the new diode subcircuit is turned off and on.

Such behavior is appropriate for small signal low-voltage diodes, but not for power diodes which are typically formed with a lightly doped layer between the p-n junctions to withstand high reverse voltages. No additional diode state variables are available in SPICE to represent the physical charge storage in the lightly doped internal regions of such a p-i-n structure. As a consequence, spurious oscillations are seen in SPICE simulations that attempt to use the standard diode model with nonzero values of TT. Not only can the simulation be inaccurate, but it can often refuse to converge. Adding to the confusion is the fact that the oscillation amplitude and duration can vary depending on the .TRAN TSTEP and RELTOL values chosen.

The enhanced diode subcircuit, DBEHAV, is shown in Table 1. The diode DMODEL models the space charge capacitance. Other characteristics are removed by making RS and TT=0 and IS small. The standard diode behavior and reverse/forward recovery effects are added using the in-line equation feature (B element) of IsSpice3. The parameter IS1 is similar to the IS diode model parameter. ISE and the RS device value are chosen for the best high current performance. N, shown in the BE element's equation (2∗VTA), is set to 2 for high level injection behavior, but can be set to 1 to function as a low-voltage p-n junction diode. Tm (transit time) and Tau (carrier lifetime) are determined from a diode turn-off current waveform. As TM increases, stored charge is removed more slowly from the base region. This has the effect of making Ta longer and Irrm larger. RMO controls the forward recovery.

High diode power dissipation often occurs during the final current tail interval as evidenced by the power dissipation and

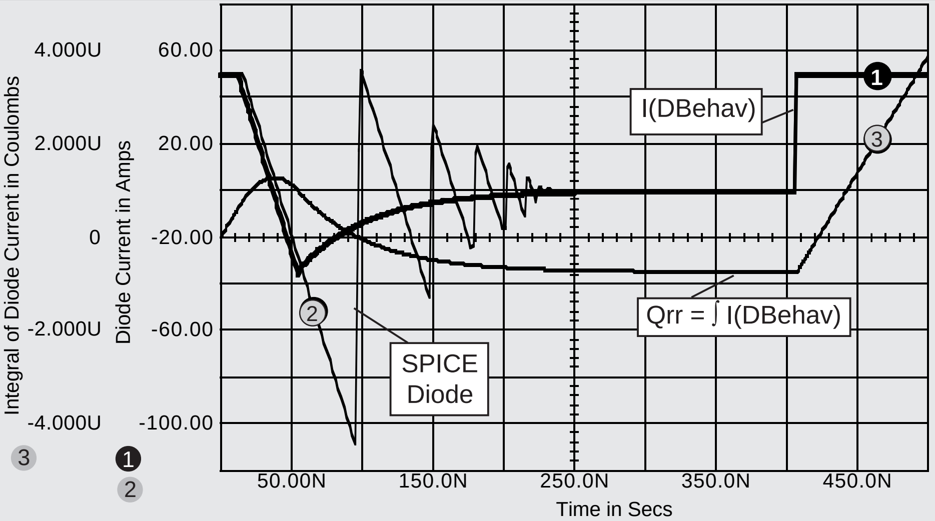


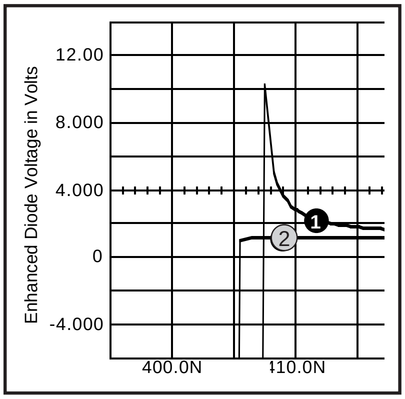
Figure 6, The diode current curves (WFM.1 & 2) reveal the softer turn off and lower Irrm response of the enhanced diode. Qrr is shown as waveform 3.

Reverse Recovery *continued*

energy curves in Figure 5. The power dissipation curve was created by the IsSpice3 Print Expression “alias dpower (v(3) - v(2)) ∗ i(vsense)”. The power was then integrated in IntuScope to get the energy. Figure 6 shows the diode current for the subcircuit diode (WFM.1) and for a comparable SPICE diode .model statement (.MODEL DMOD D IS=20F RS=4M N=1 CJO=100P M=.5 TT=100N, WFM.2). The Qrr charge curve from the new diode is shown as waveform 3. As shown in Figure 6, soft recovery diodes exhibit a faster recovery time and a lower value of Irrm than conventional diodes.

Forward Recovery

During the turn-on transient, a high forward voltage builds up across the diode because of the low conductivity in the -i region. As the injected carrier concentration increases, the voltage across the -i region soon decreases to a normal steady state diode forward drop. This effect gives rise to the initial voltage overshoot in the turn-on forward voltage as the -i region must become flooded with current carriers for its full conductivity to be effective. The forward voltage peak, as shown to the left and in Figure 5, is 10.197V. Waveform 2 shows the standard SPICE forward recovery.



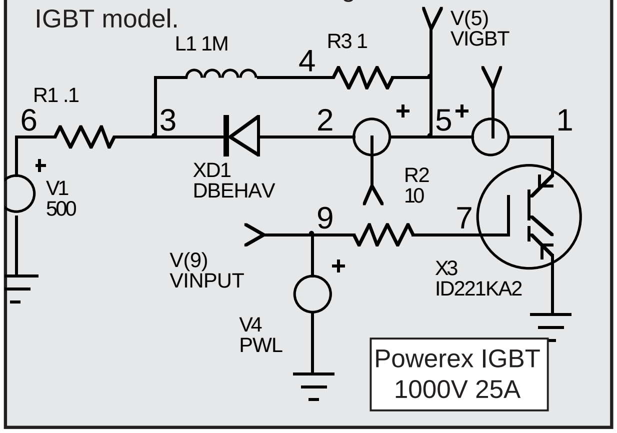
Simulating an IGBT Chopper Circuit

Figure 7 contains a simple IGBT chopper circuit using the new diode model. To simplify the circuit, the gate drive circuitry has been replaced by a piece-wise linear voltage source, V4. To study the diode behavior, the IGBT is turned on to generate current flow in inductor, L1, then turned off to cause current to flow into the diode (DBEHAV), then back on again to snap the diode off. The resulting diode turn off is shown in Figure 8.

Gear Integration Speeds Power Simulations

IsSpice3 includes a very valuable option, preferred by power engineers, to change to the Gear method of integration. Gear integration tends to find a stable time domain solution more quickly than the default trapezoidal method for many types of circuits. This results in a simulation that spends less time in high frequency calculations and is less prone to artificial numerical oscillations. SPICE simulators that do not have Gear integra-

Figure 7, IGBT chopper circuit using the enhanced diode model along with an Intusoft IGBT model.



tion can be more prone to convergence problems and often simulate more slowly than IsSpice3 for switching power circuits. Figure 8 shows the diode response using the Gear and trapezoidal integration methods.

SPICE = AHDL

Lauritzen and Ma incorrectly state in their paper [1] that macro models are normally valid only over a very narrow range of circuit operating conditions. The variety of models and verified simulation results presented in past newsletters is evidence that macro models are far more practical and robust than indicated. SPICE macromodels are the AHDL of the past and future. Macro modeling gives the user several advantages over other proposed methods; devices can be put together in a hierarchical manner, with pre-defined blocks whose convergence and behavior is well established and well understood by the user. And when the Berkeley syntax is adhered to, such as in the case of all Intusoft models, the models are portable between any vendor's SPICE simulator. Clearly, the new behavioral modeling (If-Then-Else and in-line equations) features of IsSpice3 are easy to incorporate into a model and contain virtually all of the AHDL elements often used by so-called "mathematical simulators". These features allow the user to bridge the gap between modeling with detailed physical based equations and pre-defined macro blocks in order to have the best of both modeling worlds.

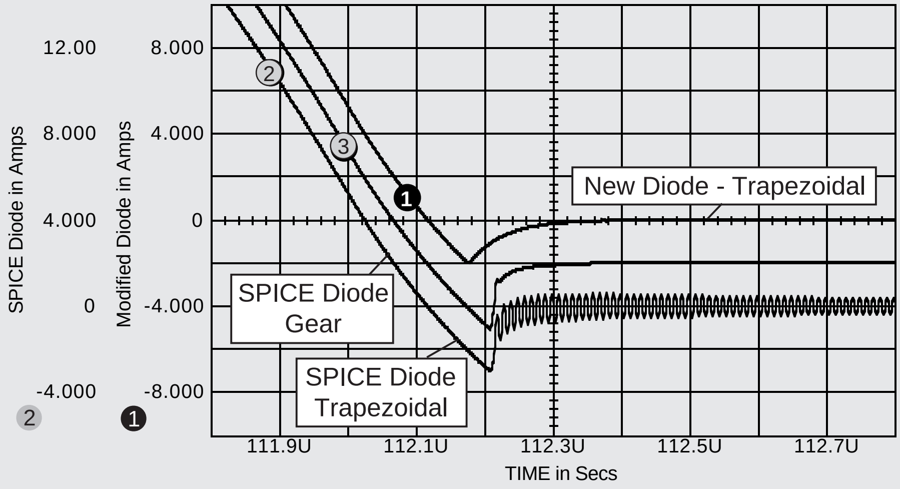


Figure 8, Turn-off response for the diode, Dbehav, in the IGBT chopper circuit, shows the different recovery and superior performance of Gear integration.

Table 1, The enhanced SPICE diode with forward/reverse recovery and the power Schottky rectifier model.

```
.SUBCKT DBEHAV 1   9 {IS1=1E-6 TM=100N TAU=100N RMO=1 VTA=.0259 CAP=100P ISE=1E-30}
* Connections         A C
* Following components include for space charge capacitance
DMODEL 1 2 DCAP
.MODEL DCAP D (IS=1E-21 RS=0 TT=0 CJO={CAP})               ; M/VJ can also be added
BD 1 2 I=V(5,6) / {TM} + {ISE} * (E^(V(1,2) / {VTA}) - 1)             ; Diode conductive currents
* Following components model reverse recovery
BE 5 0 V={IS1} * {TAU} * (E^(V(1,2) / (2*{VTA})) - 1)                  ; V(5)=QE, N = 2 in (2* {VTA})
*RE 5 0 1E6                                                              ; QE = Junction charge
BM 6 0 V=(V(5) / {TM} - I(VSENSE1)) * ({TM} * {TAU} / ({TM} + {TAU}))         ; V(6)=QM
*RM 6 0 1E6                                                              ; QM = Charge in the base region
BDM 7 0 V=V(6)
VSENSE1 7 8                                                              ; I(Vsense1)=dQM/dt
CDM 8 0 1
RDM 8 0 1E9
* Models high current effects and forward recovery
RS 2 3 4E-3
BMO 3 4 V=2*{VTA} * {RMO} * {TM} * I(VSENSE2) / ((V(6) * {RMO}) + ({VTA} * {TM}))
VSENSE2 4 9
.ENDS

.SUBCKT IR10CTQ150  1  2                          ; 150V 10A Power Schottky Diode
*CONNECTIONS            A C
LPKG 1 3 8N                                       ; Lead Inductance
RS 3 4 13MOHM
RB 4 5 105MOHM
RFR 3 6 1K                                        ; Forward Recovery
CFR 5 6 10PF                                      ; Forward Recovery
MFR 5 6 4 5 BULK
.MODEL BULK NMOS (VTO=.19 KP=320)
DIO 5 7 DMOD1
VDIO 7 2
.MODEL DMOD1 D(IS=50NA N=.9 XTI=2 EG=.69) ; Schottky Settings XTI=2, EG=.69
BDIO 5 2 I= I(VDIO) * EXP (.013 * V(2,5))         ; Voltage Sensitive Reverse Current Effect
DCAP 5 2 DMOD2
.MODEL DMOD2 D(CJO=420P M=.6 VJ=.7)
CPAR 5 2 23P
.ENDS
```

References

[1] P.O. Lauritzen and C.L. Ma, “A simple diode model with forward and reverse recovery”, IEEE Transactions on Power Electronics, Vol. 6, No. 2 Apr. 1991

[2] Gerald Stanley, “A simple SPICE3 Model for Schottky Power Diodes”, Crown International, 9-92, private correspondence

[3] Gerald Stanley, “SPICE32 Rect(n) Functions”, Crown International, 9-92, private correspondence

[4] Ned, Mohan, “Power Electronics: Computer Simulation, Analysis, and Education”, Minnesota Power Electronics Research & Education, P.O. Box 14503, Minneapolis, MN 55414, 1992

[5] International Rectifier, “Power Modules Designer's Handbook”, 1991

[6] Y. Khersonsky, M. Robinson, D. Gutierrez, “The Hexfred™ Ultrafast Diode in Power Switching Circuits”, AN-989, International Rectifier, 1992

[7] Unitrode, “Semiconductor Databook and Application Notes”, 1989

[8] P. Antognetti and G. Massobrio, “Semiconductor Device Modeling with SPICE”, 2nd Ed., McGraw-Hill, 1993

Intusoft Newsletter

Personal Computer Circuit & System Design Tools

intusoft

Issue #50 Sept. 1997
Tel. (310) 833-0710
Fax (310) 833-9658

SPARK GAP MODELING

Spark Gap Modeling
Christophe Basso
Excerpts from the September 1997 Intusoft Newsletter #50

Spark gaps or Electrical Surge Arrestors (ESAs) are highly nonlinear devices whose function is to stop transient surges on DC or AC power-supply lines. Such transients can be caused by lightning strikes, motor starts, etc. In other cases, spark gaps can also be used repetitively in ignition-type circuits. A spark gap is made of two electrodes that face each other across a short distance. The gap is fill with air or an inert gas like argon or neon. If the voltage applied to the ESA is below its striking voltage (or avalanche potential), the current flowing through the ESA is close to zero. Once the striking voltage is attained, the voltage across the ESA suddenly collapses to a value called the glow voltage. If the current still increases, the ESA voltage decreases further to a level called the arc voltage, where it stays until the surge passes. At this point, the ESA stays conductive until its current falls below a sustaining value in a manner similar to a thyristor.

Modeling such a component with SPICE can be done in several ways [1, 2]. For the sake of efficiency, we have used a macro-modeling technique. It consists of assembling SPICE primitives in a group to describe a complex electrical function. Figure 8 depicts the general ESA model we have adopted.

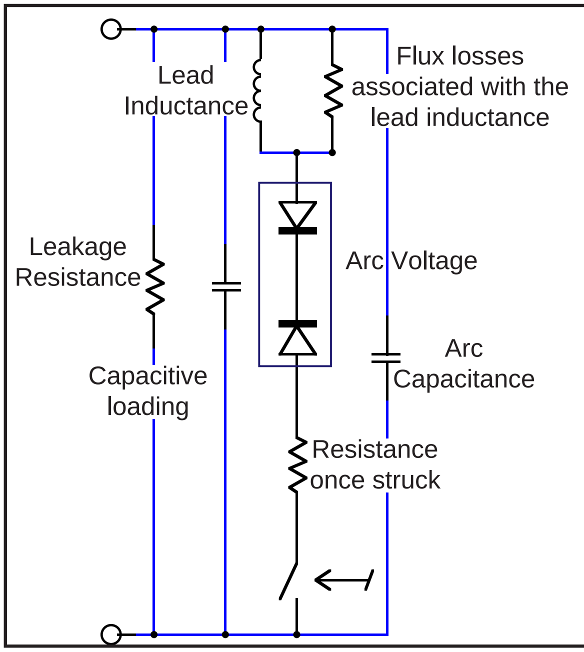


Figure 8, The structure of the spark gap model. The switch is operated using an IsSpice4 If-Then-Else expression;

If V_ESA > Vstrikes Then ON,
If I_ESA > ISUS Then ON,
Else OFF.

A table model is used to add variations with applied dV/dt.

Since the striking is usually brief, it is best to set the .OPTIONS TRTOL variable (overestimation of the transient truncation error) to 1 in order to "capture" the fast switching events.

In the OFF state, the voltage-controlled switch is open and only a leakage current circulates in the ESA. The switch stays OFF until the voltage across the ESA rises up to the striking voltage. At this point, the switch is immediately driven ON and the network made of the back-to-back zener diodes and the series resistance is applied across the ESA terminals. At this point, the voltage collapses to the arc value and the current starts to rise. When the surge passes, the ESA current decays until the sustaining value is reached and the switch opens. In the first model in Table 1, the glow transition is not taken into account, and neither is the dV/dt applied to the ESA. The netlist uses standard SPICE3 elements combined with an IsSpice4 If-Then-Else behavioral element (BARC, in Table 1 below) which performs the arcing action.

In the second model, (Table 2), a behavioral piecewise linear table function is used to account for the dV/dt applied to the device. The PWL table converts the absolute value of the applied slope in coefficients that, when multiplied by VTHRES, will increase the final BARC level. The PWL values in the PWL_001 model are extracted from an ignition voltage versus the applied dV/dt curve. The curve generally appears parabolic when the x axis uses a log scaling. The effect is modeled by the source BARC 15 0 V=ABS(V(1,2)) > {VTHRES} + {VTHRES} * V(33) ?... V(33) increases VTHRES by a ratio defined by the manufacturer’s curves.

```
.SUBCKT SPARK 1 2 {VTHRES=90 VARC=10 ISUS=500M RNEG=-0.5 LPL=130N
+ RPL=2.5K CPAR=1P CARC=3P}
* Subcircuit parameters:
* VTHRES = VOLTAGE AT WHICH THE SPARKGAP STRIKES
* VARC = VOLTAGE ACROSS THE SPARKGAP ONCE STRUCK
* ISUS = CURRENT UNDER WHICH THE ARC IS STOPPED
* RNEG = NEGATIVE RESISTANCE ONCE STRUCK
* LPL = LEAD INDUCTANCE, usually in nanoHenries
* RPL = FLUX LOSS ASSOCIATED WITH LPL, usually in kohms
* CPAR = GAP CAPACITANCE
* CARC = ARC CAPACITANCE
*
VDUM 1 10
LPL 10 11 {LPL}
RPL 10 11 {RPL}
CPAR  1 2 {CPAR}
RLEAK 1 2 10MEG
RNEG 11 17 {RNEG}
DTRK1 12 17 DCLAMP
DSRK2 12 16 DCLAMP
CARC 1 16 {CARC}
X1 16 2 13 SWITCH
BARC 15 0 V=ABS(V(1,2)) > {VTHRES} ? 10V :
+ ABS(I(VDUM)) > {ISUS} ? 10V : 10N
RDEL 15 13 10
CDEL 13 0 10P
.MODEL DCLAMP D BV={VARC}
.ENDS
```

Table 1, The IsSpice4 netlist for a surge arrestor model. The operation does not depend on the applied dv/dt. Since the striking is very fast, it is strongly advised that you set TRTOL variable in the .OPTION statement to 1. (.OPTIONS TRTOL=1). This will force SPICE to be more vigilant in the vicinity of transitions.

The user-defined parameters in the model are described at the top in bold.

```
.SUBCKT A81-A230X 1 2 {VTHRES=230 VARC=10 ISUS=500M LPL=130N RPL=2.5K
+ CPAR=1.4P CARC=3P}
VDUM 1 10
LPL 10 11 {LPL}
RPL 10 11 {RPL}
CPAR  1 2 {CPAR}
RLEAK 1 2 10G
DTRK1 12 11 DCLAMP
DSRK2 12 16 DCLAMP
CARC 1 16 {CARC}
X1 16 2 13 SWITCH
BARC 15 0 V=ABS(V(1,2)) > {VTHRES} + {VTHRES} * V(33) ? 11V :
+ ABS(I(VDUM)) > {ISUS} ? 11V : 10N
RDEL 15 13 10
CDEL 13 0 10P
A1 35 33 PWL_001
RA1 33 0 10MEG
*** INPUT SLOPE CALCULATION V/us ***
BDIFF 40 0 V=V(1,2)
EDIFF 30 0 0 31 100MEG
RDIFF 30 31 1MEG
CDIFF 40 31 1UF
ECONV 32 0 30 0 -1U
BABS 35 0 V=ABS(V(32)) < 10M ? 10M : ABS(V(32)) > 1K ? 1K : ABS(V(32))
.MODEL PWL_001 PWL(XY_ARRAY=[10.0M 1.0M   100.0M 86.0M   1.0 217.0M
+ 10.0 521.0M  100.0 956.0M   1.0K 1.6] INPUT_DOMAIN=10M FRACTION=TRUE)
.MODEL DCLAMP D BV={VARC}
.ENDS
```

Table 2, The IsSpice4 netlist for a spark gap model with the dV/dt variation effect added. The parameters shown are for the SIEMENS A81-A230X ESA.

Obtaining the PWL points is easy. For the Siemens A81-A230x, the threshold voltage of the spark gap is 230V. The curve is flat up to 10kV/s or 10mV/us. For shallow slopes, up to 10mV/us, the first coefficient will be very low, between 1mV and zero. The third point on the curve is for 10E6V/s or 1V/us. At this point, the threshold voltage is 280V. The coefficient is simply: ABS(230-280)/230=0.217, or in the PWL table format: 1.0, 217M. For 10E7V/s or 10V/us, the curve gives a threshold of 350V: ABS(230-350/230)=521M —> 10, 521M and so on. To adapt the model to a particular spark gap device, you only need to enter a few parameters (variables in curly braces) found in the manufacturer's technical specifications.

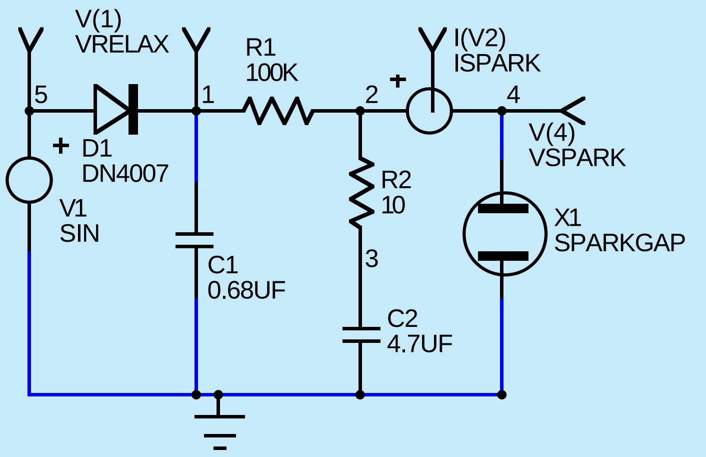


Figure 9, The self-relaxing circuit configuration used to investigate the transient behavior of the new spark gap model. Generic models and models for specific part numbers are available from Intusoft.

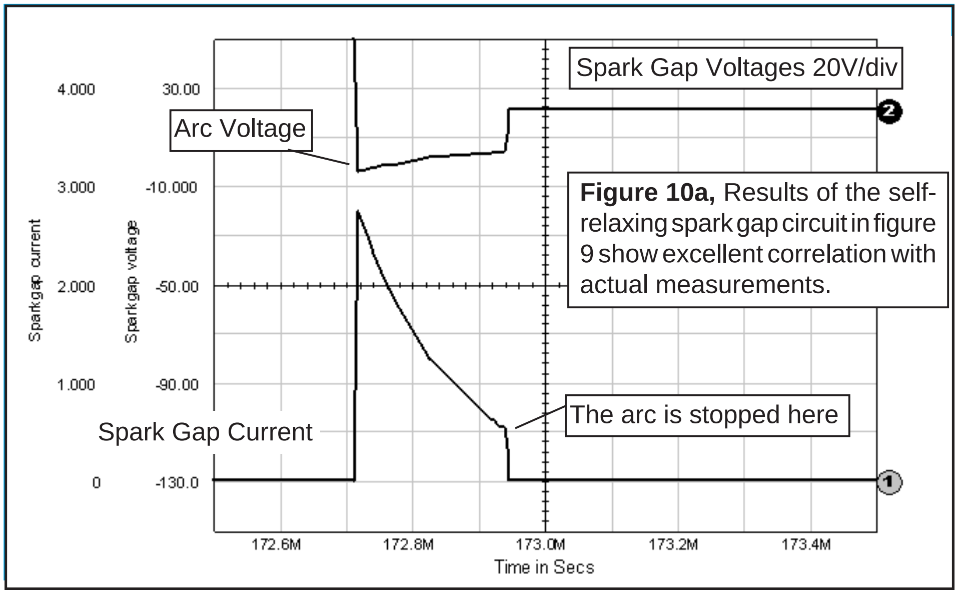


Figure 10a, Results of the self-relaxing spark gap circuit in figure 9 show excellent correlation with actual measurements.

The first IsSpice4 test is made using a self-relaxing configuration as depicted by Figure 9. Since the phenomena are very fast, you will need to view the raw non-interpolated SPICE data (internal calculated data points) simulated by IsSpice4 and not the interpolated (.PRINT) data specified by the TSTEP parameter in the .TRAN statement. Thanks to IntuScope, ICAP's waveform analysis tool, you can easily explore both types of data. Figure 10a depicts the results given by IsSpice4. Figure 10b shows an oscilloscope hardcopy of the actual tested circuit.

A second test is run using the spark gap as a real surge arrestor. The power mains supply a device protected by an

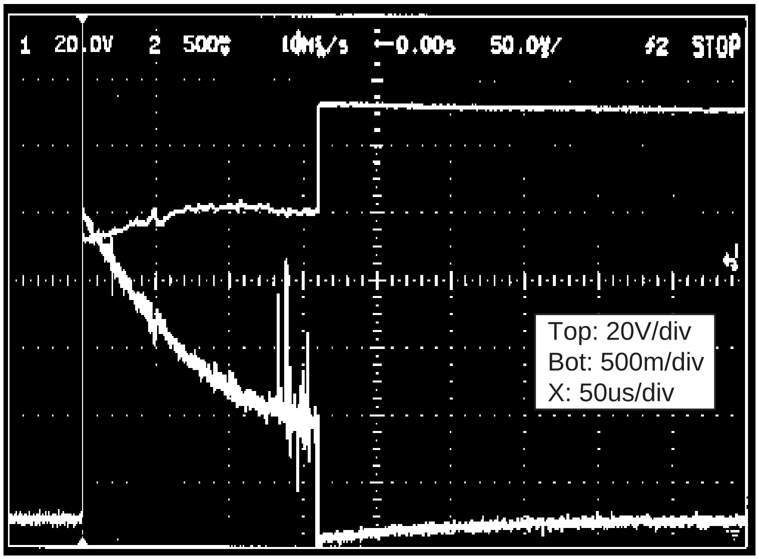


Figure 10b, The laboratory results for the self-relaxing circuit in figure 9 using the Siemens A81-C90x ESA.

ESA. A 1µs transient (PWL source: **PWL 0 0 13ms 0 13.001m 600 13.3ms 600 13.301m 0**) has been added to the sinusoidal (**SIN 0 320 50**) supply voltage in order to trigger the ESA. The results simulated by IsSpice4 are shown on Figure 11.

The model presented here runs fast and converges without difficulties. Although the model accounts for several nonlinear effects, it is still a simplification of the complex phenomenon associated with spark gap ignition and arcing.

Note that a warm or cold cathode fluorescent (CCFL/HCFL) lamp could be easily derived from this model. In the next *Intusoft Newsletter* we will explore such a model and give examples.

A pre-made set of surge arrestor models for various manufacturer's devices is available as part of the new Mechatronics model library. The price of the library is $1500 and will be available November 3, 1997. More details on the Mechatronics library will be available in the next newsletter.

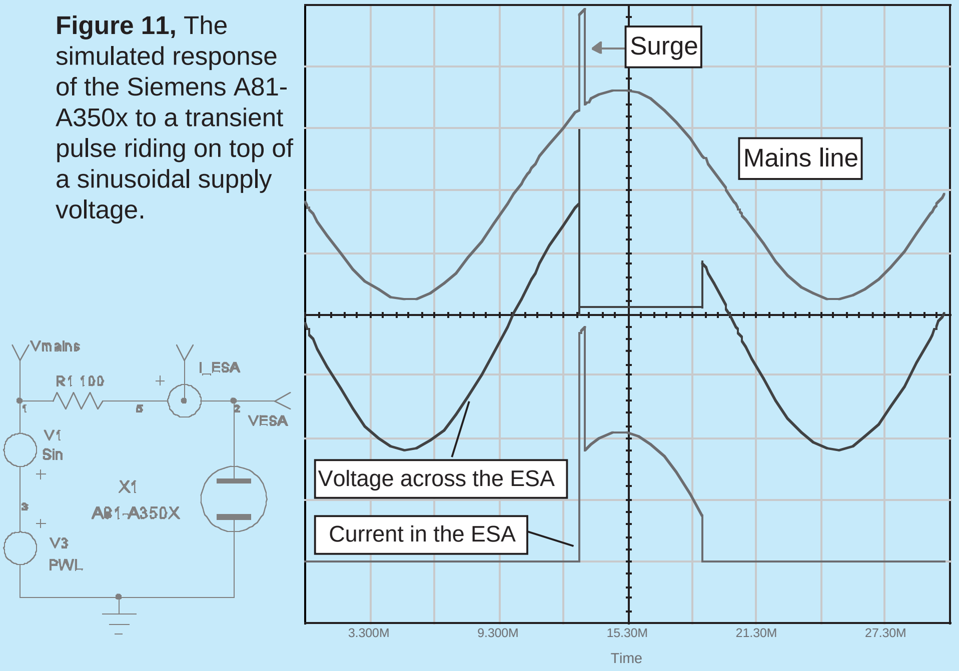


Figure 11, The simulated response of the Siemens A81-A350x to a transient pulse riding on top of a sinusoidal supply voltage.

References

[1] An Electrical Surge Arrestor (ESA) Model For Electromagnetic Pulse Analysis, Rockwell International Electronics Operations, IEEE Transactions on Nuclear Science, Vol. NS-24, No 6, December 1977

[2] A SPICE model for simulating Arc Discharge load, M. NARUI, 1991 IEEE Industry Applications Society Annual Meeting, Volume II.

Intusoft Newsletter

Personal Computer Circuit & System Design Tools

intusoft

Issue #51 Nov. 1997
Tel. (310) 833-0710
Fax (310) 833-9658

SPICE SIMULATES A FLUORESCENT LAMP

SPICE Simulates A Fluorescent Lamp

Christophe Basso
Excerpts from the November 1997 Intusoft Newsletter #51

A Hot Cathode Fluorescent Lamp (HCFL) is a device in which a gaseous mixture flows between two tungsten electrodes or filaments. In domestic applications, the mixture is made of mercury vapor and a small quantity of inert gas (krypton or argon). The role of the inert gas is to vaporize the mercury during the turn-on phase. To lengthen the filament lifetime, a preheating period is necessary in order to bring the electrodes to a sufficient temperature before the tube avalanches to its on state. The warm-up is performed by supplying the filaments with a DC or AC current during the first few hundred milliseconds. During this emissive period, the filaments increase the electron population in the tube, and consequently decrease the avalanche potential resulting in a lower striking voltage for the lamp. Once the lamp is struck, it maintains a quasi-constant voltage across its end points. This value is called the arc voltage. A practical value for the cold striking voltage for a 5 foot lamp (58W) is near the kV range with the corresponding arc voltage around 110Vrms. An HCFL can be operated at low or high frequencies. At low frequency, e.g. in a 60 or 50Hz ballast application, the conducting gas reacts faster than the AC line. Every time the polarity of the mains changes, the lamp current cancels and the tube halts its conduction process. It then has to restrike with the opposite polarity, but at a voltage lower than its cold value because of the temperature. At this slow operating rate, a second effect can be noticed; due to the negative impedance characteristics of the conducting gas, the voltage across the tube will decrease as the current grows. At higher frequencies, above a few kHz, these effects are smoothed out and we can represent the tube using resistive and weakly capacitive behavior.

The SPICE modeling for a fluorescent tube can be generated in several ways [1]. Fitting the I/V characteristics with a polynomial equation represents an elegant solution, but the parameter extraction from the manufacturer's data curves is invariably a complicated process. In contrast, the SPICE macro-modeling technique offers a simplified and efficient method for model generation. This technique consists of assembling SPICE primitives to describe a complex electrical function. Figure 7 depicts the general model we have adopted.

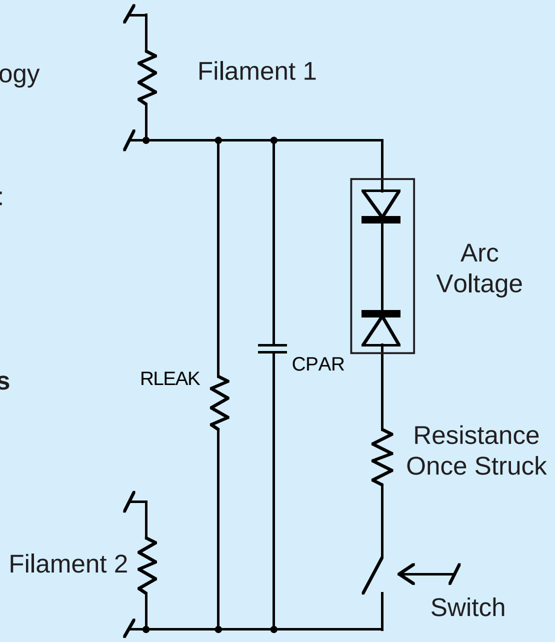


Figure 7 - The subcircuit topology for the fluorescent tube.

The switch is controlled by the following If-Then-Else function:

```
If V_Tube > V_Strike Then
    If I_Tub > I_Sus Then
    On
Else Off.
```

An example tube model is available from the Intusoft Web site at
www.intusoft.com

The model works as follows: if the voltage applied to the tube is lower than its cold striking value, no current will circulate except in the leakage elements. If the voltage is further increased and the striking voltage is reached, the voltage-controlled switch closes and the back-to-back zener network is connected across the tube. The voltage then collapses to the arc value and a current flows inside the tube. The tube will stay conductive until the current falls below the sustaining value. At this point, the switch opens and the tube needs to be restruck. The netlist is written for the IsSpice4 simulator and uses standard SPICE 3 elements combined with one of the IsSpice4's SPICE extensions, an If-Then-Else behavioral element.

As previously stated, preheating the filaments decreases the striking voltage. The model accounts for this specific behavior and monitors the RMS current flowing through the filaments prior to the first cold strike. To take advantage of this feature, you should include the UIC (Use Initial Conditions) keyword in the transient SPICE statement (.TRAN *tstep tstop* **UIC**). In AC applications, where the frequency of operation is fast enough, the thermal effects ensure a restrike voltage close to the arc value. This is especially true for high-frequency systems, e.g. electronic ballasts. The BDIFF element (see the netlist posted on the Intusoft web site) models this effect in a simplistic way.

Figure 8 shows the I/V characteristics of the IsSpice4 fluorescent tube model, where the negative effects are clearly depicted. A low frequency AC test using the tube model in a classical 50Hz application is shown in Figure 9. Figure 10

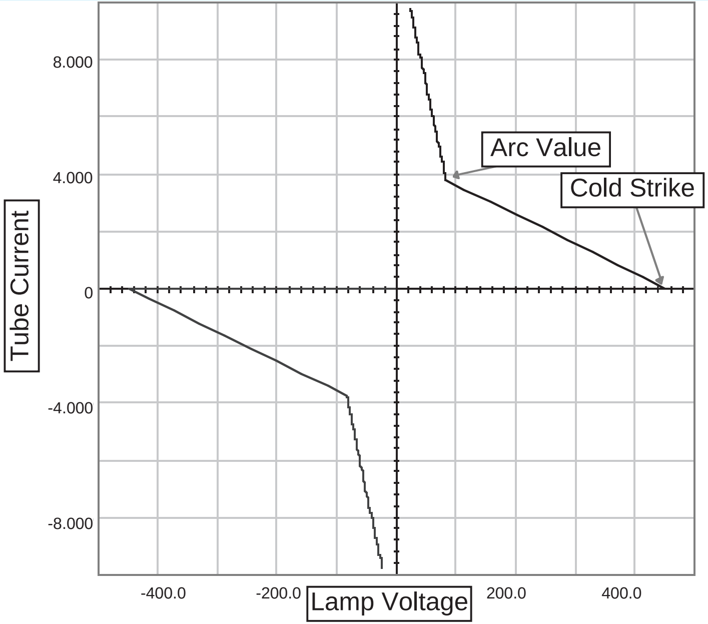


Figure 8 - The DC I/V characteristics for the fluorescent tube model.

reveals the simulation results. To operate the tube model at higher frequencies, you can simply remove the "*" comment symbol in front of the RSTK element in the netlist and place one in front of RNEG, DSTK1 and DSTK2 elements.

A typical application for a self-oscillating ballast is shown in Figure 11. RSTK can be calculated from the RMS lamp operating parameters: RSTK = VARC/INOM. For a 36W tube, RSTK = 103/.43 = 240W.

In this second example, two MOSFETs are driven by a saturable current transformer, using the IsSpice4's nonlinear magnetic core model. The start-up is driven by R17&C12 which forces X21 to enter the conduction mode a few milliseconds after the mains are applied. Square waves are delivered to the non-damped L1-C13 network and a high voltage appears across the tube ends. C13 also ensures preheating of the filaments. Once the tube is struck, the resulting load changes

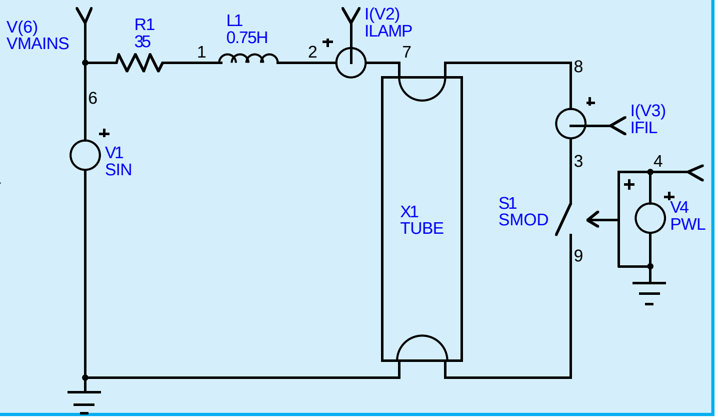


Figure 9 - Tube simulation in a classical 50Hz application. V1 is a sinusoidal waveform. A piecewise linear waveform is used to control the switch.

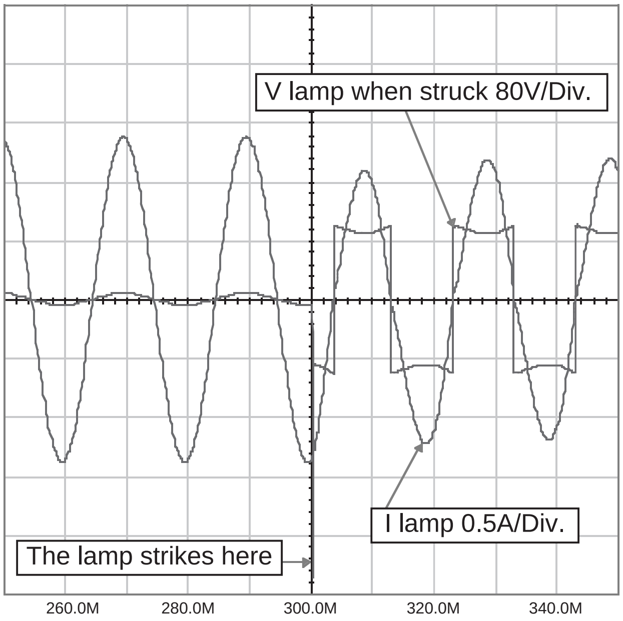


Figure 10 - The simulation results for a low frequency AC test using the tube model in a classical 50Hz application. The lamp is struck when the PWL source causes the switch to close.

the operating frequency of the ballast. Figure 11 shows the IsSpice4 simulation results cross-probed on the schematic.

The model presented here runs fast and converges without difficulties in low and high frequency applications. Thanks to its macro-modeling structure, operating parameters can be easily adapted to various lamp types. Reference 2 gives further insight into the self-oscillating ballast technique.

References

1. The SPICE book, Andrei VLADIMIRESCU, John WILEY & Sons, 0-471-60926-9
2. Energy Efficient Semiconductors for Lighting, MOTOROLA, Application note BR480/D

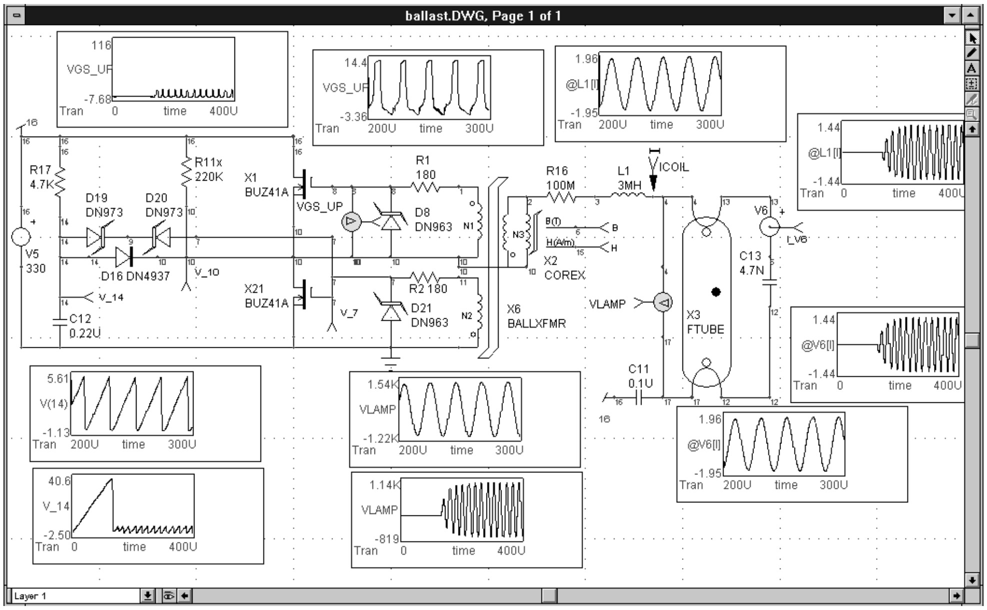


Figure 11 - The transient response of the self-oscillating ballast circuit. Two views of the results can be seen; 0-400us and 200us-300us.

SIDACTOR MODELING

Sidactor Modeling
Slim Petrie
Excerpts from the November 1997 Intusoft Newsletter #51

A SIDACTOR is a bilateral switch, similar to a triac, without a gate connection. It can be triggered into conduction regardless of polarity, but only by an overvoltage pulse. Sidactors are often used as overvoltage protection devices with clamping voltages from 20 to over 500 volts.

Upon application of a voltage exceeding the breakdown voltage, the sidactor switches on through a negative or positive resistance region to a low on-state voltage. Conduction will continue until the current is interrupted or drops below the minimum holding current. The sidactor can offer longer life and faster response (nanoseconds) than other types of protection and is able to respond without voltage overshoot. The sidactor is as fast as a zener diode, while offering a much lower impedance (leakage current <5ua) during conduction.

The latest version of the powerful SpiceMod (v2.4.3) modeling program allows you to create SPICE models for sidactors from data sheet parameters. The program takes the data sheet parameters and converts them to the appropriate SPICE model parameters. The subcircuit topology and an example netlist are shown in Figure 17. The netlist uses a SPICE 2G.6 format and is compatible with all commercial SPICE-based simulators.

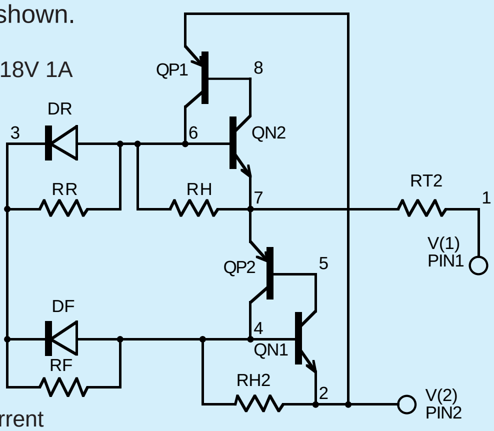
The sidactor is modeled by using two NPN/PNP transistor pairs. The base of each transistor is connected to the collector of the other. This produces positive feedback, resulting in the required switching action. Resistors and zener diodes are used to simulate the breakdown voltages and leakage currents.

Because these devices have two stable states, you may need to tell SPICE which state to use. This is accomplished by including the “OFF” parameter on the subcircuit transistors. This will set up an initial starting condition in the normal starting state.

The chief causes of SPICE convergence problems are due to abrupt changes in a circuit’s impedance. To aid IsSpice4 in converging during the abrupt sidactor switching, the following OPTIONS statement is recommended “.OPTIONS ITL1=400 ITL4=500 RELTOL=.005.

It should be noted that these are the options chosen by the Convergence Wizard feature, included in ICAP/4 version 8, when a mild convergence problem is encountered.

Figure 17, The IsSpice4 netlist for a sidactor. The parameters for a Teccor K1100E70 118Volt 1Amp device are shown.



```
*K1100E70;K1100E70;Sidacs;TECCOR; 118V 1A
.SUBCKT K1100E70  1  2
*     TERMINALS: MT2 MT1
QN1  5 4 2  NOUT; OFF
QN2  8 6 7  NOUT; OFF
QP1  6 8 2  POUT; OFF
QP2  4 5 7  POUT; OFF
DF  4 3  DZ; OFF, Forward breakdown
DR  6 3  DZ; OFF, Reverse breakdown
RF  4 3  1.18E+10 ; controls IBO
RR  6 3  1.18E+10
RT2 1 7  0.755 ; controls "on" resistance
RH  7 6  11.5 ; controls holding current
RH2 4 2  11.5 ; controls reverse holding current
.MODEL DZ   D (IS=321F RS=100 N=1.5 IBV=10N BV=118) ; controls VDRM
.MODEL POUT PNP (IS=321F BF=10 CJE=134N TF=25.5U)
.MODEL NOUT NPN (IS=321F BF=20 CJE=134N CJC=26.8N TF=1.7U)
.ENDS
```

The following NPN/PNP parameters control the performance of the sidactor at high speeds (high dv/dt):

> CJE = Controls the high speed triggering, CJC = (with RS + RGP) controls the maximum forward voltage application rate (dv/dt) before triggering occurs, and TF = Ideal Forward Transit Time (not fall time). This determines the turn-on and turn-off time

Figure 18 shows a sample application where a sidactor is added to protect a transistor during inductive load switching. Several sidactor models are included on the newsletter floppy disk and are posted on Teccor's web site at www.teccor.com.

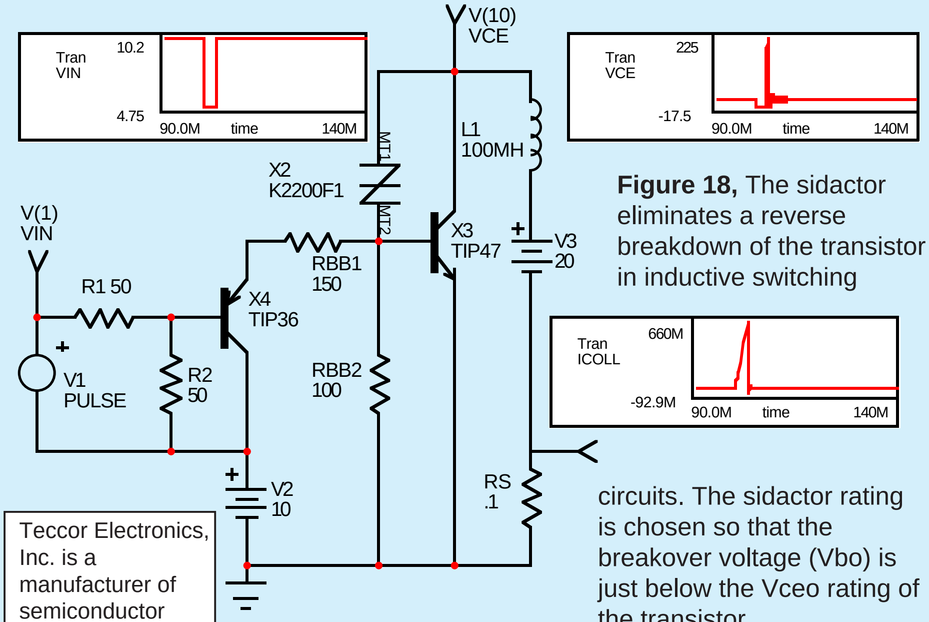


Figure 18, The sidactor eliminates a reverse breakdown of the transistor in inductive switching circuits. The sidactor rating is chosen so that the breakover voltage (Vbo) is just below the Vceo rating of the transistor.

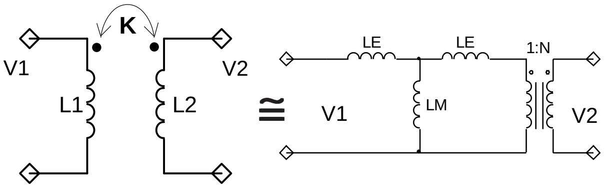
Teccor Electronics, Inc. is a manufacturer of semiconductor devices.

Transformer and Saturable Core Modeling

Larry Meares and Charles Hymowitz, Intusoft

Transformer Models

The usual method of simulating a transformer using IsSpice is by specifying the open circuit inductance seen at each winding and then adding the coupling coefficients to a pair of coupled inductors. This technique tends to loose the physical meaning associated with leakage and magnetizing inductance and does not allow the insertion of a nonlinear core. It does, however, provide a transformer that is simple to create and simulates efficiently. The coupled inductor type of transformer, its related equations and relationship to an ideal transformer with added leakage and magnetizing inductance is shown in Figure 1.



$$V1 = L1 \frac{di1}{dt} + M \frac{di2}{dt}$$

$$V2 = M \frac{di1}{dt} + L2 \frac{di2}{dt}$$

$$\text{If } K \cong 1$$

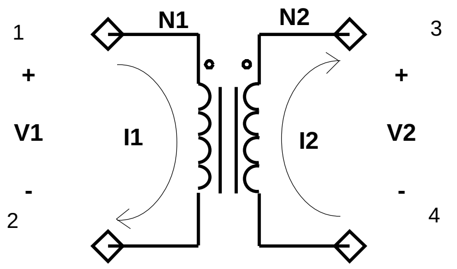
$$L1 = LM$$

$$L2 = N^2 * L1$$

$$K = 1 - LE/LM$$

Figure 1, The coupled inductor transformer, left, is computationally efficient, but it cannot provide access to LE and LM or be used as a building block. The ideal transformer with discrete inductances and their relationship to the coupling coefficient is shown on the right.

In order to make a transformer model that more closely represents the physical processes, it is necessary to construct an ideal transformer and model the magnetizing and leakage inductances separately. The ideal transformer is one that preserves the voltage and current relationships, shown in Figure 2, and has a unity coupling coefficient and infinite magnetizing inductance. The ideal transformer, unlike a real transformer, will operate at DC. This property is useful for modeling the operation of DC-DC converters.



$$V2 = V1 * N2 / N1$$

$$I1 = I2 * N2 / N1$$

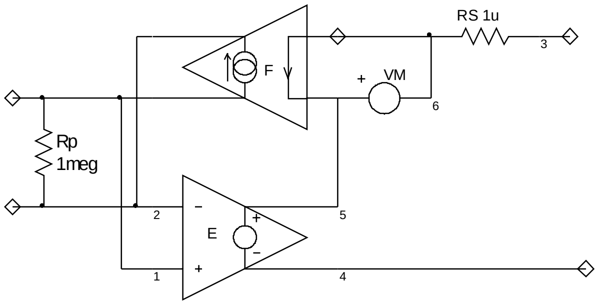
Figure 2, Symbol of an ideal transformer with the voltage to current relationships.

The coupling coefficient of a transformer wound on a magnetic core is nearly unity when the core is not saturated and depends on the winding topology when the core is saturated. The work of Hsu, Middlebrook and Cuk [3] develops the relationship of leakage inductance, showing that relatively simple measurements of input inductance with shorted outputs yield the necessary model information.

The IsSpice equivalent circuit for an ideal transformer is shown in Figure 3 and implements the following equations:

$$V1 * RATIO = V2 \qquad I1 = I2 * RATIO$$

RP and RS are used to prevent singularities in applications where terminals 1 and 2 are open circuited or terminals 3 and 4 are connected to a voltage source. RATIO is the turns ratio from winding 3,4 (secondary) to winding 1,2 (primary). Polarity "dots" are on terminals 1 and 3 as shown in Figure 2.



```
.SUBCKT XFMR 1 2 3 4
E  5  4  1  2  {RATIO}
F  1  2  VM  {RATIO}
VM  5  6
RP  1  2  1MEG
RS  6  3  1U
.ENDS
```

Figure 3, The IsSpice ideal transformer model allows operation at DC and the addition of magnetizing and leakage inductances, as well as a saturable core to make a complete transformer model. Parameter passing allows the transformer to simulate any turns ratio.

IsSpice Subcircuit Call

```
X1  1  2  3  4  5   XFMR {RATIO=xxx  }
```

IsSpice Subcircuit Definition

```
.SUBCKT  XFMR-TAP  1 2 3 4 5
E1   7 8 1 2 {RATIO}
F1   1 2 VM1 {RATIO}
RP  1 2 1MEG
RS  6 3 1U
VM1  7 6
F2   1 2 VM2 {RATIO}
E2   9 5 1 2 {RATIO}
R5  8 4 1U
VM2  9 8
.ENDS
```

IsSpice Netlist

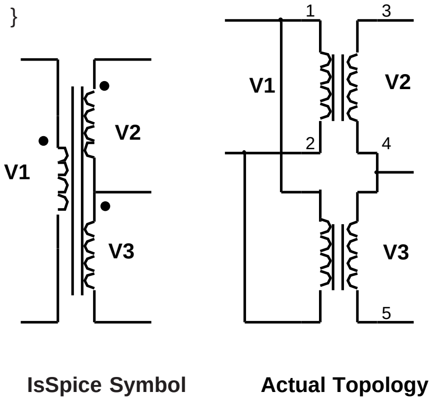


Figure 4, Multiple winding transformers may be built out of combinations of the ideal transformer. RATIO equals V2/V1. The transformer is center tapped and V2 will equal V3.

Multi-winding topologies can also be simulated by using combinations of this 2 port representation.

Now that the ideal transformer has been constructed, magnetizing inductance can be added using a separate saturable core model described next.

Saturable Reactor Model

A saturable reactor is a magnetic circuit element consisting of a single coil wound around a magnetic core. The presence of a magnetic core drastically alters the behavior of the coil by increasing the magnetic flux and confining most of the flux to the core. The magnetic flux density, B, is a function of the applied MMF, which is proportional to ampere turns. The core consists of a number of tiny magnetic domains made up of magnetic dipoles. These domains set up a magnetic flux that adds to or subtracts from the flux set up by the magnetizing current. After overcoming initial friction, the domains rotate like small DC motors, to become aligned with the applied field. As the MMF is increased, the domains rotate one by one until they are all in alignment and the core saturates. Eddy currents are induced as the flux changes, causing added loss.

The saturable reactor cannot be modeled using a single SPICE primitive element. Therefore, a saturable core "macro model", utilizing the IsSpice subcircuit feature, must be created. The saturable core model is capable of simulating nonlinear transformer behavior including saturation, hysteresis, and eddy current losses. To make the model even more useful it has been parameterized. This is a technique which allows the characteristics of the core to be determined just by the specification of a few key parameters. At the time of the simulation, the specified parameters are passed into the subcircuit. The equations in the subcircuit (inside the curly braces) are then evaluated and replaced with a value making the equation based subcircuit compatible with any SPICE program.

The parameters that must be passed to the subcircuit include:

- **Flux Capacity in Volt-Sec (VSEC)**
- **Initial Flux Capacity in Volt-Sec (IVSEC)**
- **Magnetizing Inductance in Henries (LMAG)**
- **Saturation Inductance in Henries (LSAT)**
- **Eddy current critical frequency in HZ (FEDDY).**

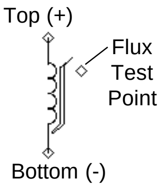
The saturable core may be added to the model of the ideal transformer to create a more complete transformer model. To use the saturable core model just place the core across the transformer's input terminals and evaluate the equations in curly braces. A special subcircuit test point has been provided to allow the monitoring of the core flux. Since there are two connections in the subcircuit, no connection need be made at the top subcircuit level other than the dummy node number.

How The Core Model Works

Modeling the physical process performed by a saturable core is most easily accomplished by developing an analog of the magnetic flux. This is done by integrating

A call to the saturable core subcircuit using IsSpice would look like the following:

X1 2 0 3 CORE {VSEC=50U IVSEC=-25U LMAG=10MHY LSAT=20UHY FEDDY=20KHZ}



```
.SUBCKT CORE 1 2 3
F1 1 2 VM1 1
G2 2 3 1 2 1
E1 4 2 3 2 1
VM1 4 5
RX 3 2 1E12
CB 3 2 {VSEC/500}
+ IC={IVSEC/VSEC*500}
RB 5 2 {LMAG*500/VSEC}
```

```
RS 5 6 {LSAT*500/VSEC}
VP 7 2 250
D1 6 7 DCLAMP
VN 2 8 250
D2 8 6 DCLAMP
.MODEL DCLAMP D(CJO={3*VSEC/
+ (6.28*FEDDY*500*LMAG)} VJ=25)
.ENDS
```

Figure 5, To make the netlist for the saturable core subcircuit SPICE compatible, just replace all the equations in the curly braces with numerical values. The key parameters are shown in bold.

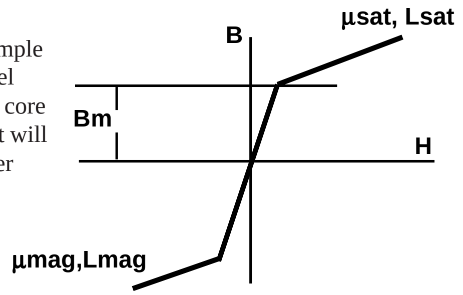


Figure 6, A simple B-H loop model detailing some core parameters that will be used for later calculations.

the voltage across the core and then shaping the flux analog with nonlinear elements to cause a current to flow proportional to the desired function. This gives good results when there is no hysteresis as illustrated in Figure 6.

The input voltage, V(2), is integrated using the voltage controlled current source, G, and the capacitor CB. An initial condition across the capacitor allows the core to have an initial flux. The output current from F is shaped as a function of flux using the voltage sources VN and VP and diodes D1 and D2. The inductance in the high permeability region is proportional to RB, while the inductance in the saturated region is proportional to RS. Voltage VP and VN represent the saturation flux. Core losses can be simulated by adding resistance across the input terminals; however, another equivalent method is to add capacitance across resistor RB in the simulation. Current in this capacitive element is differentiated in the model to produce the effect of resistance at the terminals. The capacitance can be made a nonlinear function of voltage which results in a loss term that is a function of flux. A simple but effective way of adding the nonlinear capacitance is to give the diode parameter, CJO, a value, as is done here. The other option is to use a nonlinear capacitor across nodes 2 and 6, however, the capacitor's polynomial coefficients are a function of saturation flux, causing their recomputation if VP and VN are changed.

Losses will increase linearly with frequency simulating high frequency core behavior. A noticeable increase in MMF occurs when the core comes out of saturation, an effect that is more pronounced for square wave excitation than for sinusoidal excitation as shown in Figure 8. These model properties agree closely with observed behavior [2]. The model is set up for orthonol and steel core materials which have a sharp transition from the saturated to the unsaturated region. For permalloy cores the transition out of saturation is less pronounced. To account for the different response the capacitance value in the diode model (CJO in DCLAMP), which affects core losses, should be scaled down. Also, scaling the voltage sources VN and VP down will soften the transition.

The DC B-H loop hysteresis, usually unnecessary for most applications, is not modeled because of the extra model complexity, causing a prediction of lower loss at low frequencies. The hysteresis, however, does appear as a frequency dependent function, as seen on the previous page, providing reasonable results for most applications, including magnetic amplifiers. The model shown in Figure 7 simulates the core characteristics and takes into account the high frequency losses associated with eddy currents and transient widening of the B-H loop caused by magnetic domain angular momentum. Losses will increase linearly with frequency, simulating high frequency core behavior.

Calculating Core Parameters

The saturable core model is setup to be described in electrical terms, thus allowing the engineer to design the circuitry first without knowledge of the core's physical makeup. After the design is completed, the final electrical parameters can then be used to calculate the necessary core magnetic/size values. The core model could be altered to take as its input magnetic and size parameters. The core could then be described in terms of N, Ac, Ml, µ, and Bm and would be more useful for studying previously designed circuits. But the electrical based model is better suited to

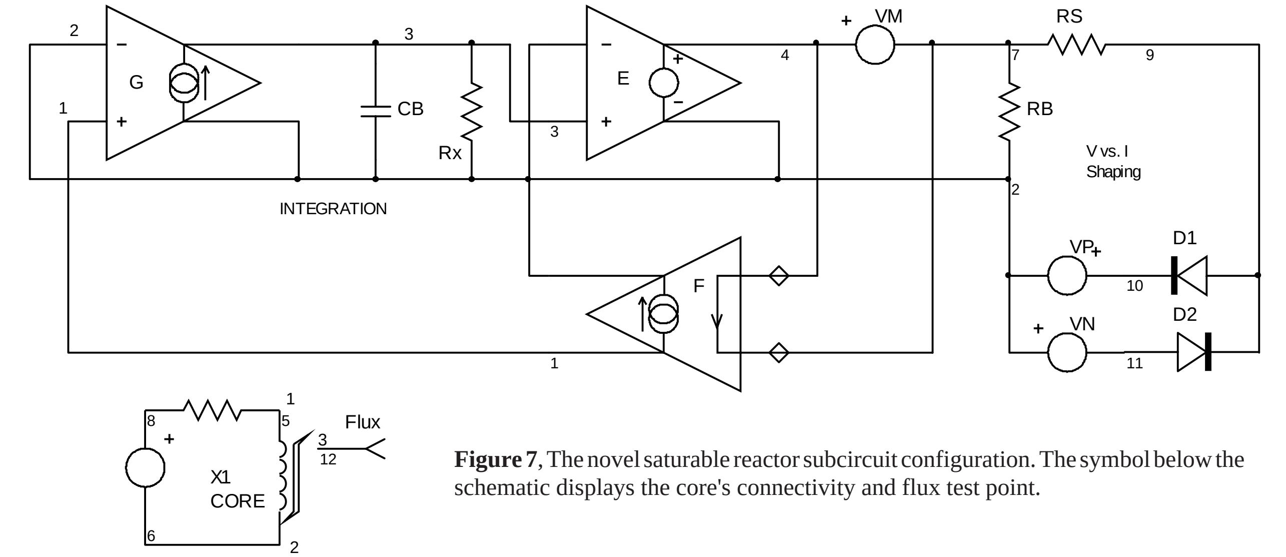


Figure 7, The novel saturable reactor subcircuit configuration. The symbol below the schematic displays the core's connectivity and flux test point.

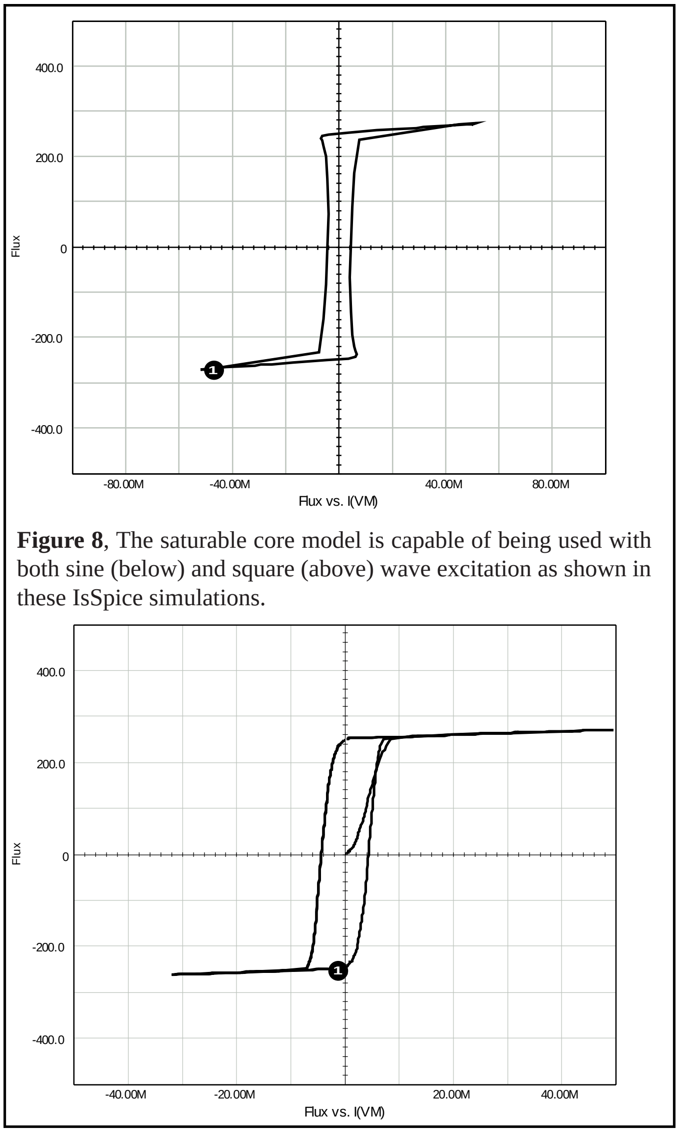


Figure 8, The saturable core model is capable of being used with both sine (below) and square (above) wave excitation as shown in these IsSpice simulations.

the natural design process. The saturable core model's behavior is defined by the set of electrical parameters, shown in Figure 6 and Figure 9. The core's magnetic/size values can be easily calculated from the following equations which utilize cgs units.

Parameters Passed To Model

VSEC	Core Capacity in Volt-Sec
IVSEC	Initial Condition in Volt-Sec
LMAG	Magnetizing Inductance in Henries
LSAT	Saturation Inductance in Henries
FEDDY	Frequency when LMAG Reactance = Loss Resistance in Hz

Equation Variables

Bm	Maximum Flux Density in Gauss
H	Magnetic Field Strength in Oersteds
Ac	Area of the Core in cm^2
N	Number of Turns
Ml	Magnetic Path Length in cm
μ	Permeability

Faraday's law, which defines the relationship between flux and voltage is:

$E = N\, d\varphi/dt * 10^{-8}$ Eq. 1

where E is the desired voltage, N is the number of turns and φ is the flux of the core in maxwells. The total flux may also be written as:

$\varphi_T = 2 \ast Bm \ast Ac$ Eq. 2

Then, from 1& 2,

$E = 4.44 \ast Bm \ast Ac \ast F \ast N \ast 10^{-8}$ Eq. 3

and

$E = 4.0 \ast Bm \ast Ac \ast F \ast N \ast 10^{-8}$ Eq. 4

where Bm is the flux density of the material in Gauss, Ac is the effective core cross sectional area in cm^2, and F is the design frequency. Equation 3 is for sinusoidal conditions while equation 4 is for a square wave input. The parameter VSEC can then be determined by integrating the input voltage resulting in:

$\int e\, dt = N\varphi_T = N \ast 2 \ast Bm \ast Ac \ast 10^{-8} =$ **VSEC** Eq. 5

also from E = L di/dt we have,

$\int e\, dt = Li$ Eq. 6

The initial flux in the core is described by the parameter **IVSEC**. To use the IVSEC option you must put the UIC keyword in the IsSpice ".TRAN" statement. The relationship between the magnetizing force and current is defined by Ampere's law as

$H = .4 \ast \pi \ast N \ast i / Ml$ Eq. 7

where H is the magnetizing force in oersteds, i is the current through N turns, and Ml is the magnetic path length in cm.

From equations 5, 6, and 7 we have

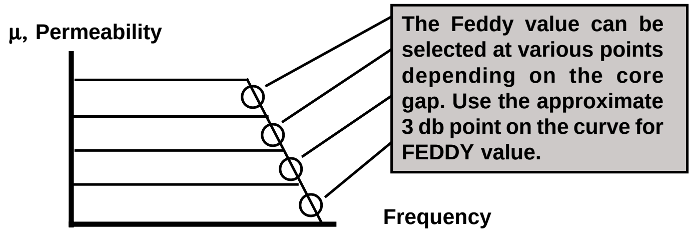
$L = N^2 \ast Bm \ast Ac \ast (.4 \ast \pi \ast 10^{-8}) / H \ast MI$ Eq. 8

with $\mu = B/H$ we have

L(mag, sat) =

$\mu(mag, sat) \ast N^2 \ast\ .4 \ast\ \pi \ast\ 10^{-8} \ast\ Ac / Ml$ Eq. 9

The values for LMAG and LSAT can be determined by using the proper value of μ in Eq. 9. The values of permeability can be found by looking at the B - H curve and choosing two values for the magnetic flux, one value in the linear region where the permeability will be maximum and one value in the saturated region. Then,



The Feddy value can be selected at various points depending on the core gap. Use the approximate 3 db point on the curve for FEDDY value.

Figure 9, The permeability versus frequency graph is used to determine the value for FEDDY.

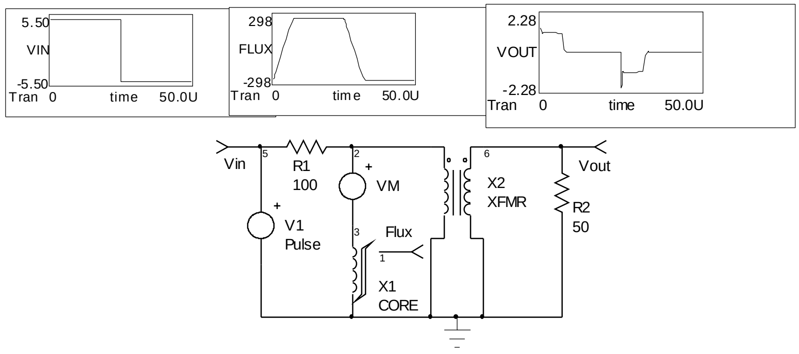


Figure 10, Saturable core test circuit schematic and IsSpice simulation results.

from a curve of permeability versus magnetic flux, the proper values of μ may be chosen. The value of μ in the saturated region will have to be an average value over the range of interest. The value of FEDDY, the eddy current critical frequency, can be determined from a graph of permeability versus frequency, shown in Figure 4. By choosing the approximate 3 db point for μ, the corresponding frequency can be determined.

It should be noted that a similar core model can be constructed using generic physical parameters as opposed to generic electrical design parameters. For example:

```
.SUBCKT COREX 1 2 3 {BI=0 N=1}
* Core Defaults shown in { }
* Defaults for parameters that are unlikely to change over
*the core family may be placed on the .SUBCKT line.
RX 3 2 1E12
CB 3 2 {N*2*BR*ACORE*1E-8/500} IC={BI/BR*500}
F1 1 2 VM1 1
G2 2 3 1 2 1
E1 4 2 3 2 1
VM1 4 5
RB 5 2 {.625*N*UMAG/(LPATH * BR)*500}
RS 5 6 {.625*N*USAT/(LPATH * BR)*500}
VP 7 2 250
D1 6 7 DCLAMP
VN 2 8 250
D2 8 6 DCLAMP
* MULTIPLIER 3 AND VJ=25 GO TOGETHER
.MODEL DCLAMP D(CJO={3*LPATH *
+ BR/(6.28*FEDDY*500*.625*N*UMAG)} VJ=25)
.ENDS
```

where the passed physical parameters are

ACORE	Magnetic cross section area in cm^2
LPATH	Magnetic path length in cm
FEDDY	Frequency when Lmag Reactance=Loss resistance
UMAX	Maximum Permeability, dB/dH
USAT	Saturation Permeability, dB/dH
BR	Flux density in gauss at H = 0 for saturated B-H loop
BI	Initial Flux density, default = 0
N	Turns, default one for use with TURNS model

An example set of parameters for Permalloy Tape Wound 80 would be BR=6K FEDDY=600K UMAX=180K USAT=10K BI=0 N=1 where ACORE and LPATH would depend on the geometry chosen.

Using And Testing The Saturable Core

The test circuit shown in Figure 10 can be used to evaluate the saturable core model. Pass the core parameters in the curly braces into the saturable core subcircuit and adjust the voltage levels in the "V2 4 0 PULSE" or "V2 4 0 SIN" statements to insure that the core will saturate. You can use Eq. 3 and 4 to get an idea of the voltage levels necessary to saturate the core. The .TRAN statement may also need adjustment depending on the frequency specified by the V2 source. The core parameters must remain reasonable or the simulation may fail. After the simulation is complete, plotting V(5) versus I(VM1) (Flux vs. Current through the core) will result in a B-H plot.

A Complete Transformer Model

A transformer is a subassembly; made up of a number of components. The actual hardware consists of a core with a number of turns, various encapsulates, insulators, fasteners and connection points. Electrically, the components are the

```
Test Circuit
*SPICE_NET
.TRAN .1US 50US
*INCLUDE DEVICE.LIB
*ALIAS  V(3)=VOUT
*ALIAS  V(5)=FLUX
*ALIAS  V(4)=VIN
.PRINT TRAN  V(3)  V(5)  I(VM1) V(4)
R1 4 2 100
R2 3 0 50
X1 1 0 5 CORE {VSEC=25U IVSEC=-25U
+ LMAG=10MHY LSAT=20UHY FEDDY=25KHZ}
X2 2 0 3 0 XFMR  {RATIO=.3}
VM1 2 1
V2 4 0 PULSE -5 5 0US 0NS 0NS
+25US
*Use the Pulse statement for square wave excitation
*V2 4 0 SIN 0 5 40K
*Use the Sin statement for sine wave excitation
*Adjust voltage levels to insure core saturation
.END
```

Figure 11, Equivalent IsSpice netlist for the saturable core test circuit.

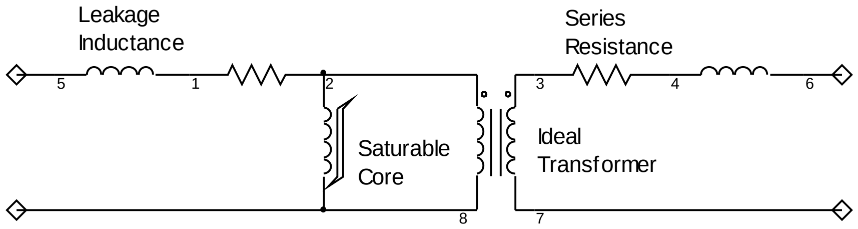


Figure 12, The saturable core may be combined with the ideal transformer, XFMR, and some leakage inductance and series resistance to create a transformer representation.

core and each of the windings. The windings have special geometrical properties that result in series resistance, inter- and intra-winding capacitance and leakage inductance. The series resistance, leakage inductance and capacitance are generally ball-park estimates with the actual values established by prototype measurements or by software such as the Magnetics Designer tool.

The magnetizing inductance is added by using the saturable reactor model across any one of the windings of the ideal transformer. Coupling coefficients are inserted in the model by adding the series leakage inductance for each winding as shown in Figure 12.

The leakage inductances are measured by finding the short circuit input inductance at each winding and then solving for the individual inductance. These leakage inductances are independent of the core characteristic shown by ref [3]. The final model, incorporating the CORE and XFMR subcircuits along with the leakage inductance and winding resistance is shown in Figure 12.

A more complete transformer representation, including geometrically defined parasitics, is shown in Figure 13.

SPICE models cannot represent all possible behavior because of the limits of computer memory and run time. This model, as most simulations, does not represent all cases.

Modeling the core (in Figure 12) as a single element referred to one of the windings works in most cases; however, some applications may experience saturation in a small region of the core, causing some windings to be decoupled faster than others, invalidating the model. Another limitation of this model is for topologies with magnetic shunts or multiple cores. Applications like this can frequently be solved by replacing the single magnetic structure with an equivalent structure using several transformers, each using the model presented here.

References

[1] SPICE2, A computer Program to Simulate Semiconductor Circuits
Laurence W. Nagel, Memorandum No. ERL-M520, 9 May 1975, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA 94720

[2] Design Manual featuring tape wound cores,
Magnetics, Inc., Components Div., Box 391, Butler PA 16001

[3] Transformer Modeling and Design for Leakage Control
Shi Ping Hsu, R.D. Middlebrook and Slobodan Cuk, Power conversion International, pg. 68, Feb, 1982

[4] Advances in Switched-Mode Power Conversion
Slobodan Cuk and R.D. Middlebrook, vol. 2, copyright 1981

[5] NEW SIMULATION TECHNIQUES USING SPICE
L.G. Meares, Applied Power Electronics Conference, (c) IEEE, April-May, 1986.

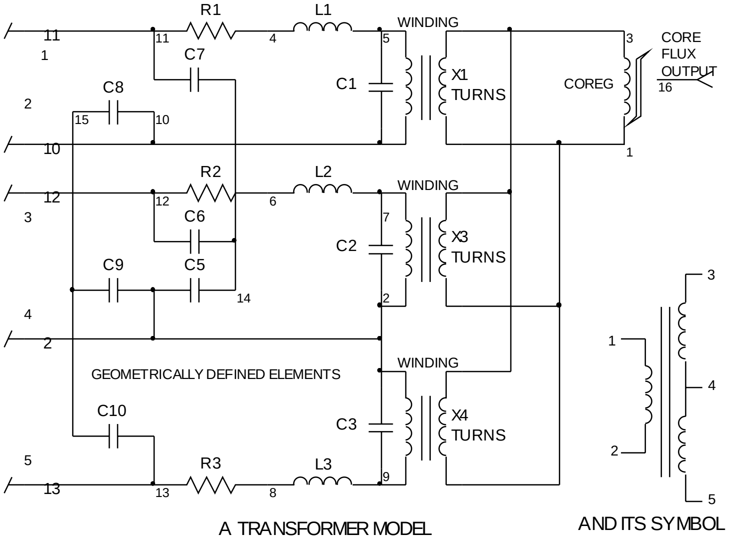


Figure 13, A more complete transformer model showing geometrically defined parasitics and an equivalent schematic symbol.

Intusoft Newsletter

Personal Computer Circuit Design Tools

#42 June 1995 Issue

intusoft

Copyright © Intusoft, All Rights Reserved

(310) 833-0710 Fax (310) 833-9658

Modeling Magnetics

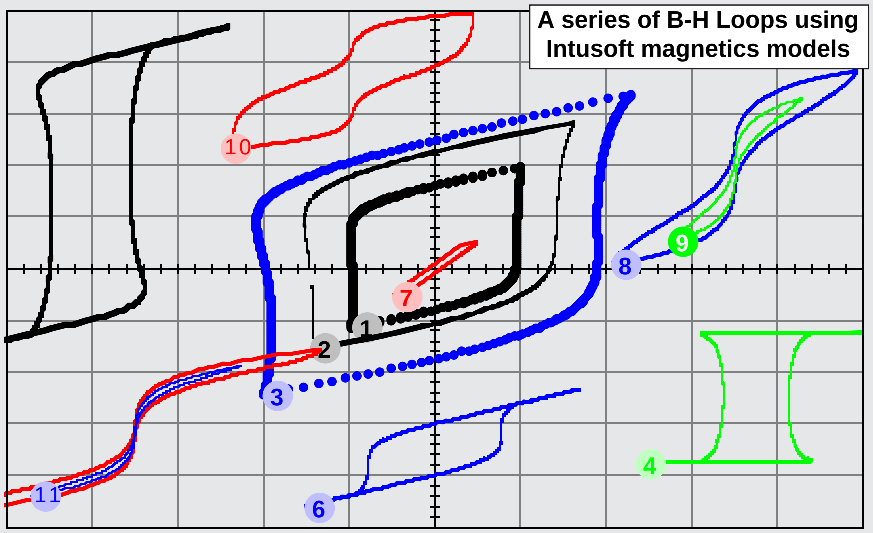


Figure 1, Nonlinear magnetic elements can be simulated in a number of ways in IsSpice. See page 2 for more details.

Modeling Nonlinear Magnetics

Larry Meares, Charles Hymowitz, Steve Sandler

Excerpts from the June 1995 Intusoft Newsletter #42

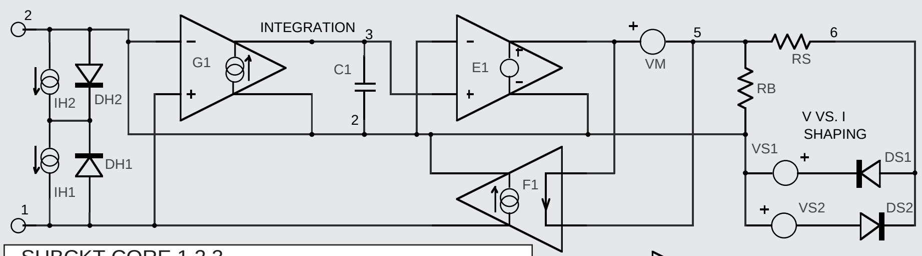
Modeling Nonlinear Magnetics

Many models for magnetic cores exist [1]. Some have been added to SPICE as built-in models while others use a subcircuit approach. In terms of modeling, however, its important to understand what SPICE behavioral modeling can offer vs. direct coding methods in order to determine at what point an AHDL implementation is appropriate. Behavioral constructs in SPICE today are plentiful and powerful. As an example, let's take a look at two subcircuit based magnetic core models that use SPICE 2G and SPICE 3 behavioral modeling.

A saturable reactor is a magnetic circuit element consisting of a single coil wound around a magnetic core. The presence of a magnetic core drastically alters the behavior of the coil by increasing the magnetic flux and confining most of the flux to the core. The magnetic flux density, B, is a function of the applied MMF, which is proportional to ampere turns. The core consists of a number of tiny magnetic domains made up of magnetic dipoles. These domains set up a magnetic flux that adds to or subtracts from the flux set up by the magnetizing current. After overcoming initial friction, the domains rotate like small DC motors, to become aligned with the applied field. As the MMF is increased, the domains rotate one by one until they are all in alignment and the core saturates. Eddy currents are induced as the flux changes, causing added loss.

Modeling the physical process performed by a saturable core is most easily accomplished by developing an electrical analog of the magnetic flux. This is done by integrating the voltage across the core and then shaping the flux analog with nonlinear elements to cause a current to flow proportional to the desired function. Intusoft has created a SPICE 2 compatible subcircuit model that accurately simulates nonlinear core behavior including saturation, hysteresis, and eddy current losses [2, 3].

The popular Jiles-Atherton magnetics model, used in some SPICE programs, is based on existing ideas of domain wall motion, including flexing and translation [4]. While a good description of core behavior, the Jiles Atherton model is difficult to use unless you have access to parameters like the pinning energy per volume, thermal energy, interdomain coupling, and domain flexing value. Invariably, the model can only be constructed by trial simulations and tweaking of the model parameters; if you know how. In contrast, the Intusoft model is much easier to use and accepts commonly available electrical or physical data sheet parameters.



```
.SUBCKT CORE 1 2 3
DH1 1 9 DHYST
DH2 2 9 DHYST
IH1 9 1 {IHYST}
IH2 9 2 {IHYST}
F1 1 2 VM 1
G1 2 3 1 2 1
E1 4 2 3 2 1
VM 4 5
C1 3 2 {SVSEC/250}  IC={IVSEC/SVSEC*250}
RB 5 2 {LMAG*250/SVSEC}
RS 5 6 {LSAT*250/SVSEC}
VP 7 2 250
D1 6 7 DCLAMP
VN 2 8 250
D2 8 6 DCLAMP
E2 10 0 3 2 {SVSEC/250}
.MODEL DHYST D
.MODEL DCLAMP D(CJO={3*SVSEC/(250*REDDY)}
+ VJ=25)
.ENDS
```

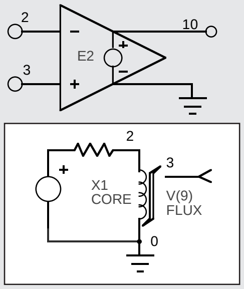


Figure 2, The modified Intusoft saturable reactor model and netlist. The SpiceNet symbol below the schematic reveals the core's connectivity and subcircuit flux test point.

SVSEC Volt-sec at Saturation = $B_{SAT} \bullet A_E \bullet N$
IVSEC Volt-sec Initial Condition = $B \bullet A_E \bullet N$
LMAG Unsaturated Inductance = $\mu_O \, \mu_R \bullet N^2 \bullet A_E / L_M$
LSAT Saturated Inductance = $\mu_O \bullet N^2 \bullet A_E / L_M$
IHYST Magnetizing I @ 0 Flux = $H \bullet L_M / N$
REDDY Eddy Current Loss Resistance

Table 1, Description of the model parameters for the modified Intusoft saturable core subcircuit shown in Figure 1.

Svsec and **Ivsec** are based on peak flux values. Lmag: For an ungapped core, L = L_M, (total path around core), For a gapped core, $\mu_R = 1$, L = gap length. A_E = core area, m^2, **Lsat**: Use core dimensions but with $\mu_R = 1$, **Reddy**: Equals Lmag reactance when permeability vs freq. is 3dB down.

The original Intusoft model had one shortcoming; it did not include low frequency hysteresis. A modified version, shown in Figure 2, solves this with the addition of 4 elements to the input. Magnetizing current associated with low frequency hysteresis is provided by current sinks IH1/IH2. With no voltage across terminals 1 and 2, these currents circulate through their respective diodes, and the net terminal current is zero. When voltage is applied, the appropriate diode starts to block and its current sink becomes active [3]. The model takes into account frequency dependent losses associated with eddy currents and transient widening of the B-H loop (Figure 1, Table 1) caused by magnetic domain angular momentum. It provides excellent results for most applications, including magnetic amplifiers.

Simulating Nonlinear Inductors

The Magnetics molypermalloy powder (MPP) core is widely used in power conversion circuits. These cores are ideal for use

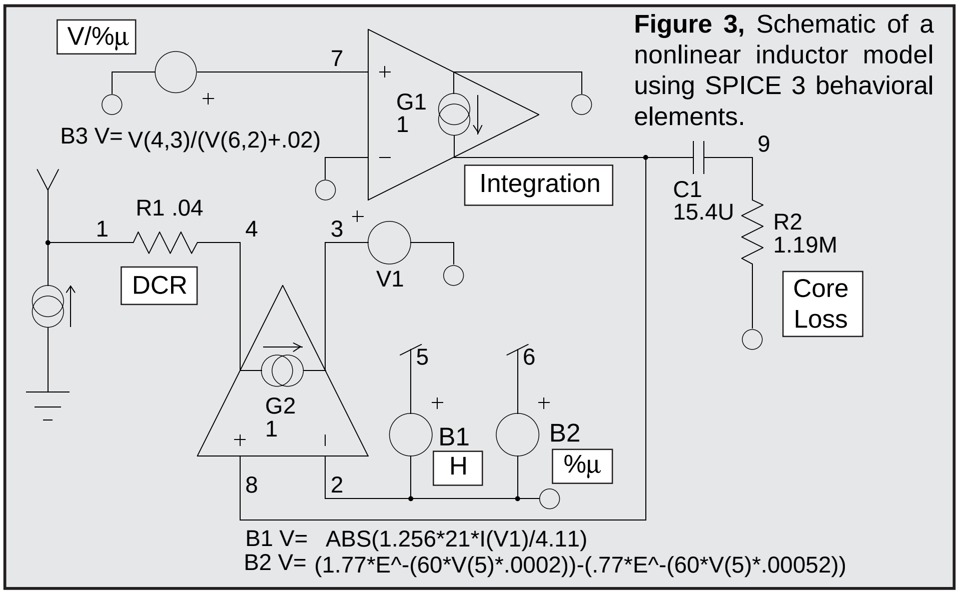

V/%µ
7
+
G1
1
−
B3 V= V(4,3)/(V(6,2)+.02)
Integration
9
C1
15.4U
R2
1.19M
Core
Loss
R1 .04
1
4
3
+
V1
DCR
G2
1
+
5
6
+
+
B1
H
B2
%µ
8
2
B1 V= ABS(1.256*21*I(V1)/4.11)
B2 V= (1.77*E^-(60*V(5)*.0002))-(.77*E^-(60*V(5)*.00052))

Figure 3, Schematic of a nonlinear inductor model using SPICE 3 behavioral elements.

in power inductors and for flyback power transformers. The cores are available in several types. The most popular are currently the 55xxx and 58xxx series. The 55xxx series has a maximum flux density of 7000 gauss and the 58xxx series has a maximum flux density of 15000 gauss, both of which are significantly higher than ferrite. Core loss of both materials are significantly higher than ferrite, and the 58xxx material has a much greater core loss than the 55xxx series.

The MPP material provides a fairly "soft" B-H loop which is ideal for applications where it is desirable to create a swinging inductance that allows a significant decrease in inductance as a function of the inductor current. This can be beneficial in applications where optimum transient response is required or to minimize the preload current of a switching power supply.

For the model to be useful, it must correctly represent the initial inductance, the incremental inductance, which represents the inductance under a DC biased condition, and the core loss.

```
.SUBCKT MP55135 1 2 10 {N=1 DCR=.01 IC=0}
R1 1 4 {DCR}
V1 3 2
G2 4 3 8 2 1
G1 2 8 7 2 1
C1 8 9 {N^2*62*1n} IC={IC}  ; {} = N^2*AL = L
R2 9 2 {48.2M*300^1.451/(62.*N^2)}
B1 5 2 V=ABS(1.256*{N}*I(V1)/0.817) ; 1.256 * NI/L, .817=LM
B2 6 2 V=(1.77*E^-(300*V(5,2)*.0002))-(.77*E^-(300*V(5,2)*.00052))
B3 7 2 V=V(4,3)/(V(6,2)+.02) ; 300 in B2 is the U(avg. permeability)
B4 10 0 V=V(6,2)*{N^2*62*1n} ; Inductance Test Point
.ENDS
```

Figure 4, SPICE 3 subcircuit listing for the nonlinear inductor model. N is the number of turns, DCR is the series resistance, and IC is an optional initial condition. The models uses the mathematical equation feature of SPICE 3.

Figure 5, In rush current for an inductor and the MPP 58121 and 55121 cores from Magnetics.

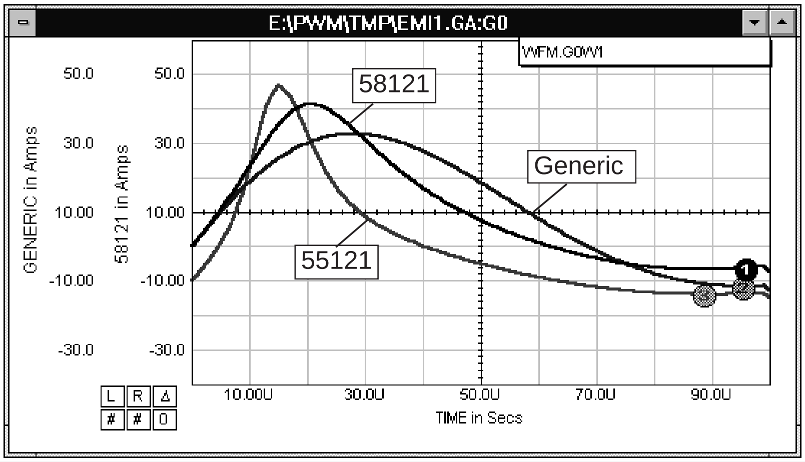


Figure 3 shows the core subcircuit while Figure 4 shows a sample netlist and key core parameters.

Initial Inductance, DC Bias, Core Loss, & Model Operation

The initial inductance is simplified by a core parameter, AL (inductance/1000 turns). The inductance in microhenries is simply computed as $N^2 * AL$. The DC bias inductance and frequency response are provided by the manufacturer in the form of graphs for each permeability available in each of the MPP materials. Using linear least squares, and some trial and error, a continuous function (difference of two exponentials) was derived which correctly calculates the % of initial permeability as a function of the DC bias in Oersteds and the initial permeability Ui. A continuous function was also derived for the loss as a function of the initial permeability Ui.

For 55xxx material: $\%Ui = 1.77e^{-0.0002\mu H} - 0.77e^{-0.005\mu H}$
For 58xxx material: $\%Ui = 1.25e^{-0.0001\mu H} - 0.25e^{-0.005\mu H}$.
55xxx material: $f = 3.3E9 * e^{-1.451}$, 58xxx material: $f = 3.3E8 * e^{-1.09}$.

V1 senses the current through the inductor. B1 uses this current to calculate the magnetizing force of the inductor in Oersteds. The magnetizing force is used by B2 to calculate the percentage of initial inductance. G1 along with C1 ($=N^2 * AL=L$) and B3 perform an integration whose result is passed to G2. G2 generates the inductor current. R2 adds a zero at the -3dB frequency of the MPP material to model the core loss.

Example - Inrush Current

A simple circuit was created to compare the results of a generic inductor and the Magnetics 55xxx and 58xxx models. The results of the simulation, in Figure 5, show that the inrush current is 25% greater with a 58121 MPP core versus the generic inductor model and 50% greater with a 55121 MPP core. In many applications this could cause concern over additional stress on the capacitors, long term reliability degra-

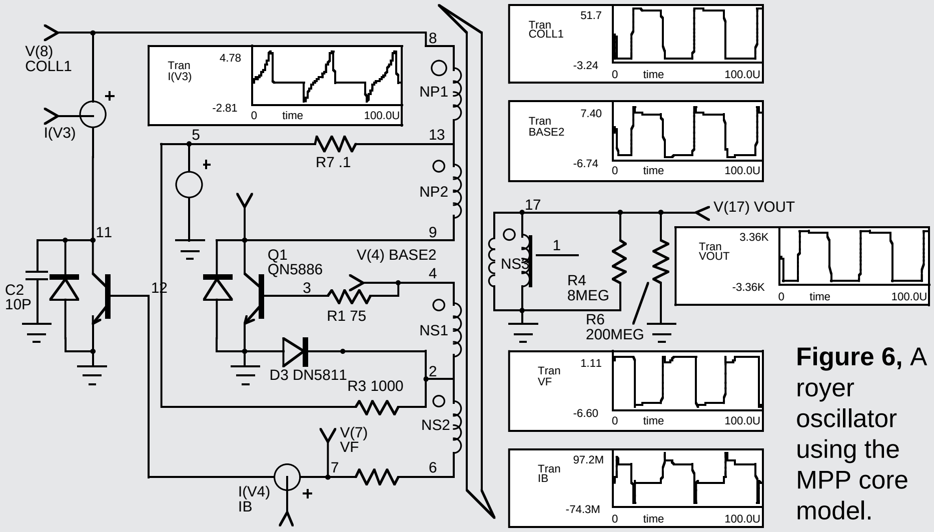


Figure 6, A royer oscillator using the MPP core model.

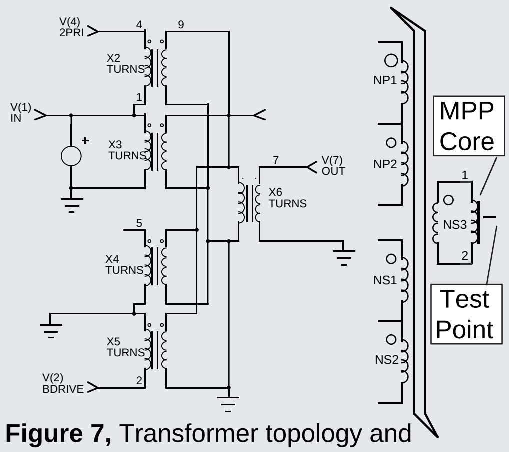


Figure 7, Transformer topology and symbol used in the Royer oscillator circuit. (Core and Inductance test point

dation, relay contacts and connector pins which could be damaged by these currents. Similar results can be seen in characteristics such as output ripple and EMI filter attenuation.

As a typical example, the MPP core was used in a Royer oscillator circuit shown in Figure 6. The transformer posed an interesting simulation challenge. The resulting structure, created from individual transformers [2], is shown in Fig 7.

Many engineers have asked Intusoft technical support if SPICE can support nonlinear magnetics. The preceding work should give the user a good indication of some of what SPICE and behavioral modeling techniques can accomplish.

Thanks to Steve Sandler, **Analytical Engineering Services** (602) 917-9727 for his contributions to this article and for the creation of the nonlinear inductor model.

References

[1] Survey of Magnetic Core Models, Peter Lauritzen, Margarita Takach, APEC Conference Proceedings 1995
[2] “Improved Spice model simulates transformer’s physical processes”, L.G. Meares, Charles E Hymowitz, EDN, August 19, 1993
[3] “An Electrical Circuit Model for magnetic Cores”, Lloyd Dixon, Unitrode Corp., Oct 1994
[4] “Theory of ferromagnetic hysteresis”, D.C. Jiles and D.L. Atherton, Journal of Magnetism and Magnetic Materials, 1986

Current Limited Power Supply

Larry Meares, Intusoft

Current limited power supplies are required to test circuits. However, the independent sources in SPICE do not have inherent current limiting capabilities, therefore, such a feature must be added separately.

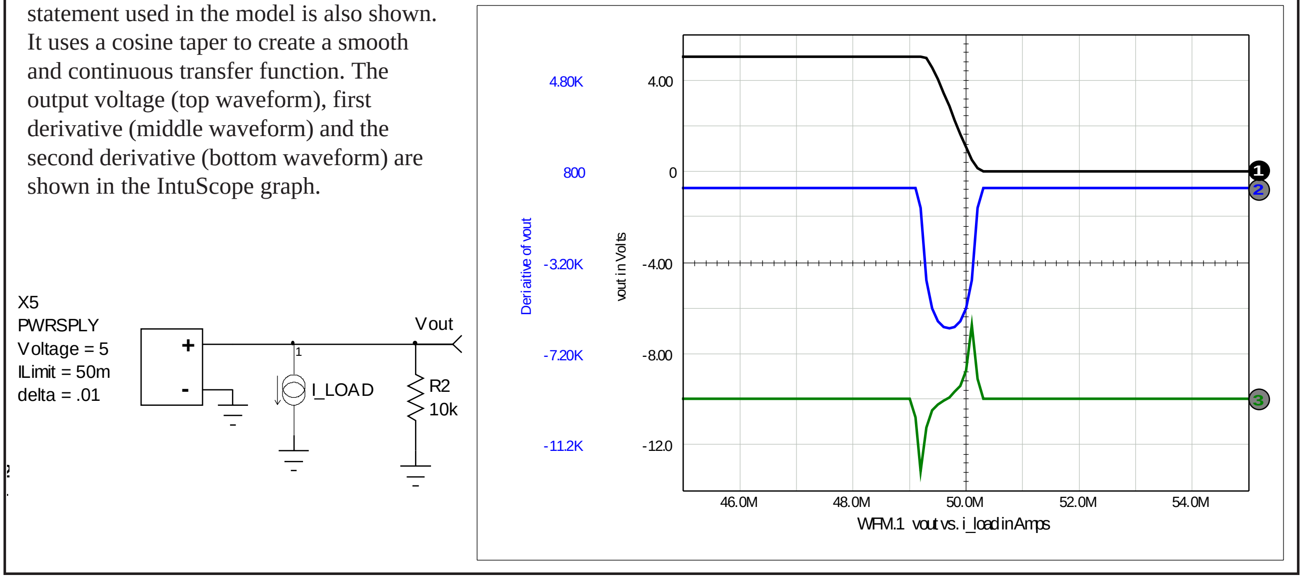
A current limiting function can be subject to convergence problems if a continuous voltage is not generated by the power supply during the current limiting operation. With traditional SPICE 2 elements there is no simple way to generate the proper response.

The IsSpice4 B element, however, allows you to create a piecewise linear response in which each region can be characterized by a nonlinear function. This is in contrast to a “table model” in which each region can only be represented by a linear response.

One such B element expression is shown in Figure 1. If uses an If-Then-Else syntax borrowed from the C programming language. Using a cosine taper function allows a smooth and controlled transition for the power supply, as illustrated in Figure 1. Function variables, set by the user on the schematic, are show in curly braces. The sharpness of the transition is controlled by delta; the threshold and low current voltage are also parameters. A complete subcircuit netlist for the current limited power supply is illustrated in Figure 2.

Figure 1, A simple test schematic for the current limited power supply is shown below. The B element If-Then-Else statement used in the model is also shown. It uses a cosine taper to create a smooth and continuous transfer function. The output voltage (top waveform), first derivative (middle waveform) and the second derivative (bottom waveform) are shown in the IntuScope graph.

```
B1 3 2 V=I(V1) < {(1-delta/2)  * ILimit } ?
+ {Voltage}  :  I(V1)  >  {(1+ delta/2)  *  ILimit }  ?  0  :
+  {Voltage * .5}*(  1+cos(3.14159*(I(V1) - ({(1-delta/2) * ILimit}))/({ILimit* delta}))  )
```



```
.SUBCKT PWRSPLY 1 2 {Voltage=5 ILimit =1 Rmin=1u delta=.01}
* Smooth cosine limiting with variable sharpness
* Parameters:
* Voltage Voltage Output in Volts,
* ILimit  Current limit in Amps for 1/2 output voltage,
* Rmin    Minimum resistance in Ohms
* delta   Fraction of Ilimit used from start of limit to 0 volts
R1 4 1 {Rmin}
V1 3 4
B1 3 2 V=I(V1) < {(1-delta/2)  * ILimit } ?
+ {Voltage}  :  I(V1)  >  {(1+ delta/2)  *  ILimit }  ?  0  :
+ {Voltage * .5}*(  1+cos(3.14159*(I(V1) - ({(1-delta/2) * ILimit}))/({ILimit* delta})) )
.ENDS
```

Figure 2, The IsSpice4 subcircuit netlist for the generic current limited power supply.

Macro Modeling Low Power DC-DC Converters

by Martin O'Hara, Newport Components Limited, U.K.

Abstract

Presented here is a macro model for low power DC-DC converters that allows fast simulation times without convergence worries. The model uses primitive circuit elements to model the transfer characteristic of the device rather than a full component model, avoiding simulation pitfalls of non convergence and speeding up the simulation time due to the simple macro model construction. The resultant macro model provides output characteristics under DC simulation conditions as well as AC noise characteristics which are shown to closely match measured device performance.

Modelling Circuits by Computer

The circuit simulation program SPICE (Simulation Program with Integrated Circuit Emphasis) has become the most popular method of designing and simulating analogue circuits by computer. Originally developed on a mini computer this is now within the means and capability of desk top PC's and available from many different software vendors. However, like any simulator it is only as good as the models it contains.

The models in a circuit simulator represent the real life behaviour of components or circuits in a mathematical form (e.g. V=IR for a simple ohmic resistor). The accuracy of these models depends on the amount of characterised behaviour available and how well the mathematics reflect the true operation of the device being modelled.

At the lowest level we have component models such as the ohmic resistor, inductors, capacitors and transistors. Component models are relatively complex for active devices such as transistors (the bipolar transistor has a 38 parameter model in SPICE) and are consequently computationally intensive (i.e. take a long time to simulate). Component models also suffer from problems of convergence, this is when the simulator is having problems resolving the actual voltage at a node due to numerical uncertainties or local computational oscillations.

The next level up from the component model is the macro model, this is where a circuit or component is described in less detail and its operation at the terminals of the device is described, possibly with quite straight forward mathematics. A typical example of a macro model would be an operational amplifier, the silicon may contain over 300 transistors to provide the circuit, however, all the user may really interested in is simulating how the gain and frequency response will change with external resistors for example. A model using simple voltage sources and a multiplier may suffice for the resistor gain characteristic and a feedback capacitor say to limit frequency. The simpler macro model would be very quick to simulate and containing only simple formulae and primitive elements should not suffer convergence problems.

The advantages of the component level model over the macro model are that the component level model will be appropriate for all users, hence the component level model should always be more accurate. Taking the amplifier circuit again for example, if one amplifier user is interested in the power supply rejection ratio and not gain, then in the amplifier model previously described the macro model will not reflect the reality of the circuit in this application. Therefore the user has to always bear in mind that with a macro model you are trading accuracy for speed of simulation and simplicity. These are considered fair trades in the electronics business and most silicon suppliers would only release a macro model and not the full component level version of their analogue IC's.

The Isolated Hybrid DC-DC Converter

The hybrid DC-DC converter has similar problems to the amplifier circuits described above, at its terminals the device looks simple, you apply say 5V DC at the input and 5V DC appears at the output. However, internally the device has to convert the input DC to AC, transfer the power via a transformer, rectify the AC, smooth back to DC and deliver at its output terminals. Also since the device is a basic power supply building block, if this takes a long time to simulate it will slow down the simulation of the whole circuit, hence macro modelling of this function is almost mandatory.

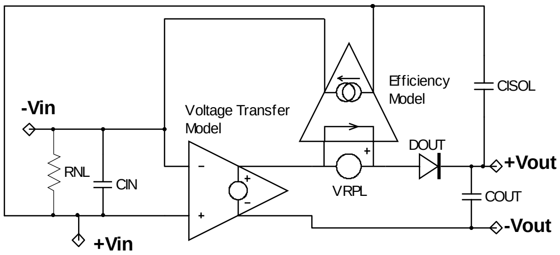
Any seasoned SPICE user may see an immediate problem for the DC-DC converter; it contains some form of oscillator to produce AC. In SPICE, if you look at circuits under DC conditions (a DC operating point for example), oscillators will have no output, hence for a 5V DC input there would be no output for a full component model of the DC-DC converter under DC simulation. Containing an oscillator and an inductively coupled transformer results in very long simulation times and many possible convergence problems, hence this approach has been discarded as impractical even for fast processors. The DC-DC converter therefore has to be examined as a macro model.

The Macro Model

The first thing to consider when looking at producing a macro model is the overall function of the device (its transfer function), then how the device appears at its terminals (i.e. high impedance, low impedance, DC blocking etc) and finally add in any additional features that need to be considered.

The hybrid DC-DC converter is a good circuit for a case study as its function is relatively simple to understand; you apply a given DC voltage at two terminals and a second isolated DC voltage appears at two other terminals. In fact this is virtually directly available in SPICE as a voltage controlled voltage source (VCVS). However, other factors that need consideration at this stage are the zero load power consumption (the converter will consume some power with no load attached), the efficiency curve, load regulation (efficiency and output voltage vary with load demand) and of course noise or ripple due to the oscillator. These are the core properties that require modelling.

The terminals of a DC-DC converter are low impedance at the input, high at the output with a DC block due to the internal rectifiers. There are also capacitors internally at input and output and these need modelling if the circuit is to simulate the effect of line filtering to and from the device. The characteristic switching noise from the oscillator is present on the DC levels at both input and output and should be included to allow users to estimate the effect this may have on their target circuit. A schematic of the subcircuit is shown below.



Primitive Elements in SPICE

The voltage controlled voltage source can be used to represent the internal transformer this should enable line regulation properties to be modelled. DC output blocking and load regulation can both be modelled using a diode model at the output. The ripple can be applied to the output as an additional voltage source, this can be used to monitor the output current drain and feed this back to an input current demand. At zero load a loss occurs due to the operation of the oscillator even without an output load, losses are usually modelled in SPICE as resistors. Finally the input, output and isolation capacitance can be added to provide suitable filter and coupling components.

Model Element Values

Deciding the elements is only half way to completing the model, values needed to be determined for all of the elements and modelling terms. The input and output capacitors were chosen from the known input and output capacitor values (COUT=CIN=1uF in the example being used here). This data is not always explicitly stated by the manufacturer, but can be determined from application notes on appropriate filter values if these are given. Isolation capacitance is in reality a measure of the capacitive coupling across the transformer, this is a very low value compared to most discrete capacitors (CISOL=24pF).

The transfer ratio of the voltage controlled voltage source should be the same as the transformer ratio, where this is not available an estimate can be gained from the voltage conversion ratio plus an additional factor for the diode drop. The loss resistor (R_L)is a simple calculation based on the known zero load power consumption (P_z) or quiescent current (I_q)and using the devices nominal input voltage (V_{nom});

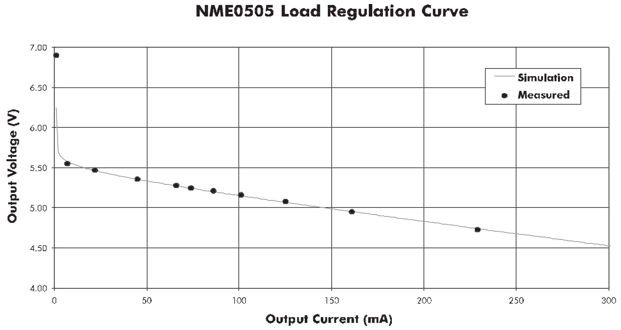
$$R_L = \frac{V_{nom}}{I_q} = \frac{Vsub\,nom^2}{P_z}$$

The AC ripple voltage was chosen to give a zero DC bias effect so as not to offset the output voltage bias levels during AC simulations, hence a small positive and negative excursion is modelled. The ripple frequency was chosen at twice the nominal switching frequency since full wave rectification is used in the actual device being modelled. The switching waveform is square from the oscillator and to model this a fast rise/fall pulse waveform is used.

```
********************************************************
*** NEWPORT COMPONENTS LIMITED
*** NME0505S DC-DC CONVERTER MODEL
*** 1W ISOLATED SINGLE OUTPUT DEVICE
********************************************************
*** NODE DESCRIPTION: VIN GND +VOUT 0V
.SUBCKT NME0505 1 2 3 4
DOUT 5 3 NCLD105
.MODEL NCLD105 D (IS=1E-7 N=2.3 RS=2.8
+ EG=1.11 XTI=3 BV=50 IBV=25.9E-3 TT=2E-9 )
COUT 3 4 0.5UF
E1 6 4 1 2 1.25
F1 1 2 VRPL -1.25
RNL 2 1 250
CIN 2 1 0.5UF
VRPL 5 6 DC 0 PULSE -0.68 0.02 0 0.25US 0.25US 4.5US 5US
CISOL 1 3 24PF
.ENDS NME0505
********************************************************
```

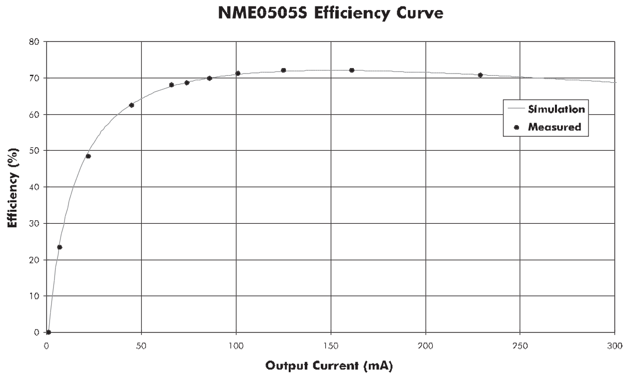
The most difficult element to determine accurately is the diode model, this has to represent the actual diode arrangement (full or half bridge) as well as junction type (p-n or Schottky). Any winding and track resistance should be included to reduce the number of elements in the finished macro model. The model was determined by starting with the supplied diode model from the diode manufacturer and adjusting various parameters until the model fitted the measured response. The basic diode model parameters were slightly changed for the load regulation characteristic, and the resistance parameter increased to include estimated track and wire resistances.

Using a current controlled current source for the feedback of load current demand also provides the method for modelling the efficiency curve. The transfer ratio of the transformer can again be used for the current feedback. The use of the noise source as the current measurement element, also results in a feedback of this noise to the input, hence no further noise source is required, plus the input and output noise is synchronised as occurs in the actual device.

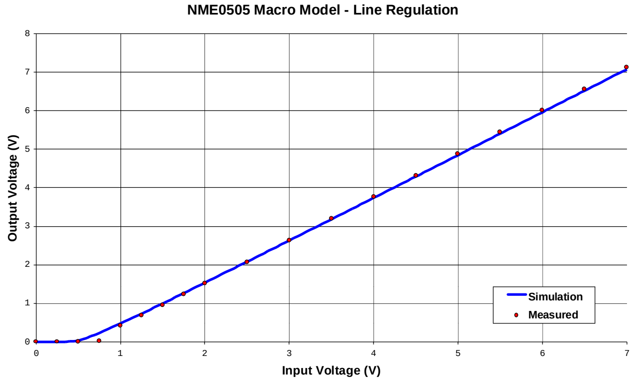


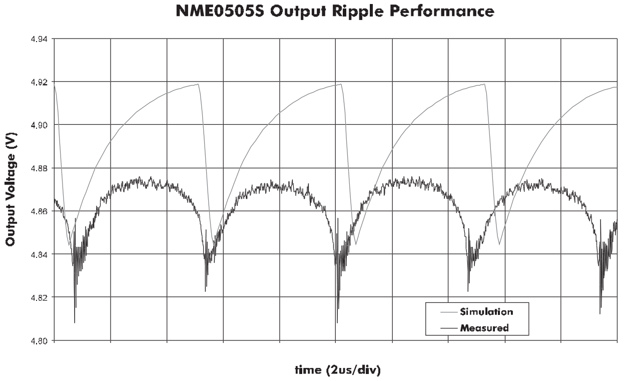
Simulation Against Measured Performance

Once the model was developed several aspects of measured performance against simulated performance were made. The two of most interest were the load regulation and the efficiency curves. These were measured under DC conditions, hence it was interesting to see that the model worked well, unlike the oscillator version, and simulated quickly. Both efficiency and load regulation curves were almost exact fits over the operating range of the device. The largest discrepancies occur when the device sees a load demand exceeding its normal limit, in the real circuit the oscillator will fail to supply the current as it is current limited itself, and the output voltage falls much more rapidly than the simulator suggests.



Other examined parameters such as line regulation (the variation of output voltage with input voltage) is well modelled as is the noise characteristic. The only problem with the noise value is that this has to be examined in an AC or transient signal simulation which is more difficult to measure than the DC parameters. The ripple from a DC-DC converter under full load was measured on a digital storage oscilloscope (DSO) and compared with the simulated result. Again both showed very close agreement, even the DC offset value is within 50mV. The ripple source model was designed to add no DC offset under simulation by having it swing between a small positive and negative value. Originally this was done to prevent the ripple source overcoming the diode block at zero input, but has also helped in preventing the output suffering a DC offset under transient simulation.





Limitations of the Macro Model

The macro model is a useful tool for giving the circuit designer an indication of how the DC-DC converter will work under various load and line conditions over the devices normal operating region. The main limitations are when the device is being driven outside of its normal operating region.

Below its quoted minimum load (10%) the device can have a relatively large excursion of output voltage, up to 50% above its nominal output at zero load. The model only predicts a slight increase at zero load, typically in the 20% above nominal region. This is primarily due to the operation of the real circuit and the model. The oscillator in the real circuit continues to charge up the capacitors at a very low charge rate until a quiescent point is achieved between the very low charge and the leakage of the output circuit. In the model the limit to the output voltage excursion at zero load is still the diode model regulation characteristic, hence it is lower than the trickle charge capacitor effect. Despite this difference the model is still useful below this minimum load level as it remains in a defined state.

At loads exceed the recommended full load of the DC-DC converter the model again fails to accurately describe the action of the real circuit. At high loads (>100%) the oscillator is current limited by design and the charge available from the output capacitors drops as the load demands increases over 100%. The net effect is that the load regulation begins to fall dramatically after around 125% load and the ripple increases significantly. The model simply continues to predict a gradual fall of the output along the same regulation curve with constant ripple. This does need to be borne in mind if the simulation of the load circuit causes the DC-DC converter to be overloaded, the resultant DC supply levels are unlikely to reflect the true situation.

The other limitation in the macro model is lack of modelling temperature effects. This is not a major problem as the effect of temperature on the operation of the DC-DC converter is relatively small. These small temperature dependant effects could be included quite easily at a later date into the primitive element models.

Although there are some limitations, this is to be expected of a macro model, especially where an oscillator is being replaced by a DC operated circuit. The limitations are very minor over the normal operating range of the DC-DC converter and are certainly worth the price for a macro model that simulates operation under DC conditions.

Summary

The model works well over the normal operating region of the DC-DC converters examined. The characteristics of line regulation, load regulation, efficiency and noise are all accurately simulated in both DC and transient signal analyses. The macro model is simple to construct from either design or specification data and provides fast simulation of the DC-DC converter circuit without convergence problems.

[1] Switching Power Supply Design, A.I. Pressman, McGraw-Hill, 1991.
[2] IsSPICE User’s Guide, Intusoft, 1994.
[3] Power Supply Design Seminar, SEM-1000, Unitrode, 1994.

newport components

Newport Components Limited, 4 Tanners Drive,
Blakelands North, Milton Keynes, MK14 5NA
Tel: 01908 615232 Fax: 01908 617545

Modeling Non-Ideal Inductors in SPICE

Martin O'Hara Technical Manager, Newport Components, U.K. November 8 1993

Abstract

The non-ideal inductor exhibits both resonance and non-linear current characteristics. These effects can be modelled in SPICE by adding only 3 additional elements to model the real inductor characteristics of dc resistance, wire capacitance and magnetic core loss. The values for these model parameters can all be obtained from standard data sheet parameters via a few simple calculations. The resulting model gives accurate impedance and phase simulations over a wide frequency range and over the peak resonance frequency. The dc current saturation characteristics is modelled by a simple 2nd order polynomial that gives a close simulation to measured performance over 2.5 times the recommended dc current limit. Comparisons of measured inductor performance and simulation results are given to illustrate the proximity of the models to real inductor behaviour.

Introduction

Modelling of inductors and inductive elements in SPICE has always been of low importance to analogue designers. This is partly because SPICE was developed primarily for IC design where inductive elements are usually parasitic and very small.

The widespread use of SPICE for discrete analogue circuit design has seen the program being used to analyze switching power supplies and filters in which the behaviour of the inductive element is critical to the accuracy of the simulation. In general these circuits operate using ideal inductors reasonably well since the current and frequency of operation are in the ideal operating region of the inductors used (i.e. relatively low frequency and well below the saturation current limit).

More recent applications employing inductive elements are electromagnetic interference (EMI) filters, in which the resonances across a very wide range of frequencies needs to be examined. Likewise employing inductors in dc supply filters can put the inductor near its dc saturation region. In both later cases the modelling of the non-ideal behaviour of the inductor is important for accurate predictions of circuit performance.

Real Inductor Behaviour

In an ideal inductor the impedance (Z) is purely reactive and proportional to the inductance (L) only; The phase of signal across the ideal inductor would always be +90° out of phase with the applied voltage and there would be no effect of DC current bias on the behaviour of an ideal inductor.

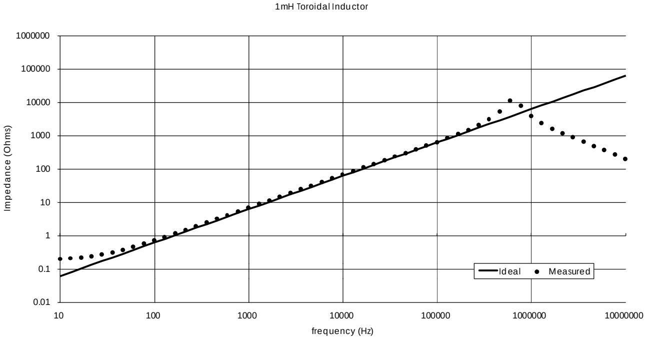


Figure 1

If we compare the measured frequency response for the impedance of a real inductor to the ideal model we can see two distinct differences at either end of the frequency spectrum (figure 1). At the low frequency (near DC) there is a dominant resistive element, observed in the constant impedance value and loss of the phase shift. At high frequency the inductor goes though a resonance peak and the impedance then falls and a voltage phase shift of -90° is observed, indicative of capacitive dominance. The frequency response is therefore observed to be non-ideal, however, it can be stated that near ideal behaviour does occur over the majority of the inductors operating region.

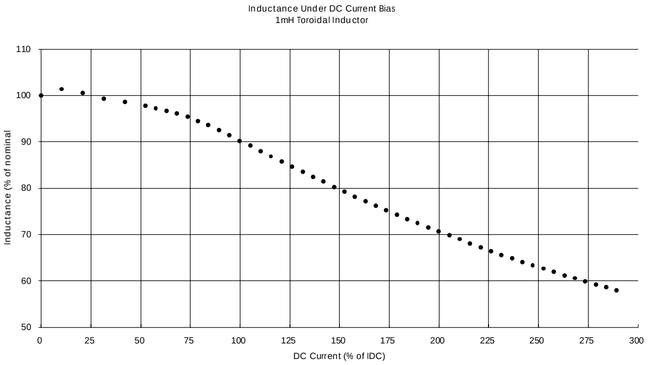


Figure 2

Under DC current bias there is a loss of inductance due to magnetic saturation. This is observed as a fall in inductance as the DC current through the inductor is increased.

Modelling Non-Ideal Behaviour

There are essentially two non-ideal characteristics[1] that are encountered when using an inductor; one is the resonance of the inductor and the other is magnetic saturation. Since these essentially act in different analyses in SPICE (i.e. AC or DC analyses), they can be considered separately, although combined into a single model.

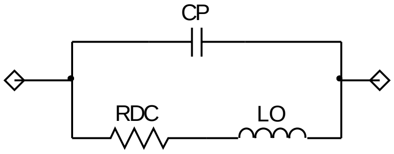


Figure 3: Basic Inductor Model

The additional parasitics that cause the behaviour of an inductor to be non-ideal over the frequency range can be easily visualised and characterised. There are essentially two additional parameters that contribute; the dc resistance of the wire and its self capacitance (figure 3). These two additional parameters can usually be easily obtained from the specification for the inductor, hence additional measurement by the circuit designer should not be required, just a few simple calculations.

The series resistance is obtained simply from the quoted dc resistance of the inductor (R_{dc}). The parallel capacitance(C_p) can be obtained from the self resonant frequency of the inductor, since at this frequency the reactance of the wire capacitance (X_C) and the reactance of the inductance (X_L) are equal. Hence the capacitance can be expressed as;

$$C_p = \frac{1}{(2\pi f_o)^2 L_o} \qquad \textbf{(1)}$$

Where f_o is the self resonant frequency.

The effect of dc current causing magnetic saturation can be modelled as a simple second order polynomial. In SPICE 2G6 this was available directly in the standard polynomial inductor model by using the POLY key word after the node description.

The polynomial is specified by the equation;

$$L_I = L_o + L_1 I + L_2 I^2 + \ldots + L_n I^n \qquad \textbf{(2)}$$

where n≤20.

An inductor specification usually gives the dc current (I_{dc}) at which the inductance falls to 90% of its nominal value (L_o). Hence, using a second order approximation, the equation becomes;

$$0.9\, L_o = L_o + L_2 I_{dc}^2 \qquad \textbf{(3)}$$

Yielding a second order coefficient of;

$$L_2 = -\frac{0.1\, L_o}{I_{dc}^2} \qquad \textbf{(4)}$$

Note that the first order co-efficient will have to be specified as zero.

In SPICE 3E2 the polynomial inductor is no longer available and a more complex method of modelling this effect is required using the non-linear element B and a zero value voltage source to measure the current through the inductor.

The complete polynomial inductor of SPICE 2G6 can be written as a subcircuit in SPICE 3E2.

```
XL 1 2 POLYL Lo L1 L2 L3 ....
```

```
.SUBCKT POLYL 1 2
V1 1 3 DC 0
LO 3 2 Lo
B1 2 3 I=I(V1)^2*L1/(2*Lo)+I(V1)^3*
+ L2/(3*Lo)+I(V1)^4*L3/(4*Lo)+....
.ENDS
```

Here we are only interested in modelling the second order polynomial, hence only the L_2 term is of interest. This can be determined from the SPICE 2G6 coefficients, or directly from the maximum DC current value.

$$L_{3E2} = \frac{L_2}{3\, L_o} = -\frac{1}{30\, I_{dc}^2} \qquad \textbf{(5)}$$

The polynomial sub-circuit can hence be rewritten;

```
.SUBCKT POLYL 1 2
V1 1 3 DC 0
LO 3 2 Lo
B1 2 3 I=I(V1)^3*L3E2
.ENDS
```

It should be noted that this subcircuit only replicates the polynomial equation from version 2G6, the additional model elements also need to be added.

Simulation and Test Results

A radial leaded bobbin inductor (14 105 40) was measured for the non-ideal characteristics described above. Impedance and phase were determined on a Hewlett-Packard HP4192A Low Frequency Impedance Analyzer, dc current characteristic was determined on a Wayne-Kerr (WK) 3245 Precision Impedance Analyzer and 3220 Bias Unit.

The effect of dc current saturation proved long winded to simulate. The reason for this is that an AC analysis cannot be performed concurrently with a DC sweep. Hence the dc current through the inductor had to be manually changed and the circuit re-simulated to get a simulation of impedance over a range of dc current (the inductance was calculated from the simulated impedance characteristic from SPICE and read directly from the WK3245).

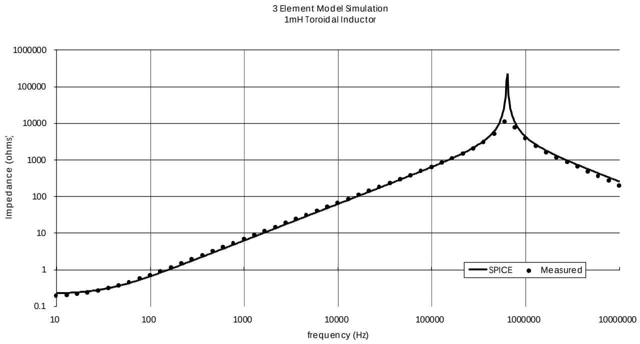


Figure 2: Impedance Analysis

Discussion

Impedance results proved to be exceptionally well matched, the only discrepancy being a slight difference in the resonant frequency. The difference in resonant frequency is purely a production variation, the model is centred on the typical value of 800kHz, whereas the sample used was resonant at 696kHz.

In simulation it is important that the measuring instrument is modelled as closely as possible so that any effects this may load onto the component is determined. The problem of the measurement system model is clearly illustrated in the phase results. If a 50O oscillator source to load impedance is used (as suggested in the HP manual) a poor simulated phase response is observed due to the loading of the source, however, using a 1MO impedance gave an accurate simulated match to the measurement characteristic (the simulated impedance response was the same with either source impedance).

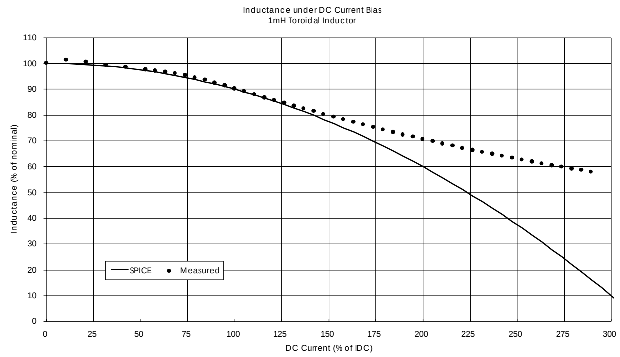


Figure 3: DC Current Analysis

The simulated dc current characteristic looks dissimilar to the measured result. The initial inductance is higher for the measured part (1.05mH) and it can be observed that the characteristic is more likely a 3rd order polynomial expression. However, the simulation is reasonably close and estimates a worse case (particularly since the I_{dc} current value for the sample used was nearer 5A; Newport inductors are always specified conservatively). The shape of the characteristic is reasonably close over 2.5 times the parts recommended operating current shown and using the 2nd order polynomial rather than a 3rd means that additional measurements are not required. What the simulation implies is that it results in the worse case characteristic for the least effort.

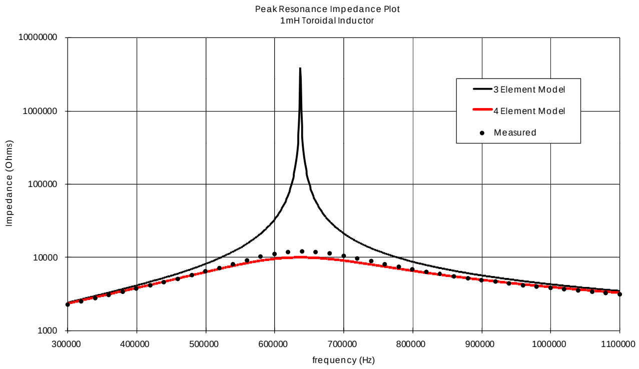


Figure 4: Peak Resonance Impedance Analysis

Peak Resonance

If the peak resonance is more closely examined it is observed that there is some disparity between simulation[2] and measurement for both the peak impedance result (figure 4) and rate of phase change (figure 5). This can be expected in that there is no provision is this simple model for the finite loss in the magnetic material.

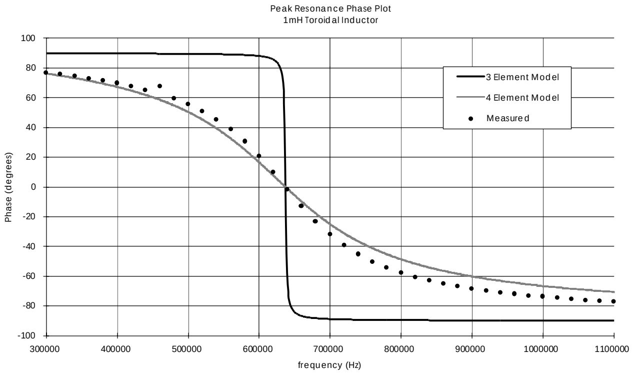


Figure 5: Peak Resonance Phase Analysis

The magnetic loss can be modelled reasonably well as a parallel resistor (R_p) across the existing model. The value can again be calculated from data sheet parameters, using the quality factor (Q). In a parallel RLC[3] circuit the relationship between the quality factor and inductance is given by;

$$Q = \frac{R_p}{2\pi \, f_o L_o} \qquad \textbf{(6)}$$

The data sheet value for the 14 105 40 inductor used here is Q=49, hence a parallel resistance of 246kO is calculated (225kO using the actual values for the sample part).

The resulting simulations for impedance and phase now match the measured results exceptionally well. This indicates the method for determining R_p is a reasonable approximation from a circuit designers point of view.

The above simulation results suggest the model gives a reasonably good approximation to the real behaviour of the inductor over a wide frequency and current range. The improvement has also been gained for no additional measurements, which means that the model can be derived from the component specification.

Limitations

The above model is now quite sophisticated for an inductive element, however, there are still limitations and this should be borne in mind. The model assumes that there is no variance of resistance and capacitance with dc current, at low values of these parameters this may be adequate as these will tend to be swamped by the rest of the circuit.

Negative inductance values can be obtained when the dc current exceeds approximately 3.16 times the I_{DC} value[1] . In the SPICE 3E2 sub-circuit this can be compensated for by putting in an IF..THEN conditional statement, this can either set the value of the inductor to zero, or some specified value. In SPICE 2G6 this facility is not available, it is therefore advisable to have a some measure of the dc current in the circuit element if it is suspected that the dc current is greater than 2.5 times the I_{DC} value.

Parameter Tolerances

The tolerance for inductance is usually specified in the data sheet (±10% for the sample used), however, few of the other parameters have a tolerance figure. In the cases of R_{dc} and I_{dc} these are worse case values and no other tolerance is required.

The tolerance of the self resonant frequency is related to the inductance value and wire capacitance. The tolerance of the wire capacitance is difficult to estimate accurately since, even with machine wound products, the value is so small that slight variations in winding cause noticeable changes in the capacitance. As an estimate, it could be expected that ±20% variation in the value of C_p would be observed.

The tolerance in the R_p value is also difficult to determine, even the core manufacturers do not usually specify tolerances of the loss parameters of the core. Again a ±20% tolerance is predicted to be sufficient to allow a Monte-Carlo analysis to accurately predict worse cases.

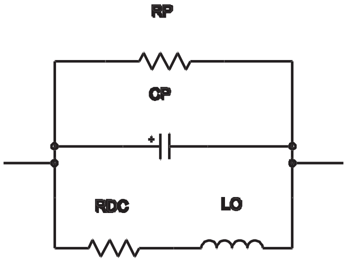


Figure 6: Completed Inductor Model

Summary

It is possible to simulate several complex aspects of inductor operation in SPICE using only 3 extra passive elements (figure 6) and a simple polynomial expression. The resulting model gives accurate inductor simulations over a wide range of operating conditions with a minimal increase in computation time (only one extra node is introduced) and no additional measurements are required.

SPICE 2G6 Example

The following example is a model for a Newport Components 1400 series 1mH inductor (14 105 40) where L_o=1mH, R_{dc}=0.173O, I_{dc}=4.0A, Q=49 and f_o=800kHz.

```
.SUBCKT IND14105  1  2
LO  3  2  POLY  1E-3  0  -6.25E-6
RDC 1  3  0.173
CP  1  2  39.6E-12
RP  1  2  250K
.ENDS
```

SPICE 3E2 Example

The following example is a model for a Newport Components 1400 series 1mH inductor (14 105 40) where L_o=1mH, R_{dc}=0.173O, I_{dc}=4.0A, Q=49 and f_o=800kHz.

```
.SUBCKT L14105  1  2
LO  1  4  1E-3
V1  4  3  DC 0
B1  4  1  I=I(V1)^3*-2.08E-3
RDC 3  2  0.173
CP  1  2  39.6E-12
RP  1  2  250K
.ENDS L14105
```

References

[1] SPICE Version 2G1 User’s Guide, A. Vladimirescu, A.R. Newton, D.O. Pederson, UCB, 1980 (+2G6 upgrade, undated).
[2] Electric Circuits, 2nd ed., J.A. Edminster, Schaum’s Outline Series, McGraw-Hill, 1983.
[3] Soft Ferrites, 2nd ed., E.C. Snelling, Butterworths, 1988.
[4] IsSPICE 3 User’s Guide, Intusoft, 1992
[5] Private communication, M. Penberth, Technology Sources Ltd, November 1993.
[6] Inductor Databook, Newport Components Limited, 1993.

newport components

Newport Components Limited, 4 Tanners Drive,
Blakelands North, Milton Keynes, MK14 5NA
Tel: 01908 615232 Fax: 01908 617545

Motor and Relay Modeling

Authors
Mike Penberth, Technology Sources, U.K.
Charles Hymowitz, Intusoft

MODELING A RELAY

by Mike Penberth, Technology Sources, U.K.

In this article we'll look at how IsSpice4 can be used to model a relay in detail. Figure 12 shows the full model which is split into the following parts:

The Coil and Drive

The transistor drives the coil which is represented by the components between the collector and supply, V3. We might usually think in terms of using a voltage source in the coil circuit to develop the back emf, but this implies differentiating the flux in the magnetic circuit. In this model, we integrate the voltage across the coil (less the resistive drop) to obtain the flux at node 7. When performing a simulation, integration is always preferable to differentiation, and the LAPLACE function provides a very fast and simple integrator.

The Magnetic Circuit & Force

The reluctance of the magnetic circuit has two series components, that of the fixed iron and air path and that of the changing solenoid air gap. For simplicity, we initially assume no fringing and a linear relation between gap and reluctance. Flux flows through the series combination of the reluctances to give a "potential" which is the ampere-turns, V(17). From the ampere-turns we have the current flow in the coil from nodes 1 to 12.

The magnetic force on a body is the integral of the square of the normal component of B over the surface. For a simple solenoid plunger with little fringing, we initially assume B squared times the plunger face area. This, in turn, is the square of the flux divided by the area.

The Moving Air Gap

The plunger acceleration is the result of the sum of the forces which are acting upon it. The acceleration can be double integrated to obtain position with respect to time. The other forces acting on the plunger are friction and the hard stop. Friction is calculated from velocity via a high gain limiter; this gives a constant force whose sign is dependent upon the direction of movement, i.e. static friction. The characteristic of the stop is obtained from a table model which gives a high spring rate for negative gap. The gap position, V(14), is fed back into the magnetic circuit to control the air gap reluctance.

Arcing

The solenoid plunger is connected to the relay contacts. The contact gap closes before the plunger hits the end stop. A limiter

Modeling A Relay

Mike Penberth, Technology Sources, U.K., Charles Hymowitz, Intusoft

Excerpts from the March 1997 Intusoft Newsletter #49

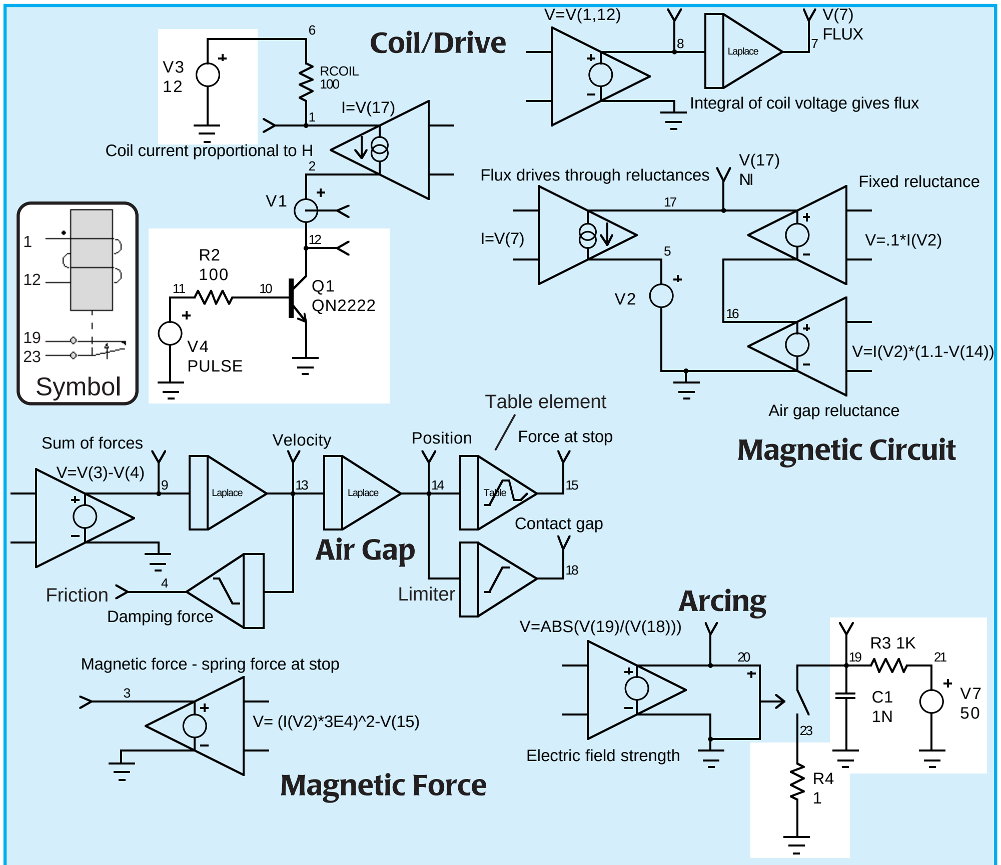


Figure 12, IsSpice4 model of a relay. The model includes the effects of coil impedance, back emf, nonlinear air gap-reluctance relationship, plunger position, plunger friction, plunger stop elasticity, and arcing. Test circuitry is shown in white.

function is used to generate the contact gap from the plunger position. Arcing will occur when the electric field strength across the gap exceeds a certain value. By calculating the field strength and applying it to a switch at node 20, we can short the relay contacts to simulate arcing.

Results

The plot of Figure 13 shows the plunger bouncing off the end stop following coil energization and the rise of current in the coil. Figure 14 shows some detail of arcing at the contacts, node 19, and the contact gap, node 18. An example IsSpice4 netlist is shown in Table 2.

Comments

The constants in this example are purely arbitrary and do not represent any real device. When modelling a real device, it would be relatively easy to introduce some second order effects such as non-linear reluctance change with air gap by using either the equation facilities in the B source, or the table model to introduce data obtained from a field modelling package.

As the air gap widens, fringing increases and the effective magnetic path length becomes longer than the gap. This can be introduced as a second order term in the B source, or added as a nonlinear behavioral resistor.

```
Rgap 16 0 R=(1.1-V(14))+(1.1-V(14))^2
```

As presented previously, the stop is perfectly elastic, energy loss in the stop can be added by subtracting another force computed from the stop force times plunger velocity. For example by changing B10 to read:

```
V=V(3) - V(4) - V(13) ∗ V(15) ∗ .0001
```

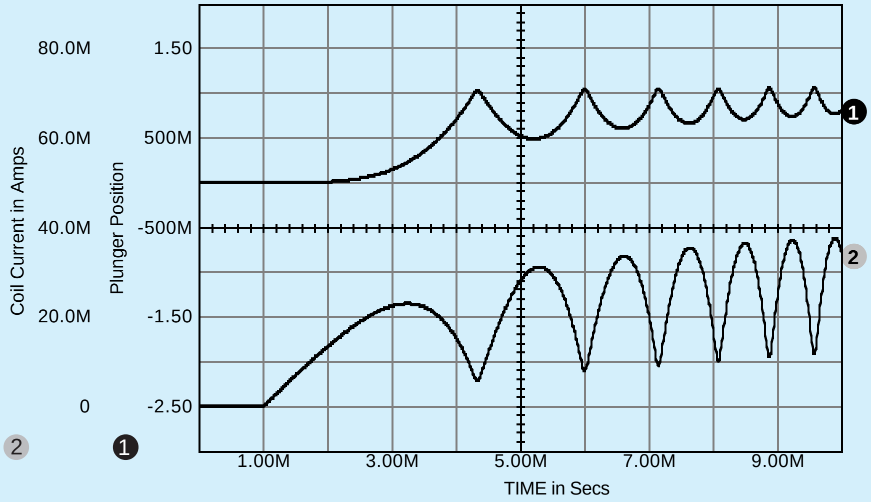


Figure 13, The plunger bouncing off the end stop can clearly be seen from this graph of the plunger position and the energizing coil current.

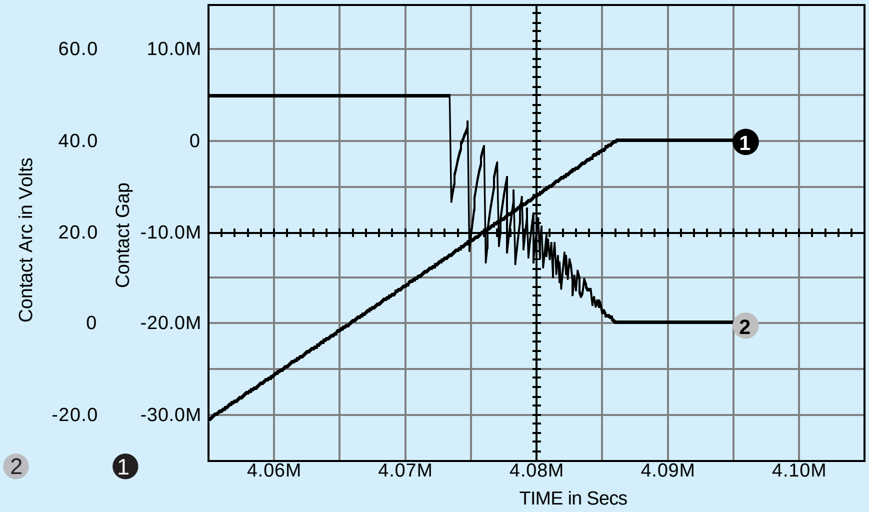


Figure 14, The relay model includes the effects of arcing as shown by this plot of the relay contact voltage as the contact gap is closed.

```
.SUBCKT RELAY 1  12  19 23
*Connections     Crtl+ Crtl-  Out+ Out-
V1 2 12
S1 19 23 20 0 SW1
.MODEL SW1 SW RON=10M ROFF=10G
+ VT=3K VH=1K
B13 20 0 V=ABS(V(19) / (V(18)))
B5 8 0 V=V(1) - V(12)
A1 8 7 S_001
.MODEL S_001 s_xfer(in_offset=0 gain=1.0
+ num_coeff=[1.0] den_coeff=[1.0  0]
+ denorm_freq=1.0)
A2 9 13 S_002
.MODEL S_002 s_xfer(in_offset=0 gain=1.0
+ num_coeff=[1.0] den_coeff=[1.0  0] out_ic=0
+ denorm_freq=1.0)
A3 13 14 S_003
.MODEL S_003 s_xfer(in_offset=0 gain=1.0
+ num_coeff=[1.0] den_coeff=[1.0  0] out_ic=0
+ denorm_freq=1.0)
A4 14 15 PWL_001
.MODEL PWL_001 Pwl(xy_array=[0  0  1.0
+  0  1.1  100.0Meg  1.2   200.0Meg]
+ input_domain=10.0M fraction=TRUE)
A5 13 4 LIM_002
.MODEL LIM_002 Limit(in_offset=0 gain=1K
+ out_lower_limit=-10 out_upper_limit=10
+ limit_range=1.0U fraction=FALSE)
A6 14 18 LIM_001
.MODEL LIM_001 Limit(in_offset=-0.8
+ gain=1 out_lower_limit=-10K
+ out_upper_limit=-1U limit_range=1U)
B7 1 2 I=V(17)
B8 17 5 I=V(7)
V2 5 0
B12 17 16 V=.1 * I(V2)
B4 16 0 V=I(V2) * (ABS(1.1-V(14)) +
+ (1.1-V(14)) * .5  ; Use Rgap for
* nonlinear air gap effects
* Rgap 16 0 R=(1.1-V(14))+(1.1-V(14))^2
B10 9 0 V=V(3) - V(4) -
+ V(13)*V(15)*0.0005   ; added for
* Imperfect plunger stop
B11 3 0 V=(I(V2) * 3E4)^2 - V(15)
* 3E4 Plunger area
.ENDS
```

Table 2, IsSpice4 relay netlist.

Relay Model Comparison

Many analog simulation tool vendors claim to have a large number of SPICE models in their libraries. But when you look more deeply into their offerings, you will see that there are significant shortcomings for which numbers of parts are not a substitute.

Take, for example, the relay we just reviewed. The relay models found in Electronic Workbench and CircuitMaker are shown in Figure 15. The relays use only the ideal SPICE (two-state) switch to model the relay. No nonlinear inductive, mechanical or arcing effects are included in their models.

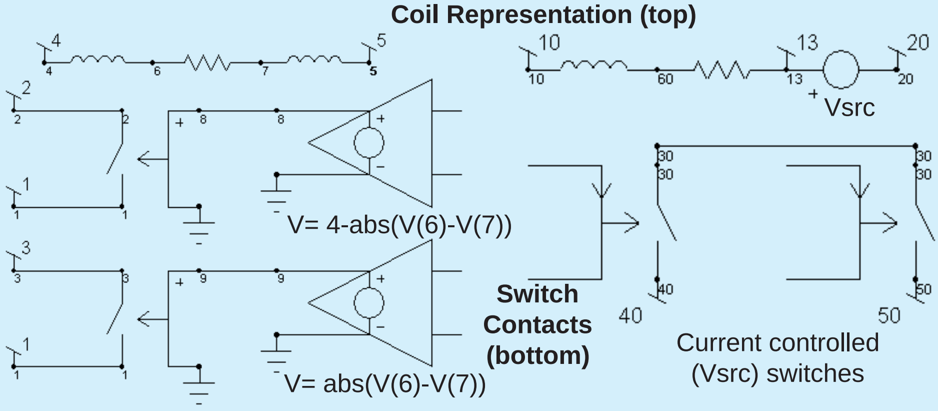


Figure 15, The relay models from CircuitMaker (left) and Electronic Workbench (right) don't include key features found in the real devices. The models are first order and support only coil inductance, turn-on current, holding current, and coil resistance.

By comparison, the models in these products are simplistic and only model “first order” phenomenon. First order models have their place in the simulation process. They are quite useful, especially for system level simulations, and Intusoft includes them in ICAP/4.

However, while these software products can be less expensive, you will find that you really get what you pay for when it comes to model features and performance. Good models are CRITICAL to a successful simulation, and they don’t come cheap. The quality of a model library should not be judged by how many models are included, but by the effects that the models emulate. The number of models simply doesn’t tell the whole story.

For further information on this article, contact Mike Penberth at: **Technology Sources Ltd. U.K. Tel: +44-01 223 516 469, Fax: +44-01 223 729916, Web - www.softsim.com.**

New SPICE features aid motor simulation

Mike Penberth, Technology Sources, U.K.

Nano-technology is increasingly featured in the science pages of serious newspapers as well as the technical journals. The developments in this field are often demonstrated by the construction of a capacitance motor built to dimensions to which few of us can relate. What is a capacitance motor and what are its characteristics? This article explores capacitance motors using SPICE as a simulation tool and taking advantage of some new features added to IntuSoft's latest version of this popular simulator.

The primary feature of a motor is that it produces torque. If this torque changes sign as a function of shaft angle the device is usually referred to as an actuator. Many types of motor consist of a number of actuators set with an angular spacing so that by switching the drive from actuator to actuator a continuous torque can be produced. Capacitance motors are in this class.

Take the simple geometry of Fig 1. Here a pair of rotor electrodes can vary in angular alignment with a pair of stator electrodes. The capacitance seen across the stator pair varies with angle reaching a maximum twice per revolution. If a voltage V is applied across the stator plates a torque T will be developed in a sense that will seek to maximise the capacitance. The torque can be calculated from the equivalence in the work done in moving through a small angle $\delta\theta$ and the change in electrical stored energy due to the change in capacitance across the small angle.

$$T\delta\theta = \tfrac{1}{2} C_1V^2 - \tfrac{1}{2} C_2V^2 = \tfrac{1}{2} \delta CV^2$$

giving:

$$T = V^2 \, dC/d\theta$$

Note: The V^2 dependence shows the torque direction is independent of the polarity of applied voltage.

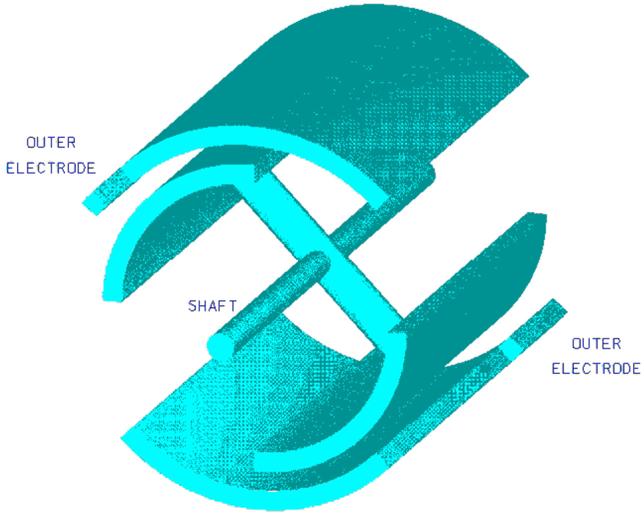


Figure 1

Our simple geometry suggests a triangular wave variation of capacitance with position, maximum at 0 & 180 degrees, zero at 90 & 270 degrees. Field fringing and construction details will modify this ideal into a more complex function.

In a real motor we could calculate the capacitance at a number of angles using field modelling software with a parameter extractor. Using this data we could then curve fit to a function.

It is at this point that we come across our first simulation problem. In looking at the motor dynamics we will want to simulate across several revolutions of the motor. This presents no problem if our curve fit is to a periodic function such as a sine wave and we could make other functions periodic by expanding them as a Fourier series. There is now an easier way. IntuSoft have introduced a number of extensions to the arbitrary source syntax one of which is a fractional function. For our purposes, if the rotor position in radians divided by 2π is the argument of the FRAC function the output will be the fractional part of the total number of revolutions. We can use this fractional part as the argument in a non-periodic function to give a repeating capacitance variation with angular position.

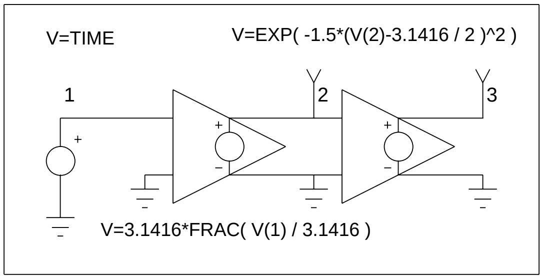

V=TIME
V=EXP(-1.5*(V(2)-3.1416 / 2)^2)
1
2
3
+
V=3.1416*FRAC(V(1) / 3.1416)

Figure 2

In the SPICE circuit of Fig 2 I have chosen to use a gaussian function to represent the change of capacitance with angle. Since there are two maxima per revolution the angular position is divided by π not 2π.

The resulting capacitance against angle plot is shown in Fig 3.

The expression for torque includes the derivative of capacitance with respect to angle. For the Gaussian function chosen this has the analytic form :

$$-2\theta e^{-\theta^2}$$

The circuit of Fig 2 is extended in Fig 4 to generate the torque output from the drive voltage input. The capacitor across the input drive voltage represents the actuator input capacitance and uses another new feature of IntuSoft's SPICE that allows capacitor (and L & R) values to be written as functions of circuit voltages and currents. An additional input allows us to rotate the torque characteristic through an offset angle.

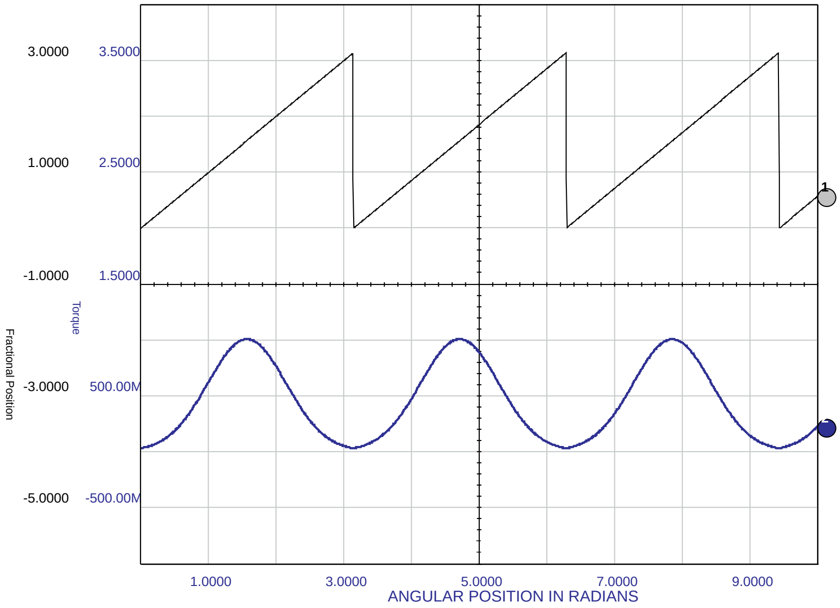


Figure 3

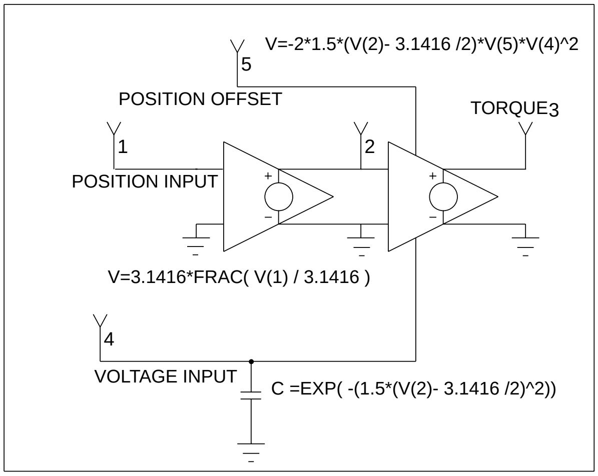

V=-2*1.5*(V(2)- 3.1416 /2)*V(5)*V(4)^2
5
POSITION OFFSET
TORQUE3
1
2
POSITION INPUT
+
−
+
−
V=3.1416*FRAC(V(1) / 3.1416)
4
VOLTAGE INPUT
C =EXP(-(1.5*(V(2)- 3.1416 /2)^2))

Figure 4

A typical motor might consist of three of these actuators disposed at 120 degree angles. The torque / rotation curves are then as shown in Fig 5 and we can see that we can always energise at least one of the actuators to produce a positive or negative torque.

The next step is to use the torque/rotation characteristic to simulate the dynamics of the motor.

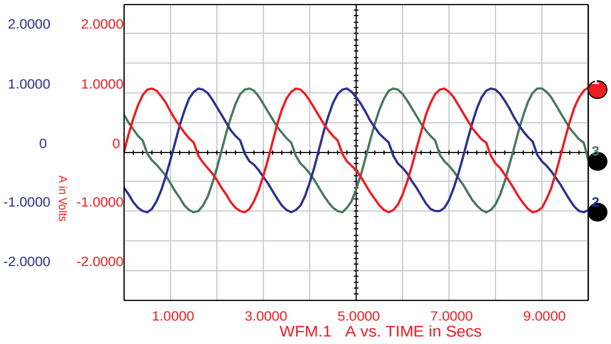


Figure 5

Motor position is the double integral of the torque/ inertia ratio. A compact way of producing the double integral is using an L and C combination around a gain stage as in Fig 6. The LC node is a virtual earth point, the current in the inductor then represents angular velocity with the output voltage representing angular position, since

$$V_{out} = \int\int V_{in}/LC$$

L is equivalent to inertia while C scales the output.

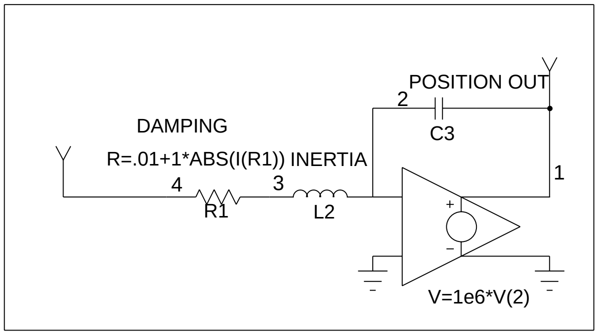


Figure 6

Any practical motor contains and drives some form of friction, friction absorbs torque that would otherwise accelerate the motor and is related to angular velocity. A resistor in series with the inductor effectively absorbs an amount of torque proportional to velocity i.e. viscous friction. Some losses (e.g. windage) increase with the square of velocity. Here we can use the ability to write expressions for resistance values to make the resistance proportional to velocity.

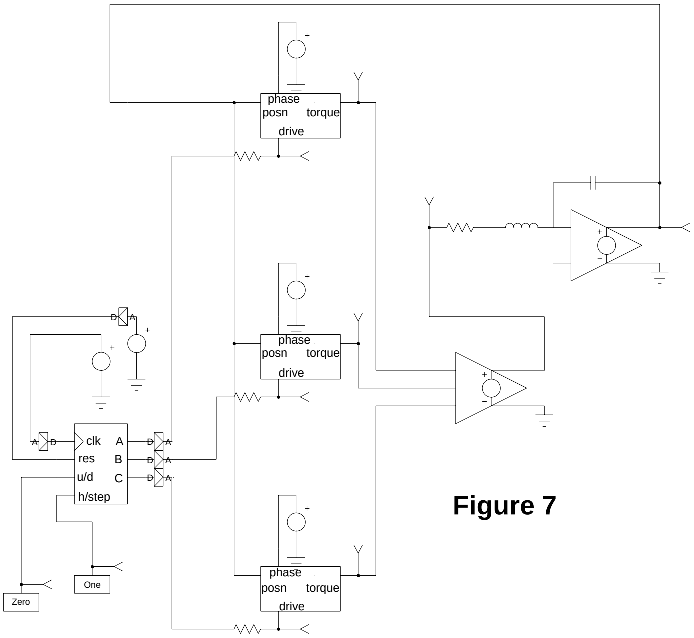


Figure 7

Fig 7 shows a complete three phase capacitance stepper motor, each actuator is represented by a sub-circuit symbol, the total torque is summed into the integrator of Fig 6 and the output angle is fed back to the actuators. The actuator drive voltages are provided from a state machine (see next page) programmable for full step or half step drive

Fig 8 shows the results obtained when driving the motor in full step and half step mode respectively. The ringing following each step is a consequence of the spring like characteristic of the torque curve. The lower frequency ringing on half steps is a result of the reduced slope of the torque/position characteristic with two phases energised.

The completed simulation model can now be used to investigate such features as the maximum stepping rates achievable, the importance of friction in damping the ringing and the effects of more complicated drive strategies such as micro-stepping.

Variable reluctance motors can be treated in a similar way though inductance in such motors is a function of both rotor position and excitation due to the non-linearities of the magnetic core.

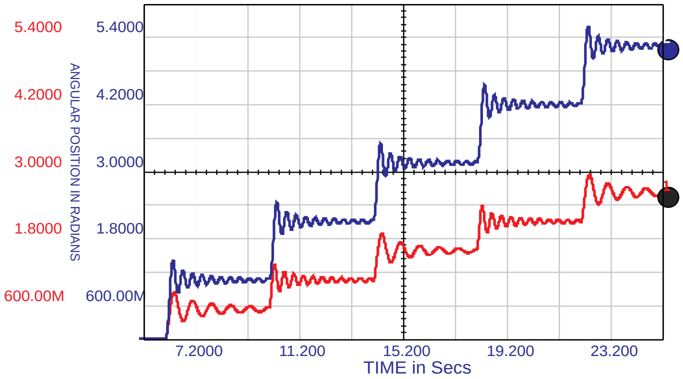


Figure 8

The motor model can be made a single sub-circuit and represented by a symbol. Some simulation packages provide such “ready wrapped” models but it is important for designers to understand the workings of these models in order to account for their limitations. The model developed here, for example, does not include voltage breakdown, more torque can always be obtained by increasing the drive voltage.

Full details of the SPICE circuits used in this model can be obtained from Mike Penberth at:

Technology Sources Ltd
Phone: +44-1638-561460
Fax: +44-1638-561721
E-mail: info@softsim.com

Test Program Development and Failure Analysis for SMPS

Authors
Larry Meares, Intusoft
Harry Dill, Consultant, Phoenix, AZ
Charles Hymowitz, Intusoft
Kyle Bratton, Chris Sparr, Naval Aviation Depot, MCAS Cherry Point, NC
Lloyd Pitzen, CCI Incorporated, Arlington VA

Automating Analog Test Design

Lawrence G. Meares, Intusoft, San Pedro California USA
email info@intusoft.com, http://www.intusoft.com

Abstract
A method of performing analog/mixed signal test will be described based on the assumption that:

1. All failure modes do not have to be defined.
2. The majority of failures are described by catastrophic faults [1-3].

Working forward from these assumptions, a comprehensive method will be presented that results in identifying a test suite that can serve either as a product acceptance test or a field test for fault isolation and repair. The resulting test suites are characterized by their percent fault coverage and number of replaceable components required for repair. The role of tolerance faults, correlated process faults and traditional IC defects are folded into the test methodology.

Introduction

Historically, product acceptance testing has stopped at specification compliance testing. While quality and safety considerations suggest that more testing is required, product cost has driven the testing toward a minimal solution. The ubiquitous desktop computer has relentlessly driven down the cost of design and test. It is now possible to use widely available Electronic Design Automation (EDA) tools to provide the specification for additional tests at a reasonable cost.

This paper will illustrate, using existing EDA technology, how it is possible to apply long-established techniques, which were originally developed for the aerospace industry, to define faults and sequence measurements in order to produce an optimal test sequence. Most tests require the design engineer to establish pass/fail limits. Methods are discussed to set these limits and to create tests that are robust, that is, tests which aren't likely to give incorrect pass/fail indications. Then a method will be described to establish optimal test sequences that will either maximize the likelihood of fault isolation, or minimize the effort that is required in order to demonstrate fault-free performance.

Defining Faults

What is a failure? A good design can accept part tolerances that are far wider than the tolerance of the individual components. For example, a 2k pull-up resistor may work just as well as the specified 1k resistor. Clearly, we want to accept the out-of-tolerance part in order to take advantage of the increased yield which is provided by a robust design. Production engineers should be able to substitute parts based on cost and availability.

In the pull-up resistor example, an open resistor could actually pass a test which is based upon functional requirements, and fail in the next higher assembly when the noise environment increases.

It's reasonable to conclude that we want to detect and reject products which contain catastrophic component failures; but accept products which may contain parametric "failures" that do not affect functional performance. Actually, these parametric failures are part of the tolerance distribution of the parts which we are using. Monte Carlo analysis will show the robustness of the design when we compare the resulting performance predictions with the product specification.

Defining failure modes: Failures are well characterized by a finite number of catastrophic failure modes. For digital circuits, there are 2 failure modes; the outputs may be stuck at logic one, or stuck at logic zero. Film and composition resistors are characterized by open circuit failures. The US Navy characterizes the catastrophic failures for many common parts for its Consolidated Automated Support System [3] and are the default failure modes used by several EDA vendors.

You may define failure modes using common sense experience, historical data or physical analysis. Depending on your application, you might want to consider additional failures; for example, IC bridging or interconnect failures which result in shorts between devices. In a PWB design, you may wish to simulate high resistance "finger print shorts" which are caused by improper handling of sensitive components.

Abstracting subassembly failure modes to the next higher assembly is a common error. For example, consider the case of an IC op-amp. Failures detected and rejected by the op-amp foundry are primarily caused by silicon defects. Once eliminated, these defects won't reappear. At the next higher assembly, failures may be caused by electrical stress due to static electricity, operator error in component testing, or environmental stress in manufacturing. Therefore, a new set of failure modes at the next production level is required. Again, we can usually describe these failure modes in terms of catastrophic events at the device or assembly interface, e.g. open, short or stuck failure modes for each interface connection.

Unusual failure modes are infrequent: Unusual failure modes get a lot of attention. For example, it is unusual to find a PNP transistor die in an NPN JANTXV package. Although these things do happen, they are very rare. If acceptance testing detects only 99% of all failed parts, then the quality of the product increases 100 fold after these tests are performed. For many products, the increased quality guarantees that products with undetected failures will not be delivered to the customer.

Detecting process faults: Process faults could cause the shift of many parameters simultaneously. When this is a consideration, as is usually the case for Ic's, the process parameters are monitored separately. If the process fails, the unit is rejected before the acceptance test is performed. Therefore, acceptance testing does not need to account for multiple parametric failure modes.

Setting Test Limits

Test limits are established using information from the product specification, simulation, analysis, laboratory testing and instrumentation capability. When parts fail or age, the measurement value for a given test may be near the test limit. Variations in circuit parameters, environment and test equipment can migrate these results across the test limit, thereby invalidating the test conclusion. Figure 1 shows why this happens when failed parts are included in the Unit Under Test, UUT.

By setting test limits suitably wide, the movement from pass to fail can be eliminated. In most cases, there isn't a product requirement which sets each test limit; test limits are determined via analysis, simulation or lab testing. Here are a few tricks we use:

- Include a liberal input voltage tolerance for power sources
- Perform a Monte Carlo analysis; include test set tolerances
- Vary ambient temperature +- 20 degrees celsius

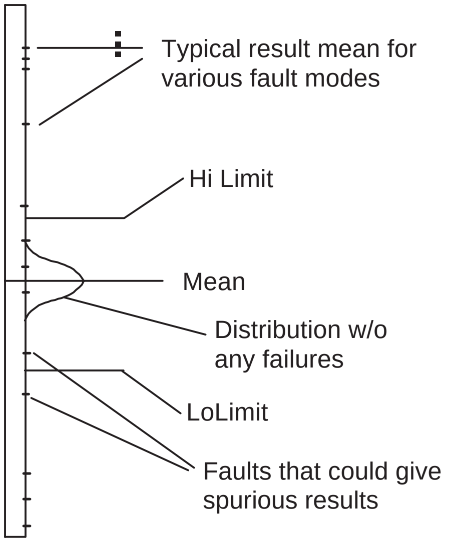

Typical result mean for various fault modes
Hi Limit
Mean
Distribution w/o any failures
LoLimit
Faults that could give spurious results

Figure 1, Failed parts can result in measurements that are near the measurement limits for normal operation, leading to false conclusions if the test is used for diagnosing those nearby faults.

After subjecting the circuit to these simulation environments, expand the measurement tolerances so that the UUT passes all of the above tests with a liberal margin (5 sigma or higher). Remember: any lack of knowledge about component models, test equipment or environment will ALWAYS necessitate a larger test limit. Increasing the limit accounts for unforeseen conditions. Movement of nearby failures across the limit, from fail to pass are handled by rejecting tests that could fail in this manner.

Seting Alarm Conditions

Another kind of simulation limit is the stress limit or alarm condition which is produced by some EDA tools [6]. When a part fails, it is frequently possible to overstress other parts, or cause an undesired circuit condition. It is recommended that power and/or current limits are set to the manufacturer's limit before part failure mode simulations are performed. In many instances, there are product safety issues such as firing an airbag or launching a rocket; these events must be prevented. Since these conditions are caused by special circuit states, you must devise measurements in order to catch these problems and group them with the stress alarms. The line between acceptance testing, built-in test and product quality begins to blur as we consider safety issues. For example, not only do we want to prevent unnecessary damage during an acceptance test, we also want to eliminate or detect safety hazards.

Resolving Alarms

Part failure modes which cause stress alarms should be detected as soon as possible. Tests can be sequenced in a manner which reveals these destructive failure modes early on, and therefore avoid the possibility of damage to any other circuit parts. Safe-to-start tests are among the tests which can reveal potentially damaging failure modes, e.g. shorts from power nodes to ground. After performing the tests which detect destructive failure modes, tests that could otherwise cause damage (if one of the destructive failure modes were present) can then be run. Using EDA tools for simulation is an ideal method for understanding circuit operation and specifying the new test limits.

Organizing a Test Sequence

Some test ordering has been described, based upon resolving stress alarms and simulation failures. The next level of ordering is by test difficulty or cost. Simple and inexpensive tests are performed first. Tests which use the same setup are grouped together. Tests which validate product performance are performed after eliminating the overstories failure modes. Product performance tests should not be used for fault detection because the tolerances are not set with failure modes in mind; they should only lead to a pass or out-of-tolerance conclusion.

Building a Test Fault-Tree

Tests are used to detect faults using the logic which is illustrated in Figure 2. Each test has one input and 2 outputs [2],[4]. The input contains a list of failure modes, and the test performs the logic which is necessary in order to classify the outcome as pass or fail. Each outcome has a list of failure modes that can be passed-on to successive tests. The process of selecting the best test in each ordered group results in a binary fault tree. After selecting a test, successive tests are placed on the pass and fail nodes of the tree until no more useful information is gained. When a new test configuration is selected, it may be possible to further isolate failure modes by expanding the tree from each node that has more than one fault. We will develop these concepts in considerable detail.

Since the world of simulation has more resources, that is, more observable information and more elaborate test capability, the job is generally acceptable at this point, although consuming excessive resources. We can gradually eliminate the most expensive test configurations, measurements and test points until the overall test objectives are met. These objectives are generally stated as a percent fault detection and if fault isolation is required, some goals regarding replaceable group size and distribution. Of course, if the objectives weren't met, you need to devise more tests, or if all else fails -- change the objectives.

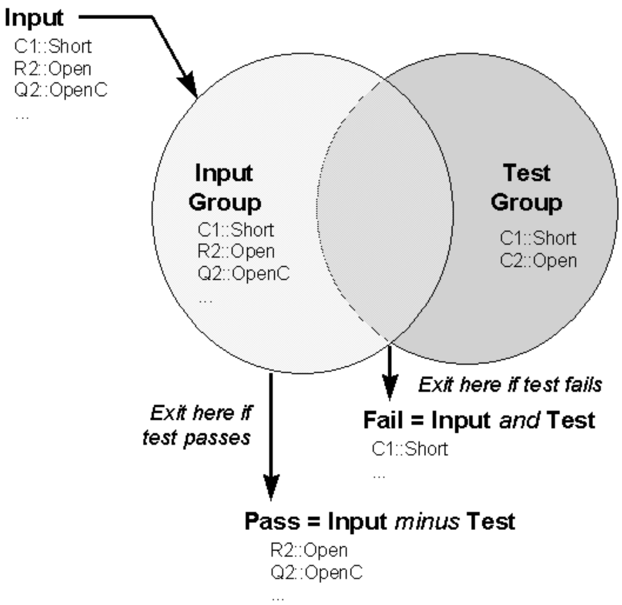


Figure 2: Test definition showing the logic used to sequence tests.

Test Synthesis

Next, we lay out a method for test synthesis that is based on standards developed for the aerospace industry [1,4,5]. Before proceeding, several concepts are required to understand the synthesis procedure.

Ambiguity Group: An ambiguity group is simply a list of failure modes. Since the failure mode references a part, all of a parts properties are implicitly included. The most important for this discussion is the failure weight.

Failure Weight: The failure weight of a part failure mode is an arbitrary number proportional to the failure rate; it is set to 1.0 for each default failure mode. The failure weight will be used to grade a test. Using these weights, each ambiguity group can be assigned a probability which is the sum of all failure weights. A selection procedure will be developed that favors tests that split these cumulative weights equally between their pass and fail outcomes.

No Fault Weight : When we begin a test, there exists an input ambiguity group that contains all of the failure modes we will consider, it's the fault universe for the UUT. It is useful to add a dummy failure to this universe, the no fault probability. The no fault probability will change depending on where, in the product life cycle, the test is being performed. An initial production test should have a high yield so we use a high no fault weight. If the product is returned from the field, it's expected to be faulty so that the no fault weigh is low. For built-in tests, the no fault weight depends on the accumulated failure rate. It turns out that the no fault weight will change the grade for tests in the "go" line; that is, the product acceptance test list. The no fault weight will tend to reduce the number of tests needed to arrive at the expected conclusion.

Test Definition

What's a test? In order to compare one test with another there must be a precise definition of a test.

A test consists of a comparison of a resultant value against limits that classify the UUT as good (pass) or bad (fail).

The resultant value can be the combination of one or more measurements and calculations. Each test has an associated ambiguity group. At the beginning of the test, there isa set of failure modes that have not been tested; these are defined by the Input Ambiguity Group. The test itself is capable of detecting a number of failure modes, these modes are grouped into a set called the Test Ambiguity Group. The pass and fail outcomes, then carry a set of failure modes that are the result of logical operations carried out between the Input Ambiguity Group and the Test Ambiguity Group such that:

Fail Ambiguity Group = Input Ambiguity Group AND Test Ambiguity Group
Pass Ambiguity Group = Input Ambiguity Group MINUS Fail Ambiguity Group

where AND represent the intersection of the lists, and MINUS removes the elements of one group from the other. Using MINUS here is a convenient way of avoiding the definition of NOT (Input…), since we really aren't interested in a universe that's greater than the union of the Input and Test groups. Figure 2 illustrates this logic.

The fail or pass outcomes can then be the input for further tests. If further tests are only connected to the pass output, then a product acceptance test is created. If tests are connected to both the pass and fail outcomes, then a fault tree is created which isolates faults for failed products. In either case, the object of each test is to reduce the size of output ambiguity groups. When no tests can be selected to further reduce these ambiguity group sizes, the test design will have been completed.

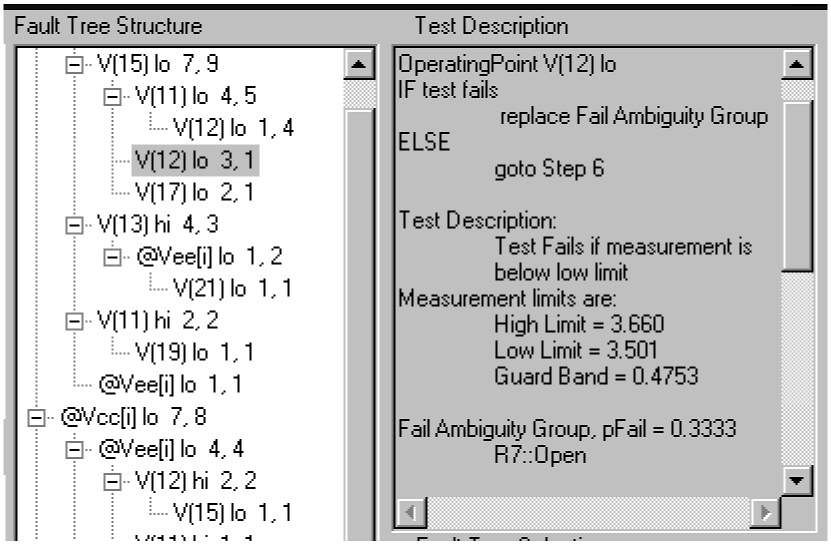
Test Strategy

The strategy used to select test and test sequences is as follows:

> To arrive at a conclusion; that is, the smallest ambiguity group using the least number of tests.
>
> The least number of tests is actually the smallest mean number of tests to reach the terminal nodes of the diagnostic fault tree. The fault tree is made by interconnecting the tests, illustrated here, in a binary tree.

The Best Test

Without performing an exhaustive search, the best test tends to be the test with the highest entropy. Variations on the exhaustive search technique, such as looking ahead one test, rarely produce better results. Exhaustive search has been shown to consume computational resources so rapidly that it is not a viable method.



```
Fault Tree Structure
  ⊟ V(15) lo  7, 9
    ⊟ V(11) lo  4, 5
        V(12) lo  1, 4
      V(12) lo  3, 1
      V(17) lo  2, 1
  ⊟ V(13) hi  4, 3
    ⊟ @Vee[i] lo  1, 2
        V(21) lo  1, 1
  ⊟ V(11) hi  2, 2
      V(19) lo  1, 1
    @Vee[i] lo  1, 1
⊟ @Vcc[i] lo  7, 8
  ⊟ @Vee[i] lo  4, 4
    ⊟ V(12) hi  2, 2
        V(15) lo  1, 1
```

```
Test Description
OperatingPoint V(12) lo
IF test fails
            replace Fail Ambiguity Group
ELSE
            goto Step 6

Test Description:
            Test Fails if measurement is
            below low limit
Measurement limits are:
            High Limit = 3.660
            Low Limit = 3.501
            Guard Band = 0.4753

Fail Ambiguity Group, pFail = 0.3333
            R7::Open
```

Selecting the best test: A terminal conclusion is defined as the pass or fail conclusion for which no more tests can be found. Then if "no fault" is present, then it is the product acceptance test result, with all remaining faults being the ones that are undetectable. Otherwise the parts in the resulting Ambiguity Group would be replaced to repair the UUT.

In general, our goal is to reach the terminal pass-fail conclusions by performing the fewest numbers of tests to reach each conclusion. The general solution to this problem using an exhaustive search technique expands too rapidly to find a solution during the lifetime of our universe[4]. Several heuristic approaches are possible [5], one of which follows.

If we take the idea of failure modes one step further, we can give each failure mode a failure weight that is proportional to the failure rate. To avoid looking up failure rates, we can default these weights to 1.0, and some time later fill in a more precise number. For each test candidate, we can compute the probability of a pass outcome and a fail outcome. From a local point of view, the summation of the pass and fail probabilities must be unity; that is, the UUT either passes or fails a particular test. Borrowing from information theory, we can compute the test entropy using the following equations [4],[5]].

Entropy = -q ∗ log(q) - p ∗ log(p)
where p and q are the pass and fail probabilities
and:
p = Σ pass weights / (Σ pass weights + Σ fail weights)
q = Σ fail weights / (Σ pass weights + Σ fail weights)

The highest entropy test contains the most information. We select the best test as the test having the highest entropy. For the case when failure weights are defaulted to unity, this method will tend to divide the number of input failures into 2 equal groups. Since no fault can only be in the pass group, a high no fault weight will steer the tests through the pass leg fastest; making the best product acceptance test. The rationale for a high no fault probability is the expectation that most units will pass the production acceptance test, a condition of an efficient and profitable business. If, on the other hand, we want to test a product that is broken, we would give the no fault probability a lower value. Then the test tree would be different, having a tendency to isolate faults with fewer tests.

Robust Tests

Tolerances can cause measurement results to migrate across the test limit boundary. As a result, a fault could be classified as good or a good part could be classified as a failure. Tolerances include:

UUT part tolerances, computer model accuracy, measurement tolerances, UUT noise, test set noise and fault prediction accuracy.

Previously; we showed that avoiding false test conclusions for tolerance failures requires setting the test limits as wide as possible. Now we will show how to set the limits as far away from expected failed results as possible. We will do this by a unique test selection procedure. But first, to compare one test with another, we need to define a measure of test robustness.

Guard Band *(the fudge factor)*

A test measurement for a good UUT has a range of values that define acceptable performance which we call a tolerance band.

> The measure of test robustness with respect to a failure mode is then the distance between the failed measurement result and the nearest test limit divided by the tolerance band. We call this value a guard band.

The test limit can be safely placed in the guard band as long as no other faults have results in this band. Normalizing all measurements using their tolerance band allows us to compare guard bands of different tests. We can then modify the entropy selection method to reject tests with small guard bands. Figure 4 shows how this works.

In this example we have 2 operating point tests. Failure modes are identified as NoFault,F!, F2, …F6. Assume test A is performed first, and test B is performed on the pass group of test A as shown in figure 5.. Test A divides the failures into a pass group containing (F4,F5,F6) and a fail group containing (F1,F2,F3) . Connecting test B to the test A pass outcome eliminates F1 from the test B failure input and the guard band for test B extends from the hi limit to F4. If test B were done first the guard band would be smaller, from the test B hi limit to F1.

An incorrect test outcome will invalidate subsequent test decisions. In order to be right most often, tests with large guard bands should be performed first because they are less likely to be wrong. Moreover, tests which were previously rejected may turn out to be excellent tests later in the sequence (as illustrated in the example). Tests with small guard bands simply shouldn't be used.

While a model of the statistical distribution was shown in figure 4, you should be aware that there usually isn't sufficient information to have a complete knowledge of the statistics of a measurement result. In particular the mean is frequently offset because of model errors; for example, a circuit that is operating from different power supply voltages than was expected. The statistics of failed measurements are even less certain because the failure mode and the failed circuit states are less accurately predicted. It is therefore necessary to increase the tolerance band as much as possible. We avoided saying exactly where in the guard band the measurement limit should be placed because it's a judgment call, depending on how much the tolerance band was widened and on the quality of the fault predication.

Summary

We have shown how to use EDA tools with various test configurations to describe normal and failed circuit behavior. Methods for establishing test tolerances for circuits without failed parts were presented. Assuming that failures mechanisms are reasonably well known, a method to recognize the existence of a single fault and further diagnose which part was responsible was developed. All of these procedures can be closely integrated in existing EDA tools to provide a cost effective method for test design [6].

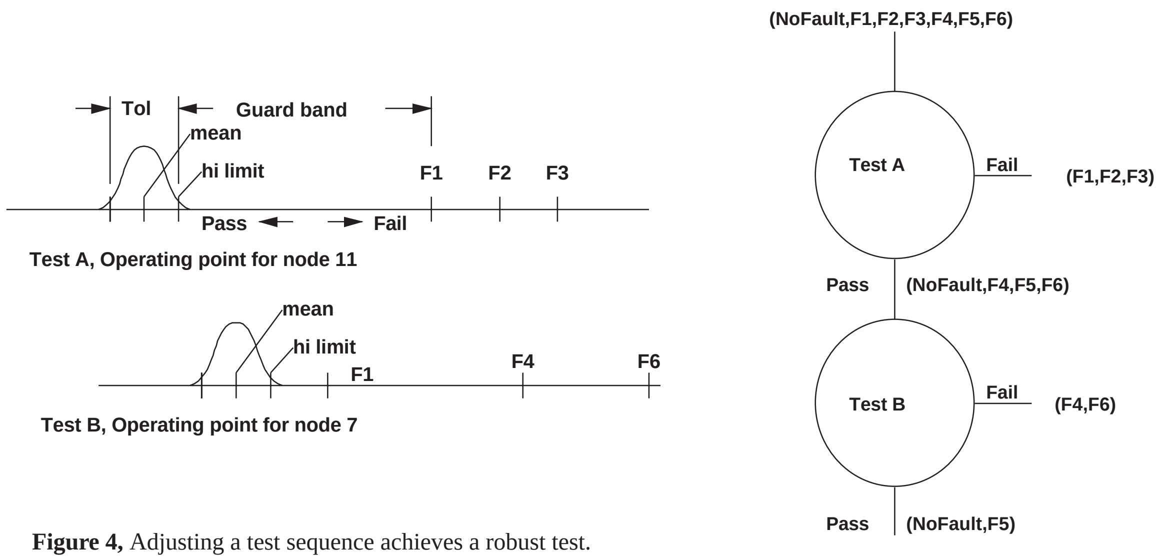


Figure 4, Adjusting a test sequence achieves a robust test.

References

[1] Cheng-Ping Wand and Chin-Long Wey, “Efficient Testability Design Methodologies for Analog/Mixed Signal Integrated Circuits”, 1997,3rd International Mixed Signal Testing Workshop, pp 68-74

[2] H. Dill et. al., “Application of Analog & Mixed Signal Simulation Techniques to the Synthesis and Sequencing of Diagnostic Tests”, 1977 Autotestcon Proceedings, pp425-434

[3] CASS Red Team Package data item DI-ATTS-80285B, Fig. 1 -SRA/SRU Fault Accountability Matrix Table, pg. 11

[4] W.R. Simpson and J.W. Sheppard, “Fault Isolation in an Integrated Diagnostic Environment, IEEE D&T, vol. 10 No. 1, March 1993

[5] Sharon Goodal, “Analog/Mixed Signal Fault Diagnosis Algorithm and Review”, 1994 AutoTestCon Proceedings, pp 351-359

[6] Test Designer, A software program from Intusoft, 1998.

Simulation measurements vs. Real world test equipment

There are fundamental differences between simulation measurement and test equipment measurements in accuracy, capability and measurement strategy. These differences will influence the way you map the simulation tests and measurements into real world test sets and measurement equipment. The following summary describes the most important differences:

Transient Simulation Accuracy:
Simulation accuracy is frequently no better than 1% to 3%; about what you expect from an oscilloscope. Accuracy limitations occur because of model accuracy, numerical errors and the user defined topology description. The implicit topology description is probably the largest source of error. Interconnect imperfections and parasitic coupling are assumed absent. You must explicitly include these effects. The designer usually adds these models sparingly because they are difficult to describe and may adversely impact the simulation. Fortunately, most parasitic elements have no important effects on a design; the trick is to identify and model the “important” ones.

Model Range:
Component models are made to be accurate in the neighborhood of a components expected useful operating point. Part failures will often place other components in an abnormal operating condition, one for which their model does not apply. To mitigate these problems, current limited power supplies should be used in the simulation. Moreover, when a simulation fails because the simulator cannot converge; it should be a warning that the failure mode cannot be studied with that particular test. It will be necessary to isolate the failure before running that particular test so that the failure mode does not need to be addressed anymore.

Visual Tests:
For items returned from the field, some component failures could have caused catastrophic damage that can be identified by visual inspection. Don’t test a smoked board!

Multimeter Tests:
Consider running “dead” circuit tests to isolate failures that result in simulation convergence failures. These tests are run without the power supplies and can be used to find shorted or opened power components. Isolation of open or shorted bypass capacitors can be done in this manner using an AC multimeter.

Measurement Range:
You don’t need to set the range or limits. There is no simulation range limitation. You automatically get IEEE double precision results.

Units:
Units are always converted to standard engineering postfix notation; for example 1.23e-9 volts will be shown as 1.23n volts. You may also use this notation for parameter input.

Coupling:
There is no need to worry about AC coupling when making measurements. You can always subtract the average value. In fact the stdDev function does just that. The simulator will work just fine at offsets of thousands of volts and your virtual instrument will not be overloaded.

Current vs. Voltage:
Current and voltage are properties of the quantity being measured. There is no need to select a current or voltage meter in the simulation.

Measurement Time:
Simulation time is a valuable resource. Measurement of period and frequency can be done on a much smaller data sample since there is no noise. Frequency measurement includes the specification of the number of events to be included so precise timing “gate” can be set in order to eliminate quantizing effects. Remember to let the circuit settle to its steady state condition before making measurements that assume the circuit has be running awhile.

AC Simulation Accuracy:

Frequency domain measurements made from AC analysis data are frequently more accurate than can be achieved with test equipment. The reason for this is that there is no noise to corrupt measurement accuracy and the numerical equations are solved exactly, rather than using the iteration procedure used in the transient simulation. For example; highly accurate estimates of oscillator frequency can be made through phase measurements. These would then map into counter/timer based measurements for the automated test.

Noise:

Thermal noise must be explicitly modeled, otherwise it is absent. This allows measurements to be made accurately without averaging them over many cycles. There is generally no benefit to adding noise unless the noise influences the value of the parameter you want to measure; for example, the lock time of a phase locked loop.

Numerical Artifacts:

Numerical artifacts are sometimes encountered in the Transient Simulation. You will have to pay attention to simulation settings such as RELTOL and integration METHOD in the simulation setup that have no counterpart in real world test equipment.

Fault Tree Design:

Simulators accumulate large chunks of data for each simulation run. These datum can be thought of as a state vector; representing the UUT state under some setup condition. These states then become a number of conventional tests; where a test is a measured value compared to some set of limits. The difficulty in acquiring data for the simulator, and probably for the hardware test are equivalent for each test. It is therefore convenient to lump all of these simulation measurements together when designing a fault tree; if you need to separate these measurement groups, you can name them differently and separate test vectors between these new measurement groups. Test sequencing is then performed by selecting the groups for a sequence, the test mode (binary, tertiary, …) and either manually or automatically completing the fault tree using the selected groups and mode.

New Techniques for Fault Diagnosis and Isolation of Switched Mode Power Supplies

C.E. Hymowitz, L.G. Meares, B. Halal
Intusoft P.O. Box 710, San Pedro CA 90733-0710, info@intusoft.com

Abstract - This paper describes new software techniques to perform analysis, diagnosis, and isolation of failures in analog and mixed-signal circuits including switched mode power supplies. Unique methods and algorithms for schematic entry, setting of failure characteristics, definition of test strategies, recording of simulation based measurements, creation of fault trees and sequencing tests are all discussed. To ensure a realistic test of the new software techniques, diagnostics were developed for a moderately complex analog and mixed signal switched mode power supply. A discussion of some of the drawbacks of sensitivity based failure analysis techniques is also included.

I. INTRODUCTION

In late 1996, the SPICE simulation tool manufacturer Intusoft initiated development of a new product, tailored to the unique and demanding needs of the test engineer. This product, called Test Designer, provides an effective, interactive design environment for the synthesis of diagnostic tests, generation of fault dictionaries and the building of diagnostic fault trees. Since the category of analog and mixed-signal test synthesis and sequencing software is relatively new, a number of unique techniques were developed to solve key parts of the FMEA (failure mode effects analysis) and test program set design process.

The motivations for improving the analog and mixed-signal test set design and failure analysis process are plentiful and well documented [1-5]. For instance, identification of early production faults, improved safety and reliability through the analysis of difficult to test failures, investigation of power supply failure mechanisms such as power switch overcurrent, FET gate over-voltage, start-up failures, and excessive component stress could all benefit from improved simulation software. Yet little dedicated software currently exists to help streamline and organize analog and mixed-signal circuit test procedures.

Most testing is done to assure product performance standards. UL standards for various types of supplies require that tests be developed for overload protection circuitry, short circuit tests, and breakdown of components[6]. In some cases, when circuit analysis indicates that no other component or portion of the circuit is seriously overloaded as a result of the assumed open circuiting or short circuiting of another component, some tests can even be bypassed.

The software can add benefits in 2 other ways. First, if it can identify failure modes that aren't tested, you will have found either unnecessary parts or a flaw in the acceptance test. Either case would improve the quality of the product. Another important aspect is the tracking of component quality during the production lifetime. Frequently, a supplier's product will evolve or the supplier will be changed, and the product's performance will drift away from its design center. This drift itself is observable from the acceptance test results, but the software also allows you to track the nearby failures, including parametric failures. These, of course, aren't really failures in the sense that the product can't be shipped; rather, they are component quality indicators.

The software, outlined in Figure 1, provides the aforementioned benefits and includes a complete system capable of design entry, simulation, analysis, and test synthesis and fault tree sequencing. The schematic entry program is specially enhanced to hold the entire design database including part and model values and tolerances, and all part failure modes. It also contains a description of the various test configurations and measurements for each test. Using Object linking and embedding (OLE) communication, the schematic builds the required netlist for IsSpice4, a SPICE 3/XSPICE based

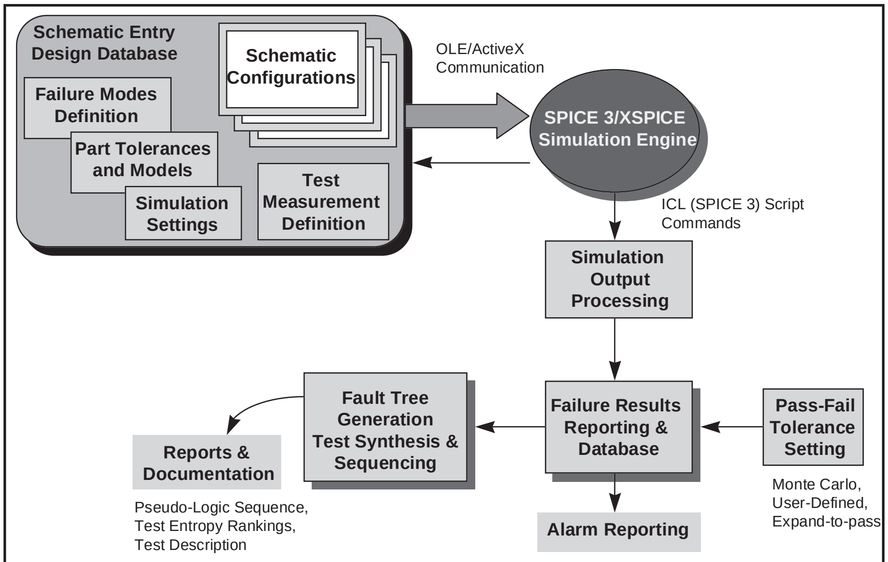

Schematic Entry
Design Database
Schematic Configurations
Failure Modes Definition
Part Tolerances and Models
Simulation Settings
Test Measurement Definition
OLE/ActiveX Communication
SPICE 3/XSPICE Simulation Engine
ICL (SPICE 3) Script Commands
Simulation Output Processing
Fault Tree Generation Test Synthesis & Sequencing
Failure Results Reporting & Database
Pass-Fail Tolerance Setting
Monte Carlo, User-Defined, Expand-to-pass
Reports & Documentation
Pseudo-Logic Sequence, Test Entropy Rankings, Test Description
Alarm Reporting

Figure 1, A block diagram of the software system implemented to provide automated failure analysis and test program set development.

analog and mixed signal simulator. The simulation output data is processed using the Berkeley SPICE Interactive Command Language (ICL) in order to extract the desired measurements. This process is continued automatically until all of the faults are simulated. The measurements are then parsed into various report forms which contain the pass-fail tolerances. Finally, the tests are sequenced into a fault tree.

A six step process, described in detail below, is utilized for the development of fault diagnostics. In chronological order, the steps are:

- Design Entry including: Schematic Layers setup, Schematic Configurations setup, Simulation Directives setup, Failure modes characteristics definition, Test Configuration definition
- Measurement definition
- Pass-Fail tolerance setting
- Fault simulation
- Results reporting
- Failure states and sequencing

II. CONFIGURABLE SCHEMATICS

A long-standing problem in electrical and mechanical circuit design has been the conflict between the needs of the designer and the needs of production and manufacturing. The main method of conveying the designer's creation is the circuit schematic which is used to describe the behavior of the circuit as well as the details of how production will build the hardware.

The designer is concerned with creating a circuit that meets specifications. This is done chiefly through various EDA tools but, mainly with circuit simulation. The designer must build multiple test configurations, add parasitic components and stimuli, and even include system elements in the simulation. A top-down design methodology, where different levels of abstraction are inserted for different components, is commonplace. Modeling electrical behavior often results in different representations for different test configurations. In general, the schematic becomes so cluttered with circuitry and data, that it must be redrawn for production, greatly raising the probability of a transcription error.

The need for a reconfigurable schematic capability becomes even more mandatory when we analyze the needs of the failure analysis and test program development engineer. In order to be effective, the simulation process can not become burdened with the bookkeeping intricacies of multiple schematic variations and analysis specifications. The designer must have a way to connect various stimuli and loads to core circuitry and to group the desired SPICE analyses and test measurements with each schematic configuration.

Until now, the best approach has been to hide these special configurations in subcircuits; for example a resistor's parasitic capacitance could be included in a subcircuit. While this approach works for hierarchical schematic entry and extending individual component models, it doesn't solve the problem of adding test equipment, different stimulus inputs, or dealing with multiple simulation scenarios.

A test setup provides loads, voltage and current stimuli and instrumentation connections at specific points on the Unit Under Test (UUT). When viewed in a broader context, the combination of the test setup circuitry and the UUT can be considered to be a circuit configuration in and of itself. Indeed, for simulation purposes, the test setup circuitry must be included as part of the circuit. Most Test Program Sets (TPSs) implement multiple setups during the testing sequence. This increases the simulation burden by requiring a separate schematic for every test setup.

The system described by Figure 2 addresses the multiple test setup problem with a unique solution. It allows the user to assign each setup/UUT combination a different configuration name and simulates all of the stand-alone configurations in a batch operation. The setup/UUT combination is called a "circuit configuration". The circuit configuration is defined during the schematic entry process. Every circuit configuration is composed of one or more schematic layers. An active layer can be thought of as a transparency that over-

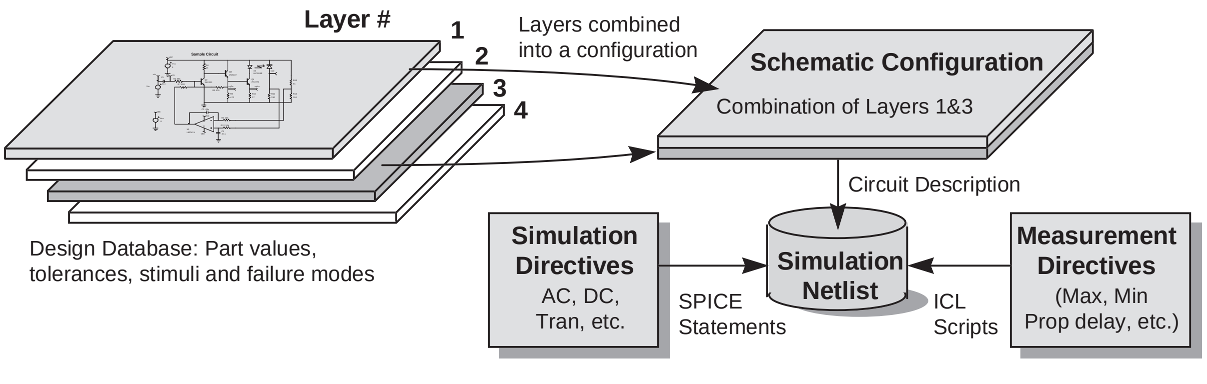


Figure 2, A unique reconfigurable schematic program allows different schematic layers to be combined in order to create various circuit configurations. Simulation and measurement directives are then added to create multiple tests descriptions.

lays other transparencies such that as you view them, you see the complete circuit configuration schematic. Circuit nodes on the top layer connect with nodes on underlying layers as if the drawing were created on a single page. The schematic allows mixing and matching of layers to form the required circuit configurations. Any circuitry, elements, or documentation can be placed on any layer.

Use of a layered concept in itself is not unique. It is generally used as a drawing management feature to remove complexity from the user's visual field, rather than making a multiplicity of configurations. While PCB layout software has had a similar feature for quite some time, a configurable schematic has not been implemented (to the best of our knowledge). This is the first known graphical entry method which is capable of solving the Test - Simulation bridge using a reconfigurable layered schematic approach.

III. SIMULATION DIRECTIVES

The system also allows different sets of SPICE analysis statements to be grouped and stored (Figure 2). For instance, an operating point analysis can be specified to run along with a brief transient analysis. In another group, a frequency response can be run with a long transient analysis. The user can then pair any set of SPICE simulation directives with any circuit configuration to create a unique "Test Configuration". For example, a single circuit configuration can be assigned multiple types of simulation analyses in order to define multiple test configurations.

IV. FAILURE DEFINITION

Each component is defined by a set of nominal device and model parameter attributes, as well as parametric tolerances. Each component also has a set of associated failure modes. Initially, parts include the failure modes as defined in the Navy's CASS (Consolidated Automated Support System) Red Team Package[7]. Users can edit the predefined failure modes or add their own catastrophic or parametric failure modes.

Failure modes are simulated by programmatically generating the proper SPICE 3 syntax to describe the failure. Failure modes can be setup for primitive (resistor, transistor, etc.) as well as subcircuit macromodel-based elements. Any node on a part can be shorted to any other node, opened, or stuck. The stuck condition allows the user to attach a B element expression. The B element is the Berkeley SPICE 3 arbitrary dependent source which is capable of analog behavioral modeling[8,9]. Behavioral expressions can contain mathematical equations, If-Then-Else directives, or Boolean logic [10]. The expressions can refer to other quantities in the design such as nodes and currents, thus creating an unlimited fashion in which to "stuck" a node. A series of examples are shown in Table 1.

It should be noted that the software builds the B element expressions and SPICE models, inserts the required elements, and generates the SPICE netlist automatically. All of the failure mode definitions are carried out in a graphical manner. No script writing or programming is necessary in order to define or simulate a fault. There is no need for the user to write or know the required SPICE syntax (examples are shown in table 1). The characteristics (open/short/stuck resistance) of each failure mode can be defined by the user (Figure 3).

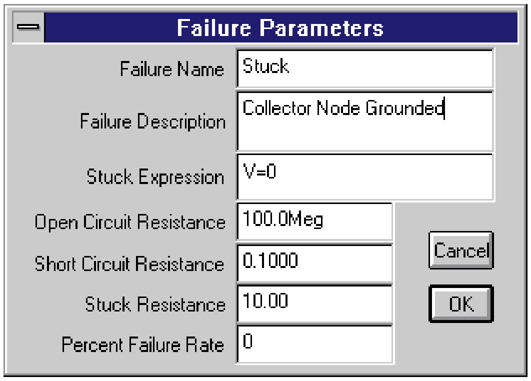
V. MEASUREMENT DEFINITION

In order to create a "test", the user must combine a circuit configuration with a set of SPICE simulation directives and a desired measurement which will be made on the resulting data. Therefore, the simulator, or some type of data post-processing program, must be included to extract and record information from each desired test point waveform. For the Test Designer software, the former was chosen and implemented using ICL which is available in SPICE 3 and Nutmeg.

The software uses the IsSpice4 simulation engine to perform the failure analysis. IsSpice4 is an enhanced version of Berkeley SPICE 3 [9] and XSPICE [11,12]. IsSpice4 includes and expands upon the standard Berkeley SPICE 3

Table 1, SPICE Syntax for Various Failure Modes

Fault	**Before Insertion**	**After Fault Insertion**
Shorted Base Emitter	Q1 12 19 24 QN2222A	Q1 12 19 24 QN2222A
		Rshort_19 19 24 .1
Open Resistor	R3 17 0 10K	R3 17_open 0 10K
		Ropen_17 17_open 17 100Meg
Low Beta	Q1 12 19 24 QN2222	Q1 12 19 24 Q1_Fail
Parametric fault	.MODEL QN2222 NPN AF=1 **BF=105**	.MODEL Q1_Fail NPN AF=1 **BF=10**
	BR=4 CJC=15.2P CJE=29.5P…	BR=4 CJC=15.2P CJE=29.5P…
Resistor Stuck	R1 6 0 1K	R1 6 0 1K
2V below Vcc		Rstuck_6 6_Stuck 6 10.00000
		Bstuck_6 6_Stuck 0 V= Vcc- 2
Timed Dependent	L2 3 0 62U	L2 3 0 62U
Inductor Fault		Rstuck_3 3_Stuck 3 10.00000
		Bstuck_3 3_Stuck 0 V= Time > 10n ? 0 : V(3)



Failure Parameters

Failure Name: Stuck

Failure Description: Collector Node Grounded

Stuck Expression: V=0

Open Circuit Resistance: 100.0Meg

Short Circuit Resistance: 0.1000

Stuck Resistance: 10.00

Percent Failure Rate: 0

Figure 3, An example of the dialog which is used to define the failure mode characteristics.

ICL. ICL is a set of commands that can direct SPICE to perform various operations such as running a particular analysis or changing a component value. The commands, which look and act like Visual Basic scripts, can be run interactively or in a batch mode. IsSpice4 contains new ICL commands which allow SPICE 3 plots, or sets of vectors (i.e. waveforms) to be scanned and measured with cursors. In contrast to traditional SPICE "dot" statements, ICL commands are performed in order, one at a time. This makes ICL scripts perfect for describing test procedures.

A "Wizard" approach is taken in order to alleviate the syntax headaches associated with script development. For example, a Cursor Wizard is employed to position one or more imaginary cursors on a waveform or set of waveforms. Y axis cursors are positioned with respect to a single waveform or vector, while X axis cursors are positioned with respect to an entire set of vectors. A Measurement Script Wizard is used to manipulate the data derived from the cursors in order to produce a single measurement.

A variety of functions are available for setting the cursor positions and for measuring the data in between the cursors. Two example scripts are shown in Table 2. As shown in figure 2, these measurement scripts are combined with traditional SPICE analysis directives and a test configuration description to form a simulatable IsSpice4 netlist.

VI . EXAMPLE

Now that we have defined how a design is setup, we can show how the software performs the simulation, and discuss the failure diagnostic and test sequencing process. This is best done through an example. The circuit shown in Figure 4 is a forward converter which uses the Unitrode UC1843 PWM and Magnetics MPP5812 core models from the Intusoft Power Supply Designer's Library. The start-up transient waveform (V(5), top right) is shown, along with a close-up view of the output ripple. Because the circuit uses a full nonlinear switch level PWM IC model and an accurate power Mosfet model, we can examine such detailed phenomenon as the Mosfet's switching characteristics, operating current into VCC, under voltage lockout threshold, and propagation delay. To simulate the start-up of the circuit, the power supplies V1 and V4 were ramped from zero to their final value, over a 100us interval.

Initially, the SMPS was simulated in full start-up mode for 1.2ms. The simulation runtime was 184.90s. It was decided that a shorter transient run could yield enough useful tests to detect the majority of the faults and be simulated more quickly. For the first failure analysis, a nominal transient analysis of .3ms in length was selected.

The selected measurements were the value at 50us, the maximum value, and the final value at .3ms. The maximum value will be useful for oscillating waveforms, while the final value will be useful for filtered and slow-changing waveforms. The 50us measurement can be used as a comparative measurement of start-up performance. The measurements were determined after looking at the nature of the waveforms and estimating which tests would best be able to detect significant differences when a fault is inserted.

For each of the three measurements, all of the available vectors (circuit test points including voltage current and power dissipation) were recorded. While not all vectors can normally be probed on a circuit card, it is important to gather all of the possible data. Later, when the test sequencing is performed, we can grade the usefulness of each measurement. By measuring all possible test points up front, we eliminate the need to perform subsequent simulations.

Group Delay Measurement	
Cursor Script	
HomeCursors	Reset cursor to waveform endpoints
Vsignal = db(???)	Use dB values
Vphase = PhaseExtend(ph(???))	Use phase values
theDelay = -differentiate(Vphase)/360	Differentiate the phase waveform
theMax = max(Vsignal)	Find the maximum
MoveCursorRight(0,Vsignal,theMax)	Move the left cursor to the maximum
Fmax = GetCursorX(0)	Store the frequency
SetCursor(1,Fmax)	Move the right cursor to the maximum
MoveCursorRight(1,Vsignal,theMax-3)	Move the right cursor to the upper -3dB point
MoveCursorLeft(0,Vsignal,theMax-3)	Move the left cursor to the lower -3dB point
Measurement Script	
groupDelay = mean(theDelay)	Store the measurement

Table 2, Example Wizard generated ICL scripts used to automate measurements of the failure analysis. The ??? fields are replaced with the desired vectors which will be processed.

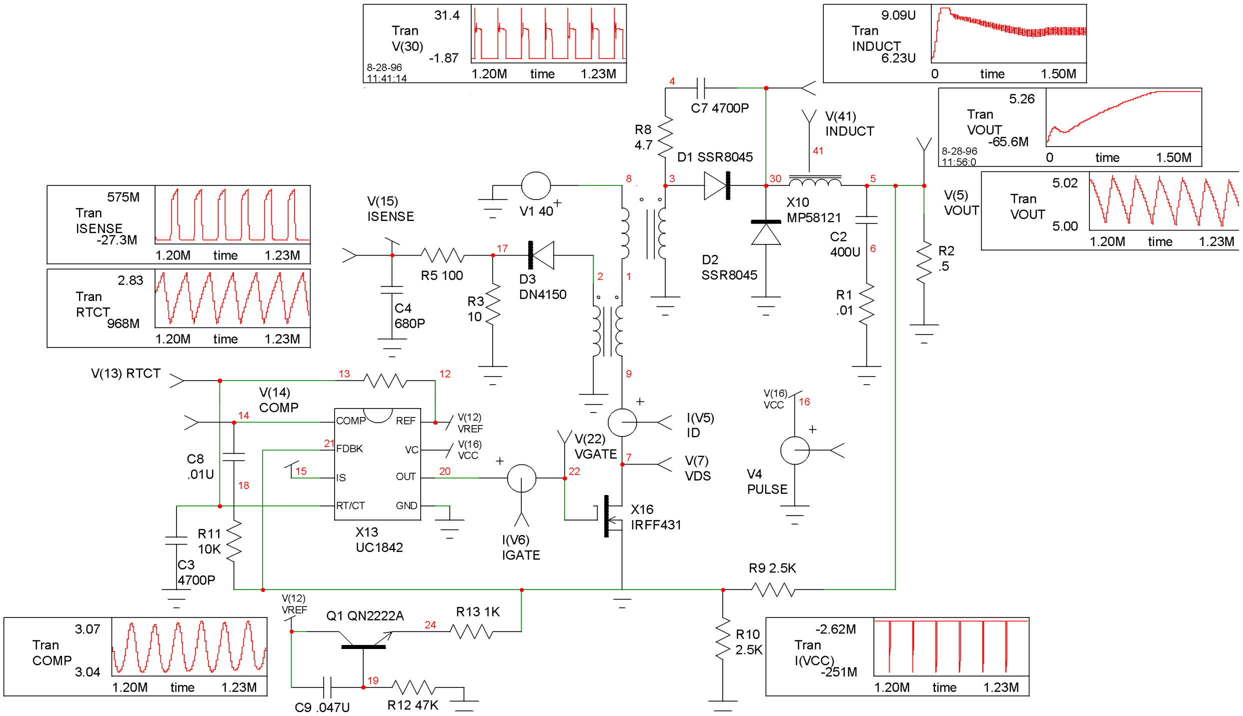
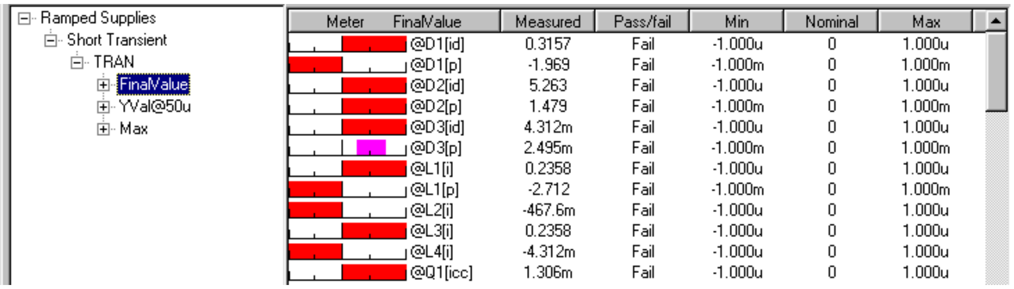


Figure 4, An SMPS forward converter example using the UC1843 PWM controller. Note the use of the nonlinear core (MPP) model. The test point in the center of X10 shows the inductance value as a function of time.

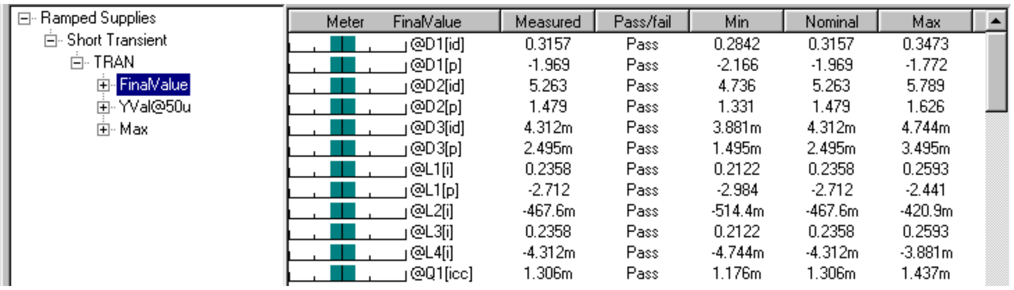
The initial results, shown in Figure 5, indicate that all of the tests fail, since no pass-fail tolerances have been applied. Figure 6 shows the results after applying default tolerances of ±100uV for voltages and ±1mA for currents.



- Ramped Supplies
  - Short Transient
    - TRAN
      - FinalValue
      - YVal@50u
      - Max

Meter	FinalValue	Measured	Pass/fail	Min	Nominal	Max
🟥	@D1[id]	0.3157	Fail	-1.000u	0	1.000u
🟥	@D1[p]	-1.969	Fail	-1.000m	0	1.000m
🟥	@D2[id]	5.263	Fail	-1.000u	0	1.000u
🟥	@D2[p]	1.479	Fail	-1.000m	0	1.000m
🟥	@D3[id]	4.312m	Fail	-1.000u	0	1.000u
🟪	@D3[p]	2.495m	Fail	-1.000m	0	1.000m
🟥	@L1[i]	0.2358	Fail	-1.000u	0	1.000u
🟥	@L1[p]	-2.712	Fail	-1.000m	0	1.000m
🟥	@L2[i]	-467.6m	Fail	-1.000u	0	1.000u
🟥	@L3[i]	0.2358	Fail	-1.000u	0	1.000u
🟥	@L4[i]	-4.312m	Fail	-1.000u	0	1.000u
🟥	@Q1[icc]	1.306m	Fail	-1.000u	0	1.000u

Figure 5, The results of an initial simulation before the default tolerances are applied to the measurements.



- Ramped Supplies
  - Short Transient
    - TRAN
      - FinalValue
      - YVal@50u
      - Max

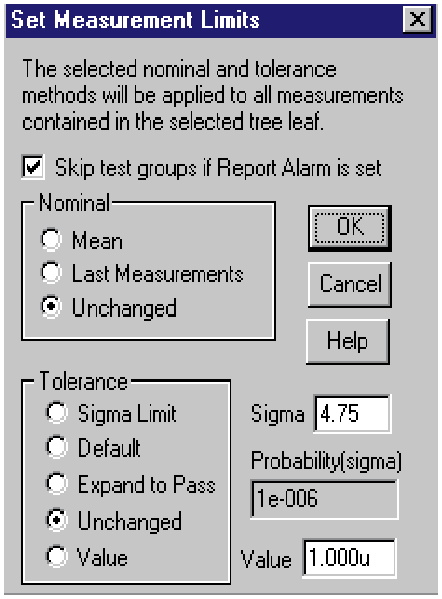
Meter	FinalValue	Measured	Pass/fail	Min	Nominal	Max
🟦	@D1[id]	0.3157	Pass	0.2842	0.3157	0.3473
🟦	@D1[p]	-1.969	Pass	-2.166	-1.969	-1.772
🟦	@D2[id]	5.263	Pass	4.736	5.263	5.789
🟦	@D2[p]	1.479	Pass	1.331	1.479	1.626
🟦	@D3[id]	4.312m	Pass	3.881m	4.312m	4.744m
🟦	@D3[p]	2.495m	Pass	1.495m	2.495m	3.495m
🟦	@L1[i]	0.2358	Pass	0.2122	0.2358	0.2593
🟦	@L1[p]	-2.712	Pass	-2.984	-2.712	-2.441
🟦	@L2[i]	-467.6m	Pass	-514.4m	-467.6m	-420.9m
🟦	@L3[i]	0.2358	Pass	0.2122	0.2358	0.2593
🟦	@L4[i]	-4.312m	Pass	-4.744m	-4.312m	-3.881m
🟦	@Q1[icc]	1.306m	Pass	1.176m	1.306m	1.437m

Figure 6, The results of an initial simulation after the default tolerances are applied to the measurements. The Results dialog shows the pass-fail status through the use of a unique histogram indicator (left side).

The results report shows the simulated (Measured column) value, whether the test passed or failed, and shows the minimum, nominal, and maximum values. A special histogram bar icon is used to provide a quick visual indication of the test status.

VII. SETTING PASS-FAIL TOLERANCES

A variety of methods are available for setting tolerances including hard limits, Monte Carlo analysis, and a unique “Expand to Pass” method (as shown in Figure 7). Expand to pass moves the min and max tolerance bands outward until the measurement is within the pass band. This allows tolerances to be set through different simulation scenarios such as high and low temperature, or high and low power supply voltage.



Set Measurement Limits

The selected nominal and tolerance methods will be applied to all measurements contained in the selected tree leaf.

☑ Skip test groups if Report Alarm is set

Nominal
○ Mean
○ Last Measurements
● Unchanged

Tolerance
○ Sigma Limit
○ Default
○ Expand to Pass
● Unchanged
○ Value

Sigma 4.75

Probability(sigma) 1e-006

Value 1.000u

Figure 7, The dialog which is used to set measurement tolerances.

Setting test limits is an iterative process of selecting highly reliable tests with regard to detection characteristics, and adjusting the limits on less-than-reliable tests in order to improve their detection characteristics.

Test set tolerances and variations can cause measurements to fail. Therefore, a convenient way to account for this is to expand the tolerances by increasing and decreasing the power source values and then using the Expand to Pass feature.

Monte Carlo analysis can also be used to set the measurement tolerances. However, Monte Carlo results tend to yield tolerances that are too tight. A combination of the two methods can also be used. Of course, tolerances can also be set manually on individual measurements or groups of measurements. Figure 8 shows the results dialog for the FinalValue measurement after increasing and decreasing the power supply values by 5% and using the Expand to Pass feature.

VIII. FAULT SIMULATION

At this point, a failure analysis is run. Measurements alarms can be set in order to flag failure modes which overstress parts. Once these failure modes are detected, separate tests can be added to find them. They can then be sequenced so that the failure mode test does not destroy functioning parts.

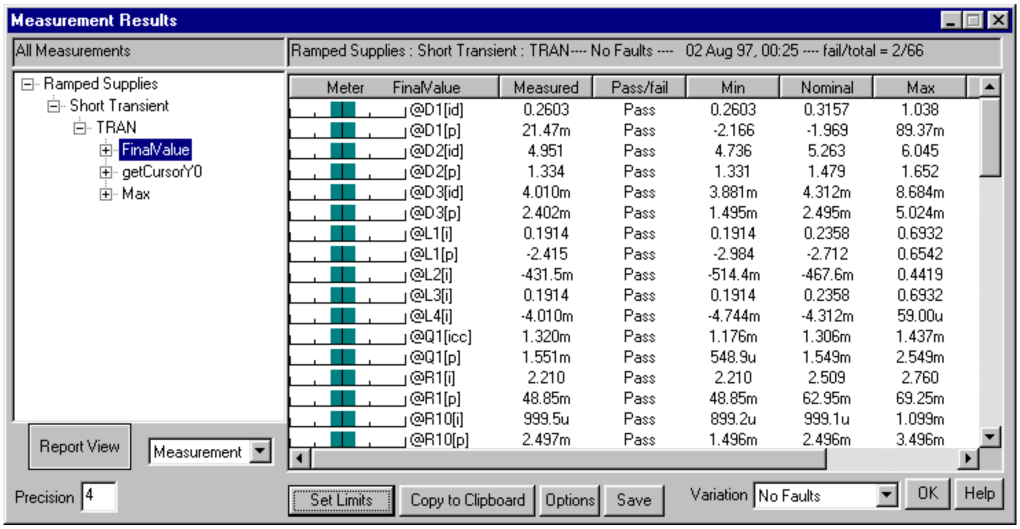
The fault universe for the SMPS consisted of 51 failure modes. Of particular interest were the PWM and power Mosfet faults. The three considered PWM IC failure modes were: Output shorted to VCC, Output shorted to ground, and Output open. The five considered power Mosfet failure modes were: (D=Drain, G=Gate, S=Source) shortedGS, shortedGD, shortedDS, OpenDS, and OpenG.

Failure mode simulation can proceed in one of several ways:

- One fault mode is inserted at a time for 1 test configuration.
- One fault mode is inserted at a time for several test configurations.
- All of the fault modes individually inserted, in succession, for 1 or more test configurations.

In this example, all failure modes are individually simulated for a ramped power supplies configuration, running a short transient analysis. The results are reported in a special dialog which is shown in Figure 9. For each failure mode, the software records the value of every user-defined measurement.

Past work [13,14] implies that simulation runtime is a major inhibitor to this methodology. However, these remarks tend to ignore recent developments in the area of model optimization and behavioral modeling, Analog Hardware Description Language (AHDL) modeling, and state-of-the-art simulator and computer performance. With the proper application of the these items and control of the simulator options, analysis of a SMPS in the transient domain using fault-by-fault simulation is clearly possible, as the indicated by the following results:



Measurement Results

All Measurements

Ramped Supplies : Short Transient : TRAN---- No Faults ---- 02 Aug 97, 00:25 ---- fail/total = 2/66

```
Ramped Supplies
  Short Transient
    TRAN
      FinalValue
      getCursorY0
      Max
```

Meter FinalValue	Measured	Pass/fail	Min	Nominal	Max
@D1[id]	0.2603	Pass	0.2603	0.3157	1.038
@D1[p]	21.47m	Pass	-2.166	-1.969	89.37m
@D2[id]	4.951	Pass	4.736	5.263	6.045
@D2[p]	1.334	Pass	1.331	1.479	1.652
@D3[id]	4.010m	Pass	3.881m	4.312m	8.684m
@D3[p]	2.402m	Pass	1.495m	2.495m	5.024m
@L1[i]	0.1914	Pass	0.1914	0.2358	0.6932
@L1[p]	-2.415	Pass	-2.984	-2.712	0.6542
@L2[i]	-431.5m	Pass	-514.4m	-467.6m	0.4419
@L3[i]	0.1914	Pass	0.1914	0.2358	0.6932
@L4[i]	-4.010m	Pass	-4.744m	-4.312m	59.00u
@Q1[icc]	1.320m	Pass	1.176m	1.306m	1.437m
@Q1[p]	1.551m	Pass	548.9u	1.549m	2.549m
@R1[i]	2.210	Pass	2.210	2.509	2.760
@R1[p]	48.85m	Pass	48.85m	62.95m	69.25m
@R10[i]	999.5u	Pass	899.2u	999.1u	1.099m
@R10[p]	2.497m	Pass	1.496m	2.496m	3.496m

Report View: Measurement

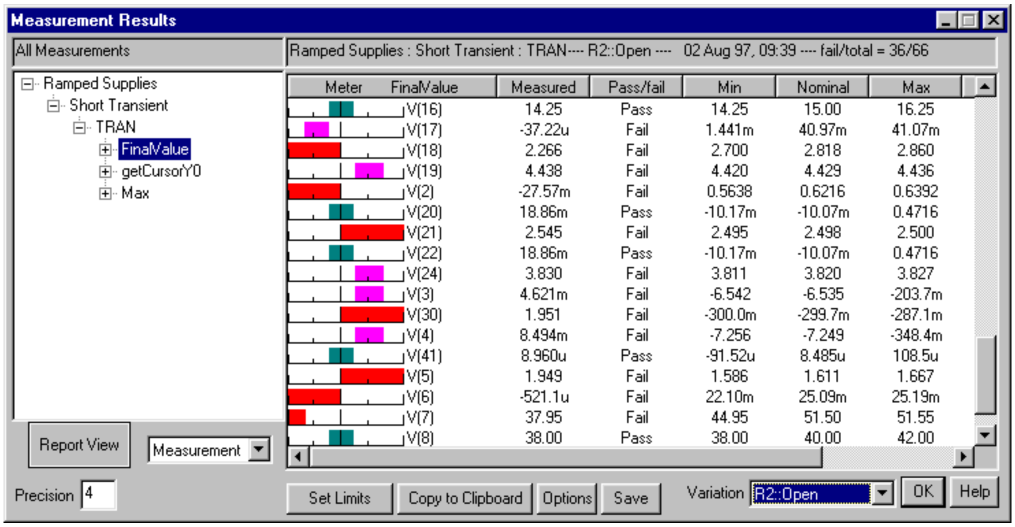
Precision 4

Variation: No Faults

Figure 8, The Results dialog after using the Expand to Pass feature to set tolerances for ± 5% power supply variations.

Type of Run	**Analyses**	**Circuit Configuration**	**Time**
Single Simulation	Full Transient	Ramped Supplies	184.90s
Single Simulation	Short Transient	Ramped Supplies	49.18s
Monte Carlo (30 Cases)	Short Transient	Ramped Supplies	23.5minutes
Failure Modes (51)	Short Transient	Ramped Supplies	47minutes

All simulations were performed on a 200Mhz Pentium® processor with 32MB RAM.



Measurement Results

All Measurements

- Ramped Supplies
  - Short Transient
    - TRAN
      - FinalValue
      - getCursorY0
      - Max

Ramped Supplies : Short Transient : TRAN---- R2::Open ---- 02 Aug 97, 09:39 ---- fail/total = 36/66

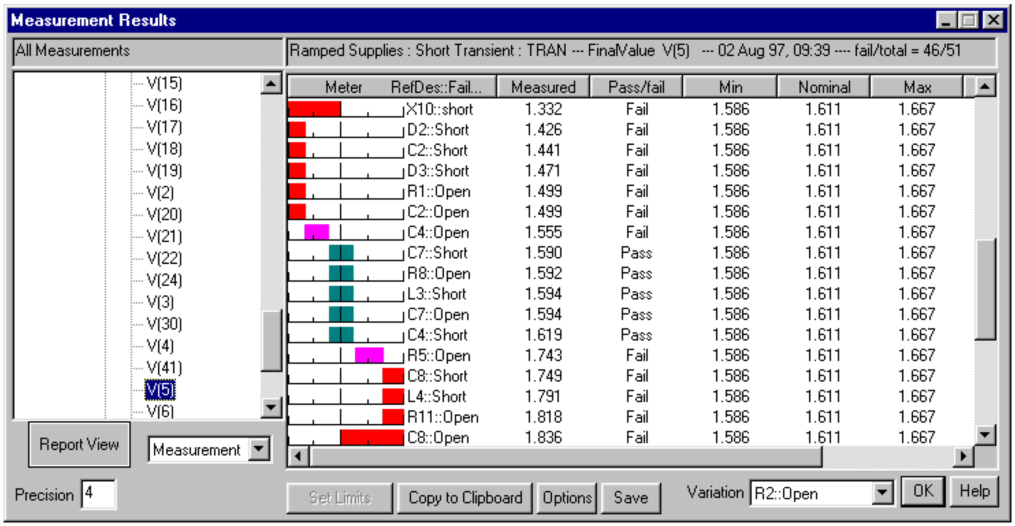
Meter	FinalValue	Measured	Pass/fail	Min	Nominal	Max
🟩	V(16)	14.25	Pass	14.25	15.00	16.25
🟪	V(17)	-37.22u	Fail	1.441m	40.97m	41.07m
🟥	V(18)	2.266	Fail	2.700	2.818	2.860
🟪	V(19)	4.438	Fail	4.420	4.429	4.436
🟥	V(2)	-27.57m	Fail	0.5638	0.6216	0.6392
🟩	V(20)	18.86m	Pass	-10.17m	-10.07m	0.4716
🟥	V(21)	2.545	Fail	2.495	2.498	2.500
🟩	V(22)	18.86m	Pass	-10.17m	-10.07m	0.4716
🟪	V(24)	3.830	Fail	3.811	3.820	3.827
🟪	V(3)	4.621m	Fail	-6.542	-6.535	-203.7m
🟥	V(30)	1.951	Fail	-300.0m	-299.7m	-287.1m
🟪	V(4)	8.494m	Fail	-7.256	-7.249	-348.4m
🟩	V(41)	8.960u	Pass	-91.52u	8.485u	108.5u
🟥	V(5)	1.949	Fail	1.586	1.611	1.667
🟥	V(6)	-521.1u	Fail	22.10m	25.09m	25.19m
🟥	V(7)	37.95	Fail	44.95	51.50	51.55
🟩	V(8)	38.00	Pass	38.00	40.00	42.00

Report View | Measurement ▾

Precision 4

Set Limits | Copy to Clipboard | Options | Save | Variation R2::Open ▾ | OK | Help

Figure 9, The results dialog after simulating all of the fault modes. The list of measurements for R2:Open are shown.



Measurement Results

All Measurements

V(15)
V(16)
V(17)
V(18)
V(19)
V(2)
V(20)
V(21)
V(22)
V(24)
V(3)
V(30)
V(4)
V(41)
V(5)
V(6)

Ramped Supplies : Short Transient : TRAN --- FinalValue V(5) --- 02 Aug 97, 09:39 ---- fail/total = 46/51

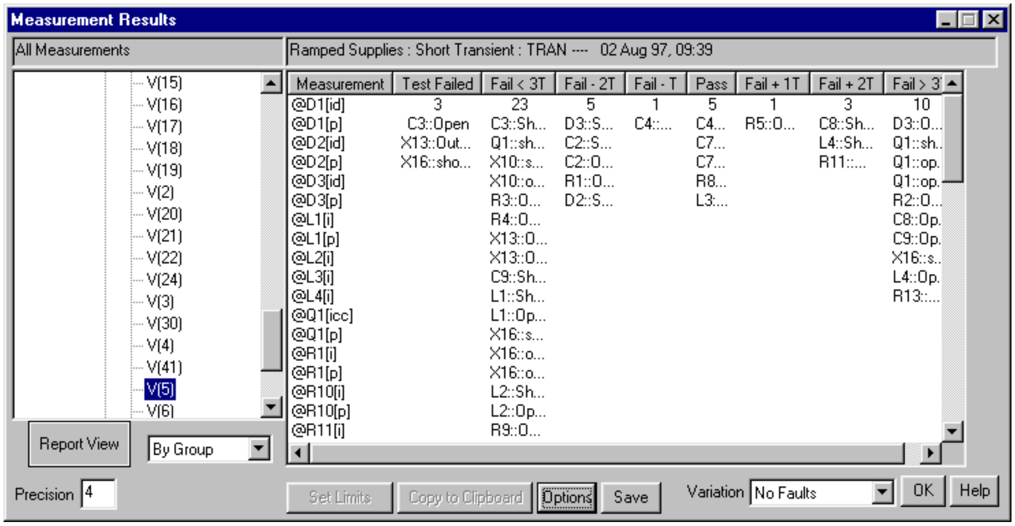
Meter	RefDes::Fail...	Measured	Pass/fail	Min	Nominal	Max
🟥	X10::short	1.332	Fail	1.586	1.611	1.667
🟥	D2::Short	1.426	Fail	1.586	1.611	1.667
🟥	C2::Short	1.441	Fail	1.586	1.611	1.667
🟥	D3::Short	1.471	Fail	1.586	1.611	1.667
🟥	R1::Open	1.499	Fail	1.586	1.611	1.667
🟥	C2::Open	1.499	Fail	1.586	1.611	1.667
🟪	C4::Open	1.555	Fail	1.586	1.611	1.667
🟩	C7::Short	1.590	Pass	1.586	1.611	1.667
🟩	R8::Open	1.592	Pass	1.586	1.611	1.667
🟩	L3::Short	1.594	Pass	1.586	1.611	1.667
🟩	C7::Open	1.594	Pass	1.586	1.611	1.667
🟩	C4::Short	1.619	Pass	1.586	1.611	1.667
🟪	R5::Open	1.743	Fail	1.586	1.611	1.667
🟥	C8::Short	1.749	Fail	1.586	1.611	1.667
🟥	L4::Short	1.791	Fail	1.586	1.611	1.667
🟥	R11::Open	1.818	Fail	1.586	1.611	1.667
🟥	C8::Open	1.836	Fail	1.586	1.611	1.667

Report View | Measurement ▾

Precision 4

Set Limits | Copy to Clipboard | Options | Save | Variation R2::Open ▾ | OK | Help

Figure 10, This version shows all of the fault modes results for the final value measurement of the output V(5).



Measurement Results

All Measurements

V(15)
V(16)
V(17)
V(18)
V(19)
V(2)
V(20)
V(21)
V(22)
V(24)
V(3)
V(30)
V(4)
V(41)
V(5)
V(6)

Ramped Supplies : Short Transient : TRAN ---- 02 Aug 97, 09:39

Measurement	Test Failed	Fail < 3T	Fail - 2T	Fail - T	Pass	Fail + 1T	Fail + 2T	Fail > 3'
@D1[id]	3	23	5	1	5	1	3	10
@D1[p]	C3::Open	C3::Sh...	D3::S...	C4::...	C4...	R5::O...	C8::Sh...	D3::O...
@D2[id]	X13::Out...	Q1::sh...	C2::S...		C7...		L4::Sh...	Q1::sh..
@D2[p]	X16::sho...	X10::s...	C2::O...		C7...		R11::...	Q1::op.
@D3[id]		X10::o...	R1::O...		R8...			Q1::op.
@D3[p]		R3::O...	D2::S...		L3:...			R2::O...
@L1[i]		R4::O...						C8::Op.
@L1[p]		X13::O...						C9::Op.
@L2[i]		X13::O...						X16::s..
@L3[i]		C9::Sh...						L4::Op.
@L4[i]		L1::Sh...						R13::...
@Q1[icc]		L1::Op...						
@Q1[p]		X16::s...						
@R1[i]		X16::o...						
@R1[p]		X16::o...						
@R10[i]		L2::Sh...						
@R10[p]		L2::Op...						
@R11[i]		R9::O...						

Report View | By Group ▾

Precision 4

Set Limits | Copy to Clipboard | Options | Save | Variation No Faults ▾ | OK | Help

Figure 11, Shows all of the faults for the final value measurement of V(5) using a histogram sorting technique. Failed tests are on the left. The rest of the faults are grouped into bins where T is the pass bandwidth (max-min).

IX. RESULTS REPORTING

The results for each failure are reported in a special dialog (shown in Figure 9). A tree list, similar to the Windows 95 Explorer tree, lists each drawing configuration, the simulation setups which are used under that configuration, and the individual analyses for each setup. Folded out from the analysis type are the various measurements that have been defined. There are two other display types; one that shows all of the failure results for each test, and another that shows a histogram of test measurements vs. failure modes, as shown in Figures 10 and 11.

The meter on the left of the report is used as a quick indicator of the measurement's pass-fail status. A long bar on the left or right of the meter center indicates that the associated failure mode moves the measured value outside of the pass/fail limits by more than 3 times the difference between the upper and lower pass/fail limits (e.g., a very high probability of failure detection). A short bar just to the right or left of center indicates that the failure is detected but is out of limits by less than one tolerance range (e.g. could be an uncertain detection and may merit further investigation using Monte Carlo techniques). The Variation drop-down contains a list of all of the simulated faults, making it easier to thumb through the results of each fault.

X. ADDING MORE TESTS

Using the test sequencing techniques which are described below, 78% of the faults were detected. X13:Out-toVcc, C3:Open, C2:Open, D1:Short, D1:Open, D2:Short, D2:Open, L2:Open, Q1:ShortCE, R1:Open, and X16:ShortGD were not be detected. Some faults were not be detected from the test data, while other faults caused simulation convergence failures. It is evident that the SMPS diagnostics require several additional test setups in order to completely detect all of the faults. Three other configurations are necessary. They are described below.

Circuit Configuration	Analyses	Faults Detected	Measurement	Description
Dead Circuit Test	AC analysis	C3:Open	Resistance	DMM resistance check
PWM Output Test	Short Transient	X13:Out-to-VCC	Peak-peak	PWM IC output short to Vcc
Reduced Supplies	Short Transient	C2:Open, D1/D2:Short D1/D2:Open, L2:Open Q1:shortCE, R1:Open X16:shortGD	Maximum	Main power supply off PWM supply ramped on slowly

One of the three configurations differed solely in its stimulus and power supply settings. Five schematic layers were created in order to implement these three configurations. The first layer, "Core Circuitry", contains all circuitry for the SMPS. The other layers contained different power supplies for the other configurations, or in the case of the impedance measurement, a DMM instrument. It should be noted that each of the configurations is a stand-alone design. Each has a unique netlist and a unique part list. The common production circuitry is carried across all configurations; this greatly helps minimize transcription errors.

In a second failure analysis pass, the three new configurations were simulated with respective failure modes inserted. The last step, discussed below, involves the sequencing of the tests into a fault tree.

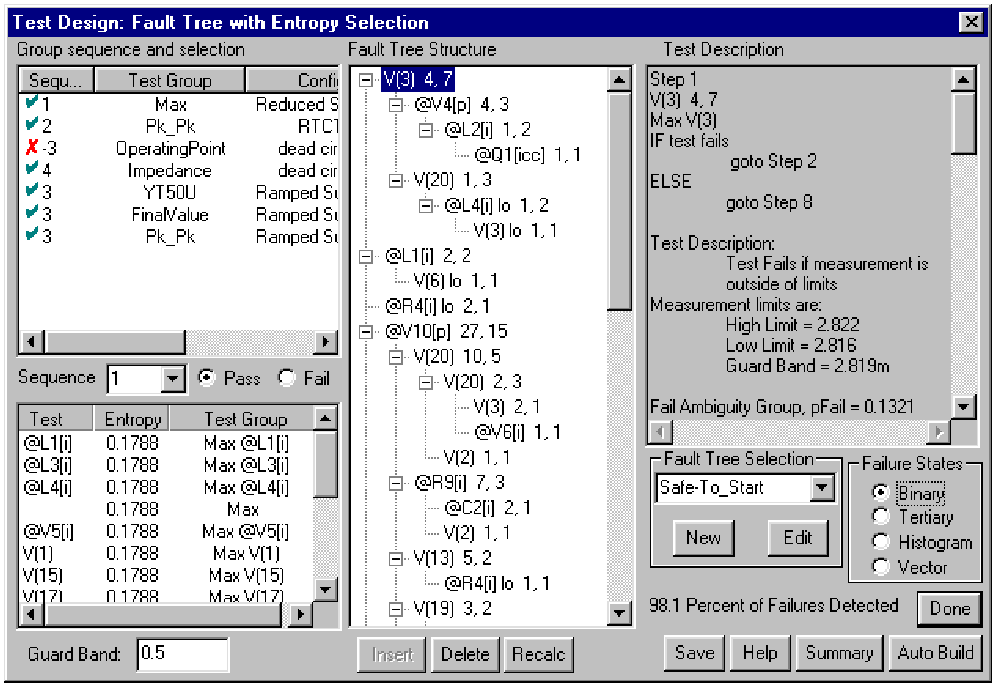
XI. RANKING OF FAILURE STATES

The process of fault detection and fault tree generation takes place in the Fault Tree design dialog (Figure 12) using a novel test sequencing technique.

It is generally accepted that the best fault tree is one that arrives at the highest probability failure conclusion with the least amount of work. The best test is then the test that produces the optimum fault tree, free from errors. Several methods for selecting the best test out of those remaining have been proposed [1,2]. Given an equal probability of occurrence for each fault, the best test is usually one that evenly divides the input group between the pass and fail group. A somewhat more complex procedure has also been proposed. Note that a general solution, made by exhaustive search, rapidly becomes intractable [2].

If the component failure rate is used to weight each fault in the ambiguity group, then we can assign a probability of detection to the pass group, p, and the fail group, q. What is really determined is the probability that a test will pass (p) and the probability that a test will fail (q). Then p + q = 1.0, because the test will either pass or fail. The input probability must be 1.0 because one of the conclusions in the current ambiguity group will be the answer. Weighting is used to reassess individual failure probability, given that the group is the answer at this point in the isolation process. Now the probability of reaching each conclusion can be predicted, based on failure weights. The best fault tree can now be defined as the one which arrives at each failure conclusion with the least amount of work. If each test is equally difficult, then the work is the summation of the probabilities of reaching each conclusion. To compute these probabilities, we simply traverse the tree for each conclusion, multiplying the probabilities of each successive outcome, and then summing the resultant probabilities for each conclusion. The best result from this procedure is 1.0. The figure of merit tells us how well we did, and can be used to compare fault trees. Clearly, we must select tests which produce high probability outcomes. To find these tests, we compute the entropy of each useful test that is available:

Entropy = -p*log(p) -q*log(q)



Test Design: Fault Tree with Entropy Selection

Group sequence and selection

Sequ...	Test Group	Confi
✓1	Max	Reduced S
✓2	Pk_Pk	RTC1
✗-3	OperatingPoint	dead cir
✓4	Impedance	dead cir
✓3	YT50U	Ramped Su
✓3	FinalValue	Ramped Su
✓3	Pk_Pk	Ramped Su

Sequence 1 ● Pass ○ Fail

Test	Entropy	Test Group
@L1[i]	0.1788	Max @L1[i]
@L3[i]	0.1788	Max @L3[i]
@L4[i]	0.1788	Max @L4[i]
	0.1788	Max
@V5[i]	0.1788	Max @V5[i]
V(1)	0.1788	Max V(1)
V(15)	0.1788	Max V(15)
V(17)	0.1788	Max V(17)

Guard Band: 0.5

Fault Tree Structure

```
V(3) 4, 7
  @V4[p] 4, 3
    @L2[i] 1, 2
      @Q1[icc] 1, 1
  V(20) 1, 3
    @L4[i] lo 1, 2
      V(3) lo 1, 1
@L1[i] 2, 2
  V(6) lo 1, 1
@R4[i] lo 2, 1
@V10[p] 27, 15
  V(20) 10, 5
    V(20) 2, 3
      V(3) 2, 1
      @V6[i] 1, 1
    V(2) 1, 1
  @R9[i] 7, 3
    @C2[i] 2, 1
    V(2) 1, 1
  V(13) 5, 2
    @R4[i] lo 1, 1
  V(19) 3, 2
```

Test Description

```
Step 1
V(3) 4, 7
Max V(3)
IF test fails
        goto Step 2
ELSE
        goto Step 8

Test Description:
        Test Fails if measurement is
        outside of limits
Measurement limits are:
        High Limit = 2.822
        Low Limit = 2.816
        Guard Band = 2.819m

Fail Ambiguity Group, pFail = 0.1321
```

Fault Tree Selection: Safe-To_Start

Failure States: ● Binary ○ Tertiary ○ Histogram ○ Vector

98.1 Percent of Failures Detected

Figure 12, The Fault Tree Design dialog sequences tests into a Fault Tree. The Fault Tree structure for the SMPS example shows the test and the number of faults in the ambiguity groups (pass first, then fail). For instance, L4[I]lo 1, 2 would have 1 fault in the pass ambiguity group and 2 in the fail ambiguity group. The description of each highlighted test is shown to the right.

According to information theory, the highest entropy test contains the most information. Proceeding in this manner tends to produce efficient fault trees. The software does not attempt to grade tests by difficulty, since this may be very subjective. Instead, tests may be grouped into a common pool for selection. This allows tests to be ordered logical requirements; for example, high temperature vs. low temperature and safe-to-start vs. operating point.

The failure state, (binary, tertiary, histogram or vector), defines the method by which measurements are used to make a test. Tests have only 2 outcomes, pass or fail; but a measurement can be compared with many different limits, creating a large number of possible tests for each measurement. Here's the way each failure state works to define tests:

1. **Binary:** The test passes if the result is within the test limits, and fails if it is outside of the limits.
2. **Tertiary:** The measurement is divided into 3 states; fail low, pass and fail high. Two tests are generated for each measurement, with outcomes of <pass, fail low> and <pass, fail high>.

3 **Histogram:** Measurements are divided into 7 groups which correspond to the values in the Results dialog's pass-fail meter. There are 6 tests, and each defines one of the failed bands; that is, <fail if in band, else pass>. The nominal band is not needed.

4 **Vector:** Let nFault be the number of faults. Then a vector can be created having nFault members, each containing the measured value for its respective fault. We arbitrarily set a limit about the fault, equal to the nominal tolerance, thus creating an ultimate histogram. Each measurement then has nFault tests, where the fail condition is the case in which the measurement is within the tolerance limits for that vector value.

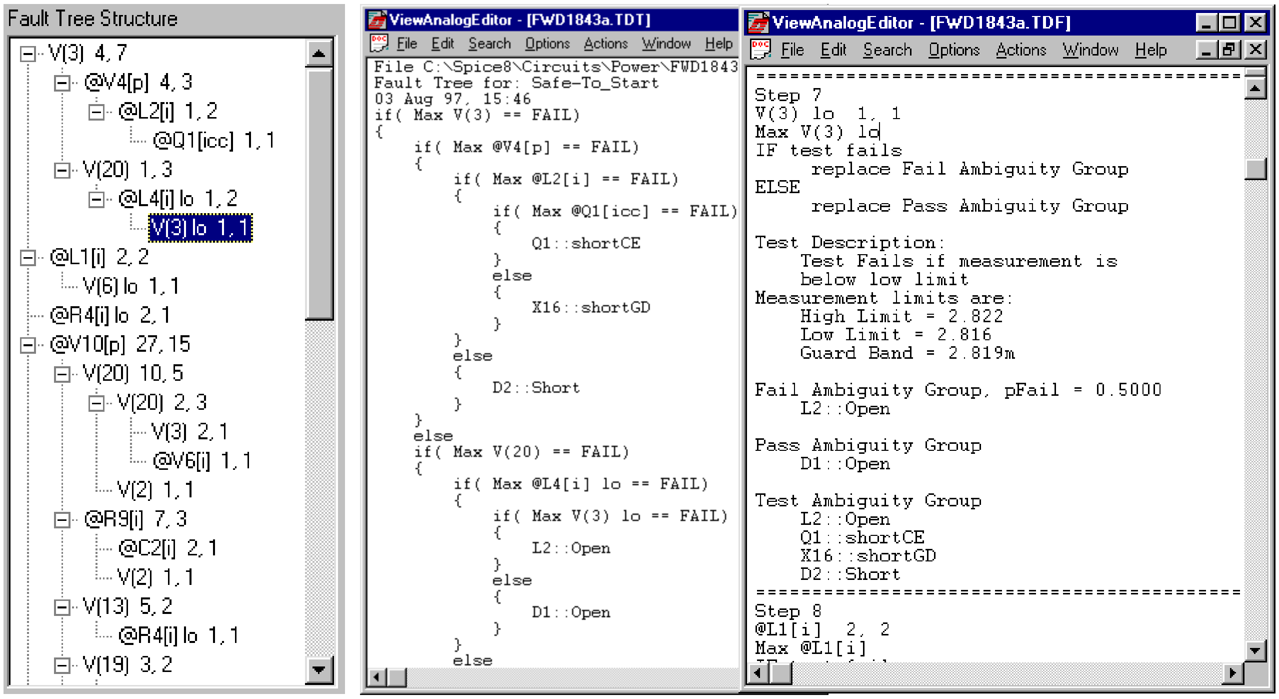
Ambiguity groups are created for each test. Test ambiguity groups contain a list of faults that can be detected; that is, faults that will be reported to the fail outcome it they are in the input fault group. The subset of the fault universe that goes into a test must be present in either the pass or fail outcome.

Vector and histogram-based tests require accurate measurements outside of a circuit's normal operating region. Measurements in this region are inherently less precise because fault models are less precise than nominal models, and faults can place parts that are not failed in unusual or high stress states for which analytic models are less accurate. These facts lead us to conclude that both vector and histogram failure states tend to produce less robust results.

Measured values can migrate across Failure State boundaries because of component and test tolerances. In order to produce robust tests, the concept of a guard band has been added. The guard band is measured in pass tolerance units, and is used to eliminate from consideration any test that has fault detections within the guard band limit of the test boundary. You can change the guard band for any Fault Tree step. The guard band width is reported in the Test Description field using the test measurement units.

When measurements are close to test boundaries, UUT tolerances and test set accuracy can move the result across the boundary, yielding a false result. This effect is mitigated by inserting a guard band just outside of the test limits. Setting the test limit to the center of the guard band produces a test that is least likely to give false results.

Referring to the Fault Tree Generation dialog in Figure 12, several groups of tests are available. They are shown in the Group sequence and selection section. Any of the test group-


Fault Tree Structure
ViewAnalogEditor - [FWD1843a.TDT]
File Edit Search Options Actions Window Help
ViewAnalogEditor - [FWD1843a.TDF]
File Edit Search Options Actions Window Help

```
V(3) 4, 7
    @V4[p] 4, 3
        @L2[i] 1, 2
            @Q1[icc] 1, 1
    V(20) 1, 3
        @L4[i] lo 1, 2
            V(3) lo 1, 1
@L1[i] 2, 2
    V(6) lo 1, 1
@R4[i] lo 2, 1
@V10[p] 27, 15
    V(20) 10, 5
        V(20) 2, 3
            V(3) 2, 1
            @V6[i] 1, 1
        V(2) 1, 1
    @R9[i] 7, 3
        @C2[i] 2, 1
        V(2) 1, 1
    V(13) 5, 2
        @R4[i] lo 1, 1
    V(19) 3, 2
```

```
File C:\Spice8\Circuits\Power\FWD1843
Fault Tree for: Safe-To_Start
03 Aug 97, 15:46
if( Max V(3) == FAIL)
{
    if( Max @V4[p] == FAIL)
    {
        if( Max @L2[i] == FAIL)
        {
            if( Max @Q1[icc] == FAIL)
            {
                Q1::shortCE
            }
            else
            {
                X16::shortGD
            }
        }
        else
        {
            D2::Short
        }
    }
    else
    if( Max V(20) == FAIL)
    {
        if( Max @L4[i] lo == FAIL)
        {
            if( Max V(3) lo == FAIL)
            {
                L2::Open
            }
            else
            {
                D1::Open
            }
        }
        else
```

```
==========================================
Step 7
V(3) lo  1, 1
Max V(3) lo
IF test fails
     replace Fail Ambiguity Group
ELSE
     replace Pass Ambiguity Group

Test Description:
    Test Fails if measurement is
    below low limit
Measurement limits are:
    High Limit = 2.822
    Low Limit = 2.816
    Guard Band = 2.819m

Fail Ambiguity Group, pFail = 0.5000
    L2::Open

Pass Ambiguity Group
    D1::Open

Test Ambiguity Group
    L2::Open
    Q1::shortCE
    X16::shortGD
    D2::Short
==========================================
Step 8
@L1[i]  2, 2
Max @L1[i]
```

Figure 13, The software produces several types of reports. Shown above is a portion of the fault tree which points to the V(3) test. The matching Test description and hardware independent C-like pseudo-logic code to duplicate the fault tree are shown to the center and right, respectively.

ings can be activated using the Sequence drop-down list. The resulting ranking of the tests, in order of best to worst, is shown in the Test-Entropy section below. In this case, 4 groups were arranged. All of the tests with the same sequence number will be evaluated at the same time. A fault tree can be sequenced from the activated tests manually, by hitting the insert button, or automatically, by hitting the AutoBuild button. This second action builds the entire fault tree, as shown in Figure 12.

From the failure analysis of our initial configuration, the pass ambiguity group from the last test in the fault tree was used to determine which faults could not detected (22% of the total shown in the "Adding More Tests" section). With all of the tests now available, 100% detection is possible.

Figure 13 displays a sample of some of the report outputs that are produced by the software.

XII. PROBLEMS WITH SENSITIVITY BASED FAILURE ANALYSIS TECHNIQUES

In several recent papers [13,14], claims are made that failures can be inferred from tolerances using sensitivity analysis. To accomplish this, a product specification is required. Based on that specification, tolerances are extracted using sensitivity analysis. The author claims that he can detect out-of-tolerance failures based upon these data. The underlying assumption is that test measurements are linearly related to parameter changes. **This is simply not true for most circuits.**

There are two key problems with the sensitivity approach. First, the frequency of part failures near the tolerance limits is but a small fraction of the total failure universe. Second, the linear nature of this prediction makes the implicit assumption that all parameters are related to each output by a first order (straight line) function. It can be shown that even many passive linear circuits fail to meet this criteria!

Using sensitivity analysis to predict circuit performance for extreme values or for component failures only works for certain linear circuits. The basic problem for most analog circuits is that the derivative of a test vector with respect to circuit parameters cannot be guaranteed to predict even the correct sign of the resulting test vector for large parameter changes. For example, consider a simple amplifier which has a gain of zero when its operating point is at either extreme, and a maximum value in between. If the test vector is proportional to gain, then the sensitivity will predict infinite gain for most operating points for one of the extreme fault cases. Even linear circuits can exhibit these characteristics for complex transfer functions; for example, parallel networks can shift complex zeros smoothly from left to right half planes by varying a single component value, yet the output waveform and its AC phase component will exhibit a discontinuity that can't be predicted using sensitivity analysis. In addition, this technique is generally not valid for circuits which contain switches or switching elements. Mixed signal circuits, by nature, are excluded. Hence, mixed-signal circuits and most power circuits such as PWM ICs can not be analyzed using sensitivity analysis.

In summary, here are a few examples that invalidate sensitivity based techniques:

- Circuits which contain switches. The sensitivity of a switched output with respect to its controlling input is always zero, so no inference is possible. **This applies to all mixed mode circuitry, including most power supply circuitry.**
- Linear circuits which have a non-minimum phase behavior (right half plane zeros). Twin T notch filters are a prime example, as are inverting transistor amplifiers.
- Control systems which have phase shifts of more than 180 degrees. AC sensitivity is based on the adjoint matrix solution at each frequency, so dependence on results at other frequencies is not possible. It is therefore not possible to distinguish between phase planes, making Nyquist stability criteria unavailable.

XIII. CONCLUSIONS

1) The simulation techniques employed here act as a "force multiplier" for development of diagnostics for analog and mixed signal circuits. Over the course of a few hours, it was possible to generate a variety of tests, determine the failure mode detection characteristics of each test, and sequence a subset of these tests into an effective diagnostic fault tree.

2) Simulation is proven to be an effective method for identifying problem areas in fault detection. Simulation reveals key problem areas: first, by identifying the low probability fault detections for individual tests; and second, by providing an infrastructure for further circuit analysis when integration results do not agree with simulator predictions.

3) Reasonable simulation times can be achieved by parsing the circuit into separate subcircuits and linking the simulation results through the use of voltage and current stimuli.

4) The software allows the user to study various power supply failure mechanisms.

Accounting for achieved detection and isolation of failure modes is a difficult problem for the TPS developer. While accountability is easy to achieve in simple circuit analysis, it becomes considerably more difficult as the circuit complexity increases. The preceding techniques provide a significant advantage, by keeping track of detection and isolation of failure modes during generation of the fault tree.

There are several ways to overcome simulation runtime issues. The first is to use a faster computer or many fast computers to perform the runs more quickly and in parallel. A more attractive method for reducing simulation time is to model the switching elements using a linearization technique called state space averaging [2]. Many faults cannot be detected when using this approach, however, the simulation time for the long switching simulations will be reduced by an order of magnitude.

REFERENCES

[1] Sharon Goodall, "Analog/Mixed Signal Fault Diagnosis Algorithm and Tool Review",1994 AutoTestCon Proceedings, pp. 351-359.

[2] W.R. Simpson and J.W. Sheppard, "Fault Isolation in an Integrated Diagnostic Environment, "IEEE D&T, Vol.10,No.1,Mar. 1993, pp52-66.

[3] C.Y. Pan and K.T. Cheng, "Test Generation for Linear, Time Invariant Analog Circuits, 3rd IEEE Intl. Mixed-Signal Testing Workshop, June 3-6, 1997.

[4] Harry H. Dill, "A Comparison of Conventional and Inference Model Based TPS Development Processes", AutoTestCon '95 Proceedings, pp 160-168.

[5] Harry H. Dill, "A Comparison of Conventional and Inference Model Based TPS Development Processes", 1997 AutoTestCon Proceedings.

[6] UL508C Standard for Power Conversion Equipment, ISBN 0-7629-0093-8, Copyright © 1996 Underwriters Laboratories Inc., Nov. 1996.

[7] CASS Red Team Package data item DI-ATTS-80285B, Fig. 1 - SRA/SRU Fault Accountability Matrix Table, pg 11.

[8] L.W. Nagel and D.O. Pederson, Simulation program with integrated circuit emphasis, ERL Memo No. ERL-M520, University of California, Berkeley, May 1975.

[9] B. Johnson, T. Quarles, A.R. Newton, D.O. Pederson, A. Sangiovanni-Vincentelli, SPICE 3F User's Guide, University of California, Berkeley, Oct. 1992.

[10] "IsSpice4 User's Guide", Intusoft, P.O. Box 710, San Pedro, CA 90733-0710, 1997.

[11] Fred Cox, W. Kuhn, J. Murray, S. Tynor, "Code-Level Modeling in XSPICE", Proceedings of the 1992 Intl. Syp. on Circuits and Systems, May 1992, IEEE 0-7803-0593-0/92.

[12] "XSPICE Users Manual", Georgia Institute of Research, Georgia Institute of Technology, Atlanta GA 30332-0800.

[13] N.B. Hamida, K. Saab, D. Marche, B. Kaminska, and G. Quesnel, "LIMSoft: Automated Tool for Design and Test Integration of Analog Circuits", 2nd IEEE International Mixed Signal Testing Workshop, Quebec City, Canada, May 15-18, 1996.

[14] N.B. Hamida and B. Kaminska, "Analog Circuit Fault Diagnosis Based on Sensitivity Computation and Functional Testing", IEEE Design and Test of Computers, March 1992, pp.30-39.

[15] Steven M. Sandler, SMPS Simulation with SPICE, McGraw-Hill, 1997.

Application of Analog & Mixed Signal Simulation Techniques to the Synthesis and Sequencing of Diagnostic Tests

Harry H. Dill, Deep Creek Technologies, Inc., Annapolis, MD 21401 USA, Kyle Bratton, Chris Sparr, Naval Aviation Depot, MCAS Cherry Point, NC USA, Lloyd Pitzen, CCI Incorporated, Arlington, VA 22202 USA

Abstract - This paper describes the fault definition, simulation, test strategy development and validation activities that were accomplished during beta testing of a new CAE tool, Test Designer. It addresses the issues of simulation convergence, component fault models, circuit model implementation, simulation run times and test strategy accuracy. To ensure a realistic test of Test Designer's capabilities, diagnostics were developed for a moderately complex analog and mixed signal UUT which was selected from the Navy's list of CASS TPS offload candidates. These diagnostics were evaluated on the CASS to measure the diagnostic sequence accuracy and to assess the ability of Test Designer to predict the nominal measurement values and fault detection characteristics of each test.

I. INTRODUCTION

In late 1996, the SPICE simulation tool manufacturer Intusoft initiated development of a new product tailored to the unique and demanding needs of the test engineer. This product, **Test Designer**, was to provide an effective, interactive design environment for the synthesis of diagnostic tests, generation of fault dictionaries and the building of diagnostic fault trees. As part of the product testing process, a beta version of Test Designer was used to synthesize diagnostic tests and build diagnostic fault trees for an analog and mixed signal UUT which was selected from the Navy's CASS offload candidate list. This paper describes the process by which the diagnostics were developed and documents the accuracy of these diagnostics as demonstrated in the Navy's Test Integration Facility in Chesapeake, VA.

A seven step process was implemented for development of the diagnostics. In chronological order, they were:

1) Circuit Parsing
2) Schematic Entry
3) Measurement Definition
4) Simulation
5) Test Synthesis
6) Fault Tree Generation
7) Integration

Details of each step in this process are provided in the sections that follow.

II. CIRCUIT PARSING

The Annunciator Light Control Power Supply (ALCPS) was chosen as the UUT for which diagnostics would be developed because of its complexity. It contains two virtually identical sections, each converting aircraft power input of 16Vdc to 30Vdc into 28Vdc in the "Bright" mode of operation and pulsed dc voltage averaging 7 to 14 Volts in the "Dimmer" mode. Since the two sections of the ALCPS are very similar, only the Emergency section was evaluated. The results are easily transferred to the Essential section of the ALCPS. Figure 1 is a block diagram of the Emergency section of the ALCPS. Subsequent reference to the ALCPS refers only to the Emergency section of the power supply.

The original plan called for simulating the entire supply as a single entity. An initial attempt at such a simulation showed that the runtime to complete diagnostics development would be excessive as far as this paper is concerned. There were several reasons :

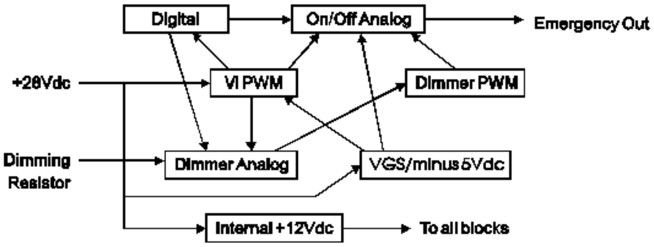
1) While the simulation time increases linearly (approximately 1.2 times [1]) with the size of the circuit, the time constants that are associated with the functional blocks the ALCPS posed a greater problem. They range from approximately 100us in the VGS/minus 5Vdc block to 250ms in the digital block. A transient analysis in SPICE must compute a sufficient number of points on the time line in order to accurately reproduce the circuit waveforms. In the case of the VGS/minus 5Vdc block the time interval for the computations must be in the 1us to 10us range. The digital block must be simulated over a total time period of several seconds in order to accurately capture its behavior since it contains several long time-constant RC delay circuits. Therefore, more than a million computations could be required to simulate the ALCPS behavior for a single test setup.

2) Test Designer computes the fault dictionaries by simulating circuit behavior for each failure mode of each component - one at a time. Approximately 300 failure modes must be considered for the entire ALCPS Emergency section. With a simulation time for a single run in excess of 8 hours on a 90Mhz Pentium®, 4 months of processing time would be required to complete the task.

There several ways of overcoming these issues. The first is to use a faster computer or a "farm" of fast computers to make the numerous runs more quickly and in parallel. Current state-of-the-art computers (i.e., 266MHz Pentium® II) would reduce the simulation time for each failure mode by approximately 25% making the problem much more manageable.

A more attractive method for reducing simulation time would be to model the switching elements using a linearization technique called state space averaging [2]. Some faults would not be detected when using this approach, however, the simulation time for the long switching simulations would be reduced by an order of magnitude.

In this case, the solution we adopted was to simulate each block of the ALCPS as a separate entity. By replacing functional blocks that interface with the block under evaluation with voltage sources that emulated both the nominal and failed modes of operation, the behavior of each block was easily simulated. For example, the digital functional block in Figure 1 provides CMOS outputs of +12Vdc or 0Vdc (depending on its stimuli) to the On/Off Analog circuit. For each digital block stimuli setup, voltage sources can represent the digital block outputs for nominal, Stuck Hi and Stuck Lo failure modes that might exist at those outputs or propagate to those outputs. These voltage sources (emulating the Digital Block outputs) were used as stimuli when evaluating the On/Off Analog block. Subsequent simulation of the digital block showed that for all failures within the block, these were the only failures observed at the digital block outputs. Since there is no clocking associated with these outputs, static voltage levels may be used. Had clocking been an issue, we could have used stimuli that generate time-varying waveforms.



The objective of the circuit parsing step was to break the circuit up into functional blocks in such a manner as to reduce simulation time while maintaining the integrity of fault effects propagation across block boundaries. The configuration in Figure 1 accomplished this objective.

III. SCHEMATIC ENTRY

Test Designer employs graphical "drag and drop" schematic entry features found in most modern EDA systems. However, there are two significant differences - component parameter data entry and circuit configuration definition.

A. Component Parameter Data Entry

Each component on the schematic has an associated graphical dialog that can be accessed to define nominal device and model parameter information, as well as parametric tolerances. The tolerances can be used in subsequent Monte Carlo analyses to assist in the definition of test pass/fail limits. Each component also has an associated failure mode definition dialog that allows you to define the parameters for each of the component's failure modes. Up to 32 failure modes can be assigned to any single component. In addition, Test Designer provides selectable failure modes that are defined in the CASS Red Team Package. Theses failure modes were used for the ALCPS. The fault universe for the modeled portion of the ALCPS consisted of 204 faults that are allocated as follows:

On/Off Analog block	66 component failure modes
Dimmer Analog block	66 component failure modes
Digital block	72 component failure modes

B. Circuit Configuration Definition

A test setup provides loads, voltage and current stimuli and instrumentation connections at specific points on the Unit Under Test (UUT). When viewed in a broader context, the combination of the test setup circuitry and the UUT can be considered to be a circuit configuration in and of itself. Indeed, for simulation purposes, the test setup circuitry must be included as part of the circuit. Most Test Program Sets (TPSs) implement multiple setups during the testing sequence. This increases the simulation burden by requiring a separate schematic for every test setup. The ALCPS diagnostics require several different test setups in order to completely test the "Bright" and "Dimmer" functions. Specific setups include the connection of high and low resistance values to the dimmer inputs and changes in the ALCPS primary power and digital inputs.

Test Designer addresses the multiple test setup problem by assigning each setup/UUT combination a unique configuration name and simulating all configurations in a batch operation. In Test Designer, the setup/UUT combination is called a "circuit configuration". The circuit configuration is defined during the schematic entry process. Every circuit configuration is composed of one or more schematic layers. An active layer can be thought of as a transparency that overlays other transparencies such that as you view them you see the complete circuit configuration schematic. Circuit nodes on the top layer connect with nodes on underlying layers as if the drawing were created on a single page. Test Designer allows mixing and matching of layers to form the required circuit configurations.

Two circuit configurations were defined for the digital block of the ALCPS. Three schematic layers were created in order to implement these two configurations. The first layer, "UUT", contained all circuitry for the digital portion of the ALCPS. The second layer, "MainStim", contained five piecewise linear voltage sources (digital signal stimuli), a pulsed voltage source (Vplus12, the power into the digital functional block), and a pulsed voltage source (Vi, filtered power). Vi and Vplus12 were given associated failure modes which were representative of their respective functional blocks. The third layer, "MainStim2", was similar to MainStim, however, the definition of the five piecewise linear voltage sources reflected the required digital stimuli for tests that differed significantly from those of MainStim. A circuit configuration was named MainLine and assigned layers "UUT" and "MainStim". Another circuit configuration was named MainLine2 and assigned layers "UUT" and "MainStim2". Circuit configurations were created in a similar manner for the On/Off Analog and Dimmer functional blocks. None of the PWM blocks were included in the analysis. Due to the low component count in these blocks, a manual analysis was performed. A transient analysis was run on the VGS/minus 5Vdc block to gauge the difficulty associated with its simulation. This block contains an SG1524 Pulse Width Modulator and simulation time using a 90Mhz Pentium® was approximately four hours. Techniques exist for reducing this time but were not implemented in the ALCPS circuit model. [3]

The developer of the ALCPS Offload TPS wished to use an existing set of verification tests and prepare diagnostic fault trees that branch from failed verification tests. The setups defined for these verification tests formed the basis for the circuit configurations "MainLine" and "MainLine2".

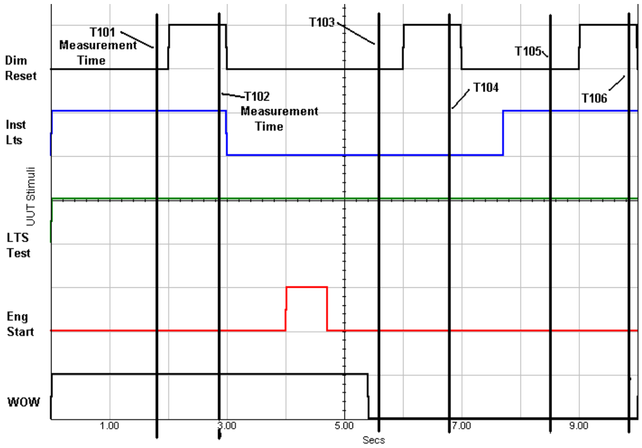


Figure 2 shows the waveforms for the digital stimuli that are defined in the "MainStim" layer of the "MainLine" circuit configuration. These waveforms were derived directly from the existing ALCPS verification tests and represent digital stimuli as the TPS progresses sequentially through verification tests T1010, T1020, T1030, T1035, T1040, T1050, T1060 and T1080. T1070 applied exactly the same stimuli to the digital block as T1060 and was not repeated in the simulation. Each test toggles only those digital stimulus lines that are needed to achieve the desired 5 bit pattern, then delays 700ms to allow the circuit to settle and makes a voltage measurement at the main output node of the ALCPS. The UUT is not de-energized between tests. Since the digital block contains two simple NOR gate flip flops, it retains the state of the circuit as defined by the stimuli that were applied during previous tests. Therefore, the sequence of these tests is important in the context of the failure modes that each test will detect. The circuit configuration "MainLine" duplicates this scenario by using identical stimuli in the simulation. The simulator was set up to record the circuit parameters at every node in the circuit at the time of each test measurement as performed by the actual TPS. At the conclusion of test T1080, power is removed from the ALCPS, then power is reapplied and testing is resumed using the stimuli duplicated in the "MainLine2" circuit configuration.

IV. MEASUREMENT DEFINITION

Test Designer uses the IsSpice4 simulation engine. IsSpice4 is an enhanced version of Berkeley SPICE 3 [4] and XSPICE [5]. This engine offers several simulation analysis options, among them DC Operating Point, AC and Transient. Each of these options provides different information regarding the circuit. Transient analysis was selected for development of the ALCPS since we are interested in dynamic circuit performance over a period of time.

A specific type of analysis must be assigned to each circuit configuration before it can be simulated. The association of a simulation analysis type with a circuit configuration is defined as a test configuration by Test Designer. A single circuit configuration can be assigned multiple types of simulation analyses in order to define multiple test configurations. Only transient (time-based) analyses were performed on the ALCPS.

Test Designer records only the simulation information that is relevant to the test measurements that have been defined and selected by the test engineer. The available information includes the voltage at each circuit node (at any specified time for the transient analysis), time relationships between events (rise times, triggered measurements, pulse widths, etc.), the current through any component and the power dissipated by any component. For the ALCPS "MainLine+tran1" test configuration, a total of 224 measurements were recorded. These measurements consist of the node voltage at each of 32 circuit nodes as recorded at 2.8 seconds, 3.7 seconds, 5.5 seconds, 6.7 seconds, 8.5 seconds, 9.8 seconds and 10.8 seconds into the simulation. These measurement times correspond to those of the verification tests T1010 through T1080. The ALCPS provides probe access to every circuit node via the back of the circuit card. Similarly, 192 test measurements were defined for the "MainLine2+tran1" test configuration. This yields a pool of 416 tests from which to choose when isolating a failure in the digital block fault universe of 72 failure modes. Not all of these measurements will necessarily be used in the final fault tree. The objective at this point is to create a large population of tests from which to choose when computing an efficient fault isolation strategy.

A similar process was used to define measurements for the On/Off Analog and the Dimmer Analog functional blocks. Eighty-seven (87) measurements were defined for the On/Off Analog functional block; 116 were defined for the Dimmer Analog block. Thus, the total measurement pool for the simulated sections of the ALCPS was 619.

V. SIMULATION

Test Designer generates a value for every defined measurement in every test configuration for every enabled failure mode of the circuit by performing the simulations that are defined in the test configurations. A measurement value is also generated for the No Fault condition. The test engineer may declare any component as “can’t fail” in order to exclude special circuitry such as external test equipment interfaces from the fault analysis computations. In addition to this automated “batch simulation” mode, Test Designer permits simulation of individual failure modes in an interactive, manual mode.

Simulation of the ALCPS test configurations posed few problems. It was quickly noted however that the first run of a simulation should be performed in manual mode for the No Faults case to ensure the circuit is correctly configured and the simulation runs to completion.

Convergence problems appeared only once. The digital signal stimuli for the UUT is applied 1 second before power is applied. Application of the power caused a transient in the circuit simulation that resulted in a non-convergence situation. This situation was quickly resolved by invoking Test Designer’s Convergence Wizard. The Convergence Wizard asks various questions and automatically manipulates the IsSPICE simulation control options in an attempt to mitigate the problem. Once the circuit converged for a test configuration, no additional problems were observed during the fault simulation process. If convergence problems do occur during an individual fault simulation in “batch” mode, the measurement value is flagged as non-convergent and can be subsequently evaluated individually in manual mode by changing simulation parameters.

“Batch” mode simulation times for all faults and all test configurations were as follows:

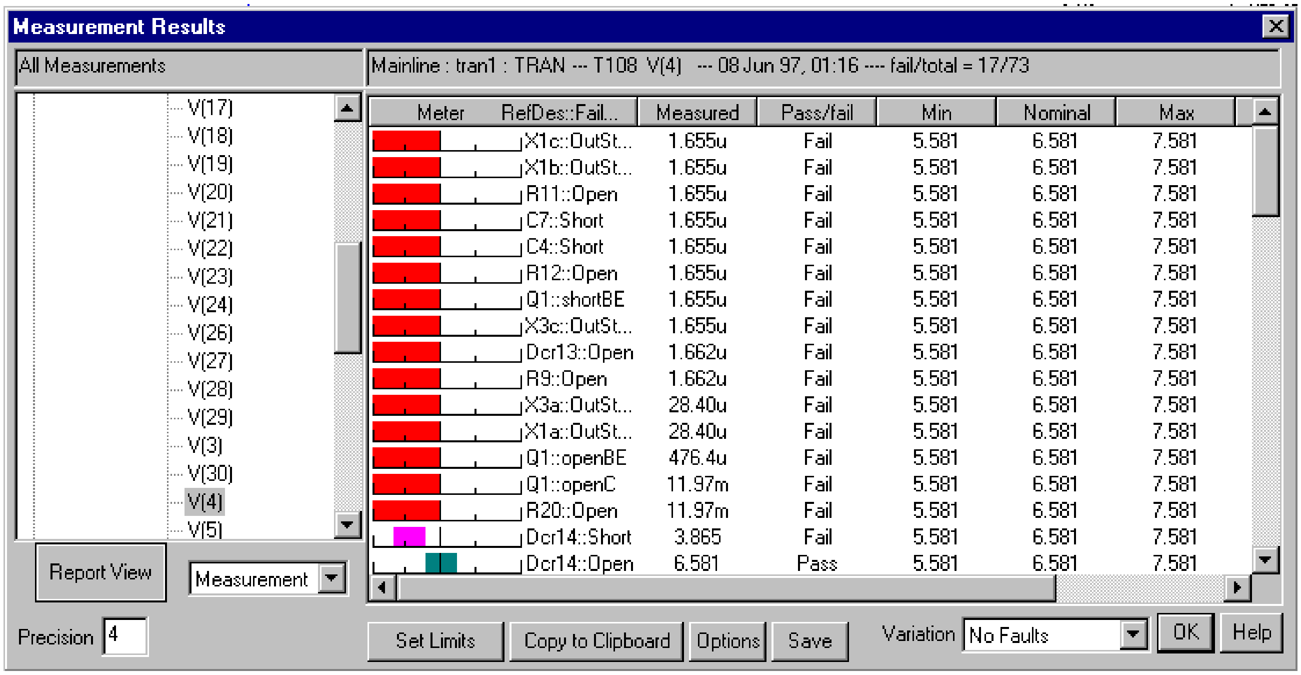
Digital block	51 minutes
Dimmer Analog block	25 minutes
On/Off Analog block	42 minutes

All simulations were performed on a 90Mhz Pentium® processor.

VI. TEST SYNTHESIS

A test is defined as a set of circuit stimuli, a parametric measurement, a circuit node where the parameter is measured and criteria for determining whether the test passes or fails [6]. At this point in the process 619 parametric measurements have been defined for the ALCPS including stimuli and measurement locations. The next step is to convert these measurements into tests by assigning pass/fail criteria.

Figure 3 shows one of the graphical methods Test Designer provides in order to facilitate test limits definition. A description of the methodology that is used to set the ALCPS digital block test limits will illustrate the application. Examination of the digital block topology revealed that most of the circuit nodes connect CMOS digital components and can be expected to fall within ±1volt of +12Vdc or 0Vdc during normal operation. Initially, the limits for all nodes were set to ±1volt about the nominal value using the global set capability in Test Designer. The limits for the non-digital nodes were then set to ±10% of the nominal values. (A more rigorous approach would have been to use the Monte Carlo feature of Test Designer to derive circuit performance). This rough cut converted the 416 measurements into 416 tests. Figure 3 shows the test results for the “MainLine+tran1” test configuration at the T1080 test time for node V(4) for every simulated failure mode. Note the



All Measurements

V(17)
V(18)
V(19)
V(20)
V(21)
V(22)
V(23)
V(24)
V(26)
V(27)
V(28)
V(29)
V(3)
V(30)
V(4)
V(5)

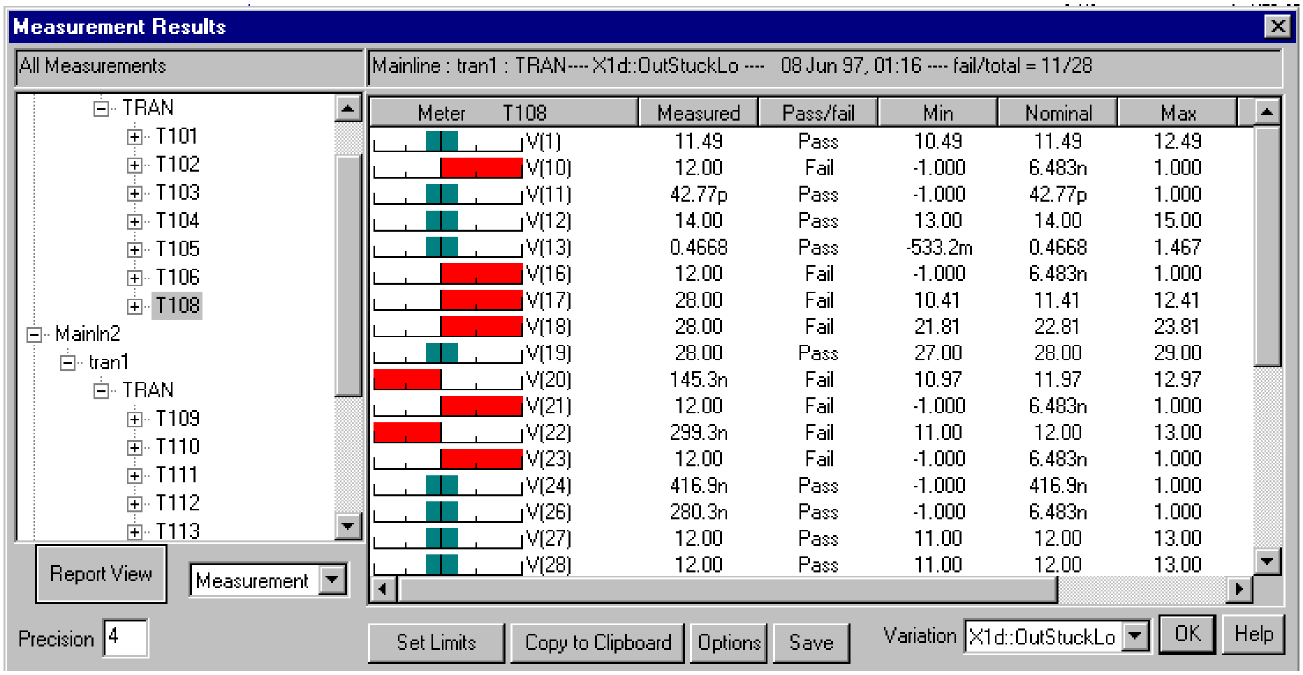
Mainline : tran1 : TRAN --- T108 V(4) --- 08 Jun 97, 01:16 ---- fail/total = 17/73

Meter	RefDes::Fail...	Measured	Pass/fail	Min	Nominal	Max
🟥	X1c::OutSt...	1.655u	Fail	5.581	6.581	7.581
🟥	X1b::OutSt...	1.655u	Fail	5.581	6.581	7.581
🟥	R11::Open	1.655u	Fail	5.581	6.581	7.581
🟥	C7::Short	1.655u	Fail	5.581	6.581	7.581
🟥	C4::Short	1.655u	Fail	5.581	6.581	7.581
🟥	R12::Open	1.655u	Fail	5.581	6.581	7.581
🟥	Q1::shortBE	1.655u	Fail	5.581	6.581	7.581
🟥	X3c::OutSt...	1.655u	Fail	5.581	6.581	7.581
🟥	Dcr13::Open	1.662u	Fail	5.581	6.581	7.581
🟥	R9::Open	1.662u	Fail	5.581	6.581	7.581
🟥	X3a::OutSt...	28.40u	Fail	5.581	6.581	7.581
🟥	X1a::OutSt...	28.40u	Fail	5.581	6.581	7.581
🟥	Q1::openBE	476.4u	Fail	5.581	6.581	7.581
🟥	Q1::openC	11.97m	Fail	5.581	6.581	7.581
🟥	R20::Open	11.97m	Fail	5.581	6.581	7.581
🟪	Dcr14::Short	3.865	Fail	5.581	6.581	7.581
🟩	Dcr14::Open	6.581	Pass	5.581	6.581	7.581

Report View: Measurement

Precision 4

Variation No Faults



Measurement Results

All Measurements

Mainline : tran1 : TRAN---- X1d::OutStuckLo ---- 08 Jun 97, 01:16 ---- fail/total = 11/28

- TRAN
  - T101
  - T102
  - T103
  - T104
  - T105
  - T106
  - T108
- MainIn2
  - tran1
    - TRAN
      - T109
      - T110
      - T111
      - T112
      - T113

Meter	T108	Measured	Pass/fail	Min	Nominal	Max
🟩	V(1)	11.49	Pass	10.49	11.49	12.49
🟥	V(10)	12.00	Fail	-1.000	6.483n	1.000
🟩	V(11)	42.77p	Pass	-1.000	42.77p	1.000
🟩	V(12)	14.00	Pass	13.00	14.00	15.00
🟩	V(13)	0.4668	Pass	-533.2m	0.4668	1.467
🟥	V(16)	12.00	Fail	-1.000	6.483n	1.000
🟥	V(17)	28.00	Fail	10.41	11.41	12.41
🟥	V(18)	28.00	Fail	21.81	22.81	23.81
🟩	V(19)	28.00	Pass	27.00	28.00	29.00
🟥	V(20)	145.3n	Fail	10.97	11.97	12.97
🟥	V(21)	12.00	Fail	-1.000	6.483n	1.000
🟥	V(22)	299.3n	Fail	11.00	12.00	13.00
🟥	V(23)	12.00	Fail	-1.000	6.483n	1.000
🟩	V(24)	416.9n	Pass	-1.000	416.9n	1.000
🟩	V(26)	280.3n	Pass	-1.000	6.483n	1.000
🟩	V(27)	12.00	Pass	11.00	12.00	13.00
🟩	V(28)	12.00	Pass	11.00	12.00	13.00

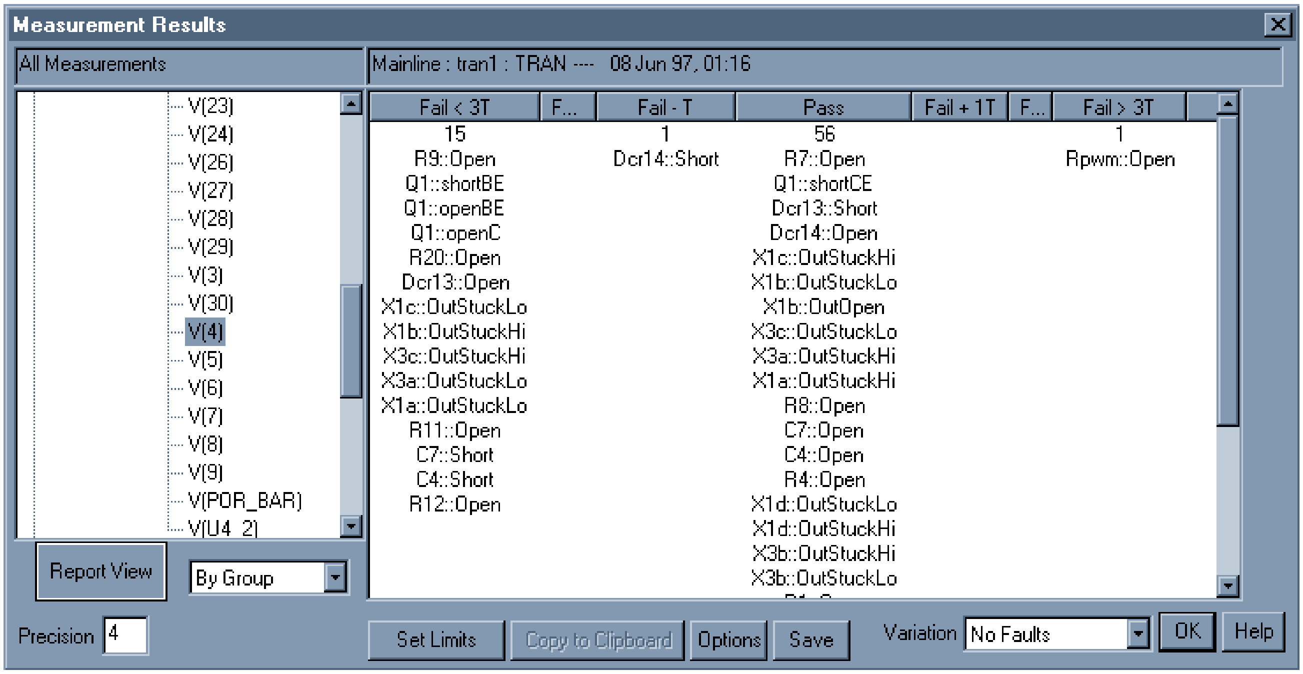
Report View | Measurement

Precision: 4

Set Limits | Copy to Clipboard | Options | Save | Variation: X1d::OutStuckLo | OK | Help

Meter on the left of the graphic. A short bar in the center of the meter indicates that the associated failure mode is not detected. This bar is colored green on the computer screen. A long bar on the left or right of the meter center indicates that the associated failure mode moves the measured value outside the pass/fail limits by more than 3 times the difference between the upper and lower pass/fail limits (e.g., a very high probability failure detection). This bar is colored bright red. A short bar just to the right or left of center indicates that the failure is detected but is out of limits by less than one tolerance range (e.g. could be an uncertain detection and may merit further investigation using Monte Carlo techniques). This bar is colored purple. A quick scan of this display gives a good indication of the reliability of an individual test in detecting the failures in its fault dictionary. There are two other display types available to display the failure detection characteristics of each test. Figure 4 shows all T1080 test results for the failure mode X1d::OutStuckLo. Figure 5 shows a histogram of test measurements vs. failure modes for the T1080 voltage test at node 4.

Setting limits for the ALCPS tests was an iterative process of selecting highly reliable tests with regard to detection characteristics (created using the rough cut limits) for inclusion in the Fault Tree and adjusting the limits on less than reliable tests in order to improve their detection characteristics when such tests were required to achieve desired isolation metrics.



Measurement Results

All Measurements

Mainline : tran1 : TRAN ---- 08 Jun 97, 01:16

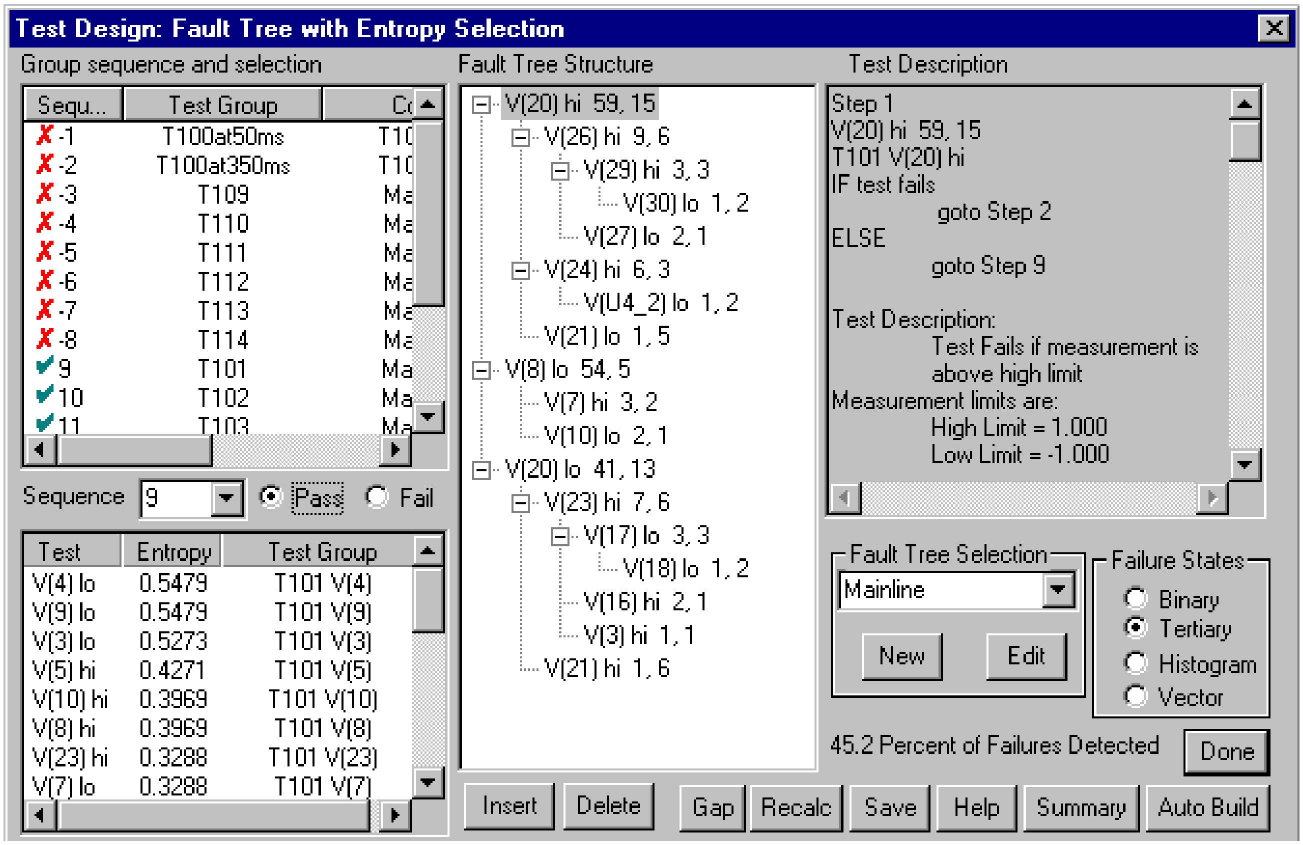
- V(23)
- V(24)
- V(26)
- V(27)
- V(28)
- V(29)
- V(3)
- V(30)
- V(4)
- V(5)
- V(6)
- V(7)
- V(8)
- V(9)
- V(POR_BAR)
- V(U4_2)

Fail < 3T	F...	Fail - T	Pass	Fail + 1T	F...	Fail > 3T
15		1	56			1
R9::Open		Dcr14::Short	R7::Open			Rpwm::Open
Q1::shortBE			Q1::shortCE			
Q1::openBE			Dcr13::Short			
Q1::openC			Dcr14::Open			
R20::Open			X1c::OutStuckHi			
Dcr13::Open			X1b::OutStuckLo			
X1c::OutStuckLo			X1b::OutOpen			
X1b::OutStuckHi			X3c::OutStuckLo			
X3c::OutStuckHi			X3a::OutStuckHi			
X3a::OutStuckLo			X1a::OutStuckHi			
X1a::OutStuckLo			R8::Open			
R11::Open			C7::Open			
C7::Short			C4::Open			
C4::Short			R4::Open			
R12::Open			X1d::OutStuckLo			
			X1d::OutStuckHi			
			X3b::OutStuckHi			
			X3b::OutStuckLo			

Report View | By Group

Precision: 4

Set Limits | Copy to Clipboard | Options | Save | Variation: No Faults | OK | Help



Test Design: Fault Tree with Entropy Selection

Group sequence and selection

Sequ...	Test Group	Co
✗ -1	T100at50ms	T10
✗ -2	T100at350ms	T10
✗ -3	T109	Ma
✗ -4	T110	Ma
✗ -5	T111	Ma
✗ -6	T112	Ma
✗ -7	T113	Ma
✗ -8	T114	Ma
✓ 9	T101	Ma
✓ 10	T102	Ma
✓ 11	T103	Ma

Sequence 9 ● Pass ○ Fail

Test	Entropy	Test Group
V(4) lo	0.5479	T101 V(4)
V(9) lo	0.5479	T101 V(9)
V(3) lo	0.5273	T101 V(3)
V(5) hi	0.4271	T101 V(5)
V(10) hi	0.3969	T101 V(10)
V(8) hi	0.3969	T101 V(8)
V(23) hi	0.3288	T101 V(23)
V(7) lo	0.3288	T101 V(7)

Fault Tree Structure

```
V(20) hi  59, 15
    V(26) hi  9, 6
        V(29) hi  3, 3
            V(30) lo  1, 2
        V(27) lo  2, 1
    V(24) hi  6, 3
        V(U4_2) lo  1, 2
    V(21) lo  1, 5
V(8) lo  54, 5
    V(7) hi  3, 2
    V(10) lo  2, 1
V(20) lo  41, 13
    V(23) hi  7, 6
        V(17) lo  3, 3
            V(18) lo  1, 2
        V(16) hi  2, 1
        V(3) hi  1, 1
    V(21) hi  1, 6
```

Test Description

Step 1
V(20) hi 59, 15
T101 V(20) hi
IF test fails
goto Step 2
ELSE
goto Step 9

Test Description:
Test Fails if measurement is above high limit
Measurement limits are:
High Limit = 1.000
Low Limit = -1.000

Fault Tree Selection: Mainline

Failure States: ○ Binary ● Tertiary ○ Histogram ○ Vector

45.2 Percent of Failures Detected

VII. FAULT TREE GENERATION

Figure 6 shows the Test Designer dialog that was used to build the diagnostic fault trees for the ALCPS. Test Designer can automatically generate the entire tree from the test pool; it can complete a partial fault tree or the test engineer can create the entire tree manually.

The first step in creating the fault tree is to select the test grouping (located the upper left of the dialog box). This selection is made by assigning a sequence number to each group of tests from which tests are to be drawn. Multiple test groups can be assigned the same sequence number when changing test setups from one test to the next in a sequence does not cause problems in the actual test process (e.g., excessive test setup times).

In the case of the ALCPS we wanted to run tests T1010, T1020, T1030, etc. in a sequential order, so we built the fault trees manually. The fault tree in Figure 6 depicts “go-line” tests as if the measurements were made at node 20 of the digital block. The actual observation point for the “go-line” tests is at the output of the On/Off Analog block. However, since the fault tree for the On/Off Analog block has a callout for digital block failure modes (recall the digital block was modeled as a voltage source in the On/Off Analog block), entry into the digital block fault tree can be made at the equivalent digital block “go-line” test entry point when the On/Off Analog block fault tree calls out the digital block failure modes. The linkage of fault trees is performed outside of the Test Designer environment and must be completed manually. The “go-line” tests of the digital block are linked instead of presented as independent trees because Test Designer can eliminate from consideration those failure modes which have been cleared from previous “go-line” tests.

To build the fault tree, a Sequence number is selected using the pull-down window on the left side of the dialog. An entropy value is computed for all tests in test groups that have this sequence number and the tests will then be displayed in the lower left of the dialog box in priority order - from best test for the active tree node to least desirable test in the context of diagnosing the highest failure rate components in the minimum number of steps. The test engineer has the option to select any test in this list. Tests that provide no new information regarding the failure modes have an entropy value of zero and are not displayed. The sequence number can be changed at any time during fault tree development in order to control which setups are used at various places in the fault tree.

The ALCPS tests used in the fault tree were selected based on their failure detection reliability. If a review of the detection characteristics for a recommended test showed “soft” detections for unknown failure modes (as indicated for failure mode Dcr14::Short in Figure 3), a different test was selected from the prioritized list. If there were no acceptable tests available, the detection characteristics of the listed tests were examined to see if limit changes would improve their reliability. With a pool of 416 tests for the digital block alone, it was easy to find reliable tests to use in the fault tree.

Fault trees for the simulated blocks of the ALCPS were manually linked together and into the pre-existing "go-line" test sequence using a word processor. These diagnostic fault trees were then passed to NADEP Cherry Point representatives for integration.

An interesting side note is that although the test pool was very large, there was substantial room for improvement in the detection characteristics of the digital block tests. Only 59.4% of these failures are detected by the "go-line" tests. Most of the non-detects occur in the RC delay circuits that filter and debounce the digital inputs from the aircraft. All "go-line" tests are designed to make measurements after the RC networks have reached steady state conditions and are not designed to detect open capacitors.

VIII. INTEGRATION

Results of inserting 15 faults and testing the ALCPS using the first three performance tests and the Test Designer generated diagnostic procedures are shown in Table 1. Results for 10 of the 15 faults were as predicted by Test Designer. The discrepancies between integration results and Test Designer predictions are discussed below.

The pass/fail limits for each test were assigned during the test synthesis phase by inspecting the "distance" of each simulated failure mode value relative to the nominal value for the measurement and selecting a limit between 5% and 10% of nominal up to a limit of 0.5 Volts. Subsequent Monte Carlo analyses reveals that these limits were well beyond those needed to accept 99.9% of unfailed UUTs as ready for issue. Furthermore, these pass/fail limits provided good discrimination between failure modes and resulted in reliable isolation of low failure count ambiguity groups. However, there was one unanticipated problem that caused many of the simulated nominal values for the On/Off analog block to be roughly 0.5 Volts above the nominal value observed on the actual ALCPS. Consequently, many pass/fail limits were also high by 0.5 Volts and tests failed when they should have passed. Recall that the input to the On/Off Analog block from the digital block was simulated using voltage sources. These voltage sources were assigned a nominal value of 11.91 Volts. On the ALCPS board used for integration, these CMOS outputs were measured at 11.2 Volts. This 0.7 Volt differential caused an adverse effect in the limit definition process. In addition, the power supply voltage for each of the integrated circuits was assumed to be 12.0 Volts. It was closer to 11.4 Volts on the actual ALCPS. This 0.6 Volt difference also contributed to the shift in test pass/fail limits. An alternative and preferable approach would have been to include tolerances on the voltage sources that represent the power supply and digital blocks and setting limits based on a Monte Carlo analysis. Subsequent analyses using more realistic tolerances for the voltage sources matched the performance observed when performing the tests on the CASS.

The fault tree that was generated with Test Designer was computed such that failure modes C20 short and Q4 collector/emitter short were isolated after detection by the T1010 "go-line" test. During integration, T1010 failed to detect C20 short and Q4 collector/emitter short. An inspection of the Test Designer Results dialog (similar to figure 3) indicated that both of these failure modes are "soft" detections for T1010 and may therefore be detected on some UUTs and not others. Expanding the limits of T1010 to ±1.0 Volts in order to make these failure modes non-detects for T1010 should eliminate this problem. Both failures can be subsequently detected by T1030.

Test Designer predicted that CR4 short and CR3 short were high probability detections for test T1020. Yet neither failure mode was detected by T1020 during fault insertion. This problem appears to be related to the architecture of the On/Off Analog circuitry. Op amp U6a of the On/Off Analog circuitry acts as a comparator. CR3 short and CR4 short

Number of cases	Result	Result agrees with model
8	fault detected and isolated successfully; that is, failed the performance test as expected and fault isolated to the correct component	Yes
2	fault not detected	Yes
4	fault detected (a function of the performance test) but not isolated correctly. Component was in Fail Ambiguity Group for a performance test that passed and was not in Fail Ambiguity Group for the subsequent performance test that failed. Test Designer predicted two of these four faults were "soft" detects for the performance tests and may not be detected for some UUTs (see text).	No
1	fault not detected	No

failure modes shift both the positive and negative inputs to the comparator away from their nominal values by a small amount; the negative moves at a faster rate than does the positive input. For some tolerance accumulations of components feeding the comparator, either of these failure modes will cause the comparator to switch states and will be “hard-over” detections for T1020. For other tolerance accumulations, the comparator will not switch states and the T1020 measurement will yield its nominal, unfailed value. Additional “go-line” tests should be devised for the ALCPS in order to detect CR3 short and CR4 short in the event they are not detected by T1020.

IX. LABOR EXPENDITURES

Following are manhour estimates for each of the tasks leading to integration of the fault tree that was prepared using Test Designer. The listed hours are within ±25% of the actual expended hours.

Circuit Parsing		6.0 hours
Schematic Entry		
	Digital block	4.0 hours
	On/Off Analog block	3.5 hours
	Dimmer block	3.0 hours
Measurement Definition		
	Digital block	3.0 hours
	On/Off Analog block	1.0 hours
	Dimmer block	0.5 hours
Simulation (Includes reruns due to circuit model errors, stimuli changes, problem solving, etc.)		
	Digital block	24.0 hours
	On/Off Analog block	10.0 hours
	Dimmer block	16.0 hours
Test Synthesis		
	Digital block	0.5 hours
	On/Off Analog block	2.0 hours
	Dimmer block	2.5 hours
Fault Tree Generation		
	Digital block	0.5 hours
	On/Off Analog block	1.0 hours
	Dimmer block	1.0 hours

Once these six steps of the process have been completed, changes can be made very quickly. To illustrate, consider that the first set of diagnostics that were generated for NADEP Cherry Point integration contained an error in the stimuli for test T1020. This error invalidated much of the logic in the fault tree so the simulation showed many detections that didn’t exist and missed a number of detections that would exist (actually, the simulator results were correct based on the stimuli defined in the model; unfortunately, the model didn’t agree with the implementation). This error was discovered at 3pm. By 9pm that evening, a completely new set of diagnostic tests for the ALCPS had been synthesized, sequenced and e-mailed to the Chesapeake Test Integration Facility.

X. LESSONS LEARNED

A. General Comments

1) The simulation techniques which are employed by Test Designer act as a “force multiplier” for development of diagnostics for analog and mixed signal circuits. Over the course of 80 hours it was possible to generate 619 tests, determine the failure mode detection characteristics of each test against a failure universe of 204 failure modes, and sequence a subset of these tests into an effective diagnostic fault tree.

2) Simulation is an effective method for identifying problem areas in fault detection. The beta testing of Test Designer surfaced problems in two ways; first, by identifying the low probability fault detections for individual tests and second, by providing an infrastructure for further circuit analysis when integration results did not match simulator predictions.

3) Reasonable simulation times can be achieved by parsing the circuit into separate subcircuits and linking the simulation results through the use of voltage and current stimuli.

4) Assumptions regarding linkage of separate simulations through the use of voltage or current sources should be validated on actual circuitry. Monte Carlo techniques should be employed to ensure that test limits are valid over the full range of values that these sources can assume.

5) The simulation model should be validated against actual circuit operation in the no fault condition prior to developing diagnostic tests.

B. NADEP Cherry Point Comments.

As a first time user of Test Designer the engineer came on line very quickly. There was a short learning curve to get an understanding of what information is required.

Accounting for achieved detection and isolation of failure modes is a difficult problem for the TPS developer. While accountability is easy for the developer to achieve in simple circuit analysis, it becomes considerably more difficult as the circuit complexity increases. Test Designer provided an advantage to the ALCPS TPS developer in that it kept track of detection and isolation of failure modes during generation of the fault tree.

Direct application of Test Designer can be made to all stages of avionics design and test program development, and also implemention of design for testability during avionics design. Models developed during this process would be of great value to subsequent test program development.

Test Designer provides the engineer a fault matrix in the form of a Fault Tree .tdf text file. The .tdf file provides an accounting of components and associated fault modes detected by the defined test. This output provides a list of detectable, undetectable, and destructive fault modes. The information required to develop the Fault Accountability Matrix (FAM) Table [7] as required in the Navy's Red Team Package resides in this .tdf file.

Once confidence is achieved in the circuit model it can be used by both contractor and customer to simplify and shorten the final product validation and acceptance process. Upon delivery of the TPS to the Fleet, the models would be maintained by In-Service-Engineering in order to provide guidance in avionics changes and upgrades to test program capability.

Test Designer is not a replacement for engineering expertise. It does however allow the engineer to focus the TPS development efforts in a fashion that targets high failure areas for additional diagnostics. After becoming familiar with its application, we anticipate that the engineer will include Test Designer as an integral part of the TPS development process.

REFERENCES

[1] L.W. Nagel and D.O. Pederson, *Simulation program with integrated circuit emphasis*, ERL Memo No. ERL-M520, University of California, Berkeley, May 1975

[2] L.G. Meares, *New Simulation Techniques Using SPICE*, IEEE APEC, 1986, pp.198-205

[3] Steven M. Sandler, *SMPS Simulation with SPICE,* Mc-Graw-Hill, 1997

[4] B. Johnson, T. Quarles, A.R. Newton, D.O. Pederson, A. Sangiovanni-Vincentelli, *SPICE 3F User's Guide*, University of California, Berkeley, Oct. 1992

[5] F. Cox, W. Kuhn, J. Murray, S. Tynor, *Code-Level Modeling in XSPICE*, Proceedings of the 1992 Intl. Syp. on Circuits and Systems, May 1992, IEEE 0-7803-0593-0/92

[6] Harry H. Dill, "A Comparison of Conventional and Inference Model Based TPS Development Processes", AutoTestCon '95 Proceedings, pp 160-168

[7] CASS Red Team Package data item DI-ATTS-80285B, Fig. 1 - SRA/SRU Fault Accountability Matrix Table, pg 11