

Installation Guide for Quartus on Ubuntu Linux

James Bassett

December 2007

Version 1.0 December 2007

Preface

This guide is for intended for those wishing to install Quartus, Modelsim and Megacore IP on an Ubuntu system and is based upon the Quartus for Ubuntu available at the Nios Wiki (<http://nioswiki.jot.com/>) which can be found [here](#).

It has been tested using Quartus 7.2, Modelsim 6.1g and Megacore IP 7.2 for Linux on an Ubuntu 7.10 system, which, at time of writing are the latest versions available although it may work for other versions and even other Linux flavours.

Install process

Before proceeding ensure that you have *tcs*h installed as Altera's install scripts are written for csh, using specific extensions only found in *tcs*h.

Next, you will need to download the installer tars from Altera's website and extract them.

Quartus

The first piece of software to be installed will be Quartus. which can be achieved by running the *install* script as root from a console. This is located in the *quartus/linux/* folder where the tar was extracted to.

The path chosen for this guide is */opt/altera7.2/quartus7.2* and if you choose another path you will need to make amend the path setup phase with the different path.

Megacore IP

Similar to quartus, run the *install* script as root from console. This is located in the *ip/linux/* folder where the tar was extracted to.

The path used by this guide is */opt/altera7.2/megacore*

Modelsim

Under the *modelsim_ae* directory edit the *install.ms* script by adding `no-same-permissions` to lines 172 and 174 so that it appears as shown below. This will clear the sticky bit when extracting the other archives which would otherwise cause `uncompress` (which calls `gzip`) to exit without extracting these other archives, preventing the installation from completing.

Line 172:

```
tar xfo {$script_path}/modeltech_altera_unix.tar --no-same-permissions
```

Line 174:

```
tar xfo {$script_path}/modeltech_altera_unix.tar $file1.Z $file2.Z $file3.Z --no-same-permissions
```

Next run *install.ms* as root from a console - */opt/altera7.2/modelsim* is the install directory used by this guide.

Path setup

The following extract will need to be inserted into a suitable location, which will set the paths to the Quartus/Modelsim binaries and various variables used by those binaries.

Before doing so however the extract below will need to be amended with any changes to the path mentioned in the installation steps. Also a server or file location for the *LM_LICENSE_FILE* variable, this will typically either be the absolute path to a *license.dat* file or in the format of `<port>@<host>` for the license server (should no port be specified in the server's *license.dat* then simply use `@<host>`). For more details on licensing please see the Quartus Linux installation manual (http://altera.com/literature/manual/quartus_unix.pdf).

Once complete the extract should be placed into the *.bashrc* file in your home folder, or */etc/bash.bashrc* should you wish for multiple users to use Quartus.

```
## Altera Quartus environment variables
LM_LICENSE_FILE=<ENTER SERVER LOCATION>
QUARTUS_64BIT=0
MWOS=linux
MWARCH=i86
QUARTUS_MWMM=allwm
QUARTUS_ROOTDIR=/opt/altera7.2/quartus7.2
PATH=$PATH:$HOME/bin:/opt/altera7.2/quartus7.2/bin:/opt/altera7.2/quartus7.2/linux
PATH=$PATH:/opt/altera7.2/modelsim/linuxaloem
LD_LIBRARY_PATH=/lib:/usr/lib:/opt/altera7.2/quartus7.2/linux
unset GCC_EXEC_PREFIX
export LM_LICENSE_FILE QUARTUS_ROOTDIR
```

Once this has been completed you can try to run Quartus and Modelsim using the commands *quartus* and *vsim* respectively.

One potential issue that may arise is that the use of the configuration menu may cause a crash when running under Ubuntu due to an incompatibility. This incompatibility appears to stem from the use of the Window managing toolkit used to port Quartus, at time of writing this issue has not been resolved.

USB Blaster setup

First, try the udev method by adding the following to */etc/udev/rules.d/40-permissions.rules* :

```
# Altera USB-Blaster
BUS=="usb", SYSFS{idVendor}=="09fb", SYSFS{idProduct}=="6001", MODE="0666", SYMLINK+="usbblaster"
```

and in your home folder create an empty file named *.jtag.conf* and plug in the USB Blaster cable. Now, when you run *jtagconfig* it should inform you that it has found your USB Blaster. However, should *jtagconfig* inform you that no JTAG devices are found when the USB Blaster is plugged in you should try the second method over leaf.

Should the SYMLINK created by udev not be picked up, then you may try this method:

```
sudo mount -t usbfs /dev/bus/usb/ /proc/bus/usb/
killall jtagd
sudo <quartus install path>/bin/jtagd
jtagconfig
```

Steps two and three are necessary as in this case there is nothing to set the permissions of the device to user writable, thus jtagd will need root permissions to lock the USB Blaster.

Should this method be successful and you wish to make it permanent then

sudoedit /etc/init.d/mountdevsubfs.sh and uncomment the code on lines 42 45 so that it appears as follows:

```
#
# Magic to make /proc/bus/usb work
#
mkdir -p /dev/bus/usb/.usbfs
domount usbfs "" /dev/bus/usb/.usbfs -obusmode=0700,devmode=0600,listmode=0644
ln -s .usbfs/devices /dev/bus/usb/devices
mount --rbind /dev/bus/usb /proc/bus/usb
```

this will ensure that usb devices are available via */proc/bus/usb* allowing jtagd to detect the USB Blaster.

To ensure that jtagd is run as root upon startup an init.d script is provided in Appendix A, this script also provides functionality to start, stop and restart jtagd.

Should you wish to utilise this, copy the script from the appendix into */etc/init.d/jtagd* taking note to update the directory where the empty *.jtag.conf* is located.

Then to install the service simply:

```
sudo chmod 755 /etc/init.d/jtagd
sudo update-rc.d jtagd defaults 20
```

Troubleshooting

One error you may encounter is Quartus exiting simply saying `Aborted`, in this case check if there is an IPv4 loopback entry in `/etc/hosts` such as:

```
127.0.0.1 <machine's hostname> localhost.localdomain localhost
```

For additional troubleshooting check the official Altera support site (<http://altera.com/support>) in particular for patches. Other locations you may wish to try are the Altera forums (<http://www.alteraforum.com/>) or the NIOS forums (<http://www.niosforum.com/>).

Appendix A jtagd init.d file

NB: You will need to enter the path to your own home folder below.

```
#!/bin/bash
#
# Init file for Altera JTAGD
#
# chkconfig: 2345 60 25
# description: JTAG daemon
#
# processname: jtagd
# config: /etc/jtagd

# source function library
#. /etc/rc.d/init.d/functions
# Rough hack to port it for ubuntu as /etc/rc.d/init.d/functions is for RedHat
. /lib/lsb/init-functions

RETVAL=0
prog="jtagd"

PID_FILE=/var/run/jtagd.pid
JTAGD=/opt/altera7.2/quartus7.2/bin/jtagd
CONFIG=<CHANGE TO YOUR HOME FOLDER>.jtag.conf

start()
{
    log_daemon_msg "Starting jtagd" "jtagd"
    if start-stop-daemon --start --quiet --oknodo --pidfile $PID_FILE --exec $JTAGD -- --config $CONFIG;
then
    log_end_msg 0
    #start-stop-daemon isn't putting correct PID via --make-pidfile option so perform manually
    pidofproc $prog > $PID_FILE
else
    pidofproc $prog > $PID_FILE
    log_end_msg 1
fi
}

stop()
{
    log_daemon_msg "Stopping jtagd" "jtagd"
    if start-stop-daemon --stop --quiet --oknodo --pidfile $PID_FILE; then
        log_end_msg 0
    else
        log_end_msg 1
    fi
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        start
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart}"
        RETVAL=1
esac
exit $RETVAL
```