

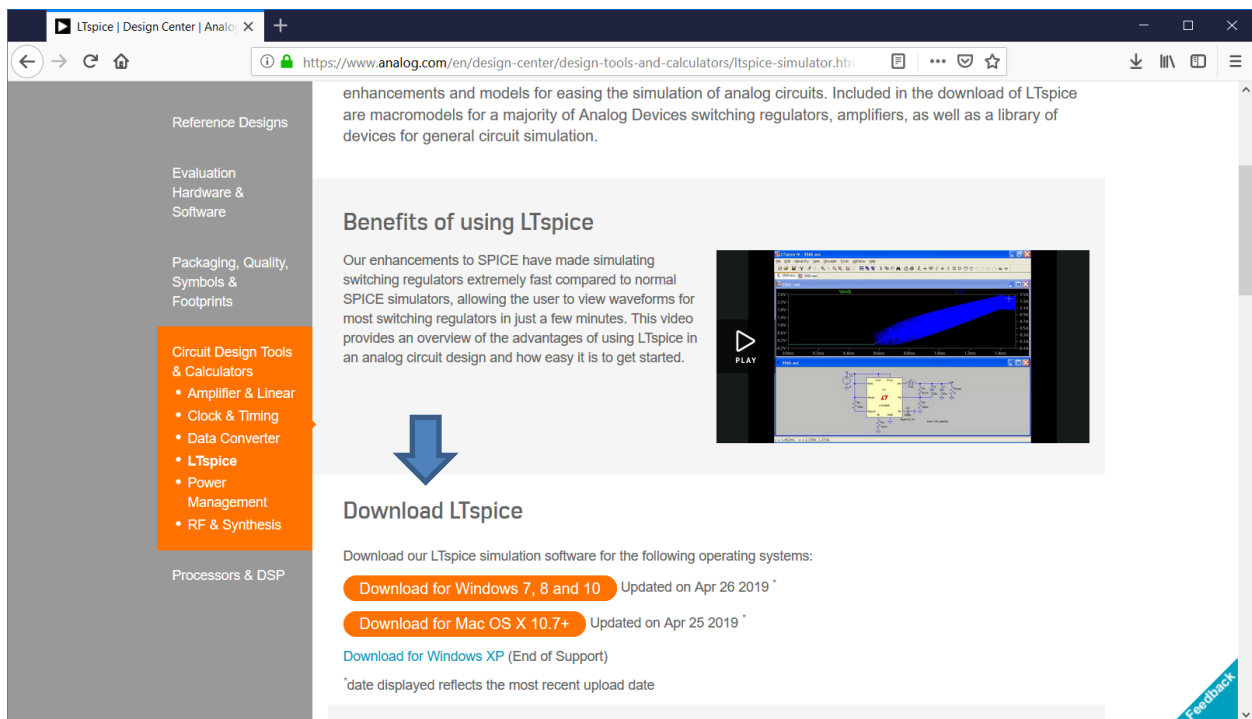
# Instructions to Install LTspice XVII

## Introduction

We will use LTspice XVII 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.

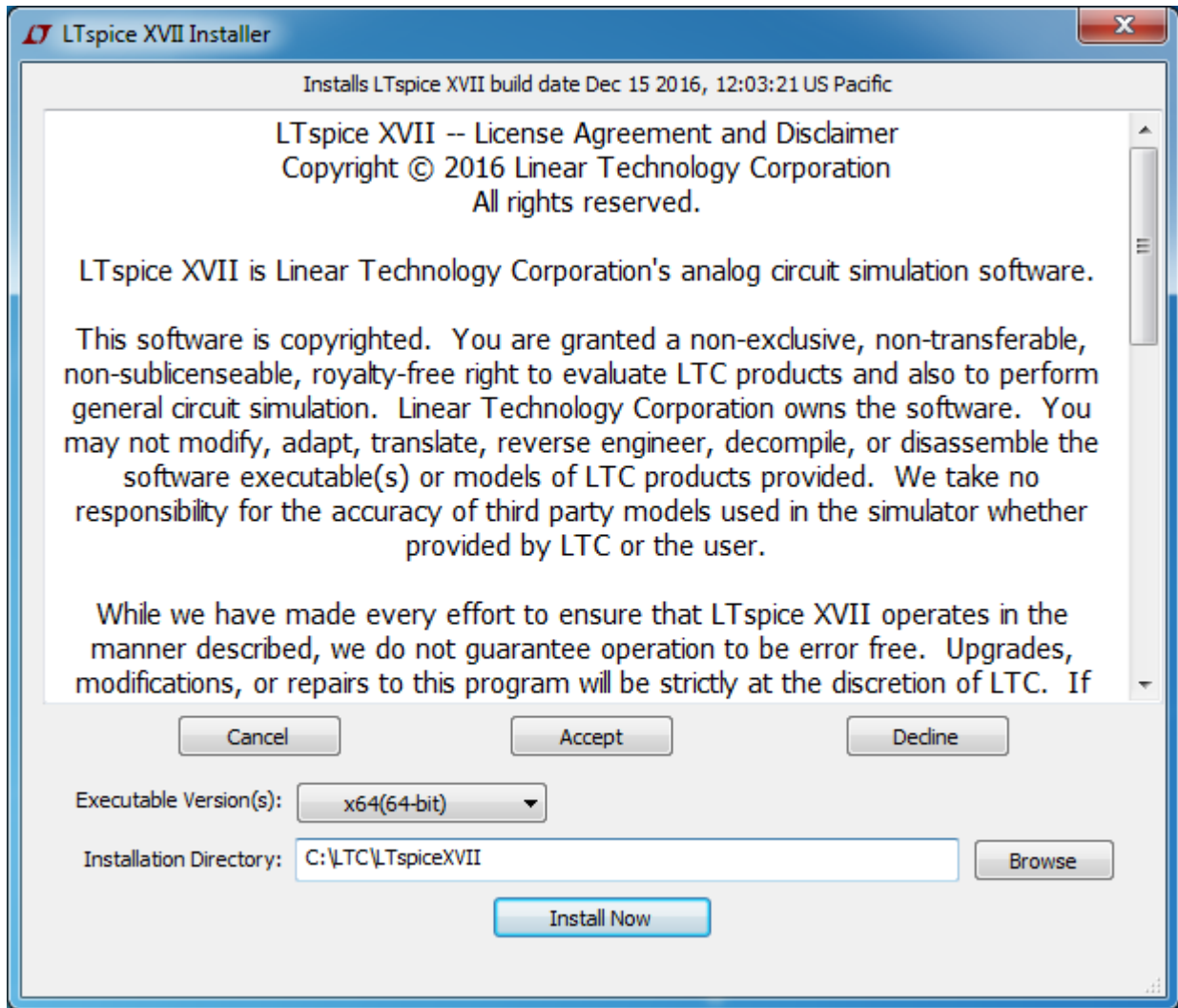
## Standard Installation

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



The screenshot shows a web browser window displaying the LTspice simulator download page. The page features a navigation menu on the left with categories like Reference Designs, Evaluation Hardware & Software, and Circuit Design Tools & Calculators. The main content area includes a section titled 'Benefits of using LTspice' with a video player showing a circuit simulation. Below this is a 'Download LTspice' section with three download buttons: 'Download for Windows 7, 8 and 10' (updated Apr 26 2019), 'Download for Mac OS X 10.7+' (updated Apr 25 2019), and 'Download for Windows XP (End of Support)'. A feedback button is visible in the bottom right corner.

Select *Download LTspice XVII for Windows 7, 8 and 10* (if on a Windows machine. *LTspice* is also available for the Mac.). You will be prompted to save the binary file *LTspiceIV.exe* from <http://ltspice.analog.com>. After the file is downloaded, select it to run the installation program, agreeing to the standard Microsoft query about installing a file downloaded from the internet. (You may need administrative privileges to install the program.)



Accept the license agreement, change the installation directory from the default C:\Program Files\LTC\LTspiceXVII to C:\LTC\LTspiceXVII, and then Install Now.

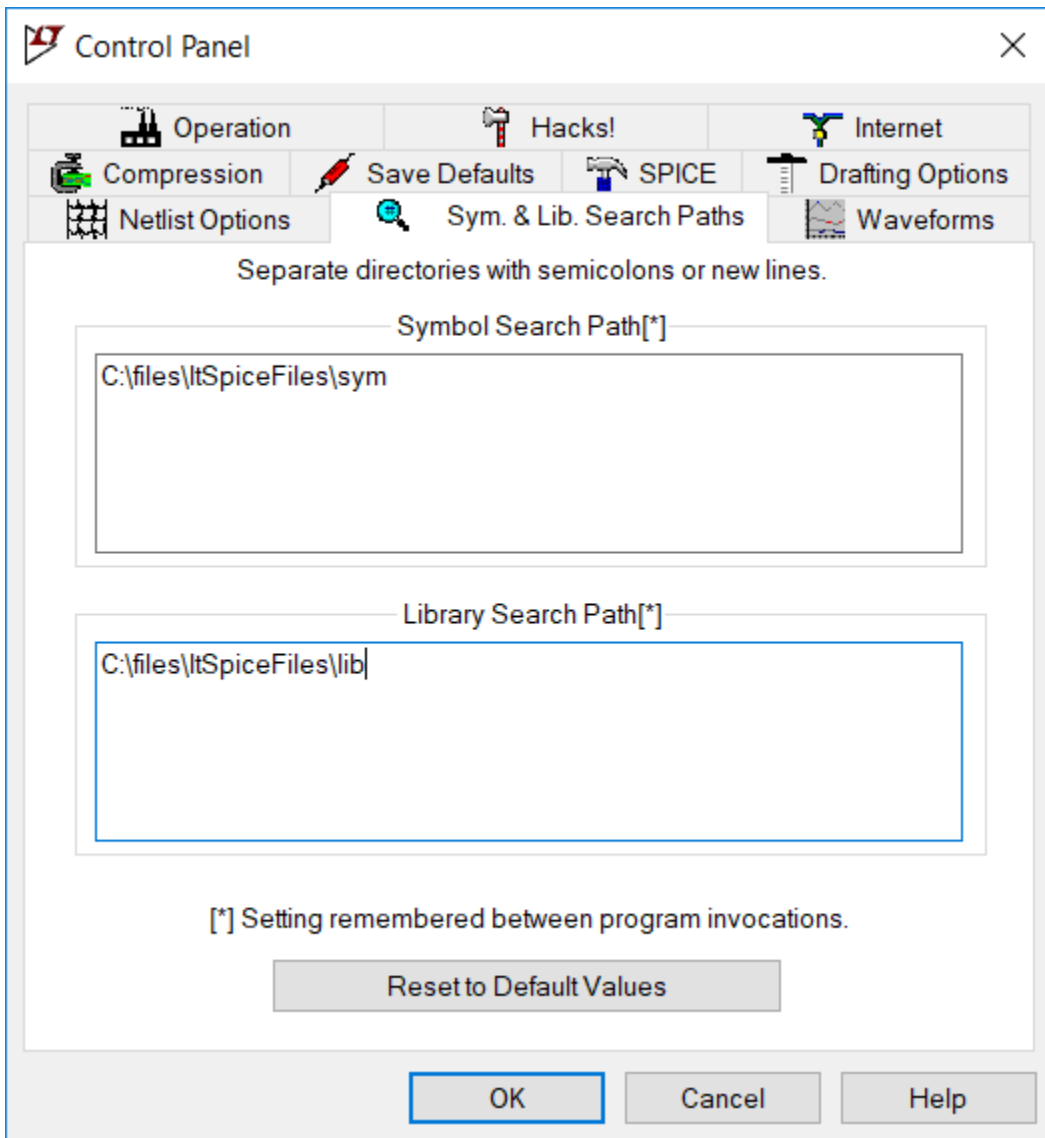
There should now be a directory C:\LTC\LTspiceXVII that includes several files, including XVIIx64.exe. The shortcut created by LTspice XVII runs this executable for a 64 bit installation. The directory also contains two subdirectories, examples and lib. On Windows 10, LTspice also creates a copy of these directories in the user space C:\Users\username\Documents\LTspiceXVII. This latter location is where LTspice looks for the files it will use in building the schematics and simulations. Finally, LTspice creates an initialization file, C:\Users\username\AppData\Roaming\LTspiceXVIII.ini, that contains the setup information.

## Customization

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. We create two

subdirectories, one for symbols that we will create and one for library parts. We create 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->Control Panel->Sym. & Lib. Search Paths  
entering the directory names and then OK. These paths are stored as the variables SymbolSearchPath  
and LibrarySearchPath, respectively, in LTspiceXVII.ini.



We will be creating IGBT files, so copy the standard symbol for an N channel IGBT into the  
C:\files\ltSpiceFiles\sym  
directory. From  
C:\LTC\LTspiceXVII\lib\sym\Misc  
copy NIGBT.asy and PIGBT.asy to  
C:\files\ltSpiceFiles\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 analog models, such as diodes, bipolar transistors, junction FETs and MOSFETs. These files are all, by default, located in the library *C:\Users\name\Documents\LTspiceXVII\lib\cmp* (for component). All of these files are ASCII files with a series of models similar to the one displayed above for the 2N2222 bipolar transistor. 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.

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 edit them from within LTspice.

### Symbol (.asy) File

We will examine NIGBT.asy to see the elements of a typical symbol file. 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, three for a bipolar transistor, MOSFET, IGBT, etc. In addition to its location, each pin requires several attributes to be defined in the .asy file. One is the PINNAME, the identifier that will be placed on the schematic. Another is the SpiceOrder, which tells the GUI how to associate the pins with the terminals defined in the actual spice definition file, other text files that reside in the lib directories and which tell LTspice the voltage and current relations at the various terminals.

Finally,

NIGBT.asy is a text file containing

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 to 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. If LTspice does not support this model, then it cannot be used in LTspice. However, LTspice does support some of these models, such as a common model for an IGBT developed at NIST. In that case, one adds the model differently from the way in which one adds a subcircuit. (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. All models compatible with Spice 2, Spice 3, or LTspice should work.)

The Help files that come with LTSpiceXVII 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

<https://www.analog.com/en/education/education-library/videos/5579239882001.html>

## Subcircuits

### *Infineon IRGP4068D*

Several semiconductor vendors have LTspice compatible models on their website. (ON Semiconductor’s site is <https://www.onsemi.com/design/resources/design-resources/models>. From this site search for *ltspice*. Infineon has models at links on <https://www.infineon.com/cms/en/tools/>.) As an example, we will import a model from Infineon.

Infineon supplies a file, *irgp4068d\_epbf.spi*, for this IGBT. This file, as are all spice input files, is an ascii text file. This file is readable; some spice files are encrypted to protect the vendor’s intellectual property. Infineon chooses to use the .spi extension for this spice model, but other suffixes are possible. Infineon has built a subcircuit for this device using resistors, capacitors, inductors, an NMOS transistor, a PNP transistor, and diodes. The subcircuit details the connections between the terminals of the elements and the file supplies the relevant parameters for each of these elements. We save this file in our personal lib folder.

We first want to create 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 *NIGBTirgp4068d.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 XVII->Schematic Capture->Creating New Symbols->Adding Attributes)

The symbol file needs to have several attributes defined:

- *SYMATTR Prefix X*  
The *X* tells LTspice that this symbol is a sub-circuit
- *SYMATTR Value name*

where *name* is replaced by the desired name on the schematic, for example *irgp4068d\_epbf*. (There are some rules I do not understand. I had to make the value name identical to the subcircuitname in order for LTspiceXVII to correctly parse this subcircuit.)

- *SYMATTR Description descript*  
where *descript* is a user supplied phrase. I am not sure how this value is used.
- *SYMATTR ModelFile filename*  
where *filename* is the name of the ASCII file containing the definition of the sub-circuit, for example *irgp4068d\_epbf.spi*. This holds true if the SPICE file is in the sub directory. (I do not know how this would be modified if it is placed elsewhere. I would try to put the absolute path name, as described in the LTspice help on the .lib command to see if that works.)
- *SYMATTR SpiceModel subcircuitname*  
where *subcircuitname* is the name in the sub-circuit definition in *filename*, for example if the definition line in *filename* is *.SUBCKT irgp4068d\_epbf 1 2 3*, then set *subcircuitname* to *irgp4068d\_epbf*.

SYMATTR Prefix X

SYMATTR Value irgp4068d\_epbf

SYMATTR Description N-Channel IGBT Infineon irgp4068d.

SYMATTR ModelFile irgp4068d\_epbf.spi

SYMATTR SpiceModel irgp4068d\_epbf

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 *NIGBTirgp4068d.asy*.

## 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. LTspiceXVII also supports this IGBT model. We implement this IGBT model differently than we do a subcircuit. The reference for this inclusion is in

LTspice XVII->LTspice->Circuit Elements->Z. MESFET and IGBT Transistors  
LTspice XVII->LTspice->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 IXGH35N120C

SYMATTR Description N-Channel IXYS IXGH35N120C

to create the file NIGBTIXGH35N120C.asy. (I tried to tell LTspiceXVII where the file was located by setting other attributes, but I was unable to get this to work.)

The vendor supplied file, IXGH35N120C.lib contains:

\* PSpice Model Editor - Version 10.0.0

\*\$

\*DEVICE=IXGH35N120C,NIGBT

\* IXGH35N120C NIGBT model

\* updated using Model Editor release 9.2.1 on 03/05/04

\* The Model Editor is a PSpice product hem.

.MODEL IXGH35N120C NIGBT

+ TAU=142.40E-9

+ KP=17.716

+ AREA=35.000E-6

+ AGD=14.000E-6

+ WB=117.00E-6

+ VT=3.8900

+ MUN=4.2000E3

+ MUP=360

+ BVF=6.5310

+ KF=.5005

+ CGS=23.422E-9

+ COXD=22.426E-9

+ VTD=-5

\*\$



In order to make this vendor supplied file compatible with LTspiceXVII, I needed to add the line  
Z35120 C G E IXGH35N120C  
immediately preceding the .MODEL line so that LTspiceXVII could recognize that the file contained the information to describe a specialized IGBT model.

As stated above, I was unsuccessful in getting LTspiceXVII 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 an IGBT to the installation

- Copy the attached files irgp4063dpmf.spi, irgp4068d\_epbf.spi, IXGH35N120C.lib, and ixgx120n60a3.lib to the directory C:\LTC\LTspiceIV\lib\

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

- Copy the attached files irgp4063dpmf.asy, NIGBTirgp4068d.asy, NIGBTIXGH35N120C.asy, and NIGBTIXGX120N60A3.asy to the directory C:\LTC\LTspiceIV\lib\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 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 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 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. One adds parameters to a schematic by adding the SPICE directive *.param*. 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 in order to ensure that the 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 LTspiceXVII help documents the allowable mathematical functions that can be used in the *.param* statements.

### 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 LTspiceXVII. 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 from the “A” icon on the menu, the Label Net selection of the Edit pull-down menu, or with the F4 key.

### 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 s key, use the .op command at the right hand side of the menu, or select the *SPICE Directive* from the *Edit* pull down menu. All of these ways opens 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

### Hot Keys

LTspice defines shortcut keys that it calls Hot Keys. A listing of the current settings can be found in the Control Panel. The control panel can be accessed by selecting the hammer icon or by choosing Tools -> Control Panel. Once in the control panel, select the Drafting Options tab and the Hot Keys button. There you can find tabs for the Schematic, Symbol, Waveform, and Netlist functions of LTspice. If desired, you can also change the hot key shortcuts on this page.

### Plot Tools

- You can plot some functions of existing waveforms. Select Plot Settings -> Add trace. 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. 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 cross-hairs 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.

## 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.

LTspiceXVII also has many useful example files, separated into two categories. The first category are generic example files. These are located in C:\LTC\LTspiceXVII\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 located in C:\LTC\LTspiceXVII\examples\jigs. 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/design-center/design-tools-and-calculators/ltspice-simulator.html>

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.

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

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

[http://www.gianlucafori.org/appunti/Spice\\_3f3\\_Users\\_Manual.pdf](http://www.gianlucafori.org/appunti/Spice_3f3_Users_Manual.pdf)

as well as some other documentation

[http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/UserGuide/overview\\_fr.html](http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/UserGuide/overview_fr.html)

There are also lectures online on the types of algorithms used to model devices and solve the circuit equations in SPICE

<http://www3.imperial.ac.uk/pls/portallive/docs/1/7292571.PDF>

<http://www3.imperial.ac.uk/pls/portallive/docs/1/7292572.PDF>

<http://www3.imperial.ac.uk/pls/portallive/docs/1/7292573.PDF>