Release Notes
CCRTNGFC (WC-CP-FIO2)

Driver	ccrtngfc (WC-CP-FIO2)	
Platform	RedHawk Linux® *(CentOS/Rocky/RHEL & Ubuntu)*, Native Ubuntu® and Native Red Hat Enterprise Linux®[1]	
Vendor	Concurrent Real-Time	
Hardware	PCIe Next Generation FPGA I/O Card (CP-FPGA-4 & 5)	
Author	Darius Dubash	
Date	November 10th, 2025	Rev 2025.2

[1] All trademarks are the property of their respective owners

This page intentionally left blank

Table of Contents

1. Introduction

This document assists the user in installing the CCRT-PCIe-NGFC Linux ***ccrtngfc*** driver and related software on the RedHawk OS, Native Ubuntu and Native Red Hat for use with the Concurrent Real-Time PCI Express Next Generation FPGA I/O Card (NGFC) and the Concurrent Real-Time PCI Express Single Daughter Card FPGA I/O Card (SDFC). The directions in this document supersede all others – they are specific to installing the software on Concurrent Real-Time's RedHawk and Native Ubuntu and Native Red Hat systems. Other information provided as part of this release, when it may contradict these directions, should be ignored and these directions should prevail.

Current versions of Native Operating Systems that are supported are:

1) Ubuntu 22.04, kernel 6.5 or 6.8, gcc-11 & gcc-12
2) Ubuntu 24.04, kernel 6.14, gcc-13
3) Red Hat RHEL 9.4, kernel 5.14 (minor 427.28.1 or less)
4) Red Hat RHEL 9.6, kernel 5.14 (minor 570.39.1 or less)

This release provides support for the new Cloning of data among peripheral components and main system *(CCRT US Patent US 11.281.584 B1, Inventor Darius Dubash)*.

For additional information on this driver and usage, refer to the ***ccrtngfc*** man page.

The NGFC is a Next Generation FPGA card with a PCI express interface.

Features and Characteristics of the ***NGFC (Device ID 9320)*** are:

- ❖ General
  - Intel Arria 10 FPGA Control
  - Six (6) MSG DMA Engines
  - Programmable Clock Generator
  - Temperature Compensated Oscillator (TCXO)
  - FPGA Temperature Sensing with Power Control
  - In System Firmware Update
  - PCI Express Gen 2 x4 Lane
  - MSI Interrupts
  - Industry Standard High-Density SCSI 68-pin Connectors
  - RJ-45 External Clock Connectors

- ❖ High Speed LVDS I/O Section
  - 32-channel LVDS Input/Output
  - Outputs Selectable per Nibble
  - Input Channel Snapshot
  - Output Channel Synchronization
  - Change Of State Sensing

- ❖ High Speed Digital I/O Section
  - 32-channel TTL Input/Output
  - 3.3V or 5V TTL Compatible Signals
  - Switchable 100 Ohm Termination
  - Input Channel Snapshot
  - Output Channel Synchronization
  - Change Of State Sensing

- ❖ Daughter Card Interface
  - Two FMC Style Positions (3" x 3.75")
  - 1.8V LPC Signaling

- HPC Signals for I/O Connectors
- DP Transceiver (4 Lanes & 8 Lanes)

Features and Characteristics of the ***SDFC (Device ID 9330)*** are:

❖ General
- Altera Cyclone V FPGA Control
- Six (6) MSG DMA Engines
- Programmable Clock Generator
- Temperature Compensated Oscillator (TCXO)
- In System Firmware Update
- PCI Express Gen 1 x4 Lane
- MSI Interrupts
- Industry Standard High-Density SCSI 68-pin Connectors
- RJ-45 External Clock Connectors

❖ High Speed Digital I/O Section
- 32-channel TTL Input/Output
- 3.3V/5V TTL Compatible Signals
- 32 Milliamp High Output Source
- 64 Milliamp Low Output Sink
- Switchable 100 Ohm Termination
- Outputs Selectable per Bit
- Input Channel Snapshot
- Output Channel Synchronization
- Change Of State Sensing

❖ Daughter Card Interface
- One FMC Style Positions (3” x 3.75”)
- 1.8V LPC Signaling
- HPC Signals for I/O Connectors
- DP Transceiver (4 Lanes)

DIFFERENCE BETWEEN NEXT GENERATION FPGA AND SINGLE DAUGHTER FPGA CARDS		
	Next Generation FPGA Card (NGFC)	**Single Daughter FPGA Card (SDFC)**
FPGA	Intel Arria 10	Altera Cyclone V
PCI Express	Gen 2, x4 Lane	Gen 1, x4 Lane
Board ID	9320	9330
Sizes (Member Codes)	048 & 115	C9
High Speed LVDS I/O support	yes	no
Number of FMC Style Positions (3" x 3.75")	2	1
Maximum ADC Daughter Card Sample Frequency	4.5 Million SPS	2.0 Million SPS
MsgDma Engine 0, 1, 2, 3 Alignment	quad-word	double-word
MsgDma Engine 0, 1, 2, 3 size multiple	16 bytes	8 bytes
Power Module support	yes	no
DP Transceiver	4 lanes and 8 lanes	4 lanes

2. Requirements

- CP-NGFC or CP-SDFC PCIe board physically installed in the system.
- This driver supports various versions of RedHawk and a selected set of Native Ubuntu and Native Red Hat. Actual supported versions depend on the driver being installed.

3. Documentation

- PCIe Next Generation FPGA I/O Card (NGFC) Software Interface by Concurrent Real-Time.

4. Running on Native Red Hat

Though this driver and hardware work best on Concurrent Real-Time ***RedHawk*** systems, the driver will also be able to run on some selected versions of ***Red Hat*** with some limitations. Some of these limitations are highlighted below. The rest of the document is applicable to all systems.

When compiling the driver, you may get the following message that can be ignored:

Skipping BTF generation for /usr/local/CCRT/drivers/ccrtngfc/driver/ccrtngfc.ko due to unavailability of vmlinux

4.1. Support to build 3rd party modules

If your system isn't setup to build 3rd party modules, you will need to install some of the following packages if they havn't already been installed before being able to compile the driver. Installation process of these modules may differ from system to system. Refer to the particular system for installation of the modules.

```
# yum install ncurses-devel        (to run curses)
# yum install gnuplot              (to run plots for various tests)
# yum install                      <any other package you want to install>
```

4.2. Support for MSI interrupts

- The driver can operate with either MSI or wired interrupts. This is a configuration option that can be selected by editing the *ccrtngfc_nomsi* parameter located in the *…/driver/ccrtngfc_config* file where the driver is installed. Reloading the driver will cause the MSI interrupt handling option to switch.

  - ccrtngfc_nomsi=0 enable MSI support *(default for RedHawk systems)*
  - ccrtngfc_nomsi=1 disable MSI support

  Red Hat systems do not have kernel level hooks like CCRT RedHawk systems to enable MSI on a per board basis for cards using a PLX chip for generating interrupts. This is specially true for the later X11SPA-TF SuperMicro Mother boards and onwards. In this case, if the user wishes to use MSI instead of wired interrupts, they can enable them in various ways as outlined below.

- If MSI interrupts are not being generated and the user wishes to continue using MSI interrupts instead of wired interrupts, they can try to resolve the problem by implementing one the following:

  - Reload the kernel with the grub option "iommu=pt"
  - Reload the kernel with the grub option "iommu=off"
  - Disable IOMMU in the BIOS
  - Reload the kernel with the grub option "intremap=nosid"
  - Reload the kernel with the grub option "intremap=off"
  - Disable VT-d in the BIOS

- ➢ Disable VT-d MSI Interrupt Remapping in the BIOS
- ➢ Disable 4G Decoding in the BIOS

- To add/remove/display the ***intremap*** command to grub, issue the following commands:

  - ➢ # grubby --update-kernel=ALL --args=iommu=pt *(add the parameter)*
  - ➢ # grubby --update-kernel=ALL --args=iommu=off *(add the parameter)*
  - ➢ # grubby --update-kernel=ALL --args=intremap=nosid *(add the parameter)*
  - ➢ # grubby --update-kernel=ALL --remove-args=intremap=nosid *(remove the parameter)*
  - ➢ # grubby --info=ALL *(display parameters)*
  - ➢ # reboot
  - ➢ After system reboots, issue the command "***cat /proc/cmdline***" to see if the added entry is present.

4.3. BIOS and Kernel Level Tuning

It is possible that some tests may get overflow or underflow errors as the card is capable of high sample rate transfers. You may need to lower the sample rates for these tests to run successfully if BIOS and kernel level tuning does not help.

BIOS tuning for real-time is specific to the mother board where the Red Hat kernel is running. The various BIOS settings need to be studied and changed accordingly to make sure that it is running at optimal performance with minimal interference from other processes.

Some Red Hat kernel level tuning can be performed to see if they are helpful in getting a more real-time performance.

Disable HyperThread in BIOS.
To check for number of hyperthreads in system:
    lscpu | grep "Thread(s)"

Disable features that allows SCHED_OTHER tasks to use up to 5% or RT CPUs.
    sysctl kernel.sched_rt_runtime_us=-1
    echo -1 > /proc/sys/kernel/sched_rt_runtime_us

Disable timer migration:
    sysctl kernel.timer_migration=0
    echo 0 > /proc/sys/kernel/timer_migration

Add follwing parameters to ***/etc/default/grub*** line and running ***update-grub*** and ***reboot***.
    GRUB_CMDLINE_LINUX="skew_tick=1 rcu_nocb_poll rcu_nocbs=1-95 nohz=on nohz_full=1-95 kthread_cpus=0 irqaffinity=0 isolcpus=managed_irq,domain,1-95 intel_pstate=disable nosoftlockup tsc=nowatchdog"

    Isolate CPUs e.g *(this command has been officially marked deprecated)*
    isolcpus=1-8,26-30 rcu_nocbs=1-8,26-30 nohz_full=1-8,26-30 rcu_nocb_poll=1-8,26-30

5. Running on Native Ubuntu

Though this driver and hardware work best on Concurrent Real-Time ***RedHawk*** systems, the driver will also be able to run on some selected versions of ***Ubuntu*** with some limitations. Some of these limitations are highlighted below. The rest of the document is applicable to all systems.

When compiling the driver, you may get the following message that can be ignored:

Skipping BTF generation for /usr/local/CCRT/drivers/ccrtngfc/driver/ccrtngfc.ko due to unavailability of vmlinux

5.1. Support to build 3rd party modules

If your system isn't setup to build 3rd party modules, you will need to install some of the following packages if they havn't already been installed before being able to compile the driver. Installation process of these modules may differ from system to system. Refer to the particular system for installation of the modules.

```
# apt install build-essential
# apt install libssl-dev
# apt install nfs-common          (to mount nfs file systems)
# apt install libncurses-dev      (to run curses)
# apt install gnuplot             (to run plots for various tests)
# apt install chrony              (for more accurate clock time)
# apt install                     <any other package you want to install>
```

5.2. Support for MSI interrupts

- The driver can operate with either MSI or wired interrupts. This is a configuration option that can be selected by editing the *ccrtngfc_nomsi* parameter located in the *…/driver/ccrtngfc_config* file where the driver is installed. Reloading the driver will cause the MSI interrupt handling option to switch.

  - ccrtngfc_nomsi=0 enable MSI support *(default for RedHawk systems)*
  - ccrtngfc_nomsi=1 disable MSI support

  Red Hat systems do not have kernel level hooks like CCRT RedHawk systems to enable MSI on a per board basis for cards using a PLX chip for generating interrupts. This is specially true for the later X11SPA-TF SuperMicro Mother boards and onwards. In this case, if the user wishes to use MSI instead of wired interrupts, they can enable them in various ways as outlined below.

- If MSI interrupts are not being generated and the user wishes to continue using MSI interrupts instead of wired interrupts, they can try to resolve the problem by implementing one the following:

  - Reload the kernel with the grub option "iommu=pt"
  - Reload the kernel with the grub option "iommu=off"
  - Disable IOMMU in the BIOS
  - Reload the kernel with the grub option "intremap=nosid"
  - Reload the kernel with the grub option "intremap=off"
  - Disable VT-d in the BIOS
  - Disable VT-d MSI Interrupt Remapping in the BIOS
  - Disable 4G Decoding in the BIOS

- To add/remove/display the ***intremap*** command to grub, issue the following commands:

  - Edit ***/etc/default/grub*** and add "iommu=pt", or "iommu=off" and/or add "intremap=nosid" to "GRUB_CMDLINE_LINUX=" entry
  - # update-grub
  - # reboot
  - After system reboots, issue the command "***cat /proc/cmdline***" to see if the added entry is present.

5.3. Compiling the driver with installed gcc

Depending on the Ubuntu kernel version supported, you will need to make sure that the driver is compiled with the same gcc as the kernel.

Currently, for Ubuntu release 22.04, the kernel 5.15 uses gcc-11 while kernel 6.4 and 6.8 uses gcc-12 and kernel 6.14 is compiled with gcc-13.

If gcc-12 is not installed, you can do the following:

```
# apt install gcc-12
```

Then create alternate entries for each available version:

```
# sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-11 11
# sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-12 12

# sudo update-alternatives --install /usr/bin/x86_64-linux-gnu-gcc x86_64-linux-gnu-gcc
/usr/bin/x86_64-linux-gnu-gcc-11 11

# sudo update-alternatives --install /usr/bin/x86_64-linux-gnu-gcc x86_64-linux-gnu-gcc
/usr/bin/x86_64-linux-gnu-gcc-12 12
```

You can select the appropriate gcc with the following commands:

```
# sudo update-alternatives --config gcc
# sudo update-alternatvies --config x86_64-linux-gnu-gcc
```

All of this will ensure you have the compiler versions that match what the kernel was compiled with.

5.4. BIOS and Kernel Level Tuning

It is possible that some tests may get overflow or underflow errors as the card is capable of high sample rate transfers. You may need to lower the sample rates for these tests to run successfully if BIOS and kernel level tuning does not help.

BIOS tuning for real-time is specific to the mother board where the Red Hat kernel is running. The various BIOS settings need to be studied and changed accordingly to make sure that it is running at optimal performance with minimal interference from other processes.

Some Red Hat kernel level tuning can be performed to see if they are helpful in getting a more real-time performance.

Disable HyperThread in BIOS.
To check for number of hyperthreads in system:

```
lscpu | grep "Thread(s)"
```

Disable features that allows SCHED_OTHER tasks to use up to 5% or RT CPUs.

```
sysctl kernel.sched_rt_runtime_us=-1
echo -1 > /proc/sys/kernel/sched_rt_runtime_us
```

Disable timer migration:

```
sysctl kernel.timer_migration=0
echo 0 > /proc/sys/kernel/timer_migration
```

Add follwing parameters to ***/etc/default/grub*** line and running ***update-grub*** and ***reboot***.

```
GRUB_CMDLINE_LINUX="skew_tick=1 rcu_nocb_poll rcu_nocbs=1-95 nohz=on nohz_full=1-95
kthread_cpus=0 irqaffinity=0 isolcpus=managed_irq,domain,1-95 intel_pstate=disable
nosoftlockup tsc=nowatchdog"
```

Isolate CPUs e.g *(this command has been officially marked deprecated)*

```
isolcpus=1-8,26-30 rcu_nocbs=1-8,26-30 nohz_full=1-8,26-30 rcu_nocb_poll=1-8,26-30
```

6. Installation and Removal

6.1. Hardware Installation

The CP-NGFC card is a Gen 2 PCI Express and the CP-SDFC is a Gen 1 PCI Express product and is compatible with any PCI Express slot. The board must be installed in the system before attempting to use the driver.

<u>Caution</u>: when installing the card insure the computer is powered off and the machine's power cord is disconnected. Please observe electrostatic discharge precautions such as the use of a grounding strap.

The ***ccrtngfc*** driver is designed to support IRQ sharing. If this device's IRQ is being shared by another device then this driver's performance could be compromised. Hence, as far as possible, move this board into a PCI slot who's IRQ is not being shared with other devices. The default driver configuration uses MSI interrupts. If the kernel supports MSI interrupts, then sharing of interrupts will not occur, in which case the board placement will not be an issue.

An ***'lspci -v'*** or the ***'lsirq'*** command can be used to determine the IRQs of various devices in the system.

```
# lspci -v -d 1542:9320

b5:00.0 System peripheral: Concurrent Real-Time Device 9320 (rev 01)
        Subsystem: Concurrent Real-Time Device 0100
        Flags: bus master, fast devsel, latency 0, IRQ 162, NUMA node 0
        Memory at fbe40000 (32-bit, non-prefetchable) [size=32K]
        Memory at fbe00000 (32-bit, non-prefetchable) [size=256K]
        Capabilities: [50] MSI: Enable+ Count=1/4 Maskable- 64bit+
        Capabilities: [78] Power Management version 3
        Capabilities: [80] Express Endpoint, MSI 00
        Capabilities: [100] Virtual Channel
        Capabilities: [200] Vendor Specific Information: ID=1172 Rev=0 Len=044 <?>
        Capabilities: [800] Advanced Error Reporting

# lspci -v -d 1542:9330

ba:00.0 System peripheral: Concurrent Real-Time Device 9330 (rev 01)
        Subsystem: Concurrent Real-Time Device 0100
        Flags: bus master, fast devsel, latency 0, IRQ 182, NUMA node 0
        Memory at fb400000 (32-bit, non-prefetchable) [size=32K]
        Memory at fb000000 (32-bit, non-prefetchable) [size=4M]
        Capabilities: [50] MSI: Enable+ Count=1/4 Maskable- 64bit+
        Capabilities: [78] Power Management version 3
        Capabilities: [80] Express Endpoint, MSI 00
        Capabilities: [100] Virtual Channel
        Capabilities: [200] Vendor Specific Information: ID=1172 Rev=0 Len=044 <?>
        Capabilities: [800] Advanced Error Reporting

# lsirq

   162     b5:00.0 Concurrent Real-Time Unknown device (rev 01)
```

After installing the card, reboot the system and verify the hardware has been recognized by the operating system by executing the following command:

```
# lspci –d 1542:9320
```

For each CP-NGFC PCIe board installed, a line similar to one of the following will be printed, depending on the revision of the system's ***/usr/share/hwdata/pci.ids*** file:

```
b5:00.0 System peripheral: Concurrent Real-Time Device 9320 (rev 01)
```

```
# lspci –d 1542:9330
```

For each CP-SDFC PCIe board installed, a line similar to one of the following will be printed, depending on the revision of the system's ***/usr/share/hwdata/pci.ids*** file:

```
ba:00.0 System peripheral: Concurrent Real-Time Device 9330 (rev 01)
```

If a line similar to the above is not displayed by the **lspci** command, the board has not been properly installed in the system. Make sure that the device has been correctly installed prior to attempting to use the software. One similar line should be found for each installed card.

6.2. Software Installation

Concurrent Real-Time™ port of the ***ccrtngfc*** software is distributed in RPM format for CentOS/Rocky and DEB format for Ubuntu OS on a DVD. Source for the API library and kernel loadable driver are not included, however, source for example test programs as well as documentation is provided in PDF format.

The software is installed in the **/usr/local/CCRT/drivers/ccrtngfc** directory. This directory will be referred to as the "top-level" directory by this document.

Warning: Before installing the software, for RedHawk kernels, the build environment **must** be set up and match the current OS kernel you are using. If you are running one of the preconfigured kernels supplied by Concurrent and have not previously done so, run the following commands while logged in as the root user before installing the driver software:

```
# cd /lib/modules/`uname –r`/build
# ./ccur-config –c -n
```

If you have built and are running a customized kernel configuration the kernel build environment should already have been set up when that custom kernel was built.

To install the ***ccrtngfc*** package, load the DVD installation media and issue the following commands as the **root** user. The system should auto-mount the DVD to a mount point in the **/media** or **/run/media** directory based on the DVD's volume label – in this case **ccrtngfc_driver**. The example's ***[user_name]*** may be ***root***, or the logged-in user. Then enter the following commands from a shell window:

```
== as root ==
      --- on RedHawk 6.5 and below ---
# cd /media/ccrtngfc_driver
      --- or on RedHawk 7.0 and above ---
# cd /run/media/[user_name]/ccrtngfc_driver
     --- or on Ubuntu RedHawk ---
# cd /media/[user_name]/ccrtngfc_driver

# rpm –ivh ccrtngfc_RedHawk_driver*.rpm   (on a RedHawk CentOS/Rocky based system)
                   --or--
# dpkg –i ccrtngfc_RedHawk_driver*.deb    (on a RedHawk Ubuntu based system)
                   --or–
# rpm –ivh ccrtngfc_RedHat_driver*.rpm    (on a Native RedHat based system)
                   --or--
# dpkg –i ccrtngfc_Ubuntu_driver*.deb     (on a Native Ubuntu based system)
```

```
# cd /
# eject
```

On successful installation the source tree for the ***ccrtngfc*** package, including the loadable kernel module, API libraries, and test programs is extracted into the **/usr/local/CCRT/drivers/ccrtngfc** directory by the rpm installation process, which will then compile and install the various software components.

The loadable kernel module is installed in the **/lib/modules/`uname –r`/misc** directory.

Once the package is installed, the driver needs to be loaded with one of the following commands:

```
== as root ==
# cd /usr/local/CCRT/drivers/ccrtngfc
# make load
      --- or on RedHawk 6.5 and below ---
# /sbin/service ccrtngfc start
      --- or on RedHawk 7.0 and above ---
# /usr/bin/systemctl start ccrtngfc
      --- or on Ubuntu RedHawk ---
# /bin/systemctl start ccrtngfc
```

Issue the command below to view the boards found by the driver:

```
# cat /proc/ccrtngfc

Version           : 2024.9.0
Built             : Tue Jan 28 08:39:36 AM EST 2025
Driver Uptime     : 0:0:22:7    (days:hh:mm:ss)
Boards            : 2
  card=0: [b8:00.0] bus=184, slot=0, func=0, irq=181, msi=1, BInfo=9320.01.01
                  FM=05/01/2025 12:00:00 (4.0) FLV=00000000 FWB=00000000 IP=0
                  ID=720335 MC=048 RLS=350  DC=1,1 DEG=41c (MultiFunc) [NGFC]
          DCard 0: Id=1 As=1579321-901 Rv='D' Dt=02/21/2023 Sn=729512 (High
                  Speed Analog Daughter Card)
          DCard 1: Id=1 As=1579321-901 Rv='D' Dt=02/21/2023 Sn=729513 (High
                  Speed Analog Daughter Card)

  card=1: [ba:00.0] bus=186, slot=0, func=0, irq=182, msi=1, BInfo=9330.01.01
                  FM=03/04/2024 00:00:00 (1.0) FLV=00000000 FWB=00000000 IP=0
                  ID= 714762 MC=C9 RLS=150  DC=2 (MultiFunc) [SDFC]
          DCard 0: Id=2 As=1579322-901 Rv='A' Dt=01/22/2024 Sn=710023 (High
                  Speed Digital Daughter Card)
```

You can issue the following command to get more information on the driver if it has loaded successfully.

```
# dmesg

[84381.069167] ccrtngfc: =========== Daughter Card 0 Identity ===========
[84381.069173] ccrtngfc:  Installed on Board[9320]: 0 (SN=720335)
[84381.069175] ccrtngfc:          Daughter Card Id: 1
[84381.069177] ccrtngfc:    Daughter Card Assembly: 1579321-901
[84381.069178] ccrtngfc:                  Revision: 'D'
[84381.069179] ccrtngfc:             Serial Number: 729512
[84381.069180] ccrtngfc:                      Date: 02/21/2023
[84381.069182] ccrtngfc:               Description: 'High Speed Analog Daughter Card'
[84381.069183] ccrtngfc:                     Notes: 'Calibrated 2/21/2023'
[84381.069184] ccrtngfc: ================================================
```

```
[84381.197187] ccrtngfc: =========== Daughter Card 1 Identity ===========
[84381.197191] ccrtngfc:  Installed on Board[9320]: 0 (SN=720335)
[84381.197192] ccrtngfc:          Daughter Card Id: 1
[84381.197193] ccrtngfc:    Daughter Card Assembly: 1579321-901
[84381.197194] ccrtngfc:                  Revision: 'D'
[84381.197195] ccrtngfc:             Serial Number: 729513
[84381.197196] ccrtngfc:                      Date: 02/21/2023
[84381.197198] ccrtngfc:               Description: 'High Speed Analog Daughter Card'
[84381.197199] ccrtngfc:                     Notes: 'Calibrated 2/21/2023'
[84381.197200] ccrtngfc: ================================================
[84381.213235] ccrtngfc: Board=0: PCI Bus# 184, Device# 0.0 BInfo=9320.01.01
                         FM=07/18/2024 12:00:00 (3.0) FLV=00000000 FWB=00000000
                         IP=0 ID=720335 MC=048 RLS=350  DC=1,1 (MultiFunc) [NGFC]

[84381.982994] ccrtngfc: =========== Daughter Card 0 Identity ===========
[84381.982998] ccrtngfc:  Installed on Board[9330]: 1 (SN=0)
[84381.982999] ccrtngfc:          Daughter Card Id: 2
[84381.983000] ccrtngfc:    Daughter Card Assembly: 1579322-901
[84381.983002] ccrtngfc:                  Revision: 'A'
[84381.983002] ccrtngfc:             Serial Number: 710023
[84381.983003] ccrtngfc:                      Date: 01/22/2024
[84381.983005] ccrtngfc:               Description: 'High Speed Digital Daughter Card'
[84381.983006] ccrtngfc:                     Notes: 'No Calibration Required'
[84381.983007] ccrtngfc: ================================================
[84381.983030] ccrtngfc: Board=1: PCI Bus# 186, Device# 0.0 BInfo=9330.01.01
                         FM=03/04/2025 00:00:00 (1.0) FLV=00000000 FWB=00000000
                         IP=0 ID= 714762 MC=C9 RLS=150  DC=2 (MultiFunc) [NGFC]
[84381.983041] ccrtngfc: driver version 2024.9.0 (Built: Tue Jan 28 09:49:34 AM
                         EST 2025) successfully inserted.
```

Note: With RedHawk 7.5 you may see a cautionary message similar to the following when the **ccrtngfc** driver is loaded on the system console or via *dmesg* command:

```
CHRDEV "ccrtngfc" major number 233 goes below the dynamic allocation range
```

As documented in the kernel driver **Documentation/devices.txt** file a range of character device numbers from 234 to 254 are officially available for dynamic assignment. Dynamic assignments start at 254 and grow downward. This range is sometimes exceeded as additional kernel drivers are loaded. Note that this was also the case with earlier kernels – the newer 7.5 kernel has added a runtime check to produce this warning message that the lower bound has been exceeded, not reduced the range of numbers officially available for dynamic assignment. If you see this message please verify the assigned number(s) isn't being used by a device installed on your system.

In addition to the above message, on some systems you may also see messages from APEI *(ACPI Platform Error Interface)* or AER *(Advanced Error Reporting)* which have these error reporting capabilities. These messages will be of the form of unrecoverable hardware errors or some other form of hardware errors for the board when the driver/firmware is loaded and started. This is because during the driver load operation, a fresh copy of the firmware is installed and started. This process of starting is equivalent to issuing a power shutdown and restart of the card. Some operating systems see the device as being no longer present, and generate the message.

On RedHawk 8.x kernels, you may see cautionary messages on the system console or via *dmesg* command similar to the following when the **ccrtngfc** driver is loaded, as this is a proprietary driver:

ccrtngfc: module verification failed: signature and/or required key missing - tainting kernel

6.3. Software Removal

The ***ccrtngfc*** driver is a dynamically loadable driver that can be unloaded, uninstalled and removed. Once removed, the only way to recover the driver is to re-install the ***rpm*** or ***deb*** from the installation DVD:

If any changes have been made to the driver package installed in ***/usr/local/CCRT/drivers/ccrtngfc*** directory, they need to be backed up prior to invoking the removal; otherwise, all changes will be lost.

```
== as root ==
# rpm –e ccrtngfc  (driver unloaded, uninstalled, and deleted – on an RPM
                    based system)
  --or--
# dpkg –P ccrtngfc (driver unloaded, uninstalled, and deleted – on an Debian
                    based system)
```

If, for any reason, the user wishes to unload and uninstall the driver and not remove it, they can perform the following:

```
== as root ==
# cd /usr/local/CCRT/drivers/ccrtngfc
# make unload            (unload the driver from the kernel)
      --- or on RedHawk 6.5 and below ---
# /sbin/service ccrtngfc stop
      --- or on RedHawk 7.0 and above ---
# /usr/bin/systemctl stop ccrtngfc
      --- or on Ubuntu RedHawk ---
# /bin/systemctl stop ccrtngfc
```

To uninstall the ***ccrtngfc*** driver, do the following after it has been unloaded:

```
=== as root ===
# cd /usr/local/CCRT/drivers/ccrtngfc
# make uninstall        (uninstall the driver and library)
```

In this way, the user can simply issue the ***'make install'*** and ***'make load'*** in the ***/usr/local/CCRT/drivers/ccrtngfc*** directory at a later date to re-install and re-load the driver.

On some Debian RedHawk systems, the following message may appear and can be ignored when the package is removed. *"dpkg: warning: while removing ccrtaicc, directory '/usr/local' not empty so not removed"*.

7. Auto-loading the Driver

The ***ccrtngfc*** driver is a dynamically loadable driver. Once you install the package or perform the ***'make install'***, appropriate installation files are placed in the /etc/rc.d/rc*.d or /usr/lib/system/systemd directories so that the driver is automatically loaded and unloaded when Linux is booted and shutdown. If, for any reason, you do not wish to automatically load and unload the driver when Linux is booted or shutdown, you will need to manually issue the following command to enable/disable the automatic loading of the driver:

```
=== as root ===
      --- on RedHawk 6.5 and below ---
# /sbin/chkconfig --add ccrtngfc       (enable auto-loading of the driver)
# /sbin/chkconfig --del ccrtngfc       (disable auto-loading of the driver)
```

```
      --- or on RedHawk 7.0 and above ---
# /usr/bin/systemctl enable ccrtngfc   (enable auto-loading of the driver)
# /usr/bin/systemctl disable ccrtngfc  (disable auto-loading of the driver)
      --- or on Ubuntu RedHawk ---
# /bin/systemctl enable ccrtngfc       (enable auto-loading of the driver)
# /bin/systemctl disable ccrtngfc      (disable auto-loading of the driver)
```

8. FPGA Chip Temperature Monitoring and Control

8.1. Monitoring

This FPGA card has components that can get very hot and damage the card due to the environment where it is placed and the custom software that is installed and running. For this reason, the card is designed with temperature sensors and the software monitors these sensors and informs the user in case they have exceeded threshold values.

There are basically two threshold values that are used to determine the software behavior of the card.

1) Warning Threshold
2) Critical Threshold

When a Warning Threshold is exceeded, the user is informed about this condition by printing a warning message every time the threshold value is exceeded on the system console and can also be viewed via the ***dmesg*** command. The card is still operating normally; however, the user needs to investigate why the card is getting hot and take corrective action as soon as possible.

When a Critical Threshold is exceeded, the software in addition to printing an appropriate error message on the system console will also attempt to shut down the card. This means that all activity to the card is abruptly terminated in 60 seconds as it is important to make sure that the card does not get damaged due to excessive heat. At this point, the user should immediately investigate the reason for the card getting hot by viewing the placement of the card in the chassis, making sure that there is good airflow and that the fans are operating properly. Additionally, they need to make sure that the chassis is not placed in a room that is excessively hot. If within the 60 second holding period, the temperature falls below the critical threshold value, the termination is aborted.

Under normal operating conditions, neither of the above warnings should ever be encountered.

8.2. Control

When the driver is loaded, it is loaded with certain default values for the monitoring frequency in seconds, the Warning Threshold and the Critical Threshold. The user can at any time change these control values with certain limitations. There is also a debugging option that the user can enable to get some useful information that is being monitored.

The following command format can be used to control the monitoring of the FPGA chip temperature

```
# echo ccrtngfc_temperature_monitor BNo,Cmd=Arg > /proc/ccrtngfc
```

Where: ***Bno*** – is a valid board number or '*' for all boards in the system. If only one board is in the system, then the user must use '0' or '*' as the board number.

Cmd – is one of the commands below:

's' – to control the temperature monitoring frequency in integer seconds
'w' – to control the Warning Threshold in integer degrees Centigrade
'c' – to control the Critical Threshold in integer degrees Centigrade
'd' – to control the temperature monitoring debugging display

Arg – is an integer number in seconds for the 's' command, degees Centigrade for the 'w' and 'c' commands and a '0' or 'non-zero' for the debug 'd' command.

The following command can be used to control the frequency of the temperature monitoring. Though the user can ***disable*** the monitoring, it is advisable ***not to do so*** as operating the card at high temperature could damage it.

```
# echo ccrtngfc_temperature_monitor 0,s=300 > /proc/ccrtngfc
```

In the above example, the monitoring is performed every 5 minutes. If the second's value is set to 0, no monitoring takes place. Seconds are integer values ranging from 0 *(off)* to 86400 *(one day).* The default is 60 seconds.

The next command can be used to control the Warning Threshold at which a message will appear on the system console. In cannot be greater than the Critical Threshold value. Warning threshold are integer values ranging from 0 C to 100 C. The default is 80 C.

```
# echo ccrtngfc_temperature_monitor 0,w=75 > /proc/ccrtngfc
```

In the above example, a Warning message will be generated on the system console if the FPGA chip temperature exceeds 75 degrees Centigrade.

You will get a message similar to below every time the warning threshold has exceeded:

```
WARNING!!!: FPGA chip temp 78C exceeded WARNING threshold of 75C [HWM=78C]
```

The command below can be used to control the Critical Threshold at which a message will appear on the system console giving a 60 second warning prior to bringing down the board to the Base Level firmware. In cannot be less than the Threshold value. Critical threshold are integer values ranging from 0 C to 110 C. The default is 100 C.

```
# echo ccrtngfc_temperature_monitor 0,c=95 > /proc/ccrtngfc
```

In the above example, a Critical message will be generated on the system console if the FPGA chip temperature exceeds 95 degrees Centigrade and will load its base level firmware.

You will get a message similar to below once the critical threshold has exceeded:

```
WARNING!!!: ###################################################################
WARNING!!!:        C R I T I C A L     S H U T D O W N     W A R N I N G
WARNING!!!:        =======================================================
WARNING!!!:
WARNING!!!:   FPGA chip temp 96C exceeded CRITICAL threshold of 95C [HWM=96C]
WARNING!!!:
WARNING!!!:   FORCING BOARD 0, SN# 720334 TO SHUT DOWN IN 60 SECONDS!!!
WARNING!!!:
WARNING!!!: ###################################################################
```

When the temperature monitor is awoken 60 seconds later, if the FPGA Chip temperature drops below the Critical Threshold value, the Critical Shutdown is aborted with the following message:

```
###################################################################
       C R I T I C A L     S H U T D O W N     A B O R T E D
       =======================================================

  FPGA chip temp 92C under CRITICAL threshold of 95C [HWM=96C]

  CRITICAL THRESHOLD SHUT DOWN ABORTED!!!

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

After the 60 second warning, if the FPGA Chip temperature is still above the Critical Threshold value, the board will shutdown with the following message. All active applications will be left in an un-predictable state if they are still running and will need to be restarted after the problem has been corrected.

```
ERROR!!!: ###################################################################################
```

```
ERROR!!!:              F O R C E D     C R I T I C A L     S H U T D O W N
ERROR!!!:              ====================================================
ERROR!!!:
ERROR!!!:   FPGA chip temp 102C exceeded CRITICAL threshold of 95C [HWM=102C]
ERROR!!!:   FORCING BOARD 0, SN# 720334 TO SHUT DOWN UNTIL APPROPRIATE COOLING IS INITIATED
ERROR!!!:
ERROR!!!:   PLEASE SHUT DOWN THE SYSTEM IMMEDIATELY UNTIL PROPER COOLING IS ENABLED
ERROR!!!:   OTHERWISE YOU COULD DAMAGE THE CARD AND THE SYSTEM
ERROR!!!:
ERROR!!!:   PLEASE CONTACT CONCURRENT REAL-TIME SUPPORT FOR FURTHER ASSISTANCE
ERROR!!!:
ERROR!!!: ##################################################################################
```

Once the card has shut down, all access to the card is prevented. Any application that is accessing the card will be abruptly stopped. Please shut down the system immediately and determine the cause of over-heating before bringing up the system again and contact Concurrent Real-Time support for further assistance.

The following command can be used to enable debugging.

```
# echo ccrtngfc_temperature_monitor 0,d=1 > /proc/ccrtngfc
```

Once debugging is enabled, information on the current FPGA chip temperature, high-water mark and warning and critical thresholds will appear everytime the threshold monitor is awoken. This is determined by the threshold monitoring frequency which is controlled by the 's' option. To disable the debugging, user can supply a '0' instead of a '1' or alternatively, disable the threshold monitoring by selecting s=0 *(an option that is not recommended).*

Please note that some of this information can also be displayed via the ***…/test/lib/ccrtngfc_sensors*** test.

9. Testing and Usage

Build and run the driver test programs, if you have not already done so:

```
# cd /usr/local/CCRT/drivers/ccrtngfc
# make test                        (build the test programs)
```

Several tests have been provided in the ***/usr/local/CCRT/drivers/ccrtngfc/test*** directory and can be run to test the driver and board.

```
=== as root ===
# cd /usr/local/CCRT/drivers/ccrtngfc
# make test                          (build the test programs)
# ./test/ccrtngfc_chip_temperature   (display chip temperature)
# ./test/ccrtngfc_disp               (display board registers)
# ./test/ccrtngfc_dump               (dump all board resisters)
# ./test/ccrtngfc_rdreg              (display board resisters)
# ./test/ccrtngfc_reg                (Display board resisters)
# ./test/ccrtngfc_regedit            (Interactive board register editor test)
# ./test/ccrtngfc_tst                (Interactive test to test driver and board)
# ./test/ccrtngfc_wreg               (edit board resisters)

# ./test/Flash/ccrtngfc_dump_license (Flash: Dump License)
# ./test/Flash/ccrtngfc_flash        (Flash: Flash FPGA)
# ./test/Flash/ccrtngfc_label        (Flash: Label FPGA)

# ./test/lib/ccrtngfc_acquire_physmem(library: acquire physical memory)
# ./test/lib/ccrtngfc_adc            (library: test ADC channel registers)
```

```
# ./test/lib/ccrtngfc_adc_calibrate  (library: test ADC calibrate)
# ./test/lib/ccrtngfc_adc_fifo       (library: test ADC FIFO channels)
# ./test/lib/ccrtngfc_adc_sps        (library: test ADC samples/channel)
# ./test/lib/ccrtngfc_check_bus      (library: test system jitter)
# ./test/lib/ccrtngfc_clock          (library: test clock)
# ./test/lib/ccrtngfc_compute_output_clock
                                     (library: compute clock)
# ./test/lib/ccrtngfc_dac            (library: test DAC channels)
# ./test/lib/ccrtngfc_dac_calibrate  (library: test DAC calibrate)
# ./test/lib/ccrtngfc_dac_fifo       (library: test DAC FIFO)
# ./test/lib/ccrtngfc_dac_setchan    (library: test DAC channels)
# ./test/lib/ccrtngfc_daughtercard_info
                                     (library: display daughter card info)
# ./test/lib/ccrtngfc_dio            (library: test DIO channels)
# ./test/lib/ccrtngfc_disp           (library: display board registers)
# ./test/lib/ccrtngfc_dma            (library: run dma test)
# ./test/lib/ccrtngfc_example        (library: run example test)
# ./test/lib/ccrtngfc_expires        (library: run expires information test)
# ./test/lib/ccrtngfc_identify       (library: identify cards in the system)
# ./test/lib/ccrtngfc_info           (library: provide information of all
                                               boards)
# ./test/lib/ccrtngfc_ldio_intr      (library: test DIO and LIO change-of-state
                                               interrupt)
# ./test/lib/ccrtngfc_lio            (library: test LIO channels)
# ./test/lib/ccrtngfc_msgdma         (library: modular scatter-gather DMA test)
# ./test/lib/ccrtngfc_msgdma_clone   (library: modular scatter-gather cloning
                                               test)
# ./test/lib/ccrtngfc_msgdma_info    (library: modular scatter-gather DMA info)
# ./test/lib/ccrtngfc_msgdma_multi_clone
                                     (library: modular scatter-gather DMA
                                               multi-cloning test)
# ./test/lib/ccrtngfc_sensors        (library: display/clear Power Module
                                               sensors)
# ./test/lib/ccrtngfc_smp_affinity   (library: display/set IRQ CPU affinity)
# ./test/lib/ccrtngfc_transfer       (library: run DMA and PIO transfer test)
# ./test/lib/ccrtngfc_tst_lib        (library: Interactive test to test driver
                                               and board)
```

10. Re-building the Driver, Library and Tests

If for any reason the user needs to manually rebuild and load an *installed* ***rpm*** or ***deb*** package, they can go to the installed directory and perform the necessary build.

Warning: Before installing the software, for RedHawk kernels, the build environment **must** be set up and match the current OS kernel you are using. If you are running one of the preconfigured kernels supplied by Concurrent and have not previously done so, run the following commands while logged in as the root user before installing the driver software:

```
# cd /lib/modules/`uname -r`/build
# ./ccur-config -c -n
```

If you have built and are running a customized kernel configuration the kernel build environment should already have been set up when that custom kernel was built.

To build the driver and tests:

=== as root ===
```
# cd /usr/local/CCRT/drivers/ccrtngfc
# make clobber    (perform cleanup)
# make            (make package and build the driver, library and tests)
```

*(Note: if you only wish to build the driver, you can enter the '**make driver**' command instead)*

After the driver is built, you will need to install the driver. This install process should only be necessary if the driver is re-built with changes.

=== as root ===
```
# cd /usr/local/CCRT/drivers/ccrtngfc
# make install    (install the driver software, library and man page)
```

Once the driver and the board are installed, you will need to ***load*** the driver into the running kernel prior to any access to the CCRT NGFC board.

=== as root ===
```
# cd /usr/local/CCRT/drivers/ccrtngfc
# make load       (load the driver)
```

11. Software Support

This driver package includes extensive software support and test programs to assist the user in communicating with the board. Refer to the *CONCURRENT PCIe Next Generation FPGA I/O Card (NGFC) Software Interface* document for more information on the product.

11.1. Device Configuration

After the driver is successfully loaded, the device to card association file ***ccrtngfc_devs*** will be created in the ***/usr/local/CCRT/drivers/ccrtngfc/driver*** directory, if it did not exist. Additionally, there is a symbolic link to this file in the ***/usr/lib/config/ccrtngfc*** directory as well. If the user wishes to keep the default one-to-one device to card association, no further action is required. If the device to card association needs to be changed, this file can be edited by the user to associate a particular device number with a card number that was found by the driver. The commented portion on the top of the ***ccrtngfc_devs*** file is automatically generated every time the user issues the ***'make load'*** or ***'/sbin/service ccrtngfc start' (on RedHawk 6.5 and below)*** or ***'/usr/bin/systemctl start ccrtngfc' (on RedHawk 7.0 and above)*** command with the current detected cards, information. Any device to card association edited and placed in this file by the user is retained and used during the next ***'make load'*** or ***'/sbin/service ccrtngfc load'*** or ***'/usr/bin/systemctl start ccrtngfc'*** process.

If the user deletes the ***ccrtngfc_devs*** file and recreates it as an empty file and performs a ***'make load'*** or if the user does not associate any device number with card number, the driver will provide a one to one association of device number and card number. For more information on available commands, view the commented section of the ***ccrtngfc_devs*** configuration file.

Warning: If you edit the ***ccrtngfc_devs*** file to associate a device to a card, you will need to re-issue the ***'make load'*** or ***'/sbin/service ccrtngfc start'*** or ***'/usr/bin/systemctl start ccrtngfc'*** command to generate the necessary device to card association. This device to card association will be retained until the user changes or deletes the association. **If any invalid association is detected, the loading of the driver will fail**.

11.2. Library Interface

There is an extensive software library that is provided with this package. For more information on the library interface, please refer to the *PCIe Next Generation FPGA I/O Card (NGFC) Software Interface by Concurrent Real-Time* document.

11.3. Debugging

This driver has some debugging capability and should only be enabled while trying to trouble-shoot a problem. Once resolved, debugging should be disabled otherwise it could adversely affect the performance and behavior of the driver.

To enable debugging, the ***Makefile*** file in ***/usr/local/CCRT/drivers/ccrtngfc/driver*** should be edited to un-comment the statement (*remove the preceding* '#'):

```
# EXTRA_CFLAGS += -DCCRTNGFC_DEBUG
```

Next, compile and install the driver

```
# cd /usr/local/CCRT/drivers/ccrtngfc/driver
# make
# make install
```

Next, edit the ***ccrtngfc_config*** file in ***/usr/local/CCRT/drivers/ccrtngfc/driver*** to un-comment the statement (remove the preceding '#'):

```
# ccrtngfc_debug_mask=0x00002040
```

Additionally, the value of the debug mask can be changed to suite the problem investigated. Once the file has been edited, the user can load the driver by issuing the following:

```
# cd /usr/local/CCRT/drivers/ccrtngfc/driver
# make load
```

The user can also change the debug flags after the driver is loaded by passing the above debug statement directly to the driver as follows:

```
# echo "ccrtngfc_debug_mask=0x00082047" > /proc/ccrtngfc
```

Following are the supported flags for the debug mask as shown in the ***ccrtngfc_config*** file.

```
###############################################################################
#                                                                             #
#        D_ENTER         0x00000001  /* enter routine */                      #
#        D_EXIT          0x00000002  /* exit routine */                       #
#                                                                             #
#        D_L1            0x00000004  /* level 1 */                            #
#        D_L2            0x00000008  /* level 2 */                            #
#        D_L3            0x00000010  /* level 3 */                            #
#        D_L4            0x00000020  /* level 4 */                            #
#                                                                             #
#        D_ERR           0x00000040  /* level error */                        #
#        D_WAIT          0x00000080  /* level wait */                         #
#                                                                             #
#        D_INT0          0x00000100  /* interrupt level 0 */                  #
#        D_INT1          0x00000200  /* interrupt level 1 */                  #
#        D_INT2          0x00000400  /* interrupt level 2 */                  #
#        D_INT3          0x00000800  /* interrupt level 3 */                  #
#        D_INTW          0x00001000  /* interrupt wakeup level */             #
#        D_INTE          0x00002000  /* interrupt error */                    #
#                                                                             #
#        D_RTIME         0x00010000  /* display read times */                 #
#        D_WTIME         0x00020000  /* display write times */                #
#        D_REGS          0x00040000  /* dump registers */                     #
#        D_IOCTL         0x00080000  /* ioctl call */                         #
#                                                                             #
```

```
#        D_DATA          0x00100000  /* data level */                        #
#        D_DMA           0x00200000  /* DMA level */                         #
#        D_DBUFF         0x00800000  /* DMA buffer allocation */             #
#                                                                            #
#        D_NEVER         0x00000000  /* never print this debug message */    #
#        D_ALWAYS        0xffffffff  /* always print this debug message */   #
#        D_TEMP          D_ALWAYS    /* Only use for temporary debug code */ #
##############################################################################
```

Another variable ***ccrtngfc_debug_ctrl*** is also supplied in the ***ccrtngfc_config*** that the driver developer can use to control the behavior of the driver. The user can also change the debug flags after the driver is loaded by passing the above debug statement directly to the driver as follows:

```
# echo "ccrtngfc_debug_ctrl=0x00001234" > /proc/ccrtngfc
```

In order to make use of this variable, the driver must be coded to interrogate the bits in the ***ccrtngfc_debug_ctrl*** variable and alter its behavior accordingly.

12. Notes and Errata

- The DACs can only operate in high-speed mode when the channels are configured as single-ended. Results are unpredictable if they operate in differential mode.
- The SDFC card (9330) has a problem with its MsgDma IP that will cause data transfer to fail if all of the following conditions are met:

  1. Board is an SDFC 9330 Single Daughter version
  2. When performing double-word transfers using MSGDMA Engines 0 to 3
  3. When using quad-word size and aligned transfers, the DMA will not fail for engines 0 to 3
  4. When performing Physical Memory to Physical Memory transfers
  5. When using more than one MsgDma descriptor
  6. When the transfer size of the descriptor is exactly 0x3BF0 plus a multiple of 0x400, e.g. 0x3FF0, 0x43F0, 0x47F0, 0x4BF0, 0x4FF0, etc.
- On the new DELL systems, e.g. HQR74-2R31C- 2KE60 motherboard with UEFI setting, a ***reboot*** command *(not power-up)* causes the BIOS to report a previously queued PCIe device error requiring the user to select ***'F1'*** at the prompt to continue when the system is booting. The error is generated due to the FPGA card going through a firmware reload on boot, which is perfectly normal. To avoid the user entering ***'F1'*** at the prompt every time a ***reboot*** command is issued, they can add the following entry to the grub line *--kopt-add="reboot=hard,pci"*.
- PLL clock synchronization feature is not supported in the first firmware release 8/25/2023 R1.0.
- In some kernel releases, when a package is installed or uninstalled, you may see a warning message on the system console similar to ***"systemd-rc-local-generator[22094]: /etc/rc.d/rc.local is not marked executable, skipping."***. This is for informational purpose only and can be ignored.
- The DIO that is located on the mother-board has each of the 32 channels controlled by 32 independent ports. Hence, each channels direction can be controlled independent of each other. The optional DIO daughter card has groups of 4 channels controlled by a single port. Hence the direction of the channels must be controlled in groups of 4 adjacent channels.
- If a square or pulse signal is supplied to an ADC channel on a daughter card, you will observe a ringing noise on the DAC channels on the daughter card corresponding to the frequency of the ADC signal. The larger the amplitude of the signal, the larger is the resulting noise on the DAC channels.
- Each multi-function analog daughter card has multiple functions like ADC, DAC, DIO, LIO incorporated into it. It is possible that when one component is operating on the daughter card it may have a performance impact on other components within the same daughter card when operated concurrently.
- Region Addressing allows a user to supply a physical address to access memory beyond its domain. Care should be taken in supplying a valid physical address, otherwise results can be unpredictable including kernel crash or system hang. ***On RH6.0 through RH7.2 kernels, a dmesg 'ioremap' kernel-warning message may be generated when kernel I/O mapping is performed***

on a user supplied physical memory address that is allocated and reserved by someone other than the user. If this is exactly what the user is intending to do, the message can be ignored as this warning should have no effect on the driver operation or the system. It may appear only once since a system reboot.

- It is meaningless to perform Cloning on a FIFO region for two reasons. Firstly, each data in a FIFO is synchronous, however, the Cloned region is accessed asynchronously. Secondly, when the FIFO runs empty *(underflow)* or cannot accept more data *(overflow)* the results are unpredictable as there is no flow control in the Cloning operation.
- If a kernel is configured with the CONFIG_DEBUG_LOCK_ALLOC define, the driver will fail to compile due to mutex_lock_nested() call being included with GPL requirement. If you want to successfully compile the driver, you will need to remove the CONFIG_DEBUG_LOCK_ALLOC define and rebuild the kernel.
- Ubuntu kernels RH8.0 onwards may have the default ***systemd-timesyncd*** daemon installed which does not accurately adjust the system clock causing the Sample/Second test to fall out of tolerance and fail. You may want to replace the default with the ***chrony*** package for a more accurate time asjustment.
- On some Debian systems, the following message can be ignored when the package is removed. *"dpkg: warning: while removing ccrtngfc, directory '/usr/local' not empty so not removed"*
- Driver and board support MSI interrupts. It can be configured for wired interrupts. MSI support is the default.

13. High Speed Analog Daughter Card (HSAD)

13.1. Overview

The High-Speed Analog Daughtercard (HSAD) is a single width FMC style card for use with the Next Generation FPGA Card (NGFC or SDFC). It contains two analog sections: 1) Twelve (12) channels of high speed 16-bit SAR analog input and 2) Twelve (12) channels of high speed 16-bit analog output. The Analog to Digital function is implemented using Analog Devices LTC2325 ADC's. The Digital to Analog function is implemented using Analog Devices AD5545 DAC's. All the analog I/O pins are routed to the motherboard front panel using the FMC HPC pins.

The DACs can only operate in high-speed mode when the channels are configured as single-ended. Results are unpredictable if they operate in differential mode.

Features of the HSAD are:

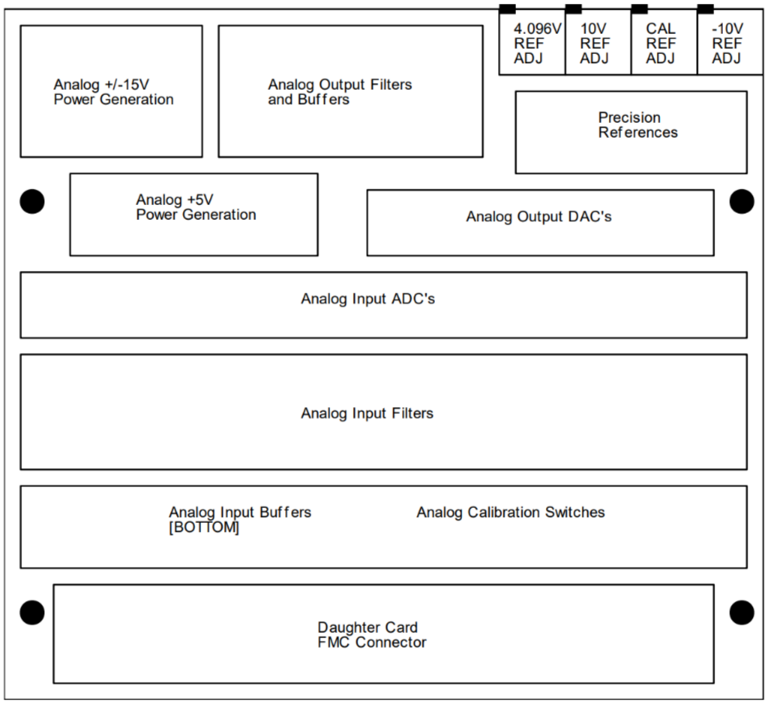
- ❖ General
  - Single Width FMC (3" x 3.75")
  - Low Noise Analog Power Generation
  - In System Calibration
  - NIST Traceable Calibration Standard
  - In System Firmware Update
  - Motherboard Front Panel I/O (Using HPC Pins)
  - Industry Standard High-Density SCSI 68-pin Connectors

- ❖ Analog Input Section
  - 16-channel 16-bit Analog-to-Digital Conversion
  - Differential or Single-ended Input
  - +/-10V Input Range
  - Input Impedance ~1Meg ohm
  - Directly Addressable Conversion Data Registers
  - 4.5Mhz Maximum Sampling Rate
  - 2.5Mhz Active Input Filter
  - 128K Word Conversion Data FIFOs with DMA

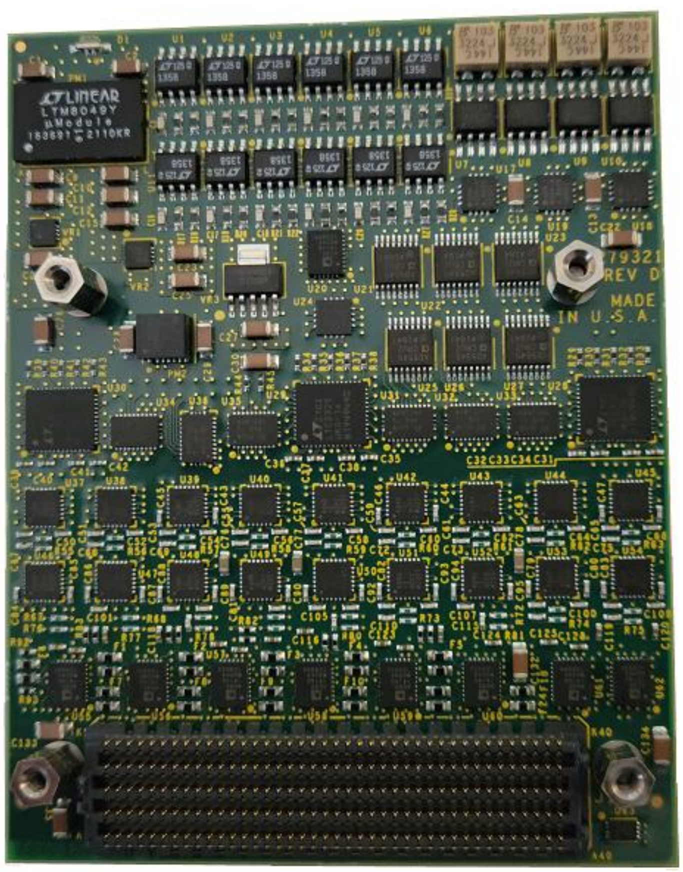
- ❖ Analog Output Section

- 16-bit Digital-to-Analog Conversion
- 12-channel Single-ended Output or 6-channel Differential Output
- +/-10V Single-ended Output Range
- +/-20V Differential Output Range
- 10 Milliamp Maximum Output Drive
- 1Mhz Maximum Updates Per Second (12-channels)
- 2Mhz Maximum Updates Per Second (6-channels)
- 1.5Mhz Active Output Filter
- 128K Word Conversion Data FIFOs with DMA

13.2. Block Diagram of ADC/DAC Daughter Card



13.3. Picture of ADC/DAC Daughter Card



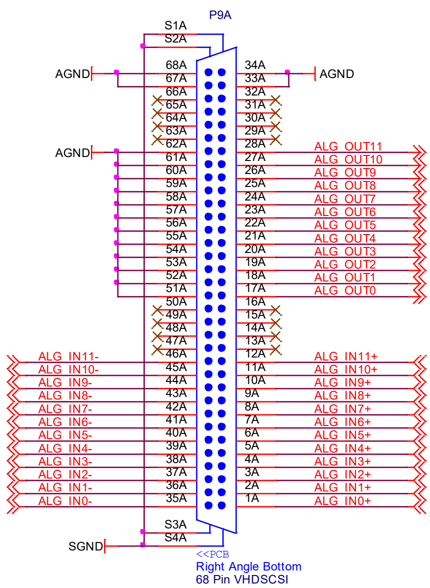
13.4. Physical Characteristics

The High Speed Analog Daughtercard (HSAD) is a single width FMC style card (3 inch long x 3.75 inch high). It consumes approximately 20 watts (1.5 amp @ 12 volt, 100mA @ 3.3 volt and 200mA @ 1.8 volt). Forced air cooling is required.

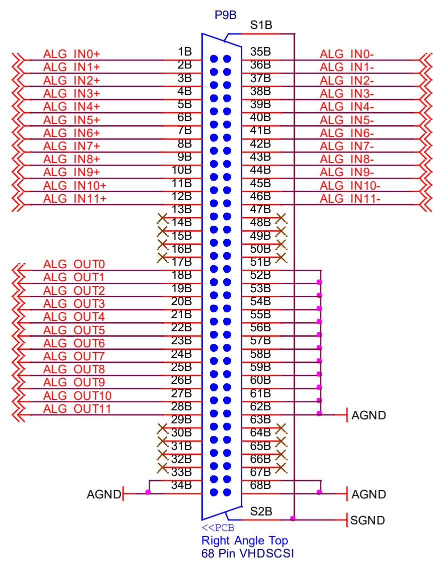
13.5. External Connectors

The NGFC I/O signals will interface to the front panel I/O connectors on the motherboard. They are industry standard 68-pin VHD SCSI type connectors with the following pin-out when looking at the motherboard:

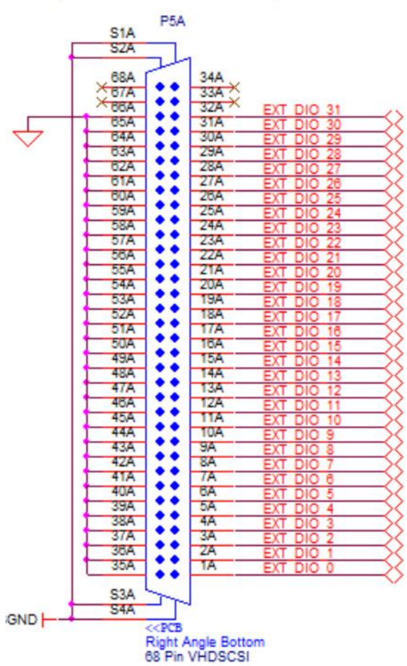
Daughter Card #1 Input/Output Connector
(Top Left Connector)



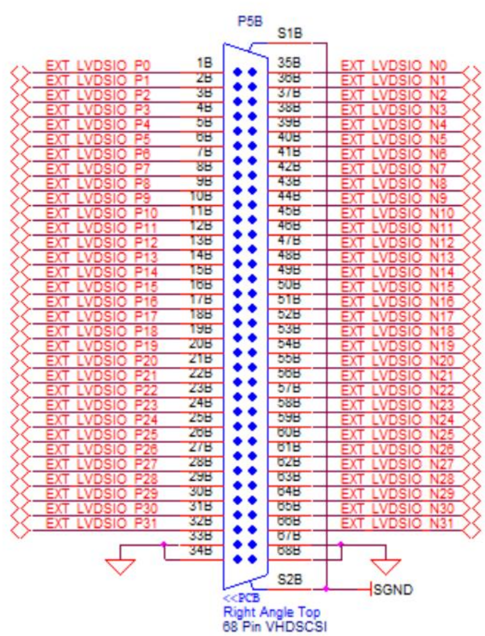
Daughter Card #2 Input/Output Connector
(Top Right Connector)



Digital Input/Output Connector
(Bottom Left Connector)



LVDS Input/Output Connector
(Bottom Right Connector)



13.6. External Connector Notes

1) The I/O signals on the HSAD are high speed and have limited protection. Do not exceed +/-15 volt into any pin or permanent damage may occur.
2) An analog ground connection is required for the ESD and over/under voltage protection circuits to function correctly for the analog signals.
3) All other connectors on the board are used for manufacturing test and should not have anything attached to them.

13.7. Board Indicator

The HSAD has one bicolor LED indicator located on the edge of the board visible. If the board is in a reset state, the indicator will be dim Red. After reset is complete, the indicator will cycle through Red and Green for approximately 1 second each as a lamp test. If the indicator remains solid or flashing Red after reset is complete or during board operation it would indicate a board malfunction. See the Power Fail Code section for more information.

Board Indicator		
Color	**Description**	**Analog I/O**
Dim Red	Board Not Enabled (Reset)	Not Active
Flashing Red	<See Power Fail Code section>	Not Active
Green	Board Enabled	Active

Appendix A: Board Indicators

The NGFC has two multicolor LED indicator located between the two front panel connector pairs. If the board is in a reset state the right indicator will be solid Red. After reset is complete, the indicators will cycle through Red, Green and Blue for approximately 1 second each as a lamp test. If the either of the indicators remain solid or flashing Red after reset is complete or during board operation it would indicate a board malfunction. See the Board Faults section for more information. Other states of the board during normal operation are indicated as follows:

Left Indicator		
Color	**Description**	**Digital Outputs**
Red	<See board faults section>	Not Active
Flashing Red	<See board faults section>	Not Active
Green	Digital Inputs Enabled	Not Active
Blue	Digital Outputs Enabled	Active

Right Indicator		
Color	**Description**	**Analog Outputs**
Red	<See board faults section>	Not Active
Flashing Red	<See board faults section>	Not Active
Green	Daughter Card Inputs Enabled	Not Active
Blue	Daughter Card Outputs Enabled	Active

Note: If either left or right LED has to represent both the blue *(output)* ***and*** the green *(input)* color, the Digital or Analog *Output* signals supersede the inputs and therefore the corresponding LED will display a blue color.

If the user selects to *identify* the board, both the left and right LEDs will flash approximately once every second while board identification is enabled. Once board identification is disabled, the left and right LEDs will display the Analog and Digital Input and Output settings.

If the FPGA card is running a custom firmware, it is possible that the meaning of the LED colors can change based on the custom firmware running on the card at that time.

Appendix B: Board Faults

The NGFC has on board monitoring of the power up/down sequence and initialization of the FPGA. The front panel indicators along with the multi-board synchronization J1 connector LED indicators will provide some level of feedback if there is a problem during board initialization and operation as follows:

Front Panel		J1 Input		J1 Output			Description	Cause
Left	**Right**	**R/G**	**Y/G**	**R/G**	**Y/G**	**#**		
Off	Red	OFF	OFF	OFF	OFF	-	Board Not Programmed	N/A
Red	Red	Red	Yellow	Red	Yellow	-	Board Reset	N/A
Red	Flashing Red	OFF	OFF	OFF	OFF	00	<Reserved>	Board Malfunction
Red	Flashing Red	OFF	OFF	OFF	Yellow	01	No 12V Power Detected	Board Malfunction
Red	Flashing Red	OFF	OFF	Red	OFF	02	Internal Power Status Error	Board Malfunction
Red	Flashing Red	OFF	OFF	Red	Yellow	03	Ext Power Status Error	Board Malfunction
Red	Flashing Red	OFF	Yellow	OFF	OFF	04	ADC Start Timeout	Board Malfunction
Red	Flashing Red	OFF	Yellow	OFF	Yellow	05	ADC Convert Timeout	Board Malfunction
Red	Flashing Red	OFF	Yellow	Red	OFF	06	Internal Power Wait*	Board Malfunction
Red	Flashing Red	OFF	Yellow	Red	Yellow	07	External Power Wait*	Board Malfunction
Red	Flashing Red	Red	OFF	OFF	OFF	08	Main Power Wait*	Board Malfunction
Red	Flashing Red	Red	OFF	OFF	Yellow	09	Ground Check Wait*	Board Malfunction
Red	Flashing Red	Red	OFF	Red	OFF	10	V0.9 Off Status Error	Board Malfunction
Red	Flashing Red	Red	OFF	Red	Yellow	11	V1.03 Off Status Error	Board Malfunction
Red	Flashing Red	Red	Yellow	OFF	OFF	12	V1.8P Off Status Error	Board Malfunction
Red	Flashing Red	Red	Yellow	OFF	Yellow	13	V1.8S Off Status Error	Board Malfunction
Red	Flashing Red	Red	Yellow	Red	OFF	14	V3.3S Off Status Error	Board Malfunction
Red	Flashing Red	Red	Yellow	Red	Yellow	15	<Reserved>	Board Malfunction

Table is continued on next page >>>

Notes:

1. Connector J1 Output is closest to the front panel side of the board.
2. * Power & ground waits will loop indefinitely for the condition to satisfy. This could be due to a power supply not being within tolerance.

Board Faults (continued)

Front Panel		J1 Input		J1 Output			Description	Cause
Left	Right	R/G	Y/G	R/G	Y/G	#		
Red	Flashing Red	OFF	OFF	OFF	OFF	16	<Reserved>	Board Malfunction
Red	Flashing Red	OFF	OFF	OFF	Green	17	12V Main Low Voltage	Board Malfunction
Red	Flashing Red	OFF	OFF	Green	OFF	18	V0.9 On Status Error	Board Malfunction
Red	Flashing Red	OFF	OFF	Green	Green	19	V1.03 On Status Error	Board Malfunction
Red	Flashing Red	OFF	Green	OFF	OFF	20	V0.9 On Fault Error	Board Malfunction
Red	Flashing Red	OFF	Green	OFF	Green	21	V0.9 Resistor On Check	Board Malfunction
Red	Flashing Red	OFF	Green	Green	OFF	22	V1.03 Resistor On Check	Board Malfunction
Red	Flashing Red	OFF	Green	Green	Green	23	V0.9 Resistor Off Check	Board Malfunction
Red	Flashing Red	Green	OFF	OFF	OFF	24	V1.03 Resistor Off Check	Board Malfunction
Red	Flashing Red	Green	OFF	OFF	Green	25	12V Main Low Voltage	Board Malfunction
Red	Flashing Red	Green	OFF	Green	OFF	26	V1.8P On Status Error	Board Malfunction
Red	Flashing Red	Green	OFF	Green	Green	27	V1.8P Resistor On Check	Board Malfunction
Red	Flashing Red	Green	Green	OFF	OFF	28	V1.8P Resistor Off Check	Board Malfunction
Red	Flashing Red	Green	Green	OFF	Green	29	12V Main Low Voltage	Board Malfunction
Red	Flashing Red	Green	Green	Green	OFF	30	V1.8S On Status Error	Board Malfunction
Red	Flashing Red	Green	Green	Green	Green	31	<Reserved>	Board Malfunction

Table is continued on next page >>>

Notes:

1. Connector J1 Output is closest to the front panel side of the board.
2. All other combinations would be N/A and indicate a board malfunction.

Board Faults (continued)

Front Panel		J1 Input		J1 Output			Description	Cause
Left	Right	R/G	Y/G	R/G	Y/G	#		
Red	Flashing Red	OFF	OFF	OFF	OFF	32	<Reserved>	Board Malfunction
Red	Flashing Red	Red	Yellow	Red	Green	33	V3.3S On Status Error	Board Malfunction
Red	Flashing Red	Red	Yellow	Green	Yellow	34	V1.8S Resistor On Check	Board Malfunction
Red	Flashing Red	Red	Yellow	Green	Green	35	V3.3S Resistor On Check	Board Malfunction
Red	Flashing Red	Red	Green	Red	Yellow	36	V1.8S Resistor Off Check	Board Malfunction
Red	Flashing Red	Red	Green	Red	Green	37	V3.3S Resistor Off Check	Board Malfunction
Red	Flashing Red	Red	Green	Green	Yellow	38	Internal Power Transfer Status	Board Malfunction
Red	Flashing Red	Red	Green	Green	Green	39	<Not Used>	Board Malfunction
Red	Flashing Red	Green	Yellow	Red	Yellow	40	<Not Used>	Board Malfunction
Red	Flashing Red	Green	Yellow	Red	Green	41	FPGA Pwr On Status Error	Board Malfunction
OFF	Flashing Red	Green	Yellow	Green	Yellow	42	FPGA Status Error	Board Malfunction
OFF	Flashing Red	Green	Yellow	Green	Green	43	FPGA Config Error *	Board Malfunction
OFF	Flashing Red	Green	Green	Red	Yellow	44	FPGA Status Error *	Board Malfunction
OFF	Flashing Red	Green	Green	Red	Green	45	<Not Used>	Board Malfunction
OFF	Flashing Red	Green	Green	Green	Yellow	46	<Not Used>	Board Malfunction
OFF	Flashing Red	Green	Green	Green	Green	47	<Reserved>	Board Malfunction

Table is continued on next page >>>

Notes:

1. Connector J1 Output is closest to the front panel side of the board
2. * FPGA configuration and status errors may be due to a non-programmed part or corrupted firmware.

Board Faults (continued)

Front Panel		J1 Input		J1 Output			Description	Cause
Left	Right	R/G	Y/G	R/G	Y/G	#		
Blue	Flashing Red	OFF	OFF	OFF	OFF	48	<Reserved>	Board Malfunction
Blue	Flashing Red	OFF	OFF	OFF	Yellow	49	V0.9 On Status Error	Board Malfunction
Blue	Flashing Red	OFF	OFF	Red	OFF	50	V1.03 On Status Error	Board Malfunction
Blue	Flashing Red	OFF	OFF	Red	Yellow	51	V1.8P On Status Error	Board Malfunction
Blue	Flashing Red	OFF	Yellow	OFF	OFF	52	V1.8S On Status Error	Board Malfunction
Blue	Flashing Red	OFF	Yellow	OFF	Yellow	53	V3.3S On Status Error	Board Malfunction
Blue	Flashing Red	OFF	Yellow	Red	OFF	54	V0.9 On Fault Error	Board Malfunction
Blue	Flashing Red	OFF	Yellow	Red	Yellow	55	IP 12V Main Low Voltage	Board Malfunction
Blue	Flashing Red	Red	OFF	OFF	OFF	56	EP 12V Main Low Voltage	Board Malfunction
Blue	Flashing Red	Red	OFF	OFF	Yellow	57	12V Main Low Voltage	Board Malfunction
Blue	Flashing Red	Red	OFF	Red	OFF	58	<Not Used>	Board Malfunction
Blue	Flashing Red	Red	OFF	Red	Yellow	59	<Not Used>	Board Malfunction
Blue	Flashing Red	Red	Yellow	OFF	OFF	60	<Not Used>	Board Malfunction
Blue	Flashing Red	Red	Yellow	OFF	Yellow	61	<Not Used>	Board Malfunction
Blue	Flashing Red	Red	Yellow	Red	OFF	62	<Not Used>	Board Malfunction
Blue	Flashing Red	Red	Yellow	Red	Yellow	63	<Reserved>	Board Malfunction

Table is continued on next page >>>

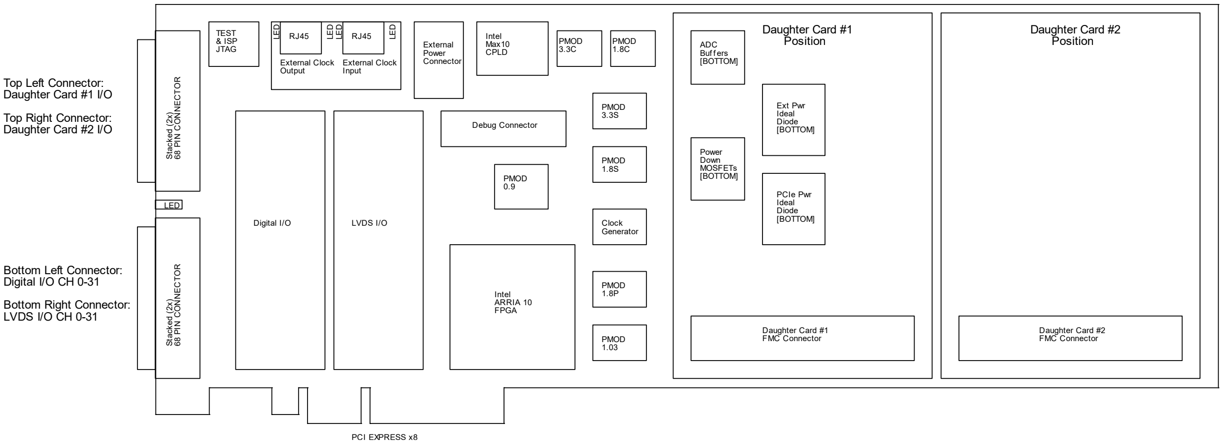
Notes:

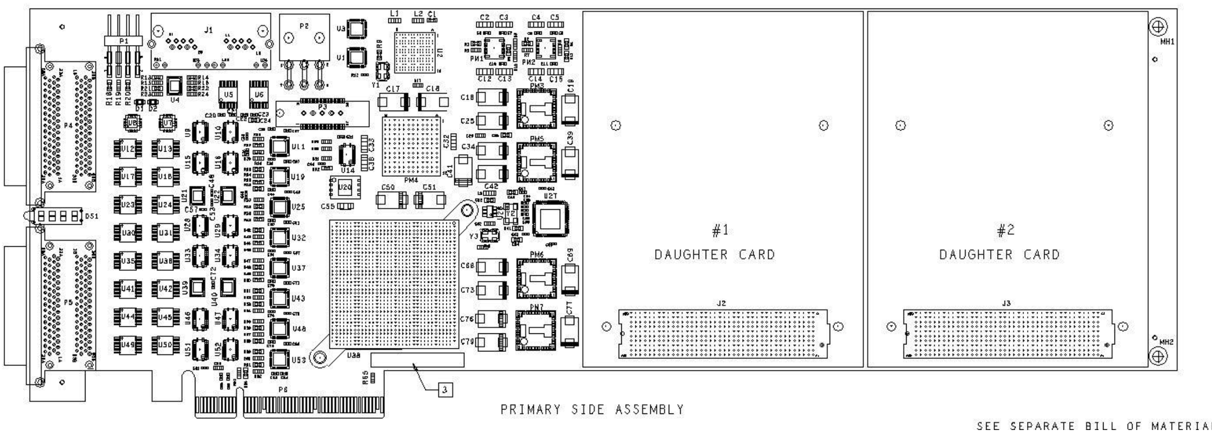
1. Connector J1 Output is closest to the front panel side of the board
2. All other combinations would be N/A and indicate a board malfunction.

Appendix C: External Connections and Pin-outs

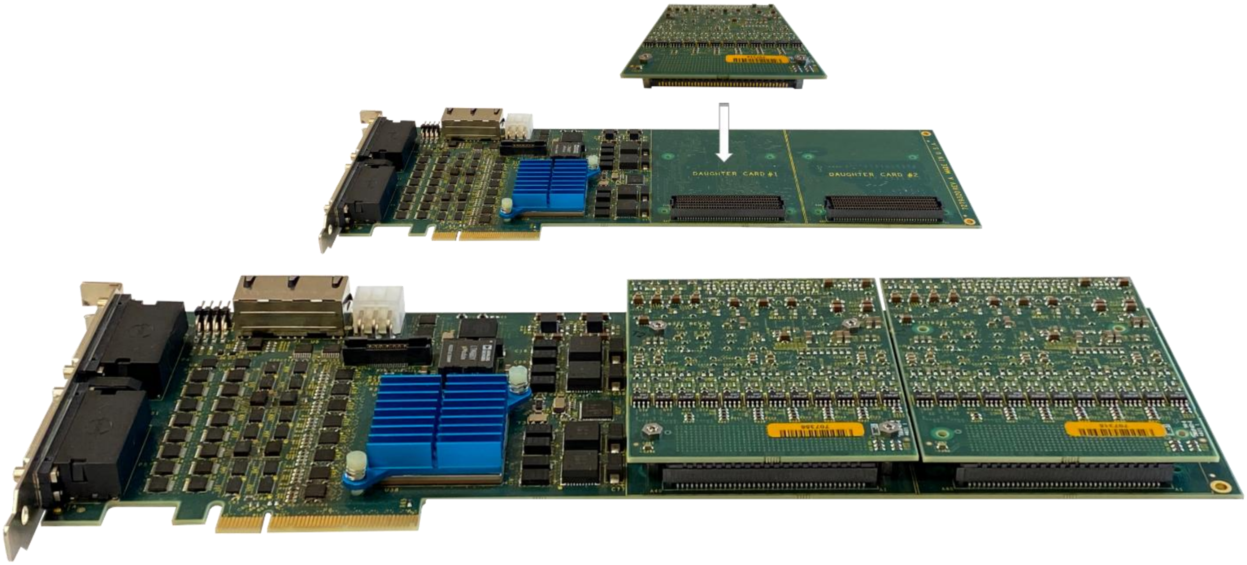
4) The DIO lines are 5V TTL compatible signal levels. Do not exceed TTL levels into any pin or permanent damage may occur.
5) The LVDS lines are LVDS signal levels. Do not exceed LVDS levels into any pin or permanent damage may occur.
6) Connector pair P4A/B connect to the daughter cards I/O. Consult the design specification for the appropriate daughter card for the actual pinout.
7) Connector J1 located near the top rear edge of the board is used for external clocking. CAT5 capable or greater shielded cable should be used. The output from the Master board is the RJ45 socket closest to the front panel. The input to the Slave board is the RJ45 socket farthest from the front panel.
8) All other connectors on the board are used for manufacturing test and should not have anything attached to them.

Appendix D: The Next Generation FPGA I/O Board





Appendix D: The Next Generation FPGA I/O Board (continued)



This page intentionally left blank