`

Rev0-2020.1-Petalinux Project Setup Guide

Author(s): R. Payne

Release: 1.0

Release Date: 21/08/2020

Document Type: Guide

Project: NuPRISM

Keywords: Vivado, Petalinux, NuPRISM, Firmware, Guide, 2020.1

History of Changes

Release No.	Date	Description	Authour(s)
1.0	21/08/2020	Initial Release	R. Payne

Introduction

Vivado and Vivado Project Setup

1) If not installed, download Vivado 2020.1 from https://www.xilinx.com/support/download.html
    a. The only chip required is the ZynqMPSoC. To limit the download size, it is recommended that only that one is downloaded
2) Clone the project located here: https://edev-group.triumf.ca/fw/exp/nuprism/xu1/rev0-2020.1-petalinux
3) Open the project using Vivado 2020.1
4) Run synthesis
5) If you want to set up the Integrated Logic Analyzer (ILA)
    a. Open synthesized design
    b. Navigate to set up a debug and select the signals you would like to view
6) Run implementation
7) Generate Bitstream
8) Go to File -> Export -> Export Hardware
    a. Select Fixed platform type
    b. Select Include Bitstream
    c. Export to default border
    d. Finish

Hardware Setup

Required Hardware:

- Mainboard
- Enclustra XU1
- Mainboard Power Cable
- Xilinx JTAG Debugger
- Power Supply
- Signal Generator
- BNC to Mainboard Cable
- USB-to-Serial Converter
- Ethernet Cable

1) Ensure the XU1 is properly in place on the mainboard
2) Plug the JTAG Debugger into the Xilinx JTAG port of the mainboard
3) Connect your USB-to-Serial Converter to the TX, RX, and GND of the J13 connector
4) Connect the two BNC-to-Mainboard Cables to and of J1-4, J6-8, J14-20
5) Use your DC power supply to deliver 12V to the mainboard

Insert pic of setup

Petalinux Setup

1) Install Ubuntu 18.04.04.
    a. If using a VM allocate 100GB of memory and 8GB of RAM.

2) To download all necessary packages, run:

```
sudo apt-get install -y iproute2 gcc g++ net-tools libncurses5-dev zlib1g:i386 libssl-dev flex bison libselinux1 xterm autoconf libtool texinfo zlib1g-dev gcc-multilib build-essential screen pax gawk python3 python3-pexpect python3-pip python3-git python3-jinja2 xz-utils debianutils iputils-ping libegl1-mesa libsdl1.2-dev pylint3 cpio vim
```

3) PetaLinux tools require that your host system /bin/sh is 'bash'. If you are using Ubuntu distribution and your /bin/sh is 'dash', consult your system administrator to change your default system shell /bin/sh with the sudo dpkg-reconfigure dash command.

4) Download the PetaLinux Tools - Installer - 2020.1 located here: https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html

5) Run: mkdir -p ~/pkg/petalinux/pl2020-1

6) In the download location, run: ./petalinux-v2020.1-final-installer.run --dir ../pkg/petalinux/pl2020-1

7) Alter ~/.bashrc to append: source ~/pkg/petalinux/pl2020-1/settings.sh

8) Close your terminal and re-open it to see if the tools have installed properly.

9) Run the following commands:
    a. mkdir ~/project
    b. cd ~/project
    c. petalinux-create -t project -n nuprism --template zynqMP
    d. cd ~/project/nuprism
    e. petalinux-config --get-hw-description <path-to-xsa-exported-from-vivado>
        i. Exit the menu that opens, to read more about this see the Petalinux 2020.1 guide
    f. petalinux-build
        i. This should generate files inside ~/project/nuprism/images/linux/
    g. petalinux-package --boot --u-boot --fpga images/linux/system.bit --format BIN

10) Copy BOOT.BIN, image.ub, and boot.scr to the root partition of your SD card
    a. To format your SD card, complete the tutorial in Appendix H of the Petalinux 2020.1 user guide: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug1144-petalinux-tools-reference-guide.pdf

11) Extract rootfs.tar.gz to the boot partition of you SD card using: tar xvf rootfs.tar.gz

12) Place the SD card into the NuPRISM board, open up a serial connection, and power on the board

MIDAS Host Setup

1) Alter ~/.bashrc to append:

```
export GIT_EDITOR="emacs -nw"
export MIDASSYS=$HOME/packages/midas
export MIDAS_EXPTAB=$HOME/online/exptab
export MIDAS_EXPT_NAME=e777
export PATH=$PATH:$MIDASSYS/bin
```

2) Run
    a. mkdir $HOME/packages
    b. cd $HOME/packages
    c. git clone https://bitbucket.org/tmidas/midas --recursive
    d. cd midas
    e. mkdir build
    f. cd build
    g. sudo cmake ../
    h. make install
    i. cd $HOME/packages
    j. git clone https://bitbucket.org/tmidas/rootana
    k. cd rootana
    l. git checkout tags/rootana-2020-03-a
    m. make
    n. cd $HOME/packages
    o. git clone https://github.com/linev/jsroot.git
    p. mkdir $HOME/online
    q. cd $HOME/online

3) Alter ~/.bashrc to append:

ROOTANASYS=$HOME/packages/rootana
JSROOTSYS=$HOME/packages/jsroot

4) Create a file in ~/online/ called exptab with the following contents
    a. e777 /home/<account>/online <account>
5) Run the following 3 programs:
    a. odbedit
    b. mhttpd
    c. mlogger
6) You should now be able to open up a browser and navigate to http://localhost:8080 (or whichever port number is provided by mhttpd) and access the MIDAS frontend panel.

Adding MIDAS to Petalinux and Running the MIDAS Demo

1) In the Petalinux project run:
    a. petalinux-create -t apps --template c++ --name midas --enable
    b. petalinux-create -t apps --template c++ --name mfe –enable
2) Copy the contents of the ds_midas folder into the relative projects generated in project-spec/meta-user/recipe-apps/
3) Run:
    a. petalinux-build -c midas
    b. petalinux-build -c mfe
4) Build the Petalinux project and copy the files over to your SD card as described in the Petalinux setup section (steps 10f,g – 12)
5) On you host run
    a. odbedit
    b. mhttpd
    c. mlogger
    d. mserver
6) Once the firmware has booted, run (on the mainboard):
    a. mfe -h <host-ip-address>:<mserver-address> -e e777
        i. example: mfe -h 192.168.1.80:1175 -e e777
        ii. you may need to add the mainboards ip address to a list of trusted ip addresses in the MIDAS control panel
7) Start a run by pressing the Start button on the frontend panel start page
8) Stop the run and check to see that data has been acquired