Intel® Acceleration Stack User Guide

Intel FPGA Programmable Acceleration Card N3000-N/2

Updated for Intel® Acceleration Stack for Intel® Xeon® CPU with FPGAs: **1.3.1**

Online Version
Send Feedback

UG-20284

ID: 683362
Version: 2021.11.01

Contents

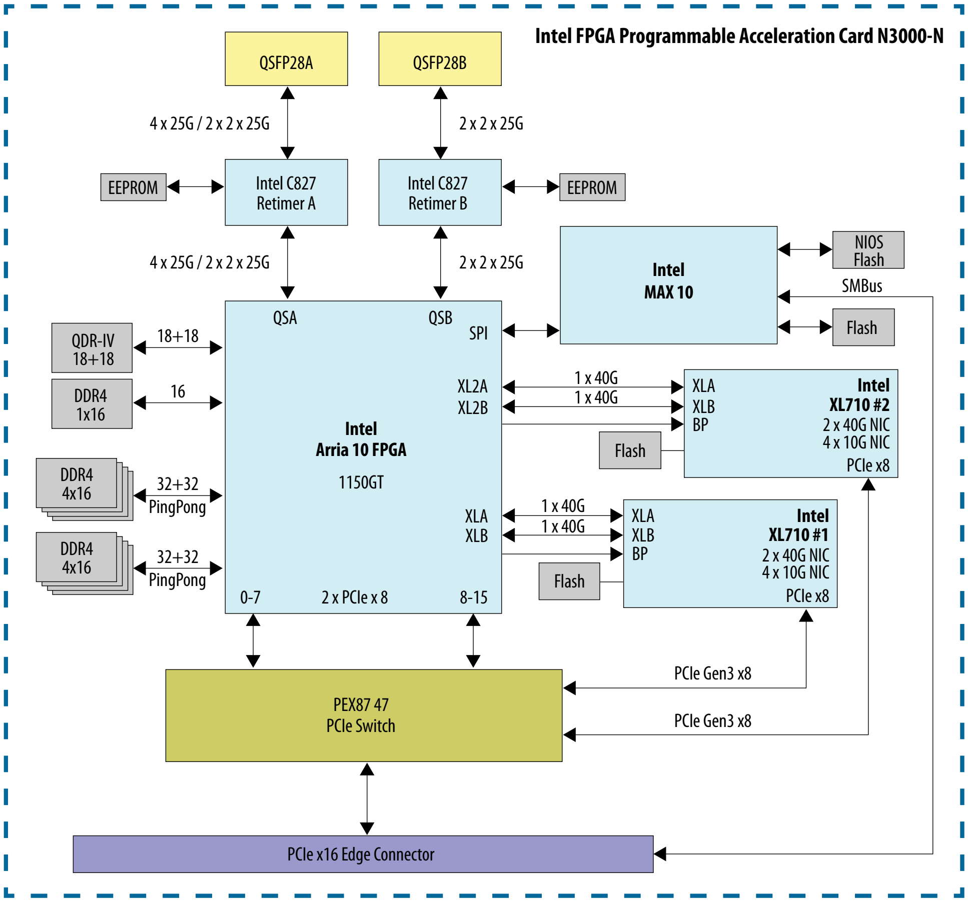
1. About this Document

This document provides:

- Instructions and requirements for installing the Intel FPGA Programmable Acceleration Card N3000-N/2 (BD-NVV-N3000-3 or BD-NVV-N3000-2) into a server.
- Instructions for installing the Open Programmable Acceleration Engine (OPAE) software on host Intel® Xeon® processor for managing and accessing the Intel FPGA PAC N3000-N/2.
- Instructions for installing the OPAE tools for validation of the Intel FPGA PAC N3000-N/2.
- Instructions for running built-in self-tests using the FPGA factory image.
- Instructions for installing the Intel XL710 driver and network testing tools for testing the Ethernet capabilities.
- Information about Graceful Shutdown and Single Event Upset (SEU) handling.

The Intel Acceleration Stack for Intel Xeon CPU with FPGAs is a collection of software, firmware, and tools that allows both software and RTL developers to take advantage of the power of Intel FPGA PAC N3000-N/2. By offloading computationally intensive networking tasks to the FPGA, the acceleration platform allows the Intel Xeon processor to execute other critical processing tasks.

Figure 1. Block Diagram



1.1. Acronym List

Acronym	Expansion	Description
Intel FPGA PAC	Intel FPGA Programmable Acceleration Card	Intel FPGA PAC N3000-N/2 is a full-duplex 100 Gbps in-system re-programmable acceleration card for multi-workload networking application acceleration.
AFU	Accelerator Functional Unit	Hardware Accelerator implemented in FPGA logic which offloads a computational operation for an application from the CPU to improve performance.
AF	Acceleration Function	Compiled Hardware Accelerator image implemented in FPGA logic that accelerates an application.
API	Application Programming Interface	A set of subroutine definitions, protocols, and tools for building software applications.
FIU	FPGA Interface Unit	FIU is a platform interface layer that acts as a bridge between platform interfaces like PCIe* and AFU-side interfaces such as CCI-P.

continued...

Acronym	Expansion	Description
OPAE	Open Programmable Acceleration Engine	The OPAE is a set of drivers, utilities, and API's for managing and accessing AFs.
FME	FPGA Management Engine	The FME provides information about the FPGA platform including the OPAE version.
RSU	Remote System Update	An RSU operation sends an instruction to the device to trigger a power cycle of the Intel FPGA PAC N3000-N/2 only. This will force reconfiguration from flash for either Intel MAX® 10 BMC image (on devices that support it) or the FPGA image.

2. System Requirements

- Operating Systems (OS):
  - CentOS Linux version 7.6 kernel 4.19 with real time kernel support.
  - Red Hat Enterprise Linux (RHEL) version 8.2 kernel 4.18 with real time kernel support.

  For information about the latest OS supported, refer to the Getting Started page.
- Connectivity hardware for testing the Ethernet interfaces:
  - 25 Gbps QSFP28

While selecting a server, ensure that the server meets the following requirements (per slot):

- Power
- Cooling air flow
- Physical dimension

For more information, refer to the Intel FPGA Programmable Acceleration Card N3000-N Data Sheet or Intel FPGA Programmable Acceleration Card N3000 Data Sheet.

Note: Intel FPGA PAC N3000-N (BD-NVV-N3000-3) has improved thermal characteristics as compared to the Intel FPGA PAC N3000 variant (BD-NVV-N3000-2), but both have the same dimensions.

Note: To compile an AFU using the Intel Quartus® Prime Pro Edition software, your server must have at least 48 GB of system RAM.

3. Hardware Installation

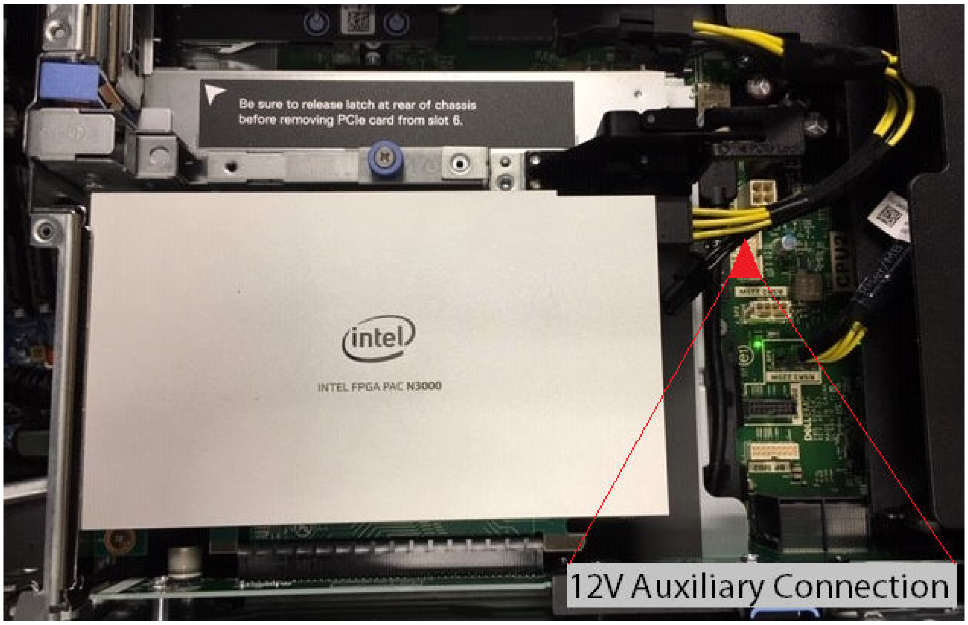
To operate the Intel FPGA PAC N3000-N/2 in your server, you must have the following:

- PCI Express* Gen3x16 slot with physical space for a full height half length PCIe form factor board
- Auxiliary 12 V 6-pin power connector
- Server that provides sufficient airflow for a given air inlet temperature

Note: The Intel FPGA PAC N3000-N/2 cannot operate without the 6-pin auxiliary power connector. Internal board circuitry prevents operation without the auxiliary power source and PCIe connector power source.

Note: If the Intel FPGA PAC N3000-N/2 is not integrated into a server closed loop fan control system, you must set the fan speed to 100%. The fan speed (100%) setting must be applied to avoid overheating when the server turns on. When the BMC detects that the card has overheated, it powers down major circuitry of the card to prevent damage.

Figure 2. Typical Intel FPGA PAC N3000-N/2 Installation in a Server


Be sure to release latch at rear of chassis before removing PCIe card from slot 6.
intel
INTEL FPGA PAC N3000
12V Auxiliary Connection

3.1. Installing the Intel FPGA PAC N3000-N/2

Complete these steps to install the Intel FPGA PAC N3000-N/2:

1. Power down the system.
2. Plug the Intel FPGA PAC N3000-N/2 into a PCIe x16 physical and x16 electrical slot on the motherboard.
3. Connect the auxiliary power to the 12 V 6-pin connector using an applicable cable.

   Note: Make sure the auxiliary power cable does not block airflow to the Intel FPGA PAC N3000-N/2.
4. Enable the following options in the BIOS:
   - Intel VT-x (Intel Virtualization Technology for IA-32 and Intel 64 Processors)
   - Intel VT-d (Intel Virtualization Technology for Directed I/O)
5. For network testing, you can insert a loopback module into each QSFP28 port.

   Note: ESD protection is required while handling the Intel FPGA PAC N3000-N/2.

   Warning: Take caution when you are inserting and removing the cards into PCIe slots. The bottom side of the card has capacitors and resistors that can be knocked off if the cards scrapes against edges or corners of the slot in the chassis.

Figure 3. Example of QSFP Non-optical Modules

This is a 25 GbE QSFP setup.

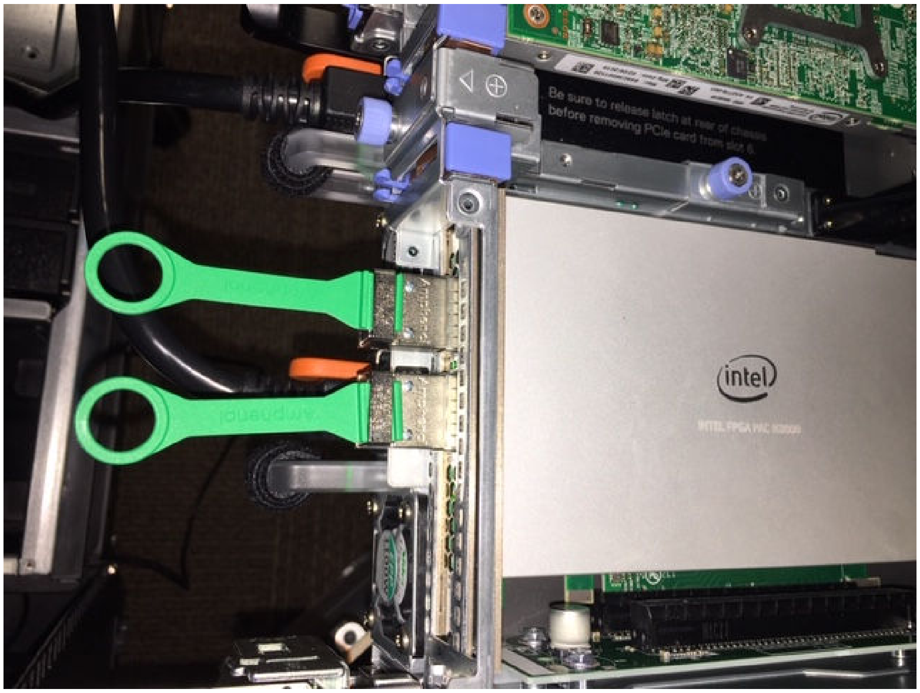
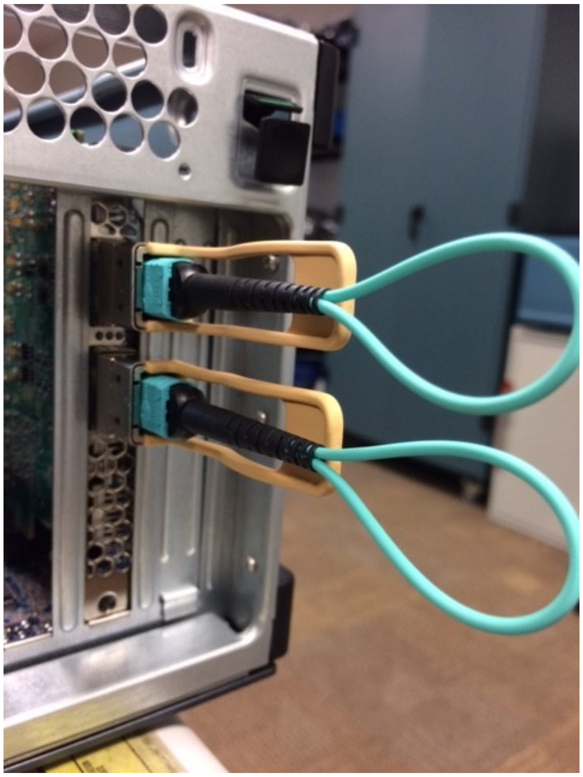

Be sure to release latch at rear of chassis before removing PCIe card from slot 6.
intel
INTEL FPGA PAC N3000
Amphenol

Figure 4. Example of QSFP Loopback Optical Modules

This figure shows the correct orientation of the QSFP module for optical loopback testing.



6. Power on the system and observe the Ethernet status LEDs which are located between the QSFP connectors. The LED operation is described in the table below:

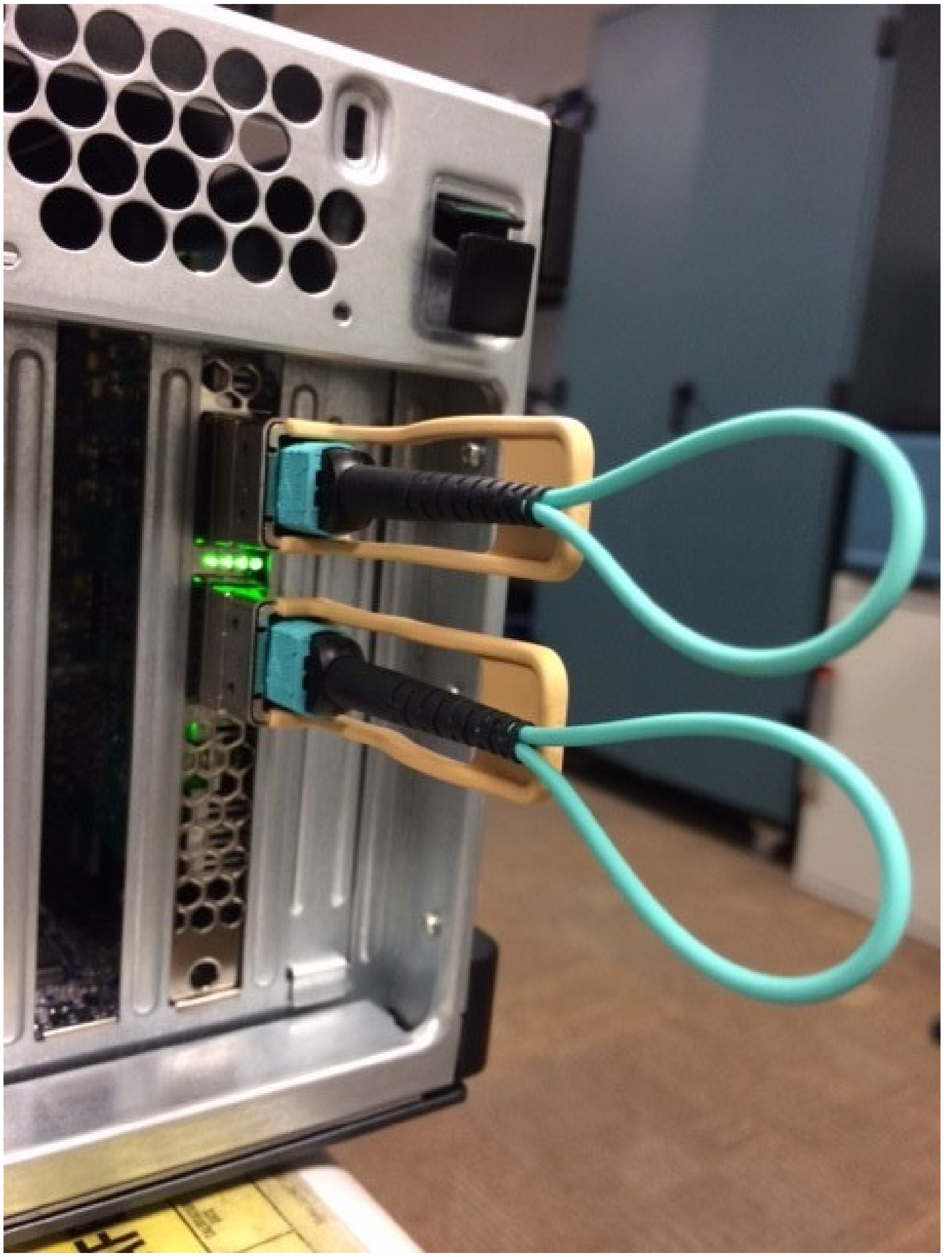
Table 1. LED Behavior

LED Type	Description
Connectivity LED	Green means link is up with link speed of 25G.
	Off means link is down.
Activity LED	Green blinking at 1 Hz means link activity present.
	Off means link is down or no activity.
All LEDs blinking yellow	It means either: • 12 V Auxiliary or PCIe edge supply voltage is below 10.46 V or • FPGA core temperature reaches 100°C or • Board temperature reaches 100°C You should check the following: • Card insertion in PCIe slot. • 12 V Auxiliary connection on the Intel FPGA PAC N3000-N/2 and on the server motherboard. • Fan setting for cooling air flow. Sufficient airflow is required whenever the Intel FPGA PAC N3000-N/2 is powered on.

For details on the location of Connectivity and Activity LEDs, refer to the Intel FPGA Programmable Acceleration Card N3000-N Data Sheet or Intel FPGA Programmable Acceleration Card N3000 Data Sheet.

Figure 5. Ethernet Status LEDs Power-On Indication

This figure is an example of 25G configuration where both Ethernet links are UP with ongoing Ethernet activity.



7. I/O Memory Management Unit (IOMMU) support is not enabled by default in CentOS Linux 7.0 distribution. IOMMU support is required for a virtual function (VF) to function properly when assigned to a virtual machine (VM).

    a. Edit the `/etc/default/grub` and add `iommu=pt iommu=on default_hugepagesz=1G hugepagesz=1G hugepages=2 hugepagesz=2M hugepages=200` to the `GRUB_CMDLINE_LINUX` entry.

For example:

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=centos/root
rd.lvm.lv=centos/\
swap rhgb quiet pci=realloc intel_iommu=pt intel_iommu=on
default_hugepagesz=1G hugepagesz=1G hugepages=2 hugepagesz=2M
hugepages=200"
```

b. **GRUB** reads its configuration either from `/boot/grub2/grub.cfg` file on traditional BIOS-based machines or from `/boot/efi/EFI/redhat/grub.cfg` file on UEFI machines. Depending on your system, execute one of the following instructions as root:

- For BIOS based machine:

  ```
  # grub2-mkconfig -o /boot/grub2/grub.cfg
  ```

- For UEFI based machine:

  ```
  # grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
  ```

```
Generating grub configuration file ...
Found linux image: vmlinuz-4.19.106-rt45
Found initrd image: initramfs-4.19.106-rt45
 Found linux image: /boot/vmlinuz-0-
rescue-594cabaaf9a84c6ea0a5167c89ad916d Found initrd image: /boot/
initramfs-0- rescue-594cabaaf9a84c6ea0a5167c89ad916d.img
```

c. Reboot your server to apply the new **GRUB** configuration file.

d. To verify the **GRUB** update, run the following command:

```
$ cat /proc/cmdline
```

Sample output showing `intel_iommu=on` on the kernel command line.:

```
BOOT_IMAGE=/vmlinuz-4.19.106-rt45 root=/dev/mapper/centos-root ro
default_hugepagesz=1G hugepagesz=1G hugepages=2 hugepagesz=2M
hugepages=200 nosoftlockup mce=ignore_ce audit=0
isolcpus=1-11,24-35,13-23,36-47 nohz_full=1-11,24-35,13-23,36-47
rcu_nocbs=1-11,24-35,13-23,36-47 pci=realloc intel_iommu=on iommu=pt
enforcing=0 crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/
swap rhgb quiet skew_tick=1
```

8. Set POSIX locale:

A locale is a set of environmental variables that defines the language, country, and character encoding settings for applications and shell session on a Linux system.

a. Run the following command to check if LC_ALL is set to UTF-8:

```
$ locale
```

Sample output:

```
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
```

```
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=
```

b. To set UTF-8 encoding, add the following in `~/.bash_profile`:

```
export LC_ALL=en_US.UTF-8
export LANG=en_US.UTF-8
export LANGUAGE=en_US.UTF-8
```

c. Restart the terminal.

Note: If your server is set up with secure boot, the OPAE Remote Setup (RSU) command will not function. This RSU limitation is due to secure boot having the following limitations:

- Using kexec to start an unsigned kernel image.
- Hibernation and resume from hibernation.
- User-space access to physical memory and I/O ports.
- Module parameters that allow setting memory and I/O port addresses.
- Writing to MSRs through /dev/cpu/*/msr.
- Use of custom ACPI methods and tables.
- ACPI APEI error injection.

While you implement secure boot in your server, you must power cycle your server to load a new FPGA image rather than using the RSU command.

4. Installing the OPAE Software

The OPAE is a software framework delivered as part of the Intel Acceleration Stack for managing and accessing the Intel FPGA PAC.

The following section describes the installation of OPAE on a freshly imaged server with supported OS and kernel. The host must have internet connectivity to retrieve additional software packages. The installation steps require `sudo` or `root` privileges on your host.

Note: The OPAE version created for Intel FPGA PAC N3000-N/2 is not compatible with any other Intel FPGA PAC.

Note: The OPAE version described in this section is only compatible with Intel FPGA PAC N3000-N/2.

4.1. Install the Release Package

Before installing the release package, ensure that the Intel FPGA PAC N3000-N/2 is installed properly as mentioned in the Hardware Installation on page 8.

Note: The Acceleration Stack 1.3.1 version only supports 25G Ethernet configuration.

The installers for Intel FPGA PAC N3000-N/2 provide easy installation of the release package. If you plan to perform FPGA development and compile FPGA RTL code, select the development (dev) installer. If you plan to develop or run server applications, select the runtime (rte) installer. Refer to the following table to understand differences between each installer:

Details	**Acceleration Stack for Runtime**	**Acceleration Stack for Development**
	runtime (rte) installer	**development (dev) installer[1]**
Purpose	Provides necessary environment to execute the AFUs as well as allows for software development of host application.	Provides necessary environment to execute the AFUs as well as allows for software development of host application. Additionally, it also includes development environment for Intel Arria® 10 GT FPGA.
OPAE Software Development Kit (SDK) Version	1.3.7-5	
Intel Quartus Prime Pro Edition	Not included or required	Included: Intel Quartus Prime Pro Edition software version 19.2 with IP licenses required to create a programmable Intel Arria 10 GT FPGA image.
Default installation location	`/home/<username>/intelrtestack`	`/home/<username>/inteldevstack`

[1] The dev installer file size is about 8.5 GB.

4.1.1. Remove Previous OPAE Packages

Remove any previous installation of OPAE or FPGA and Intel MAX 10 update package using the following:

```
$ sudo yum remove opae*
```

4.1.2. Install the Acceleration Stack for Runtime

Based on your server operating system (OS), select the appropriate runtime (rte) installer file:

Server OS	Acceleration Stack Runtime Installer	Follow Instructions
RHEL 8.2 kernel 4.18	n3000_ias_1_3_1_pv_rte_RHEL_installer.tar.gz	For RHEL on page 15
CentOS 7.6 kernel 4.19	n3000_ias_1_3_1_pv_rte_centos_installer.tar.gz	For CentOS on page 16

4.1.2.1. For RHEL

1. Unpack the runtime (rte) installer files for RHEL:

   ```
   $ tar xvfz n3000_ias_1_3_1_pv_rte_RHEL_installer.tar.gz
   ```

   ```
   $ cd n3000_ias_1_3_1_pv_rte_RHEL_installer
   ```

   It consists of:

   - n3000-1.3.8-3-rte-el8-setup.sh—OPAE installer script.
   - N3000_supplemental_files/find_RP.sh—This script finds the PCIe root port. It is used in AFU development for JTAG programming. For more information, refer to the Accelerator Functional Unit Developer Guide: Intel FPGA Programmable Acceleration Card N3000 Variants.
   - N3000_supplemental_files/hello_fpga.c—This is a simple example application.

2. The script automatically installs dependent packages. The following commands enable required repos for dependent package installation. Failure to enable these repos results in installation errors.

   Note: You must run the remaining steps in this section as root.

   ```
   # subscription-manager repos --enable codeready-builder-for-rhel-8-x86_64-
   rpms
   ```

   ```
   # dnf install https://dl.fedoraproject.org/pub/epel/epel-release-
   latest-8.noarch.rpm
   ```

3. Enable RHEL for Real Time (RT) repository and install Real Time group:

   ```
   #  subscription-manager repos --enable rhel-8-for-x86_64-rt-rpms
   ```

   ```
   # dnf --releasever=8.2 --disablerepo=epel* groupinstall RT
   ```

4. When the RHEL for RT kernel is installed, it is automatically set to be the default kernel and is used on the next boot. To verify that the RT kernel is the default kernel:

   ```
   # grubby --default-kernel
   ```

Sample output:

```
/boot/vmlinuz-4.18.0-193.14.3.rt13.67.el8_2.x86_64
```

The above output, `rt` indicates that the RT kernel is the default kernel. A reboot is required to ensure the change to RT Kernel.

5. Run the OPAE install script to install the OPAE driver, OPAE tools and any package dependencies:

```
# ./n3000-1.3.8-3-rte-el8-setup.sh -y --owner <user[:group]>
```

Sample output:

```
Running setup
Beginning installation
Processing group "OPAE Software"
Analyzing dependencies...
Installing OPAE Software dependencies
Analyzing packages to install...
    Installing OPAE Software packages...
        opae-intel-fpga-driver-2.0.1-10.x86_64.rpm
        opae.admin-1.0.3-2.el8.noarch.rpm
        opae-libs-1.3.7-5.el8.x86_64.rpm
        opae-tools-1.3.7-5.el8.x86_64.rpm
        opae-tools-extra-1.3.7-5.el8.x86_64.rpm
        opae-devel-1.3.7-5.el8.x86_64.rpm
Processing group "OPAE PACSign"
Analyzing dependencies...
Analyzing packages to install...
    Installing OPAE PACSign packages...
        opae.pac_sign-1.0.4-3.el8.x86_64.rpm
Extracting opae-1.3.7-5.tar.gz
Extracting opae.admin-1.0.3.tar.gz
Extracting opae.pac_sign-1.0.4.tar.gz
Source /root/intelrtestack/bin/init_env.sh to setup your environment.
Changing ownership on /root/intelrtestack
Installation done
see /tmp/opae_install_2020-08-09_16_43_56.log for details
```

Tip:

- `--owner` allows you to change the ownership of installation directories to a specified user. Not listing owner results in use of default setting which is root.
- `-y` option is required because interactive operation where the user is asked for specific settings rather defaults is not supported.
- `-v` option provides verbose output and is useful for debugging install issues.
- `-h` option lists all script options.

4.1.2.2. For CentOS

You must have the CentOS 7.6 kernel 3.10 with workstation settings for these steps. This section describes how to:

1. Download kernel 4.19, real time patch and file preparation
2. Build and install kernel 4.19 with real time patch
3. Install OPAE in CentOS kernel 4.19 run time environment

4.1.2.2.1. Download kernel 4.19

1. Install the required Linux packages:

   ```
   $ sudo yum install -y ncurses-devel bison flex elfutils-libelf-devel
   openssl-devel patch epel-release
   ```

2. Unpack the runtime (rte) installer files for CentOS:

   ```
   $ tar xvfz n3000_ias_1_3_1_pv_rte_centos_installer.tar.gz
   ```

   ```
   $ cd n3000_ias_1_3_1_pv_rte_centos_installer
   ```

   It consists of:

   - `n3000-1.3.8-3-rte-el7-setup.sh`—OPAE installer script.
   - `config_4.19_opae`—This file configures the kernel build and install process.
   - `N3000_supplemental_files/find_RP.sh`—This script finds the PCIe root port.
   - `N3000_supplemental_files/hello_fpga.c`—This is a simple example application.

3. Download the kernel 4.19 source and real time patch from the following:

   - Source
   - Patch

4. Copy the downloaded kernel and patch to your `n3000_ias_1_3_1_pv_rte_centos_installer` directory:

   ```
   $ cp ~/Downloads/linux-4.19.106.tar.gz .
   ```

   ```
   $ cp  ~/Downloads/patch-4.19.106-rt45.patch.gz .
   ```

5. Unpack the Linux kernel:

   ```
   $ tar xvfz linux-4.19.106.tar.gz
   ```

   ```
   $ cd linux-4.19.106/
   ```

6. Apply the real time patch to the kernel source with the following command:

   ```
   $ zcat ../patch-4.19.106-rt45.patch.gz | patch -p1
   ```

4.1.2.2.2. Build and Install kernel 4.19

1. Copy the provided `config_4.19_opae` file to the kernel source directory and rename the file as `.config`:

   ```
   $ cp ../config_4.19_opae .config
   ```

2. Build the kernel source with real time patch. The makefile for the kernel source allows you to specify the number of cores/threads to use. The kernel source takes several minutes to build. Intel recommends to use the following command for the build which uses all the cores:

   ```
   $ make -j $(nproc)
   ```

   Tip: Substitute `$(nproc)` with 4 to use reduced server resources for the build.

3. Install the kernel modules and kernel:

   ```
   $ sudo make modules_install
   ```

   ```
   $ sudo make install
   ```

4. Update your kernel boot order to make kernel 4.19 as the default kernel:

   ```
   $ sudo grubby --set-default /boot/vmlinuz-4.19.106-rt45
   ```

   To verify that the default kernel has been set correctly:

   ```
   $ sudo grubby --default-kernel
   ```

   Sample output:

   ```
   /boot/vmlinuz-4.19.106-rt45
   ```

5. Reboot the server and verify that the kernel 4.19-rt45 is running after reboot:

   ```
   $ sudo reboot

   After server comes up:

   $ uname -mrs
   Linux 4.19.106-rt45 x86_64
   ```

4.1.2.2.3. Install OPAE in kernel 4.19

Run the OPAE install script to install the OPAE driver, OPAE tools and any package dependencies:

```
$ cd <Installer unpack directory>
```

```
$ sudo ./n3000-1.3.8-3-rte-el7-setup.sh -y --owner <user[:group]>
```

Sample output:

```
Running setup
Beginning installation
Processing group "OPAE Software"
Analyzing dependencies...
Installing OPAE Software dependencies
Analyzing packages to install...
    Installing OPAE Software packages...
        opae-intel-fpga-driver-2.0.1-10.x86_64.rpm
        opae.admin-1.0.3-2.el7.noarch.rpm
        opae-libs-1.3.7-5.el7.x86_64.rpm
        opae-tools-1.3.7-5.el7.x86_64.rpm
        opae-tools-extra-1.3.7-5.el7.x86_64.rpm
        opae-devel-1.3.7-5.el7.x86_64.rpm
Processing group "OPAE PACSign"
Analyzing dependencies...
Analyzing packages to install...
    Installing OPAE PACSign packages...
        opae.pac_sign-1.0.4-3.el7.x86_64.rpm
Extracting opae-1.3.7-5.tar.gz
Extracting opae.admin-1.0.3.tar.gz
Extracting opae.pac_sign-1.0.4.tar.gz
Source /root/intelrtestack/bin/init_env.sh to setup your environment.
Changing ownership on /root/intelrtestack
Installation done
```

Tip:

- `--owner` allows you to change the ownership of installation directories to a specified user. Not listing owner results in use of default setting which is root.
- `-y` option is required because interactive operation where the user is asked for specific settings rather defaults is not supported.
- `-v` option provides verbose output and is useful for debugging install issues.
- `-h` option lists all script options.

4.1.3. Install the Acceleration Stack for Development

The development (dev) installer for Acceleration Stack for Development installs the Intel Quartus Prime Pro Edition software version 19.2 on your server for FPGA development. The Intel Quartus Prime Pro Edition software version 19.2 does not run on the operating system RHEL 8.2.

Typically, FPGA development is performed on a machine dedicated for Intel Quartus Prime Pro Edition and FPGA simulation process, while the Intel FPGA PAC N3000-N/2 is installed in a separate server with the RTE install for application testing and deployment.

The instructions below show installation on a CentOS 7.6 sever. There are tag options in the installer that only allows installation of Intel Quartus Prime Pro Edition.

1. Download, extract, and change permission for the DEV installer:

   ```
   $ tar xvzf n3000_ias_1_3_1_pv_dev_centos_installer.tar.gz
   ```

   ```
   $ cd n3000_ias_1_3_1_pv_dev_centos_installer
   ```

2. Run the script:

   ```
   $ sudo ./n3000-1.3.8-4-dev-el7-setup.sh -y --owner <user[:group]>
   ```

   The `--owner` argument allows you to change the ownership of directories to a given user. Running interactively without the -y option is not supported. For example:

   ```
   $ sudo ./n3000-1.3.8-4-dev-el7-setup.sh -y --owner john:john
   ```

   The installation will take approximately 20 minutes to complete. The above command installs OPAE driver and tools as well as Intel Quartus Prime Pro Edition. If you want to only install the Intel Quartus Prime Pro Edition and the provided RTL source files, to run FPGA compiles on a separate server, then change the install command to:

   ```
   $ sudo ./n3000-1.3.8-3-dev-el7-setup.sh -y -t quartus source --owner
   <user[: group]>
   ```

   Sample output:

   ```
   Running setup
   Beginning installation
   Processing group "OPAE Software"
   Analyzing dependencies...
   Installing OPAE Software dependencies
   Analyzing packages to install...
       Installing OPAE Software packages...
           opae-intel-fpga-driver-2.0.1-10.x86_64.rpm
           opae.admin-1.0.3-2.el7.noarch.rpm
           opae-libs-1.3.7-5.el7.x86_64.rpm
           opae-tools-1.3.7-5.el7.x86_64.rpm
   ```

```
        opae-tools-extra-1.3.7-5.el7.x86_64.rpm
        opae-devel-1.3.7-5.el7.x86_64.rpm
Processing group "OPAE Software Development Dependencies"
Analyzing dependencies...
Installing OPAE Software Development Dependencies dependencies
nothing to install
Processing group "OPAE PACSign"
Analyzing dependencies...
Analyzing packages to install...
    Installing OPAE PACSign packages...
        opae.pac_sign-1.0.4-3.el7.x86_64.rpm
Extracting opae-1.3.7-5.tar.gz
Extracting opae.admin-1.0.3.tar.gz
Extracting opae.pac_sign-1.0.4.tar.gz
Extracting pac_n3000_rtl_v1.5.7.tar.gz
Installing main Quartus package: QuartusProSetup-19.2.0.57-linux.run
    Installing update: quartus-19.2-0.01vc-linux.run
```

3. Add the Intel Quartus Prime development tool to the path. This is required for AFU compilation:

```
$ source /home/<username>/inteldevstack/bin/init_env.sh
```

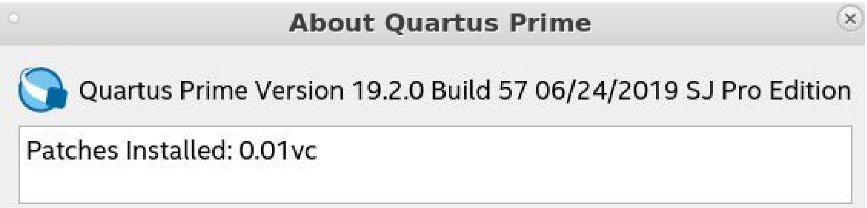
```
Adding /home/<username>/inteldevstack/intelFPGA_pro/quartus/bin to
PATH
Adding /home/<username>/inteldevstack/intelFPGA_pro/
nios2eds/bin/gnu/H-x86_64-pc-linux-gnu/bin to PATH
Adding /home/<username>/inteldevstack/bin to PATH
```

4. Verify Intel Quartus Prime installation by bringing up Intel Quartus Prime GUI and verifying the version and IP licenses:

```
$ quartus
```

   a. Click **Help** > **About Quartus Prime** to verify Intel Quartus Prime version:

Figure 6. Intel Quartus Prime Version



About Quartus Prime

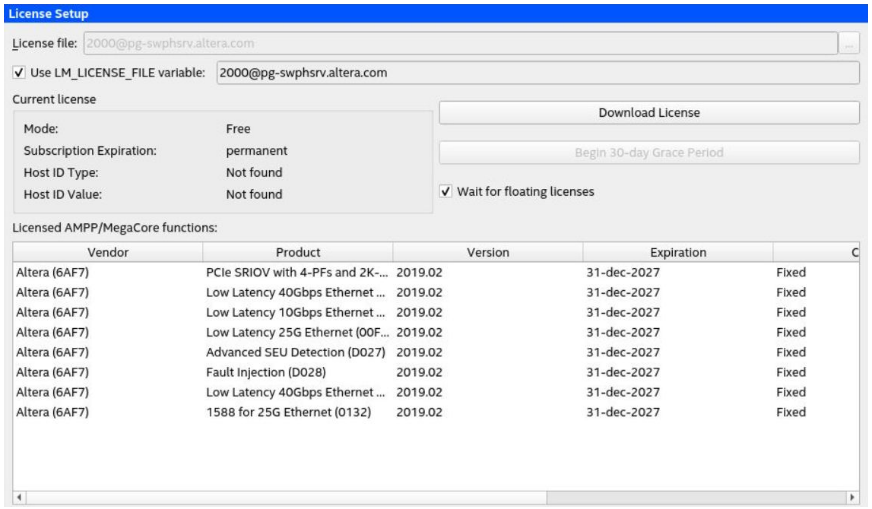
Quartus Prime Version 19.2.0 Build 57 06/24/2019 SJ Pro Edition

Patches Installed: 0.01vc

   If **patch .01vc** is not installed, then verify the execution of `init_env.sh` in the above step.

   b. Click **Tools** > **License Setup** to verify IP licenses.

Figure 7. IP Licenses


License Setup

License file: 2000@pg-swphsrv.altera.com

☑ Use LM_LICENSE_FILE variable: 2000@pg-swphsrv.altera.com

Current license

Mode: Free
Subscription Expiration: permanent
Host ID Type: Not found
Host ID Value: Not found

☑ Wait for floating licenses

Licensed AMPP/MegaCore functions:

Vendor	Product	Version	Expiration	C
Altera (6AF7)	PCIe SRIOV with 4-PFs and 2K-...	2019.02	31-dec-2027	Fixed
Altera (6AF7)	Low Latency 40Gbps Ethernet ...	2019.02	31-dec-2027	Fixed
Altera (6AF7)	Low Latency 10Gbps Ethernet ...	2019.02	31-dec-2027	Fixed
Altera (6AF7)	Low Latency 25G Ethernet (00F...	2019.02	31-dec-2027	Fixed
Altera (6AF7)	Advanced SEU Detection (D027)	2019.02	31-dec-2027	Fixed
Altera (6AF7)	Fault Injection (D028)	2019.02	31-dec-2027	Fixed
Altera (6AF7)	Low Latency 40Gbps Ethernet ...	2019.02	31-dec-2027	Fixed
Altera (6AF7)	1588 for 25G Ethernet (0132)	2019.02	31-dec-2027	Fixed

4.1.4. Verify the OPAE Installation

After executing the Acceleration Stack for Runtime installer, ensure that you have successfully installed OPAE.

- Verify the OPAE package installation:

```
$ rpm -qa | grep opae

opae.admin-1.0.3-2.el8.noarch
opae-devel-1.3.7-5.el8.x86_64
opae-intel-fpga-driver-2.0.1-10.x86_64
opae-tools-1.3.7-5.el8.x86_64
opae.pac_sign-1.0.4-3.el8.x86_64
opae-tools-extra-1.3.7-5.el8.x86_64
opae-libs-1.3.7-5.el8.x86_64
```

- Verify the OPAE driver installation:

```
$ lsmod | grep fpga

ifpga_sec_mgr           16384  1 intel_max10
intel_fpga_fme          81920  0
intel_fpga_afu          45056  0
fpga_mgr_mod            16384  1 intel_fpga_fme
intel_fpga_pci          32768  2 intel_fpga_fme,intel_fpga_afu2
```

```
$ lsmod | grep pac_n3000_net

pac_n3000_net           32768  1 c827_retimer
```

- Verify if the Linux has enumerated the Intel FPGA PAC N3000 FPGA Management Engine Device (FME):

```
$ lspci -d :0b30

61:00.0 Processing accelerators: Intel Corporation Device 0b30
```

The above output provides the PCIe BDF (Bus : Device : Function) value for the Intel FPGA PAC. Here, the `61:00.0` indicates the bus is 61, device is 00 and function is 0. The PCIe BDF value varies based on systems, therefore your values can be different. Record this value for future use.

If the verification of OPAE installation fails, refer to Troubleshooting on page 61.

5. Identify the Intel MAX 10 BMC Version

Run the following command and verify the output with the table below:

```
$sudo fpgainfo fme
```

Table 2. Intel MAX 10 BMC Versions

Intel FPGA PAC Variants	Intel MAX 10 NIOS Firmware (FW)	Intel MAX 10 Build
Intel FPGA PAC N3000-N (BD-NVV-N3000-3)	D.2.1.24	D.2.0.7
Intel FPGA PAC N3000 (BD-NVV-N3000-2)	D.2.0.19	D.2.0.6

Note: . You can identify your card by running `fpgainfo security` and comparing the BMC `root entry` hash field with the following:

Table 3. Identify your Intel FPGA PAC

Intel FPGA PAC	MMID (found on side cover of your Intel FPGA PAC)	BMC Root Entry Hash
Intel FPGA PAC N3000	999HGN (2x2x25G) 999H1K (8x10G)	0x757f524c2f45db58ac2a6c93e72b9167149979b795195d09d5e2efad82f2b031
Intel FPGA PAC N3000-N	999PJD	0xec0f42d3af138e3eca7141107f7fed5f7c13846fadbba884e51ad26bf36a3d21

5.1. FPGA Factory Image Overview

The Intel FPGA PAC N3000-N/2 has an on-board flash with two partitions (user and factory) for storing two FPGA image files known as user image and factory image. A new Intel FPGA PAC N3000-N/2 is provided with the 2x2x25G image in factory partition and 4x25G image in user partition.

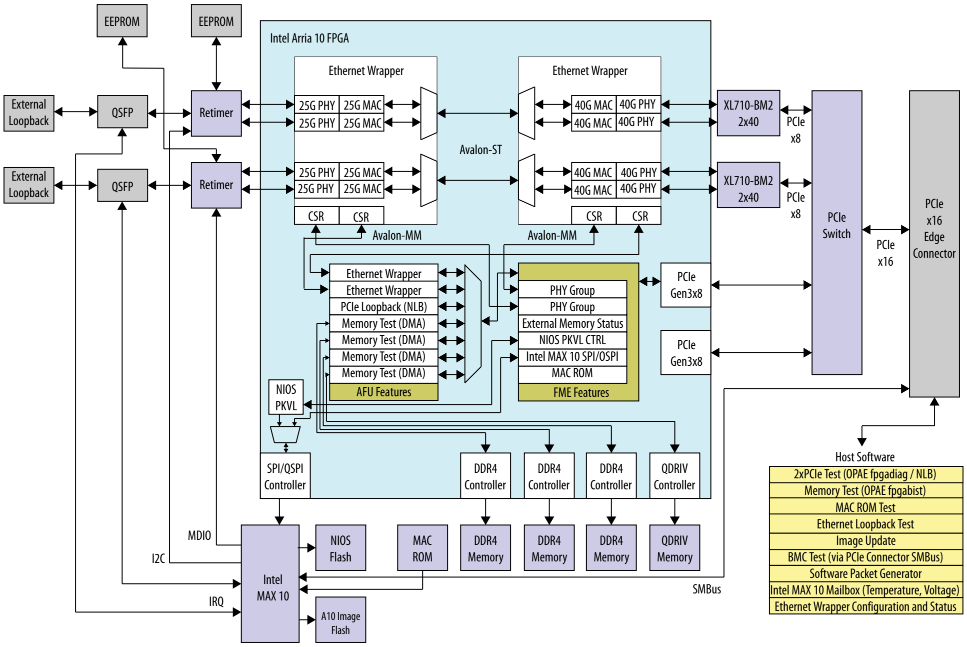
When the image in the user partition fails to load, the Intel FPGA PAC N3000-N/2 reverts back and boots from factory partition. This factory image loaded into the user partition provides basic functionality to demonstrate all the interfaces including Ethernet and external memory interfaces.

The typical use case is a specific workload (for example: FLEXRan, ipSEC, vBNG, etc.) is loaded into the user partition. When the Intel FPGA PAC N3000-N/2 is power cycled with a RSU command or a server power cycle, the user partition image is loaded into the Intel Arria 10 FPGA. The ability to load application specific images is a key benefit of the Intel FPGA PAC N3000-N/2.

The factory image includes the following Intellectual Property (IP) to support in the development of Accelerator Function (AF):

- The PCIe IP core
- The Core Cache Interface protocol (CCI-P) fabric
- DDR4 memory interface controller IP
- QDR4 memory interface controller IP
- 25 and 40 GbE physical interface and MACs with pass-through connectivity between Intel Ethernet Connection C827 Retimer and Intel Ethernet Controller XL710-BM2
- FPGA Management Engine (FME)
- Nios® core to configure the Intel Ethernet Connection C827 Retimers

Figure 8. Example: Factory Image for 2x2x25G



6. Intel XL710 Driver Installation and Firmware Update

The Intel XL710 device on the Intel FPGA PAC N3000-N/2 is widely used on server network interface controller (NIC) cards. Both the Intel FPGA PAC N3000-N/2 and server NIC cards use i40e and iavf software drivers for XL710 device.

If your server uses both Intel FPGA PAC N3000-N/2 and server XL710 NIC cards, you must ensure that the XL710 NVM firmware and drivers are compatible. Refer to the *Table: Software/NVM Compatibility for XXV710* in the Intel Ethernet Controller X710/XXV710/XL710 Feature Support Matrix for NVM releases 7.3 and later. For RHEL 8.2, use NVM 8.0 or later for OS compatibility.

For operation without a server XL710 NIC card, the Intel FPGA PAC N3000-N/2 has been tested with the following NVM and driver version:

Table 4. NVM and Driver Version

Operating System	NVM	i40e	iavf
CentOS 7.6	7.0	2.10.19.82	3.7.61.20
RHEL 8.2	7.0	2.12.6	3.9.5

The Intel FPGA PAC N3000-N/2 is shipped with NVM 7.0.

To identify the XL710 device ID on the Intel FPGA PAC N3000-N/2:

```
$ lspci -d :0d58
```

Sample output:

```
3d:00.0 Ethernet controller: Intel Corporation Device 0d58 (rev 02)
3d:00.1 Ethernet controller: Intel Corporation Device 0d58 (rev 02)
3f:00.0 Ethernet controller: Intel Corporation Device 0d58 (rev 02)
3f:00.1 Ethernet controller: Intel Corporation Device 0d58 (rev 02)
```

Table 5. XL710 Device ID

XL710 Device ID	Valid Configuration
0x0d58	2x2x25G 4x25G

Ensure that the `lspci` return entries match the device ID 0d58. Otherwise, your Intel FPGA PAC does not support 25G configurations. XL710 device ID 0d58 is specific to the Intel FPGA PAC N3000-N/2. Other NIC cards with the XL710 will have a different device ID.

6.1. Installing the Intel XL710 Driver

Select the appropriate i40e and iavf driver for the XL710 NVM based on *Table: Software/NVM Compatibility for XXV710* in the Intel Ethernet Controller X710/XXV710/XL710 Feature Support Matrix.

Remember: *Table: Software/NVM Compatibility for XXV710* is updated periodically.

Follow these steps to install the Intel XL710 driver:

1. Download the i40e driver from download center or subcribe to i40e driver on sourceforge.
2. Install driver as root.

   ```
   $ tar xvzf i40e-2.12.6.tar.gz ; cd i40e-2.12.6
   ```

   ```
   $ cd src
   ```

   ```
   $ sudo make install
   ```

   ```
   $ sudo rmmod i40e
   ```

   ```
   $ sudo insmod i40e.ko
   ```

3. Download the i40e virtual function driver from download center or subcribe to i40e virtual function driver on sourceforge.
4. Install driver as root.

   ```
   $ tar xzvf iavf-3.9.5.tar.gz ; cd iavf-3.9.5
   ```

   ```
   $ cd src
   ```

   ```
   $ sudo make install
   ```

   Insert the iavf kernel module to add support for virtual functions for Intel XL710:

   ```
   $ sudo insmod iavf.ko
   ```

Note: You may see the following errors during iavf driver install:

```
SSL error:02001002:system library:fopen:No such file or directory: crypto/bio/
bss_file.c:72
```

```
SSL error:2006D080:BIO routines:BIO_new_file:no such file: crypto/bio/
bss_file.c:79
```

It is safe to ignore these error messages. These errors are related to verification of digital signature of the driver. As the iavf Linux drivers are not digitally signed, the errors are to be expected on some operating system. These errors do not impact the functionality of the XL710 other than possible warnings in `dmesg`.

6.2. Updating the Intel XL710 Firmware

The Intel FPGA PAC N3000-N/2 is programmed with the following firmware version:

Intel XL710 Firmware	i40e driver	iavf driver
7.0	2.10.19.82	3.7.61.20

Note: Do not use firmware version 7.1 and 7.2 since these versions do not support Intel XL710 device ID 0d58.

In the future, you can choose to upgrade to Intel XL710 Firmware 7.3 or later and install the corresponding i40e and iavf driver. The following steps are instructive and provide guidance on how to perform firmware update for future versions:

1. Check the version compatibility between firmware and driver. Refer to the *Table: Software/NVM Compatibility for XXV710* in the Intel Ethernet Controller X710/XXV710/XL710 Feature Support Matrix.
2. Download the NVM Update package version 7.3 or above from download center.
3. Extract the `nvmupdate64e` tool:

   ```
   $ unzip NVMUpdatePackage_700_Series_<version number>.zip
   ```

   ```
   $ tar xvzf 700Series_NVMUpdatePackage_*_Linux.tar.gz
   ```

   ```
   $ cd 700Series/Linux_x64/
   ```

4. Upgrade Intel XL710 firmware as root:

   ```
   # ./nvmupdate64e
   ```

   For more details, refer to the corresponding Readme file.

Note: The NVM Updater checks all XL710 devices that can be updated with this firmware. The NVM Updater lists the XL710 devices for both Intel FPGA PAC N3000-N/2 as well as server NIC cards. Ensure that you update the desired XL710 devices.

Sample output:

```
# ./nvmupdate64e

Intel(R) Ethernet NVM Update Tool
NVMUpdate version 1.35.23.3
Copyright (C) 2013 - 2020 Intel Corporation.



WARNING: To avoid damage to your device, do not stop the update or reboot or
power off the system during this update.
Inventory in progress. Please wait [+*********]


Num Description                          Ver.(hex)  DevId S:B    Status
=== ================================== ============ ===== ====== ==============
01) Intel(R) Gigabit 4P X710/I350 rNDC 1.103(1.67)   1521 00:001 Update not
                                                                 available
02) Intel(R) Ethernet 10G 4P X710/I350  7.16(7.10)   1572 00:024 Update not
    rNDC                                                         available
03) Intel(R) Ethernet Controller        7.00(7.00)   0D58 00:061 Update
    XXV710 Intel(R) FPGA Programmable                            available
    Acceleration Card N3000 for
    Networking
04) Intel(R) Ethernet Controller        7.00(7.00)   0D58 00:065 Update
    XXV710 Intel(R) FPGA Programmable                            available
    Acceleration Card N3000 for
    Networking
05) Intel(R) Ethernet Controller        7.00(7.00)   0D58 00:137 Update
    XXV710 Intel(R) FPGA Programmable                            available
    Acceleration Card N3000 for
    Networking
06) Intel(R) Ethernet Controller        7.00(7.00)   0D58 00:141 Update
    XXV710 Intel(R) FPGA Programmable                            available
```

```
    Acceleration Card N3000 for
    Networking

Options: Adapter Index List (comma-separated), [A]ll, e[X]it
Enter selection: 3,4,5,6
Would you like to back up the NVM images? [Y]es/[N]o: N
Update in progress. This operation may take several minutes.
[....|*****]


Power Cycle is required to complete the update process.

Tool execution completed with the following status: All operations completed
successfully.
Press any key to exit.
```

Note: In the above sample output, XL710 devices of the Intel FPGA PAC N3000-N/2 are only updated.

After power cycling the server, verify that the NVM update was successful:

```
# ./nvmupdate64e

Intel(R) Ethernet NVM Update Tool
NVMUpdate version 1.35.23.3
Copyright (C) 2013 - 2020 Intel Corporation.


WARNING: To avoid damage to your device, do not stop the update or reboot or
power off the system during this update.
Inventory in progress. Please wait [*******+..]


Num Description                          Ver.(hex)  DevId S:B    Status
=== ================================== ============ ===== ====== ==============
01) Intel(R) Gigabit 4P X710/I350 rNDC 1.103(1.67)   1521 00:001 Update not
                                                                 available
02) Intel(R) Ethernet 10G 4P X710/I350  7.16(7.10)   1572 00:024 Update not
    rNDC                                                         available
03) Intel(R) Ethernet Controller        7.48(7.30)   0D58 00:061 Up to date
    XXV710 Intel(R) FPGA Programmable
    Acceleration Card N3000 for
    Networking
04) Intel(R) Ethernet Controller        7.48(7.30)   0D58 00:065 Up to date
    XXV710 Intel(R) FPGA Programmable
    Acceleration Card N3000 for
    Networking
05) Intel(R) Ethernet Controller        7.48(7.30)   0D58 00:137 Up to date
    XXV710 Intel(R) FPGA Programmable
    Acceleration Card N3000 for
    Networking
06) Intel(R) Ethernet Controller        7.48(7.30)   0D58 00:141 Up to date
    XXV710 Intel(R) FPGA Programmable
    Acceleration Card N3000 for
    Networking


Tool execution completed with the following status: All operations completed
successfully.
Press any key to exit.
```

7. Updating the Retimer Firmware

The Intel FPGA PAC N3000-N/2 is preloaded with Retimer firmware version 101c.1064. To verify the Retimer firmware version:

```
$ sudo fpgainfo phy
```

Sample output for 4x25G configuration:

```
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PHY ******//
Object Id : 0xED00001
PCIe s:b:d.f : 0000:8a:00.0
Device Id : 0x0b30
Numa Node : 1
Ports Num : 01
Bitstream Id : 0x23000110010310
Bitstream Version : 0.2.3
Pr Interface Id : f3c99413-5081-4aad-bced-07eb84a6d0bb
//****** PHY GROUP 0 ******//
Direction : Line side
Speed : 25 Gbps
Number of PHYs : 4
//****** PHY GROUP 1 ******//
Direction : Host side
Speed : 40 Gbps
Number of PHYs : 4
//****** Intel C827 Retimer ******//
Port0 25G : Up
Port1 25G : Up
Port2 25G : Up
Port3 25G : Up
Retimer A Version : 101c.1064
Retimer B Version : 0000.0000
```

Note: When using the 4x25G network configuration, Retimer B is held in reset to reduce power. Therefore, Retimer B firmware version is not listed in the above output.

Sample output for 2x2x25G configuration:

```
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PHY ******//
Object Id                     : 0xF100002
PCIe s:b:d.f                  : 0000:b2:00.0
Device Id                     : 0x0b30
Numa Node                     : 1
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
//****** PHY GROUP 0 ******//
Direction                     : Line side
Speed                         : 25 Gbps
Number of PHYs                : 4
//****** PHY GROUP 1 ******//
```

```
Direction                      : Host side
Speed                          : 40 Gbps
Number of PHYs                 : 4
//****** Intel C827 Retimer ******//
Port0 25G                      : Up
Port1 25G                      : Up
Port2 25G                      : Up
Port3 25G                      : Up
Retimer A Version              : 101c.1064
Retimer B Version              : 101c.1064
```

The following steps are instructive and provide guidance on how to perform firmware update for future Retimer versions:

1. Perform `fpgasupdate` to load the NIOS with Retimer firmware.
2. Perform reset:

   ```
   $ sudo rsu bmcimg [PCIe B:D.F]
   ```

3. Perform echo 1 to eeprom_load sysfs node of the card:

   ```
   $ echo 1 > /sys/class/fpga/intel-fpga-dev.0/intel-fpga-fme.0/spi-
   altera.0.auto/spi_master/spi0/spi0.0/pkvl/eeprom_load
   ```

4. Check the `eeprom_update_status` node for completion, the value must be `0x1111`.

   ```
   $ cat /sys/class/fpga/intel-fpga-dev.0/intel-fpga-fme.0/spi-altera.0.auto/
   spi_master/spi0/spi0.0/pkvl/eeprom_update_status
   ```

5. Perform reset:

   ```
   $ sudo rsu bmcimg [PCIe B:D.F]
   ```

6. Verify the Retimer firmware version:

   ```
   $ sudo fpgainfo phy
   ```

8. OPAE Tools

The following OPAE tools are provided:

- `fpgasupdate`: This tool updates Intel MAX 10 BMC image and firmware, root entry hash, and FPGA static region (SR) user image in the user partition.
- `fpgainfo`: This tool displays FPGA information derived from sysfs files.
- `bitstreaminfo`: This tool displays the authentication metadata prepended to the bitstream. For more information, refer to the Security User Guide: Intel FPGA Programmable Acceleration Card N3000.
- `fpgabist`: This tool performs board self-diagnostic tests for PCIe and external memories.
- `fpgadiag`: This tool performs board self-diagnostic tests for network loopback.
- `fpgad`: This service supports monitoring of critical sensors and prevents a server crash upon a threshold violation.
- `fpgastats`: This tool provides the FPGA Ethernet MAC statistics for both line side and host side.

8.1. Using `fpgasupdate`

The `fpgasupdate` tool updates board firmware including BMC and FPGA SR user image. This section describes how to update the FPGA SR user image.

The new Intel FPGA PAC N3000-N/2 is shipped with the 4x25G factory image in the user partition of FPGA flash. The following steps describe how to load FPGA images into the FPGA flash user partition. You may follow the steps to load or re-load the factory image if required.

Note: The Intel FPGA PAC N3000-N/2 only supports the 25G Ethernet configuration.

1. Select the Intel provided factory configuration image, `sr_vista_rot_4x25G-v1.3.16.bin` or `sr_vista_rot_2x2x25G-v1.3.16.bin`, and run the `fpgasupdate` command:

   ```
   $ sudo fpgasupdate /home/<User>/intelrtestack/bin/\
   sr_vista_rot_4x25G-v1.3.16.bin [PCIe B:D.F]
   ```

   Note: Running `fpgasupdate` involves binary file verification and writing the FPGA flash, as a result the `fpgasupdate` command takes approximately 40 minutes to complete per card.

Note: If you have programmed the static region root entry hash, then the `sr_vista_rot_*_unsigned.bin` must be signed with appropriate root key and code signing key using the appropriate Hardware Security Module (HSM). To ensure appropriate headers with metadata are included in the bitstream, every bitstream generated by AFU compile process must pass through the PACSign tool. For more information, refer to the Security User Guide: Intel FPGA Programmable Acceleration Card N3000.

2. Perform remote system update to power cycle the Intel FPGA PAC N3000-N/2 so that the updated images are loaded into FPGAs:

```
$ sudo rsu bmcimg [PCIe B:D.F]
```

Note: As a result of using the `rsu` command, the host rescans the PCIe bus and may assign a different Bus/Device/Function (B/D/F) value than the originally assigned value. The Intel XL710 Ethernet controller should be considered unavailable during the operation. Intel recommends you to stop or pause any applications until the update is complete.

8.2. Using `fpgainfo`

Command synopsis: `fpgainfo <command> [<args>]`

Table 6. `fpgainfo` Commands

command	args (optional)	Description
	--help, -h	Prints help information and exit
	--bus, -B	Provides PCIe bus number of resource
	--device, -D	Provides PCIe device number of resource
	--function, -F	Provides PCIe function number of resource
errors fme	--clear, -c	Provides/clear errors of FME
errors port	--clear, -c	Provides/clear errors of port
errors all	--clear, -c	Provides/clear errors of both FME and port
power		Provides total power in watts that the FPGA hardware consumes
temp		Provides FPGA temperature values in degrees Celsius
port		Provides information about the port
fme		Provides information about the FME
bmc		Provides BMC sensors information
mac		Provides information about MAC ROM connected to FPGA
phy	-G <group>	Provides information about Ethernet PHYs in FPGA. <group> can be 0, 1, all.
security		Provides information about whether the Intel FPGA PAC has root entry hash for programmed FIM/SR or whether the CSK ID for SR/BMC is cancelled.

Example 1. `fpgainfo fme`

```
$ sudo fpgainfo fme
```

Sample output based on 4x25G:

```
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** FME ******//
Object Id                     : 0xED00000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
Boot Page                     : user
```

Example 2. fpgainfo bmc

```
$ sudo fpgainfo bmc
```

Sample output based on 4x25G:

```
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** BMC SENSORS ******//
Object Id                     : 0xED00000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
( 1) Board Power                  : 73.07 Watts
( 2) 12V Backplane Current        : 3.26 Amps
( 3) 12V Backplane Voltage        : 12.11 Volts
( 4) 1.2V Voltage                 : 1.19 Volts
( 6) 1.8V Voltage                 : 1.79 Volts
( 8) 3.3V Voltage                 : 3.24 Volts
(10) FPGA Core Voltage            : 0.90 Volts
(11) FPGA Core Current            : 17.12 Amps
(12) FPGA Core Temperature        : 68.50 Celsius
(13) Board Temperature            : 45.00 Celsius
(14) QSFP A Voltage               : N/A
(15) QSFP A Temperature           : N/A
(24) 12V AUX Current              : 2.77 Amps
(25) 12V AUX Voltage              : 12.12 Volts
(37) QSFP B Voltage               : N/A
(38) QSFP B Temperature           : N/A
(44) Retimer A Core Temperature   : 74.00 Celsius
(45) Retimer A Serdes Temperature : 75.00 Celsius
(46) Retimer B Core Temperature   : 0.00 Celsius
(47) Retimer B Serdes Temperature : 0.00 Celsius
```

Note: The BMC supports QSFP modules compliant with SFF8436 or SFF8636 standard. If the QSFP module does not support Digital Diagnostics Monitoring, the output for QSFP voltage, temperature, temperature high fatal, and temperature high warning is N/A.

Example 3. fpgainfo phy

```
$ sudo fpgainfo phy -B 0xb2
```

Note: For 4x25G configuration, only Retimer A is functional in the Intel FPGA PAC N3000-N/2.

Sample output based on 4x25G:

```
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PHY ******//
Object Id                     : 0xED00001
PCIe s:b:d.f                  : 0000:8a:00.0
Device Id                     : 0x0b30
Numa Node                     : 1
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
//****** PHY GROUP 0 ******//
Direction                     : Line side
Speed                         : 25 Gbps
Number of PHYs                : 4
//****** PHY GROUP 1 ******//
Direction                     : Host side
Speed                         : 40 Gbps
Number of PHYs                : 4
//****** Intel C827 Retimer ******//
Port0 25G                     : Up
Port1 25G                     : Up
Port2 25G                     : Up
Port3 25G                     : Up
Retimer A Version             : 101c.1064
Retimer B Version             : 0000.0000
```

Example 4. `fpgainfo security`

```
$ sudo fpgainfo security
```

Sample output based on 4x25G:

```
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** SECURITY ******//
Object Id                     : 0xED00000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
FIM/SR root entry hash        : hash not programmed
BMC root entry hash           :
0xec0f42d3af138e3eca7141107f7fed5f7c13846fadbba884e51ad26bf36a3d21
PR root entry hash            : hash not programmed
SMB parameters update counter : 0
User flash update counter     : 2
FIM/SR CSK IDs canceled       : None
BMC CSK IDs canceled          : None
AFU CSK IDs canceled          : None
```

Note: BMC root entry hash and BMC CSK ID cancellation bitstreams are provided by Intel.

`PR root entry hash` and `AFU CSK ID` canceled parameter outputs are not applicable to Intel FPGA PAC N3000-N/2. For more information about the `fpgainfo security` command, refer to the *Accessing Intel FPGA PAC N3000 Version and Authentication Information* section of Security User Guide: Intel FPGA Programmable Acceleration Card N3000 Variants

8.3. Test PCIe and External Memories with `fpgabist`

Tests are included to demonstrate the performance of PCIe and external memories.

Requirements:

- OPAE tool `fpgabist` requires hugepage to be set:
  - For CentOS:

    ```
    $ sudo sh -c "echo 20 > /sys/kernel/mm/hugepages/hugepages-2048kB/\
    nr_hugepages"
    ```

  - For RHEL:

    ```
    # echo 200 > /proc/sys/vm/nr_hugepages
    ```

  Rerun the above command after a power cycle to the Intel FPGA PAC N3000-N/2 or after a server reboot or server power cycle.

Note: The `fpgabist` diagnostic tool only works when the Intel supplied factory images are programmed into the Intel FPGA PAC N3000-N/2.

Example 5. Using `fpgabist`

```
# fpgabist -B 0x8a -i 0x0b30
```

Note: Your bus (-B) value may be different.

Related Information

fpgabist Sample Output on page 63

8.4. Test Network Loopback using `fpgadiag`

The test requires use of an external traffic generator to send traffic through QSFP ports, the `--side host` enables loopback on the host side of the Intel Arria 10 FPGA.

```
$    sudo fpgadiag -B 0x3e -m fpgalpbk --side host –-direction remote –-enable
```

To disable:

```
$    sudo fpgadiag -B 0x3e -m fpgalpbk --side host –-direction remote --disable
```

Note: Since all four Ethernet channels in the 4x25G configuration are in QSFP A, only one QSFP port indicates linkup status through LEDs.

9. Sample Test: Native Loopback

This section describes how to run a memory copy test using the Intel provided FPGA factory image and `hello_fpga.c` host program. The FPGA factory image includes logic to support this test and an internal register with the expected **AFU UUID**. The `hello_fpga.c` only works with an FPGA image with this **AFU UUID**. The acceleration logic (NLB) in the FPGA is programmed to copy CSR_NUM_LINES (cache lines) from source to destination buffer on the host system. For more information refer to the Native Loopback Accelerator Functional Unit (AFU) User Guide for Intel FPGA Programmable Acceleration Card N3000.

Make sure the hugepage is allocated:

- For CentOS:

  ```
  $ sudo sh -c "echo 20 > /sys/kernel/mm/hugepages/hugepages-2048kB/\
  nr_hugepages"
  ```

- For RHEL:

  ```
  # echo 20 > /proc/sys/vm/nr_hugepages
  ```

Note: Commands must be run as root.

```
$    cd /home/<user>/intelrtestack/sw_sample
```

```
$    gcc -o hello_fpga -std=gnu99 -rdynamic -ljson-c -luuid -lpthread \
-lopae-c -lm -Wl,-rpath -lopae-c hello_fpga.c
```

```
$    sudo ./hello_fpga
```

Sample output:

```
Running Test
Running on bus 0x8a.
dfh = 100000008000001f
id[0] = c000c9660d824272
id[1] = 9aeffe5f84570612
dfh = 2000000080000000
id[0] = a9149a35bace01ea
id[1] = ef82def7f6ec40fc
dfh = 2000000080000000
id[0] = a9149a35bace01ea
id[1] = ef82def7f6ec40fc
dfh = 2000000080000000
id[0] = a9149a35bace01ea
id[1] = ef82def7f6ec40fc
dfh = 2000000080000000
id[0] = a9149a35bace01ea
id[1] = ef82def7f6ec40fc
dfh = 1000010080001070
id[0] = f89e433683f9040b
```

```
id[1] = d8424dc4a4a3c413
Found NLB0 at offset 0x28000
Done Running Test
```

Note: On a multi card system, pass PCIe bus argument -B 0x<xx>

10. Configuring Ethernet Interfaces

The Intel FPGA PAC N3000-N/2 contains multiple Ethernet MAC points where each point has specific naming, monitoring and configuration operations.

Note: The Intel provided factory image does not support Ethernet auto-negotiation and as a result, you must manually provision port settings to match the remote Ethernet link partner.

The following figures illustrate the Ethernet data path for each network configuration.

Figure 9. 4x25G Configuration

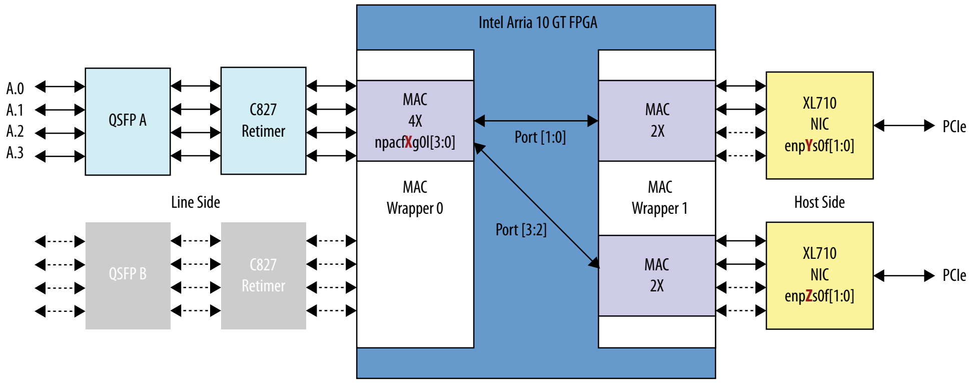
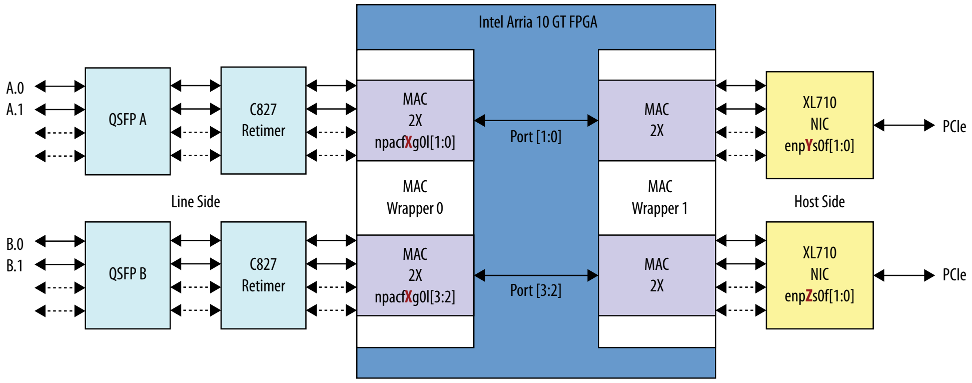


Figure 10. 2x2x25G Configuration



The above figures illustrate example device naming conventions for the XL710 `enp[Y:Z]s0f[1:0]`. Your server may have a different naming convention and numbering scheme. Additionally, the Intel Arria 10 FPGA contains Ethernet MACs with network logical naming scheme of `npacf[X]g0l[3:0]`. You must obtain the network logical names to use Linux tools for link configuration and monitoring. To find the network logical names of a specific Intel FPGA PAC N3000-N/2 in your server, perform the following steps:

1. List the available Intel FPGA PAC N3000-N/2 in your server using:

```
$ sudo fpgainfo fme
```

   Sample output:

```
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** FME ******//
Object Id                     : 0xED00001
PCIe s:b:d.f                  : 0000:1c:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
Boot Page                     : user
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** FME ******//
Object Id                     : 0xED00000
PCIe s:b:d.f                  : 0000:8a:00.0
Device Id                     : 0x0b30
Numa Node                     : 1
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
Boot Page                     : user
```

2. Use the following command to find the logical name(s) of the Ethernet interfaces on the target Intel FPGA PAC N3000-N/2:

```
ls -la /sys/class/net
```

   In the following example, if you want to change settings on the Intel FPGA PAC N3000-N/2 with PCIe b:d.f 8a:00.0, then ensure that the PCIe device ID for enp136s0f0, enp136s0f1, enp139s0f0 and enp139sof1 is in the range of 88-8b.

   Sample output:

```
drwxr-xr-x.  2 root root 0  .

drwxr-xr-x. 59 root root 0  ..

lrwxrwxrwx.  1 root root 0  enp0s20f0u4 -> ../../devices/
pci0000:00/0000:00:14.0/usb1/1-4/1-4:1.0/net/enp0s20f0u4

lrwxrwxrwx.  1 root root 0  enp136s0f0 -> ../../devices/
pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:08.0/0000:88:00.0/net/
enp136s0f0

lrwxrwxrwx.  1 root root 0  enp136s0f1 -> ../../devices/
pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:08.0/0000:88:00.1/net/
enp136s0f1

lrwxrwxrwx.  1 root root 0  enp139s0f0 -> ../../devices/
pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:10.0/0000:8b:00.0/net/
enp139s0f0

lrwxrwxrwx.  1 root root 0  enp139s0f1 -> ../../devices/
pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:10.0/0000:8b:00.1/net/
enp139s0f1
```

```
lrwxrwxrwx.  1 root root 0  enp175s0f0 -> ../../devices/pci0000:ae/
0000:ae:00.0/0000:af:00.0/net/enp175s0f0

lrwxrwxrwx.  1 root root 0  enp175s0f1 -> ../../devices/pci0000:ae/
0000:ae:00.0/0000:af:00.1/net/enp175s0f1

lrwxrwxrwx.  1 root root 0  enp26s0f0 -> ../../devices/
pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:08.0/0000:1a:00.0/net/enp26s0f0

lrwxrwxrwx.  1 root root 0  enp26s0f1 -> ../../devices/
pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:08.0/0000:1a:00.1/net/enp26s0f1

lrwxrwxrwx.  1 root root 0  enp29s0f0 -> ../../devices/
pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:10.0/0000:1d:00.0/net/enp29s0f0

lrwxrwxrwx.  1 root root 0  enp29s0f1 -> ../../devices/
pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:10.0/0000:1d:00.1/net/enp29s0f1

lrwxrwxrwx.  1 root root 0  enp59s0f0 -> ../../devices/pci0000:3a/
0000:3a:00.0/0000:3b:00.0/net/enp59s0f0

lrwxrwxrwx.  1 root root 0  enp59s0f1 -> ../../devices/pci0000:3a/
0000:3a:00.0/0000:3b:00.1/net/enp59s0f1

lrwxrwxrwx.  1 root root 0  enp94s0f0 -> ../../devices/pci0000:5d/
0000:5d:00.0/0000:5e:00.0/net/enp94s0f0

lrwxrwxrwx.  1 root root 0  enp94s0f1 -> ../../devices/pci0000:5d/
0000:5d:00.0/0000:5e:00.1/net/enp94s0f1

lrwxrwxrwx.  1 root root 0  lo -> ../../devices/virtual/net/lo

lrwxrwxrwx.  1 root root 0  npacf0g0l0 -> ../../devices/
pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:09.0/0000:1c:00.0/fpga/intel-
fpga-dev.0/intel-fpga-fme.0/pac_n3000_net.2.auto/net/npacf0g0l0

lrwxrwxrwx.  1 root root 0  npacf0g0l1 -> ../../devices/
pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:09.0/0000:1c:00.0/fpga/intel-
fpga-dev.0/intel-fpga-fme.0/pac_n3000_net.2.auto/net/npacf0g0l1

lrwxrwxrwx.  1 root root 0  npacf0g0l2 -> ../../devices/
pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:09.0/0000:1c:00.0/fpga/intel-
fpga-dev.0/intel-fpga-fme.0/pac_n3000_net.2.auto/net/npacf0g0l2

lrwxrwxrwx.  1 root root 0  npacf0g0l3 -> ../../devices/
pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:09.0/0000:1c:00.0/fpga/intel-
fpga-dev.0/intel-fpga-fme.0/pac_n3000_net.2.auto/net/npacf0g0l3

lrwxrwxrwx.  1 root root 0  npacf1g0l0 -> ../../devices/
pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:09.0/0000:8a:00.0/fpga/intel-
fpga-dev.1/intel-fpga-fme.1/pac_n3000_net.6.auto/net/npacf1g0l0

lrwxrwxrwx.  1 root root 0  npacf1g0l1 -> ../../devices/
pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:09.0/0000:8a:00.0/fpga/intel-
fpga-dev.1/intel-fpga-fme.1/pac_n3000_net.6.auto/net/npacf1g0l1

lrwxrwxrwx.  1 root root 0  npacf1g0l2 -> ../../devices/
pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:09.0/0000:8a:00.0/fpga/intel-
fpga-dev.1/intel-fpga-fme.1/pac_n3000_net.6.auto/net/npacf1g0l2

lrwxrwxrwx.  1 root root 0  npacf1g0l3 -> ../../devices/
pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:09.0/0000:8a:00.0/fpga/intel-
fpga-dev.1/intel-fpga-fme.1/pac_n3000_net.6.auto/net/npacf1g0l3

lrwxrwxrwx.  1 root root 0  virbr0 -> ../../devices/virtual/net/virbr0

lrwxrwxrwx.  1 root root 0  virbr0-nic -> ../../devices/virtual/net/virbr0-
nic
```

For example:

This listing is an example of the 4x25G network configuration. The logical device names `npacf1g0l[0, 1, 2, 3]` represents the Ethernet MAC wrapper 0 on the line side of the Intel Arria 10 FPGA. The logical device names `enp[136 and 139]s0f[1 and 0]` are the XL710 Ethernet ports.

The `pac_n3000_net` platform device driver creates the standard Linux network device interfaces for each Intel Arria 10 FPGA Ethernet MAC pair. It provides C827 re-timer information for unified network status reporting. It enables use of standard Linux tools for both link configuration and monitoring.

```
$ lsmod | grep pac_n3000_net
pac_n3000_net          28483  1 c827_retimer
```

10.1. Modifying the Interface Maximum Transmission Unit (MTU) Size

The default MTU size for 25G FPGA MAC Wrapper 0 and Wrapper 1 is 9600. The default MTU size for XL710 is 1500 bytes. You must configure FPGA MAC wrappers and XL710 to have the same MTU setting to ensure each MAC will allow your desired maximum packet size.

Command for configuring MTU of FPGA MAC wrapper 0:

```
$ sudo ip link set dev npacfXgYlZ mtu <#>

<#> = desired MTU setting
```

Command for configuring MTU of XL710:

```
$ ip sudo link set dev <XL710 interface name> mtu <#>

<#> = desired MTU setting
```

Example of current settings:

```
$ ip link show npacf0g0l0

38: npacf0g0l0: <LOWER_UP> mtu 9600 qdisc noop state UNKNOWN mode DEFAULT group
default qlen 1000
    link/generic
```

```
$ ip link show enp26s0f0

44: enp26s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
mode DEFAULT group default qlen 1000
    link/ether 64:4c:36:11:f0:c8 brd ff:ff:ff:ff:ff:ff
```

Note: The XL710 default MTU setting differs from the Intel Arria 10 FPGA MTU setting.

Example: Set MTU to 9600 for both FPGA and XL710

```
$ sudo ip link set dev npacf0g0l0 mtu 9600
```

```
$ sudo ip link set dev enp26s0f0 mtu 9600
```

```
$ ip link show npacf0g0l0

38: npacf0g0l0: <LOWER_UP> mtu 9600 qdisc noop state UNKNOWN mode DEFAULT group
default qlen 1000
    link/generic
```

```
$ ip link show enp26s0f0

44: enp26s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9600 qdisc mq state UP
mode DEFAULT group default qlen 1000
    link/ether 64:4c:36:11:f0:c8 brd ff:ff:ff:ff:ff:ff
```

The FPGA MAC Wrapper 1 MTU is an internal MAC instance for transport of Ethernet frames through the FPGA. Therefore, it does not have a logical network representation in Linux. It is unlikely you will need to provision MAC Wrapper 1 settings. The default MTU setting for MAC wrapper 1 is 9600 bytes.

To check FPGA MAC Wrapper 1 MTU:

```
$ sudo fpgadiag -B <bus> -m fpgamac --side=host --mtu
```

Sample output:

```
================================================================
maximum frame length     | transmit         | receive          |
mac 0                    | 9600             | 9600             |
mac 1                    | 9600             | 9600             |
mac 2                    | 9600             | 9600             |
mac 3                    | 9600             | 9600             |
```

If you need to change the MAC Wrapper 1 MTU setting for all ports, use the following command:

```
$ fpgadiag -B <bus> -m fpgamac --side=host --mtu <#>

<bus> = PCIe bus of FPGA in 0xYZ format
<#> = desired MTU setting
```

For example:

```
$ sudo fpgadiag -B 0x1b -m fpgamac --side=host --mtu 1500

$ sudo fpgadiag -B 0x1b -m fpgamac --side=host --mtu

Output:
=====================================================
maximum frame length     | transmit         | receive          |
mac 0                    | 1500             | 1500             |
mac 1                    | 1500             | 1500             |
mac 2                    | 1500             | 1500             |
mac 3                    | 1500             | 1500             |
```

10.2. Setting Forward Error Correction (FEC) Mode

The Intel FPGA PAC N3000-N/2 supports different forward error correction (FEC) modes to enhance data reliability for 25GbE network configurations. The default FEC mode is Reed Solomon FEC. The following FEC modes are available:

Table 7. FEC Modes

fec_mode	Mode
no	No FEC
kr	Fire Code Forward Error Correction (IEEE 802.3 Clause 74)
rs	Reed Solomon Forward Error Correction (IEEE 802.3 Clause 108)

To set FEC mode:

```
$ sudo fecmode -B <bus> <mode>

<mode> = ‘no’, ‘kr’, ‘rs’
<bus> = PCIe bus of FPGA in the format “0xYZ”
```

Note: The `fecmode` command causes a board level rsu event while changing FEC modes. The rsu event causes a board level reset which causes previously configured Ethernet settings to revert back to default settings. Intel recommends you first set FEC mode, then configure Ethernet and other board level settings. The rsu event may also cause the PCIe bus number to change.

To get FEC mode:

```
$ fecmode -B <bus>

<bus> = PCIe bus of FPGA in the format “0xYZ”
```

For example:

```
$ fecmode -B 0xb3
FEC mode in current driver: rs
FEC mode in current hardware: rs
```

```
$ sudo fecmode -B 0xb3 kr
reloading driver with new parameter 'kr'
performing remote system update
2019-11-14 10:00:34,121 - [[pci_address(0000:b3:00.0), pci_id(0x8086, 0x0b30)]]
performing RSU operation
2019-11-14 10:00:34,123 - [[pci_address(0000:ae:00.0), pci_id(0x8086, 0x2030)]]
removing device from PCIe bus
2019-11-14 10:00:34,124 - waiting 10 seconds for boot
2019-11-14 10:00:44,135 - rescanning PCIe bus: /sys/devices/pci0000:ae/pci_bus/
0000:ae
2019-11-14 10:00:49,119 - RSU operation complete
Done
```

```
$ fecmode -B 0xb3
FEC mode in configuration: kr
FEC mode in current driver: kr
FEC mode in current hardware: kr
```

10.3. Ethernet Pause Flow Control

The Intel FPGA PAC N3000-N/2 supports pause frame operation for:

- 25G as described in Flow Control section of *25G Ethernet Intel Arria 10 FPGA IP User Guide*.

The Intel provided FPGA image supports generation of pause frames in response to the internal Intel Arria 10 buffer nearing an overflow condition. The provided FPGA image does not support response to received pause frames. The pause frame generation and response to pause frame reception is disabled by default. You can read the current setting of pause frame generation using:

```
$ ethtool --show-pause npacf0g0l0
```

```
Pause parameters for npacf0g0l0:
Autonegotiate:    off
RX:         off
TX:         off
```

To turn on the pause frame generation:

1. If you have more than one Intel FPGA PAC N3000-N/2 installed in your server, first determine the proper `sysfs entry` for setting the transmit pause registers.
2. Set Transmit Pause Quanta: Configurable register used to set the desired delay time embedded in the pause frame packet requesting peer to stop transmitting for a period defined in the quanta. One quanta equals 512-bit times.
3. Set Transmit Hold-off Quanta: Configurable register used to set the desired delay time between consecutive pause frames packets in quanta.
4. Pause Frame Enable: Allows you to enable or disable pause frame behavior using the `ethtool`.

Each Ethernet port has a set of transmit pause or hold-off registers. To find the proper sysfs entry, use the following command:

```
$ ls -l /sys/class/fpga/intel-fpga-dev.*
```

Sample output:

```
lrwxrwxrwx. 1 root root 0 Apr  1 06:13 /sys/class/fpga/intel-fpga-dev.0 -
> ../../devices/pci0000:85/0000:85:00.0/0000:86:00.0/0000:87:09.0/0000:8a:00.0/
fpga/intel-fpga-dev.0
```

```
lrwxrwxrwx. 1 root root 0 Apr  1 09:01 /sys/class/fpga/intel-fpga-dev.1 -
> ../../devices/pci0000:17/0000:17:00.0/0000:18:00.0/0000:19:09.0/0000:1b:00.0/
fpga/intel-fpga-dev.1
```

In the output above, there are two Intel FPGA PAC N3000-N/2. For this example, the PCIe device ID of the desired card is 1b:00.0, hence the base sysfs path is `/sys/class/fpga/intel-fpga-dev.1`.

Setting the Transmit Pause Quanta sysfs entry to 200 quanta for port 0:

```
# echo 200 > /sys/class/fpga/intel-fpga-dev.1/intel-fpga-fme.1/
pac_n3000_net.10.auto/net/npacf1g0l0/tx_pause_frame_quanta
```

```
# cat /sys/class/fpga/intel-fpga-dev.1/intel-fpga-fme.1/
pac_n3000_net.10.auto/net/npacf1g0l0/tx_pause_frame_quanta
```

```
0xc8
```

Setting the Transmit Hold-off Quanta sysfs entry to 200 quanta for port 0:

```
# echo 200 > /sys/class/fpga/intel-fpga-dev.1/intel-fpga-fme.1/
pac_n3000_net.10.auto/net/npacf1g0l0/tx_pause_frame_holdoff
```

```
# cat /sys/class/fpga/intel-fpga-dev.1/intel-fpga-fme.1/
pac_n3000_net.10.auto/net/npacf1g0l0/tx_pause_frame_holdoff
```

```
0xc8
```

To verify:

```
# ethtool --pause npacf0g0l0 tx on
```

```
# ethtool --show-pause npacf0g0l0
```

```
Pause parameters for npacf0g0l0:
Autonegotiate:    off
RX:        off
TX:        on
```

Note: All Ethernet settings listed in this section are not persistent across power cycles or server reboots or rsu. After power cycle or server reboot or rsu, the Intel FPGA PAC N3000-N/2 returns to default settings. The rsu command causes change in the PCIe B:D.F value.

10.4. Get Link Statistics

You can get link statistics using:

- Linux `ethtool`
- OPAE `fpgastats` command

```
$ ethtool -S npacf0g0l0
```

This command provides the Ethernet statistics counts for port 0. Implementation of Ethernet statistics count is optional, check with your FPGA workload provider. For example, the FlexRAN image does not implement Ethernet statistics.

Note: The Ethernet links between the FPGA and XL710 controllers is always up when the FPGA is being programmed.

The OPAE `fpgastats` command lists all FPGA Ethernet MAC counters on the Intel FPGA PAC N3000-N/2 specified by bus number. The `fpgastats` command is useful for detecting packet drops inside the FPGA because it provides both Ethernet wrapper 0 and 1 in an easily read format.

```
$ sudo fpgastats [-h] [--segment SEGMENT] [--bus BUS] [--device DEVICE] [--
function FUNCTION] [--clear] [--debug]

optional arguments:
  -h, --help            show this help message and exit
  --segment SEGMENT, -S SEGMENT
                        Segment number of PCIe device
  --bus BUS, -B BUS     Bus number of PCIe device
  --device DEVICE, -D DEVICE
                        Device number of PCIe device
  --function FUNCTION, -F FUNCTION
```

```
                        Function number of PCIe device
  --clear, -c           Clear statistics
  --debug, -d           Output debug information
```

You can clear Ethernet counts with the clear option:

```
$ sudo fpgastats -B 0x8a -c
```

Note: The XL710 controller does not support the OPAE `fpgastats` command.

10.5. Ethernet Link Status

The external Ethernet link is connected directly to the retimer device. The link status of the external Ethernet link is obtained from the retimer. The Intel Arria 10 FPGA and Intel XL710 device have status indications for Ethernet link status, however these are status of internal links between components.

You must obtained the Ethernet link status from the Retimers. You can obtain the Ethernet link status either using the OPAE `fpgainfo phy` command or using the following sysfs node which can be read for link status:

```
$ cat /sys/bus/spi/drivers/intel-max10/spi0.0/pkvl/status
0xf
```

If your server has more than one Intel FPGA PAC N3000-N/2 installed, then the following sysfs node must correspond to the desired Intel FPGA PAC N3000-N/2.

```
 /sys/bus/spi/drivers/intel-max10/spi<X>.0
```

Use the PCIe B:D.F of the desired Intel FPGA PAC N3000-N/2 to determine the proper value of *<X>*. The value returned from this sysfs node is a bit level representation of link status where **1** corresponds to link up and **0** is link down.

- Bits [0:3] corresponds to links 0-3 of QSFP A
- Bits [4:7] corresponds to links 0-3 of QSFP B

For example, consider a returned value of 0x33:

QSFP	Link 3	Link 2	Link 1	Link 0
QSFP A	0 = Link Down	0 = Link Down	1 = Link Up	1 = Link Up
QSFP B	0 = Link Down	0 = Link Down	1 = Link Up	1 = Link Up

For 4x25G network configuration, only QSFP A is used and links 0-3 are valid. For 2x2x25G network configuration, both QSFP A and B are used, however only links 1 and 0 are valid.

11. Testing Network Loopback Using Data Plane Development Kit (DPDK)

This section demonstrates how to compile and bind the Data Plane Development Kit (DPDK) drivers for the Intel FPGA PAC N3000-N/2. It also demonstrates how to setup DPDK to test the Ethernet datapath. For more in-depth support of DPDK, reach out to the DPDK Community.

Before starting DPDK, you must perform configuration steps described in Configuring Ethernet Interfaces on page 38 as it relies on the FPGA being bound to OPAE driver (`pac_n3000_net`). While using DPDK, FPGA is unbound from this driver and bound to the `vfio-pci` driver.

Follow these steps to install DPDK for testing network loopback:

1. You must enable the Intel IOMMU driver on the host. Complete the following steps to enable the Intel IOMMU driver:

   a. Add **iommu=pt intel_iommu=on** to the GRUB_CMDLINE_LINUX entry by editing `/etc/default/grub`. For example:

   ```
   GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rhel/root
   rd.lvm.lv=rhel/\
   swap rhgb quiet pci=realloc intel_iommu=on iommu=pt"
   ```

   For RHEL: Additionally, add **pci=realloc** to `GRUB_CMDLINE_LINUX` entry.

   b. GRUB reads its configuration from either the */boot/grub2/grub.cfg* file on traditional BIOS-based machines or from the */boot/efi/EFI/redhat/ grub.cfg* file on UEFI machines. Depending on your system, execute one of the instructions below as root:

   - BIOS based machine:

     ```
     grub2-mkconfig -o /boot/grub2/grub.cfg
     ```

   - UEFI based machine:

     ```
     grub2-mkconfig -o /boot/efi/EFI/redhat/ grub.cfg
     ```

     ```
     # grub2-mkconfig -o /boot/efi/EFI/redhat/
     Generating grub configuration file ...
     Found linux image: /boot/vmlinuz-3.10.0-957.el7.x86_64
     Found initrd image: /boot/initramfs-3.10.0-957.el7.x86_64.img
     Found linux image: /boot/vmlinuz-0-
     rescue-594cabaaf9a84c6ea0a5167c89ad916d
     Found initrd image: /boot/initramfs-0-
     rescue-594cabaaf9a84c6ea0a5167c89ad916d.img
     /usr/sbin/grub2-mkconfig: line 290: /boot/efi/EFI/redhat/: Is a
     directory
     ```

c. Reboot the server to apply the new GRUB configuration file.

d. To verify the GRUB update, run the following command:

```
$ cat /proc/cmdline
```

The sample output below shows **intel_iommu=on** on the kernel command line.

```
BOOT_IMAGE=/vmlinuz-3.10.0-957.el7.x86_64 root=/dev/mapper/rhel-root ro
default_hugepagesz=1G hugepagesz=1G hugepages=64 hugepagesz=2M
hugepages=2048 nosoftlockup mce=ignore_ce audit=0
isolcpus=1-11,24-35,13-23,36-47 nohz_full=1-11,24-35,13-23,36-47
rcu_nocbs=1-11,24-35,13-23,36-47 pci=realloc intel_iommu=on iommu=pt
enforcing=0 crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap
rhgb quiet skew_tick=1
```

2. Install the required packages:

```
$   sudo yum install readline-devel libpcap libpcap-devel numactl-devel
```

You need to download these two extra Fedora packages:

- `libfdt-1.4.7-3.fc30.x86_64.rpm`
- `libfdt-devel-1.4.7-3.fc30.x86_64.rpm`

To install:

```
$ sudo rpm -i <RPM file>
```

To check installation:

```
$ rpm -qa | grep libfdt
```

3. Compile and bind drivers:

a. Download DPDK code from DPDK community and checkout release 19.11.

```
$ git clone https://github.com/DPDK/dpdk.git
```

```
$ cd dpdk
```

```
$ git pull
```

```
Already up-to-date.
```

```
$ git reset --hard 31b798a6f08e9b333b94b8bb26910209aa810b73
```

```
HEAD is now at 31b798a build: avoid overlinking
```

```
$ wget https://patches.dpdk.org/series/6821/mbox/
```

```
Length: 179795 (176K) [text/plain]
Saving to: \u2018index.html\u2019

100%[=============>] 179,795      554KB/s    in 0.3s

(554 KB/s) - \u2018index.html\u2019 saved [179795/179795]
```

```
$ git am index.html
```

```
Applying: net/i40e: i40e support ipn3ke FPGA port bonding
Applying: raw/ifpga/base: add irq support
Applying: raw/ifpga/base: clear pending bit
Applying: raw/ifpga/base: add SEU error support
Applying: raw/ifpga/base: add device tree support
```

```
Applying: raw/ifpga/base: align the send buffer for SPI
Applying: raw/ifpga/base: add sensor support
Applying: raw/ifpga/base: introducing sensor APIs
Applying: raw/ifpga/base: update SEU register definition
Applying: raw/ifpga: add SEU error handler
Applying: raw/ifpga: add PCIe BDF devices tree scan
Applying: net/ipn3ke: remove configuration for i40e port bonding
Applying: raw/ifpga/base: add secure support
Applying: raw/ifpga/base: configure FEC mode
Applying: raw/ifpga/base: clean fme errors
Applying: raw/ifpga/base: add new API get board info
Applying: raw/ifpga: add lightweight fpga image support
Applying: raw/ifpga/base: add multiple cards support
```

```
$ vim config/common_linux
// modify CONFIG_RTE_LIBRTE_PMD_IFPGA_RAWDEV=n to
CONFIG_RTE_LIBRTE_PMD_IFPGA_RAWDEV=y
```

```
$ vim config/common_base
// modify CONFIG_RTE_LIBRTE_IPN3KE_PMD=n to
CONFIG_RTE_LIBRTE_IPN3KE_PMD=y
```

b. Build DPDK and export RTE_SDK path to point to dpdk directory:

Note: Ignore the message "*Build complete [x86_64-native-linuxapp-gcc] Installation cannot run with T defined and DESTDIR undefined*"

```
$    export RTE_SDK=$PWD
```

```
$    export RTE_TARGET=x86_64-native-linuxapp-gcc
```

```
$    make config T=x86_64-native-linuxapp-gcc
```

```
$    make install -j8 T=x86_64-native-linuxapp-gcc
```

Note: $RTE_SDK points to the extracted dpdk source location.

c. Bind FPGA and NIC to `vfio-pci` driver as shown below. The sample output below shows result for 2x2x25G configuration.

Table 8. Output Differences

Configuration	Ports per Intel Ethernet Controller XL710-BM2 NIC	Device ID for Intel Ethernet Controller XL710-BM2 NIC Ports
2x2x25G	2	0d58
4x25G	2	0d58

```
$    cd $RTE_SDK
```

Check the binding between driver and device with the following command:

```
$    ./usertools/dpdk-devbind.py --status-dev net
```

```
Sample output:
Network devices using DPDK-compatible driver
============================================
<none>
Network devices using kernel driver
===================================
0000:18:00.0 'NetXtreme BCM5720 Gigabit Ethernet PCIe 165f' if=em1
drv=tg3 unused=
0000:18:00.1 'NetXtreme BCM5720 Gigabit Ethernet PCIe 165f' if=em2
drv=tg3 unused= *Active*
0000:19:00.0 'NetXtreme BCM5720 Gigabit Ethernet PCIe 165f' if=em3
```

```
drv=tg3 unused=
0000:19:00.1 'NetXtreme BCM5720 Gigabit Ethernet PCIe 165f' if=em4
drv=tg3 unused=
0000:14:00.0 'Device 0d58' if=p1p1 drv=i40e unused=
0000:14:00.1 'Device 0d58' if=p1p2 drv=i40e unused=
0000:16:00.0 'Device 0d58' if=p1p3 drv=i40e unused=
0000:16:00.1 'Device 0d58' if=p1p4 drv=i40e unused=
Other Network devices
=====================
0000:15:00.0 'Device 0b30' unused=intel_fpga_pci
```

Install the vfio-pci kernel driver:

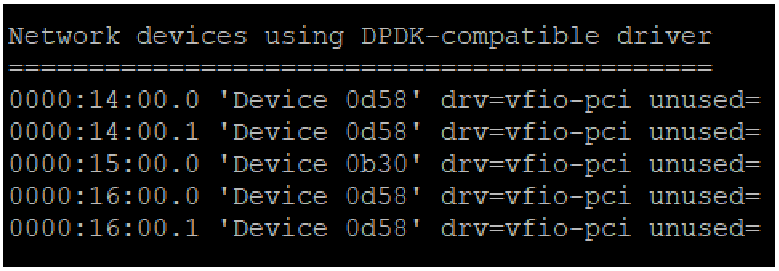
```
$    sudo modprobe vfio-pci
```

Bind the FPGA and FVL PFs to DPDK driver. Replace Bus:Device.Function with your output from above step.

```
$    sudo ./usertools/dpdk-devbind.py -b vfio-pci 14:00.0 \
14:00.1 15:00.0 16:00.0 16:00.1
```

Rerun `./usertools/dpdk-devbind.py --status-dev net` to check that FPGA and FVL PF's are bound to `vfio-pci` driver.

Figure 11. Sample Output



```
Network devices using DPDK-compatible driver
============================================
0000:14:00.0 'Device 0d58' drv=vfio-pci unused=
0000:14:00.1 'Device 0d58' drv=vfio-pci unused=
0000:15:00.0 'Device 0b30' drv=vfio-pci unused=
0000:16:00.0 'Device 0d58' drv=vfio-pci unused=
0000:16:00.1 'Device 0d58' drv=vfio-pci unused=
```

d. Reserve hugepages:

```
$ sudo mkdir -p /mnt/huge
```

```
$ sudo mount -t hugetlbfs nodev /mnt/huge
```

```
$ sudo sh -c "echo 2048 > /sys/kernel/mm/hugepages/\
hugepages-2048kB/nr_hugepages"
```

For more information on how the FPGA support is enabled in DPDK, refer to the Data Plane Development Kit Reference Manual.

11.1. Test Using an External Traffic Generator

This test can be performed on 2x2x25G and 4x25G configuration.

1. Hardware Setup: Connect cable between QSFP port of Intel FPGA PAC N3000-N/2 and external traffic generator.

   Note: In 4x25G configuration, only one QSFP port is active per the configuration.

Figure 12. External Tester with 2x2x25G Network Configuration

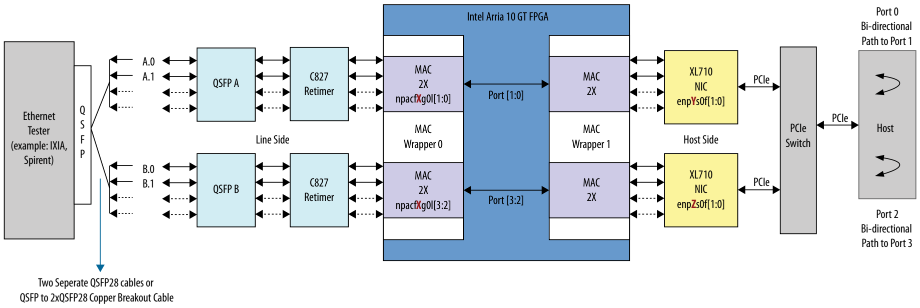
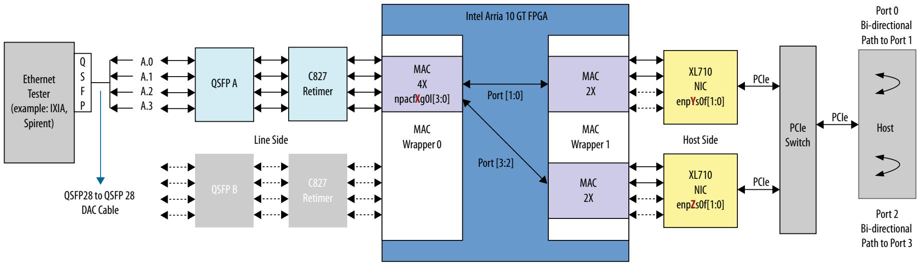


Figure 13. External Tester with 4x25G Network Configuration



2. Start the DPDK testpmd application.

   Note: Replace the FPGA B:D.F with values specific to your system.

```
$    cd $RTE_SDK
```

```
$    sudo ./x86_64-native-linuxapp-gcc/app/testpmd -l 0,1,2,3,4,5,6,7 -n 4 \
--vdev 'ifpga_rawdev_cfg0,ifpga=15:00.0,port=0' -- -i --no-numa
```

   Now, start traffic from external traffic generator:

```
testpmd>    start
```

```
testpmd>    show port stats all
```

   Sample output for 2x2x25G configuration:

```
  ######################## NIC statistics for port 0
########################
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 0          TX-errors: 0          TX-bytes:  0

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:            0

############################################################################

  ######################## NIC statistics for port 1
########################
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 0          TX-errors: 0          TX-bytes:  0

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:            0

############################################################################

  ######################## NIC statistics for port 2
########################
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 0          TX-errors: 0          TX-bytes:  0

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:            0

############################################################################

  ######################## NIC statistics for port 3
########################
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 0          TX-errors: 0          TX-bytes:  0

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:            0

############################################################################

  ######################## NIC statistics for port 4
########################
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 0          TX-errors: 0          TX-bytes:  0

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:            0

############################################################################

  ######################## NIC statistics for port 5
########################
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 0          TX-errors: 0          TX-bytes:  0

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:            0

############################################################################

  ######################## NIC statistics for port 6
########################
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 0          TX-errors: 0          TX-bytes:  0

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:            0

############################################################################

  ######################## NIC statistics for port 7
########################
```

```
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 0          TX-errors: 0          TX-bytes:  0

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:            0

############################################################################
```

Expected result: testpmd by default works in paired mode. In this mode, the packet forwarding is between pairs of ports, for example: (0,1), (2,3).

- In 2x2x25G configuration: Traffic forwarding is between ports (0,1) and (2,3)

List of the cores to run on:

In case of 2x2x25G or 4x25G, we have 4 XL710 ports and hence we assign 4 cores:

```
-l <core list>
```

Optionally, you can choose to send traffic to specific ports of the NIC rather than all ports. For this, the ports have to be explicitly white listed using the `-w <XL710 Port BDF>`. For example: Below command shows that only XL710 ports 14:00.0,14:00.1 are white listed. The FPGA BDF must also be explicitly white listed in this case:

```
sudo ./x86_64-native-linuxapp-gcc/app/testpmd -l 1,3 -n 4 -w \
0000:14:00.0 -w \
0000:14:00.1 -w \
0000:15:00.0 --vdev 'ifpga_rawdev_cfg0,ifpga=15:00.0,port=0' \
-- -i --no-numa
```

Refer to the Testpmd Application User Guide to understand the Environment Abstraction Layer arguments to testpmd.

Table 9. Specific Arguments

Arguments	Description	Capability Enabled
`-w <"FPGA BDF"> --vdev 'ifpga_rawdev_cfg0,ifpga=<"FPGA BDF">,port=0’`	The AFU name format is `FPGA BDF|Port`. Each FPGA can be divided into four blocks at most. `Port` identifies which FPGA block the AFU bitstream belongs to, but currently only Port 0 (Ethernet) is supported.	This triggers rawdev PMD driver to hotplug AFU to the IFPGA BUS.
`-w <XL710 port BDF>`	Whitelists the Intel XL710 NIC PF.	

11.2. Revert Back from DPDK to OPAE

In the following command, replace 15:00.0 with the appropriate B:D:F value that corresponds to the FPGA:

```
$ sudo rmmod vfio-pci
```

```
$ echo 0000:15:00.0 | sudo tee /sys/bus/pci/drivers/intel-fpga-pci/bind
```

```
$ sudo modprobe i40e
```

Bind the XL710 interfaces to i40e driver:

```
$ sudo ./usertools/dpdk-devbind.py -b i40e 14:00.0 \
14:00.1 16:00.0 16:00.1
```

12. Graceful Shutdown

12.1. Using OPAE

The fpgad is a service that can help you protect the server from crashing when the hardware reaches an upper non-recoverable or lower non-recoverable sensor threshold (also called as fatal threshold). The fpgad is capable of monitoring each of the 20 sensors reported by the Board Management Controller.

```
$ sudo fpgainfo bmc
```

The Intel FPGA PAC N3000-N/2 provides protective circuitry that automatically shuts down key board power supplies in the event of critical board sensors surpassing the fatal thresholds. The critical board sensors are listed below:

Table 10. Critical Sensors

Sensor ID	Sensor	Upper Fatal Threshold	Upper Warning Threshold	Lower Fatal Threshold	Lower Warning Threshold
12	FPGA Core Temperature	100°C	90°C	X	X
13	Board Temperature	100°C	85°C	X	X
25	12V Aux Voltage	X	X	10.56 V	11.40 V
3	12 V Backplane Voltage	X	X	10.56 V	11.40 V

For more information about sensors, refer to the *Board Management Controller User Guide: Intel FPGA Programmable Acceleration Card N3000-N/2*.

Note: Qualified OEM server systems should provide the required cooling for your workloads. Therefore, using fpgad may be optional.

When the opae-tools-extra-1.3.7-5.x86_64.rpm package is installed, fpgad is placed in the OPAE binaries directory (default: /usr/bin). The configuration file fpgad.cfg is located at /etc/opae. The log file fpgad.log which monitors fpgad actions is located at /var/lib/opae/.

The fpgad periodically reads the sensor values and if the values exceed the warning threshold stated in the fpgad.conf or the hardware defined warning threshold, it masks the PCIe Advanced Error Reporting (AER) registers for the Intel FPGA PAC to avoid system reset.

Use the following command to start the fpgad service:

```
$ sudo systemctl start fpgad
```

The configuration file only includes the threshold setting for critical sensor 12 V Aux Voltage (sensor 25) and 12 V Backplane Voltage (sensor 3). These sensors do not have a hardware defined warning threshold and hence fpgad relies on the configuration file. The other two critical sensor FPGA Core Temperature (sensor 12) and Board Temperature (sensor 13) have a hardware defined warning threshold and fatal threshold set to values mentioned in the above table. The `fpgad` uses this information to mask the PCIe AER register when the sensor reaches the warning threshold.

Snapshot of the `fpgad.cfg` file located at `/etc/opae/` which configures the sensor 12 V Aux Voltage (sensor 25) is shown below:

```
"fpgad-vc": {
                        "configuration": {
                                "cool-down": 30,
                                "config-sensors-enabled": true,
                                "sensors": [
                                        {
                                                "id": 25,
                                                "low-warn": 11.40,
                                                "low-fatal": 10.56
                                        },
                                ]
                        },
                        "enabled": true,
                        "plugin": "libfpgad-vc.so",
                        "devices": [
                                [ "0x8086", "0x0b30" ],
                                [ "0x8086", "0x0b31" ]
                        ]
                }
```

You must create another entry below the 12 V Aux Voltage entry for 12 V Backplane Voltage (sensor 3). The updated configuration file should have the following entry:

```
"fpgad-vc": {
                        "configuration": {
                                "cool-down": 30,
                                "config-sensors-enabled": true,
                                "sensors": [
                                        {
                                                "id": 25,
                                                "low-warn": 11.40,
                                                "low-fatal": 10.56
                                        }
                                ]
                        },
                        "enabled": true,
                        "plugin": "libfpgad-vc.so",
                        "devices": [
                                [ "0x8086", "0x0b30" ],
                                [ "0x8086", "0x0b31" ]
                        ]
                },

"fpgad-vc": {
                        "configuration": {
                                "cool-down": 30,
                                "config-sensors-enabled": true,
                                "sensors": [
                                        {
                                                "id": 3,
                                                "low-warn": 11.40,
                                                "low-fatal": 10.56
                                        }
                                ]
                        },
                        "enabled": true,
                        "plugin": "libfpgad-vc.so",
                        "devices": [
                                [ "0x8086", "0x0b30" ],
                                [ "0x8086", "0x0b31" ]
                        ]
                }
```

You can monitor the log file to see if upper or lower warning threshold levels are hit. For example:

```
tail -f /var/lib/opae/fpgad.log | grep “sensor.*warning”
fpgad-vc: sensor 'FPGA Die Temperature' warning
```

You must take appropriate action to recover from this warning before the sensor value reaches upper or lower fatal limits. On reaching the warning threshold limit, the daemon masks the AER registers and the log file will indicate that the sensor is tripped.

Sample output: Warning message when the FPGA Core Temperature exceeds the upper warning threshold limit.

```
Ex: tail -f /var/lib/opae/fpgad.log
fpgad-vc: saving previous ECAP_AER+0x08 value 0x003ff030 for 0000:5d:00.0
fpgad-vc: saving previous ECAP_AER+0x14 value 0x000031c1 for 0000:5d:00.0
fpgad-vc: sensor 'FPGA Die Temperature' still tripped.
```

Sample output: Warning message when the voltage exceeds the lower warning threshold limit.:

```
fpgad-vc: sensor '12V AUX Voltage' warning.
fpgad-vc: saving previous ECAP_AER+0x08 value 0x00100000 for 0000:ae:00.0
fpgad-vc: saving previous ECAP_AER+0x14 value 0x00002000 for 0000:ae:00.0
fpgad-vc: sensor '12V AUX Voltage' still tripped.
fpgad-vc: sensor '12V AUX Voltage' still tripped.
```

If the upper or lower fatal threshold limit is reached, then a power cycle of server is required to recover the Intel FPGA PAC N3000-N/2.

AER is unmasked by the `fpgad` after the sensor values are within the normal range which is above the lower warning or below the upper warning threshold.

Sample output when upper or lower fatal threshold is reached:

```
fpgad-vc: failed to read sensor xx
```

To stop `fpgad`:

```
$ sudo systemctl stop fpgad.service
```

To check status of `fpgad`:

```
$ sudo systemctl status fpgad.service
```

Optional: To enable `fpgad` to re-start on boot, execute

```
$ sudo systemctl enable fpgad.service
```

For a full list of `systemctl` commands, run the following command:

```
$ systemctl -h
```

13. Single Event Upset (SEU)

The Intel Manufacturing Single Event Upset (SEU) testing of Intel FPGA PAC N3000-N/2 provides the following results:

- SEU events do not induce latch-up in Intel FPGA PAC N3000-N/2.
- No SEU errors have been observed in hard CRC circuits and I/O registers.
- The cyclic redundancy check (CRC) circuit can detect all single-bit and multi-bit errors within the configuration memory.

SEU errors may occur in either of the two primary devices of the Intel FPGA PAC N3000-N/2:

- **Intel MAX 10 SEU**: An SEU event is detected by Error Detection CRC (EDCRC) circuitry in Intel MAX 10. The CRC function implemented in Intel FPGA PAC N3000-N/2 enables CRC status to be reported to FPGA via a dedicated `CRC_ERROR` pin. The CRC error output is continually polled in an interval between 5.5 and 13.6 seconds.

  If a CRC error was detected, it is not permanently logged if subsequent polls do not detect an error.

  When FPGA detects a `CRC_ERROR` assertion, it is logged in the FPGA internal register RAS_CATFAT_ERR. The system register bits are not reliable after an SEU event. To recover, you must power cycle the server.
- **FPGA SEU**: In FPGA device, the contents of the configuration RAM (CRAM) bits can be affected by soft SEU errors. The hardened on-chip EDCRC circuitry auto-detects CRC errors. Corrections of CRAM upsets are not supported. Therefore, if SEU errors are detected, you must power cycle the server.

13.1. OPAE Handling of SEU

An OPAE tool `fpgad` monitors for SEU events and records any such occurrence in the log file `/var/lib/opae/fpgad.log`

To start `fpgad`:

```
sudo systemctl start fpgad
```

- Intel MAX 10 SEU:

  The `fpgad.log` file would show the below output:

  ```
  tail -f /var/lib/opae/fpgad.log
  fpgad-vc: failed to get value object for sensor38.
  fpgad-vc: poll count = 1
  ```

```
fpgad-vc: SEU error occurred on bmc @ 0000:b2:00.0
fpgad-vc: failed to get value object for sensor15.
fpgad-vc: failed to get value object for sensor38.
```

Ignore the message: *failed to get value object for sensor*. Sensor 15 and sensor 38 indicate QSFP temperature. This failure indicates that the QSFP cable was not plugged in.

- FPGA SEU:

  The `fpgad.log` file would show the below output:

  ```
  tail -f /var/lib/opae/fpgad.log
  fpgad-vc: failed to get value object for sensor38.
  fpgad-vc: poll count = 1
  fpgad-vc: SEU error occurred on fpga @ 0000:b2:00.0
  fpgad-vc: failed to get value object for sensor15.
  fpgad-vc: failed to get value object for sensor38.
  ```

  Ignore the message: *failed to get value object for sensor*. Sensor 15 and sensor 38 indicate QSFP temperature. This failure indicates that the QSFP cable was not plugged in.

To recover from both Intel MAX 10 and FPGA SEU event, reset the Intel FPGA PAC N3000-N/2 using the following command:

```
$ rsu bmcimg [PCIe B:D.F]
```

For testing your system's response to an SEU event, Intel provides a mechanism to inject an error which will be logged by `fpgad` similar to the way an SEU event is logged.

1. Start `fpgad`

   ```
   $ sudo systemctl start fpgad
   ```

2. Terminal 2: monitor fpgad.log

   ```
   $ sudo tail -f /var/lib/opae/fpgad.log
   ```

3. Terminal 1: Inject error

   ```
   $ sudo sh -c "echo 1 > /sys/class/fpga/intel-fpga-dev.0/\
   intel-fpga-fme.0/errors/inject_error"
   ```

   Sample output:

   ```
   fpgad-vc: error interrupt event received.
   fpgad-vc: poll count = 1.
   fpgad-vc: detect inject_error 0x1 @ 0000:15:00.0
   fpgad-vc: detect catfatal_errors 0x800 @ 0000:15:00.0
   ```

   Note: `poll count` =1: indicates an error was detected.

4. To clear the error injection:

   ```
   $ sudo sh -c "echo 0 > /sys/class/fpga/intel-fpga-dev.0/intel-fpga-fme.0/
   errors/inject_error"
   ```

14. Document Revision History for Intel Acceleration Stack User Guide: Intel FPGA PAC N3000-N/2

Document Version	Intel Acceleration Stack Version	Changes
2021.11.01	1.3.1	Added a note about the RSU functionality in section: *Installing the Intel FPGA PAC N3000-N/2*.
2021.05.21	1.3.1	Updated steps in section: *For RHEL*.
2021.01.20	1.3.1	Corrected steps in the following sections: • *For RHEL* • *Download kernel 4.19* • *Build and Install kernel 4.19* • *Install the Acceleration Stack for Development*
2020.11.16	1.3.1	Corrected the Intel MAX 10 NIOS Firmware (FW) version for Intel FPGA PAC N3000 (BD-NVV-N3000-2) in section: *Identify the Intel MAX 10 BMC Version*.
2020.09.28	1.3.1	The Intel Acceleration Stack v1.3.1 also supports the following variant of the Intel FPGA PAC N3000: • BD-NVV-N3000-2 .
2020.09.08	1.3.1	Updated the following sections: • *System Requirements* • *Install the Release Package* • *Intel XL710 Driver Installation and Firmware Update* Added a new section: • *Testing Network Loopback Using Data Plane Development Kit (DPDK)*
2020.06.15	1.3	Initial release.

A. Troubleshooting

A.1. If OPAE installation verification fails, how to install OPAE manually?

Before the OPAE installation, the dependency packages must be installed.

Table 11. Dependency Packages

Distribution	OPAE Package	External Dependency Packages
Centos 7.6	opae-libs	uuid json-c
	opae-devel	libuuid-devel
	opae-tools-extra	hwloc-libs
	opae.admin	pciutils python3 sysvinit-tools
	opae-intel-fpga-driver	gcc make dkms kernel-devel kernel-headers
	Compile OPAE SDK source code	gcc-c++ cmake libxml2-devel json-c-devel hwloc-devel libpng12 rsync python3-devel python3-libs python3-sphinx python3-pip python3
RHEL 8.2	opae-libs	uuid json-c
	opae-devel	libuuid-devel
	opae-tools-extra	hwloc-libs
	opae.admin	pciutils python3
	opae-intel-fpga-driver	gcc make dkms kernel-devel kernel-headers
	Compile OPAE SDK source code	gcc-c++ cmake libxml2-devel json-c-devel hwloc-devel libpng12 rsync python3-devel python3-libs python3-sphinx python3-pip python3

1. Download the Acceleration Stack runtime (rte) installer.
2. Extract the OPAE packages:

```
$ ./n3000-1.3.8-*-setup.sh extract
```

```
$ cp opae-intel-fpga*.rpm n3000-1.3.8-rte/opae/
```

```
$ cd n3000-1.3.8-rte/opae/
```

3. Remove any previously installed OPAE:

```
$ sudo yum remove opae*
```

4. Manually install OPAE driver:

```
$ sudo yum install opae-intel-fpga*.rpm
```

5. Verify installation:

```
$ lsmod | grep fpga
```

```
ifpga_sec_mgr 16384 1 intel_max10
intel_fpga_fme 65536 0
intel_fpga_afu 32768 0
fpga_mgr_mod 16384 1 intel_fpga_fme
intel_fpga_pci 24576 2 intel_fpga_fme,intel_fpga_afu
```

   If the verification fails, analyze the message log of kernel installation for hints to fix the issue.

6. Manually install OPAE libraries and tools:

```
$ sudo yum install opae*.rpm
```

7. Verify installation of OPAE libraries and tools:

```
$ rpm -qa | grep opae

opae.admin-1.0.3-2.el8.noarch
opae-devel-1.3.7-5.el8.x86_64
opae-intel-fpga-driver-2.0.1-10.x86_64
opae-tools-1.3.7-5.el8.x86_64
opae.pac_sign-1.0.4-3.el8.x86_64
opae-tools-extra-1.3.7-5.el8.x86_64
opae-libs-1.3.7-5.el8.x86_64
```

B. fpgabist Sample Output

Related Information

Test PCIe and External Memories with fpgabist on page 35

B.1. For 4x25G Configuration

```
==========================================================

Beginning FPGA Built-In Self-Test

==========================================================
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** FME ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
Boot Page                     : user


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PORT ******//
Object Id                     : 0xF000000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
Accelerator Id                : 9aeffe5f-8457-0612-c000-c9660d824272


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** TEMP ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
(12) FPGA Core Temperature        : 46.00 Celsius
(13) Board Temperature            : 30.00 Celsius
(15) QSFP A Temperature           : N/A
(38) QSFP B Temperature           : N/A
(44) Retimer A Core Temperature   : 54.50 Celsius
(45) Retimer A Serdes Temperature : 55.50 Celsius
```

```
(46) Retimer B Core Temperature   : 0.00 Celsius
(47) Retimer B Serdes Temperature : 0.00 Celsius


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** POWER ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
( 1) Board Power              : 67.57 Watts
( 2) 12V Backplane Current    : 3.10 Amps
( 3) 12V Backplane Voltage    : 12.10 Volts
( 4) 1.2V Voltage             : 1.19 Volts
( 6) 1.8V Voltage             : 1.80 Volts
( 8) 3.3V Voltage             : 3.26 Volts
(10) FPGA Core Voltage        : 0.90 Volts
(11) FPGA Core Current        : 14.31 Amps
(14) QSFP A Voltage           : N/A
(24) 12V AUX Current          : 2.49 Amps
(25) 12V AUX Voltage          : 12.10 Volts
(37) QSFP B Voltage           : N/A


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** FME ERRORS ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
PCIe0 Errors                  : 0x0
PCIe1 Errors                  : 0x0
Catfatal Errors               : 0x0
Seu Emr                       : 0x0
Inject Error                  : 0x0
Nonfatal Errors               : 0x0
Next Error                    : 0x0
First Error                   : 0x0
Errors                        : 0x0
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PORT ERRORS ******//
Object Id                     : 0xF000000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
Accelerator Id                : 9aeffe5f-8457-0612-c000-c9660d824272
First Malformed Req           : 0x0
First Error                   : 0x0
Errors                        : 0x0


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PHY ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3e:00.0
```

```
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
//****** PHY GROUP 0 ******//
Direction                     : Line side
Speed                         : 25 Gbps
Number of PHYs                : 4
//****** PHY GROUP 1 ******//
Direction                     : Host side
Speed                         : 40 Gbps
Number of PHYs                : 4
//****** Intel C827 Retimer ******//
Port0 25G                     : Up
Port1 25G                     : Up
Port2 25G                     : Up
Port3 25G                     : Up
Retimer A Version             : 101c.1064
Retimer B Version             : 0000.0000


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** MAC ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3e:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000110010310
Bitstream Version             : 0.2.3
Pr Interface Id               : f3c99413-5081-4aad-bced-07eb84a6d0bb
Number of MACs                : 8
MAC address 0                 : 64:4C:36:12:9D:D0
MAC address 1                 : 64:4C:36:12:9D:D1
MAC address 2                 : 64:4C:36:12:9D:D2
MAC address 3                 : 64:4C:36:12:9D:D3
MAC address 4                 : 64:4C:36:12:9D:D4
MAC address 5                 : 64:4C:36:12:9D:D5
MAC address 6                 : 64:4C:36:12:9D:D6
MAC address 7                 : 64:4C:36:12:9D:D7


Running mode: nlb
Running fpgadiag lpbk1 vh0-vh0 test...
found the NLB offset=0x28000

Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss   Eviction 'Clocks(@200 MHz)'   Rd_Bandwidth   Wr_Bandwidth
      1024   97681728    97680872            0            0
0             0          0        200032043     6.251 GB/s     6.251 GB/s

VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count VL0_Wr_Count
    97681740     97680873            0            0            0            0

Running fpgadiag lpbk1 vh0-vh1 test...
found the NLB offset=0x28000

Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss   Eviction 'Clocks(@200 MHz)'   Rd_Bandwidth   Wr_Bandwidth
      1024   97694604    97693772            0            0
0             0          0        200030128     6.252 GB/s     6.251 GB/s

VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count VL0_Wr_Count
    97694616     97693773            0            0            0            0

Running fpgadiag lpbk1 vh1-vh0 test...
found the NLB offset=0x28000
```

```
Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss   Eviction 'Clocks(@200 MHz)'  Rd_Bandwidth   Wr_Bandwidth
      1024   97676772    97675900            0            0
0             0          0        200031350     6.250 GB/s     6.250 GB/s

VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count VL0_Wr_Count
    97676772     97675901            0            0            0            0

Running fpgadiag lpbk1 vh1-vh1 test...
found the NLB offset=0x28000

Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss   Eviction 'Clocks(@200 MHz)'  Rd_Bandwidth   Wr_Bandwidth
      1024   97687552    97686640            0            0
0             0          0        200031698     6.251 GB/s     6.251 GB/s

VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count VL0_Wr_Count
    97687552     97686641            0            0            0            0

Finished Executing NLB (FPGA DIAG) Tests

Running mode: dma_afu
Running fpga_dma_test test on DDR4_A...

Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Buffer pointer = 0x7fb65c5fc000, size = 0x100000000 (0x7fb65c5fc000 through
0x7fb75c5fc000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6028.495598 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6238.024430 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
DDR sweep with unaligned pointer and size
Buffer pointer = 0x7fb65cffd03d, size = 0xffffffbe (0x7fb65cffd03d through
0x7fb75cffcffb)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6023.680411 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6227.207827 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fb65cffd003, size = 0xfffffffd (0x7fb65cffd003 through
0x7fb75cffd000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6040.148285 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6278.841591 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fb65cffd007, size = 0xfffffff6 (0x7fb65cffd007 through
0x7fb75cffcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6075.907357 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6193.636128 Megabytes/sec
```

```
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fb65cffd000, size = 0xfffffffd (0x7fb65cffd000 through
0x7fb75cffcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6075.847398 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6286.483893 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fb65cffd000, size = 0xffffffc3 (0x7fb65cffd000 through
0x7fb75cffcfc3)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6077.481368 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6239.577290 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fb65cffd000, size = 0xfffffff9 (0x7fb65cffd000 through
0x7fb75cffcff9)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6075.056363 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6282.301315 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Running fpga_dma_test test on DDR4_B...

Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Buffer pointer = 0x7fe2a87fc000, size = 0x100000000 (0x7fe2a87fc000 through
0x7fe3a87fc000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6075.915975 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6354.786148 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
DDR sweep with unaligned pointer and size
Buffer pointer = 0x7fe2a91fd03d, size = 0xffffffbe (0x7fe2a91fd03d through
0x7fe3a91fcffb)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6008.463097 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6317.263668 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fe2a91fd003, size = 0xfffffffd (0x7fe2a91fd003 through
0x7fe3a91fd000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6077.102232 Megabytes/sec
Clear buffer
```

```
DDR Sweep FPGA to Host
Measured bandwidth = 6295.423917 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fe2a91fd007, size = 0xfffffff6 (0x7fe2a91fd007 through
0x7fe3a91fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6076.641461 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6308.306961 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fe2a91fd000, size = 0xfffffffd (0x7fe2a91fd000 through
0x7fe3a91fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6078.123251 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6403.779597 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fe2a91fd000, size = 0xffffffc3 (0x7fe2a91fd000 through
0x7fe3a91fcfc3)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6077.587999 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6442.061446 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fe2a91fd000, size = 0xfffffff9 (0x7fe2a91fd000 through
0x7fe3a91fcff9)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6076.830038 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6434.268346 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Running fpga_dma_test test on DDR4_C...

Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Buffer pointer = 0x7f9f08bfc000, size = 0x40000000 (0x7f9f08bfc000 through
0x7f9f48bfc000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2359.378051 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2441.195532 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
DDR sweep with unaligned pointer and size
Buffer pointer = 0x7f9f095fd03d, size = 0x3fffffbe (0x7f9f095fd03d through
0x7f9f495fcffb)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
```

```
Measured bandwidth = 2358.980944 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2439.859968 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f9f095fd003, size = 0x3ffffffd (0x7f9f095fd003 through
0x7f9f495fd000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2359.307560 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2440.822373 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f9f095fd007, size = 0x3ffffff6 (0x7f9f095fd007 through
0x7f9f495fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2359.178753 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2440.677334 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f9f095fd000, size = 0x3ffffffd (0x7f9f095fd000 through
0x7f9f495fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2359.269882 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2440.602513 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f9f095fd000, size = 0x3fffffc3 (0x7f9f095fd000 through
0x7f9f495fcfc3)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2359.528387 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2441.371818 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f9f095fd000, size = 0x3ffffff9 (0x7f9f095fd000 through
0x7f9f495fcff9)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2359.232654 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2399.931406 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Running fpga_dma_test test on QDR...

Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Buffer pointer = 0x7f2de37fc000, size = 0x1000000 (0x7f2de37fc000 through
0x7f2de47fc000)
Allocated test buffer
Fill test buffer
```

```
DDR Sweep Host to FPGA
Measured bandwidth = 933.236849 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 913.672934 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Finished Executing DMA Tests


Built-in Self-Test Completed.
```

B.2. For 2x2x25G Configuration

```
==========================================================

Beginning FPGA Built-In Self-Test

==========================================================
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** FME ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3f:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
Boot Page                     : user


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PORT ******//
Object Id                     : 0xF000000
PCIe s:b:d.f                  : 0000:3f:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
Accelerator Id                : 9aeffe5f-8457-0612-c000-c9660d824272


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** TEMP ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3f:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
(12) FPGA Core Temperature        : 45.50 Celsius
(13) Board Temperature            : 29.50 Celsius
(15) QSFP A Temperature           : N/A
(38) QSFP B Temperature           : N/A
(44) Retimer A Core Temperature   : 47.50 Celsius
(45) Retimer A Serdes Temperature : 47.50 Celsius
(46) Retimer B Core Temperature   : 46.50 Celsius
(47) Retimer B Serdes Temperature : 46.50 Celsius
```

```
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** POWER ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3f:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
( 1) Board Power              : 66.06 Watts
( 2) 12V Backplane Current    : 3.03 Amps
( 3) 12V Backplane Voltage    : 12.11 Volts
( 4) 1.2V Voltage             : 1.19 Volts
( 6) 1.8V Voltage             : 1.80 Volts
( 8) 3.3V Voltage             : 3.26 Volts
(10) FPGA Core Voltage        : 0.90 Volts
(11) FPGA Core Current        : 13.86 Amps
(14) QSFP A Voltage           : N/A
(24) 12V AUX Current          : 2.43 Amps
(25) 12V AUX Voltage          : 12.11 Volts
(37) QSFP B Voltage           : N/A


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** FME ERRORS ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3f:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
PCIe0 Errors                  : 0x0
PCIe1 Errors                  : 0x0
Catfatal Errors               : 0x0
Seu Emr                       : 0x0
Inject Error                  : 0x0
Nonfatal Errors               : 0x0
Next Error                    : 0x0
First Error                   : 0x0
Errors                        : 0x0
Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PORT ERRORS ******//
Object Id                     : 0xF000000
PCIe s:b:d.f                  : 0000:3f:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
Accelerator Id                : 9aeffe5f-8457-0612-c000-c9660d824272
First Malformed Req           : 0x0
First Error                   : 0x0
Errors                        : 0x0


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** PHY ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3f:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
//****** PHY GROUP 0 ******//
Direction                     : Line side
Speed                         : 25 Gbps
Number of PHYs                : 4
//****** PHY GROUP 1 ******//
Direction                     : Host side
Speed                         : 40 Gbps
Number of PHYs                : 4
//****** Intel C827 Retimer ******//
Port0 25G                     : Up
Port1 25G                     : Up
Port2 25G                     : Down
Port3 25G                     : Down
Retimer A Version             : 101c.1064
Retimer B Version             : 101c.1064


Board Management Controller, MAX10 NIOS FW version D.2.1.24
Board Management Controller, MAX10 Build version D.2.0.7
//****** MAC ******//
Object Id                     : 0xF100000
PCIe s:b:d.f                  : 0000:3f:00.0
Device Id                     : 0x0b30
Numa Node                     : 0
Ports Num                     : 01
Bitstream Id                  : 0x23000410010310
Bitstream Version             : 0.2.3
Pr Interface Id               : a5d72a3c-c8b0-4939-912c-f715e5dc10ca
Number of MACs                : 8
MAC address 0                 : 64:4C:36:12:9D:D0
MAC address 1                 : 64:4C:36:12:9D:D1
MAC address 2                 : 64:4C:36:12:9D:D2
MAC address 3                 : 64:4C:36:12:9D:D3
MAC address 4                 : 64:4C:36:12:9D:D4
MAC address 5                 : 64:4C:36:12:9D:D5
MAC address 6                 : 64:4C:36:12:9D:D6
MAC address 7                 : 64:4C:36:12:9D:D7


Running mode: nlb
Running fpgadiag lpbk1 vh0-vh0 test...
found the NLB offset=0x28000

Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss   Eviction 'Clocks(@200 MHz)'   Rd_Bandwidth   Wr_Bandwidth
      1024   97702252    97701372            0            0
0             0          0        200032256     6.252 GB/s     6.252 GB/s

VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count VL0_Wr_Count
    97702252     97701373            0            0            0            0

Running fpgadiag lpbk1 vh0-vh1 test...
found the NLB offset=0x28000

Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss   Eviction 'Clocks(@200 MHz)'   Rd_Bandwidth   Wr_Bandwidth
      1024   97707972    97707108            0            0
0             0          0        200032174     6.252 GB/s     6.252 GB/s

VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count VL0_Wr_Count
    97707972     97707109            0            0            0            0

Running fpgadiag lpbk1 vh1-vh0 test...
found the NLB offset=0x28000

Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss   Eviction 'Clocks(@200 MHz)'   Rd_Bandwidth   Wr_Bandwidth
      1024   97738752    97737964            0            0
0             0          0        200031550     6.254 GB/s     6.254 GB/s
```

```
VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count VL0_Wr_Count
    97738752     97737965            0            0            0            0

Running fpgadiag lpbk1 vh1-vh1 test...
found the NLB offset=0x28000

Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss   Eviction 'Clocks(@200 MHz)'   Rd_Bandwidth   Wr_Bandwidth
      1024   97721588    97720760            0            0
0             0          0        200031448     6.253 GB/s     6.253 GB/s

VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count VL0_Wr_Count
    97721588     97720761            0            0            0            0

Finished Executing NLB (FPGA DIAG) Tests

Running mode: dma_afu
Running fpga_dma_test test on DDR4_A...

Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Buffer pointer = 0x7fce887fc000, size = 0x100000000 (0x7fce887fc000 through
0x7fcf887fc000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5600.550050 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5592.111580 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
DDR sweep with unaligned pointer and size
Buffer pointer = 0x7fce891fd03d, size = 0xffffffbe (0x7fce891fd03d through
0x7fcf891fcffb)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5608.426199 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5523.782897 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fce891fd003, size = 0xfffffffd (0x7fce891fd003 through
0x7fcf891fd000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5575.882161 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5496.642558 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fce891fd007, size = 0xfffffff6 (0x7fce891fd007 through
0x7fcf891fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5648.824812 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5521.200156 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fce891fd000, size = 0xfffffffd (0x7fce891fd000 through
0x7fcf891fcffd)
```

```
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5609.697316 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5630.765624 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fce891fd000, size = 0xffffffc3 (0x7fce891fd000 through
0x7fcf891fcfc3)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5612.841992 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5586.903681 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7fce891fd000, size = 0xfffffff9 (0x7fce891fd000 through
0x7fcf891fcff9)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5651.709522 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5633.353851 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Running fpga_dma_test test on DDR4_B...

Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Buffer pointer = 0x7f17d3bfc000, size = 0x100000000 (0x7f17d3bfc000 through
0x7f18d3bfc000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5612.720106 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5694.609999 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
DDR sweep with unaligned pointer and size
Buffer pointer = 0x7f17d45fd03d, size = 0xffffffbe (0x7f17d45fd03d through
0x7f18d45fcffb)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5606.467502 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5648.229561 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f17d45fd003, size = 0xfffffffd (0x7f17d45fd003 through
0x7f18d45fd000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5615.607668 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5612.448868 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f17d45fd007, size = 0xfffffff6 (0x7f17d45fd007 through
0x7f18d45fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5649.737109 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5672.415577 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f17d45fd000, size = 0xfffffffd (0x7f17d45fd000 through
0x7f18d45fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5624.444398 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5618.423113 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f17d45fd000, size = 0xffffffc3 (0x7f17d45fd000 through
0x7f18d45fcfc3)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5622.371837 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5659.481637 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7f17d45fd000, size = 0xfffffff9 (0x7f17d45fd000 through
0x7f18d45fcff9)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5606.902116 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 5660.070136 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Running fpga_dma_test test on DDR4_C...

Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Buffer pointer = 0x7ff68effc000, size = 0x40000000 (0x7ff68effc000 through
0x7ff6ceffc000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2358.764387 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2439.997796 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
DDR sweep with unaligned pointer and size
Buffer pointer = 0x7ff68f9fd03d, size = 0x3fffffbe (0x7ff68f9fd03d through
0x7ff6cf9fcffb)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2358.690375 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2439.482419 Megabytes/sec
```

```
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7ff68f9fd003, size = 0x3ffffffd (0x7ff68f9fd003 through
0x7ff6cf9fd000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2358.861401 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2439.375992 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7ff68f9fd007, size = 0x3ffffff6 (0x7ff68f9fd007 through
0x7ff6cf9fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2359.751621 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2438.391193 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7ff68f9fd000, size = 0x3ffffffd (0x7ff68f9fd000 through
0x7ff6cf9fcffd)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2350.635176 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2440.036198 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7ff68f9fd000, size = 0x3fffffc3 (0x7ff68f9fd000 through
0x7ff6cf9fcfc3)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2353.521378 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2439.492240 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Buffer pointer = 0x7ff68f9fd000, size = 0x3ffffff9 (0x7ff68f9fd000 through
0x7ff6cf9fcff9)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 2350.056518 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 2439.847035 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Running fpga_dma_test test on QDR...

Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Buffer pointer = 0x7f80ee3fc000, size = 0x1000000 (0x7f80ee3fc000 through
0x7f80ef3fc000)
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 932.759871 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
```

```
Measured bandwidth = 917.245179 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
Finished Executing DMA Tests


Built-in Self-Test Completed.
```