

Software Install Notes

David W. Hawkins- anaconda section modified by Michael Medrano (David.W.Hawkins@jpl.nasa.gov)

Version 0.1

December 29, 2020

Contents

1	Introduction	3
2	Development Machines	4
3	Operating Systems	5
3.1	Windows 7/10	5
3.2	Linux	5
4	Environment Variables (Licensing)	5
5	Development Tools Support	6
5.1	Mentor/Siemens Support Site	6
5.2	Synplify Support Site	6
5.3	Xilinx	6
6	Windows Installation Notes	7
6.1	Cygwin	7
6.2	Cygwin ssh keys	7
6.3	Mentor QuestaSim 2019.4	8
6.4	Mentor Questa Verification IP 2019.4	8
6.5	Mentor Questa and QVIP Setup	9
6.6	Mentor HDL Designer Series 2019.4	10
6.7	Xilinx ISE 13.2	10
6.8	Xilinx ISE 13.2 Virtex-5QV SIRF Overlay	11
6.9	Build Xilinx ISE 13.2 Simulation Libraries	11
6.10	Xilinx ISE 14.7	13
6.11	Xilinx Vivado 2019.2	13
6.12	Build Xilinx Vivado Simulation Libraries	13
6.13	MATLAB	14
6.14	MATLAB Support Packages	15
6.15	Latex (MiKTeX)	16
6.16	Inkscape	16
6.17	IrfanView	16
6.18	VirtualBox	16
6.19	Git LFS	17
6.20	Anaconda (Python)	17
6.21	Synopsys Synplify Pro and Premier	19

The technical data in this document is controlled under the U.S. Export Regulations;
release to foreign persons may require an export authorization.

7	VirtualBox Virtual Machines	20
7.1	Development Tools Virtual Drive (/software)	20
7.2	Centos	20
7.3	Ubuntu	20
7.4	Xilinx Vivado 2019.2	20
7.5	Xilinx PetaLinux 2019.2	20
8	Full Ubuntu Installation on 64GB USB Drive	21

1 Introduction

This document contains software installation notes for my JPL development machines.

2 Development Machines

The following are details of my JPL development machines.

LMC-049267 (retired/returned 1/1/2020)

- Provider: Lockheed Martin Corporation (LMC)
- Operating System: Windows 7 Enterprise (64-bit) Service Pack 1
- Ethernet MAC: 9C-EB-E8-20-E9-62
- Ethernet IP: 137.78.253.177
- WiFi MAC: CC-3D-82-F4-40-39
- WiFi IP: 137.79.220.132
- Windows Product ID: 00392-918-5000002-85349
- Processor: Intel Core i7-4712HQ CPU @2.30GHz
- RAM: 16GB

MT-210792 (received 12/16/2019)

- Provider: ManTech
- System Model: Dell Precision 7540
- Operating System: Windows 10 Enterprise
- OS Version: 10.0.18632 Build 8632
- Ethernet MAC:
- Ethernet IP:
- WiFi MAC:
- WiFi IP:
- Windows Product ID:
- Processor: Intel Core i9-9980HK CPU @2.40GHz
- RAM: 64GB
- UEFI Secure Boot: On

The UEFI Secure Boot setting needs to be changed to off when working with FPGA boards attached to the laptop via a Thunderbolt 3 to PCIe bridge. Ubuntu can be booted with UEFI Secure Boot enabled, however, Linux places the kernel in lockdown mode, and disables hardware access via `/dev/mem` and PCI BARs, making it hard to develop PCIe device drivers.

Windows 10 will not boot without Secure Boot enabled, so the BIOS setting needs to be changed back after booting Ubuntu for PCIe development.

3 Operating Systems

3.1 Windows 7/10

Install all tools into `c:/software`, so that it is clear what has been custom installed into the machine.

3.2 Linux

Users would typically install custom tools into `/opt`, however, for VMs it is convenient to install tools onto a separate virtual drive, so that the drive can be mounted on multiple VMs, eg., both Centos and Ubuntu. Since `/opt` is already in use on most Linux distros, the tools should be installed on a different mount point, eg., `/software`.

4 Environment Variables (Licensing)

```
MGLS_LICENSE_FILE      = 2020@cae-lm-mentor1
XILINXD_LICENSE_FILE   = 2200@cae-lmgr1,2200@cae-lmgr2,2200@cae-lmgr3
SNPSLMD_LICENSE_FILE   = 9998@cae-lmgr1,9998@cae-lmgr2,9998@cae-lmgr3
```

5 Development Tools Support

5.1 Mentor/Siemens Support Site

- Support site: <https://support.sw.siemens.com/en-US/signin>
- Site ID: 15208

5.2 Synplify Support Site

- Support site: <https://solvnet.synopsys.com>
- Site ID: 1993

5.3 Xilinx

Download site: <https://www.xilinx.com/support/download.html>

Xilinx ISE 13.2 download:

- Under *Version*, click on the *ISE Archive* link.
- Click on the ISE 13.2 link.
- Download *ISE Design Suite - 13.2 Full Product Installation, All Platforms*
[Xilinx_ISE_DS_13.2_O.61xd.0.0.tar](#)
- Download *Software Development Kit - 13.2 Full Product Installation, All Platforms*
[Xilinx_SDK_13.2_O.61xd.0.0.tar](#)

Xilinx ISE 14.7 download:

- Under *Version*, click on the *ISE Archive* link.
- The archive link *14.7 Windows 10* is for an ISE version that only supports Spartan-6 devices.
- Click on the *14.7* link (below the *14.7 Windows 10* link).
- Download *ISE Design Suite - 14.7 Full Product Installation, Full DVD Single File Download Image*
[Xilinx_ISE_DS_14.7_1015_1.tar](#)

Vivado 2019.2 download:

- From the main download page scroll down to *Vivado HLx 2019.2: All OS installer Single-File Download* (26.55GB) and download
[Xilinx_Vivado_2019.2_1106_2127.tar.gz](#)
- Download Update 1 (9.03GB)
[Xilinx_Vivado_Vitis_Update_2019.2.1_1205_0436.tar.gz](#)

6 Windows Installation Notes

6.1 Cygwin

Install the 64-bit version of Cygwin

- Install into `c:\cygwin64`
- Change the package directory to `c:\cygwin64\packages`
- Select a mirror, eg., `http://mirrors.xmission.com`
- Change the *View* pull-down to full.
- Browse and add packages, eg.,
 - `gcc-g++ 7.4.0-1`
 - `git 2.21.0-1`
 - `make 4.2.1-2`
 - `libreadline-dev 7.0.3-3` (`libreadline7 7.0.3-3` was already selected)
 - `tree 1.7.0-1`
 - `vim 8.1.1772-1`

After Cygwin installation `git --version` is 2.21.0, `gcc --version` is 7.4.0, and `make --version` is 4.2.1.

Cygwin TODO:

- Install the Cygwin packages for HDF5 and NetCDF or build from scratch (since the Cygwin packages are too old)?
- Git LFS (requires a separate Windows installation tool)

6.2 Cygwin ssh keys

Run `ssh-keygen` to generate ssh keys and copy the public key in `.ssh/id_rsa.pub` to your JPL github account.

6.3 Mentor QuestaSim 2019.4

The Windows installation of the Questa Verification IP Suite only supports the 32-bit version of QuestaSim.

- Install QuestaSim 32-bit
 - Run `questasim-win32-2019.4.exe`
 - Install to `c:\software\mentor\questasim_2019.4`
 - Do not add a desktop shortcut
 - Do not add the tool to the path
 - Do not install the hardware security key driver
 - When the installer completes, it opens the licensing setup tool. Just close it.
 - Setup the environment variable `MGLS_LICENSE_FILE`.
 - Start the tool from the Windows start menu and confirm it works, eg.,

```
QuestaSim> vsim -version
# Questa Sim   vsim 2019.4 Simulator 2019.10 Oct 15 2019
```
- Install gcc 32-bit
 - Unzip `questasim-win32-gcc-4.2.1-mingw32vc12.zip`
 - Move the folder `gcc-4.2.1-mingw32vc12` to `C:\software\mentor\questasim_2019.4`
- Install Register Assistant (UVM)
 - Run `regassistuvvm_2019.4_win.exe`
 - Selecting the QuestaSim installation `c:\software\mentor\questasim_2019.4` to install the tool into `c:\software\mentor\questasim_2019.4\RUVM_2019.4`
 - Complete the installation
- Install size (`C:\software\mentor\questasim_2019.4`): 1.92GB

6.4 Mentor Questa Verification IP 2019.4

- Run `setup.exe`
- The Mentor software installer will indicate the correct source folder, but the destination folder will be wrong (it will be the last place you installed a Mentor product).
- Change the installation folder to `c:\software\mentor\questasim_vip_2019.4`
- Complete the installation
- Install size (`C:\software\mentor\questasim_vip_2019.4`): 1.97GB

The QVIP installation also includes PCIe design examples (`2019.4_PCIE_starter_kit.tar.gz`) for Xilinx and Altera IP cores. These do not need to be installed, but should be a useful resource for PCIe development.

6.5 Mentor Questa and QVIP Setup

- Define the environment variables

```
QUESTA_HOME = c:/software/mentor/questasim_2019.4
```

```
QUESTA_MVC_HOME = c:/software/mentor/questasim_vip_2019.4
```

- Edit the QuestaSim file

```
c:/software/mentor/questasim_2019.4/modelsim.ini
```

and change MvcHome = \$MODEL_TECH/.. to MvcHome = \$QUESTA_MVC_HOME.

The file is read-only, so will need to be made read/write before editing, and then set back to read-only after editing.

Test the Questa and QVIP setup by running one of the AXI4-Stream tutorial examples. For example, under my Windows 7 machine, the multiply-add example output was

```
# Compiling C:/Users/dhawkins/AppData/Local/Temp/dhawkins@LMC-049267_dpi_3364\
win32_gcc-4.2.1\exportwrapper.c
# Loading C:/Users/dhawkins/AppData/Local/Temp/dhawkins@LMC-049267_dpi_3364\
win32_gcc-4.2.1\vsim_auto_compile.dll
# Loading C:/software/mentor/questasim_2019.4/uvm-1.2/win32\uvm_dpi.dll
# Loading c:/software/mentor/questasim_vip_2019.4/questa_mvc_core/win32_gcc-4.2.1/
libaxi4stream_IN_SystemVerilog_MTI_full.dll
#
# // Questa Verification IP
# // Version 2019.4_win : (axi4stream) 20191003 win32 10/17/2019:09:11
# //
# // Copyright 2007-2019 Mentor Graphics Corporation
# // All Rights Reserved.
# //
# // THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION
# // WHICH IS THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS
# // LICENSORS AND IS SUBJECT TO LICENSE TERMS.
#
# ** Note: (vsim-50000)
# -----
# MVC instances summary: MVCs 2, ends 8
#   /top/master/axi4stream_master_if ( AXI4_ID_WIDTH = 1, AXI4_USER_WIDTH = 64,
AXI4_DEST_WIDTH = 1, AXI4_DATA_WIDTH = 64 ) { master:1 slave:1 clock_source:1 reset_source:1 }
#   /top/slave/axi4stream_slave_if ( AXI4_ID_WIDTH = 1, AXI4_USER_WIDTH = 64,
AXI4_DEST_WIDTH = 1, AXI4_DATA_WIDTH = 64 ) { master:1 slave:1 clock_source:1 reset_source:1 }
# -----
#
```

This output shows the Questa and QVIP 2019.4 paths, and confirms gcc-4.2.1 works. The output under my Windows 10 machine was very similar (the machine name in the messages changed from LMC-049267 to MT-210792).

6.6 Mentor HDL Designer Series 2019.4

- Run `HDS_2019.4_win.exe`
- Install to `c:\software\mentor\HDS_2019.4`
- Install size (`c:\software\mentor\HDS_2019.4`): 1.54GB

6.7 Xilinx ISE 13.2

Xilinx ISE 13.2 is not officially supported by Xilinx under Windows 10, however, the Xilinx Answer Record [AR#62380: ISE Install - Guide to Installing and Running ISE 10.1 or 14.7 on a Windows 8.1 or Windows 10 machine](#) contains the steps needed to get the tool working. The `libportability` DLL fix needs to be applied to four directories for ISE 13.2 and ten directories for ISE 14.7. A Windows batch file to perform this operation exists on github for ISE 14.7 <https://github.com/cbureriu/xilinx-14.7-patch-for-Win10-32-64>. A bash equivalent of this script was created that uses the `XILINX` environment variable to determine which version of the tool to update.

Installation procedure:

- Extract `Xilinx_ISE_DS_13.2_0.61xd.0.0.tar`
- Run `xsetup.exe`
- Accept the default installation of *ISE Design Suite: System Edition*
- On the *Select Installation Options* page, deselect all options, including the cable drivers. Xilinx Vivado is supported under Windows 10, so installation of that tool will install the correct cable drivers.
- Install to: `c:\software\Xilinx\13.2` (uncheck the option to import preferences from previous versions).
- Set the Xilinx license manager environment variable.
- Set the environment variable
`XILINX = C:/software/Xilinx/13.2/ISE_DS/ISE`
- Open a Cygwin terminal and run the bash script, `xilinx_ise_win10_fix.sh`.
- Edit the PlanAhead desktop shortcut and start menu properties to add the 32-bit flag, `-m32`.
- **Licensing issue:** The ISE feature is not detected.

The *Xilinx License Configuration Manager* can be opened by selecting *Help*→*Manage Licenses*. The *Manage Xilinx Licenses* tab shows the licenses provided by the JPL CAE servers. Under Windows 7 this list has 70 entries: the *Search Order* column lists the license entries from 02 to 71. The ISE feature is search order 060.

The *Server Name* under Windows 7 was `2200@cae-lmgr1`, i.e., the first entry in the Xilinx license environment variable, and the *File Name* was the value in the Xilinx environment variable. The Windows 10 machine had *Server Name* empty and *File Name* set to `c:\software\Xilinx\13.2\ISE_DS\EDK\data\core_licenses\Xilinx.lic`. This implies that the CAE license servers were not resolved correctly.

Under Windows 10, `ping cae-lmgr1` and `ping cae-lmgr1.jpl.nasa.gov` did not resolve the server. Under Windows 7, `ping cae-lmgr1` resolved to `cae-lm-prod2.jpl.nasa.gov`. Under Windows 10, `ping cae-lm-prod2` resolved correctly.

Temporary work-around: change `XILINXD_LICENSE_FILE = 2200@cae-lm-prod2`

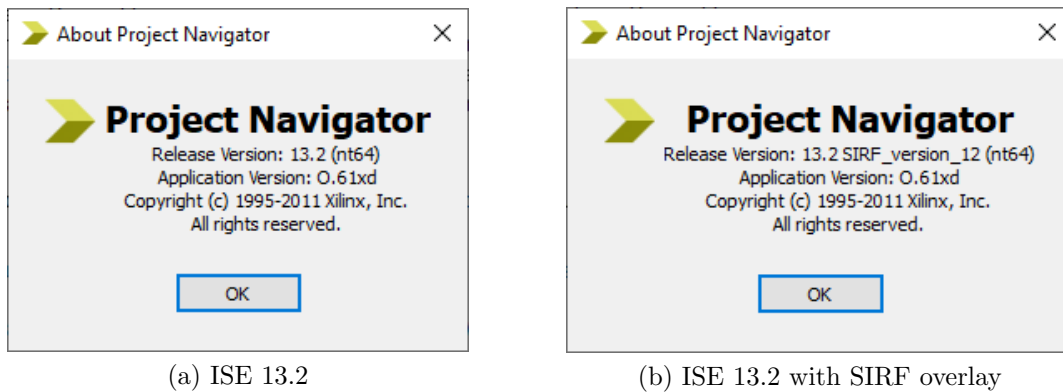


Figure 1: Xilinx ISE *Help*→*About* dialog for ISE 13.2 and ISE 13.2 with SIRF overlay.

6.8 Xilinx ISE 13.2 Virtex-5QV SIRF Overlay

SIRF overlay installation instructions:

- Unzip `sirf_overlay_13.2_2016_01_20.zip` (the version I received from Xilinx in 2016)
- Navigate down a couple of folders into the unzipped archive.
- Unzip `sirf_overlay_13_2_v12.zip`. The overlay folder contains ISE and PlanAhead folders.
- Copy the overlay folders into `c:/software/Xilinx/13.2/ISE_DS`, so they replace the existing folders.
- Open a Cygwin terminal and re-run the bash script, `xilinx_ise_win10_fix.sh`.

Figure 1 shows the ISE *Help*→*About* dialog for ISE 13.2, and for ISE 13.2 with SIRF overlay.

6.9 Build Xilinx ISE 13.2 Simulation Libraries

The following procedure was used to build the ISE 13.2 simulation libraries:

- From the Windows 10 start menu open the 64-bit version of the *Simulation Library Compilation Wizard* tool
- Select **QuestaSim**
- Set the QuestaSim path to `c:\software\mentor\questasim_2019.4\win32`
- Accept the remaining defaults and click *Next*
- Accept the default *Both VHDL and Verilog* and click *Next*
- Uncheck all devices and then select **Virtex5** and **Virtex-5QV**, and click *Next*
- Uncheck the **EDK Simulation Library** (it takes a long time to build), and click *Next*
- The output path for the compiled libraries should be (if it is not, then the Questasim path was not set correctly):
`c:\software\Xilinx\13.2\ISE_DS\ISE\<language>\<simulator>\<version>\<platform>`
Click *Launch Compile Process* to start the library compilation (which will fail). Exit the application (using the back buttons is not sufficient for the next steps).

```

*****
*                                     COMPILATION SUMMARY                                     *
*                                                                                         *
* Simulator used: questa                                                         *
* Compiled on: Tue Dec 24 12:12:15 2019                                         *
*                                                                                         *
*****
*      Library      | Lang | Mapped Name(s) | Err#(s) | Warn#(s) *
*-----*-----*-----*-----*-----*
* secureip          | verilog | secureip      | 0       | 0       *
*-----*-----*-----*-----*-----*
* unisim            | vhdl   | unisim        | 0       | 0       *
*-----*-----*-----*-----*-----*
* unisim            | verilog | unisims_ver   | 0       | 2       *
*-----*-----*-----*-----*-----*
* simprim           | vhdl   | simprim       | 0       | 0       *
*-----*-----*-----*-----*-----*
* simprim           | verilog | simprims_ver  | 0       | 2       *
*-----*-----*-----*-----*-----*
* xilinxcorelib     | vhdl   | xilinxcorelib | 0       | 260     *
*-----*-----*-----*-----*-----*
* xilinxcorelib     | verilog | xilinxcorelib_ver | 0       | 4       *
*-----*-----*-----*-----*-----*

```

Figure 2: Xilinx ISE 13.2 Library Compilation Summary.

- Library compilation fails due to the options present in the Xilinx ISE Questasim configuration file. The configuration file is generated by the first run through the library compilation process.
- Edit the file `c:\software\Xilinx\13.2\ISE_DE\ISE\compplib.cfg` and remove `-novopt` from within the file.
- Re-run the compilation wizard, and the *Launch Compile Process* will run.

Figure 2 shows the ISE 13.2 simulation libraries compilation summary.

RAMBo-FPGA build test:

- Check out the RAMBo repo


```
$ git clone --recursive git@github.jp1.nasa.gov:REASON-DES/RAMBo_FPGA.git
```
- Change into the RAMBo directory and source the setup script


```
$ cd RAMBo_FPGA
$ source setup.sh
```

Note: the script is sourced, not run via `./setup.sh`, so that the environment variables are defined for the current bash shell.
- Change into the `verif` directory and run the smoke test


```
$ cd verif $ vrun smoke -rmdb regression.rmdb -vrmdata $RAMBO_DATADIR
```
- Windows 10 will block the tool `vrun.exe` until you allow it.
- The smoke test takes about 10 minutes.

6.10 Xilinx ISE 14.7

When I installed both ISE 13.2 and 14.7 under Windows 7, the ISE 13.2 Impact and ChipScope tools stopped working: they could not detect JTAG cables or devices. The 14.7 version worked correctly, so I assumed that the 14.7 installation had done something to over-ride the 13.2 installation. For now, I have decided not to install 14.7 under Windows 10.

6.11 Xilinx Vivado 2019.2

Installation procedure:

- Copy the installation tar file to `c:/temp` and extract it (delete the files after installation)
- Run `xsetup.exe`
- Accept the default tool *Vivado HL System Edition*
- Install to `c:/software/Xilinx` (install for all users)
- The installation will also install DocNav, the Xilinx Documentation Navigator tool.
- Close the license manager GUI when it appears. No license setup is required, since the license setup for ISE also works for Vivado.
- The installer generates additional pop-up windows. If the installer appears to have stopped, move the main installer window to see if there is a pop-up waiting for user input.
- Create the environment variable
`VIVADO = c:/software/Xilinx/Vivado/2019.2`
My Tcl scripts targeting devices supported by Vivado use this environment variable to locate the simulation libraries.
- **TODO:** Install the software update?

6.12 Build Xilinx Vivado Simulation Libraries

Run the compilation tool and have it place the libraries in

`C:\software\Xilinx\Vivado\2019.2\compile_simlib\questasim_32bit`

It takes about 45 minutes to compile the simulation libraries.

6.13 MATLAB

Installation procedure:

- Login to the JPL CAE portal: <https://opencae.jpl.nasa.gov/portal>
- Select *Tools*→*All Tools*
- Scroll down and click on the MATLAB link
- The MATLAB version is 2019a (the installer refers to it as MATLAB 9.6).
- The CAE portal page instructions are (along with my updated comments):
 - Download the license file: [license.dat](#)
 - Download the 15.2GB installation package for Windows: [R2019a-Win.7z](#)
(there are separate packages for Windows, Linux, and MACs)
 - Extract the installer (do not extract to a USB drive as it takes too long)
 - Edit the installer control file `installer_input.txt` to specify the installation path as `c:/software/MATLAB/R2019a`
 - Install MATLAB from a Windows Command prompt by running the command
`setup.exe -inputFile installer_input.txt`
 - The installer will prompt for the license file.
 - The installer runs for about 30 minutes.
- MATLAB does not require a license environment variable.

6.14 MATLAB Support Packages

export_fig

The package `export_fig` is useful for generating PDF and PNG files from MATLAB. The figures generated by this package look a lot nicer than the figures that MATLAB natively generates (the README.md for `export_fig` describes many of the issues).

- Install Ghostscript (eg., the 64-bit version); <https://www.ghostscript.com>
Install to `cd c:/software/ghostscript`. The first time you run `export_fig` a dialog box will prompt you to point to this installation location (select the versioned folder within the install directory).
- Clone the git repo https://github.com/altmany/export_fig. The repo can be cloned into the MATLAB installation directory, eg.,

```
$ cd c:/software/MATLAB
$ git clone https://github.com/altmany/export_fig
```

- Add the repo path to your MATLAB `startup.m` script. The MATLAB command `userpath` returns the directory name where MATLAB searches for the startup file. Under Windows, the path to the startup file was:

```
c:/Users/dhawkins/Documents/MATLAB/startup.m
```

The startup script should look something like:

```
if (~isdeployed)
    addpath('c:/software/MATLAB/export_fig');
end
```

- Start MATLAB, create a simple plot, and generate a PDF and PNG

```
x = 1:10;
plot(x,x,'b')
hold on
plot(x,1-x,'r')
xlabel('x-axis')
ylabel('y-axis')
title('Title')
export_fig fig1.pdf -transparent
export_fig fig1.png -m4 -transparent
```

The `-transparent` option changes the grey background to transparent. The `-m4` option increases the number of pixels in the PNG.

6.15 Latex (MiKTeX)

I use Latex (MiKTeX) for documentation.

Installation procedure:

- Web site: <https://miktex.org>
- Download the latest 64-bit version: [basic-miktex-2.9.7269-x64.exe](#)
- Install to `c:\software\MiKTeX`
- Allow MiKTeX to check for updates and install any it finds.
- Process several of my Latex documents and install any missing packages.

6.16 Inkscape

I use Inkscape for drawing block diagrams, timing diagrams, and state machines.

Installation procedure:

- Web site: <https://inkscape.org>
- Download the latest 64-bit version (v0.92.4): [inkscape-0.92.4-x64.exe](#)
- Install to `c:\software\inkscape`

6.17 IrfanView

I use IrfanView (or sometimes Microsoft Paint) for manipulating and cropping screen captures.

Installation procedure:

- Web site: <https://www.irfanview.com>
- Download the latest 64-bit version (v4.54): `iview454_x64_setup.exe`
- Install to `c:\software\irfanview`

6.18 VirtualBox

I use VirtualBox for running Windows and Linux virtual machines.

Installation procedure:

- Web site: <https://www.virtualbox.org/wiki/Downloads>
- Download and install the latest version (6.1.0)
- Also install the extensions pack (the link is on the download page).
- Install to `c:\software\Oracle\VirtualBox`

6.19 Git LFS

Git LFS support can be added to Cygwin git by installing the Git LFS package.

Installation procedure:

- Web site: <https://git-lfs.github.com>
- Download file: [git-lfs-windows-v2.9.1.exe](#)
- Install to `c:\software\git-lfs`

6.20 Anaconda (Python)

- Install Anaconda

Download and install the *Individual Edition* (open source distribution) from

<https://www.anaconda.com/products/individual>

Select Python 3.7, 64-bit graphical installer (466MB). I ran the installer for all users (as administrator).

Install into `c:/software/Anaconda3`.

After installation, the Spyder GUI indicated that an update was available, so I updated the Anaconda installation.

- Update Anaconda.

Start the conda console and issue the command

```
conda update anaconda
```

This will update the installation, including the Spyder version.

Note: when I ran this command under Windows 10 the console had a warning about handles that could not be deleted and a reboot was recommended. I rebooted and re-ran the conda command, and received the message

```
# All requested packages already installed.
```

- Install `netcdf4`

Per the installation procedure at <https://anaconda.org/anaconda/netcdf4>, start the conda console and issue the command

```
conda install -c anaconda netcdf4
```

- Import anaconda environment

git clone the waveform analysis repo. In the anaconda page select the Environments tab. Then select import. A dialog window called 'Import new environment' will open. Select the file `base_env.yml` in the waveform analysis repo and give the environment an appropriate name (eg `zcu102_adrv9009_env`). Click okay and wait for the environment to be imported. Since `iio` and `adi` are not tracked by conda, we'll use pip to install them in the new environment.

- Install `libiio` (for `zcu102/adrv9009` setup)

Per the installation procedure at the bottom of the page at <https://github.com/analogdevicesinc/libiio>, download the primary installer package for Windows and run the executable. On my machine, the executable installed `libiio.dll` in `C:\Windows\System32` and `iio.py` in `C:\software\Anaconda3\pkgs\libiio-0.21-py38he3d0fc9_0\Lib\site-packages`. Now that the bindings have been installed in the terminal type:

```
pip install pylibiio
```

This should put `iio.py` in the same directory as the other modules: `C:\software\Anaconda3\lib\site-packages\iio.py`. Check that the module installed successfully by opening python and entering:

```
import iio
```

- Install `pyadi-iio`

Per the instructions at <https://wiki.analog.com/resources/tools-software/linux-software/pyadi-iio> in Cygwin or in the python anaconda terminal (administrator mode)

```
cd C:/github
git clone https://github.com/analogdevicesinc/pyadi-iio.git in C:/github/pyadi-iio.
cd pyadi-iio
python setup.py install
```

Check that `pyadi-iio` was installed correctly. In a python terminal enter

```
import adi
```

- Install `peakdetect`

In a python terminal enter:

```
pip install peakdetect
```

Check that the module was installed correctly by entering

```
import peakdetect
```

- Ethernet Adapter settings

To setup the ethernet adapter settings for the `zcu102/adrv9009` setup, connect the ethernet and USB cables from the `zcu102` to your laptop. Open a serial terminal to the OS on the `zcu102`. On my computer, there exists `COM3`, `COM4`, `COM5`, and `COM6`. The correct serial port is `COM4`. When asked for username and password enter `root` and `analog`. Then change the ethernet IP address to `192.168.1.21`, eg.

```
ifconfig eth0 down
ifconfig eth0 192.168.1.21 netmask 255.255.255.0
```

Now put the laptop on the same subnet as the `zcu102` OS. Go to Control Panel\Network and Internet\Network Connections\Ethernet (On my machine there's only one ethernet port). Double click Ethernet and highlight Internet Protocol Version (TCP/IPv4). Select Properties and a new dialog window called Internet Protocol Version 4 (TCP/IPv4) Properties should open. Select Use the following IP address and enter

IP address: 192.168.1.10
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.1.1

Click OK, to put the laptop on the same subnet as the zcu102.

6.21 Synopsys Synplify Pro and Premier

My Synopsys SolvNet account had expired, so I could not download the latest tools. I submitted a service request to re-enable SolvNet access.

7 VirtualBox Virtual Machines

7.1 Development Tools Virtual Drive (/software)

7.2 Centos

7.3 Ubuntu

7.4 Xilinx Vivado 2019.2

7.5 Xilinx PetaLinux 2019.2

8 Full Ubuntu Installation on 64GB USB Drive

- Download Ubuntu Live ISO.
- Burn to USB using Rufus.
- Boot and install the full version onto USB.

Redo this, but download the ISO and boot a VM, and from there I should be able to install a full version onto USB directory.

Create a list of tools to install;

- gparted
- readline
- vim, gvim (gvim-gtk3)
- Kernel headers for device driver building
- Wireshark