Circuit Simulation Using EPC SPICE Models

Robert Beach, and Aydin Babakhani; Efficient Power Conversion Corporation

EPC’s enhancement mode gallium nitride (GaN) power transistors are a new generation of power switches offering unsurpassed performance over silicon power MOSFETs in switching speed and conduction losses with superior thermal characteristics. An accurate circuit and device model is a valuable tool for developing new topologies, building successful designs, and shortening time to market. This article describes the status and use of EPC device models, and illustrates some important considerations when incorporating EPC GaN devices into a circuit model.

Status of current models:

Years of refinement have gone into the development of silicon power MOSFET device models. Early attempts were based on a fitting of device behavior with functions of the approximate shape, polynomials of many orders, or simple look-up tables. Recent trends have been toward solving multidimensional electrostatic conditions from the basic underlying physics—a daunting task—but great simplification in the final solution has been achieved using this approach. Only a few papers have been published on the development of spice models for GaN. In addition, GaN has many new properties, such as spontaneous and piezoelectric polarization, that have only recently begun to be incorporated into physics based models. EPC V091 models presented in this paper, are a hybrid of physics-based and phenomenological functions to achieve a compact spice model with acceptable simulation and convergence characteristics. Temperature effects have also been included for conductivity and threshold parameters. Although quantum-based effects have not been incorporated, the models accurately reproduce the basic response of the devices under circuit operation conditions. A number of improvements are under development to include field dependent mobility and gate injection current. Models with these, and other, improvements will be made available regularly from the EPC web site (www.epc-co.com).

Structure of EPCs GaN Power Transistors and Model

EPC’s GaN power transistor operation is very similar to that of the conventional silicon power MOSFET. The basic structure of EPC’s GaN transistor is shown in fig 1 below.

The gate (G), source (S) and drain (D) terminals, are defined in the same way as they are in silicon power MOSFETs. A positive voltage higher than the threshold between the gate and source will turn the device ON, and a voltage lower than threshold will turn the transistor OFF. An electron generating layer is incorporated to provide a channel to direct the current from the drain to the source when the gate is activated. Once ON, the device can conduct current in either direction form drain to source or visa-versa.

The DC current characteristics of the devices have been implemented similar to a level 3 MOSFET model. The non-linear current response is modeled as the product of a saturation current that is gate to source voltage dependent with a drain to source voltage dependent shaping function. Some improvements to the accuracy of the sub-threshold region have been included since this region plays an important role in the negative drain bias operation of the device in circuit. Figure 2 shows an example of the model’s device transfer and output characteristics.

Unlike standard silicon MOSFETs, the GaN based device does not have a source-connected p-type...
region under the gate. In place of the p layer is a highly insulating layer of GaN. This difference results in an interesting feature for the conduction characteristics of the device. It is nearly symmetric. A positive gate to drain voltage will enhance the channel as well as a positive gate to source voltage. The device model accounts for this by having two parallel FETs connected in opposing orientation, i.e. the source of the first FET is connected with the drain of the second, and vice-versa). Figure 3 shows the forward vs reverse output curves of an EPC1001 part. The main difference between the forward and reverse directions is the reduction in saturation current due to the added series resistance of the extended drain-side channel. Each FET is restricted to operation in its forward state by a step function in drain to source voltage, such that only one contributes to the circuit at a time.

Capacitance models were developed and fit to data based on the underlying device geometry.

As shown in Figure 5:

1) Drain current (I_D): I_D is a nonlinear function of internal nodes D, G, and S. For V_D>V_S: I_D>0 and for V_D<V_S: I_D<0
2) Gate-source capacitance (C_{GS}): C_{GS} is a nonlinear function of internal nodes D, G, and S
3) Gate-drain capacitance (C_{GD}): C_{GD} is a nonlinear function of internal nodes D, G, and S
4) Drain-source capacitance (C_{DS}): C_{DS} is a nonlinear function of internal nodes D and S
5) Drain parasitic resistance (R_D): R_D is a constant resistance that depends on the device and package parasitic resistances
6) Source parasitic resistance (R_S): R_S is a constant resistance that depends on the device and package parasitic resistances
7) Gate parasitic resistance (R_G): R_G is a constant resistance that depends on the device and package parasitic resistances

Figure 4 diagrams the elements of the capacitance used in the model. The elements consist of constant metal to metal capacitances in parallel with voltage-dependent functions to model depletion in the channel layers. Shown in the figure are C_{G1}, the gate metal to source metal field plate capacitance, C_{G2}, the gate metal to source side channel, C_{G3}, the gate to drain side drift region, C_{G4}, the source to drain side drift region capacitance under the field plate, and C_{G5}, the source to drain region fringe capacitance. The gate to channel region is shown as C_{G3}. C_{G1}, as well as other metal to metal capacitances not shown in the figure were treated as constants and fit using measured capacitance-voltage data. C_{G1} is modeled as a Gaussian distribution of parallel MOS capacitors, while C_{G2} is a fringe capacitance that is dependent on the depletion width from the field plate edge. The gate to channel capacitance, C_{G3}, is treated similarly; however, the assignment of this capacitance requires some care. Due to the symmetric nature of the device, the channel capacitance can be gate to source, gate to drain, or shared, depending on the node conditions. An instance of this is when the drain swings negative during the dead time of a buck converter. The gate to drain capacitance will be large as the drain swings past V_{TH}, filling the gate to channel capacitance. Many SPICE programs do not allow the use of higher level functions when representing capacitors. To ensure flexibility of the model file, the capacitor functions have been converted to high order polynomial fits of the capacitance vs voltage. In appendix A, a copy of the EPC1001_V091 model is included for reference. The three capacitor model functions listed as C_{G1}, C_{G2}, and C_{G3} are the resulting polynomial expressions after conversion.

The nonlinear capacitors and current sources are combined to produce the time dependent electrical behavior of the device. A device circuit schematic is shown in Figure 5. The main components are the voltage controlled current source, I_D, Nonlinear capacitors, C_{G1}, C_{G2}, and C_{G3}, and input resistors R_D, R_S, and R_G.

Using the Model

Each device model is provided as a sub-circuit written in SPICE.

The model is defined using SPICE's .subckt command and can be copied directly into a SPICE netlist, or loaded in a SPICE simulator using the command, .include.
Example:

.include (directorypath)/EPC1001_V091.sp
XEPC1001_1 drain gate source EPC1001

In the above example, directory path is the location in which the model file was saved, and can be omitted if the file is located in the default model folder of your SPICE compiler. XEPC1001_1 could be any arbitrary name with the starting letter ‘X’. The model assumes that the first node is a drain node, the second one is a gate node, and the third one is a source node. At right is the SPICE net list used to generate the data of figure 3. Three resistors are connected to the device and one voltage source. Rsgrd is a small resistor used to probe current, Rgs is connected between gate and source nodes and Rgd is connected between gate drain nodes. They are used to form a voltage divider, setting gate voltage to half of the drain voltage. Vdd is connected between drain and gnd (ground). A DC analysis is performed between -5 and 5 V with .05 V steps, and the current through Rsgrd is sent for plotting.

Example Circuit and Comparison

As a demonstration of device model and circuit considerations, a simple circuit was built and tested to compare device performance with that predicted by the model.

The circuit consisted of a voltage source charging a 13 µF capacitor through a 10 kΩ resistor used to isolate the voltage source from the device under test. The FET is driven with a 5 V pulse and the capacitor is discharged through a 0.8 Ω resistor and the device with a 0.1 Ω stray resistance. Of particular interest are the gate and drain side inductances. Although not intentionally added to the circuit, stray inductance in the PCB plays a key role in the behavior of the circuit. Figure 6 shows the effect even small inductance can have on oscillations given the fast switching time of the GaN parts. Ringing of greater than the input voltage occurs with drain loop inductance of less than 2 nH. Example SPICE code is shown at right. This can be used to investigate the influence of stray inductance and resistance from PCB on the switching wave form in this simple circuit. To use the example, first uncomment the lines in blue and run. This simulates the charging of the large supply cap, which takes a long time and only needs to be done once. Re-comment blue lines, and uncomment green lines. The file is then ready for running.

*first line (The first line is always ignored by SPICE)  
.include EPC1001subckt_V091.sp
Rin in 2 10k
Cap_1 2 3 13u
Resr 3 Gnd .1
Rl2 4 .8
Ll2 4 drain .6n
XEPC1001_1 drain gate source EPC1001
Ls source source2 .3n
Rs source2 Gnd .001
Ldrive gate drive1 2n
Rdrive drive1 drive .1
* Include the device model
* Include the text within file EPC1001subckt_V091.sp
* An instance of EPC1001 is placed between drain, gate and source nodes
* A resistor between source and ground
* A resistor between gate and source
* A resistor between gate and drain
* A voltage source is set between drain and ground
* Request for DC solution for Vdd from -5 to 5 V in .05V steps
* Send the current of Rsgrd from DC solution to be plotted

* include EPC1001subckt_V091.sp
XEPC1001_1 drain gate source EPC1001
Rsgrd source gnd .0001
Rgs gate source 100
Rgd gate drain 100
Vdd drain gnd
.dc Vdd -5 5 .05
.print DC (Rsgrd)

********* Simulation Settings - Additional SPICE commands *********
.include EPC1001subckt_V091.sp
* An instance of EPC1001 is placed between drain, gate and source nodes
* A resistor between source and ground
* A resistor between gate and source
* A resistor between gate and drain
* A voltage source is set between drain and ground
* Request for DC solution for Vdd from -5 to 5 V in .05V steps
* Send the current of Rsgrd from DC solution to be plotted

********* Simulation Settings - Analysis section *********
.Vin in Gnd pulse(0 12.0 10u 1001m 1002m)
.Vin in Gnd 12
.Vdrive_1 drive Gnd 0
.Vdrive_1 drive Gnd pulse(0 500n 2n 6n 150n 300n)
.save file=testboardnodesetv1 type=nodeset time=1000m
.load file=testboardnodesetv1
.print tran v(gate,source) v(drain,source)

Figure 6: Schematic of demo circuit number 1.
Comparison of the simulated results for the demo circuit show reasonable correlation with the measured values. Although not perfect, overshoot and ringing is qualitatively reproduced. Figure 8 shows the overlay of the gate and drain voltages vs time for the measured and simulated circuit.

**Delay line**

Another example circuit is a scaled model of the device used to make a delay line: A pull up resistor in series with a FET to make an inverter, repeated to form a delay line. A schematic is shown below in figure 9. The purpose of this example is to demonstrate the first pulse effect that can result from this model. It is important to simulate through the startup conditions to get an accurate idea of performance when operating continuously in the circuit.

The delay line is for a 30-FET series. The input voltage was set to 5 V, and a 5 V pulse train with 5 ns rise and fall time is sent to the first gate. In figure 10, there is a significant difference between the shape and delay times of the first and following pulses. This is caused by some internal charge redistribution when moving away from the DC solution.

It is recommended to use a soft start approach to simulating circuits using the GaN models: Ramping voltage sources from 0 V, and increasing AC signal amplitudes over several cycles to avoid this type of first switch result. As can be seen in the figure, delay times are approximately 180 ps/FET, and the switch time has narrowed significantly from the 5 ns input pulse to <1 ns. Inductance has been ignored in this example, but the FETs are fundamentally capable of high switching speeds.
Simulation of a 48 V - 1.0 V, 5 A Converter at 250 kHz

Figure 11 shows the schematic of this buck converter. A number of parasitic resistances are added to account for the PCB trace, inductor, and package resistance. The following SPICE script is used to simulate this circuit.

In this SPICE code, the optional method of Gear is used to improve the stability. Please refer to Appendix B for a detailed description of the Gear method.

The .measure commands used in the SPICE code calculate the delivered load power as well as the total DC power. These values can be found at the end of the .out file after completion of the transient simulation. Based on this simulation the average load current is 5A, the output dc power is 5.13 W, the total input dc power is 5.88 W, and the conversion efficiency is 87%. Measured efficiency for this circuit was 85.9% in reasonable agreement.

Figure 11: Schematic of a buck converter

Pulse widths for 5 A output

Figure 12: Switch node voltage vs time obtained from a) simulated 48 V - 1 V buck converter and b) measured data for switch node voltage and output voltage during operation of 48 V - 1 V buck converter.
with simulated values. Figure 13 is a graph of the above conversion efficiency over a range of output current levels. For a first generation device model, there is very good agreement between simulated and measured efficiency. These results show the remarkable performance of the EPC GaN parts. Obtaining greater than 85% efficiency in high frequency 48 V - 1 V converters is well beyond state of the art Si capability.

Cut-Off Frequency Simulation of the EPC GaN Device

Figure 14 shows the schematic cut-off frequency simulation. Cut-off frequency of a transistor is the frequency at which the current gain becomes one. The SPICE script at right is used to simulate this circuit.

Based on these simulations, the maximum cut-off frequency of EPC1001 GaN device is 4.6 GHz.

Conclusions

Device simulation for enhancement mode GaN devices from EPC was presented with an eye toward basic understanding of the available models, while highlighting some unique features of the devices and models. The components of the model were presented, and where appropriate, the underlying device geometry and function were discussed to help the user understand the differences from standard Si models. Although early in the development, models to date perform well in replicating the basic circuit performance of the EPC devices. A few circuit models were used as examples to show the effect of stray inductance and simulation setup. Stray inductance was found to play an important role in circuit oscillation, and needs to be considered during device simulation, and PCB layout. A soft start approach was shown to improve the accuracy of the simulation results, and should be incorporated into simulation programs. A comparison between simulated and measured circuit performance was shown for a few circuits, and qualitative accuracy was obtained.
Appendix A: SPICE netlist for EPC1001 device V091

*First line

.SUBCKT EPC1001 N_2 N_1 N_0 T=25 aGc1=6.34e-10 aGc2=1.85 aGc3=1.156 aGc4=1.7 dgs1=4.3e-7 dgs2=2.6e-13 dgs3=8 aWg=1077 aTc=.05 aStc=0.005

rd N_2 drain .0001
rs source N_0 .0001
rg N_1 gate .6

gswitch drain source cur='aWg*(1-aTc*(T-25))/1077*18*0.5*(1+tanh(1000*v(drain,source)))*(v(gate,source)+2*aGc3*log(cosh((v(gate,source)-aGc2)/(2*aGc3)))/cosh((aGc2)/(2*aGc3)))*(1-exp(-aGc4*(1-aStc*(T-25)))*v(drain,source)))'
gswitchr drain source cur='aWg*(1-aTc*(T-25))/1077*18*0.5*(1+tanh(1000*v(source,drain)))*(v(gate,drain)+2*aGc3*log(cosh((v(gate,drain)-aGc2)/(2*aGc3)))/cosh((aGc2)/(2*aGc3)))*(1-exp(-aGc4*(1-aStc*(T-25)))*v(source,drain)))'
ggdiode gate source cur='0.5*aWg/1077*(dgs1*(exp((v(gate,source))/dgs3)-1)+dgs2*(exp((v(gate,source))/dgs4)-1))'
ggddiode gate drain cur='0.5*aWg/1077*(dgs1*(exp((v(gate,drain))/dgs3)-1)+dgs2*(exp((v(gate,drain))/dgs4)-1))'
gcd2 gate2 drain cur='0.5*aWg/1077*(dgs1*(exp((v(gate,source))/dgs3)-1)+dgs2*(exp((v(gate,source))/dgs4)-1))'

cg12 gate gate2 5n*aWg/1077
rg12 gate gate2 1k
cgs gate source poly '7.628006667448896e-010*(aWg/1077)-4.408939083898130e-011*(aWg/1077)+2.247410988968877e-010*(aWg/1077)+1.583482512913506e-010*(aWg/1077)+1.019108516596830e-010*(aWg/1077)+1.283449181368e-010*(aWg/1077)+3.9088390674128e-012*(aWg/1077)+3.4940576832156e-013*(aWg/1077)+8.243562765756245e-014*(aWg/1077)+2.462476595030476e-014*(aWg/1077)+5.90496283932310e-015*(aWg/1077)-1.271029589443909e-015*(aWg/1077)-4.80525028032811e-017*(aWg/1077)+6.995190873289643e-018*(aWg/1077)-1.315174814787596e-018*(aWg/1077)+2.53651174799662e-020*(aWg/1077)+4.0348254277123e-022*(aWg/1077)+6.88941399069026e-024*(aWg/1077)+8.08978966881882e-026*(aWg/1077)

cgd drain source poly '1.048094953309892e-009*(aWg/1077)-8.78953882860264e-012*(aWg/1077)-1.36334321380894e-012*(aWg/1077)+5.36177486203130e-014*(aWg/1077)+6.4029177341298e-016*(aWg/1077)+6.48380282336875e-022*(aWg/1077)+3.66141024640568e-023*(aWg/1077)+2.32728995426371e-025*(aWg/1077)+1.51163970135620e-026*(aWg/1077)-7.4160315063339e-029*(aWg/1077)+4.7669533795129e-030*(aWg/1077)+6.02928424363992e-032*(aWg/1077)+8.393338235783209e-034*(aWg/1077)+2.5967852827443e-036*(aWg/1077)+1.40466102001129e-037*(aWg/1077)+2.32926500178870e-040*(aWg/1077)+1.05198369073076e-041*(aWg/1077)+1.47466230611065e-044*(aWg/1077)+6.283401076826703e-046*(aWg/1077)+5.40998812240192e-049*(aWg/1077)+2.19132452124918e-052*(aWg/1077)+8.74250687843623e-054*(aWg/1077)+3.387065085419279e-055*(aWg/1077)

cds drain source poly '1.048094953309892e-009*(aWg/1077)+8.78953882860264e-012*(aWg/1077)-5.36177486203130e-014*(aWg/1077)+2.912807254237602e-015*(aWg/1077)+1.73507791923614e-016*(aWg/1077)+3.68101611042899e-018*(aWg/1077)+2.6009539198188e-019*(aWg/1077)+3.0575186477400e-019*(aWg/1077)+2.3376367530291e-022*(aWg/1077)+1.76623972849919e-024*(aWg/1077)+1.382697582328767e-025*(aWg/1077)+7.342313034375557e-028*(aWg/1077)+5.65326532767369e-029*(aWg/1077)+2.24421376864322e-031*(aWg/1077)+1.643128169139548e-032*(aWg/1077)+5.101113648226566e-035*(aWg/1077)+3.43440683154685e-036*(aWg/1077)+8.65424503707054e-039*(aWg/1077)+5.21633518517069e-040*(aWg/1077)+1.09060441529558e-042*(aWg/1077)+5.65702606911167e-044*(aWg/1077)+1.005704150025597e-046*(aWg/1077)+4.2805062320799e-048*(aWg/1077)+6.587984884820379e-051*(aWg/1077)+2.145441078734759e-052*(aWg/1077)+2.901553225388414e-055*(aWg/1077)+6.389923790968234e-057*(aWg/1077)+7.00774690118621e-060*(aWg/1077)+8.5933699904114e-062*(aWg/1077)-9.30356253437635e-065*(aWg/1077)

.ends
Circuit Simulation, SPICE Models

Appendix B: Gear’s BDF Method

T-SPICE’s alternate method for transient analysis uses Gear’s backward differentiation formulas (BDF). In this method, the time derivative of charge in the KCL equations is replaced by an approximation involving the solution at the last few time points. The first-order BDF method uses only one previous time point, and it is equivalent to the well-known Backward Euler method. In this method, the discretization error is a linear function of the step size. The second order method uses two previous time points, and its discretization error is proportional (for small time step sizes) to the time step size squared. In general, the kth order BDF method uses k previous time points.

T-SPICE uses a variable-step-size, variable-order, and variable-coefficient implementation of the BDF method. T-SPICE automatically adjusts the time step size and BDF order (between 1 and 4) to minimize the number of time steps required to meet the given error tolerances. The maximum order used can be adjusted with the maxord option. The variable-coefficient implementation was chosen over the fixed coefficient and fixed-leading-coefficient methods because it offers the best stability properties, especially with frequently varying time step sizes. At each time step, the BDF discretization results in a nonlinear system of equations (representing KCL) which is solved iteratively as described above. If the iteration succeeds, the discretization error is examined (by comparison with an explicit predictor). For example, in the order 1 case, the difference between the Forward Euler predictor and the computed BDF (Backward Euler) solution provides a bound on the discretization error. If the error is within the prescribed tolerance (defined by chargetol and relchargetol), the step is accepted, and the error is used to adjust the step size for the next time step. If the error is too large, the time step is rejected and reattempted with a smaller step size. This will produce answers which approach a more stable numerical solution. Gear integration often produces superior results for power circuitry simulations, due to the fact that high frequency ringing and long simulation periods are often encountered.