

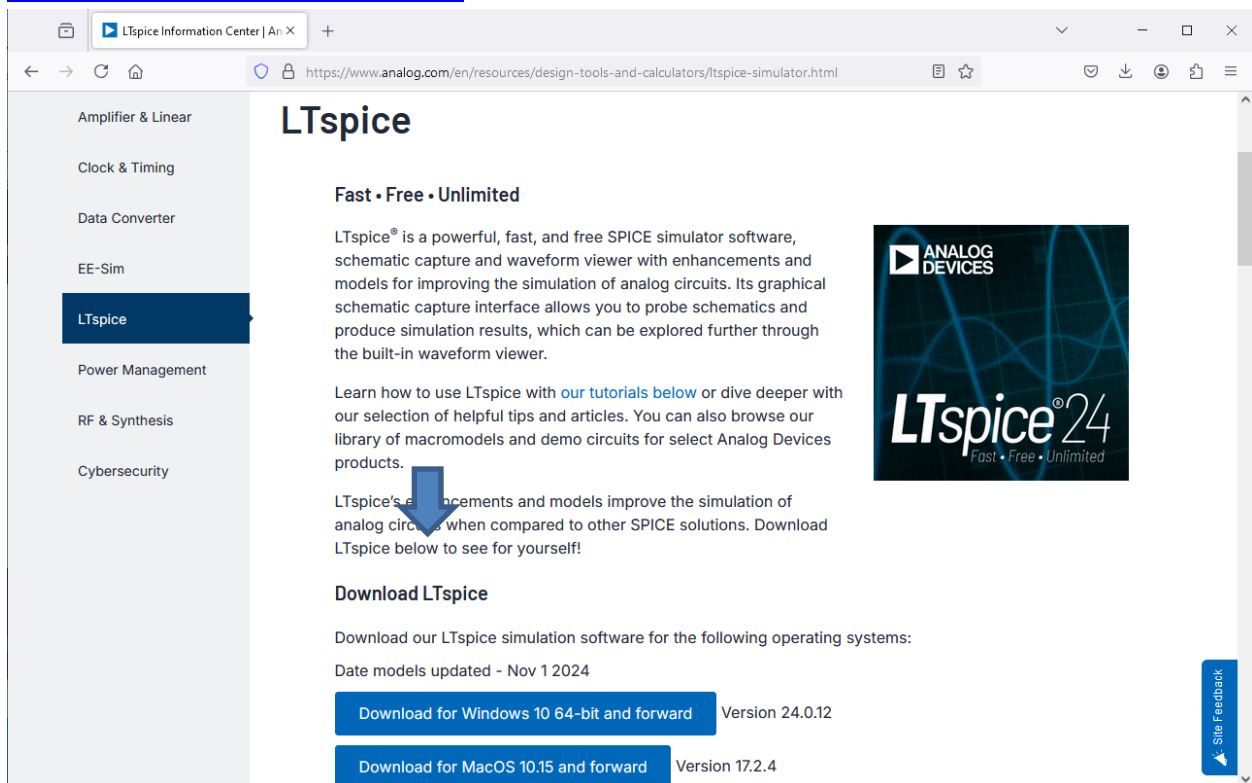
Instructions to Install LTspice

Introduction

We will use LTspice in the software lab for the Accelerator Power Electronics course. We have slightly modified the standard installation in order to include some parts that are not in the standard distribution. These instructions detail how to both install the standard distribution and how to modify it to include our additions.

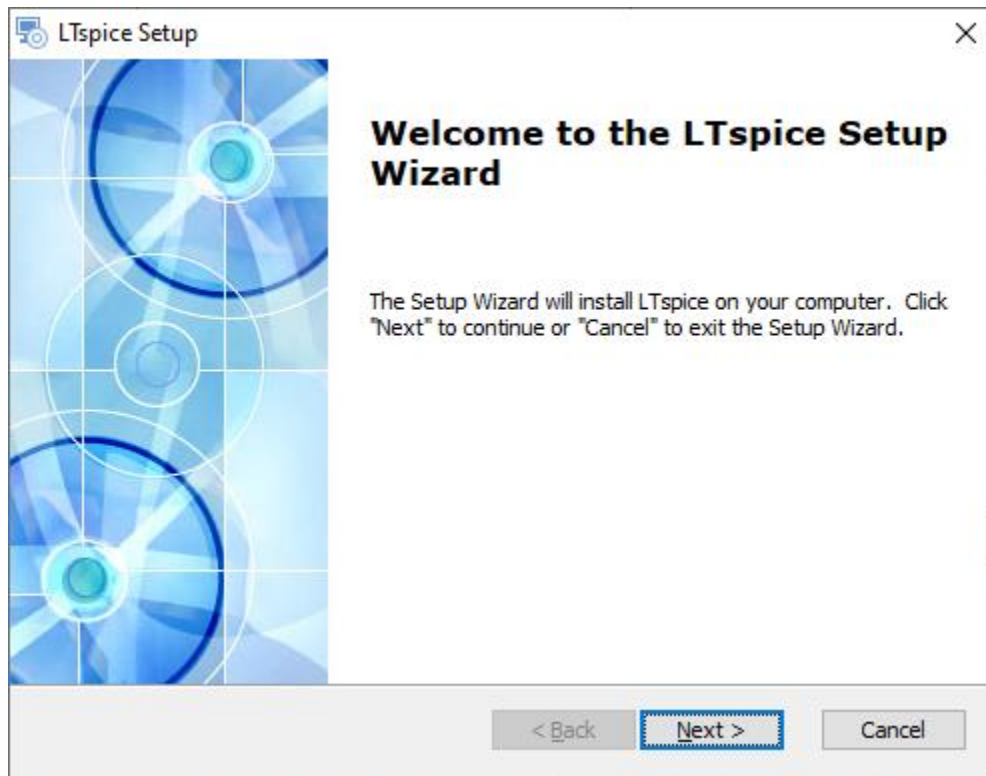
Standard Installation

The software was developed by Linear Technology, Inc. It since has been bought by Analog Devices, Inc., who now maintains the software. The software, the current version is 24.0.12, can be freely downloaded from Analog Devices at its website <https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html>

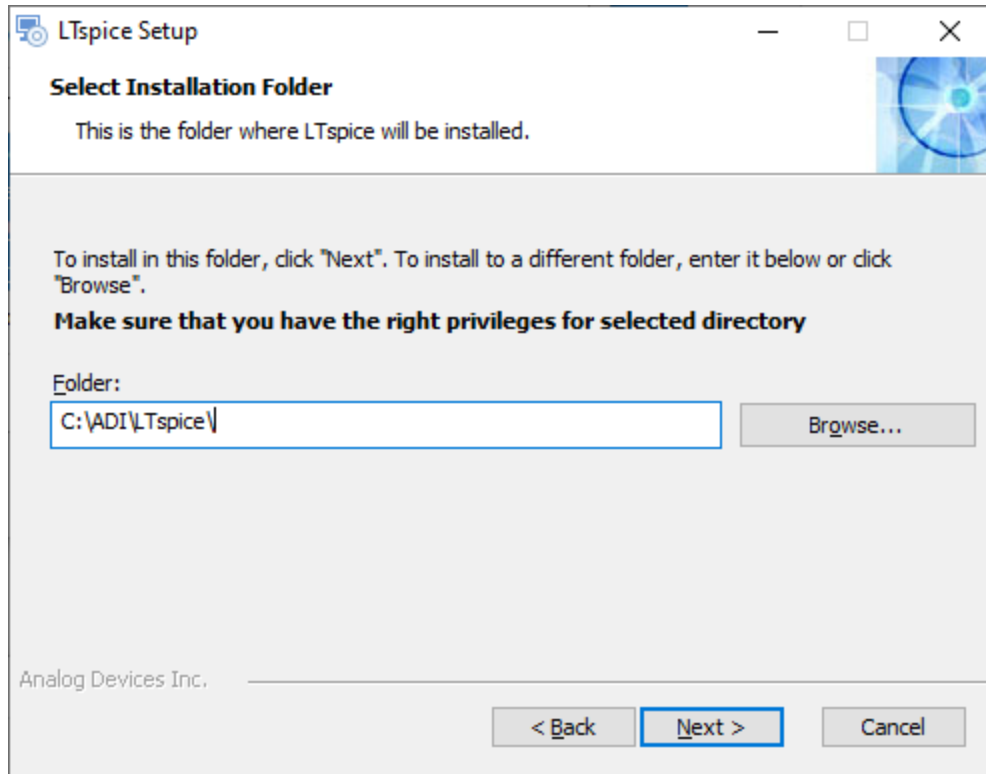


The screenshot shows a web browser window displaying the LTspice website. The page title is "LTspice" and the URL is "https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html". The page features a navigation menu on the left with categories like "Amplifier & Linear", "Clock & Timing", "Data Converter", "EE-Sim", "LTspice", "Power Management", "RF & Synthesis", and "Cybersecurity". The main content area is titled "LTspice" and includes the tagline "Fast • Free • Unlimited". Below this, there is a description of LTspice as a powerful, fast, and free SPICE simulator software. A blue arrow points to the text "LTspice's enhancements and models improve the simulation of analog circuits when compared to other SPICE solutions. Download LTspice below to see for yourself!". The "Download LTspice" section lists two download options: "Download for Windows 10 64-bit and forward" (Version 24.0.12) and "Download for MacOS 10.15 and forward" (Version 17.2.4). A "Site Feedback" button is visible in the bottom right corner.

Select *Download for Windows 10 64-bit and forward* (if on a Windows machine. *LTspice* is also available for the Mac.). This downloads the file *LTspice64.msi*. After the file is downloaded, select it to run the *LTspice Setup Wizard*. (Windows may ask if you are sure that you want to launch the file.) (You may need administrative privileges to install the program.)



Accept the terms in the license agreement, install LTspice for Only for me, and then when asked to Select Installation Folder



change the installation directory from the default C:\Users\sebek\AppData\Local\Programs\ADI\LTspice to C:\ADI\LTspice, then Next, Install to begin the installation, and then finish and launch LTspice. You may be prompted to return to the ADI website to update the installation with the latest versions of ADI generated files. You also can choose to use their new GUI shortcuts.

There should now be a directory C:\ADI\LTspice that includes several files, including LTspice.exe. The shortcut created by LTspice runs this executable for a 64 bit installation. The directory also contains two zip files, examples.zip and lib.zip. The .zip files, examples.zip and lib.zip, are installed in C:\ADI\LTspice. These files are copied and then exploded in C:\Users\username\AppData\Local\LTspice. This is where LTspice looks for the files it will use in building the schematics and simulations. On Windows 10, LTspice also creates a directory in the user space C:\Users\username\Documents\LTspice and it creates an initialization file, LTspice.ini, in C:\Users\username\AppData\Roaming that contains initialization information.

Customization

Configure LTspice

We would like to have a directory in which we store our work files. We create a directory structure C:\files\ItSpiceFiles

This structure will also store our custom devices, those not in the LTspice distribution. Many semiconductor manufacturers create SPICE files to enable designers to correctly model their devices. If

we want to use these devices in LTspice, we must download these SPICE files and make them available to LTspice.

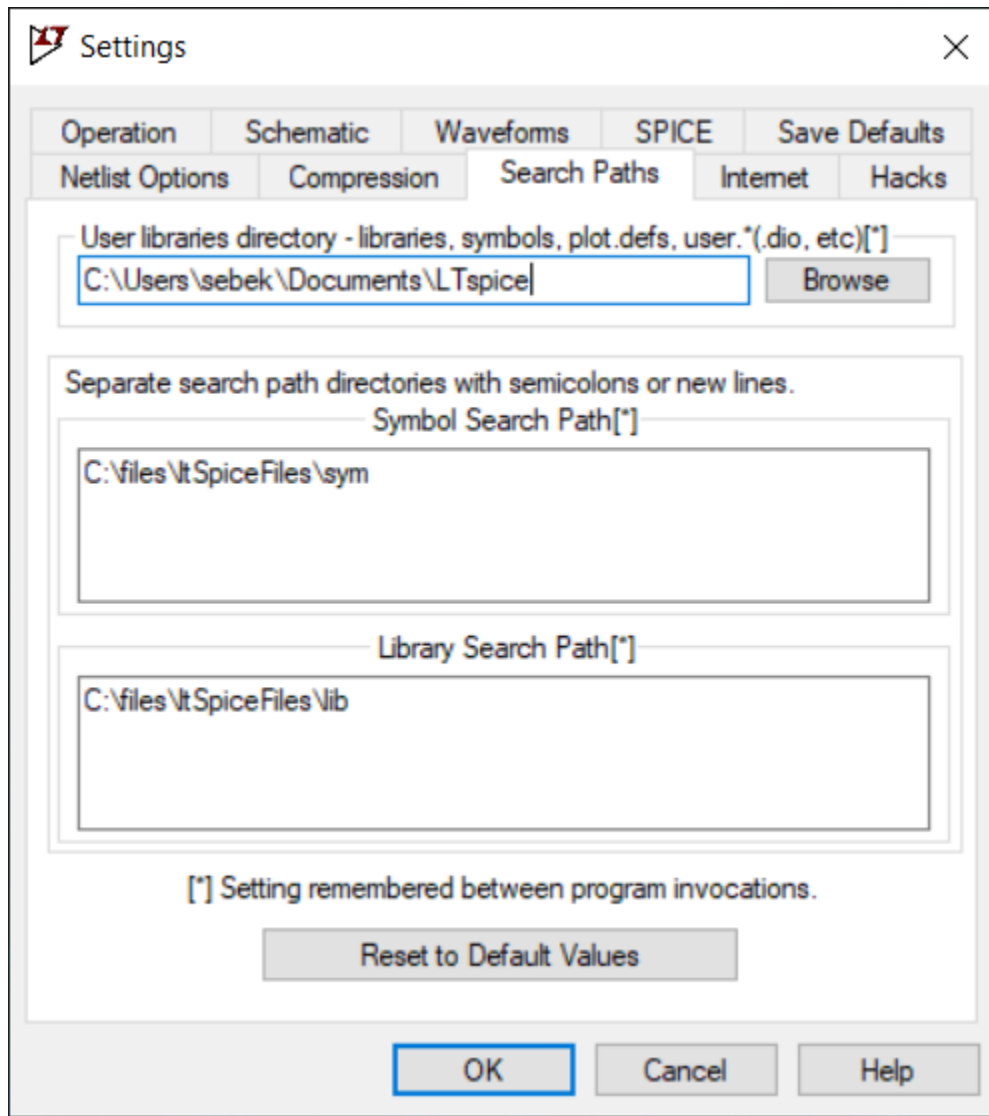
LTspice requires library files containing SPICE directives that describe the behavior of each device. LTspice is also GUI driven, so it requires symbol icons to correctly display the device on the GUI. We create two subdirectories, one for library parts that we add to the distribution and the other for their symbols. We create the subdirectories

C:\files\ltSpiceFiles\sym

C:\files\ltSpiceFiles\lib

and include them in the LTspice search path. We enter these directories after navigating to LTspice->Tools->Settings->Search Paths

entering the directory names and then OK. These paths are stored as the variables SymbolSearchPath and LibrarySearchPath, respectively, in C:\Users\username\AppData\Roaming\LTspice.ini.



We also create the directory C:\file\ItSpiceFiles\homework as a directory in which we will save our homework.

Adding Custom Devices

We will be creating IGBT files, so copy the standard symbol for N and P channel IGBTs into the C:\files\ItSpiceFiles\sym directory. From C:\Users\username\AppData\Local\LTspice\lib\sym\Misc copy NIGBT.asy and PIGBT.asy to C:\files\ItSpiceFiles\sym

General Notes

Adding Models

The basic element of a SPICE circuit is a model. SPICE uses equations that define the current and voltage characteristics for each model. An example of a model is a bipolar transistor. To individualize each type of transistor, we then create a transistor model, initializing the value of each model parameter to that transistor characteristic.

A sample model for a 2N2222 transistor is

```
.model 2N2222 NPN(IS=1E-14 VAF=100
+ BF=200 IKF=0.3 XTB=1.5 BR=3
+ CJC=8E-12 CJE=25E-12 TR=100E-9 TF=400E-12
+ ITF=1 VTF=2 XTF=3 RB=10 RC=.3 RE=.2 Vceo=30 Icrating=800m mfg=Philips)
```

In this example, *.model* is the keyword telling SPICE a model definition follows, *2N2222* is the identifier of this model, *NPN* says that this element is an NPN transistor, and most of the assignment equations in the parenthesis are valid SPICE parameters for an NPN transistor model. The last three parameters, *Vceo*, *Icrating*, and *mfg* appear to be LTspice specific informational parameters. LTspice extracts them and displays them in the menu that allows us to select the particular transistor for the circuit. The + signs at the beginning of the lines tell SPICE that these lines are continuations of a previous command.

The standard LTspice installation includes many files of standard active analog components, such as bipolar transistors, diodes, junction FETs and MOSFETs. These files are all, by default, located in the library C:\Users\username\AppData\Local\LTspice\lib\cmp (for component). All the files for the active components are ASCII files with a series of models similar to the one displayed above for the 2N2222 bipolar transistor. (The file standard.bjt contains the model definition of the 2N2222 transistor discussed above.) To add additional devices to these files, we use an ASCII text editor and copy a SPICE model of the new device, typically obtained from a manufacturer's website, to an existing, but renamed file. If we also want to add information to the model so that LTspice can display the information during the component selection, we add the appropriate equalities to the entry, using the existing entries as a model to determine the names of the valid informational parameters. We do not need to add models in these LTspice files. We can also create ASCII files with this information and place these files in the directory we added to the library search path above.

In addition to the model files containing semiconductors, there are also model files for passive devices, ferrite beads, capacitors, diodes, and resistors. These files are stored in a binary format, but we can edit them from within LTspice.

Symbol (.asy) File

Each time we add a device we want to associate it with a symbol file so that our new device can be displayed correctly in the LTspice GUI.

We will examine NIGBT.asy to see the elements of a typical LTspice symbol file. This file is located in C:\Users\username\AppData\Local\LTspice\lib\sym\Misc. A symbol file has information for the GUI. It contains drawing commands, often starting with LINE, for the ICON, and other location commands, such as WINDOW. It also has locations for pins. Therefore the number of pins in the symbol must match the number of terminals on a device, two for a resistor, capacitor, inductor, diode, etc., three for a bipolar transistor, MOSFET, IGBT, thyristor, etc. In addition to its location, each pin requires several attributes, labeled PINATTR, to be defined in the .asy file. One is the PinName, the identifier that will be placed on the schematic. For an IGBT, these identifiers are C, G, and E, for the collector, gate, and emitter of the IGBT. Another attribute is the SpiceOrder, which tells the GUI how to associate the pins with the terminals defined in the actual SPICE definition file, a complementary text file that resides in a lib directory and which defines the attributes of the SPICE model and tells LTspice the voltage and current relations at the various terminals. When we copy and customize the .asy file, we will modify its SYMATTR so that the part number of our device is displayed in the GUI.

The contents of NIGBT.asy are copied below.

Version 4

SymbolType CELL

LINE Normal 16 80 32 88

LINE Normal 48 96 28 92

LINE Normal 48 96 32 84

LINE Normal 28 92 32 84

LINE Normal 16 48 40 56

LINE Normal 16 8 16 24

LINE Normal 16 40 16 56

LINE Normal 16 72 16 88

LINE Normal 0 80 8 80

LINE Normal 8 16 8 80

LINE Normal 48 0 32 8

LINE Normal 16 16 36 12

LINE Normal 16 16 32 4

LINE Normal 36 12 32 4

WINDOW 0 56 32 Left 0

WINDOW 3 56 72 Left 0

SYMATTR Value NIGBT

SYMATTR Prefix Z
SYMATTR Description N-Channel IGBT symbol for a model you supply.
PIN 48 0 NONE 0
PINATTR PinName C
PINATTR SpiceOrder 1
PIN 0 80 NONE 0
PINATTR PinName G
PINATTR SpiceOrder 2
PIN 48 96 NONE 0
PINATTR PinName E
PINATTR SpiceOrder 3

Importing and Using IGBT Models

SPICE files can appear in different ways. The basic SPICE construct is an *element*. Examples of standard SPICE elements are resistors, capacitors, inductors, voltage sources, current sources, diodes, bipolar junction transistors, MOSFETs, MESFETs, etc. With each element is associated a model that gives its voltage-current behavior, typically non-linear relations, as a function of various parameters based on device physics. If LTspice does not include an element from a particular manufacturer, one can add an element in the appropriate LTspice file, such as standard.dio, standard.mos, etc., by adding a line with the appropriate parameters for that particular element.

For more complicated structures, non-elemental structures, such as an IGBT, there are several options for adding the device to LTspice. The options depend on the information supplied by the manufacturer. In some cases, the manufacturer builds up the IGBT by combining elements with well-defined SPICE models into a subcircuit. In other cases, the manufacturer uses a “non-standard” model, one that is not in the base version of SPICE. Some of these non-standard models are designed for variants of the base SPICE models. (Note that there is a standard set of components that are used in all versions of SPICE. Different distributions of SPICE, in particular PSpice, may have added enhancements that accept non-standard model definitions. These non-standard models likely will not work in LTspice.) We can only use the non-standard models in LTspice if it supports them.

LTspice does support some of these models, such as a common model for an IGBT developed at NIST. All models compatible with SPICE 2, SPICE 3, or LTspice should work. If LTspice supports the non-standard model, one adds it differently from the way in which one adds a subcircuit.

In other cases, the vendor does not use any basic SPICE elements as building blocks. Rather, they fit theoretical or measured performance data of the devices and then create a SPICE model using numerical equations within SPICE to simulate the performance of the device.

The Help files that come with LTSpice describe the process required to add these models. One can search the help for subcircuit and/or IGBT, as appropriate. Analog Devices also has a video that describes the addition of subcircuits and models (note that this video is from 2011 and an earlier

version of LTspice, but it still is likely correct)

<https://www.analog.com/en/resources/media-center/videos/5579239882001.html>

Vendor Model Resources

Several semiconductor vendors have LTspice compatible models on their website. The ease of obtaining models appropriate for LTspice varies from vendor to vendor. With some vendors one searches for SPICE files in a centrally located area. With other vendors one must first select the part, then see if a SPICE file exists, and finally access that file. The list of vendors below is not meant to be all inclusive. They are just some power semiconductor vendors listed as examples.

Infiniteon

Infiniteon has models at links on <https://www.infineon.com/cms/en/tools/>. From there one selects the Infiniteon Simulation Models (SPICE, PLECS) to get to the web page from which one can select SPICE model language simulation files for the Infiniteon components.

Wolfsped

Wolfsped has developed models specifically for LTspice. One can obtain the LTspice models from a link to All SPICE models on their website <https://www.wolfsped.com/tools-and-support/power/ltspice-and-plecs-models/>. This site also has a link to a User Guide PRD-07913 that describes the process of downloading and installing LTspice models.

OnSemi

I did not find a convenient way to search for and find all LTspice files from OnSemi. They provide an application note describing the methods to find, install, and execute a model on LTspice.

<https://www.onsemi.com/pub/collateral/and9783-d.pdf>

The method described in this note requires that one know the device of interest and start with the OnSemi product page for that device. From there, one selects the Technical Documentation tab which leads to a link to the LTspice model and symbol for, for example, the SuperFET3 650V. This link is to a .zip file that appears to contain all models for the many devices in the same OnSemi family. The LTspice models are encoded in a binary format in order to protect the OnSemi intellectual property.

Subcircuits

Infiniteon IKW75N65EH5 IGBT

As an example of the process of creating a custom model, we will import an IGBT model from Infiniteon.

We select an active and preferred Infiniteon IGBT with a built-in freewheeling SiC Schottky barrier diode, the IKW75N65EH5. Infiniteon supplies a file, IGBT_650V_TRENCHSTOP5_H5_L1.lib, that includes a subcircuit for this device, as well as for several other related devices. This file is an ascii text file. This file is readable; some SPICE files are encrypted to protect the vendor's intellectual property. Infiniteon chooses to use the .lib extension for this SPICE model, but other suffixes are possible. To create this model, Infiniteon has built an analytical model of the device that simulates measured device behavior. Infiniteon calls this a Level 1 model. We save this Level 1 file in our personal lib folder. Infiniteon also supplies Level 2 models that are derived from other device models and data in the file IGBT_650V_TRENCHSTOP5_H5_L2_LTSpice.lib.

We next assign an icon for this device to use in LTspice. (LTspice has a method to create a rectangular box icon, but we prefer to create an icon that looks like the standard IGBT symbol.) We start with the LTspice symbol NIGBT.asy, another text file. We copy that file and rename it NIGBTikw75n65eh5.asy. In this assembly file we need to define attributes that will associate the file with the vendor subcircuit model. (A discussion of attributes is in the LTspiceHelp,

LTspice Help->Schematic Capture->Creating New Symbols->Adding Attributes)

One can either edit the file NIGBTikw75n65eh5.asy with a text editor and then save the changes or, alternatively, from the LTspice GUI, File->Open->C:\files\ltSpiceFiles\sym\NIGBTikw75n65eh5.asy. This will bring up the renamed NIGBT.asy symbol created by LTspice. From the Edit menu, select Attributes (or enter Ctrl-A).

The symbol file needs to have several attributes defined:

- *SYMATTR Prefix X*
The X tells LTspice that this symbol is a sub-circuit
- *SYMATTR SpiceModel IKW75N65EH5_L1*
This is the name of the .SUBCKT in the file that will define the behavior of this device
- *SYMATTR Value IKW75N65EH5_L1*
This is the desired name on the schematic. (I do not understand the difference between these two attributes.)
- *SYMATTR Description *descript**
where *descript* is a user supplied phrase. I am not sure how this value is used. This may be a label included on the drawing.
- *SYMATTR ModelFile IGBT_650V_TRENCHSTOP5_H5_L1.lib*
IGBT_650V_TRENCHSTOP5_H5_L1.lib is the name of the ASCII file containing the definition of the sub-circuit. This holds true if the SPICE file is in the sub directory. I found it to also hold if the .lib file is contained in a directory defined in the Settings->Library Search Path. (I did not try the case in which this file was not in library search path. I think that if it is elsewhere, one needs to include the absolute path name, as described in the LTspice help on the .lib command.)

To summarize, we add the following SYMATTR lines to the symbol file:

SYMATTR Prefix X

SYMATTR Value IKW75N65EH5_L1

SYMATTR Description Infineon IKW75N65EH5 IGBT

SYMATTR SpiceModel IKW75N65EH5_L1

SYMATTR ModelFile IGBT_650V_TRENCHSTOP5_H5_L1.lib

Finally, the attributes of the pins need to be set so that they match up with the sub-circuit pins. In this model, the collector is identified with pin 1, the gate with pin 2, and the emitter with pin 3.

PINATTR PinName C

PINATTR SpiceOrder 1

PINATTR PinName G
PINATTR SpiceOrder 2
PINATTR PinName E
PINATTR SpiceOrder 3

Then we save the newly created .asy file into our personal sym folder.

In order to use this device in a schematic, add a component, navigate to our custom directory, C:\files\ltSpiceFiles\sym, and select NIGBTikw75n65eh5.asy.

Wolfspeed C3M0015065D SiC MOSFET

Wolfspeed has an application note, *Wolfspeed SiC MOSFET LTspice Model Quick Start Guide*, that is an excellent reference for adding Wolfspeed SiC MOSFETs to LTspice and using them in circuits. This file is included in the Documentation directory created after the LTspice model files are downloaded and unzipped. I will document my installation of the C3M0015065D SiC N type MOSFET.

We want to first add both the library and symbol files to the custom directories I created and are visible to LTspice. I copy the SPICE file C3M0015065D.lib from the Discretes directory downloaded in the Wolfspeed .zip file to our custom working directory C:\files\ltSpiceFiles\lib\. Since the C3M0015065D is a MOSFET in a TO247-3L (three leg) package, I want to customize the Wolfspeed generic three pin package for this device. I copy the Wolfspeed supplied symbol file, nmos_TO247_3L.asy, to our custom working directory, C:\files\ltSpiceFiles\sym\ and then open the file from within LTspice, using either File->Open, Ctrl-O, or the Open icon on the toolbar. (One may need to Zoom to Fit (View->View to Fit, Space, or the Zoom to Fit icon on the toolbar) in order to center the icon on the screen. Ctrl-A then opens the Symbol Attribute Editor. I change the Value from NMOS to C3M0015065D. I enter C3M0015065D, the name of the .subckt in the .lib file, as the value of the SpiceModel attribute. I change the description to N-Channel C3M0015065D MOSFET transistor. Then I enter C3M0015065D.lib, the name of the file that contains the model, as the value of the ModelFile attribute. (I know that the MOSFET will work in LTspice when I enter these attributes. I did not test to see if the subcircuit would work with fewer attributes.) Finally, I save the edited file into our personal sym folder.

Model

Some vendors do not use a subcircuit for an IGBT; rather they use a phenomenological model, developed at NIST, that is new since the original Berkeley SPICE installation. PSpice, a product of Cadence/Orcad was the first variation of SPICE to support this model. Since this is not a standard SPICE model, some vendors refer to their IGBT models as P-Spice models. LTspice also supports this IGBT model. We implement this IGBT model differently than we do a subcircuit. The reference for this inclusion is in the two help pages

LTspice->Help->Circuit Elements->Z. MESFET and IGBT Transistors

LTspice->Help->F.A.Q.->Third-party Models

I was able to use these model files by trivially modifying the .MODEL files and adding a SPICE directive to the schematic.

As with the subcircuit, I first copied the NIGBT.asy file and renamed the copy. I kept the SYMATTR Prefix Z line to signify this file contained the parameters for an IGBT and then I replaced the two existing lines describing attributes
SYMATTR Value NIGBT
SYMATTR Description N-Channel IGBT symbol for a model you supply.
with the two attribute lines
SYMATTR Value IXGT35N120B
SYMATTR Description N-Channel Littlefuse IXGT35N120B
to create the file NIGBTIXGH35N120C.asy. I also included the name of the model file
SYMATTR ModelFile IXGT35N120B.lib
(I tried to tell LTspice where the file was located by setting other attributes, but I was unable to get this to work. Check if adding ModelFile fixed this.)

The vendor supplied file, IXGH35N120C.lib contains (note the typos in the comments for the GH rather than GT device):

```
*DEVICE=IXGH35N120B,NIGBT
```

```
* IXGH35N120B NIGBT model
```

```
* updated using Model Editor release 9.2.1 on 10/16/03 at 12:42
```

```
* The Model Editor is a PSpice product.
```

```
*$
```

```
.MODEL IXGT35N120B NIGBT
```

```
+ TAU=168.81E-9
```

```
+ KP=7.6109
```

```
+ AREA=35.000E-6
```

```
+ AGD=14.000E-6
```

```
+ WB=117.00E-6
```

```
+ VT=6.1400
```

```
+ MUN=1.0000E6
```

```
+ MUP=250
```

```
+ BVF=6.2000
```

```
+ KF=.5005
```

```
+ CGS=16.344E-9
```

```
+ COXD=22.190E-9
```

```
+ VTD=-5
```

```
*$
```

In order to make this vendor supplied file compatible with LTspice, I needed to add the line

```
ZGT35120 C G E IXGT35N120B
```

immediately preceding the .MODEL line so that LTspice could recognize that the file contained the information to describe a specialized IGBT model. (I also added two comment lines above the Z line.)

Check if adding ModelFile fixed this. As stated above, I was unsuccessful in getting LTspice to find the location of the .lib file by including the information in the .asy file. Therefore I had to add the SPICE directive

```
.lib IXGH35N120C.lib
```

to the schematic.

Modifications to Base Setup

The first modification is to add the subcircuit SPICE models that define the custom devices we want to use to the installation

- Copy the attached files IGBT_650V_TRENCHSTOP5_H5_L1.lib, C3M0015065D.lib, and IXGT35N120B.lib to the directory C:\files\ltSpiceFiles\lib\

The next modification is to add the files that contain the symbols for the IGBT to the installation

- Copy the attached files NIGBTikw75n65eh5.asy, nmosC3M0015065D.asy, and NIGBTIXGT35N120B.asy to the directory C:\files\ltSpiceFiles\sym\

SPICE Design Constraints

One of the first steps SPICE takes in solving its simulation equations is to find a DC operating point. Therefore, you must ensure that all of your circuits have a finite DC solution. This may involve adding additional components to the circuit.

For example, an inductor cannot be placed in parallel with a voltage source, since a non-zero DC voltage source will create a linearly increasing current, without bound, in the inductor. One can solve this problem by placing a very low resistance in series with the source and inductor. The low resistance will mean that a finite DC solution exists, even if the magnitude of the calculated DC current is unphysical. Similarly, one cannot feed a current source into a capacitor. One solves this problem by adding a very large resistance in shunt with the current source and capacitor.

Useful SPICE Tools

Parameters

There are often many correlated parameters in a circuit. Changing the value of one requires that others also be changed. It is therefore useful to define and use parameters. Parameters make the schematic more readable and easier to maintain.

One adds parameters to a schematic by adding the SPICE directive *.param*. This is selected either by just typing “.” on the schematic, selecting Edit->. or by selecting the .t icon. One can use arithmetic as well as many numeric functions in *.param* statements. In general, one is unable to perform arithmetic in arguments of other SPICE directives. Variables in *.param* statements can be defined in terms of either constants or other parameters. It is useful to enclose any arithmetic statements in parenthesis to ensure that the order of evaluation is completed as desired. In order for a *.param* or any other statement to use another parameter, the parameter must be enclosed in curly braces {}.

I did not find the modification of existing parameter values intuitive. Right clicking on the parameter statement opens the “.param Statement Editor”. I could modify the string in the bottom message box, but I could not figure out how to get this information back into the schematic. What did work was to “Cancel” the .param Statement Editor, which opened up an “Edit Text on the Schematic” window. I could successfully edit the .param statement in this window. An “OK” would update the parameter on the schematic. Ctrl-Right-Click on the .param statement takes me directly to the “Edit Text on the Schematic” window.

The LTspice help documents, LTspice->Help->LTspiceSimulator->Dot Commands->.PARAM gives the various arithmetic functions available for parameters as well as some rules for usage.

Net Labels

Net labels are useful in SPICE for two reasons. First a net label allows the user to easily identify the nodes with a meaningful name, rather than a default node number selected by LTspice. This knowledge is useful when selecting nodes for plotting.

The other reason is to enable cleaner schematic drawings. Rather than connecting all links to a dense node together, one can name that node with a net label and then connect the links to different instances of the same net label.

A net label can be inserted either by typing the “n” key on the schematic, Edit->Label Net, or the “net” icon on the menu.

Parametric Analysis

This section describes how to perform a parametric analysis on the resistor in a Praeg filter.

Vary the parametric component by right clicking on it. Under Component *Properties (Resistor, capacitor, or Inductor)* enter **{R}**, or **{C}**, or **{L}** in the *Resistance, Capacitance, or Inductance* box. This tells LTspice the component value is a global parameter. Close the component box by selecting the OK button.

Add a SPICE directive to the circuit in one of three ways. Depress the period key, ., use the .t command at the right hand side of the menu, or select the *SPICE Directive* from the *Edit* pull down menu. All of these ways open a dialog box *Edit Text on the Schematic*. Ensure highlight of the *SPICE Directive* radio button. Type the needed command (see the example below) then left click the directive onto the circuit diagram.

For example, the command `.step dec param R 0.01 1.0 4`, steps the parameter **R** logarithmically from 0.01 ohms to 1.0 ohms in four steps per decade when the *Run* command is chosen. Display the plots for all of the steps by then selecting the voltage or current of interest and displaying the plot. To select only specific plots, choose *Select Steps* to display the traces of interest from the *Plot Settings* drop down menu.

In order to step through the parameter in linear increments, we modify the command to `.step param R 0.05 0.15 0.02`, which starts with **R** at 0.05 ohms, and steps in increments of 0.02 ohms to 0.15 ohms.

To return to the original value we can either delete the `.step` command and replace it with a command used to set **R** `.param R 0.1`, or remove the command altogether, to return to the damping resistor properties and replace **{R}** with the desired value.

Step Response

We can series cascade two voltage sources to obtain a step response at a fixed working point. For example, if we are analyzing a pulse-width modulated power supply, we set the first pulsed voltage source to the original duty cycle and wait for the output to reach steady state. We limit the total number of pulses from this supply with the *Ncycles* parameter. Then we start the second pulsed voltage source with the final duty cycle. We delay its turn on time using the *Tdelay* parameter.

Fourier Transform

We can plot the Fourier transform of a displayed waveform. Once displayed, select the plot window, either by selecting the *.raw* tab or by selecting the body of the plot. Then either from the *View* tab or by right clicking the plot body and selecting *View*, select *FFT* to compute and plot an FFT of the desired parameter.

LTspice Utilities

Keyboard Shortcuts

LTspice defines keyboard shortcuts. A listing of the current settings can be found in the Help->Keyboard Shortcut Cheat Sheet. There are shortcuts for the Schematic, Symbol, WaveForm, and Netlist functions of LTspice. If desired, you can also change most of the shortcuts on this page.

Plot Tools

- You can plot some functions of existing waveforms on the WaveForm window. Select Plot Settings -> Add trace, or type A. A window with the available traces will appear. Select the traces of interest and apply the operations, such as addition, subtraction, multiplication, or division to these traces in the Expressions to add box.
- Holding down the Alt key and selecting a line will plot the current on that line.
- You can plot a differential measurement between two nodes by selecting the first node and then dragging the cursor to the second node. During this action a red probe is visible on the first node and a black probe is visible on the second node.
- You can plot the power dissipated in an element by holding Alt and then clicking on the body of the desired element. A thermometer will appear to signify power. The dissipated power is calculated by summing, with the correct signs, all of the real power flows into the element.
- On a multiplot graph of the same quantity, such as a Bode plot, you can hide one of the traces by moving the cursor over the desired axis until the cursor changes into a ruler and then selecting the axis. A button at the bottom allows you to Don't plot phase. Pressing that button removes the phase from the plot. Even if that quantity of the plot is hidden, you can make it reappear by dragging the cursor to the now vacant axis area and selecting the axis when the ruler appears. Selecting OK returns the trace and axis to the plot.
- If you have plotted a family of traces as the result of a parametric analysis, you can select which of the family to display on the plot using the Plot Settings -> Select Steps command (also available by right clicking on the plot window), then selecting the traces from the menu.
- You can enable the cursor to show the plotted values at each step along the plot by placing the cursor over the title of the plotted quantity and selecting it. A crosshairs will appear on the screen and a box will pop up with the values of the plot corresponding to the value of the

independent parameter selected by the cross-hairs. Multiple cursors can be enabled. You associate different labels with different cursors by right clicking on the label and selecting the Attached Cursor.

- You can find global values by right placing the cursor over the label, depressing Ctrl, and then selecting the label. If the plot is a time plot, it will give an average value and an area under the curve (total energy for a power curve). If the plot is a frequency plot, the calculation will give the 3 dB bandwidth of the system. To have LTspice calculate these values over a desired time span, move the cursor down to the time axis until a ruler appears. Then right click and a window appears that will enable you to enter the desired times at the left and right edges of the plot.

Graphic Output

You can produce an .emf file of a window, either a schematic or plot window, by selecting the window, then using the menu selection Tools->Write image to .emf file.

References

There are several references useful in learning LTspice.

The first place to start, of course, is the documentation provided by Analog Devices. This includes the file *LTspiceGettingStartedGuide.pdf* and the Help from within LTspice.

LTspice also has many useful example files, separated into two categories. The first category are generic example files. These are in C:\Users\username\AppData\Local\LTspice\examples\Educational. They have examples of general SPICE circuits that are useful models for many types of general circuits. The other category is meant for applications of Linear Technology/Analog Devices devices, especially switch mode power supply controllers. These files are in

C:\Users\username\AppData\Local\LTspice\examples\Applications. One can either navigate to the directory listed above and open a file or use the component menu to find the device of interest. Once that device is selected, one can select "Open this macromodel's test fixture" to load the appropriate test jig.

Analog Devices has many resources, including SPICE files, application notes, and videos on their web site, with a link to many of them at

<https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html>, the site from which we downloaded the program.

Analog Devices also has a large number of demo circuits for many common applications of their parts at <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator/lt-spice-demo-circuits.html>

(Analog Devices is very closely linked to LTspice because the software was developed by Linear Technology to support their design efforts and those of their customers. When Analog Devices bought Linear Technology, it also assumed responsibility for LTspice and developed models of their integrated circuits that work with LTspice. Note that other vendors, such as Texas Instruments, support other versions of SPICE or SPICE like simulators and supply models of their integrated circuits that work on those simulators.)

Another resource is an online tutorial hosted by Laurier University, in Waterloo, Ontario, Canada. This was mainly written for LTspiceIV, so the screens shown on the page are obsolete.

<http://denethor.wlu.ca/ltspace/>

There also is a rather complete 1993 SPICE reference manual from UC Berkeley

http://www.gianlucafiori.org/appunti/Spice_3f3_Users_Manual.pdf