Circuit Simulation Using EPC Device Models

Robert Beach, Aydin Babakhani and Robert Strittmatter; Efficient Power Conversion Corporation

EPC's enhancement mode gallium nitride (eGaN®) 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 eGaN 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 piezo-electric 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 eGaN power transistors operation is very similar to conventional, enhancement mode silicon power MOSFETs. The basic structure of EPC's eGaN FET is shown in Figure 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 from drain to source or visa-versa.

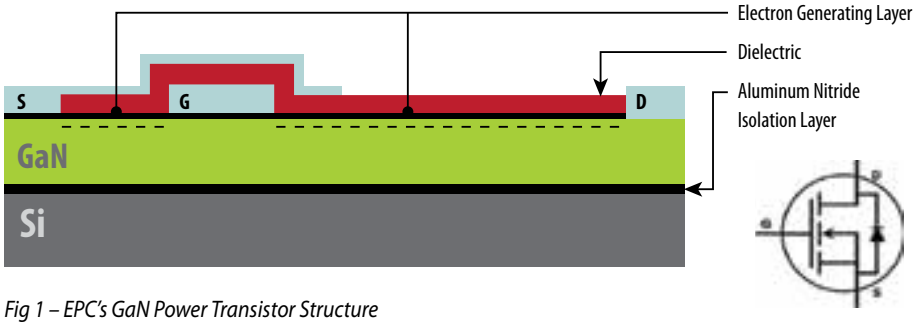


Fig 1 – EPC's GaN Power Transistor Structure

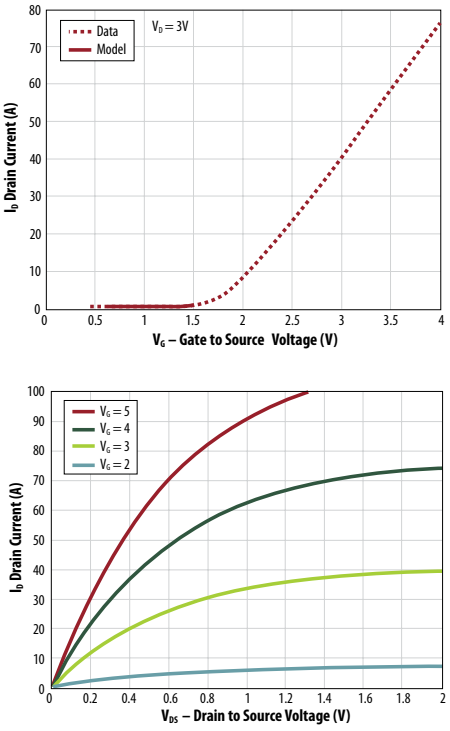


Figure 2: Transfer and output curves for EPC1001 device model

The DC current characteristics of the devices have been implemented similar to a level 3 MESFET 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 subthreshold 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.

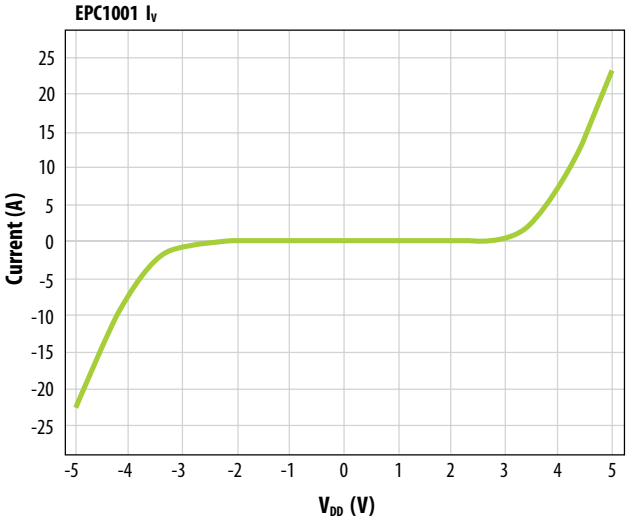


Figure 3: Forward and reverse current with V_G at V_{DD}/2, showing the symmetric nature of the device.

Unlike standard silicon MOSFETs, the eGaN 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.

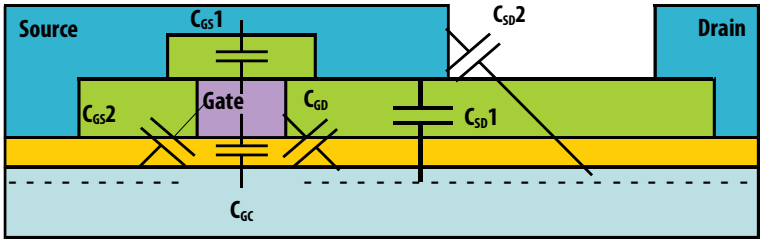


Figure 4: Schematic of capacitance sources

Capacitance models were developed and fit to the data based on the underlying device geometry. Figure 4 illustrates the physical locations of the lumped capacitor elements used in the model. The elements consist of constant metal to metal capacitances in parallel with voltage-dependent functions to model depletion characteristics in the channel layers. Shown in the figure are:

- $C_{GS}1$, the gate metal to source metal field plate capacitance
- $C_{GS}2$, the gate metal to source side channel
- $C_{GD}1$, the gate to drain side drift region
- $C_{SD}1$, the source to drift region capacitance under the field plate
- $C_{SD}2$, the source to drift region fringe capacitance
- C_{GC}, the gate to channel capacitance

$C_{GS}1$, as well as other metal to metal capacitances not shown in the figure, were treated as constants and fit using measured capacitance-voltage data. $C_{SD}1$ is modeled as a Gaussian distribution of parallel MOS capacitors, while $C_{SD}2$ is a fringe capacitance that is dependent on the depletion width from the field plate edge. The gate to channel capacitance, C_{GC}, 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 Vth, filling the gate to channel capacitance. The non-linear capacitance characteristics of eGaN devices are modeled using semi-empirical fits to measured capacitance vs voltage data. Rather than using high order polynomials, the models employ a sum of sigmoid (or Fermi) functions. These functions provide a close fit to the data, and have better stability and convergence properties during simulation. In Appendix A, a copy of the EPC2001 LTSPICE model is included for reference. The formulae for the three capacitor model functions (labeled as C_{GS}, C_{GD}, and C_{SD}) are seen at the bottom of the model listing. To take advantage of behavioral modeling built into most modern SPICE simulators, the non-linear capacitance is implemented as a voltage dependent charge source.

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

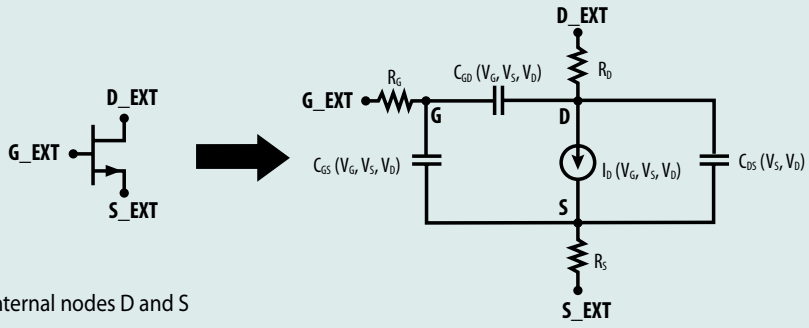


Figure 5: EPC's GaN model

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, ID, Nonlinear capacitors, C_{GD}, C_{GS}, and C_{SD}., and input resistors R_S, R_D, 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. XECP1001_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. Below is the SPICE net-list used to generate the data shown in figure 3. The simulation comprises three resistors that 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. These resistors 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 by varying the VDD voltage from -5 through 5 V in .05V steps, and the current through Rsgrd is sent for plotting using the .print command.

Example Circuit and Comparison

As a demonstration of the 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 13uF cap through a 10kOhm resistor used to isolate the voltage source from the device under test. The FET is driven with a 5V pulse and the capacitor is discharged through a 0.8 Ohm resistor and the device with a 0.1 Ohm stray resistance. Of particular interest are 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

```
* test circuit
.include EPC1001subckt_V091.sp      * Include the text within file EPC1001subckt_V091.sp
XEPC1001_1 drain gate source EPC1001 * An instance of EPC1001 is placed between drain, gate and
                                    * source nodes
Rsgrd source gnd .0001              * A resistor between source and ground
Rgs gate source 100                 * A resistor between gate and source
Rgd gate drain 100                  * A resistor between gate and drain
Vdd drain gnd                       * A voltage source is set between drain and ground
.dc Vdd -5 5 .05                    * Request for DC solution for Vdd from -5 to 5 V in .05V steps
.print DC i(Rsgrd)                  * Send the current of Rsgrd from DC solution to be plotted
```

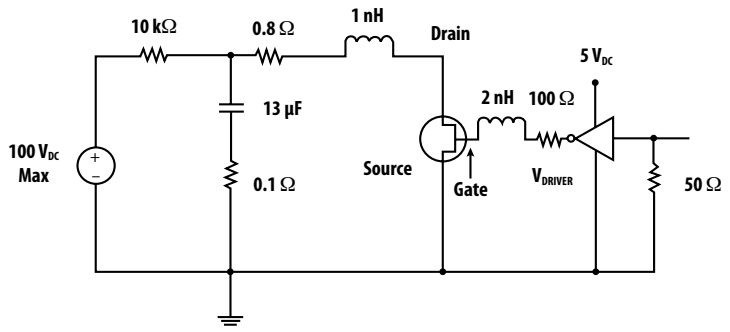


Figure 6: Schematic of demo circuit number 1.

```
*first line (The first line is always ignored by SPICE)
.include EPC1001subckt_V091.sp      * Include the device model
Rin in 2 10k                        * an isolation resistor between input voltage and the
                                    rest of the circuit
Cap_1 2 3 13u                       * A cap between node 2 and node 3 used as current
                                    supply when device is turned on
Resr 3 Gnd .1                       * Parasitic resistance of capacitor and PCB
Rl2 2 4 .8                          * Main resistor between capacitor and device
Ll2 4 drain .6n                     * Stray inductance in circuit from PCB, the value
                                    can be modified to look at the effect on switching
XEPC1001_1 drain gate source EPC1001 * The EPC1001 device reference
Ls source source2 .3n               * A source side inductance that can also be varied
Rs source2 Gnd .001                 * A small source side PCB resistance
Ldrive gate drive1 2n               * Gate loop inductance that can be varied to look at
                                    the drive side inductance effect on switching
Rdrive drive1 drive .1              * Added driver resistance
* Uncomment lines in blue and run, this creates a file with the DC starting solution for the circuit
* It 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
* VIN in Gnd pulse(0 12 0 10u 10u 1001m 1002m )  * ( Uncomment this for first run)
* VIN in Gnd 12                                  * (Uncomment this for other runs)
* VDriver_1 drive Gnd 0                          * ( Uncomment this for first run)
* VDriver_1 drive Gnd pulse(0 5 100n 2n 6n 150n 300n )  * (Uncomment this for other runs)
* save file=testboardnodesetv1 type=nodeset time=1000m  * ( Uncomment this for first run)
*.load file=testboardnodesetv1                  * (Uncomment this for other runs)
.print tran v(gate,source) v(drain,source)
********* Simulation Settings - Analysis section *********
.op
* tran 10u 1000m                                *( Uncomment this for first run)
*.tran 1n 300n                                  * (Uncomment this for other runs)
********* Simulation Settings - Additional SPICE commands *********
.end
```

the effect even small inductance can have on oscillations given the fast switching time of the GaN parts. Voltage ringing of amplitude greater than the input voltage occurs with drain loop inductance of less than 2nH. Example SPICE code is also shown. This can be used to investigate the influence of stray inductance and resistance from PCB traces and pads 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, and then creates a node-set file with the solution. This reduces the time required for running the switching simulation later. Second, re-comment the blue lines, and uncomment the lines in green text. The file is then ready for simulating the switching of the circuit. The rise time of gate drive, parasitic inductances of drain side, gate side, and source side, and various resistances can be modified to look at effect on circuit performance.

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 versus time for the measured and simulated circuit.

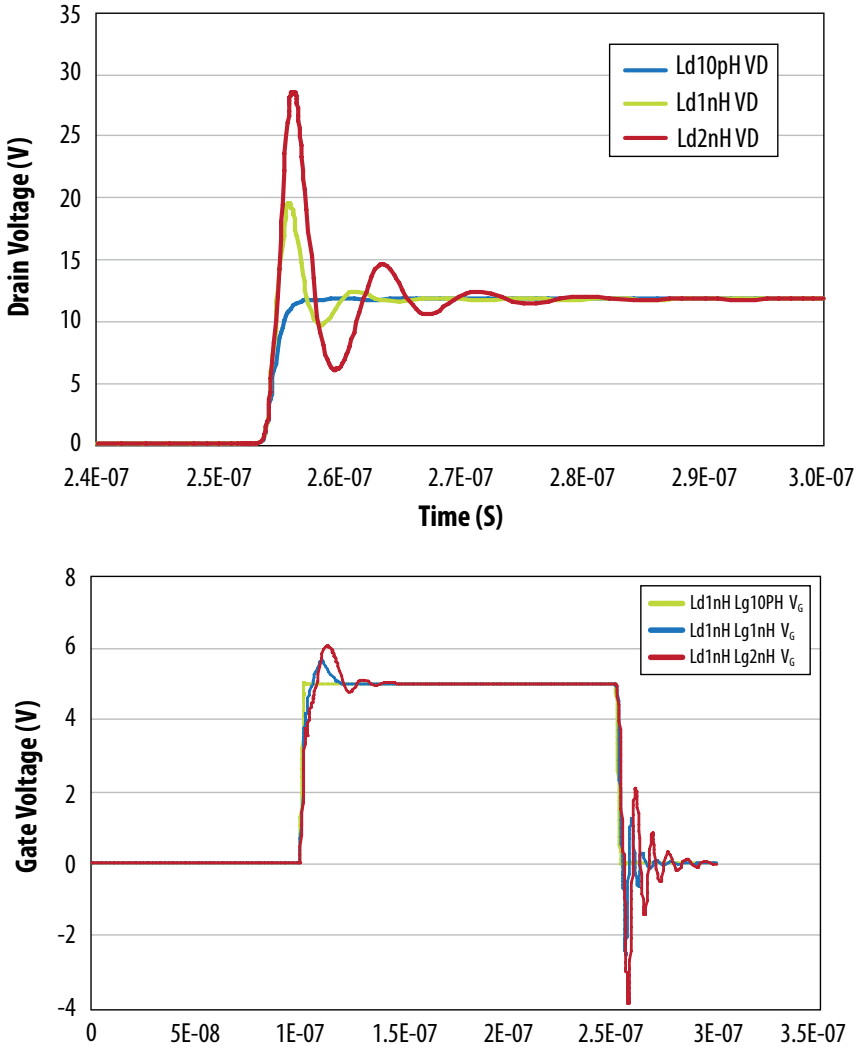


Figure 7: A) Comparison of drain node oscillations vs different stray inductance levels. B) gate node oscillations vs stray gate inductance.

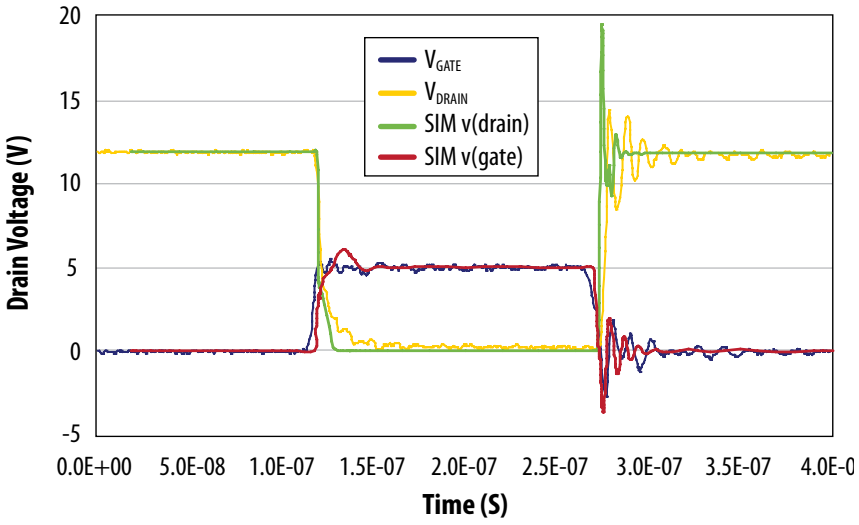


Figure 8: Comparison of simulated and measured demo circuit 1.

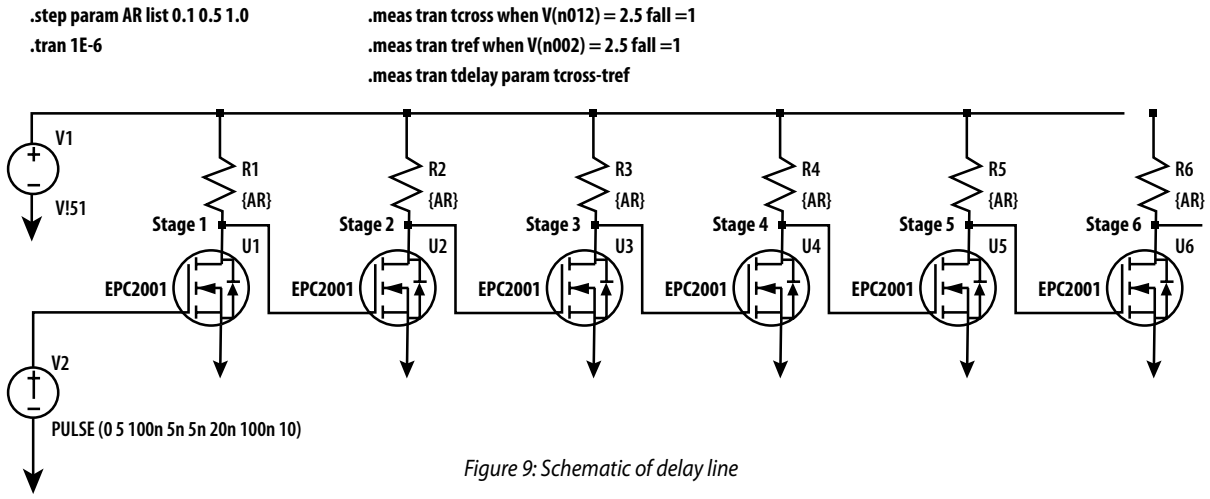


Figure 9: Schematic of delay line

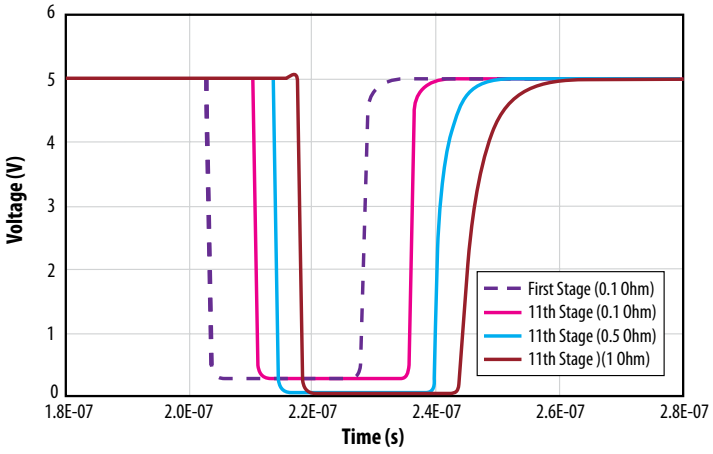
Delay line

An example simulation with a large number of eGaN FETs is the delay line shown in Figure 9. A pull-up resistor in series with an EPC2001 FET forms a single inverter stage. These stages are then cascaded multiple times. An input pulse at the gate of the first stage will propagate down the line, with each stage making a contribution of time delay.

The schematic contains 30 stages, although they are not all shown in the figure. The common drain voltage to the delay line was set to 5V (voltage source V1). A 5V pulse train is sent to the first gate (source V2), and a transient analysis is performed to simulate the pulse as it travels down the line. From the point of view of simulation stability, there are two features of the input pulse that are worthy of mention. For one, a 5 ns rise/fall time has been specified. Using a finite slew rate on a gate drive signal instead of abrupt transitions promotes improved convergence in the simulation. Secondly, a 100 ns offset has been specified before the first pulse. This delay aids convergence by allowing the simulator to reach a stable dc steady state solution prior to signals changing. Both of these "soft start" practices are recommended for circuit simulation in general, but particularly for eGaN FETs with their relatively high transconductance.

The ".meas" command is used in this example to find the time at which the stage 11 and stage 1 node voltages cross 2.5V on the falling edge. These times are then used to calculate the delay time for the first 10 stages. Figure 10(a) shows the voltage vs time for different values of the pull-up resistor. The dashed blue line is for the first stage, while the other curves are for the 11th stage. Large resistor values cause slow voltage rise times, with non-abrupt transitions, as can be seen in the brown curve (1 ohm). A small pull-up resistor leads to a resistor divider and non-zero node voltage when the device is on, as can be seen in the pink trace (0.1 ohm). Figure 10(b) is a graph of delay time per stage vs pullup resistor value, showing the high switching speeds that can be attained with an EPC2001.

This example demonstrates simulation with a large number of eGaN® FETs. Owing to the unique properties of these devices, the part models can not take advantage of parameterized transistor models built into SPICE (e.g. MOSFET,MESFET,JFET). Instead, eGaN models contain custom mathematic formulae which are fit to detailed measurements. As a result, the part models can be expected to simulate slower than a conventional transistor. However, even with a large number of parts, simulation results can be attained in a reasonable time.



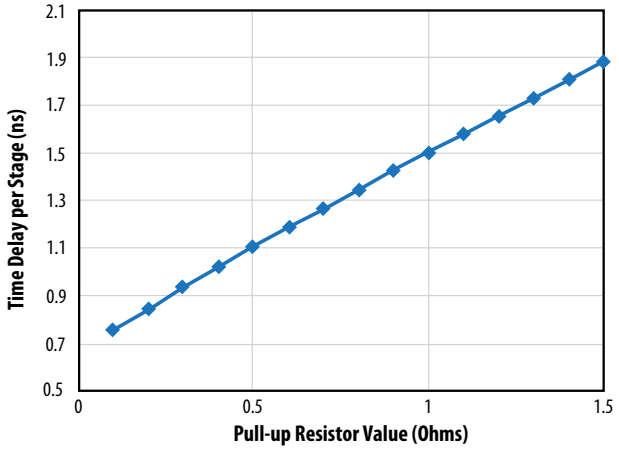
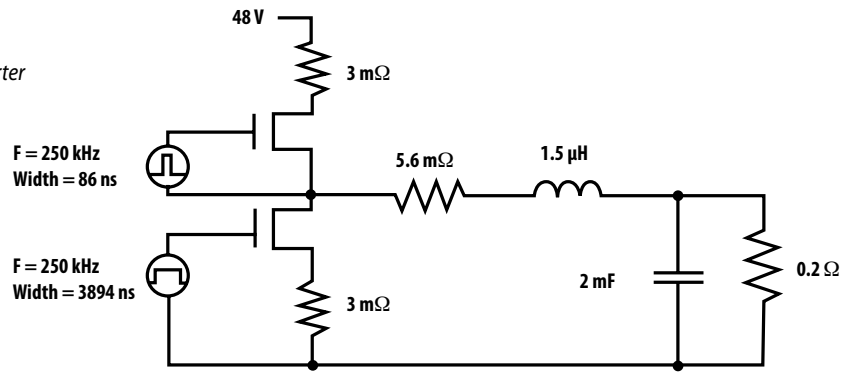


Figure 10: (A) Simulated delay line voltages at the 1st and 11th stages. Different colors correspond to varying values of the pull-up resistance used in the delay line. (B): Time delay per stage versus the pull-up resistance value.

Figure 11:
Schematic of a buck-boost converter

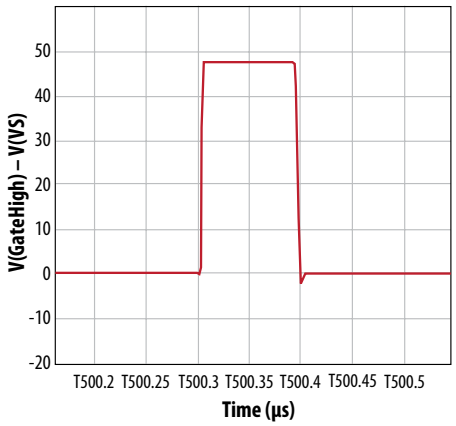


Example simulation of a 48-1.0V 5A converter at 250kHz

Figure 11 shows the schematic of this buck-boost 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:

```
*first line
.include EPC860subckt_V091.sp

********* Simulation Settings - Parameters and SPICE Options *********

XEPC860_1 VS GateLow sLow EPC860          *Sync Fet large die
XEPC199_1 IN GateHigh VS EPC199           *Control Fet small die

rstab1 IN VS 1MEG                         *Stabilization resistor to assist in convergence

VIN IN1 Gnd DC 48                         *Input Voltage

rpachigh IN1 IN .003                      *PCB resistance on high side

VDriver_1 GateLow2 Gnd PULSE(0 5 0 5n 5n 3892n 4u)   *Gate Driver low side
VDriver_2 GateHigh2 VS PULSE(0 5 3899n 5n 5n 88n 4u) *Gate Driver high side

rpaclow sLow GND .003                     *PCB resistance low side
LgLow GateLow GateLow2 .01n               *Gate inductance low side
LgHigh GateHigh GateHigh2 .01n            *Gate inductance high side
RL1 VS VS2 .0056                          *Inductor resistance
L1 VS2 VO 1.5u                            *Output inductor
C1 VO Gnd 2000u                           *Output Cap
Rload VO Gnd .2                           *Load resistor

.PRINT TRAN 'V(VS)-V(Gnd)'
.PRINT TRAN V(VO)
.PRINT TRAN '-((i(VIN)))'
.PRINT TRAN 'V(GateHigh)-V(VS)'
.PRINT TRAN V(GateLow)V(GateHigh,VS)
.measure tran dcpower avg '-(i(VIN)*v(IN)+i(VDriver_1)*v(GateLow2)+i(VDriver_2)*(v(GateHigh2)-v(
VS)))' from=500u to=750u
.measure tran Loadpower avg 'i(Rload)*v(Rload)' from=500u to=750u

********* Simulation Settings - Analysis section *********
.op

.tran 1n 760u

********* Simulation Settings - Additional SPICE commands *********

.options method = gear
*.options rmax = .1
*.options abstol=1e-10
.options reltol=1e-5

.end
```

In the SPICE code here, 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.



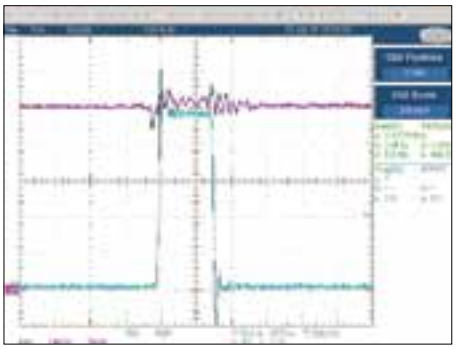


Figure 12: Switch node voltage vs time obtained from a) simulated 48V-1 V buck converter and b) measured data for switch node voltage and output voltage during operation of 48V-1V buck converter.

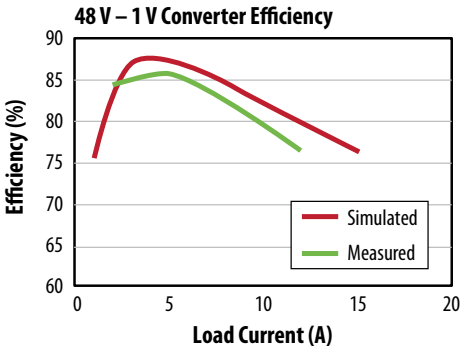


Figure 13: Comparison of simulated and measured efficiency for 48 V to 1 V converter.

Based on this simulation the output dc power is 5.35W, the total input dc power is 6.12W, and the conversion efficiency is 87%. Measured efficiency for this circuit was 85.9% in reasonable agreement 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 eGaN FETs. Greater than 85% efficiency in 48-1 V converters is well beyond state of the art Silicon capability.

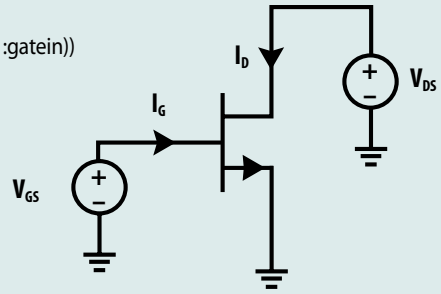
Figure 15 shows the simulated fT for various bias states of an EPC2015 device. As can be seen, fT reaches a maximum value exceeding 10 GHz with gate-source voltage near 2.5V. The models developed and presented here have not been tested for accuracy at this high frequency range. Furthermore, packaging and PCB layout have a strong influence on circuit performance in the GHz range. RF device models are currently under development, and will be made available soon.

Example cut-off frequency simulation of the EPC eGaN device

Figure 14 shows the schematic for a cut-off frequency simulation. The cut-off frequency (fT) of a transistor is the frequency at which the current gain becomes one. At this frequency, the ac magnitudes of the gate and drain current are equal. The following LTSPICE net-list is used to simulate this circuit using the EPC2015 part:

```
XU1 N002 N001 0 EPC2015
VGS N002 0 2.2 AC 0.1 Rser=0.0
VDS N001 0 1 Rser=0
.ac dec 50 1E3 5E10
.measure ac fT when mag(Ix(U1:drainin))=mag(Ix(U1:gatein))
.step VGS 2 2.6 0.1
.step VDS 0.0 10 0.1

.lib EPCGanlibrary.lib

.end
```

Figure 14: A schematic for cut-off frequency measurement



The operation of this netlist can be somewhat confusing on first inspection. The two ".step" commands cause SPICE to iterate thru a matrix of gate-source ("VGS") and drain-source ("VDS") values. For each bias state, an ac analysis is performed over the range of 1 kHz–50 GHz (with logarithmic steps in frequency). The ".measure" command operates on the output of a single frequency sweep. It determines the frequency at which the gate and drain current cross (i.e. have equal ac magnitudes). At the end of the simulation, the fT values can be extracted from the LTSPICE output log, where they are listed in tabular form next to the dc bias values for each measurement.

RF Impedance Matching Simulation

A common problem in RF applications is matching the input impedance of a transistor with the characteristic impedance of the gate drive electronics (typically 50Ω). Figure 16 shows a LTSPICE schematic of a simple LC impedance matching network at the gate of an eGaN FET. This network acts as a step down transformer, making the voltage on the device side lower and the current higher. For this example, the component values were chosen to achieve 50Ω at 500 MHz. The FET is treated as a 1-port network, with source and drain tied together, and gate biased with a dc 2.2V.

The phase and magnitude of the input impedance are plotted as a function of frequency at the bot-

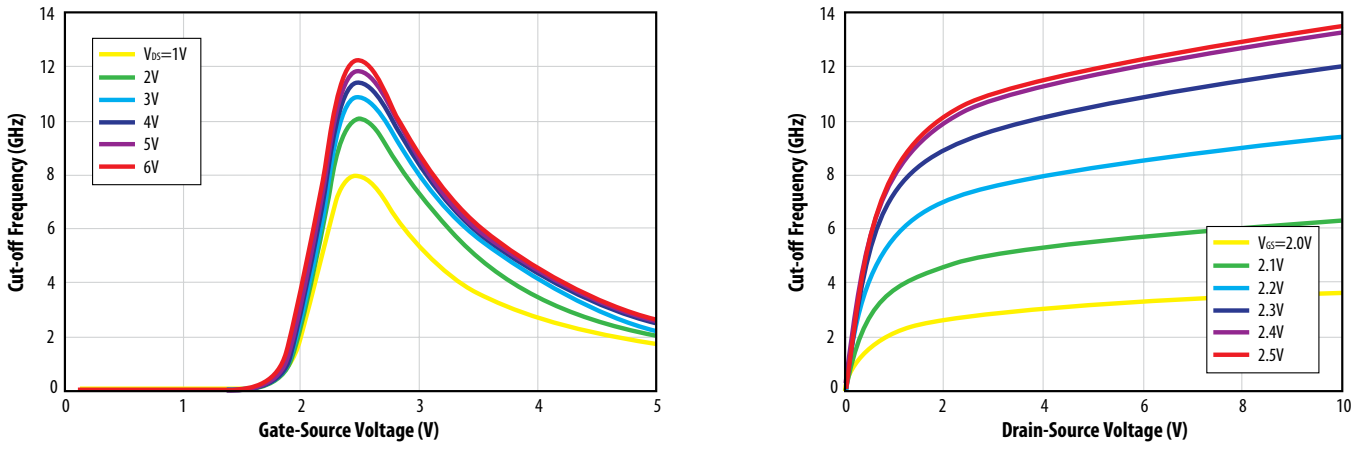


Figure 15: (Left) Simulated cut-off frequency versus VGS for VDS=1,2,3,4,5,6V; (Right) Cut-off frequency versus VDS for VGS=2.0-2.5V (EPC2015)

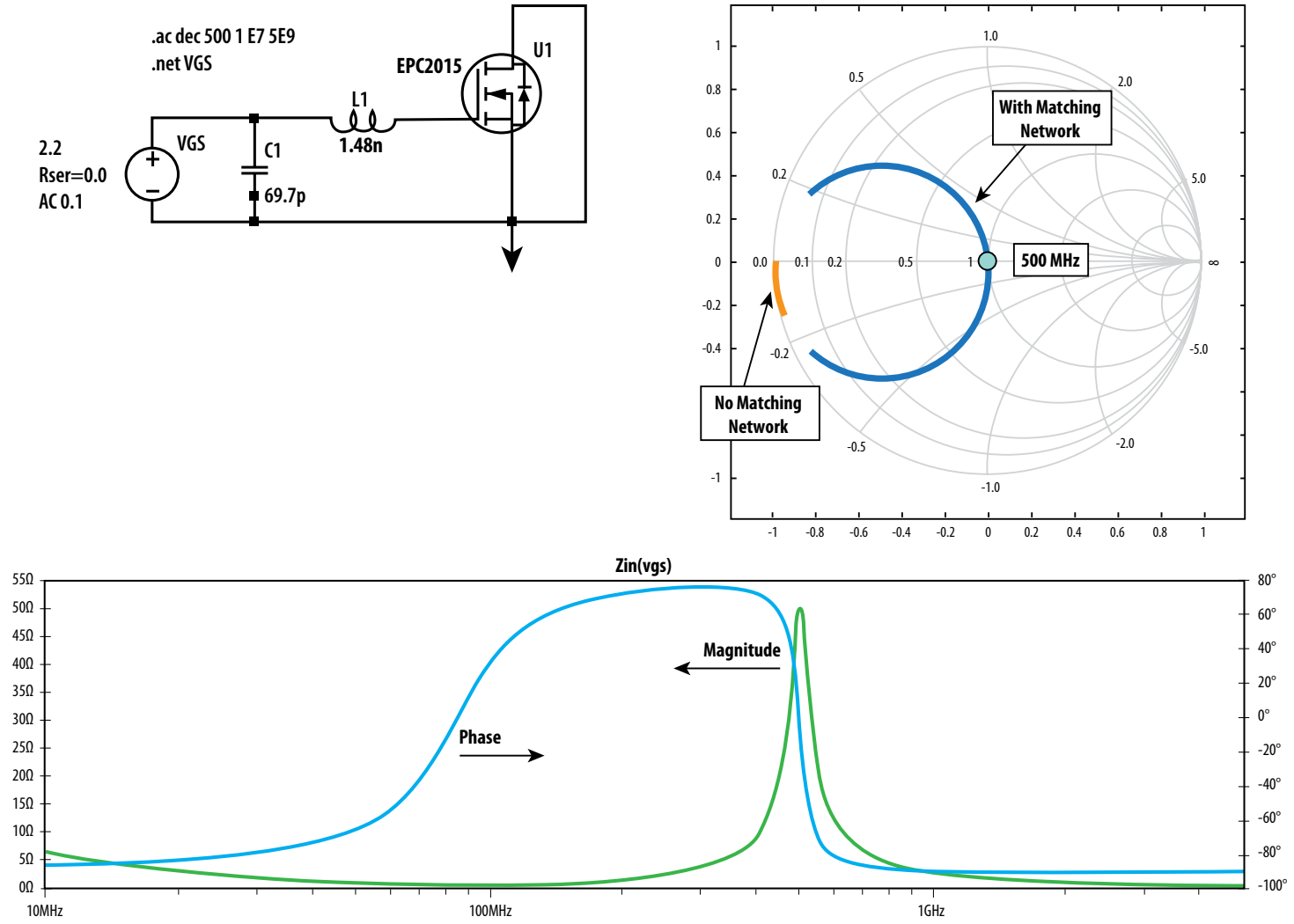


Figure 16: (Top Left) Circuit schematic for gate impedance matching simulation.
(Bottom) Input impedance (magnitude and phase) versus frequency looking into gate with matching network tuned for 500 MHz.
(Top Right) Smith chart of gate input impedance, both with and without matching network.

tom of the figure. As can be seen, the LC section provides good matching at 500 MHz (50 Ω real), but the performance over frequency is somewhat narrow-band. The impedance data was generated by invoking the ".net" directive in the SPICE netlist. For a 1-port network, this directs SPICE to compute the input impedance (or admittance) referred to an independent voltage source ("VGS") connected to the port. It can also be used to compute S-parameters of a 2-port network.

The same impedance data is also shown in the Smith chart at the upper right of Figure 16. The blue trace corresponds to the FET with gate matching network over the frequency range of 400-600 MHz. The input impedance crosses 50Ω (center of the Smith chart) at 500 MHz. The orange trace corresponds to the FET without any matching network. As can be seen, the impedance is very low (left side of the chart) over this frequency range owing to the gate capacitance (~1 nF) of the FET. The Smith chart was generated using data exported from LTSPICE. The impedance, admittance, and reflection coefficient (S11) can all be exported to a tabular text file which is readily imported into other analysis software.

Conclusions

Device simulation for enhancement mode eGaN® devices from EPC was presented with focusing on 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: LTSPICE netlist for EPC2001 device

This appendix contains a netlist for a LTSPICE circuit model of an EPC2001 FET. The model is implemented in a language very similar to the original Berkley SPICE, although some of the syntax is unique to LTSPICE. All device models offered by EPC (LTSPICE, PSPICE, TSPICE, Spectre) share the same mathematical models, and syntax very similar to the listing below. In most cases, these models can be ported to other SPICE simulation packages with relative ease.

```
********************************************************************************
* source EPC2001DEV1
.subckt EPC2001 gatein drainin sourcein
.param aWg=1077 A1=41.7998 k2=2.259866e+000 k3=1.2e-001 rpara=4.463059e-003
+      alTc=5.486028e-003 arTc=-4.699671e-003 ax0Tc=0.75E-4 x0_0=-0.75 x0_1=1.10
+      dgs1=4.3e-7 dgs2=2.6e-13 dgs3=.8 dgs4=.23
+      ags1=8.6952e-010 ags2=5.3168e-010 ags3=1.9975e+000 ags4=2.8377e-001
+      ags5=-1.4751e-010 ags6=-7.5163e+000 ags7=7.2121e+000
+      agd1=1.4182e-011 agd2=2.1475e-010 agd3=-3.8030e+000 agd4=5.9551e+000
+      asd1=3.3621e-010 asd2=6.3080e-010 asd3=-1.2803e+001 asd4=2.2690e+000
+      asd5=2.5818e-010 asd6=-4.0599e+001 asd7=2.0638e+001

*Parasitic drain, source, and gate resistances
rd drainin drain {(0.75*rpara)}
rs sourcein source {(0.25*rpara)}
rg gatein gate {(.6*1077/aWg)}

*Resistors for convergence (establish dc operating point)
Rcsdconv drain source {100000Meg/aWg}
Rcgsconv gate source {100000Meg/aWg}
Rcgdconv gate drain {100000Meg/aWg}

*Voltage controlled current source- main output characteristics of FET
bswitch drain source I=if(v(drain,source)>0,
+      (A1*(1-alTc*(Temp-25))*log(1.0+exp((v(gate,source)-k2)/k3))*
+      v(drain,source)/(1 + max((x0_0+x0_1*v(gate,source))/(1+ax0Tc*(Temp-25)*(Temp-25)),0.5)*v(drain,source)) ),
+      (-A1*(1-alTc*(Temp-25))*log(1.0+exp((v(gate,drain)-k2)/k3))*
+      v(source,drain)/(1 + max((x0_0+x0_1*v(gate,drain))/(1+ax0Tc*(Temp-25)*(Temp-25)),0.5)*v(source,drain)) ) )

*Parasitic gate to source leakage (non-linear voltage controlled current source)
bgsdiode gate source I=if( v(gate,source)>10,
+      (0.5*aWg/1077*(dgs1*(exp((10.0)/dgs3)-1)+dgs2*(exp((10.0)/dgs4)-1))),
+      (0.5*aWg/1077*(dgs1*(exp((v(gate,source))/dgs3)-1)+dgs2*(exp((v(gate,source))/dgs4)-1)))   )

*Parasitic gate to drain leakage (non-linear voltage controlled current source)
bgddiode gate drain I=if( v(gate,drain)>10,
+      (0.5*aWg/1077*(dgs1*(exp((10.0)/dgs3)-1)+dgs2*(exp((10.0)/dgs4)-1))),
+      (0.5*aWg/1077*(dgs1*(exp((v(gate,drain))/dgs3)-1)+dgs2*(exp((v(gate,drain))/dgs4)-1))) )

*Gate to source capacitance: fixed capacitor in parallel with a non-linear behavioral charge source
C_GS        gate source  {ags1} TC=0,0
C_CGS1      gate source Q=(0.5*ags2*ags4*log(1+exp((v(gate,source)-ags3)/ags4))+
+      ags5*ags7*log(1+exp((v(source,drain)-ags6)/ags7)) )

*Gate to drain capacitance: fixed capacitor in parallel with a non-linear behavioral charge source
C_GD        gate drain  {agd1} TC=0,0
C_CGD1      gate drain Q=(0.5*ags2*ags4*log(1+exp((v(gate,drain)-ags3)/ags4))+
+      agd2*agd4*log(1+exp((v(gate,drain)-agd3)/agd4)) )

*Source to drain capacitance: fixed capacitor in parallel with a non-linear behavioral charge source
C_SD        source drain  {asd1} TC=0,0
C_CSD1      source drain Q=(asd2*asd4*log(1+exp((v(source,drain)-asd3)/asd4))+
+      asd5*asd7*log(1+exp((v(source,drain)-asd6)/asd7)) )

.ends
```

Appendix B: Using EPC Device Models- Some Rules of Thumb

Some users of the EPC SPICE models have reported issues with convergence and stability. Though some of these issues are platform dependent, they generally relate to the high transconductance gain (gm) of the eGaN devices. Just as high gain can cause oscillation in real circuits, it can cause instability in simulated circuits as well. Below is a list of practical rules of thumb to address convergence problems.

1. In a transient analysis, ensure that FETs are in the off state initially (i.e. VGS ≤ 0).
    a. This is especially important if any FETs have a large (transient) drain to source voltage at startup

2. Make sure that SPICE converges at the dc operating point (voltages and currents) corresponding to the very beginning of a time domain simulation.
    a. If it does not converge there, try changing voltages. For instance, setting all voltage and current sources to zero should lead to convergence. From there, sequentially turn the sources back on until the offender(s) is found.

3. Wherever possible, avoid abrupt (instantaneous) changes in voltage/current sources
    a. A common offender is a square wave gate drive signal
    b. Replace abrupt gate transitions with a linear ramp (finite slew rate)
    c. To begin with, set the slew rate low. If there are no convergence problems, iteratively increase the slew rate until the desired gate drive characteristics are attained.

4. As a last resort, try setting the time step to a very small number.
    a. Most SPICE platforms allow the user to specify a fixed (or maximum) time step.
    b. In general, circuit simulators use the current solution as the convergence starting point for the next time step. If external stimuli do not change very much because the time step is small, the simulator is better able to find the solution.
    c. Here again, employ an iterative approach, starting with a very small step (e.g. 10 ps) and gradually increasing it until convergence problems begin to appear.

5. Divide and conquer: Break up large, complex circuits into smaller sub-circuits. Test each sub-circuit separately in an attempt to isolate the source of the convergence problem.

6. Reality checks: Often times, a convergence problem is caused by an oversight in the circuit design.
    a. An example is a circuit which drives an eGaN FET at voltage or current levels which it could not realistically survive
    b. Probe voltage and current levels at the terminals of the FET
    c. Common circuit errors include:
        i. Excessively high forward bias on the gate. The gate should never be > 6V more positive than either source or drain.
        ii. Excessive source-drain voltage. Make sure VDS does not exceed spec. sheet limit.
        iii. Excessive drain current. Continuous current in the hundreds of amps indicates a circuit design error.

Appendix C: Detailed Instructions for Downloading and Running the LTSPICE Models

Some users of the EPC SPICE models have reported issues with convergence and stability. Though some of these issues are platform dependent, they generally relate to the high transconductance gain (gm) of the eGaN devices. Just as high gain can cause oscillation in real circuits, it can cause instability in simulated circuits as well. Below is a list of practical rules of thumb to address convergence problems.

1. Download the LTSPICE EPC model library from the web:
   a. http://epc-co.com/epc/ToolsandDesignSupport/DeviceModels.aspx

2. Unzip the compressed folder. There are three files in the folder:
   a. EPCGanLibrary.lib: Library containing the device models for all of the EPC parts
   b. EPCGaN.asy: A generic part symbol for EPC FETs
   c. Example EPC1001.asc: A LTSPICE schematic capture file containing an example circuit with eGaN devices

3. Launch LTSPICE and open a new schematic (File->New Schematic)

4. Save the (empty) schematic

5. Move the library (EPCGanLibrary.lib) and symbol (EPCGaN.asy) to the same directory which contains your schematic (.asc)

6. In LTSPICE, select: Edit->Component
   a. In the dialog box which appears, change the Top Directory to the directory containing your schematic
   b. In the list below, select EPCGaN. This will select the generic eGaN FET symbol.
   c. Place the symbol in your schematic

7. This symbol applies to all eGaN parts in the model library
   a. By default, it is set to the EPC1001 part
   b. To change the part model associated with this symbol, right click on the symbol
      i. Under "Value", change EPC1001 to any other part in the library (e.g. EPC2015)
      ii. Click O.K.

8. At this point, you are ready to complete your circuit and start simulating.

Note: As an alternative to moving the files to your working directory, the files can be copied into folders that are part of the permanent search path of LTSPICE. In this way, the symbol and models become globally available. The symbol (EPCGaN.asy) should be copied to the lib\sym subfolder of the main LTSPICE directory. The model library (EPCGanLibrary.lib) should be copied into the lib\sub folder.

Appendix D: 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.

[1] *MOSFET Modeling for Circuit Analysis and Design, Galup-Montoro and Schneider, World Scientific Publishing 2007 ISBN 9812568107*
[2] *Analysis Through Numerical Solution of Poisson's Equation, Schroeder, J. E., Muller, R.S. iEEE Trans. Elect. Devices ED-15 No12, pg954 (1968)*
[3] *Mosfet Modeling for VLSI Simulation: Theory and Practice, Narain Arora, World Scientific (2007) ISBN 981256862*
[4] *Spice model of AlGaN/GaN MODFETs and simulation of VCO and power amplifier, Syed S. Islam and A.F.M. Anwar, International J. of High Speed Electronics and Systems, V14 (3), pp853 (2004)*
[5] *An accurate charge control model for spontaneous and piezoelectric polarization dependent two-dimensional electron gas sheet charge density of lattice mismatched AlGaN/GaN MODFETs, Rashmi, Abhinav Kranti, S. Haldar, and R.S. Gupta, Solid-State Electronics, V46, p621, (2002)*