



AN 908: Enabling 4G Wireless Acceleration in FlexRAN

for the Intel® FPGA Programmable Acceleration Card N3000



Online Version



Send Feedback

AN-908

ID: **683736**

Version: **2020.01.30**

Contents

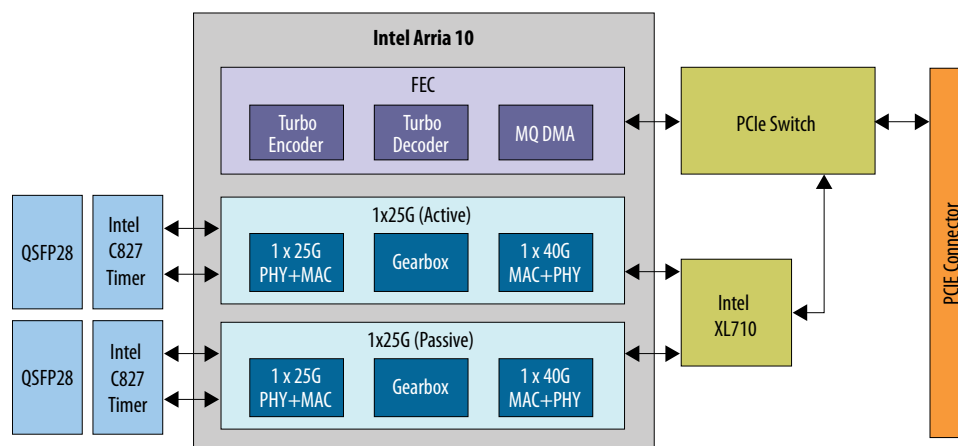
1. About the 4G Wireless Acceleration Reference Design	3
1.1. 4G User Image Features.....	3
1.2. About the Intel PAC N3000.....	4
1.2.1. Factory Image for 2x2x25 GbE.....	5
2. 4G User Image Description.....	7
2.1. User Image Power Management.....	7
2.2. 4G Channel Coder.....	8
2.2.1. 4G Channel Coder Throughput.....	9
2.2.2. 4G Turbo-V Encoder and Decoder Tests.....	10
2.2.3. 4G VRAN Universal Verification Methodology.....	10
2.3. Fronthaul IO.....	11
2.3.1. O-RAN Compression and Decompression.....	11
2.4. User Image Software.....	12
3. Document Revision History for AN 908: Enabling 4G Wireless Acceleration in FlexRAN for the Intel® FPGA Programmable Acceleration Card N3000.....	13

1. About the 4G Wireless Acceleration Reference Design

The 4G Wireless Acceleration reference design provides additional IP (Intel FPGA IP and software drivers) to support fronthaul IO and 4G channel coding (forward error correction (FEC)).

The Intel FPGA PAC N3000 provides an on-board PCIe switch that connects fronthaul and 4G channel coding functions to a PCIe Gen3x16 edge connector. The Intel FPGA PAC N3000 is a general-purpose acceleration card for networking.

Figure 1. Data flow for the user image, FEC, and Fronthaul IO



1.1. 4G User Image Features

FEC features:

- Functionality independent of 25G I/O (look-aside model)
- Support for one physical function (PF) and 8 virtual functions (VFs) simultaneously accessing acceleration
- 64 queues supported equally split between uplink and downlink.
- Long-term evolution (LTE) Turbo encoding with interleaving and rate matching
- LTE Turbo decoding with sub-block de-interleaving of reverse rate matching.
- Load balancer distributes the pending requests to transmitter and receiver

- Early termination CRC24A and CRC24B
- Software enablement by baseband device (bbdev) API (targeted to upstream to Data Plane Development Kit (DPDK))
- Function-level reset

Fronthaul IO features:

- 25G MAC and 25G PHY IP connectivity to retimer and a quad small form factor pluggable (QSFP).
- 40G MAC and 40G PHY IP connectivity to Fortville networking device
- Gearbox to enable 25G connectivity to QSFP.
- In-line compression and decompression.
- Software enablement by Open Platform Acceleration Environment (OPAE), DPDK and bbdev.

1.2. About the Intel PAC N3000

You enable the Intel PAC N3000 through six main firmware components. Five components are not specific to wireless but are for FPGA workload. Intel also provides a software package for the Intel PAC N3000.

The Intel PAC N3000 supports the factory image with RSU capability in on-board 1 Gb flash in page 0 as a fail over image. The user image is stored in 1 Gb flash.

Intel develops and owns all of the following Intel PAC N3000 components (including all updates) except the Intel® Arria® 10 flash page 1 user image:

- Intel MAX® 10 Nios flash.
 - Fixed configuration. RSU capable. Intel loads the binary image.
- PCIe software.
 - Intel flashes the binary images.
 - Fixed configuration for PCIe configuration.
 - Not RSU capable.
- Intel C827 retimer.
 - Intel flashes the binary EEPROM.
 - Power-up configuration initialization by Intel Arria 10 soft Nios processor through Intel MAX 10.
 - Fixed configuration for XCVR.
 - Encrypted.

- Intel XL710.
 - Intel flashes the binary images.
 - Fixed configuration for XCVR configuration.
 - RSU capable.
- Intel Arria 10 flash factory image page 0.
 - Intel flashes the binary images.
 - Not RSU capable.
- Intel Arria 10 flash page 1 user image.
 - RSU capable.
 - Intel provides the top-level reference design under a software license agreement.
 - Contains multiple encrypted IP blocks provided under a software license agreement.
 - You own the production image and design.

Related Information

- [Intel® FPGA PAC N3000 AFU Developer Guide](#)
- [Intel® FPGA PAC N3000 Data Sheet](#)
- [Security User Guide: Intel FPGA Programmable Acceleration Card N3000](#)
- [Intel Acceleration Stack User Guide: Intel FPGA Programmable Acceleration Card N3000](#)

1.2.1. Factory Image for 2x2x25 GbE

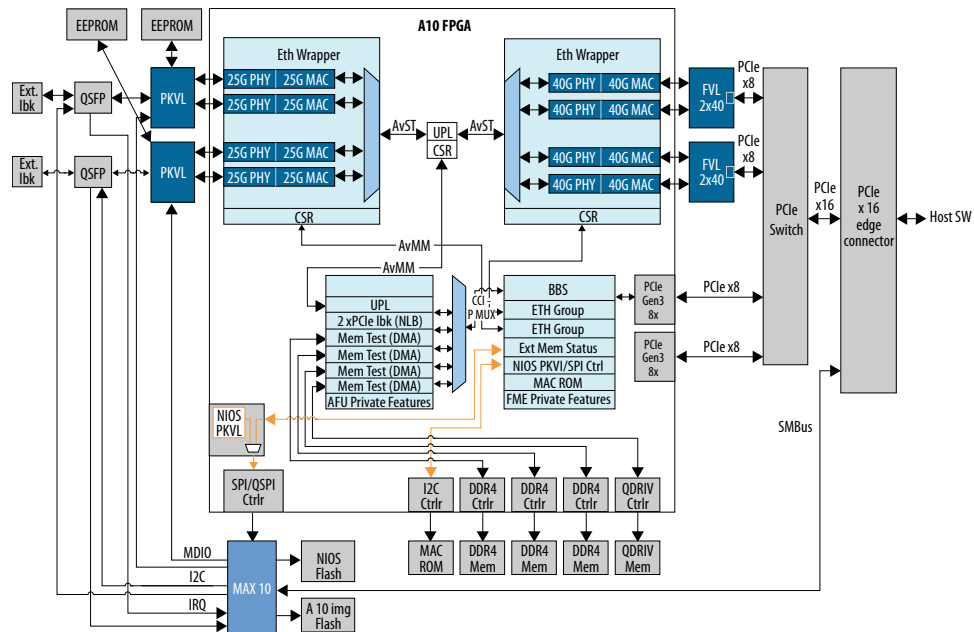
Page 0 of the flash contains the factory image. This image tests and diagnoses the Intel PAC N3000.

The factory image:

- Tests the image that enables PCIe, Ethernet, and memory diagnostics:
 - PCIe near-end loopback testing
 - Memory testing using DMA reads and writes
 - Ethernet loopback test
- Enables the RSU for the user image in flash

If the user image update fails, the Intel PAC N3000 restarts with the factory image, you can then reload the image.

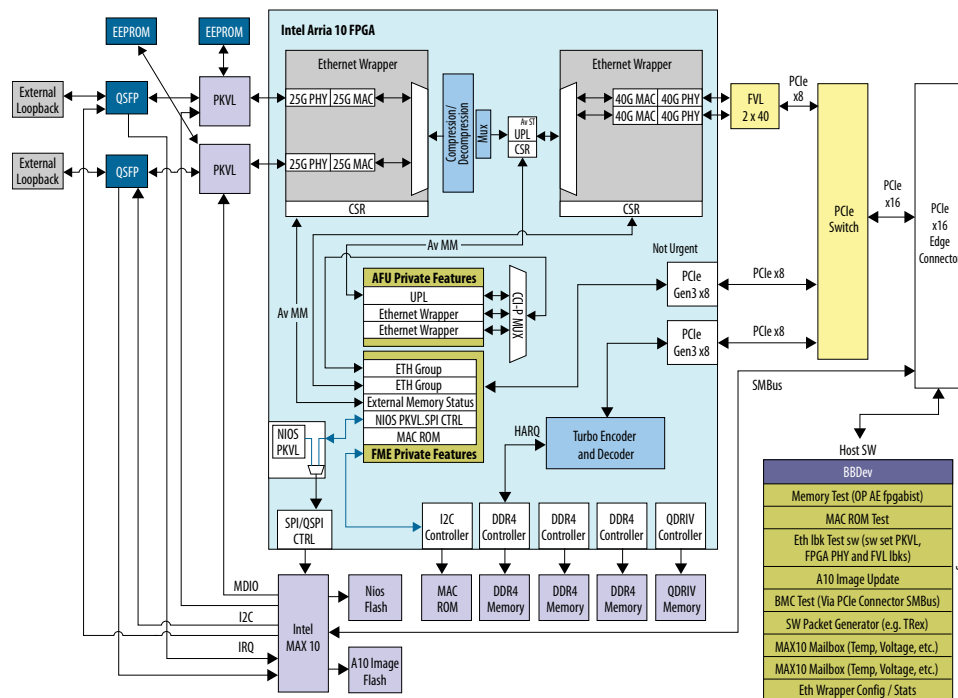
Figure 2. Factory Image Block Diagram for 2x2x25 GbE



2. 4G User Image Description

The user image performs fronthaul IO and 4G channel coding.

Figure 3. 4G User Image



2.1. User Image Power Management

On board power monitoring restricts the board temperature to 100°C. In the event of reaching this limit, the board is automatically shut down. The user image power consumption and thermal profile must fit within this envelope.

For different situations with different functions, the power consumptions are different. As a reference point, the raw power consumption of an FPGA is about 60 W @ 100°C junction temperature. The Intel PAC N3000 card power consumption is about 100 W.

2.2. 4G Channel Coder

You should send and receive code blocks for the encoder and decoder over PCIe to or from the host using the descriptor format defined in the Data Plane Development Kit (DPDK) and the baseband device (bbdev).

The channel coders queue and process these blocks based on the load balancing decisions.

Figure 4. 4G Channel Coder

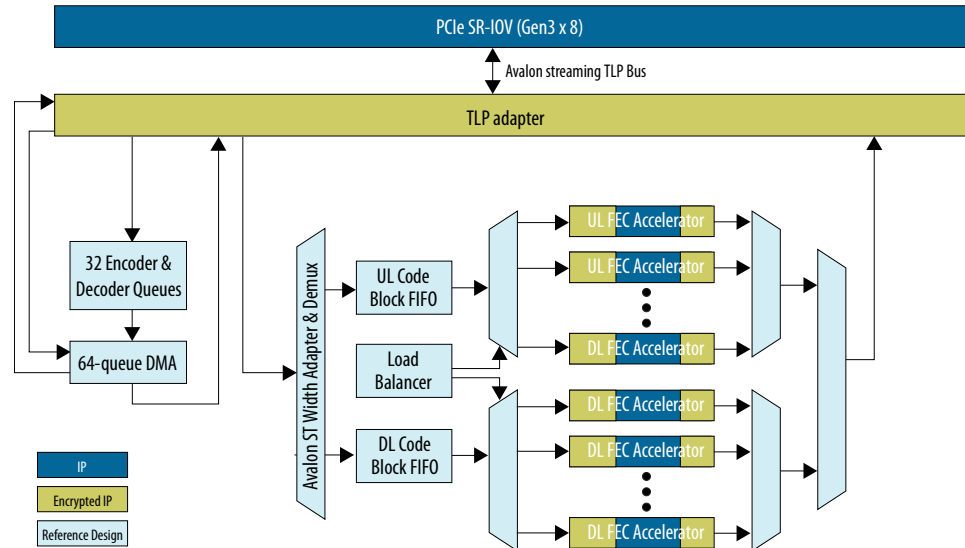


Figure 5. 4G Channel Downlink FEC Accelerator

The downlink FEC accelerator ((Intel Turbo-V FPGA IP)) consists of a code block CRC attachment block and a Turbo encoder (Intel Turbo FPGA IP) and rate matcher. The input data is 8-bit wide and the output data is 24-bit wide. The rate matcher consists of three interleavers and bit selector and bit collector.

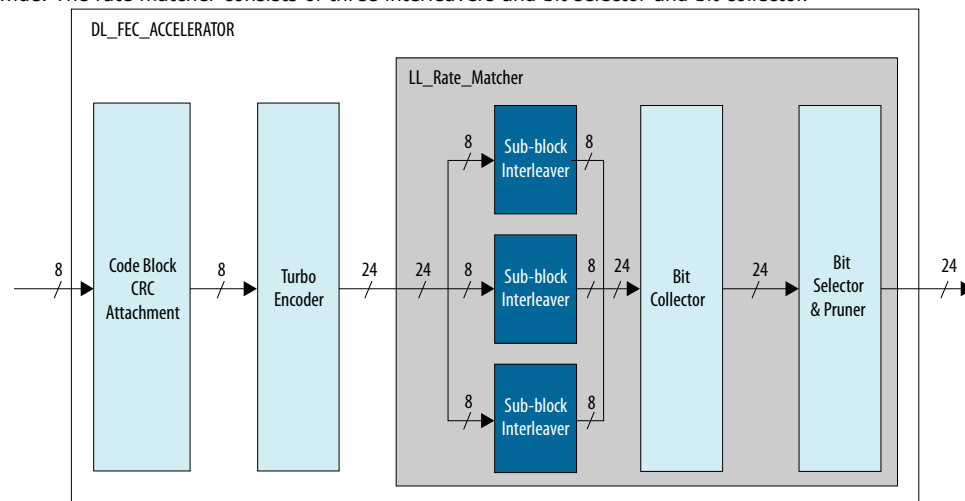


Figure 6. 4G Channel Uplink FEC Accelerator

The uplink FEC accelerator (Intel Turbo-V FPGA IP) consists of an interleaver and a turbo decoder (Intel Turbo FPGA IP).

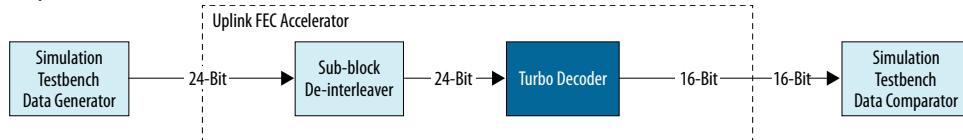


Figure 7. Deinterleaver

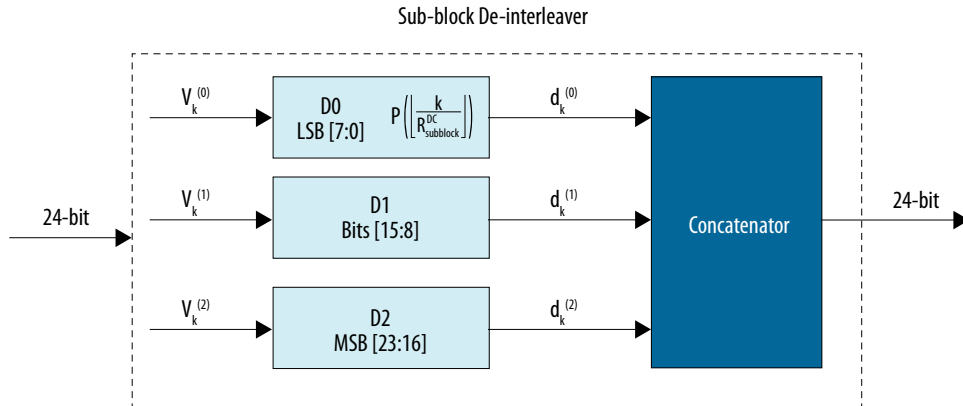
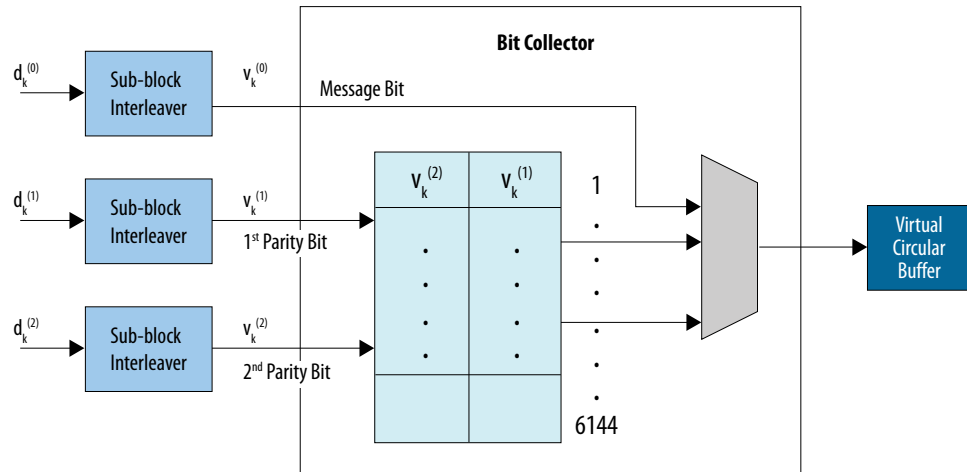


Figure 8. Bit Collector



Related Information

- [Data Plane Development Kit Website](#)
- [Baseband Device](#)
- [Turbo IP Core User Guide](#)

2.2.1. 4G Channel Code Throughput

The Intel FPGA PAC N3000 4G channel coder contains four encoders and six decoders. The throughput depends on the traffic model.

Baseline values include:

- Max code block size 6,144 downlink, 5,824 uplink
- Max transport block size 75,376 down and uplink
- 1 ms TTI
- Fmax = 275 MHz

Uplink throughput (decoding path) is $10 \times (75376 + 13 \times 24 + 24)$ bits in 500 μ s, which is 1.5142 Gbits/s @ 8 iterations. The decoders can decode $10 \times 13 = 130$ code-blocks of length 5,824 bits in 500 μ s @ 8 iterations

Downlink throughput (encoding path) is $10 \times 2 \times (75376 + 13 \times 24 + 24)$ bits in 333 μ s, which is 4.5473 Gbits/s. The encoders can encode $10 \times 2 \times 13 = 260$ code-blocks of length 6,144 bits in 333 μ s.

2.2.2. 4G Turbo-V Encoder and Decoder Tests

Intel tests the encoder and decoder by simulating with 5,100 test patterns. Approximately half of these place the rate matcher in bypass mode. Each test uses different values and sizes for E value and K size

Figure 9. Turbo-V Downlink Tests

The first 10 test cases.

Test	Test Case Na	Test Vector	E valu	K si	Test Case De
1	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_100_BS_Tx_CbMode/TIP_100_K_80_E_1632_Tx_0_2layers.txt	1632	80	DLTIP_Sanity
2	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_101_BS_Tx_CbMode/TIP_101_K_80_E_1376_Tx_0_2layers.txt	1376	80	DLTIP_Sanity
3	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_102_BS_Tx_CbMode/TIP_102_K_80_E_1120_Tx_0_2layers.txt	1120	80	DLTIP_Sanity
4	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_103_BS_Tx_CbMode/TIP_103_K_80_E_864_Tx_0_2layers.txt	864	80	DLTIP_Sanity
5	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_104_BS_Tx_CbMode/TIP_104_K_80_E_1536_Tx_0_2layers.txt	1536	80	DLTIP_Sanity
6	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_105_BS_Tx_CbMode/TIP_105_K_80_E_1440_Tx_0_2layers.txt	1440	80	DLTIP_Sanity
7	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_106_BS_Tx_CbMode/TIP_106_K_80_E_1344_Tx_0_2layers.txt	1344	80	DLTIP_Sanity
8	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_107_BS_Tx_CbMode/TIP_107_K_80_E_1632_Tx_0_2layers.txt	1632	80	DLTIP_Sanity
9	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_108_BS_Tx_CbMode/TIP_108_K_80_E_1376_Tx_0_2layers.txt	1376	80	DLTIP_Sanity
10	DL FEC wrapper	test_vectors/DLTIP_Sanity/DLTIP_Sanity/TIP_FEC_4G_test_109_BS_Tx_CbMode/TIP_109_K_80_E_1120_Tx_0_2layers.txt	1120	80	DLTIP_Sanity

Figure 10. Turbo-V Uplink Tests

The first 10 test cases.

Test	Test Case Na	Test Vector	Test Case De	K S	E V	Test status
1	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1000_BS_Rx_CbMode/TIP_1000_K_768_NMAP_0_E_5112_Rx_0.txt	ULTIP_Sanity	768	5112	Pass
2	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1001_BS_Rx_CbMode/TIP_1001_K_768_NMAP_0_E_5864_Rx_0.txt	ULTIP_Sanity	768	5864	Pass
3	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1002_BS_Rx_CbMode/TIP_1002_K_768_NMAP_0_E_7460_Rx_0.txt	ULTIP_Sanity	768	7460	Pass
4	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1003_BS_Rx_CbMode/TIP_1003_K_768_NMAP_0_E_4768_Rx_0.txt	ULTIP_Sanity	768	4768	Pass
5	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1004_BS_Rx_CbMode/TIP_1004_K_768_NMAP_0_E_7776_Rx_0.txt	ULTIP_Sanity	768	7776	Pass
6	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1005_BS_Rx_CbMode/TIP_1005_K_768_NMAP_0_E_6218_Rx_0.txt	ULTIP_Sanity	768	6218	Pass
7	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1006_BS_Rx_CbMode/TIP_1006_K_768_NMAP_0_E_7298_Rx_0.txt	ULTIP_Sanity	768	7298	Pass
8	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1007_BS_Rx_CbMode/TIP_1007_K_992_NMAP_0_E_6992_Rx_0.txt	ULTIP_Sanity	992	6992	Pass
9	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1008_BS_Rx_CbMode/TIP_1008_K_992_NMAP_0_E_5614_Rx_0.txt	ULTIP_Sanity	992	5614	Pass
10	UL FEC wrapper	ULTIP_Sanity/ULTIP_Sanity/TIP_FEC_4G_test_1009_BS_Rx_CbMode/TIP_1009_K_992_NMAP_0_E_7250_Rx_0.txt	ULTIP_Sanity	992	7250	Pass

Related Information

- [Turbo-V Intel FPGA IP User Guide](#)
- [Turbo IP Core User Guide](#)

2.2.3. 4G vRAN Universal Verification Methodology

The vRAN universal verification methodology (UVM) simulation test environment for the 4G channel coder incorporates the encoder and decoder and the DMA subsystem. The test environment does not include the preverified transaction layer packet (TLP) adapter

The tests use the same test patterns as in the encoder chain to test randomization (and functional coverage) of system scenarios such as PF and VF access, queue flushing, and reset. The reference design includes the UVM test plan, VLAN_UVM_Test_Plan.xls.

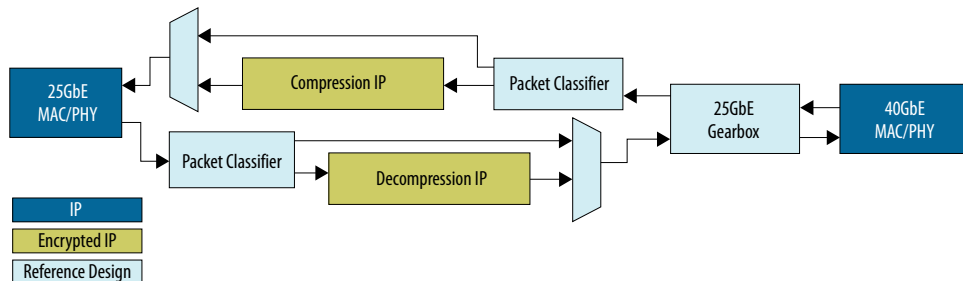
Figure 11. UVM Tests

Coverage Matrix		Verification Requirements
Sect: Title	Description	Constraints
1 Register		
1.1 access property	RW : read-write RO : read-only W1C : write 1 to clear detail refer to sect2.9 in design spec	Based on reg field operation. Every reg field will be tested in PF side. Besides, static configuration reg fields 'vf queue flush status write back address' and regs 'ring control registers' will also be tested in each VFs side, because they support VF write.
1.2 static configuration registers	Every field in this registers can be configured by PF, and they are read only in each VFs side, except 'queue flush status write back address' related fields. 'queue flush status write back address' related fields can be configured by each VFs and PF. detail refer to sect2.9.1 in design spec	Write reg fields with random value by each VFs and PF, and then read them back by each VFs and PF. The read data (except 'queue flush status write back address' related fields) should match the write data by PF (based on access property). The read data of 'queue flush status write back address' related fields should match the write data by the same Fid
1.3 queue mapping registers	Every field in these registers can be configured by PF, and they are read only in each VFs side. detail refer to sect2.9.2 in design spec	configure the registers to map the queue to random Fid by PF, read them back by each VFs and PF for check. PF: read data should match the write data VFX: read data should be 'hfff...fff'. set static configuration register field 'Queue-PF/VF mapping done' to 1, and then read Queue mapping registers back by each VFs and PF for check. PF: read data should match the write data VFX: read data should equal the queue ID if this queue is assigned to VFX, otherwise, the read data should be 'hfff...fff'. repeat above operations to make sure each Queue has been mapped to each

2.3. Fronthaul IO

The fronthaul IO is a simple passthrough pipe.

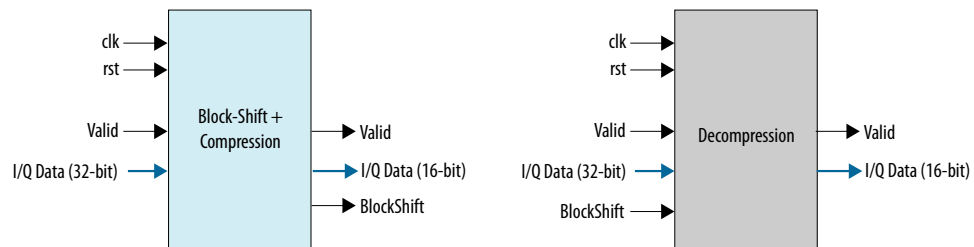
Figure 12. Fronthaul IO



2.3.1. O-RAN Compression and Decompression

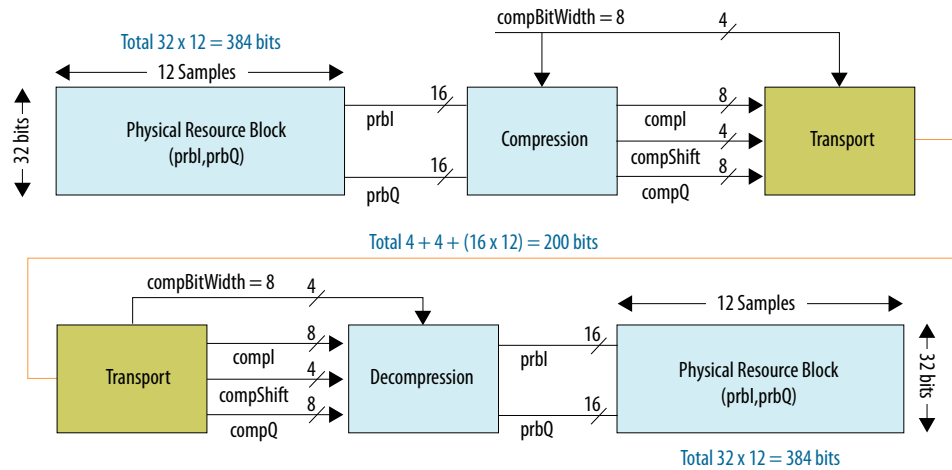
The compression and decompression IP supports both block floating point and Mu-Law compression methods.

Figure 13. Compression and Decompression IO



Internally, the design collects the 12 resource elements in a resource block and determines the maximum magnitude. It then performs block floating-point shifting and Mu-Law compression or decompression.

Figure 14. Compression and Decompression 16:8 bit Example



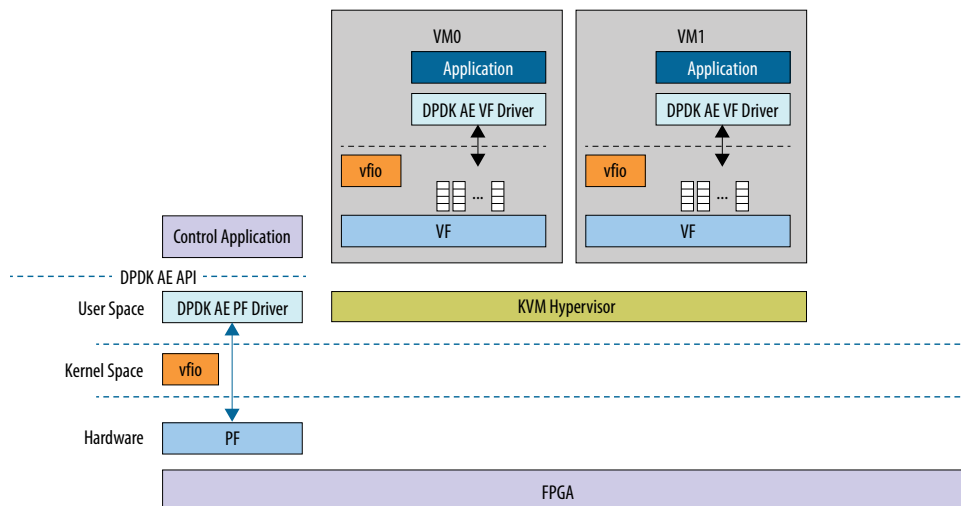
Related Information

ORAN Alliance

2.4. User Image Software

The user image requires software components and drivers, which support the descriptor format and physical function drivers.

Figure 15. User Image Architecture



3. Document Revision History for AN 908: Enabling 4G Wireless Acceleration in FlexRAN for the Intel® FPGA Programmable Acceleration Card N3000

Document Version	Changes
2020.01.30	Initial release.