Building HSR DANH on PetaLinux with ZedBoard using FMC-GbE-RJ45

Version 0 Revision 2

May 16, 2019 (May 1, 2019)

Future Design Systems
www.future-ds.com / contact@future-ds.com

Abstract

This manual describes details of building HSR DANH on PetaLinux with Avnet ZedBoard using Future Design Systems FMC-GbE-RJ45 board.

Table of Contents

1 HSR DANH

As shown in Figure 1, HSR (High-availability Seamless Redundancy) DANH (Dual Attached Node with HSR) supports two HSL Links and is built using MAC[3] and HSR[4].

- Link-A port: it is one of two HSR Ethernet ports
- Link-B port: it is one of two HSR Ethernet ports

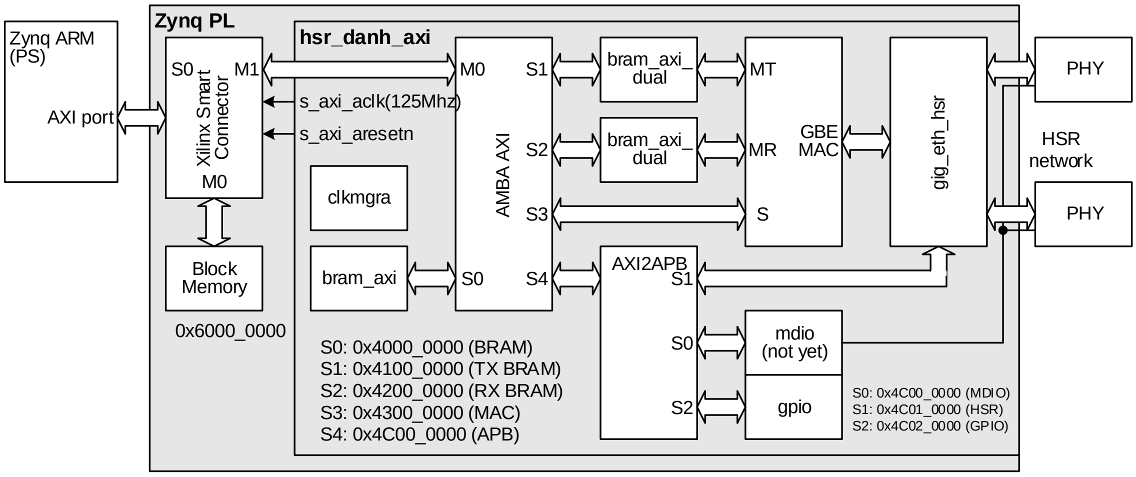


Figure 1: HSR DANH structure

ARM processor in the PS region can access blocks in the PL region through following addresses.

- S0: 0x4000_0000 (BRAM)
- S1: 0x4100_0000 (TX BRAM)
- S2: 0x4200_0000 (RX BRAM)
- S3: 0x4300_0000 (MAC)
- S4: 0x4C00_0000 (APB)
- P0: 0x4C00_0000 (MDIO)
- P1: 0x4C01_0000 (HSR)
- P2: 0x4C02_0000 (GPIO)

HSR DAMH is a kind of Ethernet MAC supporting dual-Ethenrt ports for HSR feature, where a processor send an Ethernet packet and it is duplicated and sent through two HSR port after adding HSR header as shown in Figure 2. HSR packet received from the HSR Link-A/B is forwarded to the upstream and the other HSR Link depending on MAC address and other information. More detaails should be referred to HSR specification[1].

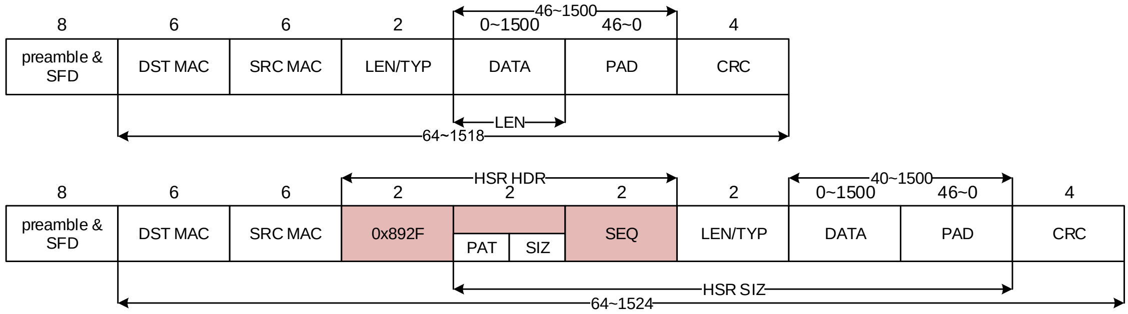


Figure 2: Normal packet and HSR packet

This document describes a details of building DANH using Future Design Systems HSR RTL[3] on the following hardware borads.

- Future Design Systems, FMC-GbE-RJ45[5]
- Avnet, ZedBoard[6]
- Xilinx, PetaLinux[7]

2 HSR DANH on ZedBoard

Figure 3 shows how HSR DANH is implemented on ZedBoard.

- Zedboard: FPGA boarad to implement HSR design
- FMC-GbE-RJ45: 3 port Ethernet board

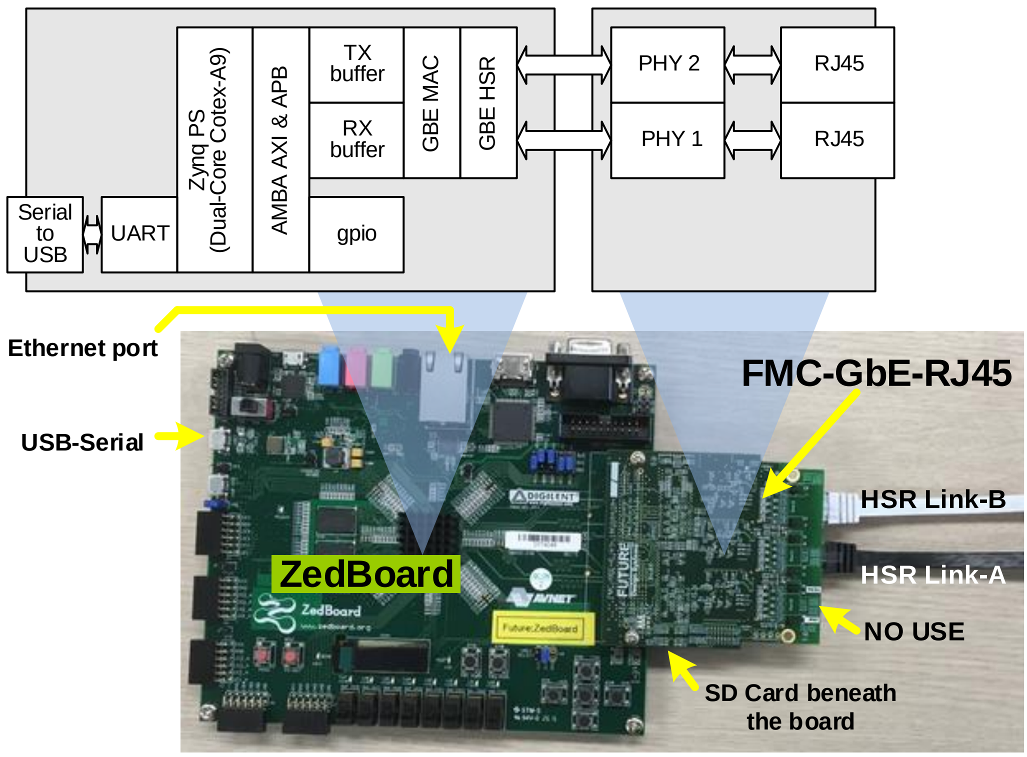


Figure 3: HSR DANH implementation on ZedBoard

2.1 Directory structure

'hw' directory contains hardware building project. 'sw.arm' directory contains software building project. 'bootgen' directory contains project to prepare bitstream that contains both hardware and software.

directory			remarks
petalinux	PetaLinux building projects		
	Makefile	It build 'zed-plnx' and calls 'make' in the 'zed-fsbl'.	
	zed-fsbl	FSBL[1] building project. This project prepares boot loader.	
		Makefile	
	zed-plnx	Test-bench	
sw.arm.petalinux	Software building projects		
	hello	'Hello world' project	
		Makefile	
	eth_send_receive	HSR DANH testing program	
		Makefile	
NOTE: This project uses HW platform prepared in the 'hsr.danh.arm' project[8].			

2.2 Testing structure

Figure 4 shows hardware structure to simulate a HSR system, where several HSR nodes are connected to build ring and each HSR node consists of 'hsr_danh_axi' and 'tester'.

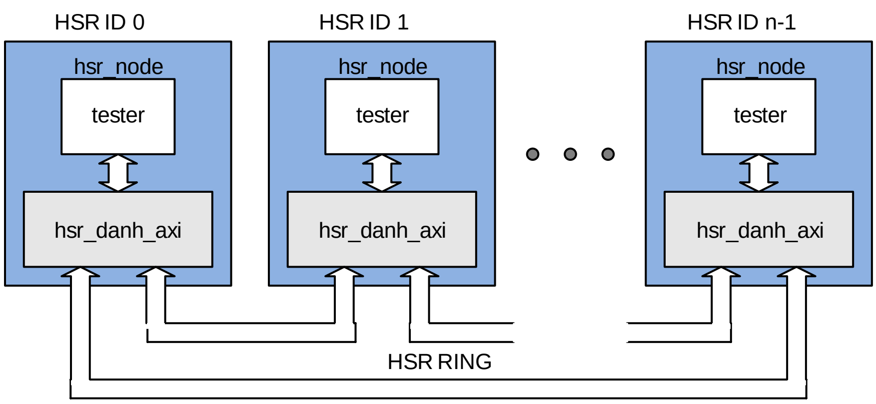


Figure 4: Testing structure

[1] FSBL: First Stage Boot Loader

HSR nodes are connected through GMII instead of PHY and each 'tester' in the HSR node is assigned a unique MAC address related to node id[2] such that '48'hF0_12_34_56_78_{node_id[7:0]}'.

2.3 Simulation

Simulation can be carried out in 'hsr.danh.arm' project[8].

3 Building system

3.1 Preparing bit files for HW

HW platform can be built in 'hsr.danh.arm' project[8].

Simply run 'RunAll.sh' with 'hw' option as follows.

```
$ cd …/hsr.danh.arm
$ ./RunAll.sh –step hw
```

At last, following files should be ready as a result of building HW platform.

- hw/impl/zedboard.lpc/zed_bd_wrapper.bit
- hw/impl/zedboard.lpc/zed_bd_wrapper_sysdef.hdf

3.2 Preparing PetaLinux image

This step requires following packages.

- PetaLinux SDK
- ZedBoard BSP

3.2.1 Get BSP

Download "ZED BSP" from https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html.

- avnet-digilent-zedboard-v2018.3-final.bsp

3.2.2 Install PetaLinux SDK

First download installer and install it.

1. Download installer "PetaLinux 2018.3 Installer" from https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html

[2] Board ID[7:0] is read from board sliding switch through GPIO. Refer to Figure 5.

- petalinux-v2018.3-final-installer.run

2. Move the installer file to a user director, say ${HOME}/tmp and run it with installation directory as argument. (You may need to change the file permission to executable.)

   - $ ./petalinux-v2018.3-final-installer.run ${HOME}/work/PetaLinux
   - While installing PetaLinux SDK, error can raise due to pre-requisites program. Install the pre-requisites and then re-run the installer.

3. Make 'bsp' directory in the installation directory, i.e., ${HOME}/work/PetaLinux/bsp, and mv the BSP file into that.

   - $ cp ~/Download/avnet-digilent-zedboard-v2018.3-final.bsp ${HOME}/work/PetaLinux/bsp

3.2.3 Building PetaLinux

Now building PetaLinux.

Simply run make.

- $ cd …/hsr.danh.petalinux/petalinux
- $ source ${HOME}/work/projects/PetaLinux/settings.sh
- $ source /opt/Xilinx/SDK/2018.3/settings64.sh
- $ make

While running make, there are two steps of 'petalinux-config' you should set configure as follows.

For the first 'petalinux-config'
[For HDF]
- Yocto Settings→Enable Network sstate feeds: disable[3]
- Image Packaging Configuration→Copy final Iimages to ftpboot: disable

Configuring takes some time and then the second 'petalinux-config'

[For ROOTFS]
- Filesystem Packages→misc→gcc-runtime→{libstdc++, libstdc++-dev}: enable
- Filesystem Packages→console→network→dropbear→dropbear: enable
- Filesystem Packages→console→network→openssh→openssh-sftp-server: enable

'petalinux-build' is following this Root File System configuration.

---

[3] Press space-key to toggle enabled or disabled

At last following two files should be ready.

- zed-plnx/images/linux/BOOT.BIN: Boot loader and FPGA bit-stream
- zed-plnx/images/linux/image.ub: Linux kernel image

3.3 Booting from SD Card

ZedBoard can be configured using SD Card. The boot configuration DIP switches should be set properly; consult manual for details.

1. Prepare ZedBoard with FMC-GbE-RJ45
   - Port 1 for HSR Link-A
   - Port 2 for HSR Link-B
   - For this DANH, Upstream link is not used (i.e., port 0)
2. Connect USB-to-Serial and invoke terminal emulator
3. Connect Ethernet
4. Go to 'petalinux' directory and copy following two files to SD Card.
   - zed-plnx/images/linux/BOOT.BIN: Boot loader and FPGA bit-stream
   - zed-plnx/images/linux/image.ub: Linux kernel image
5. Insert SD Card to the ZedBoard
   - Make sure ZedBoard is turned off
   - Make sure Boothing mode selection switches are properly set as shwon in Figure 5.
6. Turn on ZedBoard
   - Configuration DONE LED should be on.
   - Two LEDs should be on as shown in Figure 5; one for HSR ready and the other for DANH indicator.

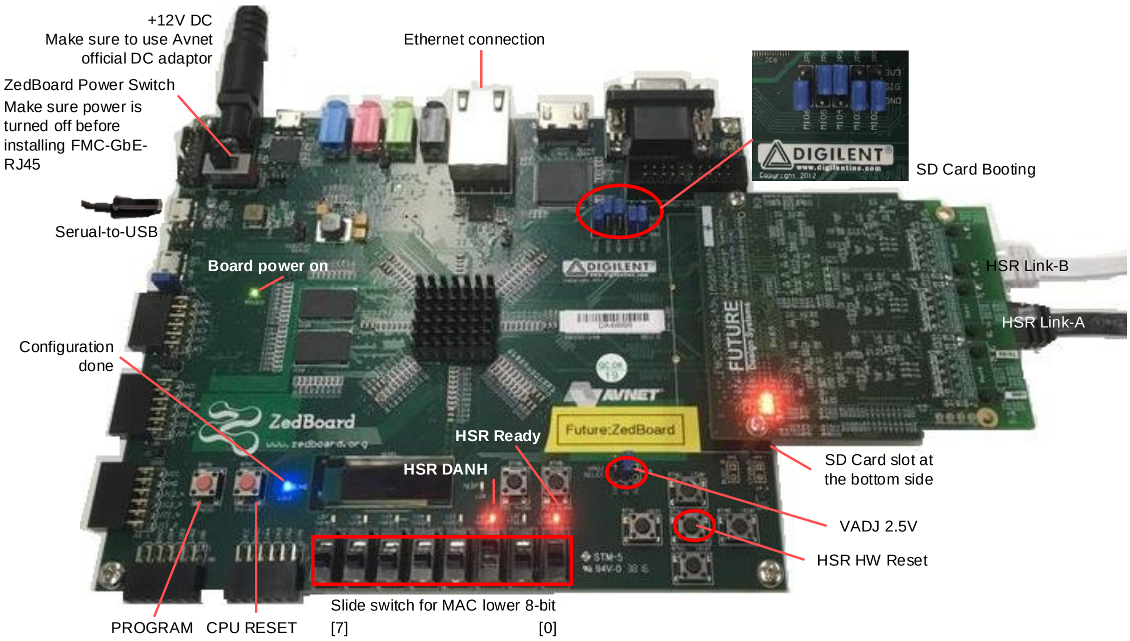


Figure 5: ZedBoard and FMC-GbE-RJ45

Push button switches:
- HSR HW Reset: It raises reset for PL and does not affect PS.
- CPU Reset: It raises reset for PS and makes restart PetaLinux.
- PROGRM: It makes FPGA program.

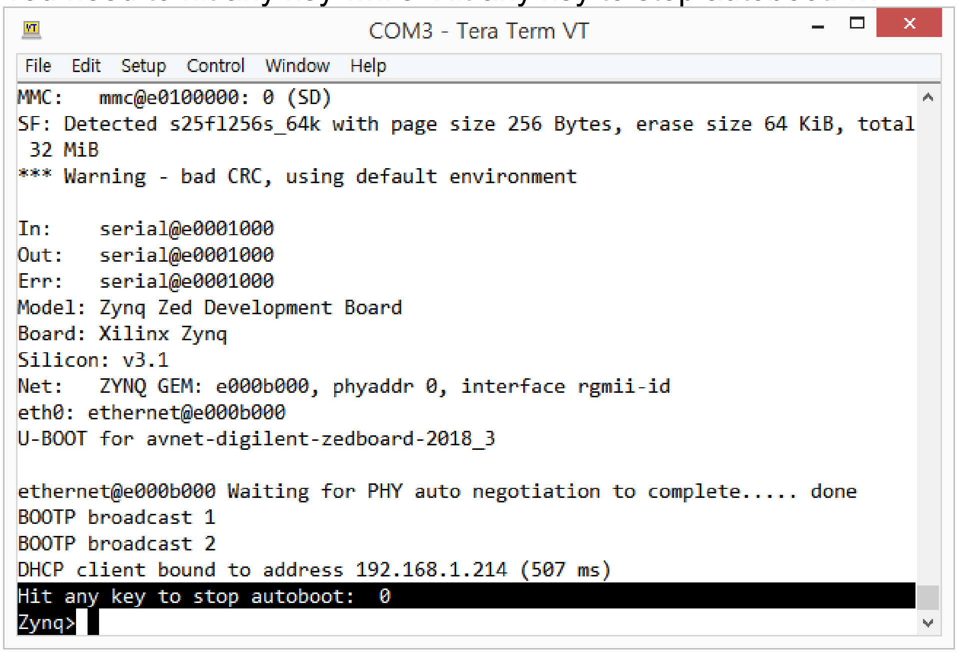
3.4 MAC addresses

As shown in Figure 5, the system requires two MAC addresses.

- PetaLinux uses Ethernet to communicate host computer, which is used to compile and download application programs.
  - ✧ Each PetaLinux should be set a unique MAC HW address when there are more than one PetaLinux since PetaLinux uses static MAC HW address.
- MAC for HSR uses a unique MAC HW address
  - ✧ It is automatically determined by reading 8-bit Board ID switch.

In order to set user specific MAC HW address, do as follows.

- Boot PetaLinux as described in Section 3.3.
  - ✧ (This step may not require.) For some case following steps are also required in the PentaLinux
    - $ rm /etc/udev/rules.d/70-persistent-net.rules
    - $ sync
    - $ reboot
- While booting, stop before starting Linux. Now it is uBoot command line.
  - ✧ You need to hit any key while 'Hit any key to stop autoboot: …'.

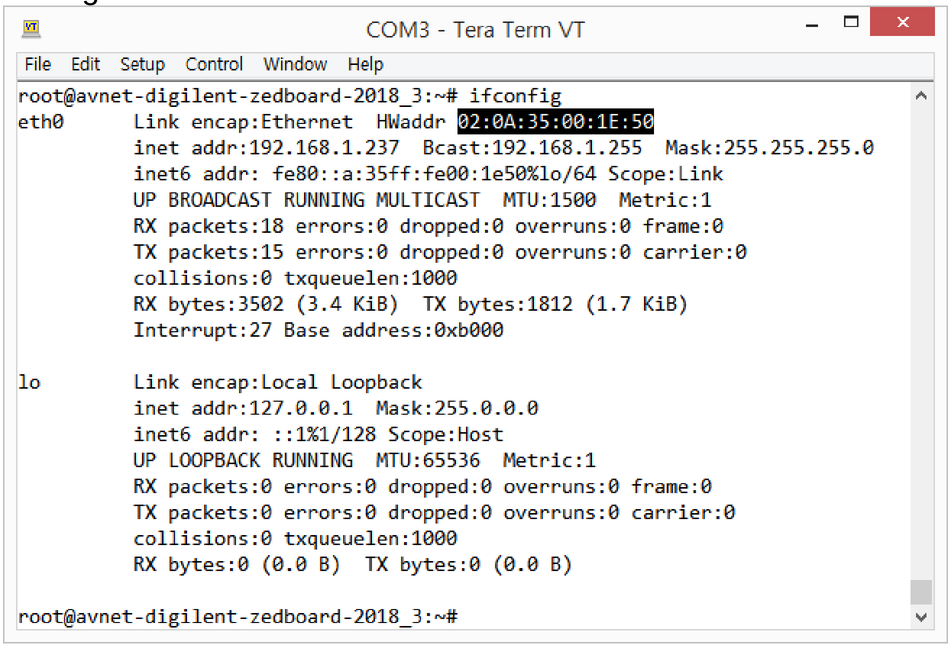


- Set MAC HW address something like

- ✧ (Note that this step update flash memory. It means MAC HW address updated permanently)
- ✧ (Make sure that the second bit of the first octet should be 1, which indicates locally managed address)
- ✧ Zynq> setenv ethaddr 02:0A:35:00:1E:50[4]
- ✧ Zynq> saveenv

  Saving Environment to SPI Flash...
  SF: Detected s25fl256s_64k with page size 256 Bytes, erase size 64 KiB, total 32 MiB
  Erasing SPI flash...Writing to SPI flash...done
- ✧ Zynq> reset (NOT 'boot')

- Now PetaLinux brings up
- Petalinux login will be "root" and its passwd will be "root".
- Check MAC HW address as follows.
  - ✧ $ ifconfig



4 Testing with two DANHs

4.1 Setup

When all has been done as shown in Figure 5, a HSR system shown in Figure 6 can be set up using two DANH. In addition to this, USB-to-Serial connections are required to interact with ARM processor. Make sure that FMC-GbE-RJ45 port 1 and port 2 are connected (do not use port 0).

[4] It is recommended to use 'locally administered' one and it should start '02:…'.

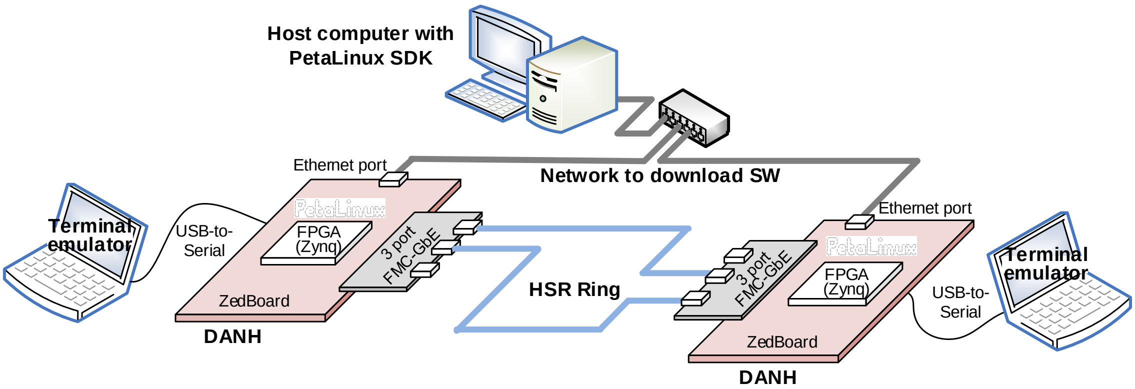


Figure 6: Two DANH conceptual

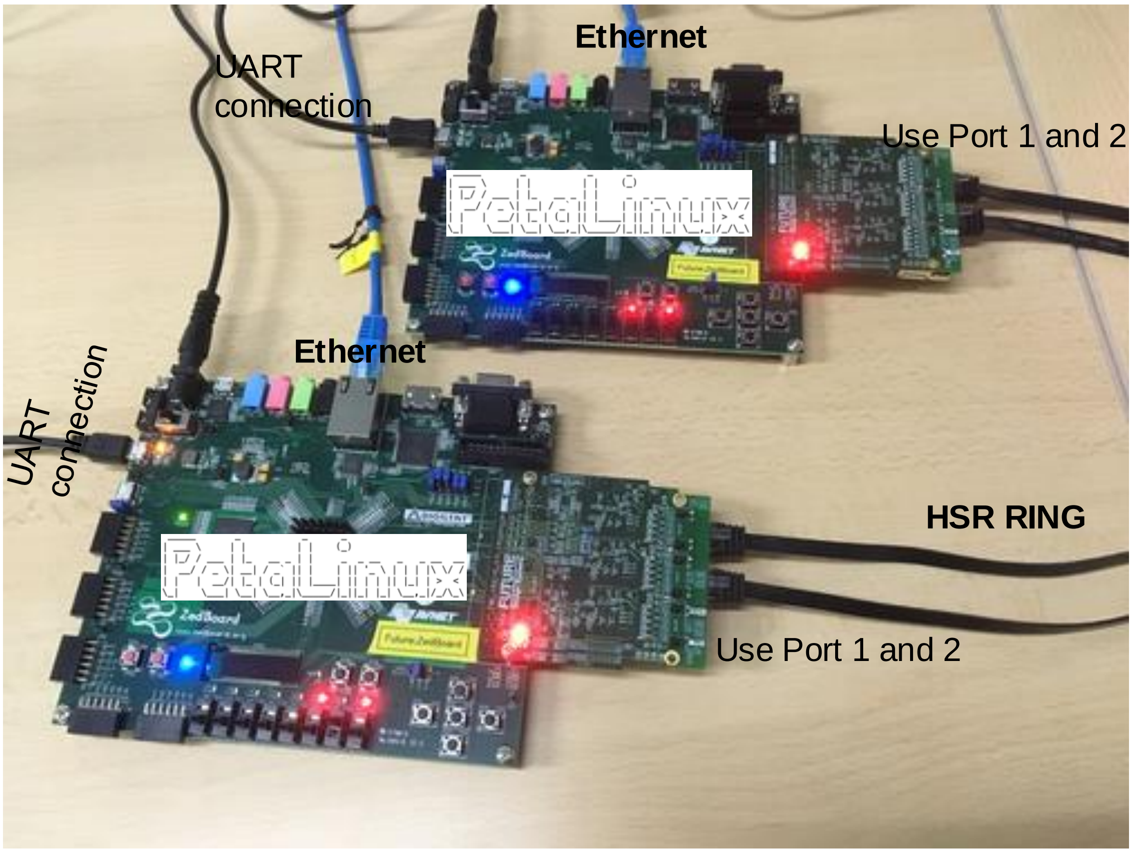


Figure 7: Two DANH with two ZedBoard

The serial port used following setings.

- baud rate: 115,200
- bit-width: 8-bit
- parity: no parity
- stop bit: 1-bit

One of following terminal emulator can be used.

- For Windows
  - ✧ Teraterm
- For Linux[5]
  - ✧ Call up (use '~.' to terminate)
    - $ cu -l /dev/ttyACM0 -s 115200
  - ✧ Picocom[6]
    - $ sudo picocom -b 115200 /dev/ttyACM[7]0
  - ✧ GTKterm
    - $ gtkterm -s 115200 -p /dev/ttyACM1
      - You may need add yourself to 'dialout' group
      - $ sudo addgroup ${USER} dialout
    - You may need to kill processes accessing the device (e.g., the device is '/dev/ttyACM0')
      - $ fuser –k /dev/ttyACM0

4.2 Prepare application program and download the program

It is carried on at the host computer, where PetaLinux SDK is installed.

Go to $PROJECT/hsr.danh.petalinux/sw.arm.petalinux/eth_send_receive directory

- $ cd … sw.arm.petalinux/eth_send_receive

Set PetaLinux environment, e.g., PetaLinux SDK is installed in ${HOME}/work/

- $ ${HOME}/work/PetaLinux/settings.sh

Run make

- $ make

Now download ARM executable to the ZedBoard using scp (secure copy command) through Ethernet. The program should be download to each ZedBoard.

- $ scp eth_send_receive root@192.168.1.235:/home/root

In which, '192.168.1.235' should be the correct IP address of each ZedBoard and it can be checked using 'ifconfig' command.

---

[5] Check which tty port by '$ dmesg | grep --i tty' or '$ sudo setserial -g /dev/ttyACM[0123456789]'
[6] Use 'Ctl-AX" to exit from picocom.
[7] ACM stands for a USB communication device (CDC) of sub-type "abstract control model'.

If there is any error regarding SSH, do again after removeing SSH key something like as follows.

- $ ssh-keygen –f “${HONE}/.ssh/known_hosts” –R 192.168.1.235

4.3 Running

It is carried on at the PetaLinux computer, where HSR design is running, i.e., ZedBoard.

Following shows a typical case of testing with two ZedBoard and each row shows user interfaction with FDS-Monitor.

Terminal A	Terminal B	Remarks
$ ./eth_send_receive	$ ./eth_send_receive	
monitor> **mac_init**	monitor> **mac_init**	initialize
monitor> **mac_addr -r** MAC 0x021234567801 HSR 0x021234567801	monitor> **mac_addr -r** MAC 0x021234567802 HSR 0x021234567802	check mac and hsr address. MAC and HSR should be the same.
monitor> **pkt_snd -b *0x0213456788902***		send a packet to the other
	monitor> **pkt_rcv -v 3** ETH mac dst: 0x02123456782 ETH mac src: 0x021234567801 ETH type leng: 0x0001 [60] 02:12:....	receive a packet
monitor> **pkt_snd -b 0x021234567801 -r**	monitor> **pkt_rcv -v 3 -r**	let receive packets send and receive packets One sends packets continuously. The other receives packets. Try to dis-connect network cable in order to check HSR recovery.

References

[1] IEC 62439-3, Industrial communication networks –High availability automation networks –Part 3: Parallel Redundancy Protocol (PRP) and :q

[2] High-availability Seamless Redundancy (HSR), Edition 2.0, 2012-07.
[3] Future Design Systems, Gigabit Ethernet Media Access Controller, FDS-TD-2018-10-001, 2018. (company confidential)
[4] Future Design Systems, High-availability Seamless Redundancy Controller on Gigabit Ethernet, TD-2018-10-002, 2018. (company confidential)
[5] Future Design Systems, FMC-GbEGbE -RJ45 User Manual- Avnet ZedBoard, TD-2018-10-004, 2018.
[6] Avnet, ZedBoard (Zynq Evaluation and Development) Hardware User's Guide, Jan. 2014.
[7] Xilinx, PetaLinux Tools Documentation Reference Guide, UG1144.
[8] Future Design Systems, Building HSR DANH on ZedBoard using FMC-GbE-RJ45, FDS-TD-2019-01-003.

Revision history

- ☐ 2019.05.01: Document started by Ando Ki (adki@future-ds.com)

– End of document –