Importing USCi Models into LTSPICE

John Bendel

Abstract

United Silicon Carbide provides standard text based SPICE models to all their commercially released products. To fully utilize these models they need to be imported into a circuit simulator. This application note details the process to add USCi models to LTSPICE, and apply them to a simple example.

1.0 Adding Devices to LTSPICE

In LTSPICE, each device requires a SPICE text model (supplier generated) and a corresponding symbol to represent that model in a schematic. In this note a UJC1210K and UJ2D1205T will be imported into LTSPICE and used in a "double pulse" test circuit. Once the devices are added, the "save as" command can be liberally applied to add additional devices in the future.

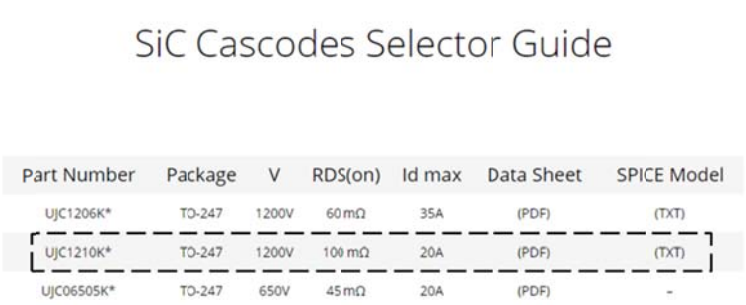

Figure 1: JFET and Cascode Selector Page

1.1 Importing the SPICE text model

Every component listed on the USCi website has a corresponding SPICE file, which is located on the product selector page. Figure 1 highlights the UJC1210K. To create a file from the website, click on the SPICE model in the selector guide and copy and paste into Notepad. Save this file using the naming convention subcktname.sub (.subckt UJC1210K nd ng ns).

The file should be stored in a location that can be built up when adding future devices. A good location is either in a folder off the document directory or within the LTSPICE Libraries (may require a right click to run as administrator). It is not a good idea to add the files directly in with the models that come standard with LTSPICE (sub) for fear of them being overwritten during an upgrade. Figure 2 illustrates the cascode and two diode models saved in a folder called USCi under LTC's "sub" folder.

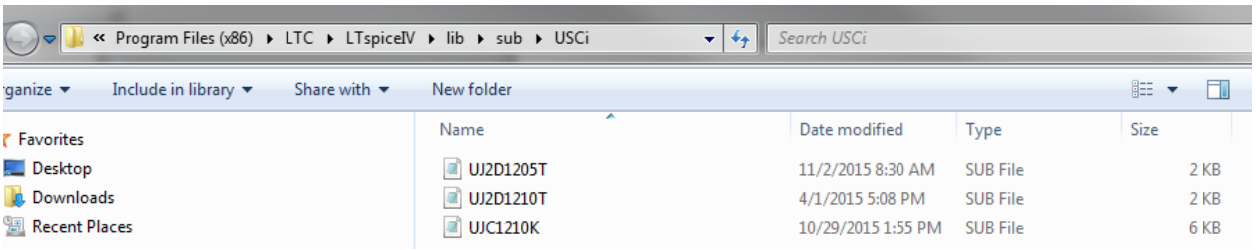

Figure 2: USCi Spice Models in LTC… Sub/USCi Directory

1.2 Using an existing Symbol

Now that the SPICE files have been added to the USCi subckt folder, one needs to associate a symbol to the file. The name of this symbol file will be the part number for the new device. For standard devices like a diode, an existing symbol can be renamed and associated to the SPICE file.

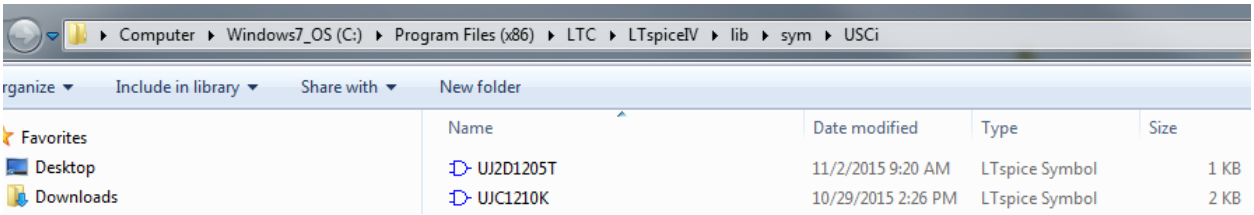

Figure 3: USCi Symbol Files in LTC… Sym/USCi Directory

To associate the UJ2D1205T SPICE file to a diode symbol, open the generic diode symbol in the LTC "Sym" folder, and save it in a folder just under the LTC "sym" folder using the devicename.asy (UJ2D1205T.ASY) naming convention (Figure 3).

To complete the association, open "Edit Attributes" from under the Edit menu (Figure 4). As the UJ2D1205T model is a subckt, change "D" in the prefix field to "X", and in the value field, change "D" to the subckt name: UJ2D1205T. The Symbol and models are now associated by name.

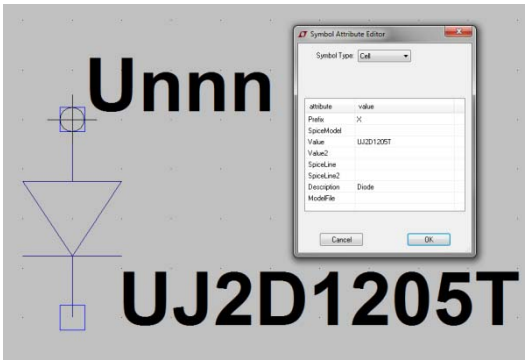

Figure 4: Edit Attributes

To use the device, just as in spice netlists, a .LIB statement needs to be added to a schematic to direct LTSPICE to the location of the spice file. Skipping ahead, Figure 6 shows several .LIB statements in a schematic.

The .LIB statements may seem tedious, but they are convenient when one wishes to simulate customized models of the same device without having to create a new device for each variation. Copy the spice file to a different location, modify it, and by swapping out different .LIB path statements, one can run simulations with different device parameters without creating new symbols, or affecting other schematics that rely on the original model.

1.3 Creating a Cascode Symbol

In the case of the cascode, an existing symbol does not exist. The cascode symbol can be created from scratch, but it is simpler to modify an existing symbol. For the cascode, the NMOS symbol is a good choice. Open the NMOS symbol from the LTC library, and save it as UJC1210K.ASY in the same USCi folder as the UJ2D1205T (Figure 3).

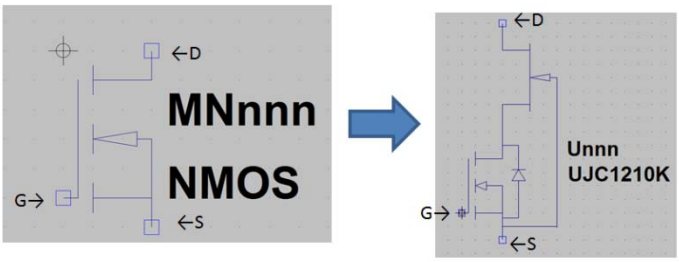

Figure 5: Converting NMOS Symbols to Cascode

With the newly renamed UJC1210K.ASY, a few cut, copy, and pastes can create a cascode symbol (Figure 5). The most important element in the symbol is that it has the same number of pins as the subckt, and they are in the same numeric sequence as listed in the SPICE model. The sequencing of the NMOS aligns with the UJC1210K.SUB subckt statement, so no change is required.

If there is any question of a pin's name or number, a quick right click on a pin can confirm its name and sequence. To complete the symbol process, as with the UJ2D1205T, open up Edit Attributes and type "X" into the Prefix and the model name "UJC1210K" in the Value field.

This Symbol is now complete, and the device is ready to be used in a circuit. The .LIB statement requirements still apply. By the time this paper is posted, a cascode symbol may be available for download on the USCi website.

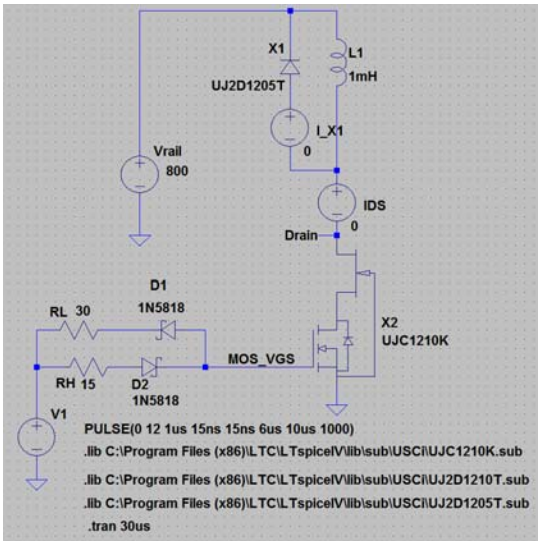

Figure 6: Double Pulse Circuit

1.4 Double Pulse Tester

It is now time to use the devices. For those unfamiliar with a double pulse tester (Figure 6), it is a methodology used to measure the reverse recovery of a diode, or switching losses in a power device.

In operation, the Power Switch (UJC1210K, X2) is turned on, and the current ramps through the inductor, L1. The amount of time X2 is "on" sets the current for the measurements. When the test current is reached, X2 is cycled off/on.

Some results are shown In Figure 7. Eon, Eoff and Q_{RR} are quantified, and a selection of waveforms are shown. Although it is always best to consider simulations as "relative" vs. "absolute", a bevy of useful information can still be pulled from the data. By adding layout considerations into the schematic/simulation, the designer can continue to hone a collection of circuits that mimic the real work, and shorten design time. A well thought out simulation may highlight a nuance that needs to be accounted for in a system design, and thus be caught early and save days if not weeks of development time. In the power/analog world, simulations are not a substitute for the real lab experience, but they can make that lab experience much more pleasant.

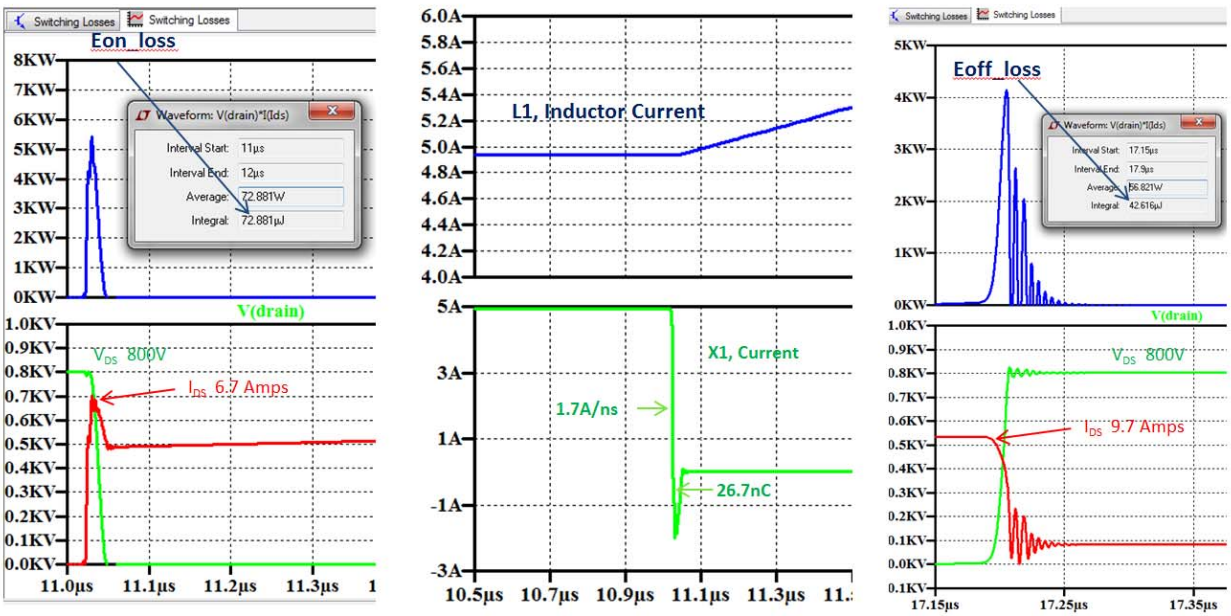

Figure 7: Switching Dynamics