ORAN.WG7.IPC-HRD-Opt8.0-v03.00
Technical Specification

O-RAN WG7

Hardware Reference Design Specification for Indoor Picocell (FR1) with Split Architecture Option 8

This is a re-published version of the attached final specification.

For this re-published version, the prior versions of the IPR Policy will apply, except that the previous requirement for Adopters (as defined in the earlier IPR Policy) to agree to an O-RAN Adopter License Agreement to access and use Final Specifications shall no longer apply or be required for these Final Specifications after 1st July 2022.

ORAN.WG7.IPC-HRD-Opt8.0-v03.00
Technical Specification

O-RAN WG7

Hardware Reference Design Specification for Indoor Picocell (FR1) with Split Architecture Option 8

O-RAN ALLIANCE e.V.
Buschkauler Weg 27, 53347 Alfter, Germany
Register of Associations, Bonn VR 11238
VAT ID DE321720189

Revision History

Date	Revision	Author	Description
07/12/2021	v03.00	WG7	Final Published Version

Contents

Tables

Figures

Chapter 1 Introductory Material

1.1 Scope

This Technical Specification has been produced by the O-RAN.org.

The contents of the present document are subject to continuing work within O-RAN WG7 and may change following formal O-RAN approval. Should the O-RAN.org modify the contents of the present document, it will be re-released by O-RAN Alliance with an identifying change of release date and an increase in version number as follows:

Release x.y.z

where:

- x the first digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have x=01).
- y the second digit is incremented when editorial only changes have been incorporated in the document.
- z the third digit included only in working versions of the document indicating incremental changes during the editing process. This variable is for internal WG7 use only.

The present document specifies system requirements and high-level architecture for the FR1 Picocell Indoor deployment scenario as specified in the Deployment Scenarios and Base Station Classes document [1].

In the main body of this specification (in any “chapter”) the information contained therein is informative, unless explicitly described as normative. Information contained in an “Annex” to this specification is always informative unless otherwise marked as normative.

1.2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

[1] ORAN-WG7.DSC.0-V01.00 Technical Specification, ‘Deployment Scenarios and Base Station Classes for White Box Hardware’. https://www.o-ran.org/specifications

[2] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[3] 3GPP TR 38.104: "NR; Base Station (BS) radio transmission and reception". http://www.3gpp.org/ftp//Specs/archive/38_series/38.104/38104-g10.zip

[4] ORAN-WG4.CUS.0-v03.00 Technical Specification, ‘O-RAN Fronthaul Working Group Control, User and Synchronization Plane Specification’. https://www.o-ran.org/specifications

[5] 3GPP TS 38.113:"NR: Base Station (BS) Electromagnetic Compatibility (EMC)". http://www.3gpp.org/ftp//Specs/archive/38_series/38.113/38113-f80.zip

[6] ORAN-WG7. IPC. HAR-v01.00 Technical Specification, ‘Indoor Pico Cell Hardware Architecture and Requirement Specification’. https://www.o-ran.org/specifications.

[7] ORAN.WG7. IPC-HRD-Opt7-2.0-v03.00 Technical Specification, 'Indoor Pico Cell BS Hardware Reference Design Specifications with Fronthaul Split Option 7-2 and FR1'. https://www.o-ran.org/specifications

1.3 Definitions and Abbreviations

1.3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [2]. For the base station classes of Pico, Micro and Macro, the definitions are given in 3GPP TR 38.104 [3].

Carrier Frequency: Center frequency of the cell.

F1 interface: The open interface between O-CU and O-DU_8 as defined by 3GPP TS 38.473 between CU and DU.

Integrated architecture: In the integrated architecture, the O-RU and O-DU_x are implemented on one platform. Each O-RU and RF front end is associated with one O-DU_8. They are then aggregated to O-CU and connected by F1 interface.

Split architecture: The O-RU_x and O-DU_x are physically separated from one another in this architecture. A switch may aggregate multiple O-RU_xs to one O-DU_x. O-DU_x switch and O-RU_xs are connected by the fronthaul interface as defined in WG4.

Transmission Reception Point (TRxP): Antenna array with one or more antenna elements available to the network located at a specific geographical location for a specific area.

1.3.2 Abbreviations

For the purposes of the present document, the abbreviations given in [2] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, as in [2].

7-2	Fronthaul interface split option as defined by O-RAN WG4, also referred to as 7-2x
3GPP	Third Generation Partnership Project
5G	Fifth-Generation Mobile Communications
5GC	5G Core
ACS	Adjacent Channel Selectivity
ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuit
ATA	Advanced Technology Attachment
BBDEV	Baseband Device
BH	Backhaul
BMC	Baseboard Management Controller
BPSK	Binary Phase Shift Keying
BS	Base Station

CISPR	International Special Committee on Radio Interference
CFR	Crest Factor Reduction
CU	Centralized Unit as defined by 3GPP
COM	Cluster Communication
CPRI	Common Public Radio Interface
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DAC	Digital to Analog Converter
DDC	Digital Down Conversion
DDR	Double Data Rate
DIMM	Dual-Inline-Memory-Modules
DL	Downlink
DPD	Digital Pre-Distortion
DPDK	Data Plane Development Kit
DSP	Digital Signal Processor
DU	Distributed Unit as defined by 3GPP
DUC	Digital Up Conversion
ECC	Error Correcting Code
eCPRI	evolved Common Public Radio Interface
EMC	Electro Magnetic Compatibility
EVM	Error Vector Magnitude
FCC	Federal Communications Commission
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FH	Fronthaul
FHGW	Fronthaul Gateway
FHM_x	Fronthaul Multiplexer with no FH protocol translation, supporting an $O\text{-}DU_x$ with split option x and an $O\text{-}RU_x$ with split option x, with currently available options 6→6, 7-2→7-2 and 8→8
$FHGW_{x\rightarrow y}$	Fronthaul Gateway that can translate fronthaul protocol from an $O\text{-}DU_x$ with split option x to an $O\text{-}RU_y$ with split option y, with currently available option 7-2→8.
FHHL	Full Height Half Length
FPGA	Field Programmable Gate Array
GbE	Gigabit Ethernet
GNSS	Global Navigation Satellite System
GPP	General Purpose Processor
GPS	Global Positioning System
HARQ	Hybrid Automatic Repeat request
HHHL	Half Height Half Length
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
IMD	Inter Modulation Distortion

I/O	Input/Output
JTAG	Joint Test Action Group
L1	Layer 1
LDPC	Low-Density Parity Codes
LRDIMM	Load-Reduced Dual In-line Memory Module
LTE	Long Term Evolution
LVDS	Low-Voltage Differential Signaling
MAC	Media Access Control
MCP	Multi-Chip Package
MH	Midhaul
MIG	Memory Interface Generator
MII	Media-Independent interface
MIMO	Multiple Input Multiple Output
MU-MIMO	Multiple User MIMO
NEBS	Network Equipment-Building System
NetConf	Network Configuration Protocol
NFV	Network Functions Virtualization
NIC	Network Interface Controller
NR	New Radio
O-CU	O-RAN Centralized Unit as defined by O-RAN
O-DU_x	A specific O-RAN Distributed Unit having fronthaul split option x where x may be 6, 7-2 (as defined by WG4) or 8
O-RU_x	A specific O-RAN Radio Unit having fronthaul split option x, where x is 6, 7-2 (as defined by WG4) or 8, and which is used in a configuration where the fronthaul interface is the same at the O-DU_x
OCXO	Oven Controlled Crystal Oscillator
PCIe	Peripheral Component Interface Express
PDCP	Packet Data Convergence Protocol
PHY	Physical Layer
PMBus	Power Management Bus
POE	Power over Ethernet
PPS	Pulse Per Second
PRACH	Physical Random Access Channel
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
QSFP	Quad Small Form-factor Pluggable
RAN	Radio Access Network
RDIMM	Registered Dual In-line Memory Module
RF	Radio Frequency
RoE	Radio over Ethernet
RU	Radio Unit as defined by 3GPP

1	RX	Receiver
2	SATA	Serial ATA
3	SDU	Service Data Unit
4	SFP	Small Form-factor Pluggable
5	SFP+	Small Form-factor Pluggable plus
6	SOC	System On Chip
7	SPI	Serial Peripheral Interface
8	SSD	Solid State Drive
9	TCXO	Temperature Compensate Crystal Oscillator
10	TDP	Thermal Design Power
11	TR	Technical Report
12	TS	Technical Specification
13	TX	Transmitter
14	UL	Uplink
15	USB	Universal Serial Bus
16	WG	Working Group

Chapter 2 Hardware Reference Design 1

This chapter describes a white box hardware reference design example for indoor Picocell deployment scenario. It includes O-CU, O-DU$_8$ O-RU$_8$, and FHM$_8$ for IPC deployment scenario.

2.1 O-CU Hardware Reference Design

The O-CU white box hardware is the platform that perform the O-CU function of upper l2 and l3. The hardware systems specified in this document meet the computing, power and environmental requirements of use cases configurations and feature sets of RAN physical node. These requirements are described in the hardware architecture and requirement specification [6] as well as in the use cases document [1]. The O-CU hardware includes the chassis platform, mother board, peripheral devices and cooling devices. The mother board contains processing unit, memory, the internal I/O interfaces, and external connection ports. The midhaul (MH) and backhaul (BH) interface are used to carry the traffic between O-CU and O-DU$_8$ as well as O-CU and core network. The other hardware functional components include: the storage for software, hardware and system debugging interfaces, board management controller, just to name a few; the O-CU designer will make decision based on the specific needs of the implementation.

The HW reference design of O-CU is the same as o-du$_8$ except for the need of HW accelerator, thus detail design will be described in o-du$_8$ section 2.2.

2.2 O-DU$_8$ Hardware Reference Design

The O-DU$_8$ white box hardware is the platform that performs the DU function of L1 and lower L2. The hardware systems specified in this document meet the computing, power and environmental requirements of use cases configurations and feature sets of RAN physical node. These requirements are described in the early hardware requirement specification as well as in the use cases document. The O-DU$_8$ hardware includes the chassis platform, mother board, peripheral devices and cooling devices. The mother board contains processing unit, memory, the internal I/O interfaces, and external connection ports. The fronthaul and backhaul interface are used to carry the traffic between O-RU$_8$/FHM$_8$ and O-DU$_8$ as well as O-CU and O-DU$_8$. The O-DU$_8$ design may also provide an interface for hardware accelerator if that option is preferred. The other hardware functional components include: the storage for software, hardware and system debugging interfaces, board management controller, just to name a few; the O-DU$_8$ designer will make decision based on the specific needs of the implementation.

Note that the O-DU$_8$ HW reference design is also feasible for O-CU and integrated O-CU/ O-DU$_8$.

2.2.1 O-DU$_8$ High-Level Functional Block Diagram

Figure 2-1 shows the major functional blocks of O-DU$_8$. The digital processing unit handles the baseband processing workload. To make the processing more efficient, an accelerator can be used to assist with the baseband workload processing. The memory devices include the random-access memory (RAM) for temporary storage of data while flash memory is used for codes and logs. The storage device is for persistent storage. The external network cards can be used

for fronthaul or backhaul connection. The baseboard management controller (BMC) is a microcontroller which monitors hardware operation on motherboard. The clock circuits provide digital processing unit with required clock signals.

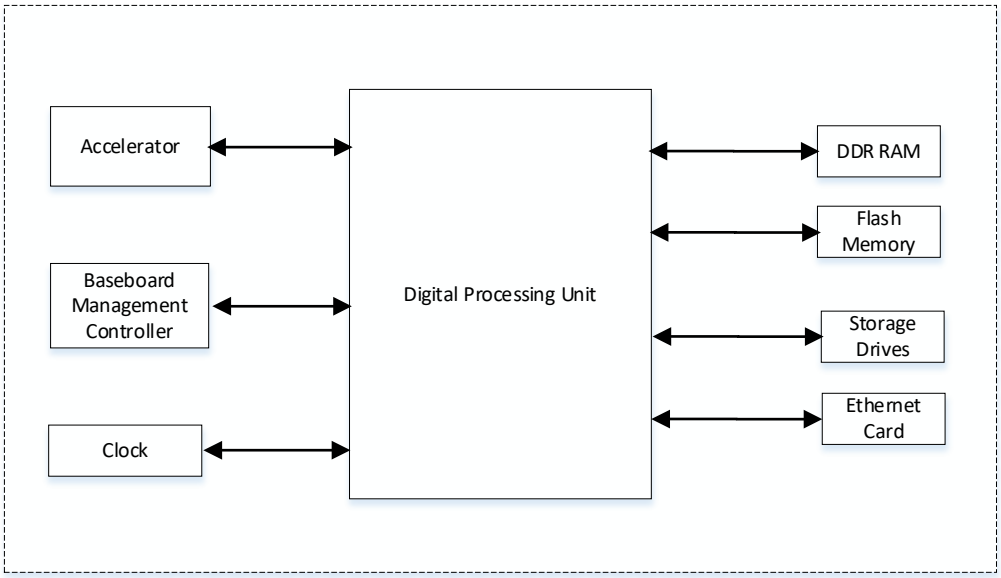


Figure 2-1: O-DU8 Functional Block Diagram.

2.2.2 O-DU8 Hardware Design Description

Figure 2-2 describes the components and connections inside the O-DU8 white box.

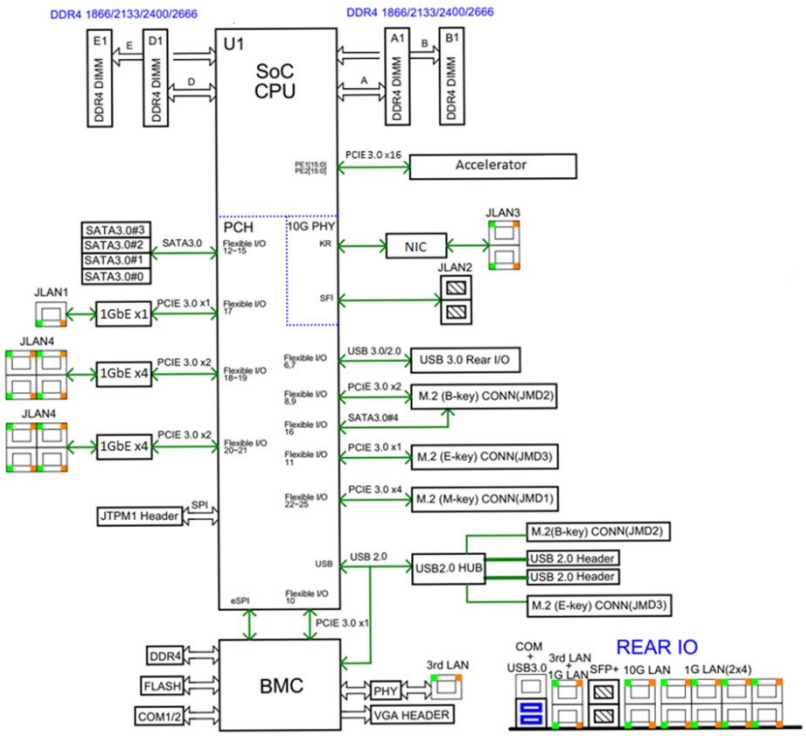


Figure 2-2: O-DU$_8$ Hardware Block Diagram

As described in the previous section, the O-DU$_8$ hardware can be implemented with difference design choices. Here, a system-on-a-chip (SoC) based design is presented as an example which processes most of the O-DU$_8$ workload. The accelerator can be used to perform some O-DU$_8$ workload functions based on the overall performance requirement. Several Ethernet controllers are used for front haul link, back haul link and remote console control connection. The other parts include: RAM, flash memory, and hard drive storage. The JTAG and USB ports are provided for hardware debug and local connection if needed. Figure 2-2 describes the components and connections of O-DU$_8$ hardware. Finally, BMC block is mainly responsible for monitoring the hardware status.

2.2.3 O-DU$_8$ Hardware Components

In this section, the details of the O-DU$_8$ hardware component's requirements, features and parameters are described. The components selection is based on the use case requirements which are listed in the hardware architecture and requirements document [6].

2.2.3.1 Digital Processing Unit

This example of the digital processing unit in O-DU$_8$ is based on the General Purpose Processor (GPP).

a. Digital Processing Unit Requirement

The GPP requirements are listed in the following table.

Table 2-1: The Processor Feature List

Item Name	Description
# of Cores	16
# of Threads	32
Base Frequency	2.20 GHz
Max Turbo Frequency	3.00 GHz
Cache	22 MB
Thermal Design Power (TDP)	100W
Max Memory Size (dependent on memory type)	512 GB
Memory Types	DDR4
Max # of Memory Channels	4

The interface specifications on the main board are as follows:

Memory Channel Interfaces: The system memory capacity, type and related information are described in the following table.

Table 2-2: The Memory Channel Feature List

Item Name	Description
Memory Types	DDR4
# of Memory Channels	4
ECC LRDIMM	Up to 512GB
ECC RDIMM	Up to 256GB
Memory Speed	2666/2400/2133MHz

DIMM Sizes	128GB, 64GB, 32GB, 16GB
Memory Voltage	1.2 V

PCIe: PCIe Gen 3 should be supported by the processor. There are total of 32 PCIe lanes with 128 Gb/s bandwidth. The 32 PCIe lanes can be divided into two x16 slots by using a riser card.

Ethernet: The system should be capable to offer aggregated 48 Gb/s Ethernet bandwidth. The breakout the ports are discussed in later section. When higher Ethernet bandwidth required, an Ethernet card can be installed in one of the PCIe slot.

b. Digital Processing Unit Design

The digital processing unit is a SoC device which is a 64-bit multi-core server class processor. This SoC includes an integrated Platform Controller Hub (PCH), integrated high-speed I/O, Integrated Memory Controllers (IMC), and four integrated 10 Gigabit Ethernet ports.

The SOC supports 512-bit wide vector processing instruction set. It also supports hardware virtualization to enable dynamic provisioning of services as communication service providers extend network functions virtualization (NFV). Figure 2-2 shows the major functional blocks of the digital processing unit.

2.2.3.2 Hardware Accelerator

Hardware accelerators can be used in O-DU$_8$ to offload computationally intensive functions and to optimize the performance under varying traffic and loading conditions. The acceleration functional requirements and implementation are system designer's choice; however, the O-DU$_8$ shall meet the minimum system performance requirements under various loading and deployment conditions. In most cases, a Field Programmable Gate Array (FPGA) or Application Specific Integrated Circuit (ASIC) based PCIe card can be used to optimize the system performance. The FPGA(s) are part of a Network Interface Controller (NIC) that further provides connectivity services.

2.2.3.2.1 Accelerator Design Solution 1

The O-DU$_8$ system is typically implemented using a multi-core processor and one or more hardware accelerators. Parts of O-DU$_8$ protocol stack can be implemented in software running on the multi-core processors, some of the computationally intensive L1 and L2 functions are offloaded to FPGA-based or similar hardware accelerators. This is a programmable hardware, which provides both flexibility and high computing capabilities.

a. Accelerator Requirements

The accelerator unit comprises one or more FPGAs (e.g., two FPGAs), sufficient amount of DDR4 memory, and synchronization circuitry where one of FPGAs is used for L1 functional offload and the other one is used to perform fronthaul connectivity functions/protocols. The FPGA for L1 offload uses dedicated cores for channel encoding/decoding as well as FPGA and processing resources for running L1 functions such as but not limited to rate matching and de-matching, interleaving and scrambling, demodulation and HARQ buffer management as well as OFDM modulation/demodulation and channel estimation.

Key features of the FPGA-based accelerator include:

- 2X10/25G CPRI or RoE fronthaul interface
- Built-in SyncE/IEEE1588v2 synchronization + external reference timing
- L1 offloading options
  - LDPC encoding and decoding
  - Polar encoding and decoding

- HARQ management with on board DDR memory (including DDR controller and interfaces)
- Other L1 offloading candidates include PRACH detection, MIMO encoding and decoding, channel estimation
- Partial or full L1 functions can be offloaded. It is recommended to offload the user-plane channel coding chain and part of or the entire control-plane channel coding chain to the hardware accelerator.

- PCIe Gen3 x16, two Gen4 x8, or PCIe Gen3 x16 bifurcated to two Gen3 x8
- GPP supported
- Standard PCIe FHHL card (It is assumed that the hardware accelerators further perform NIC functions).
- 8GB DDR4 memory for buffering
- Power consumption not to exceed 75W

The accelerator requirements in hardware have been summarized inTable 2-3.

Table 2-3: Accelerator Hardware Component List

Item Name	Description for FPGA 1	Description for FPGA 2
SoC Resources	System Logic cells - 930K CLB LUT - 425K SDFEC -8 DSP Slices - 4,272 BRAM - 38.0Mb URAM - 22.5Mb	System Logic cells - 1,143K CLB LUT - 523K CLB Flip-Flops -1,045K DSP Slices - 1,968 BRAM - 34.6Mb URAM - 36.0Mb
Form Factor	FHHL PCIe Form Factor	
PCIe Interface	x8 Gen1, Gen2, Gen3 interface to FPGA2 and x8 Gen1, Gen2, Gen3 interface to FPGA1 have x16 to two x8 bifurcation	
On Board Memory	FPGA1	FPGA2
	Total Capacity 4 GB in PL, upgradeable to 8GB Total Capacity 2 GB in PS, upgradeable to 4GB	Total Capacity 4 GB in PL, upgradeable to 8GB Total Capacity 2 GB in PS, upgradeable to 4GB
Network Interface(s)	N/A	2xSFP28 optical interfaces to FPGA2 (User Configurable, includes 10/25 Ethernet)
Other External Interface(s)	Micro USB for JTAG support (FPGA programming and debug) and access to BMC	
Graphical User interface	GUI for monitoring the basic board parameters, monitoring temperature alerts, firmware upgrades for BMC	

Board Management Controller	• Power sequencing and reset • Field upgrades • FPGA configuration and control • Clock configuration • I2C bus access • USB 2.0
Operating Temperature	-5°C - 55°C
Power	< 75 W
Clocking Options	Low-Jitter, configurable clock ranging from10MHz to 750MHz 1 PPS input and output with assembly option for OCXO and TCXO

The following are the accelerator requirements in function and interface.

Functional offload requirements

One of the candidate functions for offloading is the LDPC encoder and decoder, which typically consists of computationally intensive and relatively highly power-consuming functions. It must be noted that neither software implementation in CPU nor soft FPGA logic implementation would provide a highly power-efficient solution while meeting/exceeding 3GPP NR user-plane encoder/decoder throughput and latency requirements, rather a hardened implementation of the FEC functions would be very power efficient. Downlink and uplink throughputs of up to 40Gbps and 18Gbps, respectively, are shown feasible with this architecture. Other candidate L1 functions for acceleration include CRC generation, segmentation, bit-level/sub-block interleaving and scrambling as well as FFT/IFFT processing, for which an FPGA can be used.

For other symbol processing L1 functions, which require heavy multiply and accumulation operations, FPGAs 1 and 2 have DSP blocks that can efficiently perform these operations. Polar encoding and/or decoding on the control-plane can also be offloaded to FPGA1 resulting in high throughputs and low latencies.

Interface requirements

- PCIe: PCIe interface is widely used to provide interface between the GPP and hardware accelerators. FPGA devices have dedicated PCIe hard IP which facilitate implementation and quick setup of PCIe interface. They support both PCIe Gen3 x16 or PCIe Gen4 x8, which allow the FPGA device to interface with any GPP supporting either PCIe Gen3 x16 or PCIe Gen4 x8 interface.

- Fronthaul: FPGA devices can support various speed grades and any fronthaul protocols, customers can use off-the-shelf CPRI or RoE IPs to quickly implement and configure any fronthaul interface protocol.

- Serial transceivers: FPGA devices have SerDes resources to implement various connectivity speeds (e.g., 33 Gb/s) per SerDes, 10G/25G CPRI, 10G/25G/50G/100G Ethernet connections can use these SerDes resources.

- Ethernet MAC speed: FPGA devices have hardened implementations of Ethernet MAC that support speeds of 100 Gb/s and above. The Ethernet MAC IP allows power-efficient implementation of high speed Ethernet connectivity. In the example shown in Figure 2-3, FPGA 1 and FPGA 2 can use the hard 100Gbps MAC IPs to connect each other, allowing the L1 and fronthaul functions to be distributed across these two FPGAs with less

connectivity overhead. For other Ethernet MACs such as 10G/25G Ethernet, they provide soft Ethernet MAC IPs, so when implementing CPRI or RoE fronthaul functions, 10G/25G Ethernet MAC can be used.

b. Accelerator Design

The hardware accelerator supports GPP. Figure 2-3 illustrates a two-chip acceleration architecture comprising two FPGAs with multi-lane PCIe interfaces toward the CPU and external connectivity toward O-RU$_8$(s) via CPRI/RoE and O-CU(s) through GbE connectivity. The example architecture further depicts multi-lane Gen3 or Gen4 PCIe interfaces between each FPGA and the CPU. The FPGAs communicate through high-bandwidth Ethernet (GbE) transport.

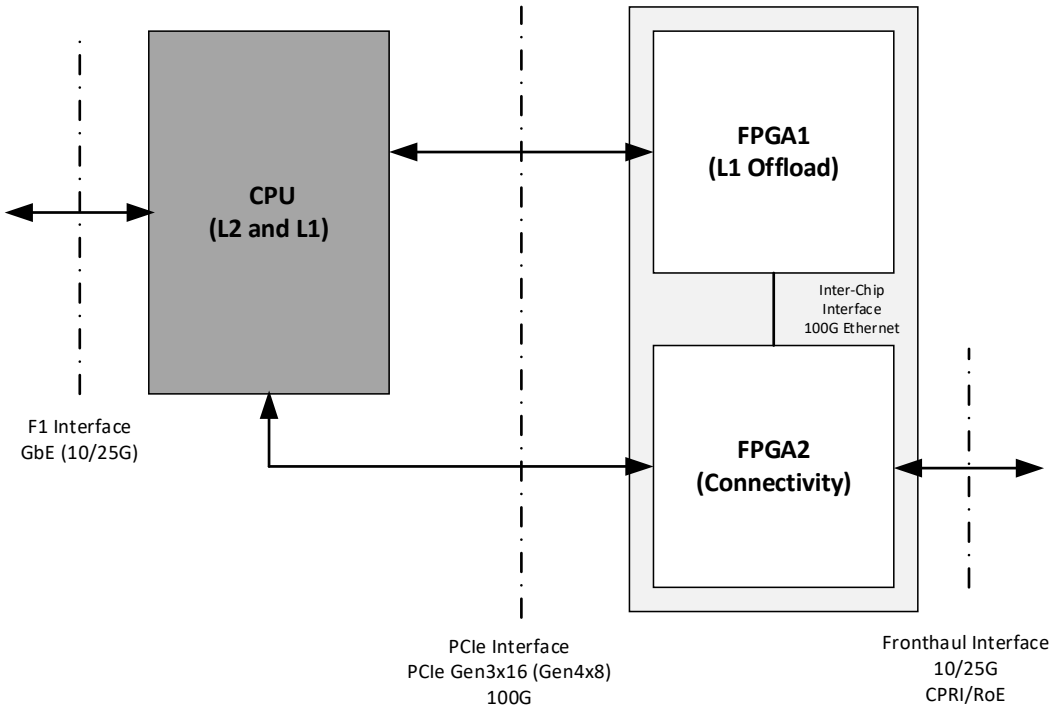


Figure 2-3: Example 2-chip FPGA-based Hardware Acceleration in O-DU$_8$

2.2.3.2.2 Accelerator Design Solution 2

This section illustrates a one-chip acceleration architecture comprising one FPGA with PCIe interfaces toward the CPU and external connectivity toward O-RU$_8$(s). The example architecture further depicts Gen3 PCIe interfaces (for example, or Gen4) between FPGA and the CPU.

An example one-chip FPGA-based hardware acceleration architecture is shown in Figure 2-4.

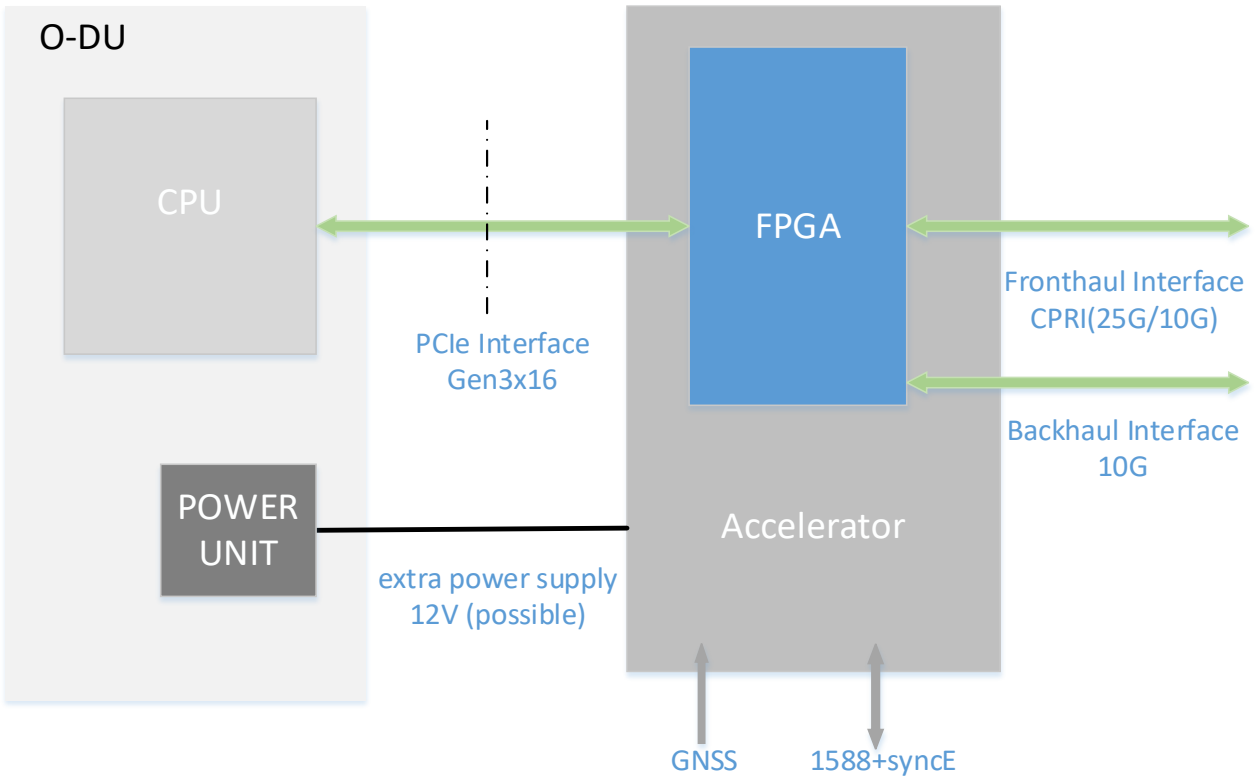


Figure 2-4: Example of 1-chip FPGA-based Hardware Acceleration in O-DU$_8$

a. Accelerator Requirement

The accelerator unit comprises one or more FPGAs (e.g., two FPGAs), sufficient amount of DDR4 memory, and synchronization circuitry where one of FPGAs is used for L1 functional offload and the other one is used to perform fronthaul connectivity functions/protocols. The FPGA for L1 offload uses dedicated cores for channel encoding/decoding as well as FPGA and processing resources for running L1 functions such as but not limited to rate matching and de-matching, interleaving and scrambling, demodulation and HARQ buffer management as well as OFDM modulation/demodulation and channel estimation.

Key features of the FPGA-based accelerator include:

- 4X25G/10G CPRI optical interface
- 1x10G Ethernet optical interface
- Support GNSS/IEEE1588v2+syncE reference timing
- L1 offloading options
  - LDPC encoding and decoding
  - HARQ management

- Other optional L1 offloading such as FFT/IFFT, CP, CRC, rate matching, PRACH detection.

- PCIe Gen3X16, 4 hard IP blocks
- FHHL form factor and operation temperature from -5℃ to +55℃
- 2GB DDR4 memory for Buffering
- Specific power supply design is required with 1-chip FPGA, since the maximum power consumption of PCIe specification (75w) could be exceeded, additional external auxiliary power may be required.

Functional offload requirements

- L1 offloading: One of the functions for L1 offloading is the LDPC encoder and decoder, which typically consists of computationally-intensive and relatively highly power-consuming functions. Other L1 functions for acceleration include CRC generation, CP, PRACH detection and rate matching as well as FFT/IFFT processing.

- Fronthaul: The data transmission and protocol processing of fronthaul interface is completed on the FPGA device.

Interface requirements

- PCIe: PCIe interface is widely used to provide interface between the CPU and hardware accelerators. FPGA device has dedicated PCIe hard IP which facilitate implementation and quick setup of PCIe interface. FPGA support PCIe Gen3 x16, which allow the FPGA device to interface with any CPU supporting either PCIe Gen3 x16 interface.

- Fronthaul: FPGA device can support various speed grades and CPRI protocol for fronthaul interface between O-DU_8 and O-RU_8.

- Backhaul: FPGA device can support backhaul interface between O-DU_8 and core network function with an 10G optical interface.

- Synchronization: Accelerator has external interfaces to support synchronization, including GNSS, IEEE1588v2 and SyncE reference timing.

b. Accelerator Design

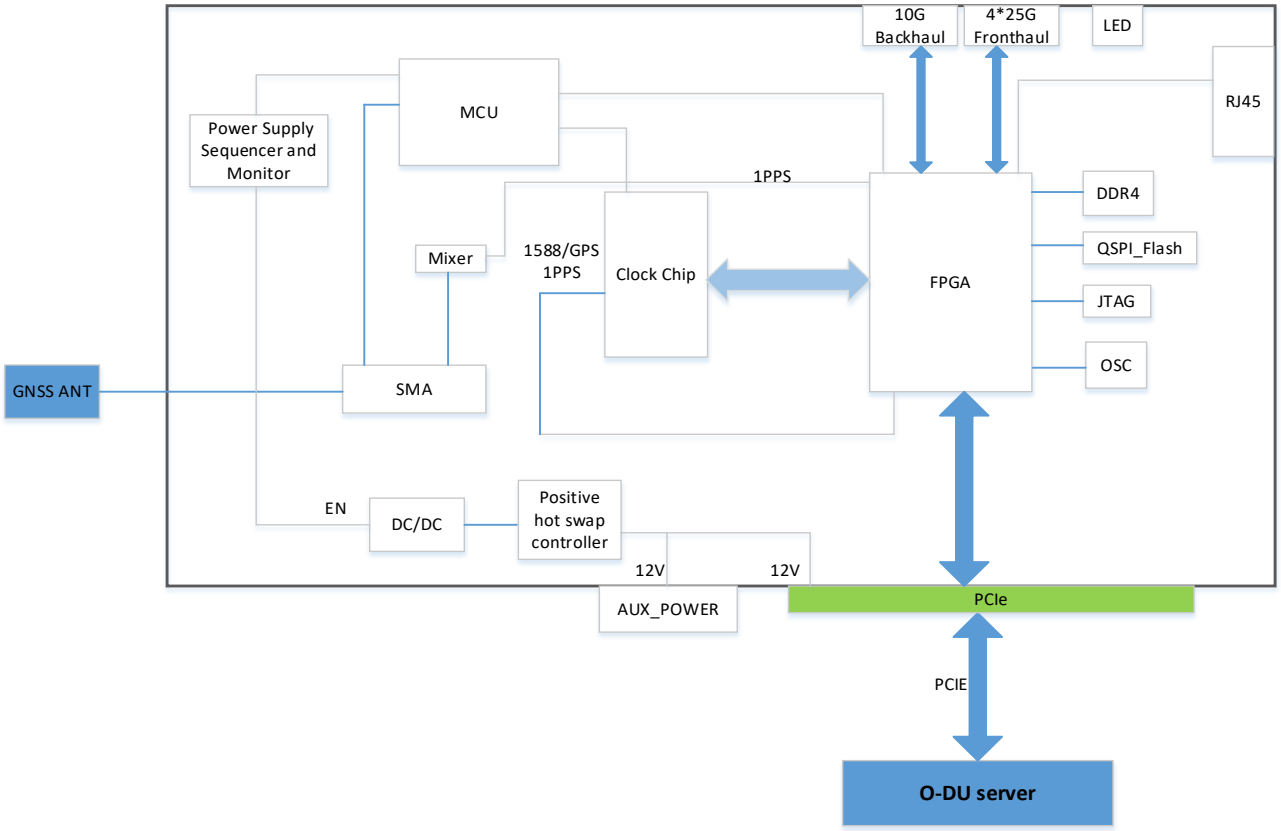


Figure 2-5: Example of Accelerator Design

Figure 2-5 shows the accelerator design, description of block diagram of board card:

12V power: The maximum power consumption of the whole card is estimated to exceed 75W. The PCIe cannot meet the power consumption demand, so it needs an external 12V auxiliary power supply. The extra power consumption can be included by the power unit of O-DU$_8$. Only power cord is needed, and it has no effect on the whole machine design. This part is not necessary if there are future optimizations.

DC/DC: The module contains all the power sources used in the system.

Power Supply Sequencer and Monitor: Since the system has the power on timing requirement, this module controls the output of each DC/DC to realize the normal power on/off of the system.

Mixer and GNSS antenna: The system uses an external antenna, which can be placed outdoors to improve the receiving sensitivity of synchronizing signal. After mixer's processing of the received signal, the signal source of 1pps is output to clock chip for system input synchronous clock signal.

MCU (micro control unit): As the auxiliary processor of the board card, MCU realizes the temperature detection of the board card, the operation state indication and the configuration of power monitor and clock chip in the initialization process, and real-time monitoring of the working state of each module, exception handling and other functions.

Clock chip: The chip realizes system-level clock input, which can synchronously output all kinds of clocks for clock synchronization of subordinate modules.

FPGA: As the core chip of the board card, FPGA forwards the data of the downlink of the core network and the uplink data gathered at the O-RU$_8$(s), and interacts with the O-DU$_8$ server through PCIe x16 to complete the calculation of 5G NR data.

2.2.3.2.3 Accelerator Design Solution 3

Channel coding for LDPC and fronthaul compression requires a significant amount of bit level processing and is well suited to a fine-grained FPGA architecture and/or low cost/power structured ASIC. Options include:

- Look-aside FEC: LDPC (de)coding, Polar (de)coding, Rate (De)Matcher, (De) Interleaver, CRC, HARQ
- L1 Processing: (i)FFT & CP+/-, PRACH, channel estimation & beamforming
- Bump-in-wire Fronthaul: compression / decompression for latency and bandwidth reduction

a. Accelerator Requirement

Hardware requirements are listed in Table 2-4.

Table 2-4: Accelerator Hardware Feature List

Item Name	Description
PCIe (Interface with digital processing unit)	Gen4 x16 (and lower)
Form factor	FHHL
Connectivity	2x QSFP28/56
FPGA	Logic Elements: 1437K M20K Memory: 139Mb

	Quad-core GPP Hard Processor Sub-system
NIC Device	100Gb xHAUL for FH, BH & MH traffic shaping. Optional FPGA co-processing.
DDR Main	8+8+1GB DDR4
Flash (FPGA images)	>=1 Gbit
BMC	Telemetry, Security, remote upgrade
Clocking	For O-RAN C1, C2, C3 & C4
Fronthaul	CPRI
GPS	SMA for 1PPS & 10MHz (in/out)
Operating Temperature (ambient)	NEBS Compliant
Power	<75W (without optional NIC device)
Clock Accuracy	Low-Jitter, configurable clock ranging from 10MHz to 750MHz. Option for OCXO (TCXO as standard)

Firmware Requirements are listed in Table 2-5.

Table 2-5: Accelerator Firmware Feature List

Item Name	Description
Remote system upgrade	Securely upgradable FPGA flash image
Queuing	64 Queues supported equally split between UL & DL.
LDPC Acceleration	NR LDPC Encoding with, interleaving and rate-matching. NR LDPC Decoding with sub-block de-interleaving function of reverse rate matching. Early Termination, CRC attachment and HARQ buffering. 5G Throughput: DL 14.8Gbps, UL 3.2Gbps
Load Balancing (channel coding)	Load balancer distributes the pending encoder/decoder requests to encoder/decoders
Descriptor Format (channel coding)	Code block based interface. Software enablement by BBDEV API (DPDK) https://www.dpdk.org/ https://doc.dpdk.org/guides/prog_guide/bbdev.html
Fronthaul Compression	In-line compression/decompression for Mu-Law, block-floating point and quantization according to the O-RAN WG4 specification.
Open programmable acceleration	Support for:

environment	• FPGA Flashing upgrade • Firmware version reporting • PCIe diagnostics • Ethernet diagnostics • Temperature and voltage telemetry information

b. Accelerator Design

Accelerator design solution 3 has 2 options as shown in the figures below.

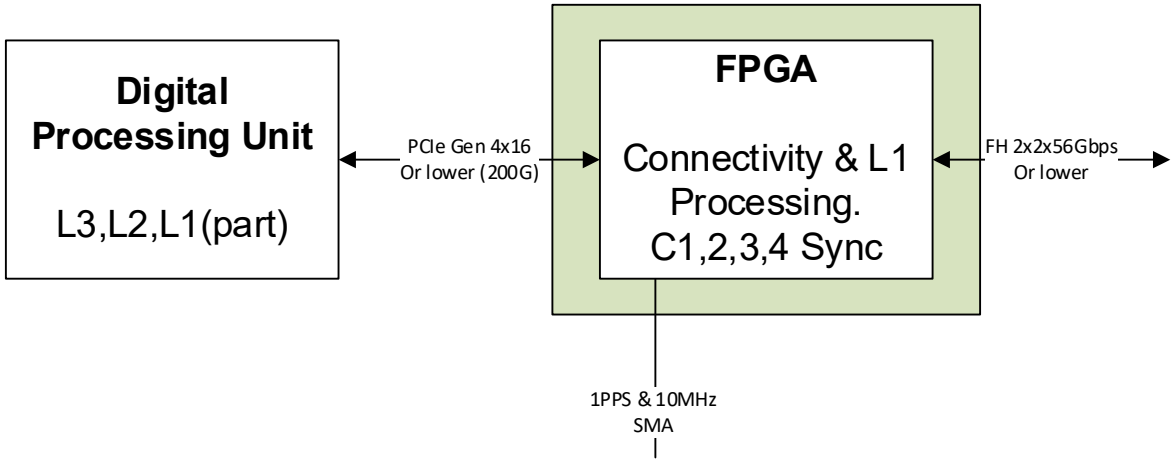


Figure 2-6: Accelerator Design 1 without optional NIC Device

The accelerator with optional NIC device.

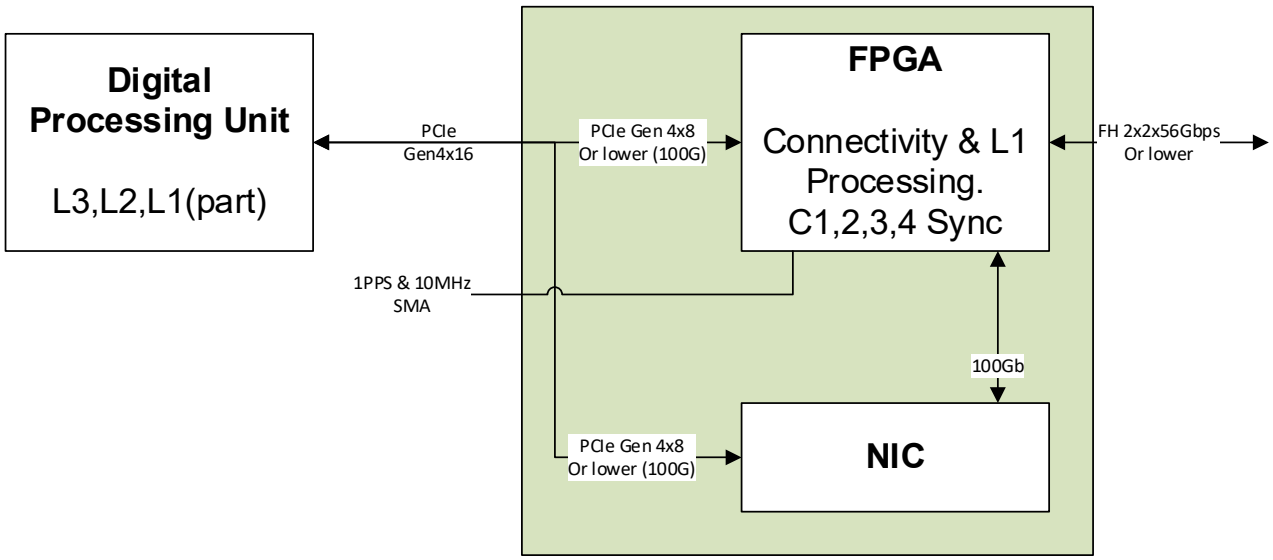


Figure 2-7: Accelerator Design 1 with optional NIC Device

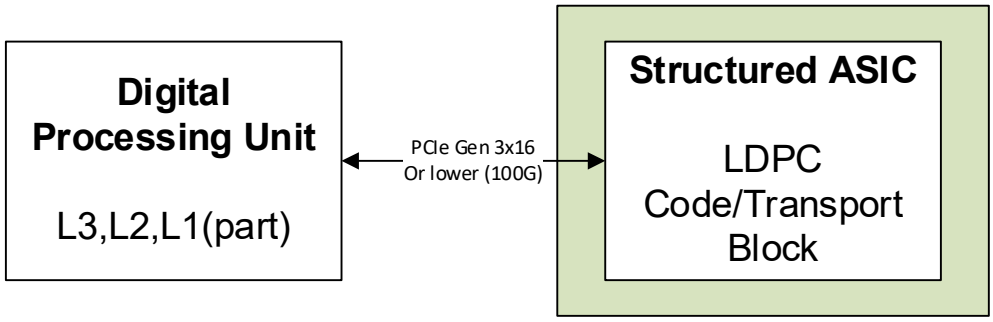


Figure 2-8: Structured ASIC Accelerator Design

2.2.3.2.4 Accelerator Design Solution 4

Channel coding for LDPC and fronthaul compression requires a significant amount of bit level processing and is well suited to a fine-grained structured ASIC. Features include:

- Look-aside FEC: Turbo (de)coding, LDPC (de)coding, Rate (De)Matcher, (De) Interleaver, CRC, HARQ retransmission & (de)interleaver.

a. Accelerator Requirement

Hardware features are listed in Table 2-6.

Table 2-6: Accelerator Hardware Feature List

Item Name	Description
PCIe	Gen3 x16
Form factor	HHHL
DDR	DDR4 (64-bit +ECC), 2667Mbps Interface for HARQ buffering
Board Management Controller	Telemetry, Security.
Power	<35W

Firmware Requirements are listed in Table 2-7.

Table 2-7: Accelerator Firmware Feature List

Item Name	Description
Queuing	64 Queues supported equally split between UL & DL.
LDPC Acceleration	NR LDPC Encoding with, interleaving and rate-matching. NR LDPC Decoding with sub-block de-interleaving function of reverse rate matching.

	Early Termination, CRC attachment and HARQ buffering. 5G Throughput: DL 23Gbps, UL 8Gbps
Load Balancing (channel coding)	Load balancer distributes the pending encoder/decoder requests to encoder/decoders
Descriptor Format (channel coding)	Code block and transport block based interface. Software enablement by BBDEV API (DPDK) https://www.dpdk.org/ https://doc.dpdk.org/guides/prog_guide/bbdev.html

b. Accelerator Design

The following diagram shows the structured ASIC based accelerator design.

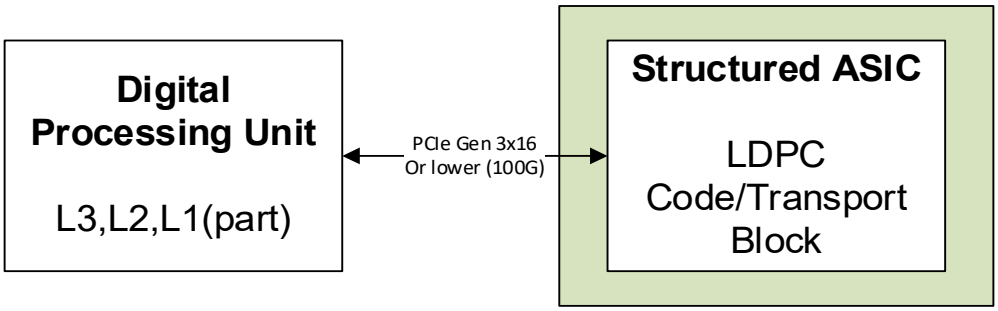


Figure 2-9: Structured ASIC Accelerator Design

2.2.3.3 Baseboard Management Controller

BMC is used to perform hardware power control (power on, power off and power cycle), monitor hardware status (temperatures, voltages, etc.), monitor Basic I/O System (BIOS)/ Unified Extensible Firmware Interface (UEFI) firmware status, and log system events. It provides remote access via shared or dedicated NIC. System user can do console access via serial or physical/Kernel-based Virtual Machine (KVM). The BMC has dedicated RAM and flash memory. It provides access via serial port or Ethernet port. Figure 2-2 describes the BMC connections with related components.

2.2.4 Synchronization and Timing

2.2.4.1 Synchronization and Timing Design 1

This section describes the synchronization and timing mechanism that is used in the O-DU[8].

a. Hardware Requirements

The O-DU$_8$ shall support following timing synchronization methods:

1. GPS Synchronization
2. Ethernet based IEEE1588v2 Synchronization
3. BeiDou Synchronization
4. BeiDou and GPS switching

b. ***Hardware Design***

Depending on the timing distribution topologies used, the O-DU$_8$ system clock is able to synchronize with the Grand Master Clock (GMC) using IEEE1588 via either the front haul NIC or backhaul NIC. O-DU$_8$ can also synchronize timing using Global Navigation Satellite System (GNSS). In the case of IEEE1588, the Physical Hardware Clock (PHC) within the NIC is synchronized with the GMC first, then the O-DU$_8$ system clock is synchronized with the PHC. The O-DU$_8$ is also capable to provide clock to the O-RU$_8$ via front haul if needed. When GNSS becomes available to O-DU$_8$, it will be able to synchronize the system clock to Coordinated Universal Time (UTC). Figure 2-10 outlines the O-DU$_8$ timing synchronization mechanisms.

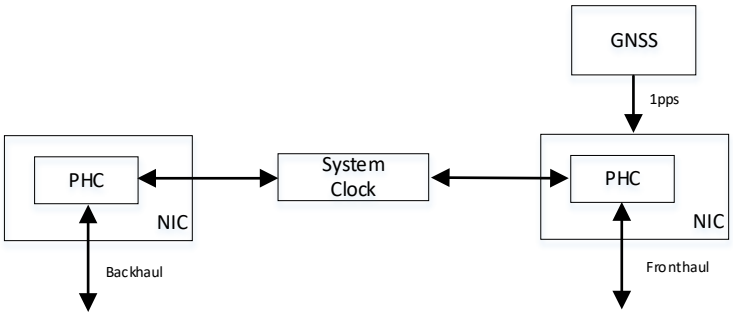


Figure 2-10: O-DU$_8$ Timing Synchronization

2.2.4.2 Synchronization and Timing Design 2

This section describes the synchronization and timing mechanism that is used in the hardware accelerator board.

a. Hardware Requirements

Each hardware accelerator board that supports connectivity functions in O-DU$_8$ must be able to support external synchronization I/O and to receive or transmit 1 PPS reference clock source in order to ensure synchronization across network.

b. Hardware Design

The accelerator board can operate in the master or slave mode when supporting IEEE 1588v2 timing/synchronization. It can generate 1 PPS reference clock for synchronization in the master mode and can receive the 1 PPS reference clock in the slave mode for internal synchronization. The timing circuitry of the accelerator board is shown in Figure 2-11. Two FPGA SerDes transceivers are used to receive and transmit SyncE TX and RX clocks.

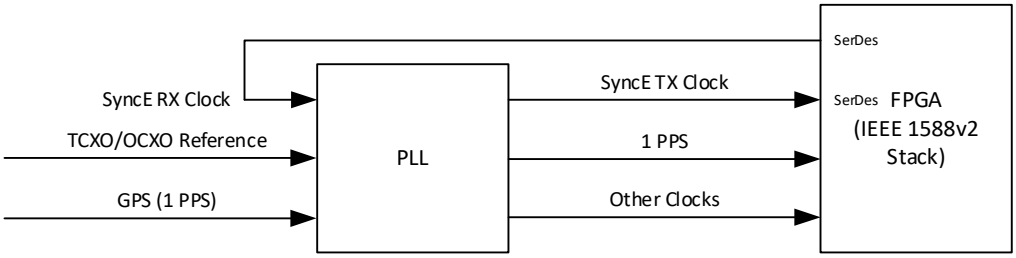


Figure 2-11: HW Acceleration Board Timing and Synchronization

2.2.5 External Interface Ports

The external interfaces of O-DU8 are described below.

a. Hardware Requirements

The following table shows the external ports or slots that the system provided.

Table 2-8: External Port List

Item Name	Description
Ethernet	Octave Gigabit Ethernet LAN connectors
	Dual 10GbE Base-T Ethernet connectors
	Dual 10GbE SFP+ Faber Ethernet connectors
	2x100G QSFP28 or 2x25G SFP28
USB	2 USB 3.0 ports
Serial Port	1 COM port via RJ45
Antenna port	1 SMA connector for GNSS

b. Hardware Design

The digital processing unit is a SoC device which provides the external ports described in the hardware requirement section. The system includes 2 USB 3.0 ports, and the serial RS232 port that can be used for Console Redirection, e.g. Out-of-Band Management. The system provides eight 1Gbps and four 10Gbps Ethernet ports. There are two or four 25G CPRI ports in system depends on the accelerator card used. The system also provides a RF interface to connect GNSS antenna. The following diagram outlines the external interfaces that supported.

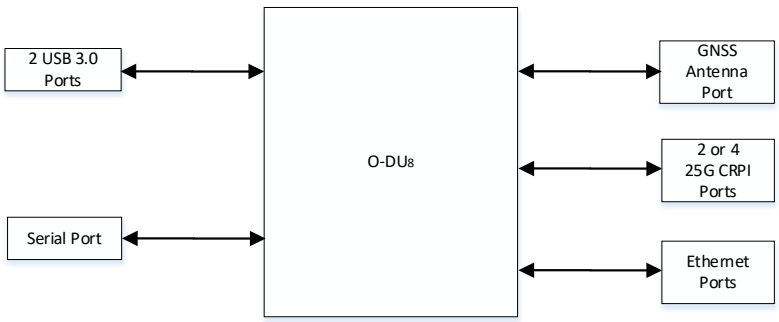


Figure 2-12: External interfaces reference design

2.2.6 O-DU8 Firmware

BIOS and BMC firmware are needed in the system and shall be installed.

2.2.7 Mechanical

The mechanical design for mother board, chassis, and cooling are listed in this section.

- **Mother Board**

  The mechanical layout of the mother board shows the location of major components and interface ports. The following diagram also provides the dimension of the board.

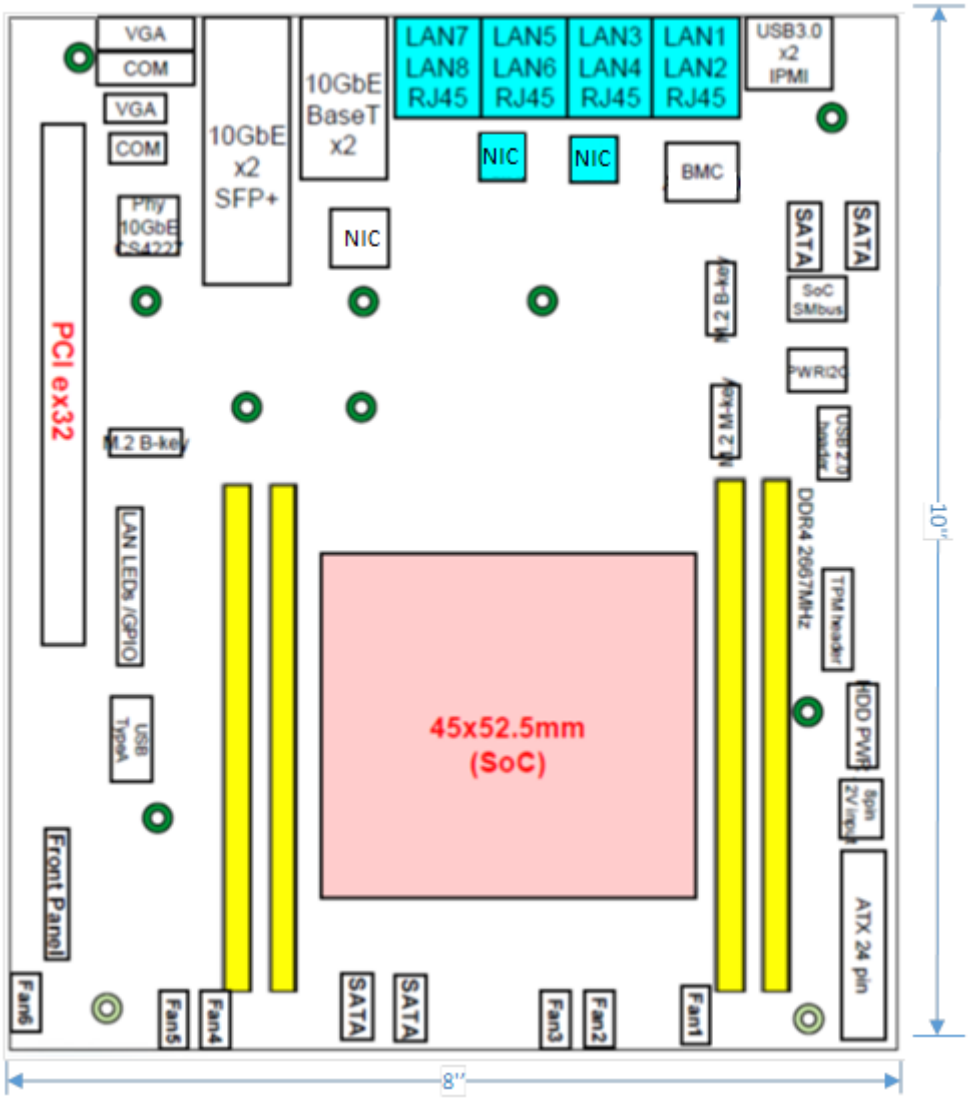


Figure 2-13: Mother Board Layout Diagram

- **Chassis**

The 1U rack mount chassis contains the layout of the power supply, Solid State Drive (SSD) and fans. The chassis dimension is showed in following figure.

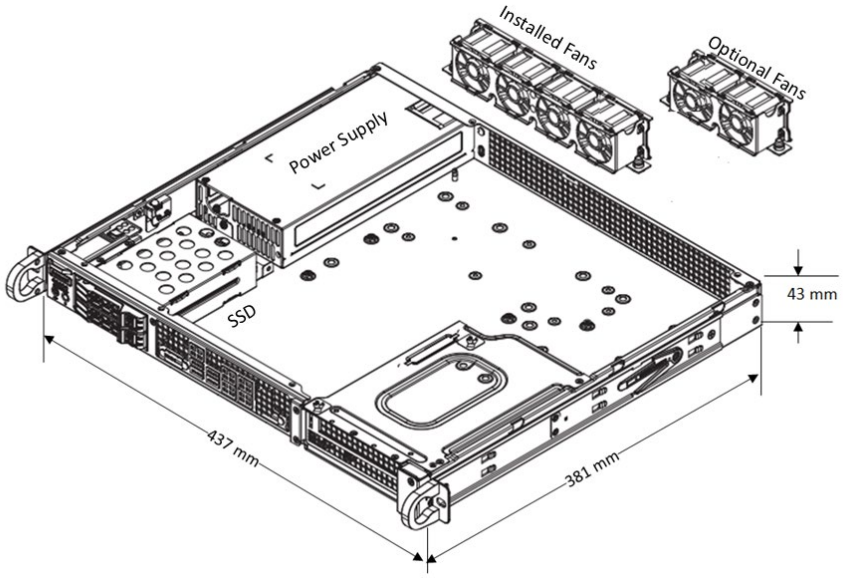


Figure 2-14: Chassis Mechanical Diagram

- **Cooling**

The system installs 4x 40x28mm PWM fans for the cooling. Up to 6 fans can be installed if needed.

2.2.8 Power Unit

In a fully loaded system with two PCIe slots populated with 75W each, the system power consumption should be less than 400W. The total system power requirement shall be kept less than 80% of the power supply capacity.

a. Hardware Requirements

The power is provided by 500W High-Efficiency power supply with Power Management Bus (PMBus) 1.2. The power support input and output power rails are listed below.

AC Input: 100-240V, 50-60Hz, 6.6A max

DC Output: +3.3V: 12A

+5V: 15A

+5V standby: 3A

+12V: 41A

-12V: 0.2A

1

b. Hardware Requirements

The O-DU$_8$ chassis includes one 500W power supply unit. The power supply unit is auto-switching capable, which enables it to automatically sense and operate at a 100v to 240v input voltage. The power supply unit features are listed in the following table.

Table 2-9: Power supply unit feature list

Item Name	Description
Output connectors	24pin/8pin/4+4pin/HDD/I2C
Dimension (W x D x H)	3.9 x 7.1 x 1.6 inch
Maximum Output Power	+3.3V: 12A +5V: 15A +12V: 41A -12V: 0.2A +5Vsb: 3A
Rated Input Voltage/Current	100-240Vac / 6.6A max
Rated Input Frequency	50-60HZ
Inrush current	Less than 30A

2.2.9 Thermal

Active cooling with up to 6 fans is integrated in the chassis.

The hardware acceleration cards described in Section 2.2.3.2 use passive cooling and a custom heatsink and is equipped with temperature sensors. It is designed to operate in temperatures ranging from -5°C to +55°C.

2.2.10 Environmental and Regulations

The O-DU$_8$ hardware system is RoHS Compliant. The power supply unit is EMC FCC/CISPR Class B compliant. Table 2-10 lists the environmental related features and parameters.

Table 2-10: Environmental Features

Item Name	Description
Operating Temperature	-5°C to +55°C
Non-operating Temperature	-40°C to 70°C
Operating Relative Humidity	8% to 90% (non-condensing)
Non-operating Relative Humidity	5% to 95% (non-condensing)

The hardware accelerator described in Section 2.2.3.2 is designed to operate in indoor environments and in temperatures ranging from -5°C to +55°C.

2.3 O-RU$_8$ Hardware Reference Design

In this section we describe, the HW reference design of all O-RU$_8$ components including block diagram, HW components, synchronization and timing, mechanical design, power supply, thermal, and environment requirements.

2.3.1 O-RU$_8$ High-Level Functional Block Diagram

Figure 2-15 provides a high-level functional block diagram depicting the major HW/SW components. It also highlights the internal/external interfaces that are required. This document shows how to implement the system defined by the IPC-HAR [6] document.

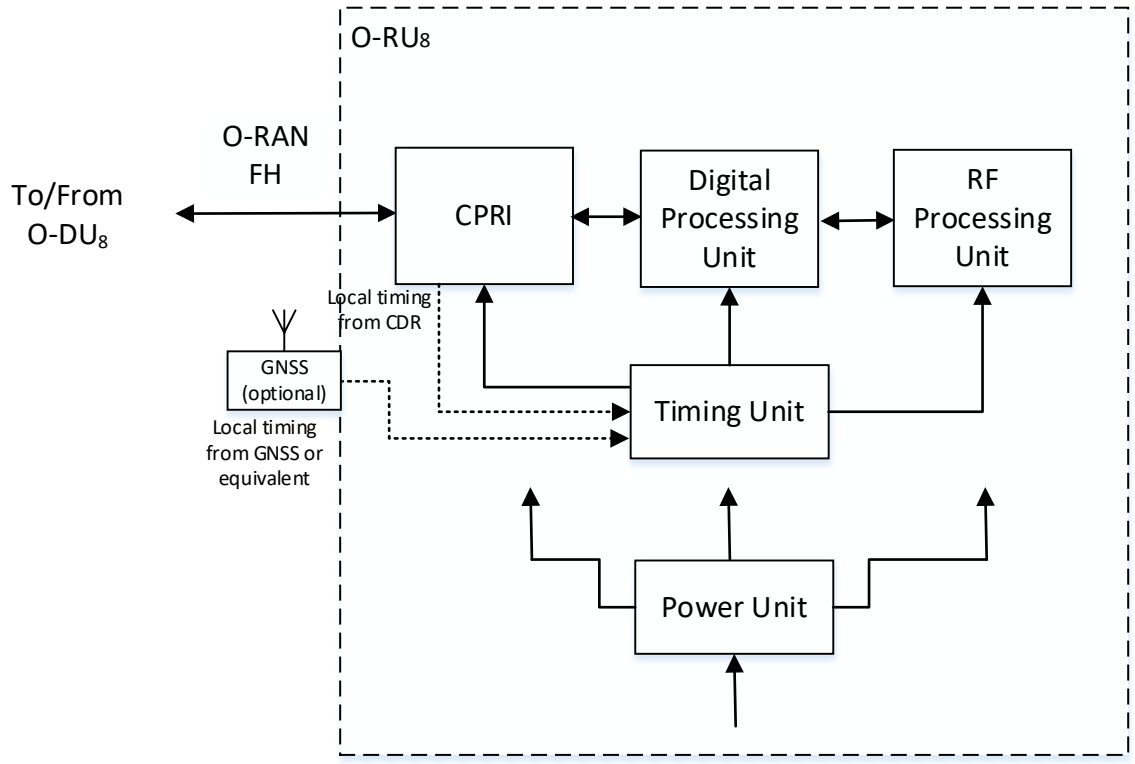


Figure 2-15: High-Level Functional Block Diagram

2.3.2 O-RU$_8$ Hardware Components

General block diagrams for 2T2R and 4T4R O-RU$_8$ examples are shown.

Figure 2-16 shows a 2T2R implementation and Figure 2-17 shows an 4T4R implementation. In each of these diagrams, the Digital Processing Unit is further detailed in 2.3.2.1. Items under the umbrella of RF

Processing Unit, including the Transceiver, RFFE and other RF items are reviewed in additional detail in 2.3.2.2. Clock and Synchronization are reviewed in 2.3.3. The Power Unit is detailed in 2.3.6.

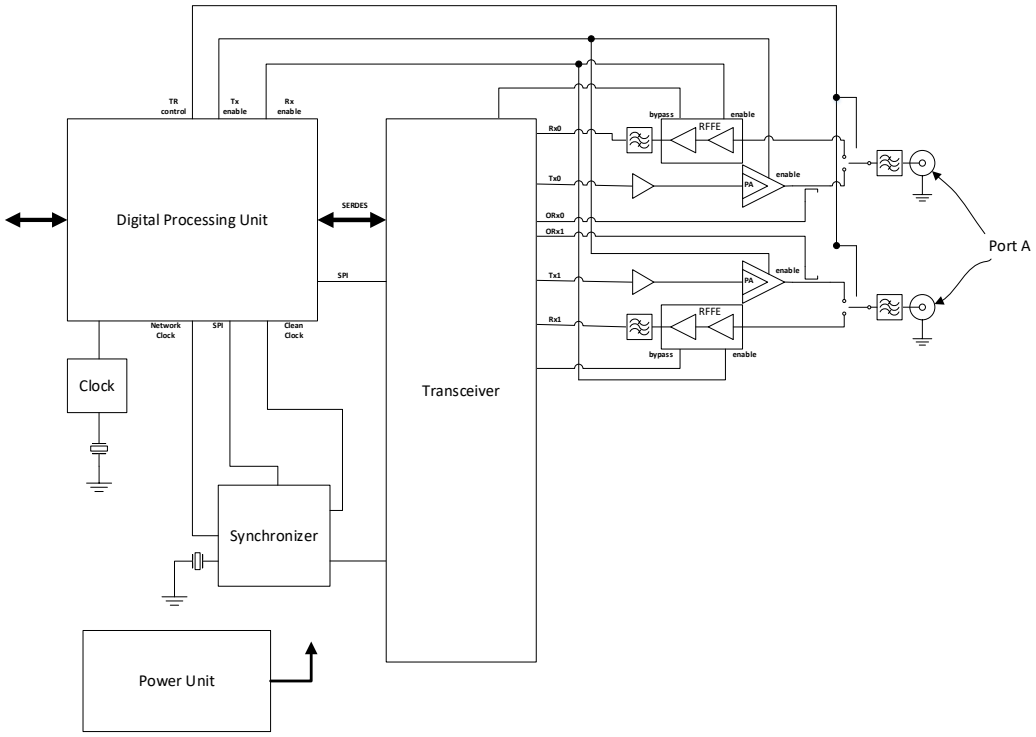


Figure 2-16: 2T2R General Block Diagram with TR switch

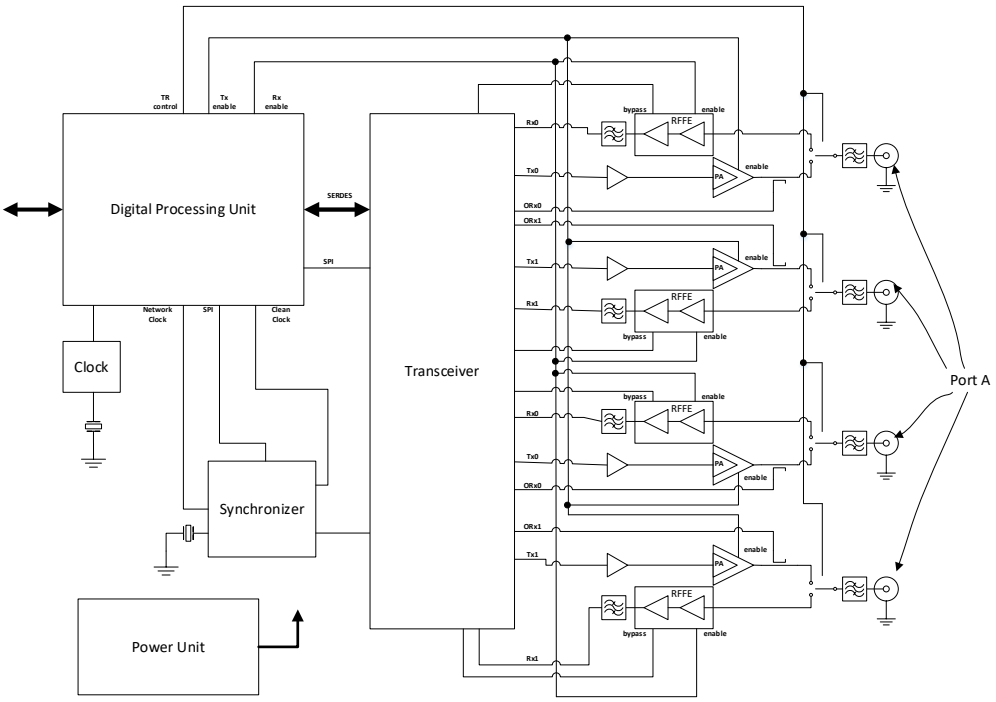


Figure 2-17: 4T4R General Block Diagram with TR switch

In following section, we describe their functionality, interfaces and performance of every block in the RF Processing Unit. Since device integration is an ongoing activity, chip boundaries may be fluid and some functionalities may move from one block to another or entire functionalities may be absorbed into other blocks. The sections below describe the functional blocks independent of which physical device they may reside in.

2.3.2.1 Digital Processing Unit

The digital processing unit of O-RU$_8$ is mainly for performing tasks related to FH interface, RF interface, and OAM.

i. FPGA Solution Design 1

The digital processing unit of O-RU$_8$ is mainly for performing FH interface, RF interface, and OAM functions.

a. FPGA Requirement

The following items are the main requirement for the O-RU$_8$:

- Interface requirement: One lane of bi-direction SerDes targeting CPRI will be @10Gbps for FH split option 8. Four lanes of bi-direction JESD204B SerDes will be used for 2T2R. Eight lanes of bi-direction JESD204B SerDes will be used for 4T4R
- Resource requirement: FPGA resource requirements for 2T2R and 4T4R are shown in Table 2-11 and Table 2-12.

Table 2-11: Resource requirement for 2T2R O-RU$_8$

Module	FF	LUT	BRAM18	DSP
DUC	4820	4346	0	108
DDC	4820	4346	0	108
CFR	11470	6136	22	36
DPD	34269	13250	188	87
JESD204B	4314	4285	0	0
Fronthaul(CPRI)	4210	2756	1	0
Other	8000	5000	100	12
Total	71903	40119	311	351

Table 2-12: Resource requirement for 4T4R O-RU$_8$

Module	FF	LUT	BRAM18	DSP
ChFIR	2832	2686	0	84
DUC	5148	3756	0	36
CFR	25676	12372	48	96
DPD	45770	17643	217	105

AGC	1000	600	4	8
DDC	1716	1252	0	12
UL_ChFIR	2832	2686	0	84
JESD2048	9900	11127	0	0
CPRI	10080	8960	32	0
Total	104954	61082	301	425

- Processor requirement: For device model A, the processor will be used, one is used for device control and management plane functions, the other one is for Digital Pre-Distortion (DPD) feedback path. For device model C, dual-core ARM cortex-A53 will be used, similarly one is used for device control and management plane functions, the other one is for DPD feedback path.

- Power requirement: Power estimations for 2T2R and 4T4R are shown in Figure 2-18 and Figure 2-19.

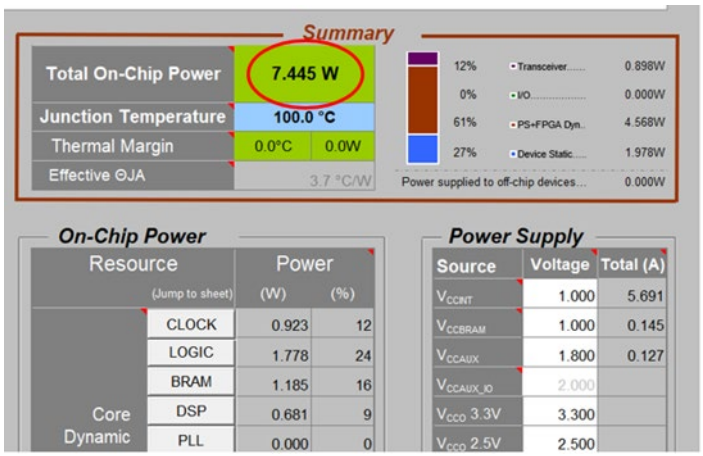


Figure 2-18: Power estimation for 2T2R O-RU8

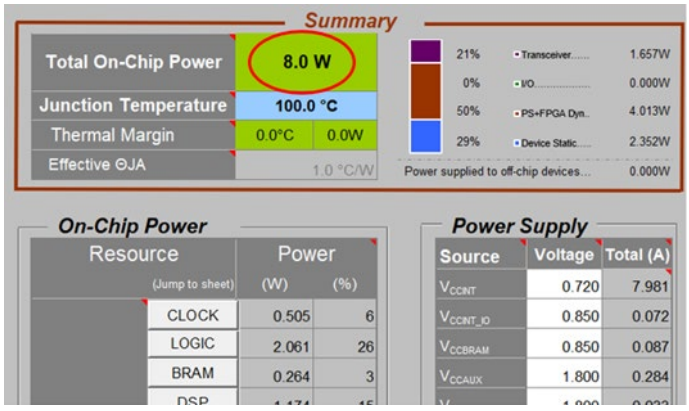


Figure 2-19: Power estimation for 4T4R O-RU8

- Speed grade, environmental requirement: -2L, 0℃ to 110℃

b. FPGA Design

This solution of the digital processing unit incorporates FPGA and a processor. The FPGA handles high speed digital processing such as FH, DDC, DUC, CFR and so on. All functions are listed in the previous section. The processor is used for hardware device configuration and the OAM function. The FPGA and the processor core can be integrated into one SoC or implemented into two devices. Here the FPGA and the processor core are integrated into one SoC device.

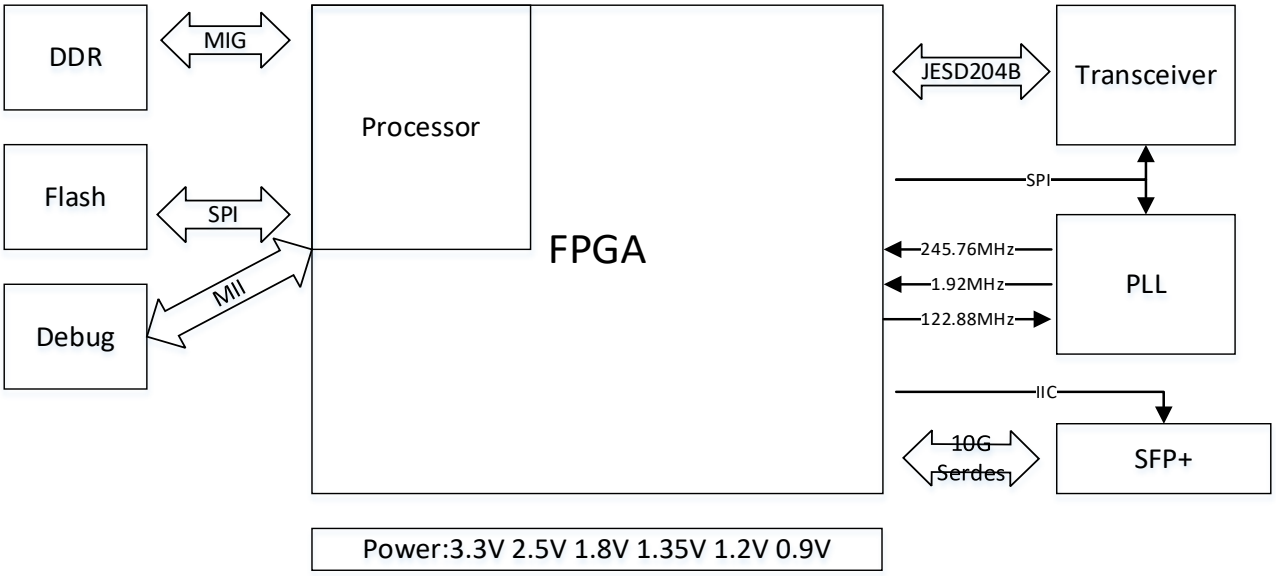


Figure 2-20: FPGA Reference Design Diagram

For the processor portion, the internal RAM resource may not be enough. So the external DDR is needed to let the processor handle more RAM consuming functions such as operation system or stack protocols like Network Configuration Protocol (NetConf) Client. For the O-RU$_8$ design, the DDR3 with 256Mb*16bit memory capacity is enough. The interface between the DDR and processor can be memory interface generator (MIG).The external Flash is used to store operation system related files, calibration information of the RF portion, NetConf related files, FPGA firmware and so on. For this O-RU$_8$ design, the flash with 2Gb memory capacity is enough. The online debug function is performed by external Ethernet PHY with an RJ45 connector. This allows the administrator to visit the internal function of the O-RU$_8$ and control it. The interface between the DDR and processor can be Media-Independent interface (MII). The Ethernet PHY device can be very general 100M Ethernet Transceiver. The FPGA has one SerDes lane connected to optical module to perform the fronthaul link between O-RU$_8$ and O-DU$_8$ /FHM$_8$. Another 4 SerDes lanes of the FPGA are needed to connect one transceiver of the O-RU$_8$ to transmit or receive IQ sample by the interface of JESD204B while the FPGA needs synchronized clock signals to work well. The interface between PLL and FPGA should be Low-Voltage Differential Signaling (LVDS).

2.3.2.2 RF Processing Unit

For the RF processing unit of O-RU$_8$, it will perform functionalities of ADC, DAC, LO, down converter, up converter and etc.

2.3.2.2.1 Transceiver Reference Design

For the O-RU$_8$ the sampling function and frequency conversion function can be performed by transceiver. The purpose of using the transceiver is to saving power and size of the PCB. The Transceiver is to convert between high speed baseband data and a low-level RF for both transmit and receive signal chain. In addition, the transceiver is responsible for orchestration of control signals not limited to the PA enable, LNA enable, LNA bypass as well as other required system level signals.

*a. **Hardware Requirements***

Include the requirements for this component.

- Interface Requirements: The interface requirements for the Transceivers are list in Table 2-13.

Table 2-13: Interface requirements of the Transceiver

Item Name	Description
High Speed Data	High speed data represents the baseband information being transmitted or received. Depending on the configuration of the O-RAN device, various bandwidths may be supported leading to a range of payload rates. Options for data include parallel data paths, JESD204B and JESD204C. Up to 8 lanes in each direction may be supported although fewer is preferred. Options such as DPD and numeric precision will impact the payload rate. Several options are shown in the following table. All data represents IQ 16-bit (N=16) precision. Some devices support IQ 12 bit (N'=12) which may reduce the required data rates accordingly. The tables below assume 1 ORx for 2 TRx. From Table 2-14 and Table 2-15, the number of lanes required may be determined by dividing the total bit rates shown by the capacity of a lane, typically 12.5 GBPS for JESD204B and 25 GBPS for JESD204C.
Reference Clock (Device Clock)	The transceiver should receive a reference for internal clock and LO synthesis needs. This reference clock can function as the JESD204 Device Clock where the interface is by SERDES. The specific clock frequency is determined by the operation mode of the transceiver and may range from 1Hz upward.
SYSREF	If the transceiver supports SERDES, then it should accept a SYSREF signal from the clock or data source as appropriate. The number and configuration for the SYSREF is dependent on the operating mode of the transceiver.
SYNCB	If the transceiver supports SERDES, then it should also support a SYNCB for each link as appropriate.
Control	Control of the transceiver is by way of 3 or 4 wire SPI or IIC functioning as a slave. Support for 1.8V control is required and tolerance of 3.3V is preferred. The transceiver may optionally include a separate SPI master for control of peripheral devices as required.
GPIO	The transceiver may optionally include GPIO for controlling peripherals including but not limited to PAs, LNAs and other devices. These GPIOs should at a minimum support 1.8V outputs but the specifics will be determined by the connected devices. The GPIO should

	also support input from peripheral devices. Input should at a minimum support 1.8V logic with tolerance of 3.3V preferred.
Tx Enable	The transceiver should provide an output to support enabling and disabling the external devices in the transmit chain such as a TxVGA (optional) and PA.
Rx Enable	The transceiver should provide an output to support enabling and disabling the external devices in the transmit chain such as a RF Front End Module or LNA.
LNA Bypass	The transceiver should provide an output to support bypassing the LNA appropriately in the condition of a large blocker if so required.
RF Outputs	RF outputs including the main Tx signal should support 50 ohm or 100 ohms signalling. These outputs can be either single ended or differential.
RF Inputs	RF inputs including the main Rx and the Observation Rx (ORx) (for DPD) should support 50 ohm or 100 ohms signalling. These inputs can be either single ended or differential. The device should support at least 1 ORx.

Table 2-14: Payload with DPD in Digital Device (GBPS Rx/Tx)

Bandwidth (MHz)	**Parallel**	**JESD204B**	**JESD204C**	**Parallel**	**JESD204B**	**JESD204C**
	2T2R			4T4R		
20	2.95/5.9	3.69/7.37	3.04/6.08	5.9/11.8	7.37/14.75	6.08/12.17
50	5.9/11.8	7.37/14.75	6.08/12.17	11.8/23.59	14.75/29.49	12.17/24.33
100	11.8/23.59	14.75/29.49	12.17/24.33	23.59/47.19	29.49/58.98	24.33/48.66

Table 2-15: Payload with DPD in Transceiver (GBPS Rx/Tx)

Bandwidth (MHz)	**Parallel**	**JESD204B**	**JESD204C**	**Parallel**	**JESD204B**	**JESD204C**
	2T2R			4T4R		
20	1.97/1.97	2.46/2.46	2.03/2.03	3.93/3.93	4.92/4.92	4.06/4.06
50	3.93/3.93	4.92/4.92	4.06/4.06	7.86/7.86	9.83/9.83	8.11/8.11
100	7.86/7.86	9.83/9.83	8.11/8.11	15.73/15.73	19.66/19.66	16.22/16.22

- Algorithm Requirements: The transceiver is required to provide appropriate algorithms to sustain RF operation including but not limited to Rx AGC, Tx Power control. DPD and CFR may be included either in the transceiver or in the digital baseband device making the trade-offs in Payload shown in Table 2-11 and Table 2-12.

- Device Configuration: The transceiver should support either 2T2R or 4T4R. In addition, at least one ORx path should be supported. Additional ORx paths are allowed as required for the application.

- Power Dissipation: Total dissipation of the TRx should be less than 6W for 4T4R.
- RF Specifications are given in Table 2-16

Table 2-16: RF specifications in Transceiver

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions/Comments
TRANSMITTERS						
Center Frequency		650		6000	MHz	
Transmitter Synthesis Bandwidth				450	MHz	
Transmitter Large Signal Bandwidth				200	MHz	
Transmitter Attenuation Power Control Range		0		32	dB	Signal-to-noise ratio (SNR) maintained for attenuation between 0 dB and 20 dB
Transmitter Attenuation Power Control Resolution			0.05		dB	
Adjacent Channel Leakage Ratio (ACLR)						20 MHz LTE at −12 dBFS
			−66		dB	75 MHz < f ≤ 2800 MHz
In Band Noise Floor						0 dB attenuation; in band noise falls 1 dB for each dB of attenuation for attenuation between 0 dB and 20 dB
			−154.5		dBm/Hz	600 MHz < f ≤ 3000 MHz
Out of Band Noise Floor						0 dB attenuation; 3 × bandwidth/2 offset
			−153		dBm/Hz	600 MHz < f ≤ 3000 MHz
Maximum Output Power						0 dBFS, continuous wave tone into 50 Ω load, 0 dB transmitter attenuation
			6		dBm	600 MHz < f ≤ 3000 MHz
Third Order Output Intermodulation Intercept Point	OIP3					0 dB transmitter attenuation
			27		dBm	600 MHz < f ≤ 5700 MHz
Error Vector Magnitude (Third Generation Partnership Project (3GPP) Test Signals)	EVM					
1900 MHz LO			0.6		%	50 kHz RF PLL loop bandwidth
3800 MHz LO			0.53		%	300 kHz RF PLL loop bandwidth
OBSERVATION RECEIVER	ORx					
Center Frequency		450		6000	MHz	
Gain Range			30		dB	IIP3 improves decibel for decibel for the first 18 dB of gain attenuation; QEC performance optimized for 0 dB to 6 dB of

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions/Comments
						attenuation only
Analog Gain Step			0.5		dB	For attenuator steps from 0 dB to 6 dB
Receiver Bandwidth				450	MHz	
Maximum Useable Input Level	P_{HIGH}					0 dB attenuation; increases decibel for decibel with attenuation; continuous wave corresponds to −1 dBFS at ADC
			−11		dBm	75 MHz < f ≤ 3000 MHz
Integrated Noise			−58.5		dBFS	450 MHz integration bandwidth
			−57.5		dBFS	491.52 MHz integration bandwidth
Third-Order Input Intermodulation Intercept Point	IIP3					Maximum observation receiver gain; test condition: P_{High} - 11 dB/tone
Narrow Band						
1900 MHz			15		dBm	
2600 MHz			16.5		dBm	
3800 MHz			18		dBm	
Wide Band						IM3 products>130 MHz at baseband; test condition: P_{High} - 11 dB/tone ; 491.52 MSPS
1900 MHz			13		dBm	
2600 MHz			11		dBm	
38000 MHz			13		dBm	
Third-Order Intermodulation Product	IM3					
			−70		dBc	600 MHz < f ≤ 3000 MHz
Spurious-Free Dynamic Range	SFDR		64		dB	Non IMx related spurs, does not include HDx; (P_{HIGH} − 11) dB input signal
Harmonic Distortion						(P_{HIGH} − 11) dB input signal
Second Order Harmonic Distortion Product	HD2		−80		dBc	In band HD falls within ±100 MHz
			−73		dBc	Out of band HD falls within ±225 MHz
Third-Order Harmonic Distortion Product	HD3		−70		dBc	In band HD falls within ±100 MHz
			−65		dBc	Out of band HD falls within ±225 MHz
RECEIVERS						
Center Frequency				6000	MHz	
Gain Range			30		dB	
Analog Gain Step			0.5		dB	Attenuator steps from 0 dB to 6

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions/Comments
					dB	
			1		dB	Attenuator steps from 6 dB to 30 dB
Receiver Bandwidth				200	MHz	
Maximum Useable Input Level	P_{HIGH}					0 dB attenuation, increases decibel for decibel with attenuation; continuous wave = 1800 MHz; corresponds to −1 dBFS at ADC
			−11		dBm	75 MHz < f ≤ 3000 MHz
Noise Figure	NF					0 dB attenuation, at receiver port
			12		dB	600 MHz < f ≤ 3000 MHz
Input Third-Order Intercept Point	IIP3					
Difference Product,	IIP3, d					
2600 MHz (Wideband)			17		dBm	Two (P_{HIGH} − 9) dB tones near band edge
2600 MHz (Midband)			21		dBm	
Sum Product,	IIP3, s					
2600 MHz (Wideband_			20		dBm	Two (P_{HIGH} − 9) dB tones, at bandwidth/6 offset from the LO
HD3	HD3					(P_{HIGH} − 6) dB continuous wave tone at bandwidth/6 offset from the LO
			−66		dBc	600 MHz < f ≤ 4800 MHz

1
2

b. *Hardware Design*

For the O-RU$_8$ the sampling function and frequency conversion function can be performed by transceiver. Usually the transceiver is integrated by ADC, DAC, LO, down converter, up converter and so on. The block diagram of transceiver design is shown in Figure 2-21.

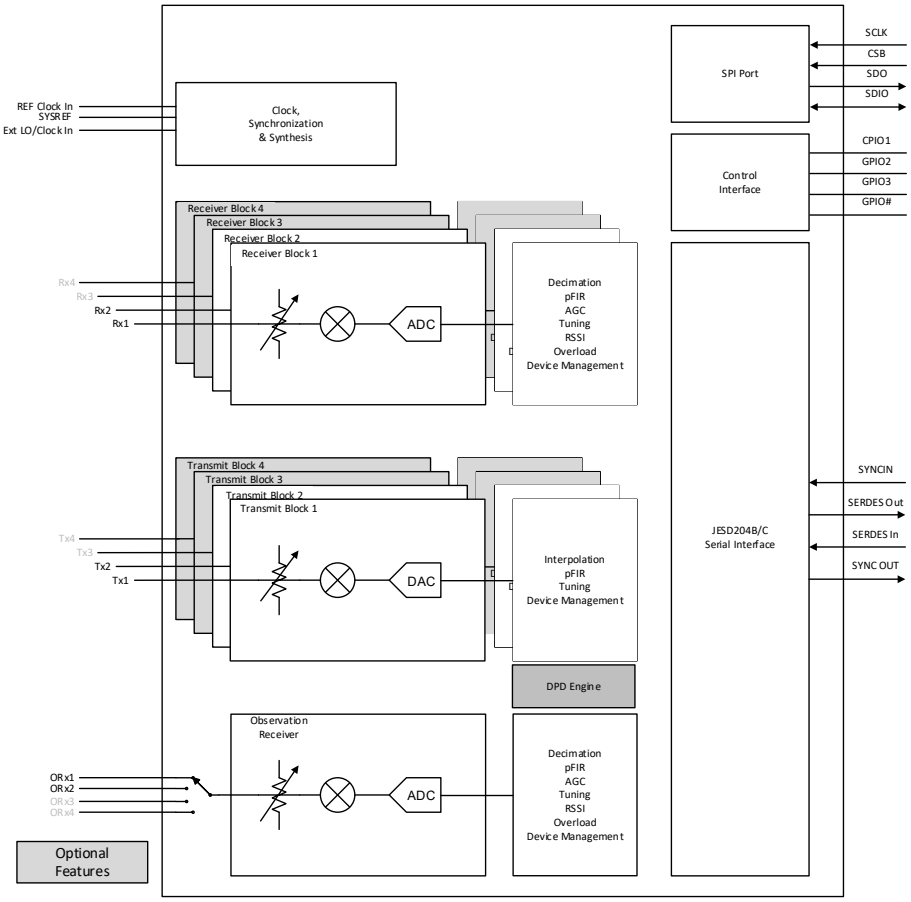


Figure 2-21: Transceiver Reference Design Diagram (Optional elements are highlighted in Grey)

The definition of the input/output lines are as follows:

- REF Clock In is the differential reference input to the on-chip synthesizer. This may also function as the SerDes Device Clock.
- SYSREF is a JESD204B/C differential synchronization signal.
- Ext LO/Clock Input is an optional input that can be used as an alternate LO or clocking signal.
- Rx1 through Rx4 are differential receiver inputs to their respective cores.
- Tx1 through Tx4 are differential transmitter output from their respective cores.
- ORx1 through ORx4 are differential observation receiver inputs.
- SCLK is the serial control clock input.
- CSB is the active low device select for the control interface.
- SDO is the serial data output for the control interface. This pin may be omitted for 3-pin control implementations.
- SDIO is a bidirectional serial data interface. In 4-pin mode, this pin functions as the serial data input.

- GPIO1 – GPIO# are general purpose IO signals used for interrupts, enables, test mode and resets. These are used as interface and control for peripheral devices including controllers, TxVGAs, PA, LNA and similar devices where SPI control is too slow.
- SYNCIN are differential pins associated with the receiver channels of the JESD204 interface. In not used, they are typically grounded. Up to 4 pair may be supported.
- SerDes Out are differential JESD204B/C data output interfaces. Up to 8 lanes may be active.
- SerDes In are differential JESD204B/C data input interfaces. Up to 8 lanes may be active.
- SYNCOUT are differential pins associated with the transmitter channels of the JESD204 interface. If not used, do not connect. Up to 4 pair may be supported.

2.3.2.2.2 Power Amplifier (PA) Reference Design

The Power Amplifier boosts the RF output to the level required for the base station class.

*a. **Hardware Requirements***

The PA should have large enough gain to reduce the need for an additional driver. This will reduce cost and PCB space. The output power should be at least 27dBm (30dBm for 500mW/port to compensate for the loss of switch and antenna filter). The ACLR should be greater than 47dBc according to the related 3GPP test mode. DPD is needed to reduce the power consumption. The P1 dB requirement is closely related to the DPD algorithm.

Table 2-17: PA Requirements

Frequency band	**Band41**	**Band78**	**Additional Information**
Gain	>33dB	>33dB	
P3dB	>34dBm	>34dBm >37dBm(500mW/port)	ACLR>=47dBc with DPD@100MHz NR 27dBm (30dBm for 500mW/port)
Input return loss	<-15dB	<-15dB	
Output return loss	<-15dB	<-15dB	
Switching Speed	<1us	<1us	
HD2	>28dBc	>28dBc	CW 27dBM
HD3	>30dBc	>30dBc	CW 27dBm

- Interface Requirements:

The interface requirements of the transmit PA are listed in Table 2-18.

Table 2-18: Interface requirements of the transmit PA

Item Name	Description
Enable	The enable input should be compatible with 1.8V logic and tolerate 3.3V as required. A logic high enables the PA. A logic low disables the device and places it in a minimum dissipation mode.
RF Outputs	RF outputs support 50-ohm single ended to properly interface to a directional coupler, isolator, switch or antenna.
RF Inputs	RF inputs should support 50 ohm, single ended match to the transceiver output or preamp.
RF power detector	Power detector (optional)

- Power Dissipation: Dissipation while enabled should not exceed ~3W.

b. Hardware Design

RF_{in} is the input to the PA, RF_{out} is the output of the PA. V_{cc} and V_{bias} are the power inputs of the PA. PAEN is the control pin to disable or enable the PA. The input and output should match to 50 ohm as much as possible to reduce the reflection. V_{cc} and V_{bias} need capacitors to reduce the DC power ripples and give a short route to reduce the RF energy leakage. Figure 2-22 shows the details.

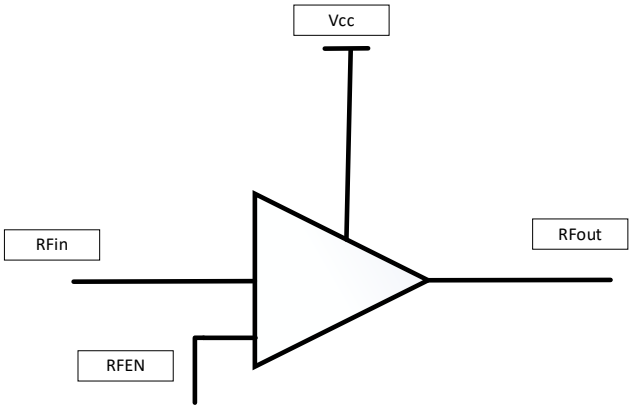


Figure 2-22: PA Reference Design Diagram

2.3.2.2.3 Low Noise Amplifier (LNA) Reference Design

The purpose of the LNA is to boost the Rx signal to a level that can nominally interface directly to the transceiver. This block will typically be a 2-stage amplifier with a 2nd stage bypass. The frontend will also include a TR switch to shunt any Tx signal to a termination away from the amplifier.

a. Hardware Requirements

The requirements of the LNA are listed here.

- Interface Requirements: The interface requirements of the transmit LNA are listed in Table 2-19.

Table 2-19: One stage LNA Requirements

Parameter	Band 41	Band 78	Additional Information
NF	<1dB	<1dB	
Gain	>15dB	>15dB	
P1dB	>18dBm	>18dBm	
Input return loss	<-12dB	<-12dB	
Output return loss	<-15dB	<-15dB	
Switching Speed	<1us	<1us	
IP3	>30dBm	>30dBm	2 Tone CW -15dBm
K	>1	>1	up to f = 12.5 GHz
Max input power	>15dBm	>15dBm	

- Device Configuration: Single or dual device.
- Power Dissipation: Less than 500 mW for a dual when device is fully enabled in receive mode.
- RF Specifications: For LNA unit, it should have larger enough gain to reduce extra driver amplifier for cost and PCB space. The NF figure of LNA should small enough to overcome the loss of switch and filter.

Table 2-20: Two stage LNA Requirements

Frequency band	Band41	Band78	Additional Information

NF	<1 dB	<1.5 dB	
Gain	High gain >=32 dB Low gain >=16 dB	High gain >=32 dB Low gain >=16 dB	
P1dB	>18dBm	>18dBm	
Input return loss	<-12dB	<-12dB	
Output return loss	<-15dB	<-15dB	
Switching Speed	<1us	<1us	
IP3	>30dBm	>30dBm	2 Tone CW -15dBm
K	>1	>1	up to f = 12.5 GHz
Max input power	>15dBm	>15dBm	

b. Hardware Design

The reference designs for one stage LNA and two-stage LNA are given.

- One stage LNA

RFin is the input port of the LNA. RFout is the output of the LNA. Vcc is the power input of the LNA. LNA EN is the control pin to disable or enable the LNA. The input match to 50 ohm as much as possible to reduce the reflection. But for output of the LNA, it is hard to get best Noise figure and the output return loss. Compromise is needed between NF and return loss. Usually the return loss should be around -10dB or -12dB. Also is should be very careful to assure the stability of the LNA in a large bandwidth. Vcc needs capacitors to reduce the DC power ripples and give a short route to reduce the RF energy leakage. Figure 2-23 shows the details.

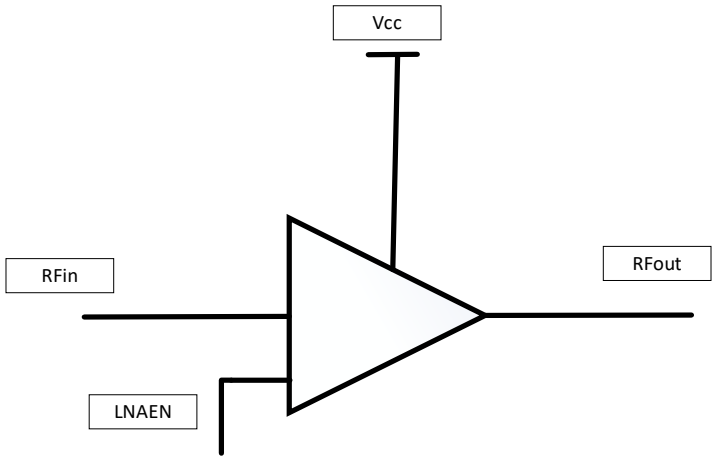


Figure 2-23: One Stage LNA Diagram

- Two stage LNA

RFin is the input port of the LNA. RFout is the output of the LNA. Vcc is the power input of the LNA. LNA EN is the control pin to disable or enable the LNA. The input matches to 50 ohm as much as possible to reduce the reflection. But for output of the LNA, it is hard to get best Noise figure and the output return loss. Compromise is needed between NF and return loss. Usually the return loss should be around -10dB or -12dB. Also is should be very careful to assure the stability of the LNA in a large bandwidth. Vcc needs capacitors to reduce the DC power ripples and give a short route to reduce the RF energy leakage. Figure 2-24 shows the details.

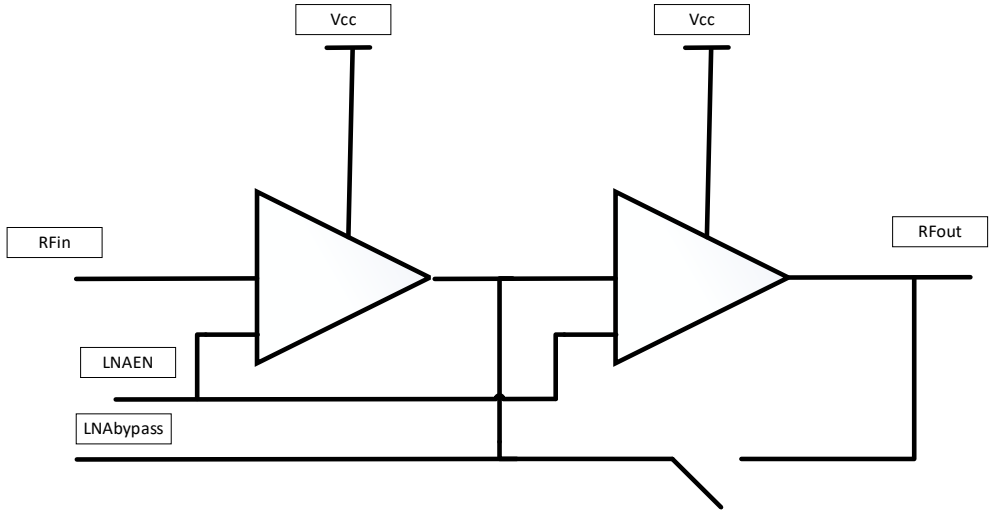


Figure 2-24: Two-Stage LNA Diagram

2.3.2.2.4 RF Switch Reference Design

For TDD use, the TX and RX links work spiritedly by time duplex. The switch is used to change the RF link according to the TDD duplex. In the TDD TX mode, the switch is switched to connect PA and antenna. In TDD RX mode, the switch is switched to LNA and antenna.

*a. **Hardware Requirements***

The requirements of the RF switch are listed here.

- Interface Requirements

The interface requirements of the switch are listed in Table 2-21.

Table 2-21: Interface requirements of the RF switch

Item Name	Description
Control	Control the switch working mode
RF COM	The port connected to the antenna RF inputs should support 50 ohm, single ended to match the antenna.

RF Outputs	The port connected to the LNA. RF inputs should support 50 ohm, single ended to match the LNA.
RF Inputs	The port connected to the PA. RF inputs should support 50 ohm, single ended to match the PA output.

- RF Specifications

For switch, it should have larger P1dB to not degrade the ACLR or damage the switch itself. The output power should be at least 26dBm (29dBm for 500mW/port) count on the loss of antenna filter so it must have low loss. Also the isolation needs to be high to protect the LNA. Higher HD2 and HD3 are needed to reduce the out of band emissions.

Table 2-22: RF Switch Requirements

Frequency band	Band 41	Band 78	**Additional Information**
Loss	<0.8dB	<0.8dB	
P1dB	>36dBm	>36dBm >39dBm(500mW/port)	
Input return loss	<-15dB	<-15dB	
Output return loss	<-15dB	<-15dB	
Switching Speed	<1us	<1us	
Isolation	>30dB	>30dB	
HD2	>70dBc	>70dBc	CW 27dBm
HD3	>70dBc	>70dBc	CW 27dBm

b. Hardware Design

RF Com is the input port of the switch, also can be the output of the switch. RF in is the input of the switch. RF out is the output of the switch. Control 1 and control 2 are the control pin to switch the RFin to RF Com or from RF Com to RF out. Figure 2-25 shows the details.

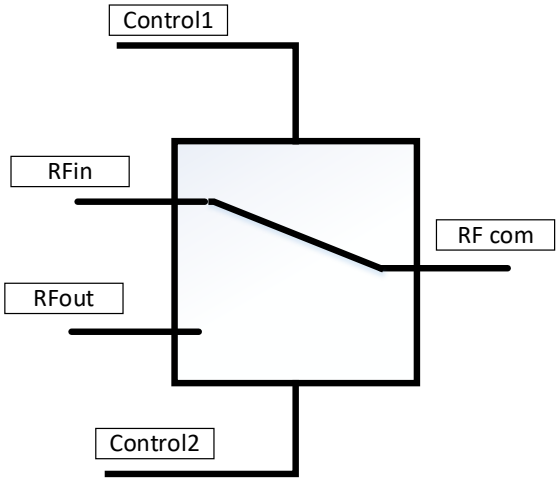


Figure 2-25: RF Switch Reference Design Diagram

2.3.2.2.5 Antenna Reference Design

The antenna is used to radiate the TX power on to the air and receive the RX power from the air. For indoor Picocell scenario, the isotropic antenna is the first choice.

a. ***Hardware Requirements***

The following table shows the antenna requirement for the O-RU$_8$.

Table 2-23: Antenna Requirements

Frequency band	Band 41	Band 78
VSWR	≤1.8	≤1.8
Power capacity	≥1 W	≥2 W
Gain	≥2 dBi	≥2.5 dBi
Beam width on vertical direction	≥35°	≥35°
pattern roundness on horizontal direction	±3 dB	±3 dB

b. ***Hardware Design***

One possible choice of the isotropic antenna is the whip antenna. Following figure shows a design of one kind of whip antenna.

Figure 2-26: Whip Antenna

2.3.3 Synchronization and Timing

The purpose of the Clocking device is to accept the network reference clock, typically 1 PPS, and generate a jitter free reference clock(s) for other devices in the system including the RF transceiver and digital block. The clock is typically

part of an IEEE 1588 implementation either controlled by an external stack or implemented in the clock device itself which could be part of the baseband implementation.

a. Hardware Requirements

Hardware requirements are:

- Interface Requirements: The interface requirements of the transmit Clocking are listed in Table 2-24.

Table 2-24: Interface requirements of the clocking

Item Name	Description
Reference Clock Input	The clock device should receive a network reference clock from the FPGA/ASIC. This typically could be a 1pps, 10 MHz or other standard reference as determined by the specific implementation. More than one input is allowed that may be selected between when the reliability of one reference is compromised. Standard differential clocking should be used to preserve signal integrity.
Control	Control of the transceiver is by way of 3 or 4 wire SPI or IIC functioning as a slave. Support for 1.8V control is required and tolerance of 3.3V is preferred. Typically, the clock devices will be part of an IEEE1588v2 protocol loop controlled by way of this control interface or other GPIO lines as required by the hardware implementation.
Clock Outputs	One or more clock outputs are required to drive the digital device and transceiver clock inputs. Each output should be independently programmable in frequency to suit the application. The outputs should nominally be differential to preserve clocking edge and to maximize tolerance of high common mode signals.
RF Inputs	RF inputs should support 50 ohm or 100 ohms, single ended or differential signalling to match the transceiver output.

- Power Dissipation: Total dissipation should be about less than 2W.
- RF Specifications: Clocking RF requirements are given in the following table.

Table 2-25: Clocking RF requirements

Absolution Phase Noise, Dual Loop Mode - LVDS OUTPUT	Typical	Additional Information
f_{OUT} = 122.88 MHz		Example is using an external 122.88 MHz VCXO (Crystek CVHD-950); reference = 122.88 MHz; channel divider = 10 or 1; PLL2 loop bandwidth (LBW) = 450 kHz.
10 Hz Offset	−86 dBc/Hz	
100 Hz Offset	−106 dBc/Hz	
1 kHz Offset	−126 dBc/Hz	
10 kHz Offset	−135 dBc/Hz	
100 kHz Offset	−139 dBc/Hz	
800 kHz Offset	−147 dBc/Hz	

1 MHz Offset	−148 dBc/Hz	
10 MHz Offset	−157 dBc/Hz	
40 MHz Offset	−158 dBc/Hz	
Absolute Phase Noise, Single Loop Mode – LVDS OUTPUT	Typical	
LVDS OUTPUT		
f_{OUT} = 122.88 MHz		Example is using an external 122.88 MHz reference (SMA100A generator); reference = 122.88 MHz; channel divider = 10; PLL2 LBW = 450 kHz.
10 Hz Offset	−111 dBc/Hz	
100 Hz Offset	−113 dBc/Hz	
1 kHz Offset	−123 dBc/Hz	
10 kHz Offset	−135 dBc/Hz	
100 kHz Offset	−140 dBc/Hz	
800 kHz Offset	−147 dBc/Hz	
1 MHz Offset	−148 dBc/Hz	
10 MHz Offset	−157 dBc/Hz	
40 MHz Offset	−157 dBc/Hz	
CLOCK OUTPUT ADDITIVE TIME JITTER (BUFFER MODE)	Typical	Example is using an external 122.88 MHz source driving VCXO inputs (distribution section only, does not include PLL and VCO)
f_{OUT} = 122.88 MHz	79 fS	Integrated BW = 200 kHz to 5 MHz
	101 fS	Integrated BW = 200 kHz to 10 MHz
	140 fS	Integrated BW = 12 kHz to 20 MHz
	187 fS	Integrated BW = 10 kHz to 40 MHz
	189 fS	Integrated BW = 1 kHz to 40 MHz
	176 fS	Integrated BW = 1 MHz to 40 MHz

1

2

b. *Hardware Design*

For the clocking function, it is usually performed by a synchronization IC which may include one or more PLLs. And it also can supply numbers of output port to support different frequency clocking.

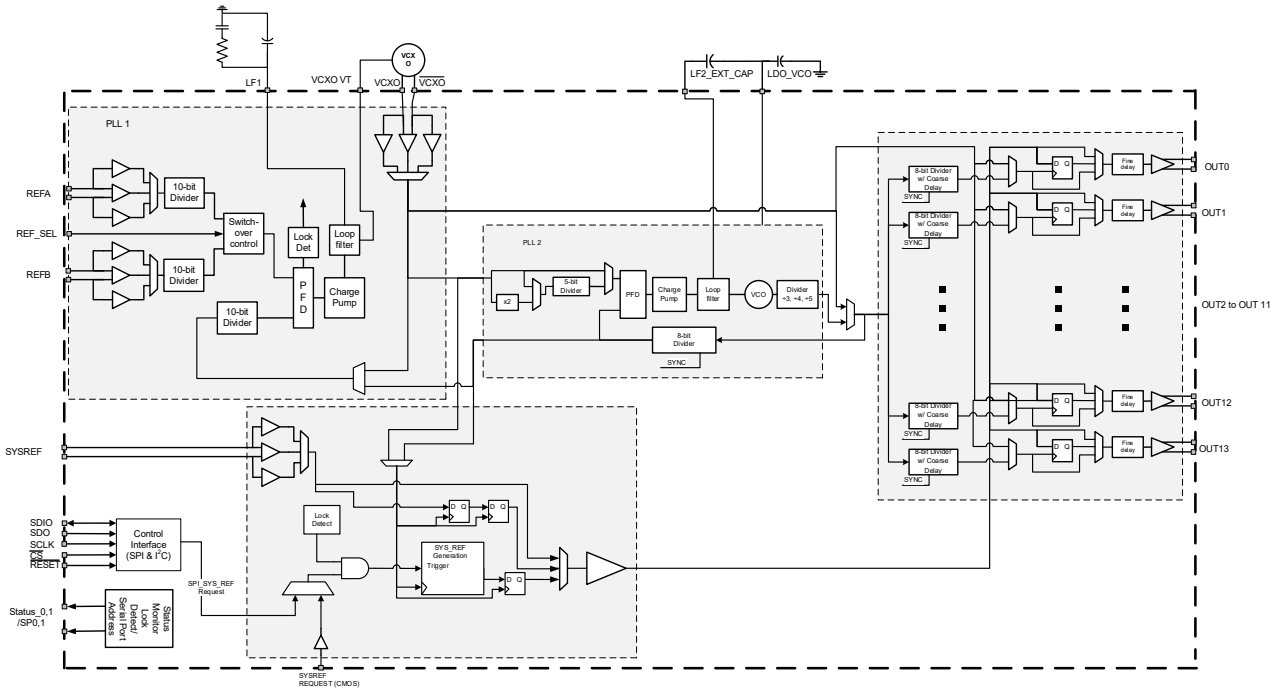


Figure 2-27: PLL Reference Design Diagram

REFA and REFB are differential reference clock inputs from the source to be cleaned up and used as the system reference. REF_SEL is the control pin used to select between REFA and REFB signals. LF1 is PLL1 external loop filter. VCXO_VT is the VCXO control voltage. This pin is connected to the voltage control pin of the external VCXO. VCXO_IN are differential signals from the external VCXO. These typically can be configured for single ended operation as well. LF2_CAP is the external loop filter capacitor for loop 2. This cap is connected between this pin and LDO_VCO pin. LDO_VCO is the on-chip LDO regular decoupling for the VCO. RESET(bar) is an active low pin to reset the internal logic to their default states. CS(bar) is an active low chip select for serial control. SCLK/SCL is a serial control port clock for both SPI and I2C. SDIO/SDA is a bidirectional serial data pin for both SPI and I2C. SDO is the serial data out pin for 4-wire mode. OUT0 through OUT13 are differential programmable output clock signals. These are the primary outputs of this device and provide high performance clock signals to the transceiver, baseband device and other key elements. This device is also responsible for providing SYSREF to various devices in the system if required. It may also use an external SYSREF as a source that may be retimed for local timing if necessary. STATUS0/SP0 is a lock detect and other status signal. STATIS1/SP1 is a lock detect and other status signal. SYSREF_IN is an external SYSREF input clock is a differential input representing the JESD204 synchronization from the external source.

2.3.4 External Interface Ports

a. *Hardware Requirements*

The following table shows the external ports or slots provided by O-RU$_8$.

Table 2-26: External Port List

Port Name	Feature Description
Fronthaul interface	One 10Gbps SFP+ transceiver interface
	Or one RJ45 Ethernet interface
Debug interface	RJ45 for debug usage
Power interface	Two-pin male plug for power cable

b. *Hardware Design*

- SFP+ case and connector: The SFP+ case and connector are standardized component on the market; following figure describe the form factor of one SFP case which is integrated with connector.

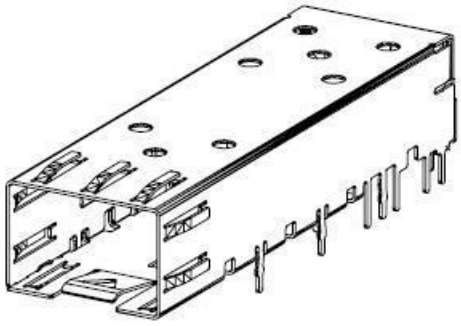


Figure 2-28: SFP+ case and connector

- RJ45 Ethernet interface: The RJ45 Ethernet interface is standardized component on the market; following figure describe the form factor of one RJ45 interface.

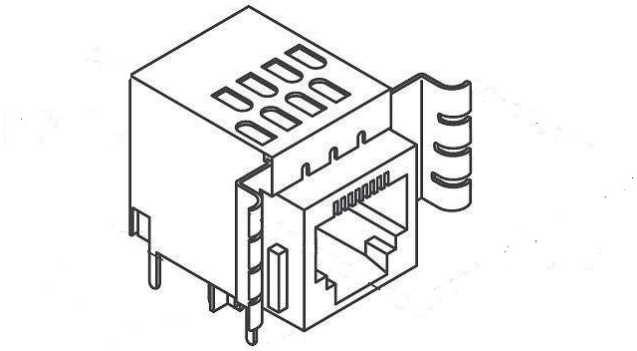


Figure 2-29: RJ45 interface

2.3.5 Mechanical

The shell of O-RU$_8$ is showing in the following figure. The O-RU$_8$ should be quiet, so it depends on the natural heat dissipation method. Usually the bottom of the shell is built by metal. All hot component should make its surfaces contact to metal shell through thermal pad.

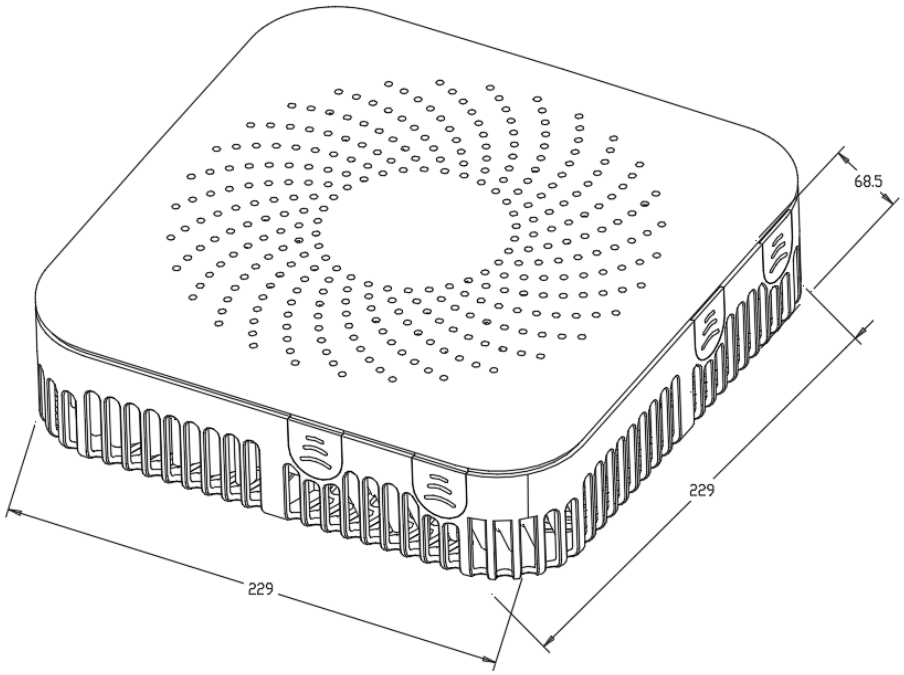


Figure 2-30: O-RU$_8$ Shell Mechanical Diagram

2.3.6 Power Unit

For Picocell, for the power solution of the O-RU$_8$ can be over Ethernet cable (cat5E, cat6A) by POE (POE+, POE++) or directly power cable with fibre. And for the board power solution, LDO and DCDC can be used.

a. Hardware Requirements

For PA unit, it should have larger enough gain to reduce extra driver amplifier for cost and PCB space. The output power should be at least 27dBm count on the loss of switch and antenna filter. The ACLR should larger than 47dBc according to the related 3GPP test mode. To reduce the power consumption, DPD is needed. And the P1 dB requirement is closely related to the DPD algorithm.

Table 2-27: Power unit requirement for 2T2R and 4T4R O-RU$_8$

Module	Power consumption			
	2T2R	4T4R	Unit	
FPGA	8	9	W	
Transceiver	5.74	11.48	W	
PLL	1.78	1.78	W	
PA	5	10	W	0.75 TDD ratio
LAN	0.15	0.3	W	0.25 TDD ratio
Ethernet PHY	1	1	W	
DDR	0.9	0.9	W	
Flash	0.1	0.1	W	
others	1.65	1.65	W	
component total power consumption	24.3	36.2	W	
O-RU$_8$ power consumption	28.6	42.6	W	on board power efficiency 0.85

Table 2-28: Power unit function

Function	Priority	
Enable/Disable	Mandatory	
Power good	Optional	
Input voltage	Mandatory	48V
Output voltage	Mandatory	0.85V 1.2V 1.3V 1.5V 1.8V 2.5V 3.3V 5V 12V etc.

b. Hardware Design

The block diagram of POE reference design is shown in Figure 2-31. Data Pair and Spare Pair are the POE connections on the data transformers used to source the power over Ethernet. V_{AUX} is a local backup power source if desired. Isolated Output is the isolated raw output from the POE sub-circuit.

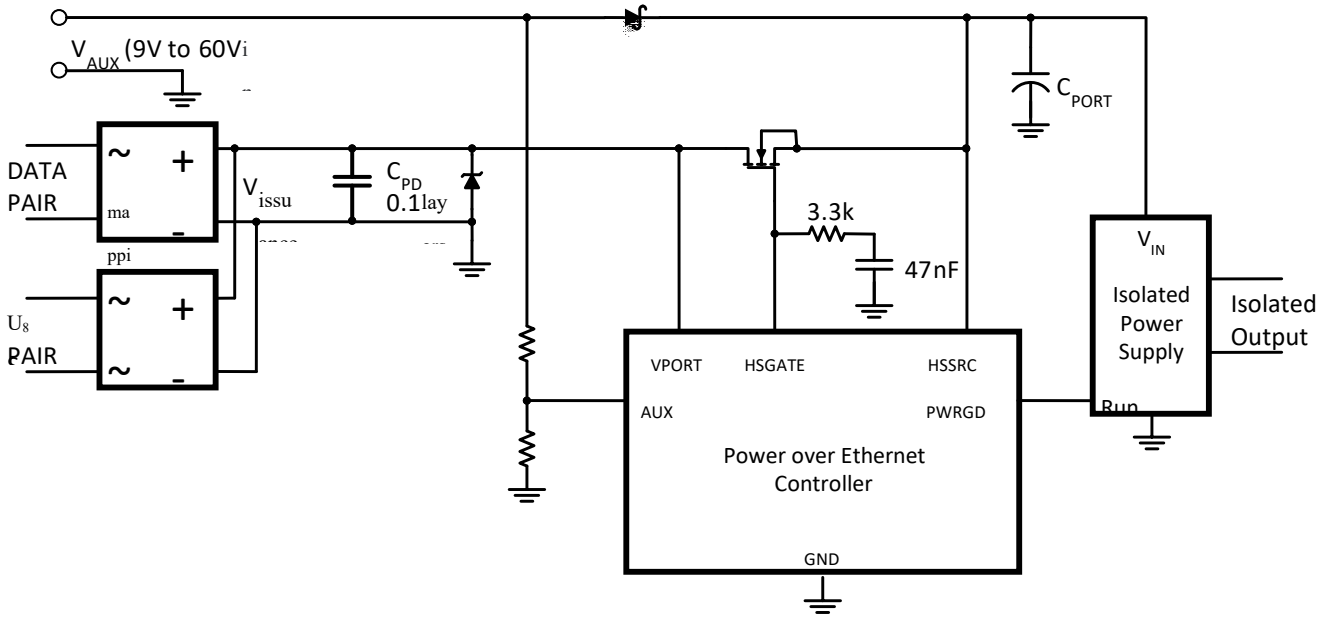


Figure 2-31: POE Reference Design Diagram

2.3.7 Thermal

O-RU_8 will use passive cooling.

2.3.7.1 Environmental and Regulations

The O-RU_8 hardware system is RoHS Compliant. The power supply unit is EMC FCC/CISPR Class B compliant. Table 2-29 lists the environmental related features and parameters.

Table 2-29: Environmental Features

Item Name	Description
Operating Temperature	-5°C to +55°C
Non-operating Temperature	-40°C to 70°C
Operating Relative Humidity	8% to 90% (non-condensing)
Non-operating Relative Humidity	5% to 95% (non-condensing)

2.4 FHM$_8$ Hardware Reference Design

The FHM$_8$ is used to bridge the connation between O-DU$_8$ and O-RU$_8$. It needs to power the O-RU$_8$ and reduces the front haul bandwidth.

2.5 FHM$_8$ High-Level Functional Block Diagram

Figure 2-32 shows the FHM$_8$ block diagram for this reference design. Following sections below describe the functionality, interface and performance for each respective block of the Digital Processing Unit. As device integration is an ongoing activity, chip boundaries may be fluid and some functionality may move from one block to another or entire functionalities may be absorbed into other blocks. The sections below describe the functional blocks independent of which physical device they may reside in.

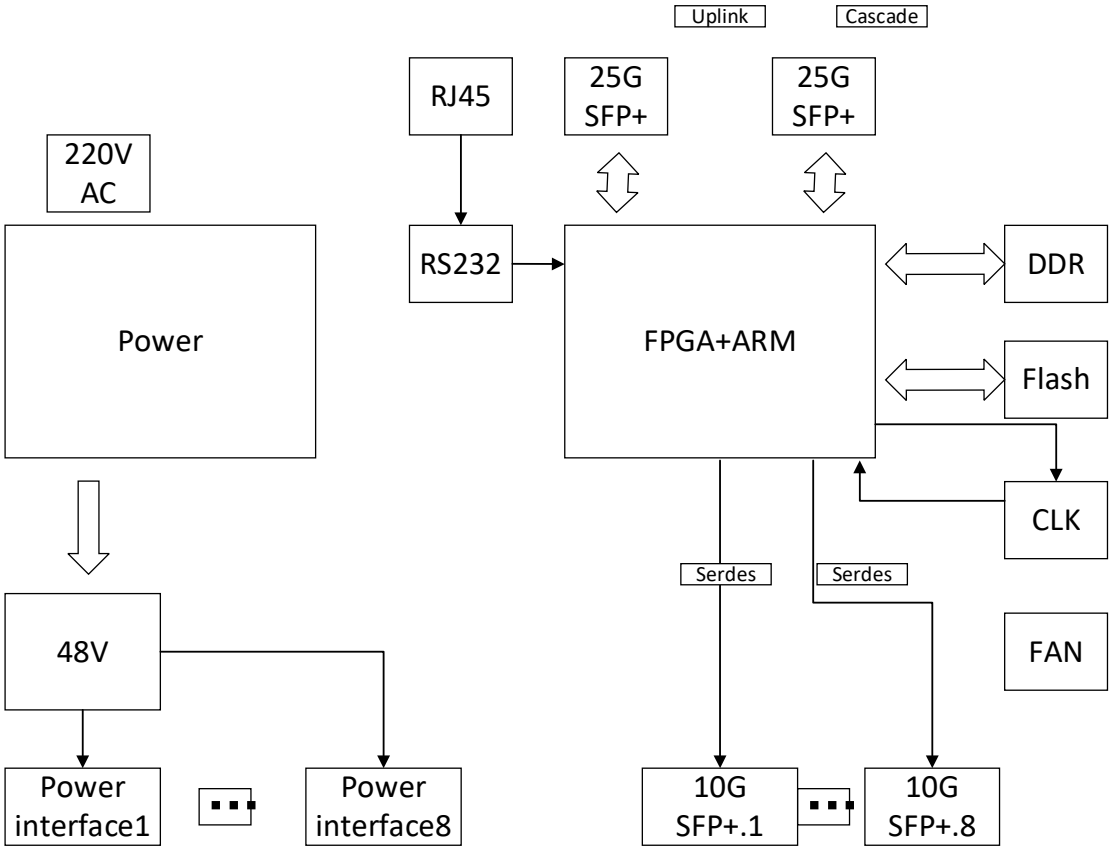


Figure 2-32: FHM$_8$ General Block Diagram

2.6 FHM$_8$ Hardware Components

2.6.1 Digital Processing Unit

The FHM$_8$ need a lot of high speed interface. And for the shared cell application, the uplink need combining function.

i. ***FPGA Solution Design 1***

In this section, the requirement and reference design based on FPGA are described.

a. ***FPGA** Requirement*

The requirements are as follows:

- Interface requirement

Table 2-30: Interface requirements of the FPGA

Item Name	Description
SerDes lane	At least 8 lanes of bi-direction10Gb SerDes
	At least 2 lanes of bi-direction25Gb SerDes

b. ***FPGA** Design*

This unit handles multiple high speed interfaces and the digital signal processing. The main chipset is based on FPGA with ARM multi-core processor while FPGA is responsible for high speed data processing and ARM cores are mainly for configuration and management.

The FPGA function is shown below:

For the uplink processing, the CPRI interface module is used to receiver CPRI link form O-RU$_8$ and generate the reference frequency and time slot for time and frequency synchronization. Then the CPRI de-frame module get the uplink IQ bit stream of the time domain the carrier from O-RU$_8$. If the uplink IQ stream is compressed, then it needs the de-compression module to translate them into original bit width.

After that all the O-RU$_8$ uplink IQ streams are combined together and the precondition for the combine function is that each stream from different O-RU$_8$ are aligned in time domain.

If the FHM$_8$ can support two separate cells then the two cell uplink signal should be aggregated or interleaved together. Then capture the IQ streams in the CPRI frame and transmitted by CPRI interface to the O-DU$_8$.

For the downlink processing, the procedure is much the same with uplink but with an inverse sequence. The difference is that the downlink signal is duplicated with 8 copies and sends to different O-RU$_8$.

There is also a CPRI OAM module; actually this module will work with the processor (Arm) to accommodate the OAM of different O-RU$_8$ and the FHM$_8$ itself.

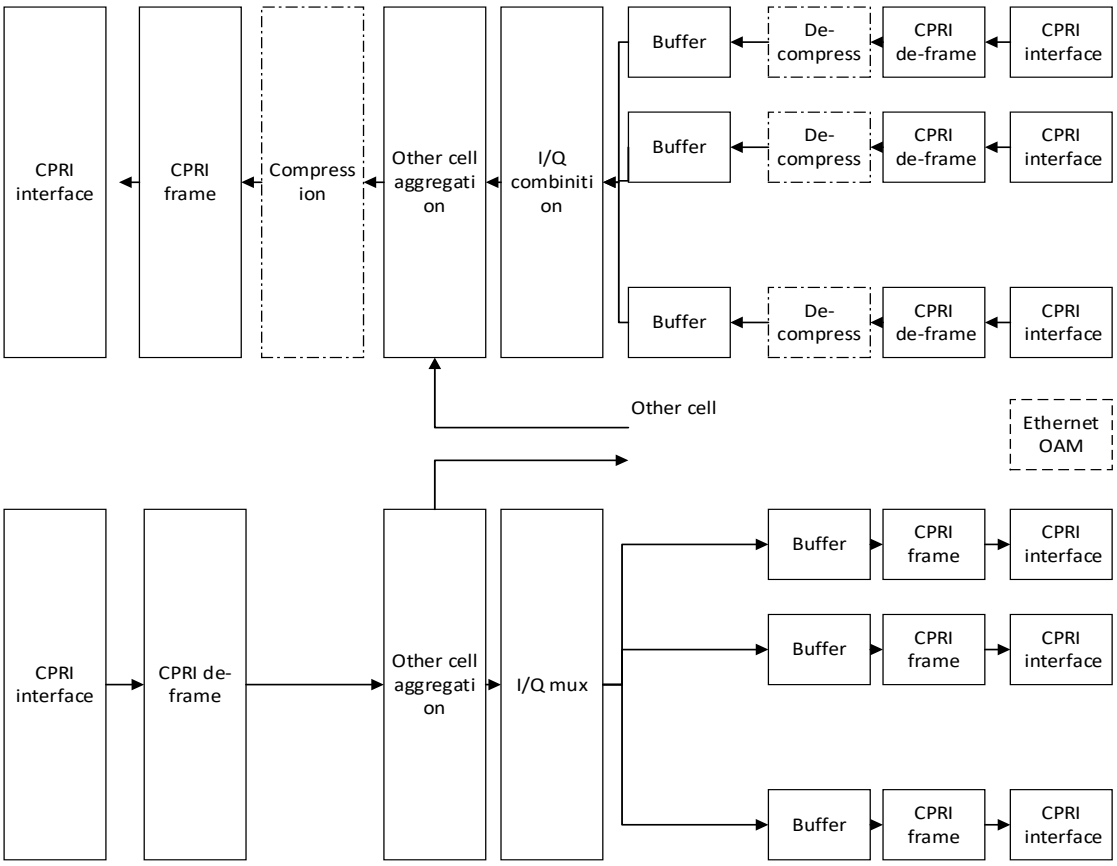


Figure 2-33: FHM8 Digital Processing Block Diagram

The resource needed for the FPGA is listed below:

Table 2-31: FPGA Resource usage for FHM8

Device	FF	LUT	URAM	BRAM	DSP
IQ MUX	1000	1000	0	0	0
Cascade	2000	2000	0	0	0
IQ combine	2000	2000	0	32	0
CPRI deframe*9	45000	45000	0	0	0
CPRI frame*9	45000	45000	0	0	0
Fronthaul(CPRI)x10	47500	31000	0	10	0
compress*1	3000	3000	0	24	32
de_compress*8	24000	24000	0	192	208
ethernet_cm	3000	3000	0	15	0
Other	20000	15000	0	200	0

2.6.2 Synchronization and Timing

In this section we describe the internal and external timing and synchronization that are managed by this entity.

a. Hardware Requirements

- CLK requirement

The FHM_8 may support some kinds of synchronization method:

1) Support GPS/GLONASS/GALILEO/BEIDOU

So the PLL must support 1pps or CPRI CDR as the reference frequency.

Table 2-32: Requirements of the PLL device

Device	Description
1PPS	Supported
Synchronizer number	at least 1
Output channel	At least 5
VCO integrated	Supported

b. Hardware Design

This unit is to recover clock from external source and generate the synchronized clock to other devices. Upon scenarios, there will be external sync source via CPRI.

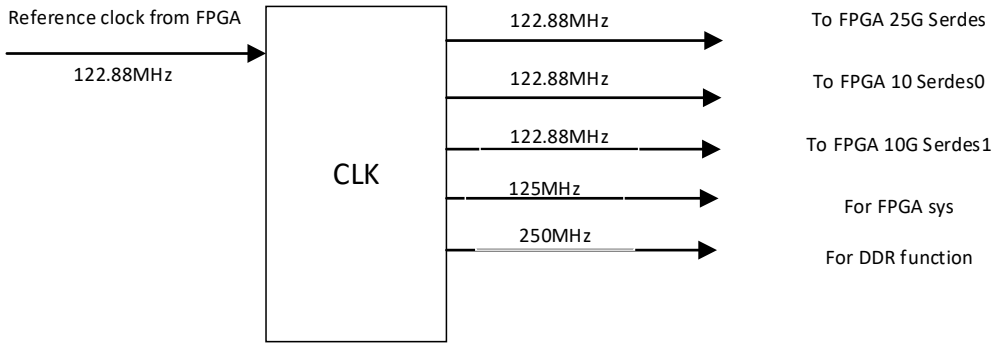


Figure 2-34: CLK reference design for FHM_8

For general FPGA device, each bank may have four SerDes channels. It is better to have separated CLK signal to each bank. Three CLK signals of 122.88MHz are needed for the FHM_8. One CLK signal of 125MHz is used for FPGA system, while one CLK signal of 250MHz is used for DDR device function. The input of the CLK module comes from FPGA CDR function which can get reference CLK from the CPRI line rate. Usually, one PLL device integrated with VCO is needed.

2.6.3 External Interface Ports

List and provide description of all external interfaces.

a. Hardware Requirements

The following table shows the external ports or slots provided by FHM_8.

Table 2-33: External Port List

Port Name	Feature Description
Fronthaul interface	Eight 10Gbps SFP+ transceiver interfaces
	Two 25Gbps SFP+ transceiver interfaces
Debug interface	RJ45 for debug usage
Power interface	PLUG AC FEMALE for power cable

b. Hardware Design

- SFP+ case and connector

The SFP+ case and connector are standardized component on the market; following figure describes the form factor of one SFP case which is integrated with connector.

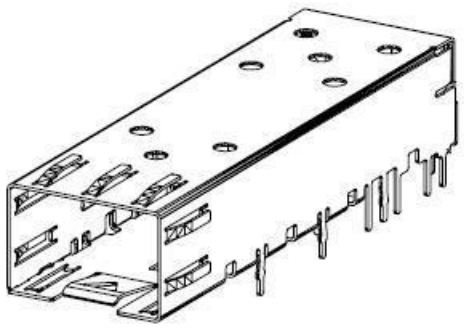


Figure 2-35: SFP+ case and connector

- RJ45 Ethernet interface

The RJ45 Ethernet interface is standardized component on the market; following figure describes the form factor of one RJ45 interface.

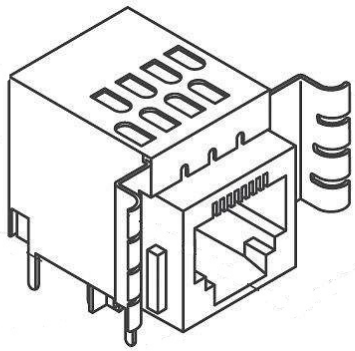


Figure 2-36: RJ45 interface

- Power interface

The 220V AC power connector is standardized component on the market; following figure describes the form factor of one 220V AC power connector.

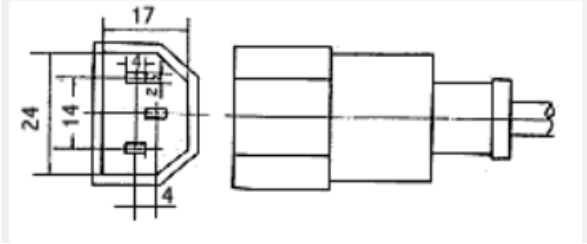


Figure 2-37: AC power interface

2.6.4 Mechanical

The 1U rack mount chassis can contains the layout of the power unit and processing unit. The shell of FHM_8 is showing in the following figure. The power consumption of FHM_8 is huge, so it may need a fan to accelerate the heat dissipation.

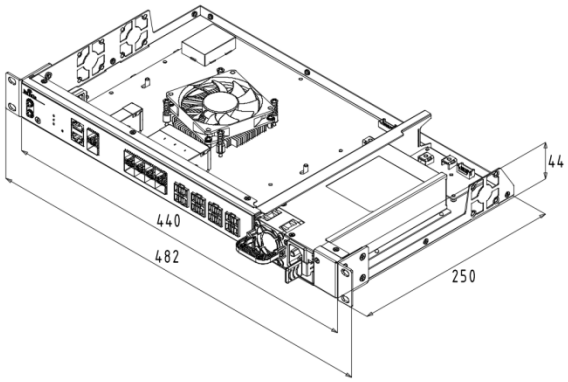


Figure 2-38: Shell Mechanical Diagram

2.6.5 Power

At minimum, fully describe the power consumptions for this white box. Include all AC/DC input or outputs and their ratings. Not mandatory but if you like you can add summary of all component's power requirement and overall white box.

a. Hardware Requirements

- Power requirement: The power solution is divided into two parts.

1. Input power module: This power module must support AC to DC conversion. Usually it converts 220V AC to 48V DC.
2. Output power module: It will supply power to the $O\text{-}RU_8$ connected to the FHM_8 and also the device on the FHM_8 itself.

Table 2-34: Requirements of the power unit

Item	Description
Input voltage	AC:100V~264V
Output voltage	DC: 48V
Output power	At least 600W

b. Hardware Design

Describe hardware reference design for this power unit.

This unit has two main functions which are internal power supply and external/remote power supply. The input power is normally AC (100V to 250V) but DC input could be optional. For remote power supply, it will support -48V DC via either standalone cable or cat-6A cable.

Usually a separated AC-DC power supply is used for the 220V to 48V conversion. It is very common power supply on the market.

It should have fan to cool itself and handle for easy plug in and out.

Then the output power is divided into two portions. One for O-RU$_8$ power supply, the other is for the device on the FHM$_8$ board.

For the on-board power solution, 48V to 12V converter is needed as standard 1/8 brick module. Then the power voltage is further changer to lower level such as 5V, 3.3V, 1.1V, 1.0V and so on by DC/DC or LDO devices.

2.6.6 Thermal

Active cooling.

2.6.7 Environmental and Regulations

The FHM$_8$ will fulfil the requirement in 4.6.3 of [6].

Annex 1 Parts Reference List

According to WG7 scope and charter, component selection for example implementation of white box hardware is allowed for WG7 but would not be mandatory in any Specification.

Recommended O-DU$_8$ parts reference is given in table below:

No.	Part Number	Descriptions
1	Intel® Xeon® Processor D-2183IT	Intel® Xeon® Skylake D D-2100 SoC processor
2	Intel® I350	Intel® 1GbE x4 Ethernet controller
3	Intel® X557-AT2	Intel® 10GbE x2 Ethernet controller
4	Intel® I210	Intel® 1GbE Ethernet controller
5	ASPEED® AST-2500	ASPEED® BMC controller
6	UniIC® SCQ08GU03H1F1C-26V	UniIC® DDR4 8GB
7	Powerleader® MTFDDAK480TDC-1AT1ZABYY	Powerleader® SSD 480GB
8	Accelink® RTXM228-551	Accelink® 10Gbps Optical module

Annex 2 CPRI specification reference design

A2.1 Scope

The present document specifies the CPRI Control Plane, User Plane and Synchronization Plane management plane protocols used over the fronthaul interface in 5G-NR.

A2.2 Reference

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in Release 15.

[1] 3GPP TR 21.905: “Vocabulary for 3GPP Specifications”

[2] Interface Specification: “CPRI Specification Common Public Radio Interface (CPRI)”, V7.0, October 2015.

A2.3 Definitions and Abbreviations

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

Abbreviations	Full name
C&M	Control and Management
CPRI	Common Public Radio Interface
LOF	Lost of Frame
LOS	Lost of Signal
LSB	Least Significant Bit

M-Plane	Management Plane
MSB	Most Significant Bit
O-DU$_8$	O-RAN Distributed Unit
FHM$_8$	Fronthaul Multiplexer
O-RU$_8$	O-RAN Radio Unit
RAI	Remote Alarm Indication
S-Plane	Synchronization Plane
SAP	Service Access Point
SDI	SAP Defect Indication
U-Plane	User Plane

A2.4 High Level Description

A2.4.1 System Architecture

This specification defines the protocol of the fronthaul interface for 5G indoor picocell system.

5G digital indoor distribution system is composed of O-DU$_8$, FHM$_8$ and O-RU$_8$, providing wireless network access function for users. Optical fiber is used to carry signals between O-DU$_8$ and FHM$_8$, and photoelectric composite cable or network cable can be used to carry signals between FHM$_8$ and O-RU$_8$. Due to upgrade evolution, cost and other considerations, optoelectronic composite cable connection is preferred between FHM$_8$ and O-RU$_8$.

Each functional entity in the system architecture is defined as follows:

O-RAN Distributed Unit (O-DU$_8$): A O-DU$_8$ logical node that includes the eNB/gNB functional subset with Split-8x listed in Section 4.2. Mainly realize the modulation and demodulation of baseband signals.

O-RAN Radio Unit (O-RU$_8$): A O-RU$_8$ logical node that includes the eNB/gNB functional subset with Split-8x listed in Section 4.2. Realize the transmission and reception of radio frequency signals. Receive the downlink signal from the FHM$_8$, modulate it into a radio frequency signal and then transmit it through the antenna; receive the radio frequency signal from the antenna, perform corresponding signal processing, and send it to the O-DU$_8$ through the FHM$_8$ for processing.

O-RAN FHM$_8$: Work with O-DU$_8$ and O-RU$_8$. Receive downlink data sent by O-DU$_8$ and transmit to O-RU$_8$ after shunt processing; the uplink data sent by O-RU$_8$ is sent to O-DU$_8$ after a certain junction processing to realize communication with O-DU$_8$.

This specification is based on option 8 CPRI fronthual interface, which can be applied to the interface between O-DU$_8$ and FHM$_8$, as well as the interface between FHM$_8$ and O-RU$_8$. The system architecture is shown in following figure.

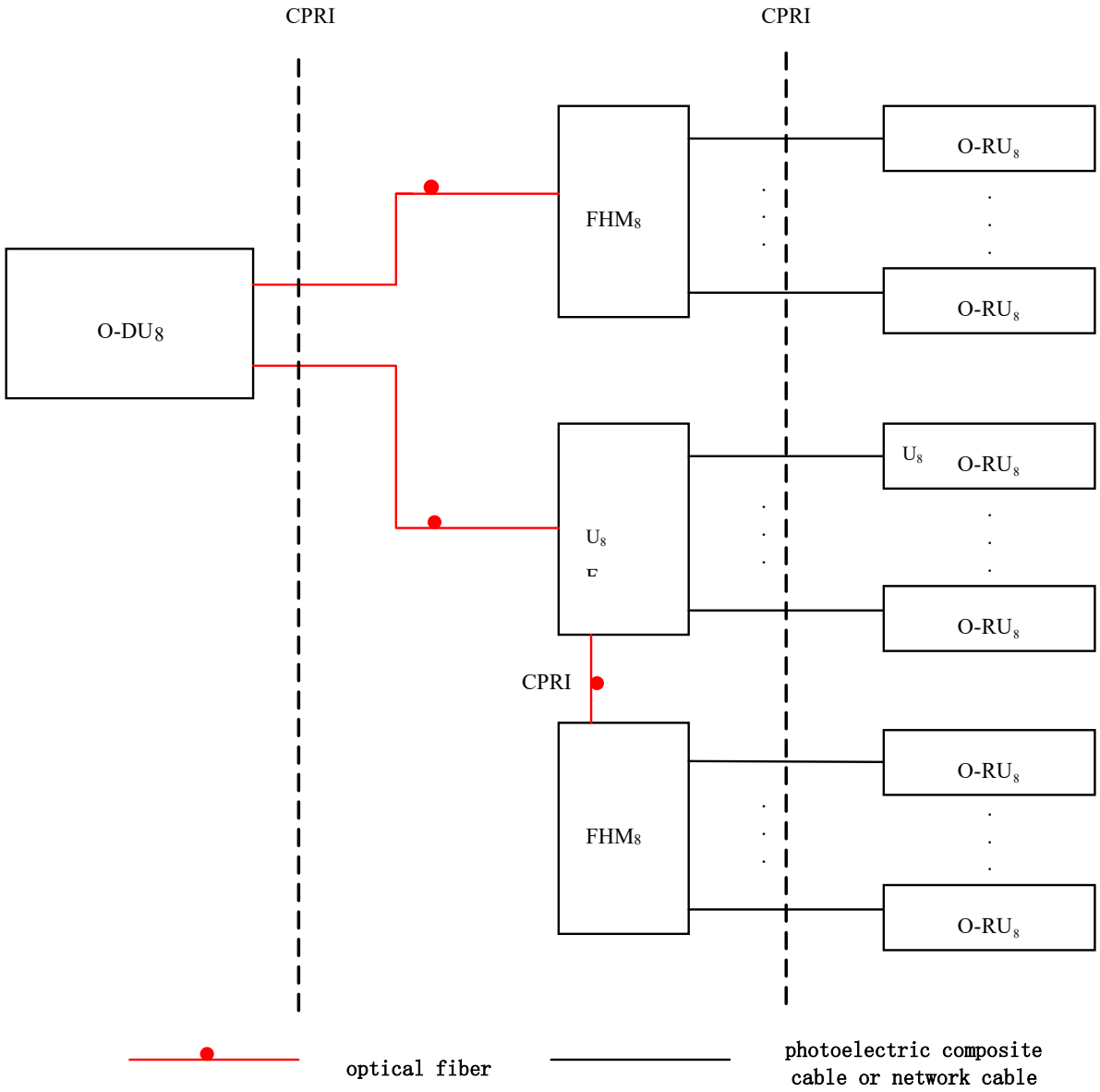


Figure Annex2-1: system architecture

A2.4.2 Functional Split

There are many ways to split the function between O-DU$_8$ and O-RU$_8$. This specification adopts the Option 8 of O-RAN functional split option.

Following figure shows the functional split of the Option 8. In order to show the split point more intuitively, the FHM_8 and $O\text{-}RU_8$ are regarded as a whole, which is represented by $O\text{-}RU_8$ in the figure.

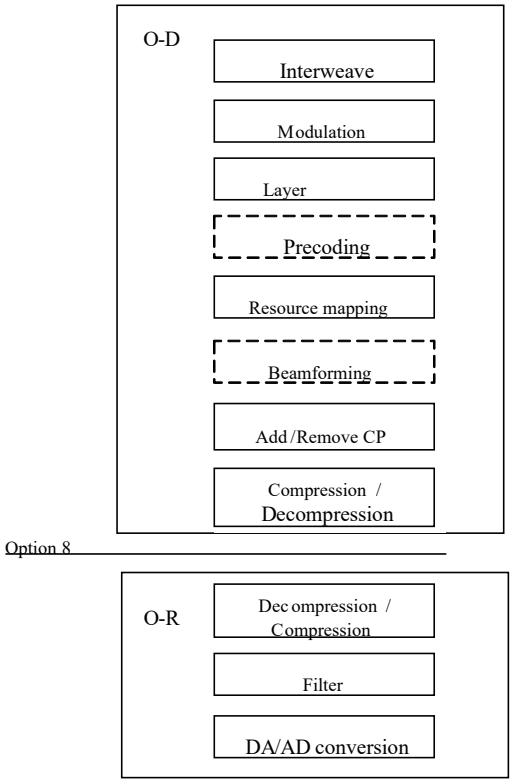


Figure Annex2- 2: option 8 functional split

A2.4.3 Data Flow

In order to exchange information between $O\text{-}DU_8$ and $O\text{-}RU_8$, the data flows required by the fronthaul interface based on the split option 8 can be categorized as follows. Moreover, Table 1 shows the content mapping of the data flow. ID in the table below is just for ease of description.

- User Plane
  - Data flow 1a：IQ data flow in DL time domain;
  - Data flow 1b：IQ data flow in UL time domain.
- Control Plane
  - Data flow 2a：Real-time control command issuance;
  - Data flow 2b：Real-time information reporting from lower layers.
- Synchronization Plane
  - Data flow S：Synchronization information.
- Management Plane

- Data flow M：Management information.

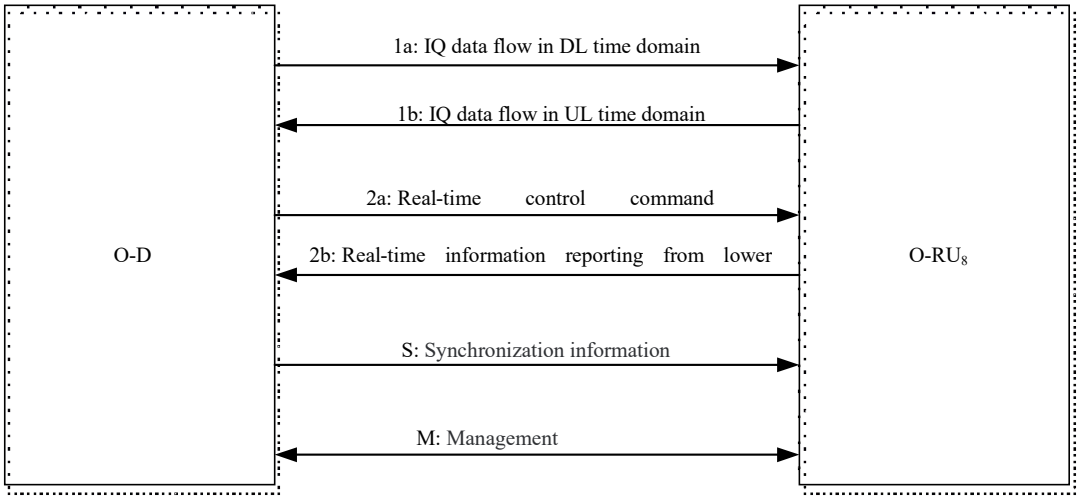


Figure Annex2- 3: Fronthual data flow

Table Annex2- 1 Data flow information mapping

Plane	ID	Name	Content
User Plane	1a	IQ data flow in DL time domain	IQ data flow in DL time domain
	1b	IQ data flow in UL time domain	IQ data flow in UL time domain
Control Plane	2a	Real-time control information	Ethernet speed configuration, RESET command, etc
	2b	Real-time information reporting	Power failure alarm, LOS, LOF, etc
Synchronization Plane	S	Synchronization information	Super frame start indicator, HFN,BFN
Management Plane	M	Management information	Parameter configuration, file management, fault management, etc

A2.5Interface Protocol Overview

Based on the CPRI protocol, refer to section 4.1 of the CPRI v7.0 specification, which is divided into user plane, control plane, synchronization plane and management plane.

The user plane is carried by IQ data. The control plane transmits real-time control information, which is mainly carried by the vendor's specific control words and the L1 inband protocol. The synchronization plane ensures the stability of clock recovery through the assistance of line coding, and at the same time transmits time information through the control word. The management plane is carried by Ethernet.

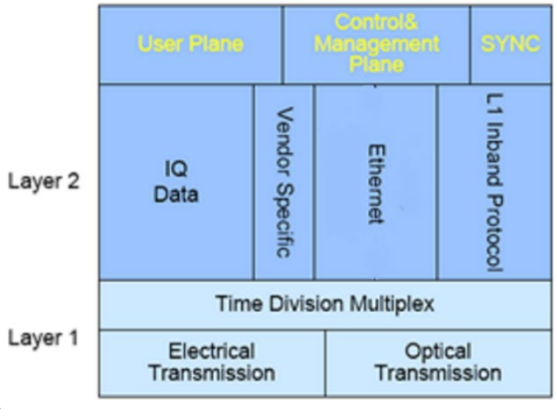


Figure Annex2- 4 : Interface protocol architecture

A2.6 Functional Requirements

A2.6.1 Supported Scenarios

This specification supports CPRI interface to carry 5G NR single-mode transmission. The specific scenarios supported include but are not limited to:

1. The CPRI interface carries a single or two 5G NR cells with a 50MHz bandwidth of 2T2R.
2. The CPRI interface carries a single or two 5G NR cells with a bandwidth of 50MHz and 4T4R.
3. The CPRI interface carries a single or two 5G NR cells with a bandwidth of 100MHz (or 80MHz) and 2T2R.
4. The CPRI interface carries a single or two 5G NR cells with a bandwidth of 100MHz (or 80MHz) and 4T4R.
5. The CPRI interface carries four 5G NR cells with a bandwidth of 100MHz (or 80MHz) and 2T2R.

This specification also supports CPRI interface to carry 5G NR and 4G LTE dual-mode transmission. The specific scenarios supported include but are not limited to:

1. The CPRI interface carries a single 5G NR cell with 100MHz (or 80MHz) bandwidth and 4T4R and a

single 4G LTE cell with 20MHz bandwidth and 2T2R.

2. The CPRI interface carries a single 5G NR cell with 100MHz (or 80MHz) bandwidth and 4T4R and two 4G LTE cell with 20MHz bandwidth and 2T2R.
3. The CPRI interface carries two 5G NR cell with 100MHz (or 80MHz) bandwidth and 4T4R and two 4G LTE cell with 20MHz bandwidth and 2T2R.

A2.6.2 Line Bit Rate

This specification only selects part of the line bit rates and line coding method from the CPRI specification. This specification focuses on the scenario with the maximum capacity that a line rate can carry, that is, only the upper limit is specified.

1. CPRI line bit rate option 10: 24330.24Mbps, 64B/66B line coding (48 x 491.52 x 66/64 Mbit/s).
   Optical fiber is used for transmission, and the optical interface rate used should not be lower than 25G.

   This rate is applicable to the following scenarios and other scenarios with lower rate requirements:

   1) Upper limit of dual-mode: The CPRI interface carries two 5G NR cell with 100MHz bandwidth and 4T4R and two 4G LTE cell with 20MHz bandwidth and 2T2R.
   2) Upper limit of single-mode: The CPRI interface carries two 5G NR cell with 100MHz bandwidth and 4T4R.
2. CPRI line bit rate option 9: 12165.12Mbps, 64B/66B line coding (24 x 491.52 x 66/64 Mbit/s).
   Optical fiber is used for transmission, and the optical interface rate used should not be lower than 12.5G.

   This rate is applicable to the following scenarios and other scenarios with lower rate requirements:

   1) Upper limit of dual-mode: The CPRI interface carries a single 5G NR cell with 100MHz bandwidth and 4T4R and two 4G LTE cell with 20MHz bandwidth and 2T2R.
   2) Upper limit of single-mode: The CPRI interface carries a single 5G NR cell with 100MHz bandwidth and 4T4R.
3. CPRI line bit rate option 8: 10137.6Mbps, 64B/66B line coding (20 x 491.52 x 66/64 Mbit/s).
   If optical fiber and photoelectric composite cable are used for transmission, the optical interface rate should not be lower than 10G. If a network cable is used for transmission, the electrical interface should support a rate of 10GE or above. (This specification temporarily does not make mandatory requirements for electrical interface transmission and its transmission format.)

   This rate is applicable to the following scenarios and other scenarios with lower rate requirements:

   1) Upper limit of dual-mode: The CPRI interface carries a single 5G NR cell with 100MHz bandwidth and 4T4R and a single 4G LTE cell with 20MHz bandwidth and 2T2R.
   2) Upper limit of single-mode: The CPRI interface carries a single 5G NR cell with 100MHz bandwidth and

4T4R.

A2.6.3 Network Form

This specification supports two topologies: star topology and chain topology.

Star topology means that one O-DU$_8$ can connect to multiple FHM$_8$s, and one FHM$_8$ can connect to multiple O-RU$_8$s.

Chain topology means that FHM$_8$ can be cascaded.

A2.6.4 Frame Synchronization

Interface synchronization includes providing accurate frame information, and the FHM$_8$ also needs to provide frame information to subordinate FHM$_8$ and O-RU$_8$. The frame timing information is calibrated by O-DU$_8$ at regular intervals to ensure the normal operation of the entire system.

A2.6.5 IQ Bit Width of User Plane

User plane data is transmitted in the form of IQ data with a width of 16 bits. However, due to the actual bandwidth requirements, IQ data compression technology is required. The compression algorithm and compression ratio are shown in Chapter 8.2.

A2.6.6 Link Maintenance

Four link-related alarms are defined in the CPRI Specification: LOS, LOF, RAI and SDI. For details, please refer to the CPRI v7.0 specification, section 4.2.10.

A2.6.7 Hot Plug Function

Support hot plug function, that is, during O-DU$_8$ operation, the CPRI interface can still resume normal communication after the FHM$_8$ is unplugged and plugged. During the operation of the FHM$_8$, after the O-RU$_8$ connected to it is unplugged and plugged, the CPRI interface between of them can still resume normal communication.

A2.6.8 Working Mode

In this specification, the CPRI interfaces can be applied to interfaces in the following connection modes:

1. O-RU$_8$ connected to FHM$_8$
2. FHM$_8$ connected to O-DU$_8$
3. FHM$_8$ cascaded to FHM$_8$

A2.7 Performance Requirement

A2.7.1 Synchronization

Refer to CPRI V7.0 specification, section 4.2.8.

A2.7.2 Link Delay Accuracy

Refer to CPRI V7.0 specification, section 4.2.9.

A2.7.3 Bit Error

Refer to CPRI V7.0 specification, section 4.2.6.

A2.7.4 Layer 1 Start-up Timer

Refer to CPRI V7.0 specification, section 4.5.2.

A2.8 Interface physical layer

A2.8.1 Frame structure

The frame structure refers to cpriv 7.0 specification, section 4.2.7. As shown in the figure below.

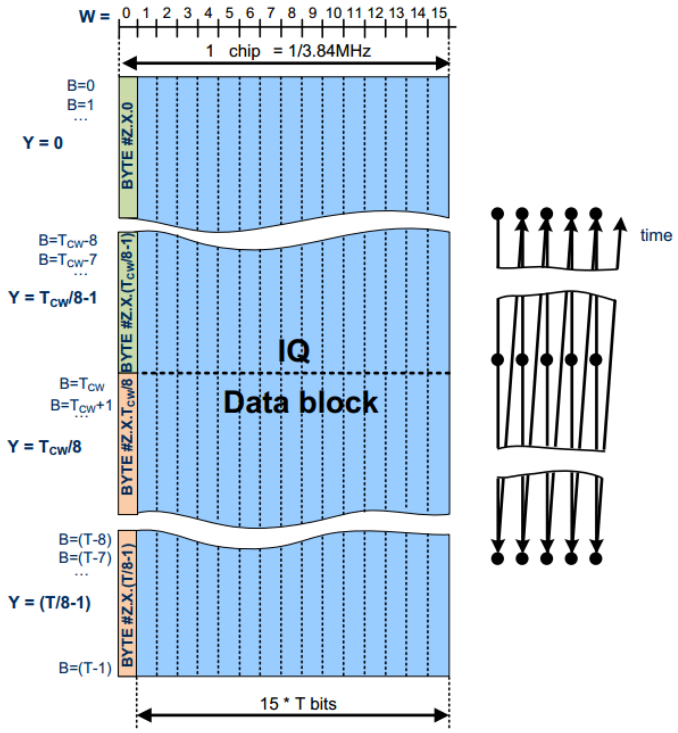


Figure Annex2- 5：CPRI frame structure

The length of a basic frame is 1 TC = 1 / FC = 1 / 3.84 MHz = 260.416667ns. A basic frame contains 16 words, and the corresponding index is w = 0…15。 The length t of each word is related to the corresponding line rate. Each bit in each word has an index B, B = 0 is LSB, B = T-1 is MSB. Every byte in a word is indexed by Y, B = 0 is the LSB of y = 0, B = 7 is the MSB of y = 0, B = 8 is the LSB of y = 1, and so on.

When the line rate is 10137.6Mbps, the range of Y is 0 ~ 19. When the line speed is 12165.12Mbps, the range of Y is 0 ~ 23. When the line rate is 24330.24Mbps, the range of Y is 0 ~ 47.

Table Annex2- 2 Control Word Length

CPRI[Mbit/s]	[bit]	Control word length[bit]	BYTES# Control word contains BYTES
10137.6	T=160	T_{CW}=128	#Z.X.0,#Z.X.1,#Z.X.2,#Z.X.3,#Z.X.4,#Z.X.5,#Z.X.6,#Z.X.7,#Z.X.8,#Z.X.9,#Z.X.10,#Z.X.11,#Z.X.12,#Z.X.13,#Z.X.14,#Z.X.15
12165.12	T=192		
24330.24	T=384		

According to CPRI protocol, the first Tcw bits of the first word (w = 0) in each basic frame are used as control words. If Tcw is less than t, the remaining T-Tcw bits in the word w = 0 can be used to transmit real-time control information.

A2.8.2 IQ data compression

In the specification, we stipulate that IQ data compression is implemented by block floating point compression algorithm, the specific algorithm is listed in the appendix A.1, after compression the bit width of IQ data is eight bits.

The unit of IQ compression is called a compression block. The corresponding compression information bit width (shared exponent of compression block) is four bits in each compression block.

In the specification, all samples of same antenna-carrier in a basic frame are treated as a compression block.

Take an antenna-carrier of one NR cell of 100MHz bandwidth for example, compression block contains 32 samples of IQ data. If CPRI is carrying four antennas of an NR cell, there are four compression blocks in a basic frame.

Take an antenna-carrier of one LTE cell of 20MHz bandwidth for example, compression block contains 8 samples of IQ data. If CPRI is carrying two antennas of an LTE cell, there are two compression blocks in a basic frame.

A2.8.3 IQ data mapping

The unit of IQ data mapping is a basic frame, the total area starts form W=1,Y=0,which contains compression information (shared exponent of compression block), IQ data and automatic gain control(AGC) in basic frame.

In this specification there are default mapping modes of IQ data in different scenarios, which are listed in the appendix B. Start position and end position for IQ data and compression information data for each antenna are configured by M plane parameters.

Samples of each antenna-carrier, compression information data of each compression block and arrangement of bits of AGC should follow the rules below.

1. IQ data. In one sample of antenna-carrier, the bits of I data and the bits of Q data arrange alternately. As shown in the figure 6, I0 is LSB, I7 is MSB.

2. Compression information data. The arrangement of bits in compression information data is shown in the figure 7. E0 is LSB, E3 is MSB in this.

3. Automatic gain control (AGC) data. We use four bits of binary data to describe the link gain of each antenna, Zero corresponds to the maximal link gain. Every time the receiving link gain go down by 2db, the AGC value go up by 1. The field should be filled with default value 0 if it's not used. The internal bit sequence is the same as bit sequence of compression information, as shown in the figure 7, in which E0 is LSB,E3 is MSB.

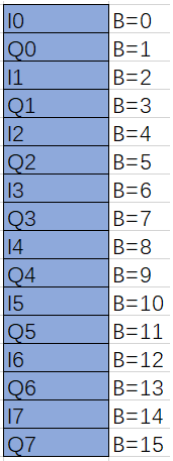


Figure Annex2- 6: IQ arrangement in one sample

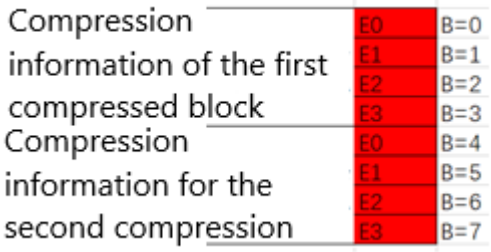


Figure Annex2- 7: Bit Arrangement in Compression Information

A2.8.4 Superframe Structure

Refer to section 4.2.7.3 of CPRI v7.0 specification.

A2.8.5 Subchannel Definition

Refer to section 4.2.7.4 of CPRI v7.0 specification

A2.8.6 Synchronize Data

Refer to section 4.2.7.5 of CPRI v7.0 specification

A2.8.7 L1 inband protocol

Refer to section 4.2.7.6 of CPRI v7.0 specification.

A2.8.8 C&M data channel

This specification support C&M channel option 2 in CPRI protocol: Fast C&M channel that is based on Ethernet. The initial position of Ethernet channel is indicated by pointer p in the hyperframe. Pointer p is carried by #Z.194.0. Refer to section 4.2.7.6 of CPRI v7.0 specification about the definition of pointer p.

In this specification, the value of pointer p is fixed value 20.

A2.8.9 Reserved control word

In order to ensure the extensibility of future protocols, some control words need to be retained. Refer to section 4.2.7.8 of CPRI v7.0 specification.

A2.9Data link layer

The data link layer supports fast C&M channel. For the definition of fast C&M channel, please refer to section 4.2.7.7.2 of CPRI v 7.0 specification.

The data rate of fast C&M channel (i.e. Ethernet rate) is shown in the table below.

Table Annex2-3 Fast C&M Channel Rate

CPRI [Mbit/s]	[bit] Control word length	Control word contains BYTES#	[Mbit/s] Ethernet bit rate

			[Mbit/s] (#Z.194.0 =rr010100)
10137.6， 12165.12, 24330.24	128	#Z.X.0,#Z.X.1,#Z.X.2,#Z.X.3,#Z.X.4,#Z.X.5,#Z.X.6,#Z.X.7,#Z.X.8,#Z.X.9,#Z.X.10,#Z.X.11,#Z.X.12,#Z.X.13,#Z.X.14,#Z.X.15	337.92

A2.10 Appendix - Compression Methods

A2.10.1 Block floating point compression

The following is the representation of block floating point compression. I sample data of each compression unit and Q sample data of each compression unit are converted to floating point data. A original compression unit contains several (for example 32) sample data that are represented by fixed bit width and signed integers. Each IQ sample is represented by sign and mantissa (for example 8 bits) and a shared exponent (see figure 8) in compressed compression unit.

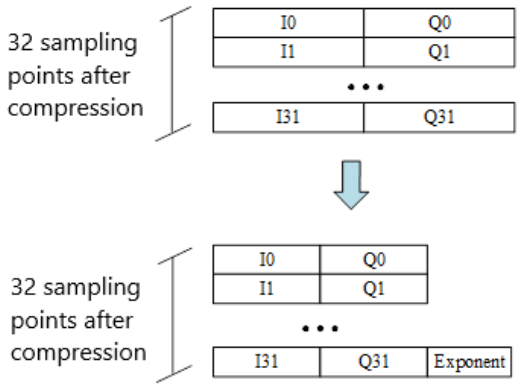


Figure Annex2- 8 : Representation of block floating point compression

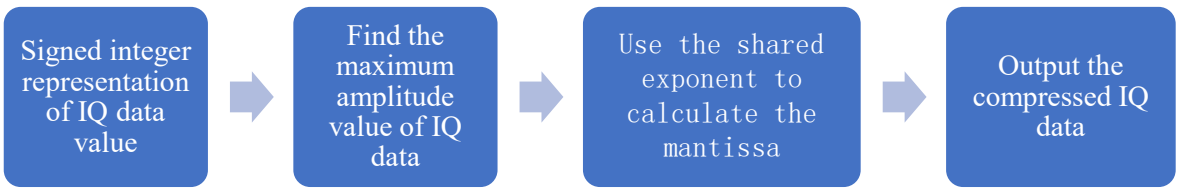


Figure Annex2- 9 Process of block floating point compression

A2.10.2 Block floating point compression algorithm

The following is the pseudocode for a reference implementation of the compression algorithm.

Input

- fBlock - Original data block that is of the size of a compression block, for example,32 samples with the original word length, each sample contains a 16 bits I sample data and a 16 bits Q sample data.
- iqWidth - Compressed word length (contains signed bit), for example, compressed I and Q are respectively represented by 8 bits.

Output：

- cBlock - Compressed data block. The size is a compression block, for example, 32 samples with the original word length
- exponent – shared exponent of compression block. I and Q samples use the same exponent which contains expWidth bits. For example, expWidth = 1+log2(16-8) = 4;

// Find max and min
$maxV$ = max(Re(fBlock), Im(fBlock)), $minV$ = min(Re(fBlock), Im(fBlock))

// Determine max absolute value
$maxValue$ = max($maxV$, |$minV$|-1) (msb of negative value can be one higher)

// Calculate exponent
raw_exp = [floor($\log_2$($maxValue$) +1)] (msb of $maxValue$)

// Calculate shift value and limit to positive
$exponent$ = max(raw_exp – iqWidth + 1, 0)

// Determine right shift value
$scaler = 2^{-exponent}$

For iSample = 1:length(fBlock)

//Scale and round:

*Re(*cBlock(iSample)) = Quantize ($scaler$ × Re(fBlock(iSample))) /* mult. could be bit-shift, Quantize could be or-round */

*Im(*cBlock(iSample)) = Quantize ($scaler$ × Im(fBlock(iSample))) /* mult. could be bit-shift, Quantize could be or-round */

End

A2.10.3 Block floating point decompression algorithm

The following is the pseudocode for a reference implementation of the decompression algorithm.

Input：

- cBlock - Compressed data block, which contains, for example, 32 samples with word width iqWidth.
- exponent - Compressed data block 's shared exponent.

Output：

- fBlock - Decompressed data block, for example, 32 samples with the original word length, each sample contains a 16bits I sample data and a 16bits Q sample data.

```
//Determine scaler
scaler = 2^exponent
For iSample = 1:length(cBlock)
    //Scale
    Re(fBlock(iSample)) = scaler × Re(cBlock(iSample))  /* this could be replaced with a bit-shift operation */
    Im(fBlock(iSample)) = scaler × Im(cBlock(iSample))  /* this could be replaced with a bit-shift operation */
End
```

A2.11 Appendix - Typical mode of IQ mapping

A2.11.1 IQ mapping mode(line bit rate of 10137.6Mbps)

This article presents the IQ mappings modes at a line bit rate of 10137.6Mbps for different scenarios.

Scenario 1: One 5G cell with 4T4R antennas and 100M/80M bandwidth at line bit rate of 10137.6Mbps and one 4G cell with 2T2R antennas and 20M bandwidth at line bit rate of 10137.6Mbps, whose IQ data mapping mode is shown in the figure below. The W=1, #Z.X.0-2 area is compression information data(compression block 's shared exponent) area, The W=15, #Z.X.17-19 area is AGC area. The order of data for antennas in each of these two areas is the same as the order in IQ data area.

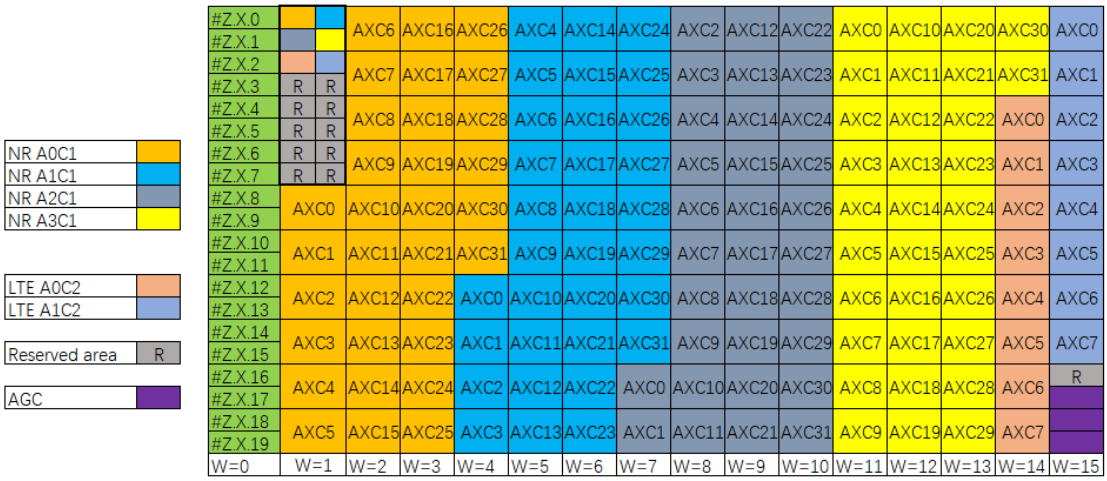


Figure Annex2- 10: Dual mode IQ data compression at a line rate of 10137.6Mbps

Situation 2 : IQ data compression of two neighborhoods by 2T2R with 100M/80M bandwidth at a line rate of 10137.6Mbps is shown in the figure below. The W=1, #Z.X.0-1 area is compression information data(compression block 's public index) area, The W=15, #Z.X.18-19 area is AGC area. The marshalling sequence of the internal antennas of each of the two areas is the same as marshalling sequence of antennas of IQ data area.

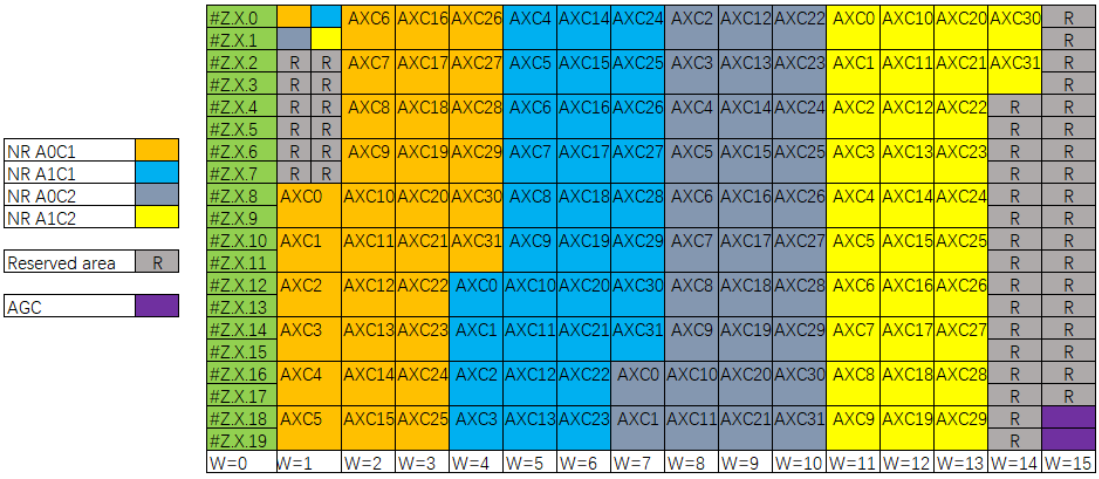
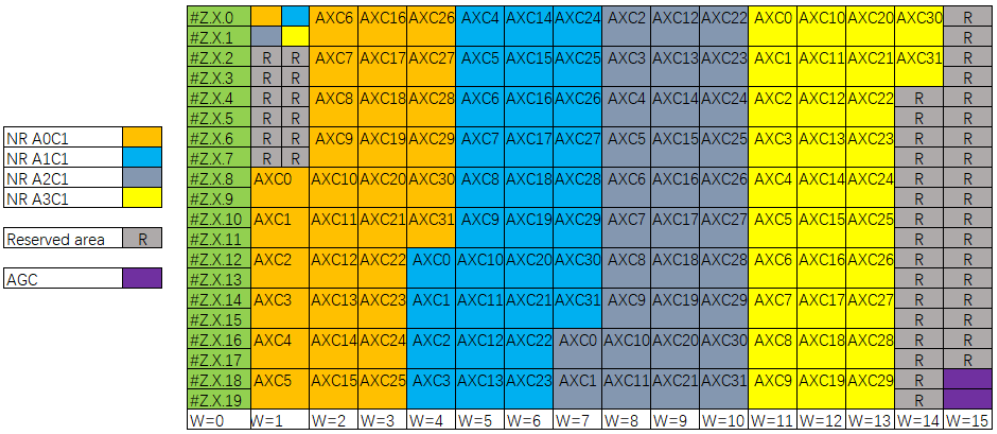


Figure Annex2- 11 : IQ data compression of two neighborhoods by 2T2R with 100M bandwidth at a line rate of 10137.6Mbps

Situation 3: IQ data compression of one neighborhood by 4T4R with 100M/80M bandwidth at a line rate of 10137.6Mbps is shown in the figure below. The W=1, #Z.X.0-1 area is compression information data (compression

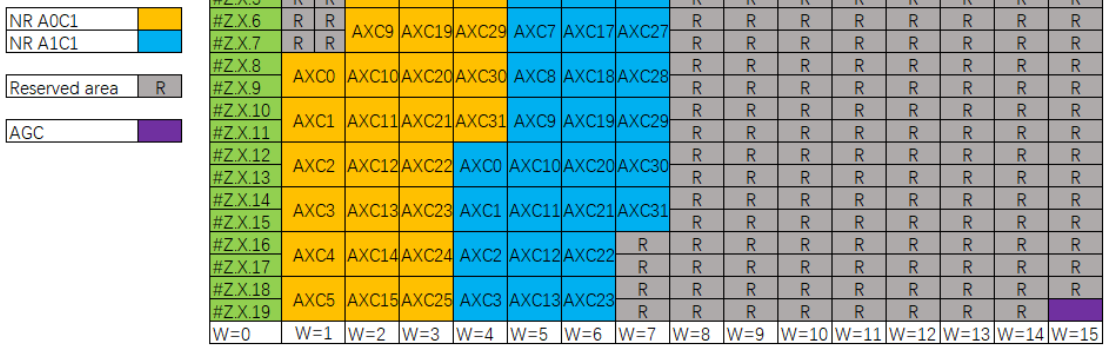
block's public index) area, The W=15, #Z.X.18-19 area is AGC area. The marshalling sequence of the internal antennas of each of the two areas is the same as marshalling sequence of antennas of IQ data area.



	W=0	W=1	W=2	W=3	W=4	W=5	W=6	W=7	W=8	W=9	W=10	W=11	W=12	W=13	W=14	W=15
#Z.X.0			AXC6	AXC16	AXC26	AXC4	AXC14	AXC24	AXC2	AXC12	AXC22	AXC0	AXC10	AXC20	AXC30	R
#Z.X.1																R
#Z.X.2	R	R	AXC7	AXC17	AXC27	AXC5	AXC15	AXC25	AXC3	AXC13	AXC23	AXC1	AXC11	AXC21	AXC31	R
#Z.X.3	R	R														R
#Z.X.4	R	R	AXC8	AXC18	AXC28	AXC6	AXC16	AXC26	AXC4	AXC14	AXC24	AXC2	AXC12	AXC22	R	R
#Z.X.5	R	R													R	R
#Z.X.6	R	R	AXC9	AXC19	AXC29	AXC7	AXC17	AXC27	AXC5	AXC15	AXC25	AXC3	AXC13	AXC23	R	R
#Z.X.7	R	R													R	R
#Z.X.8	AXC0		AXC10	AXC20	AXC30	AXC8	AXC18	AXC28	AXC6	AXC16	AXC26	AXC4	AXC14	AXC24	R	R
#Z.X.9															R	R
#Z.X.10	AXC1		AXC11	AXC21	AXC31	AXC9	AXC19	AXC29	AXC7	AXC17	AXC27	AXC5	AXC15	AXC25	R	R
#Z.X.11															R	R
#Z.X.12	AXC2		AXC12	AXC22	AXC0	AXC10	AXC20	AXC30	AXC8	AXC18	AXC28	AXC6	AXC16	AXC26	R	R
#Z.X.13															R	R
#Z.X.14	AXC3		AXC13	AXC23	AXC1	AXC11	AXC21	AXC31	AXC9	AXC19	AXC29	AXC7	AXC17	AXC27	R	R
#Z.X.15															R	R
#Z.X.16	AXC4		AXC14	AXC24	AXC2	AXC12	AXC22	AXC0	AXC10	AXC20	AXC30	AXC8	AXC18	AXC28	R	R
#Z.X.17															R	R
#Z.X.18	AXC5		AXC15	AXC25	AXC3	AXC13	AXC23	AXC1	AXC11	AXC21	AXC31	AXC9	AXC19	AXC29	R	
#Z.X.19															R	

Figure Annex2- 12 : IQ data compression of one neighborhood by 4T4R with 100M bandwidth at a line rate of 10137.6Mbps

Situation 4: IQ data compression of one neighborhood by 2T2R with 100M/80M bandwidth at a line rate of 10137.6Mbps is shown in the figure below. The W=1, #Z.X.0 area is compression information data (compression block 's public index) area, The W=15, #Z.X.19 area is AGC area. The marshalling sequence of the internal antennas of each of the two areas is the same as marshalling sequence of antennas of IQ data area.



	W=0	W=1	W=2	W=3	W=4	W=5	W=6	W=7	W=8	W=9	W=10	W=11	W=12	W=13	W=14	W=15
#Z.X.0			AXC6	AXC16	AXC26	AXC4	AXC14	AXC24	R	R	R	R	R	R	R	R
#Z.X.1	R	R							R	R	R	R	R	R	R	R
#Z.X.2	R	R	AXC7	AXC17	AXC27	AXC5	AXC15	AXC25	R	R	R	R	R	R	R	R
#Z.X.3	R	R							R	R	R	R	R	R	R	R
#Z.X.4	R	R	AXC8	AXC18	AXC28	AXC6	AXC16	AXC26	R	R	R	R	R	R	R	R
#Z.X.5	R	R							R	R	R	R	R	R	R	R
#Z.X.6	R	R	AXC9	AXC19	AXC29	AXC7	AXC17	AXC27	R	R	R	R	R	R	R	R
#Z.X.7	R	R							R	R	R	R	R	R	R	R
#Z.X.8	AXC0		AXC10	AXC20	AXC30	AXC8	AXC18	AXC28	R	R	R	R	R	R	R	R
#Z.X.9									R	R	R	R	R	R	R	R
#Z.X.10	AXC1		AXC11	AXC21	AXC31	AXC9	AXC19	AXC29	R	R	R	R	R	R	R	R
#Z.X.11									R	R	R	R	R	R	R	R
#Z.X.12	AXC2		AXC12	AXC22	AXC0	AXC10	AXC20	AXC30	R	R	R	R	R	R	R	R
#Z.X.13									R	R	R	R	R	R	R	R
#Z.X.14	AXC3		AXC13	AXC23	AXC1	AXC11	AXC21	AXC31	R	R	R	R	R	R	R	R
#Z.X.15									R	R	R	R	R	R	R	R
#Z.X.16	AXC4		AXC14	AXC24	AXC2	AXC12	AXC22	R	R	R	R	R	R	R	R	R
#Z.X.17								R	R	R	R	R	R	R	R	R
#Z.X.18	AXC5		AXC15	AXC25	AXC3	AXC13	AXC23	R	R	R	R	R	R	R	R	R
#Z.X.19								R	R	R	R	R	R	R	R	

Figure Annex2- 13 : IQ data compression of one neighborhood by 2T2R with 100M bandwidth at a line rate of 10137.6Mbps

A2.11.2 IQ mapping mode(a line rate of 12165.12Mbps)

Situation 1: One NR neighborhood by 4T4R with 100M bandwidth at a line rate of 12165.12Mbps and two LTE neighborhoods by 2T2R with 20M bandwidth at a line rate of 12165.12Mbps, who's the IQ data mapping mode is shown in the figure below in the situation. The W=1, #Z.X.0-3 area is compression information data (compression block's public index) area, The W=15, #Z.X.20-23 area is AGC area. The marshalling sequence of the internal antennas of each of the two areas is the same as marshalling sequence of antennas of IQ data area.

Figure Annex2- 14 : Dual mode IQ data compression at a line rate of 12165.12Mbps

A2.11.3 IQ mapping mode(a line rate of 24330.24Mbps)

This article presents the IQ mappings modes at a line rate of 24330.24Mbps for different situation.

Situation 1: Two NR neighborhoods by 4T4R with 100M/80M bandwidth at a line rate of 24330.24Mbps and two LTE neighborhoods by 2T2R with 20M bandwidth at a line rate of 24330.24Mbps, whose IQ data mapping mode is shown in the figure below in the situation. The W=1, #Z.X.0-5 area is compression information data (compression block's public index) area, The W=15, #Z.X.42-47 area is AGC area. The marshalling sequence of the internal antennas of each of the two areas is the same as marshalling sequence of antennas of IQ data area.

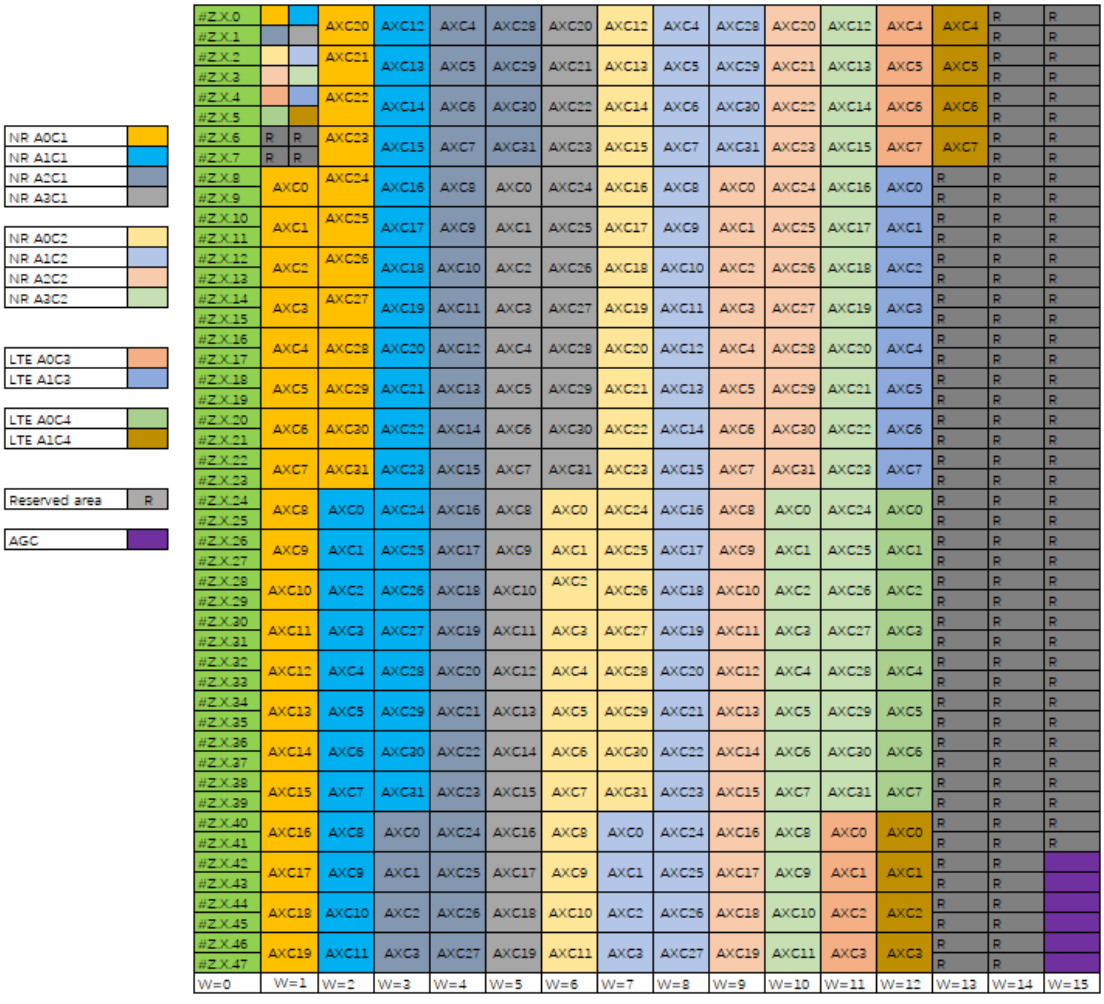


Figure Annex2- 15 : Dual mode IQ data compression at a line rate of 24330.24Mbps

Situation 2: IQ data compression of two neighborhoods by 4T4R with 100M/80M bandwidth at a line rate of 24330.24Mbps is shown in the figure below. The W=1, #Z.X.0-3 area is compression information data (compression block's public index) area, The W=15, #Z.X.44-47 area is AGC area. The marshalling sequence of the internal antennas of each of the two areas is the same as marshalling sequence of antennas of IQ data area.

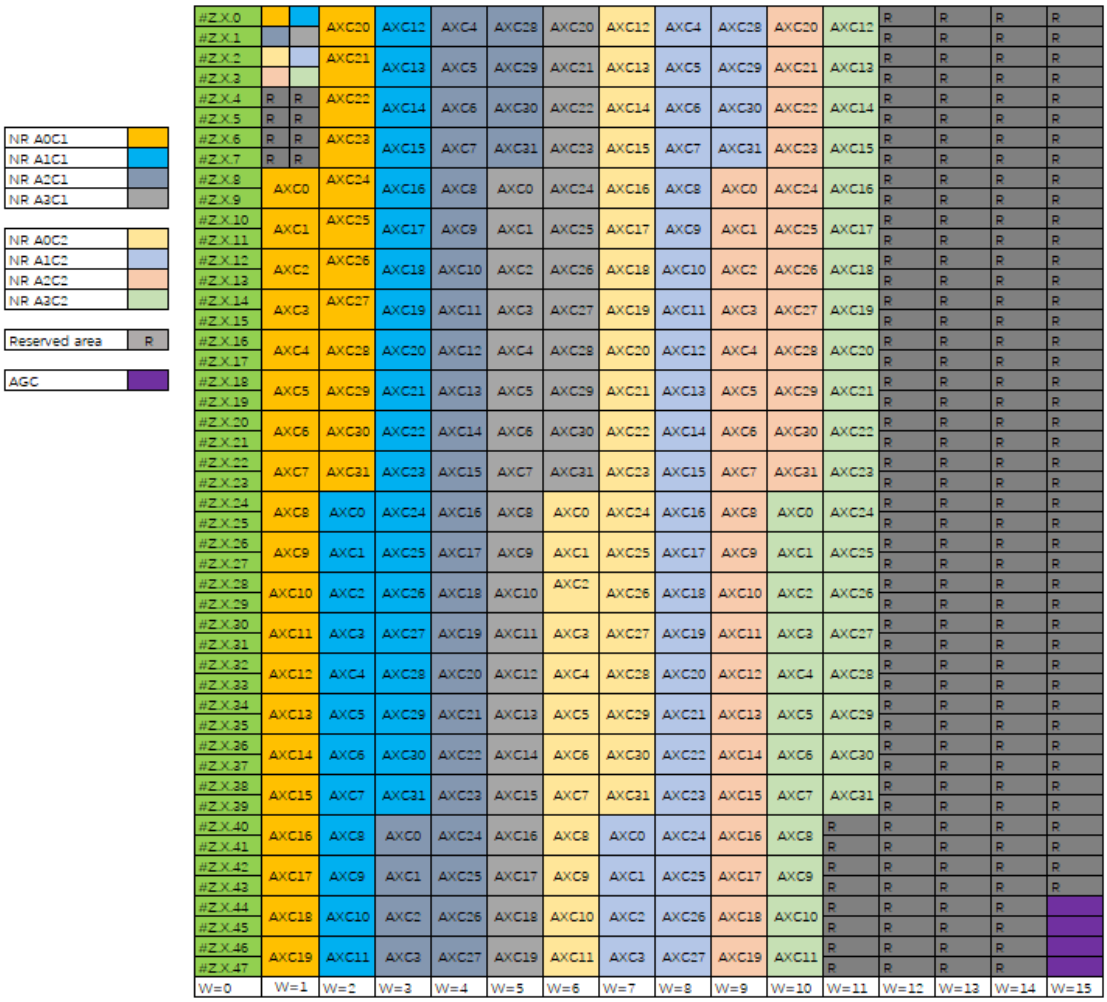


Figure Annex2- 16 : IQ data compression of two neighborhoods by 4T4R with 100M/80M bandwidth at a line rate of 24330.24Mbps

Annex 3 Option 8 CPRI spec reference design – Management plane

A3.1 Scope

This chapter includes management of O-RU_8 based on Option8 CPRI protocol connection.

A3.2 References

[1] 3GPP TR 21.905: “Vocabulary for 3GPP Specifications”

[2] ORAN-WG4.MP.0-v03.00.00

[3] ORAN-WG4.CUS.0-v03.00

[4] RFC 6241, “Network Configuration Protocol (NETCONF)”, IETF, June 2011

[5] SFF-8472v11, “Diagnostic Monitoring Interface for Optical Transceivers”, SFF Committee, September 2010

[6] SFF-8636v2.9.3, “Specification for Management Interface for Cabled Environment”, SFF Committee, April 2019

[7] IEEE 802.1ag, “IEEE Standard for Local and Metropolitan Area Networks Virtual Bridged Local Area Networks Amendment 5: Connectivity Fault Management”, IEEE, 2007

[8] MEF.38, “Service OAM Fault management YANG Modules”, Metro Ethernet Forum, April 2012

[9] G.8275.1 Precision time protocol telecom profile for phase/time synchronization with full timing support from the network ITU June 2016.

[10] G.810 Definitions and terminology for synchronization networks ITU August 1996.[11] 1588v2-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems IEEE 2008

A3.3 Overview of NETCONF Configuration Management

A3.3.1 NETCONF RPC Basic Operation

NETCONF operation layer RPC is shown as following table.

Table Annex3- 1 NETCOF basic operation

Operation	Mandatory /Optional	Description
<get>	mandatory	for querying the status data, you can use the filter for conditional queries
<get-config>	mandatory	for querying configuration data
<edit-config>	mandatory	for modifying the contents of a specified configuration database, supported operations: merge: a merging operation, which is a default operation replace: a replacing operation, which replaces the object if it already exists, and creates one if it doesn't exist create: a creating operation, an error "data-exists" is reported if the object already exists delete: a deleting operation, which deletes the object if it exists, or an error "data-missing" is reported if it doesn't exist remove: a removing operation, which removes the object if it exists, and the operation is ignored if it doesn't exist
<copy-config>	optional	for replicating a source database to a target database
<delete-config>	optional	for deleting a database, wherein a running database cannot be deleted
<lock>	optional	for getting a lock for a specified database, which is optional

<unlock>	optional	for releasing a lock for a specified database, which is optional
<close-session>	mandatory	for closing gracefully a NETCONF connection
<kill-session>	mandatory	for closing compulsively a NETCONF connection

A3.3.2 O-RAN Self-defining RPC Operation

Self-defined RPC of O-RAN is in a content layer, as shown in following table:

Table Annex3- 2 Self-defined RPC of O-RAN

RPC	Mandatory /Optional	Description	YANG Model
activate-beamforming-config	optional	activating a beamforming configuration	o-ran-beamforming.yang
file-upload	mandatory	uploading files to O-DU$_8$	o-ran-file-management.yang
retrieve-file-list	mandatory	obtaining a file list via O-DU$_8$	o-ran-file-management.yang
file-download	mandatory	instructing O-RU$_8$ to download files via O-DU$_8$	o-ran-file-management.yang
reset-interface-counters	optional	indicating O-RU$_8$ to reset an interface counter via O-DU$_8$	o-ran-interfaces.yang
start-measurements	optional	indicating O-RU$_8$ to start a measurement via O-DU$_8$	o-ran-laa-operations.yang
reset	mandatory	indicating O-RU$_8$ to restart via O-DU$_8$	o-ran-operations.yang
software-download	mandatory	indicating O-RU$_8$ to download a	o-ran-software-

		specified software via O-DU$_8$	management.yang
software-install	mandatory	indicating O-RU$_8$ to install a specified software via O-DU$_8$	o-ran-software-management.yang
software-activate	mandatory	indicating O-RU$_8$ to activate a specified software via O-DU$_8$	o-ran-software-management.yang
supervision-watchdog-reset	mandatory	resetting a “watchdog” timer of O-RU$_8$ by O-DU$_8$	o-ran-supervision.yang
start-trace-logs	mandatory	triggering O-RU$_8$ to start to collect trace files	o-ran-trace.yang
stop-trace-logs	mandatory	Triggering O-RU$_8$ to stop collecting trace files	o-ran-trace.yang
start-troubleshooting-logs	mandatory	indicating O-RU$_8$ to start to collect fault location log via O-DU$_8$	o-ran-troubleshooting.yang
stop-troubleshooting-logs	mandatory	indicating O-RU$_8$ to stop collecting fault location log via O-DU$_8$	o-ran-troubleshooting.yang
chg-password	mandatory	resetting account passwords	o-ran-usermgmt.yang

A3.4M-Plane Connection Establishment

A process of establishing M-plane connections is described in this chapter, including: a network layer connection, a secure transport layer SSH connection, and a NETCONF connection.

A3.4.1 Specific Conventions

DHCP related requirements refer to RFC2131， RFC2132

SSHv2 related requirements mainly refer to RFC4252， RFC4253

NETCONF connection related requirments refer to RFC8071， RFC6242， RFC8341

A comparison of the differences between this chapter and the O-RAN M-plane Specification [1] is shown in the following table.

Table Annex3- 3

	O-RAN M-plane Specification	This Specification
O-RU$_8$ identification in DHCP	DHCPv4 Option 60； DHCPv4 Option 124； DHCPv6 Option 16	only DHCPv4 Option 60 is supported
Detection discovery of M-plane client	DHCPv4 OPTION_V4_ZEROTOUCH_REDIRECT DHCPv6 OPTION_V6_ZEROTOUCH_REDIRECT； DHCPv4 Option 43； DHCPv6 Option 17	only DHCPv4 Option 43 is supported
Prefix of vendor-class of O-RU$_8$	o-ran-ru o-ran-ru2	o-ran-ru
TCP port number for establishing SSH connection	Port 830	supported

Authentication way	password authentication X.509 certificate	only password authentication is supported default username: oranuser default password: o-ran-password
User access privilege	sudo nms fm-pm swm	only sudo is supported
NETCONF capability	Writable-running configuration capability Candidate configuration capability and associated commit operation Activation configuration capability Discard change operation Lock and un-lock operations Confirmed commit capability Cancel commit operation Rollback on error capability Validate capability URL capability XPATH capacity Notifications NETCONF session reuse capability	Writable-running configuration capability XPATH capability Notifications NETCONF session reuse capability

A3.4.2 Pre-conditions of M-plane Connection

Both ends of a CPRI link can be negotiated to a state F according to a specified configuration (see 4.3.1.CPRI Interface Management for details); Ethernet over CPRI is connected and can begin to establish a secure connection to an O-RU$_8$ M-plane.

A3.4.3 Network Layer Connection

A3.4.3.1 O-RU$_8$ identification in DHCP

When an IP address is assigned to the O-RU$_8$ or when O-RU$_8$ controller (O-DU$_8$) information of an M-plane is configured in the O-RU$_8$, a DHCP server uses this information.

The DHCPv4 Option 60 Vendor Class Identifier as defined in RFC2132 can be used as identification of O-RU$_8$ in DHCPv4, as shown in following table:

Table Annex3- 4 DHCPv4 Vendor identifier

DHCPv4	
Option	Vendor Class Option
	60
Vendor Class String	"o-ran-ru/<vendor>", such as "o-ran-ru/vendor" "o-ran-ru/<vendor>/<product-code>", such as "o-ran-ru/vendorA/ORUAA100" "o-ran-ru/<vendor>/<product-code>/<serial-number>", such as "o-ran-17 ru2/vendorA/ORUAA100/FR1918010111"

A3.4.3.2 VLAN detection and discovery

In the current version of this specification, an untagged interface is used for sending and receiving messages of the M-plane by the M-plane.

A3.4.3.3 IP address assignment

Please refer to RFC2131 and RFC4361 for a DHCP discovery process which is involved in obtaining IP addresses of O-RU$_8$ M-plane.

The IP address of O-RU$_8$ M-plane is assigned automatically, and the DHCP server should perform static binding operation, i.e., ensure that the same IP address of M-plane are always assigned to O-

RU8s identified by the same client hardware address, e.g., the IP address of M-plane assigned remains unchanged after an O-RU8 reboot.

A3.4.3.4 M-plane client detection and discovery

The DHCP server uses Vendor Specific Option 43 of DHCPv4 to send NETCONF client information to O-RU8.

Following table defines the encoding format for NETCONF client information when the prefix type of DHCPv4 Option 43 at the Vendor Class is o-ran-ru.

Table Annex3- 5 TLV format of NETCONF client information

O-RU8 uses an **o-ran-ru** prefix to report vendor-class	**DHCPv4 Option 43**		
	Type	**Length**	**Value**
	Type: 0x01-O-RU8 controller IP address	Hexadecimal encoding of a length value in bytes	Hexadecimal IP address

A3.4.4 Secure Transport Layer SSH Connection

A3.4.4.1 Establishing NETCONF Call Home M-plane TCP connection

O-RU8 initiates a NETCONF session to O-DU8 after it obtains address information of M-plane and O-DU8 (NETCONF client) information. If it fails to establish the session, the O-RU8 will perform a Call Home process to O-DU8 repeatedly and periodically according to a timer “recall-home-to-ssh-timer”, until the session is established successfully.

Using port 4334 of NETCONF Call Home client O-DU8 receives the Call Home TCP connection initiated by O-RU8, please refer to RFC8071 for the process.

M-plane of a fronthual interface is established by using a method of SSHv2 password authentication.

A3.4.4.2 Establishing M-plane SSH connection

The O-RU$_8$ serves as the SSHv2 server, and the O-DU$_8$ serves as the client. Before the O-RU$_8$ sends or receives password authentication-based data or any configuration/status data, the O-DU$_8$ (which is also a NETCONF client) should verify and authenticate O-RU$_8$'s ID.

The O-DU$_8$ authenticates the O-RU$_8$ by using public key-based host authentication. Refer to RFC4253 for the main process.

The O-RU$_8$ shall verify and authenticate the O-DU$_8$'s ID before it can receive any configuration or reporting status data sent by the O-DU$_8$. Refer to RFC4253 for the main process.

After the initialization of the starting system, the O-RU$_8$ is configured as a default account, e.g., the default user account is named "oranuser".

It needs to define a default password for the default account and configure the default password in the O-RU$_8$, for example, as "o-ran-password".

A3.4.5 NETCONF Connection

A3.4.5.1 M-plane interchange capability set

The O-RU$_8$ notifies its NETCONF capability in the NETCONF Hello message. The Hello message indicates standard features defined in the supported NETCONF RFC6241, as well as the specific namespace.

The NETCONF capability for interaction between the O-RU$_8$ and the NETCONF client is shown in following table:

Table Annex3- 6 NETCONF capability support

Capacity	Description
writable-running configuration capability	support
candidate configuration capability and associated commit operation	optional

activation configuration capability	optional
discard change operation	optional
lock and un-lock operations	optional
confirmed commit capability	optional（support if the candidate configuration capability and associated commit operation is supported）
Cancel commit operation	optional（support if the candidate configuration capability and associated commit operation is supported）
rollback on error capability	optional
validate capability	optional
URL capability	optional
XPATH capability	support
notifications	support
NETCONF session reuse capability	support

A3.4.5.2 Basic information query

During the startup phase, after the NETCONF connection is established, by the NETCONF configuration query process, the O-DU$_8$ obtains the basic information of the O-RU$_8$ as follows:

hw/hardware/component

mfg-name: name of O-RU$_8$ manufacturer;

serial-num: serial number of O-RU$_8$;

software-rev: O-RU$_8$ software component.

o-ran-hardware/hardware/component/

product-code: product code defined by O-RAN;

o-ran-operations/operational-info/declarations

supported-mplane-version: version of O-RAN M-plane interface;

supported-cusplane-version: version of O-RAN CUS-plane interface;

supported-header-mechanism: type of CU-plane message header supported by O-RU$_8$;

o-ran-operations/operational-state

restart-cause: reasons for the most recent reboot;

o-ran-sync/sync

sync-state: synch state of O-RU$_8$.

A3.4.5.3 Access privilege of M-plane

In order to support interoperability of management control of the fronthual interface, the O-RU$_8$, which serves as an NETCONF server, should use the IETF NETCONF Access 34 Control Model defined by RFC8341.

The O-DU$_8$ accesses the O-RU$_8$ as a NETCONF Client [A1], and uses “sudo” privilege.

Following table defines model operation privilege for the sudo user.

Table Annex3- 7 NETCONF capability support

Model rules	Sudo user
“urn:o-ran:supervision:x.y”	--X
“urn:o-ran:hardware:x.y”	R-X
“urn:ietf:params:xml:ns:yang:ietf-hardware”	RWX
“urn:ietf:params:xml:ns:yang:iana-hardware”	RWX
“urn:o-ran:user-mgmt:x.y”	RW- note 1
“urn:o-ran:fm: x.y”	R-X
“urn:o-ran:fan: x.y ”	R--
“urn:o-ran:sync: x.y ”	RW-
“urn:o-ran:delay: x.y”	RW-
“urn:o-ran:module-cap: x.y ”	R--
“urn:o-ran:udpecho: x.y ”	RW-
“urn:o-ran:operations: x.y ”	R-X
“urn:o-ran:uplane-conf: x.y ”	RWX
“urn:o-ran:beamforming: x.y”	RWX

“urn:o-ran:lbm: x.y”	RWX
“urn:o-ran:software-management: x.y”	RWX
“urn:o-ran:file-management: x.y”	--X
“ urn:o-ran:message5: x.y ”	RW-
“urn:o-ran:performance-management: x.y ”	RWX
“urn:o-ran:transceiver: x.y ”	RW-
“urn:o-ran:externalio: x.y ”	RWX
“urn:o-ran:ald-port: x.y ”	RWX
“urn:o-ran:interfaces: x.y ”	RWX
“urn:ietf:params:xml:ns:yang:ietf-ip”	RW-
“urn:ietf:params:xml:ns:yang:ietf-interfaces”	RW-
“urn:ietf:params:xml:ns:yang:iana-if-type”	R--
“urn:ietf:params:xml:ns:yang:ietf-inet-types”	R--
“urn:o-ran:processing-elements: x.y”	RW-
“ urn:o-ran:mplane-interfaces: x.y”	RW-
“urn:o-ran:dhcp: x.y “	R--
“urn:ietf:params:xml:ns:yang:ietf-dhcpv6-types”	RW-
“urn:o-ran:ald: x.y”	--X
“urn:o-ran:troubleshooting: x.y”	--X
“urn:o-ran:trace: x.y”	--X
“urn:o-ran:laa: x.y”	RW-
“urn:o-ran:laa-operations: x.y”	--X
“urn:o-ran:antcal: x.y”	RWX
“urn:ietf:params:xml:ns:yang:ietf-netconf-acm”	RWX
“urn:ietf:params:xml:ns:yang:ietf-yang-library”	RWX
“urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring”	RWX
Note 1: the rule list for “urn:o-ran:usermgmt:1.0” denies any NETCONF	

client to read the password leaf node.

A3.4.5.4 Configuring user account

The accounts are all sudo privileges.

Username: a string of 3-32 characters, the first character must be a lowercase letter, and the rest can be lowercase letters or numbers.

Password: a string of 8-128 characters. Characters allowed in the password field include upper and lower case letters, numbers, and special characters: ! $ % ^ ( ) _ + ~ { } [ ] . -.

The O-RU$_8$ configures a default account, for example, the account name of the default user is “oranuser”.

The default account has an account type PASSWORD, in which case the default password needs to be defined and configured in O-RU$_8$, e.g., “o-ran-password”.

A3.4.5.5 Keeping M-plane connection alive

When performing a session with an O-DU8 (NETCONF Client) with “sudo” access privileges, the O-RU$_8$ operates a “watchdog” timer to monitor durability of the session with the O-DU$_8$ (NETCONF Client). The O-RU$_8$ indicates the O-DU$_8$ (NETCONF Client) that its management system is running via NETCONF notifications.

Whenever the O-RU$_8$ establishes a NETCONF session with an O-DU$_8$ with “sudo” privileges, the O-DU$_8$ shall enable the “watchdog” timer by creating a <supervision-notification> subscription operation.

After monitoring the subscription, the O-RU$_8$ should send monitoring notifications repeatedly. The O-DU$_8$ is responsible for sending the <supervision-watchdog-reset>RPC, and the O-RU$_8$ should reply with a timestamp for sending the next notification.

When reset the “watchdog” timer, O-DU$_8$ (NETCONF Client) can set a new value for the “watchdog” timer, without receiving a <supervision-notification> from O-RU$_8$. The new value will take effect immediately based on the contents of the <supervision-watchdog-reset> RPC, and the next <supervision-notification> should be no later than the timestamp provided in the RPC reply.

The O-RU$_8$ will enter a “monitoring failure” state when the O-RU$_8$ fails to receive the <supervision-watchdog-reset>RPC after the “watchdog” timer times out. The O-RU$_8$ should close the “watchdog” timer immediately when all NETCONF sessions between O-RU$_8$ and O-DU$_8$ (NETCONF Client) with “sudo” privileges have been closed.

Basic procedure:

Figure 4-x shows the monitoring subscription and notification process for the NETCONF connection between O-RU$_8$ and O-DU$_8$.

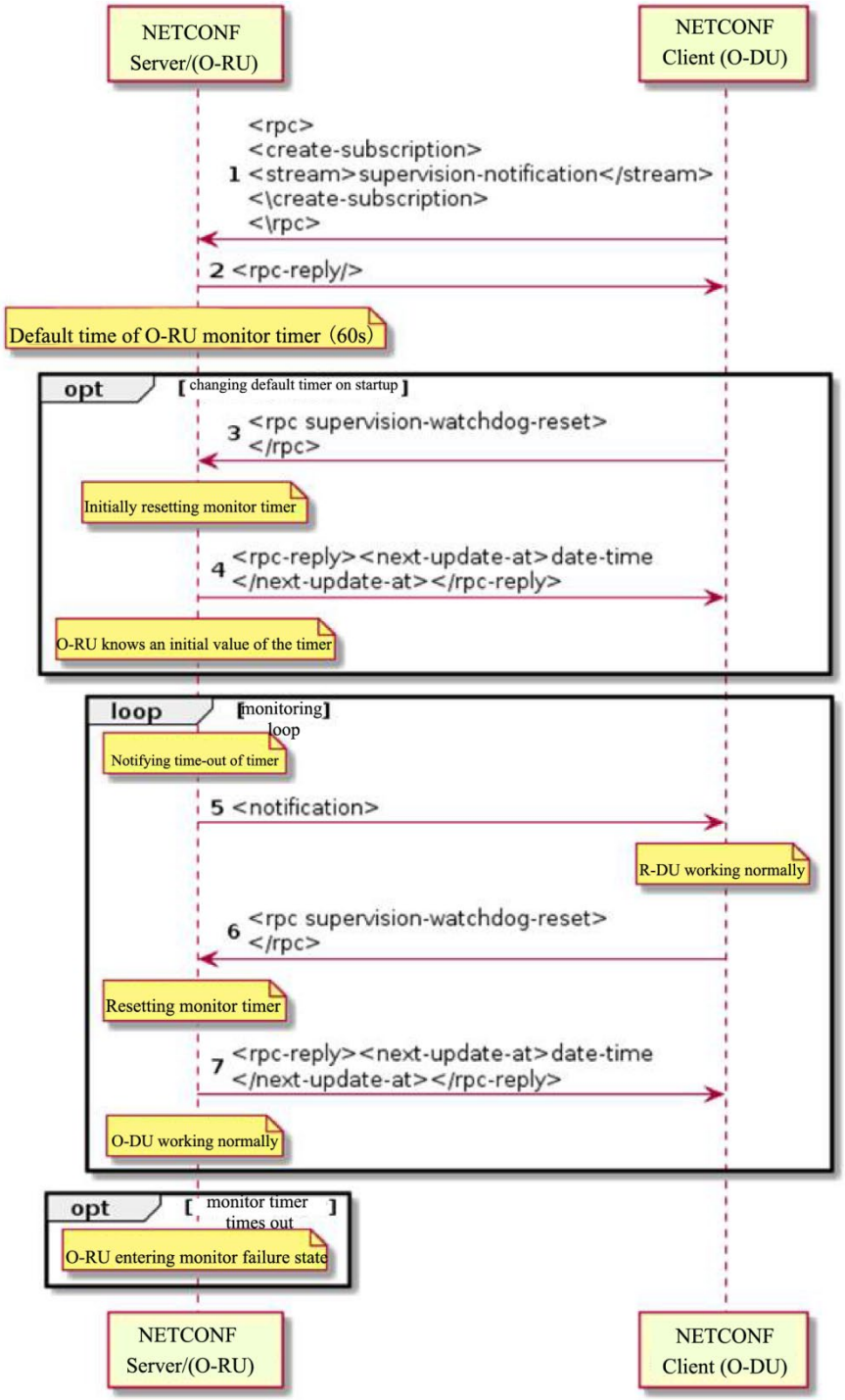


Figure Annex3- 1

1) The DU subscribes to the O-RU$_8$ for monitoring supervision-subscription by rpc: create-

subscription

2) After the RU receives the O-DU$_8$ monitoring subscription, rpc-reply replies with successful 200 or corresponding error code
3) The DU resets the “watchdog” timer by rpc: supervision-watchdog-reset
4) The RU updates reporting time by rpc：next-update-at
5) The RU reports supervision-notification to the O-DU$_8$ (the default time is 60s)
6) The DU resets the “watchdog” timer by rpc：supervision-watchdog-rese after the supervision-notification is received from the O-RU$_8$
7) The RU updates reporting time by rpc: next-update-at

Notes:

- Steps 3-4 may be performed after O-RU$_8$ activation or Step 1.
- Steps 5-7 are repeated.
- The O-RU$_8$ will enter a “monitoring failure” state when the O-RU$_8$ fails to receive the supervision-watchdog-reset after the “watchdog” timer times out.

A3.4.5.6 Disconnecting M-plane

The O-DU$_8$ closes an existing NETCONF session by executing a RPC<close-session> command. The O-RU$_8$ responds to the command and closes an SSHv2 session. The O-RU$_8$ then restarts the call home procedure.

A3.4.5.7 System Time Synchronization

Since the clock in o-ran-operations.yang only synchronizes time zone, it cannot synchronize system time information, o-ran-system-time.yang is added to set the time of FHM$_8$ and O-RU$_8$, which can be used for clock configuration of information such as log.

Basic procedure:

1. Time synchronization is mandatory when a NETCONF session is established. Time synchronization also needs to be triggered when the O-DU$_8$’s system time changes.

2. In order to avoid the time difference between O-DU$_8$ and FHM$_8$ and O-RU$_8$ caused by transmission delay, O-DU$_8$ needs to determine time interval with RPC message when it receives RPC Reply, if the interval is more than 1 second, re-synchronization is required.

A3.5 Interface Management

A3.5.1 CPRI Interface Management

Content of CPRI interface management mainly includes configurating CPRI basic attribute parameters, as well as establishing CPRI U-plane data flow model, and transceiver parameter acquisition and the like.

A3.5.1.1 YANG model

Please refer to o-ran-interface-cpri.yang for CPRI basic attribute parameters configuration.

Please refer to o-ran-processing-element-cpri.yang for CPRI U-plane data flow configurartion.

Please refer to o-ran-tranceiver.yang for transceiver parameter acquisition.

A3.5.1.2 Dataflow model of CPRI U-plane

Dataflow model of CPRI U-plane, o-ran-processing-elements-cpri.yang, which is formed based on extension and definition of the existing O-RAN data flow model, o-ran-processing-elements.yang, specifically includes:

- creating a new type of transport session, and the new transport-session-type is CPRI interface type, CPRI-INTERFACE;

- abstracting an IQ datablock portion in a CPRI frame structure into multiple CPRI dataflows, cpri-flow, depending on the logical function carried;

- specifying a physical CPRI interface corresponding to each CPRI dataflow via interface-name parameters;

- distinguishing different dataflows transported over the physical CPRI interface by different string name parameters; and

- indicating finally that each device can support one or more physical CPRI interfaces, and each physical CPRI interface can carry multiple CPRI dataflows.

Each CPRI dataflow cpri-flow includes the following parameters:

Iq-data-start-bit: identifying a start bit of IQ data for this cpri-flow;

iq-data-end-bit: identifying an end bit of IQ data for this cpri-flow.

Moreover, one additional parameter is required:

compression-flag: for describing whether compression is performed on the IQ data. No if the bit is false, and yes if the bit is true.

If this parameter is set as false, the inband-compression-start-bit and inband-compression-end-bit no longer take effect.

inband-compression-start-bit: for describing a start bit of compression information for this cpri-flow. It is recommended to arrange from the initial position of the CPRI to avoid fragmentation.

inband-compression-end-bit: for describing an end bit of the compression information of this cpri-flow. It is recommended to arrange from the initial position of the CPRI to avoid fragmentation.

agc-start-bit: for describing an agc start bit of this cpri-flow. To avoid fragmentation, it is recommended that the AGC be arranged near the end position of the CPRI to avoid fragmentation. This parameter is ignored when AGC is not used.

agc-end-bit: for describing an agc end bit of this cpri-flow. It is recommended that the AGC be arranged near the end position of the CPRI to avoid fragmentation. This parameter is ignored when AGC is not used.

10.1376Gbps CPRI Basic frame (1/To)

			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	Z	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	S
0	A	Control word	0	160	320	480	640	800	960	1120	1280	1440	1600	1760	1920	2080	2240	Control word	
	B		1	161	321	481	641	801	961	1121	1281	1441	1601	1761	1921	2081	2241		
	C		2	162	322	482	642	802	962	1122	1282	1442	1602	1762	1922	2082	2242		
	D		3	163	323	483	643	803	963	1123	1283	1443	1603	1763	1923	2083	2243		
	E		4	164	324	484	644	804	964	1124	1284	1444	1604	1764	1924	2084	2244		
	F		5	165	325	485	645	805	965	1125	1285	1445	1605	1765	1925	2085	2245		
	G		6	166	326	486	646	806	966	1126	1286	1446	1606	1766	1926	2086	2246		
	H		7	167	327	487	647	807	967	1127	1287	1447	1607	1767	1927	2087	2247		
1	A	Control word	8	168	328	488	648	808	968	1128	1288	1448	1608	1768	1928	2088	2248	Control word	
	B		9	169	329	489	649	809	969	1129	1289	1449	1609	1769	1929	2089	2249		
	C		10	170	330	490	650	810	970	1130	1290	1450	1610	1770	1930	2090	2250		
	D		11	171	331	491	651	811	971	1131	1291	1451	1611	1771	1931	2091	2251		
	E		12	172	332	492	652	812	972	1132	1292	1452	1612	1772	1932	2092	2252		
	F		13	173	333	493	653	813	973	1133	1293	1453	1613	1773	1933	2093	2253		
	G		14	174	334	494	654	814	974	1134	1294	1454	1614	1774	1934	2094	2254		
	H		15	175	335	495	655	815	975	1135	1295	1455	1615	1775	1935	2095	2255		
2-18																			
19	A	Control word	152	312	472	632	792	952	1112	1272	1432	1592	1752	1912	2072	2232	2392	Control word	
	B		153	313	473	633	793	953	1113	1273	1433	1593	1753	1913	2073	2233	2393		
	C		154	314	474	634	794	954	1114	1274	1434	1594	1754	1914	2074	2234	2394		
	D		155	315	475	635	795	955	1115	1275	1435	1595	1755	1915	2075	2235	2395		
	E		156	316	476	636	796	956	1116	1276	1436	1596	1756	1916	2076	2236	2396		
	F		157	317	477	637	797	957	1117	1277	1437	1597	1757	1917	2077	2237	2397		
	G		158	318	478	638	798	958	1118	1278	1438	1598	1758	1918	2078	2238	2398		
	H		159	319	479	639	799	959	1119	1279	1439	1599	1759	1919	2079	2239	2399		

Figure Annex3- 2 Bit position under 10.1376Gbps line rate

The following example 1 is a configuration example with IQ data compressed, where cpri flow 0 corresponding to 100 MHz NR cell's antenna 0 and cpri flow 1 corresponding to 100 MHz NR cell's antenna 1:

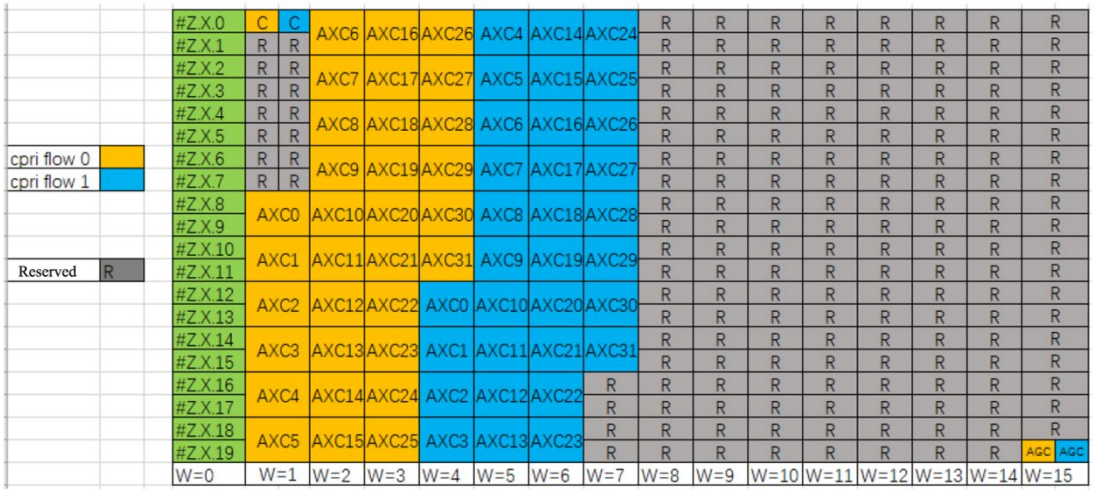


Figure Annex3- 3 10.1376Gbps line rate 100MHz 2T2R mapping relationship

Table Annex3- 8

The following example 2 is a configuration example with IQ data uncompressed, where cpri flow 0 corresponding to 100 MHz NR cell's antenna 0 and cpri flow 1 corresponding to 100 MHz NR cell's antenna 1:

	compression-flag	iq-start-bit	iq-end-bit	inband-compression-start-bit	inband-compression-end-bit	agc-start-bit	agc-end-bit
Cpri flow 0	true	64	575	0	3	2392	2395
Cpri flow 1	true	576	1087	4	7	2396	2399

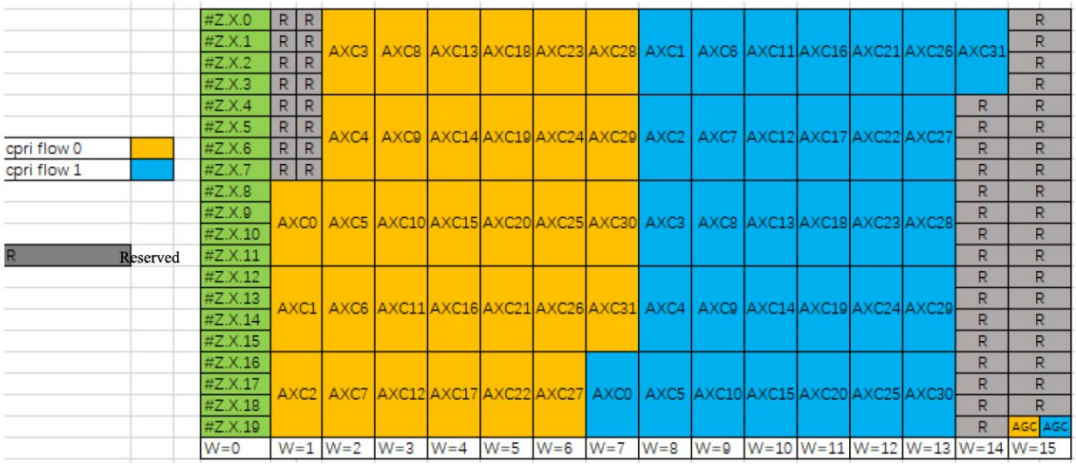


Figure Annex3- 4 Two uncompressed cpri-flows under 10.1376Gbps line rate of example 2

Table Annex3- 9 Parameter configuration of two uncompressed cpri-flows under 10.1376Gbps line rate

	compression-flag	iq-start-bit	iq-end-bit	inband-compression-start-bit	inband-compression-end-bit	agc-start-bit	agc-end-bit
Cpri flow 0	false	0	1087	0	0	2392	2395
Cpri flow 1	false	1088	2111	0	0	2396	2399

A3.5.1.3 Transceiver parameter acquisition

The eCPRI transceiver parameter acquisition defined by O-RAN specification can be used in the CPRI transceiver parameter acquisition.

O-ran-interface-cpri.yang and o-ran-tranceiver.yang are associated via a port-number under cpri-interface, to represent corresponding receiver parameters on the port-number interface.

A3.5.1.3.1 Procedure

A3.5.1.3.2 Transceiver Procedure

Pre-condition:

M-plane connection between the NETCONF client and the O-RU$_8$ is established.

Post-condition:

At the end of this procedure, the NETCONF client obtains the O-RU$_8$ transceiver module information.

Basic procedure:

1) During a start-up phase, the O-RU$_8$ detects the transceiver module and stores data information from a transceiver module in a file;

2) The NETCONF client obtains the file according to an upload process of netconf-yang file.

3) Exception:

- If the transceiver module is inserted during the file upload process, then the process will provide a file with previous content, or fails.
- If the FHM$_8$ is unable to extract data from the transceiver module, or the file does not

exist, then the FHM_8 will not create file or delete the file previously created (note: the file upload will fails because the upload process requests a file that does not exist).

A3.5.1.3.3 M-plane configuration procedure of CPRI interface

Pre-condition:

M-plane connection between the NETCONF client and the $O\text{-}RU_8$ is established.

Post-condition:

At the end of this procedure, the $O\text{-}RU_8$ CPRI U-plane configuration is completed.

Basic procedure:

CPRI link U-plane information is configured according to newly defined o-ran-processing-element-cpri model. The configuration information includes whether to compress IQ data, start bit position of IQ data, end bit position of IQ data, start bit position of compression information, end bit position of compression information, AGC start bit position, AGC end bit position and the like. After completing IQ configuration process for the CPRI interface, CPRI mapping relationship on a southbound CPRI interface is determined. Different IQ configurations are required on the CPRI interfaces for different cell bandwidth and antenna configurations.

A3.6 Software Management

The Software Management function provides a set of operations allowing the desired software build to be downloaded, installed and activated at the $O\text{-}RU_8$. Successful software activation operation does not mean an $O\text{-}RU_8$ is running the just activated software build. An RPC<reset> operation is required to trigger the $O\text{-}RU_8$ to take the activated software build into operational use.A single software build may be considered as set of internally consistent files compliant within such a build. Software build is a subject of versioning and maintenance and as such cannot be broken. Replacement of files within a build is prohibited, as this will cause software version incompatibility. The use of compression and ciphering for the content of the software build is left to vendor implementation. The only file which shall never be ciphered is the manifest.xml file.It is also Vendor's responsibility to handle software build/ package/ file integrity check.The $O\text{-}RU_8$ provides a set of so called "software slot" or "slot". Each slot provides an independent storage location for a single software build. The number of slots offered by $O\text{-}RU_8$ depends on the devices' capabilities. At least two writable slots shall be available at the $O\text{-}RU_8$ for failsafe update operation. Presence of read only software slot is optional. The software slots are resources provided by the $O\text{-}RU_8$ and as such are not the subject of creation and deletion. The size of individual software slot is fixed and determined by the $O\text{-}RU_8$'s vendor and sufficient to accommodate the full software build.

A3.6.1 Specific Conventions

A3.6.1.1 Comparison of specification requirements

A comparison of differences between this chapter and the O-RAN M-plane Specification [1] is shown in the table below:

Table Annex3- 10 Comparison of differences with software management related specification

	specification of O-RAN M-plane	Requirements of the present specification
software slot	at least two writable software slots are supported, and read only software slot is optionally supported	the same as the O-RAN specification
sFTP transport certification	password-based O-RU$_8$ certification and public key(DSA/RSA) list-based sFTP server certification; certificate based O-RU$_8$ and sFTP server certification	password-based O-RU$_8$ certification and public key(DSA/RSA) list-based sFTP server certification

A3.6.1.2 YANG model

Please refer to o-ran-software-management.yang model for relative detailed definitions of NETCONF.

A3.6.1.3 Conception and terms

Translation of basic concepts in this chapter:

Package--software package: a package contains one or more builds.

Build--software compilation package: a build can be considered as set of internally consistent files, and a specific hardware type corresponds to the corresponding build.

Software-slot--software storage area: Each software-slot is an independent storage space, and each software build is stored in a different software-slot. The number of software-slots offered by O-RU$_8$ depends on the device's storage capacity.

Files -- software files.

A3.6.2 Package

The package is delivered by the O-RU$_8$ vendor.

Each package includes:

- manifest.xml
- software files to be installed on O-RU$_8$

The name of package should follow the following format:

"<Vendor Code><Vendor Specific Filed>[#NUMBER].EXT"

Where:

- *Vendor Code* is a mandatory part which has two capital characters,
- *Vendor Specific Field* is a mandatory part and consists of characters used to define a filename. The value must not include character "_" (underscore) or "#" (hash). The value can be defined per vendor for the human readable information. Version information is necessary in the Vendor Specific Field which defines load version,
- *NUMBER* is optional and used when the original file is split into smaller pieces – number after "#" indicates the number of a piece. Numbering starts from 1 and must be continuous;
- *EXT* is a mandatory part which defines the extension of filename. A vendor provides one or more software packages. Each software package shall be compressed by zip.

Note: the name of package needs a <Vendor Code> prefix to avoid issue if two vendors provide files with same name.

Note: the NETCONF client shall support ZIP functionality to enable support of compressed files types.

Note: the operator needs to manage and control which O-RU$_8$ files will be stored and used in the file server, e.g., based on O-RU$_8$ files provided by O-RU$_8$ vendors and network configurations. The operator needs to make sure that only the expected version of O-RU$_8$ files will be transferred from the file server to the O-RU$_8$. Different versions of files for O-RU$_8$ with the same vendor and same product code should be avoided.

The content of the manifest.xml file allows to maintain software update process correctly in terms of compatibility between O-RU$_8$ hardware and software build to be downloaded. The content of the Manifest file prohibits the O-RU$_8$ from installing software builds designed for device based on different hardware. The format of the manifest.xml file is:

```
<xml>

<manifest version="1.0"> ///

@version describes version of file format (not the content)
```

<products>

<product vendor="“XX” **code**="0818820\.x11”

name="RUXX.x11” **build-Id**="1”/>

<product vendor="“XX” **code**="0818820\.x12”

name="RUXX.x12” **build-Id**="1”/>

<product vendor="“XX” **code**="0818818\…” name="RUYY” **build-Id**="2”/>

/// @vendor is as reported by

O-RU[8]

/// @code is a regular

expression that is checked against productCode reported by O-RU[8]

/// @name is optional and

used for human reading – MUST NOT be used for other purposes!

/// @buildId is value of

build@id (see below)

</products>

<builds>

<build id="1” **bldName**="xyz”

bldVersion="1.0”>

/// @id is index of available

builds.

/// @bldName and @bldVersion

are used in YANG (build-name, build-version)

<file fileName="xxxx” **fileVersion**="1.0”

path="full-file_name-with-path-relative-to-package-root-folder” **checksum**="FAA898”/>

<file fileName="yyyy” **fileVersion**="2.0”

path="full-file_name-with-path-relative-to-package-root-folder”

checksum="AEE00C”/ >

/// @fileName and

@fileVersion are used in YANG (name, version)

/// @path is full path (with

name and extension) of a physical file, relative to

package

root folder,used in YANG (local-path)

/// @checksum is used to

```
chech file integrity on O-RU8 side
</build>
<build id="2" bldName="xyz" bldVersion="1.0">
<file fileName="xxxx" fileVersion="1.0"
path="full-file_name-with-path-relative-to-package-root-folder"
checksum="FAA898"/>
<file fileName="yyyy" fileVersion="2.0"
path="full-file_name-with-path-relative-to-package-root-folder" checksum="AEE00C"/>
<file fileName="zzzz" fileVersion="1.5"
path="full-file_name-with-path-relative-to-package-root-folder" checksum="ABCDEF"/>
</build>
</builds>
</manifest>
</xml>
```

Note: Keywords in manifest.xml example are in bold, the keywords must be strictly followed considering of cross-vendor cases.

Note: Correspondency between content of manifest.xml tags and content of o-ran-software-management.yang is:

- XML tag "product vendor" corresponds to content of leaf "vendor-code",
- XML tag "code" corresponds to content of leaf "product-code",
- XML tag "build-Id" corresponds to content of leaf "build-id",
- XML tag "bldName" corresponds to content of leaf "build-name",
- XML tag "bldVersion" corresponds to content of leaf "build-version",
- XML tag "fileName" corresponds to content of leaf "name" in list "files",
-XML tag "fileVersion" corresponds to content of leaf "version" in list "files".

A3.6.3 Software Inventory Query Procedure

Pre-condition:

When M-plane NETCONF session is established, this procedure can be started.

Post-condition:

NETCONF client successfully collected the software inventory information from NETCONF server at the end of this procedure.

Basic procedure:

The figure below shows an example of the software inventory query procedure.

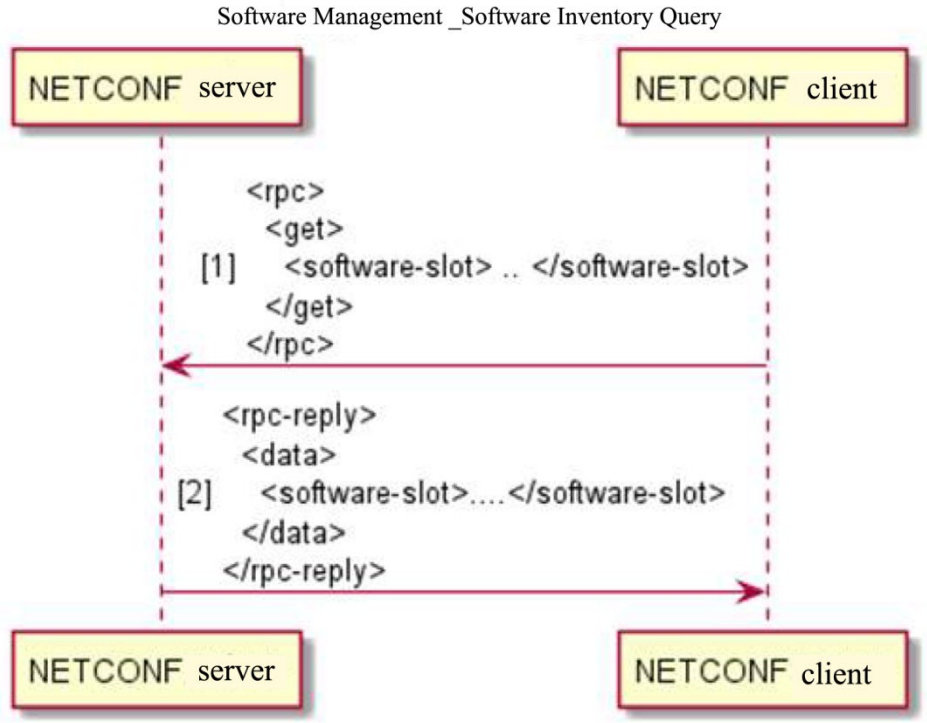


Figure Annex3- 5 Example of software inventory query procedure

1) NETCONF client sends rpc get operation to NETCONF server to query software-slot.

2) The NETCONF server sends rpc-reply to the NETCONF client to return detailed information about the software-slot.

Notes:

- Please refer too-ran-software-management.yang model for detailed information about software-slot.
- If a slot contains a file with integrity NOK, the O-RU$_8$ shall mark the whole slot with status as INVALID.
- The content of a software-slot is fully under O-RU$_8$'s management - including removal of the content occupying the slot (in case the slot is subject of software update procedure), control of file system consistency and so on. The slot content shall not be removed until there is a need for new software to be installed.
- The empty slot parameters shall be set by NETCONF server as follows:

name: up to vendor, not empty;

status: INVALID;

active: False;

running: False;

access: READ_WRITE;

product-code: up to vendor;

vendor code: up to vendor;

build-name: null;

build-version: null;

files: empty.

A3.6.4 Software Download Procedure

Pre-condition:

When M-plane connection is established, software download procedure can be started.

Post-condition:

O-RU$_8$ downloads all files specified and successfully stores the downloaded files in the O-RU$_8$'s file system at the end of this procedure.

Basic procedure:

The figure below shows an example of the software download procedure.

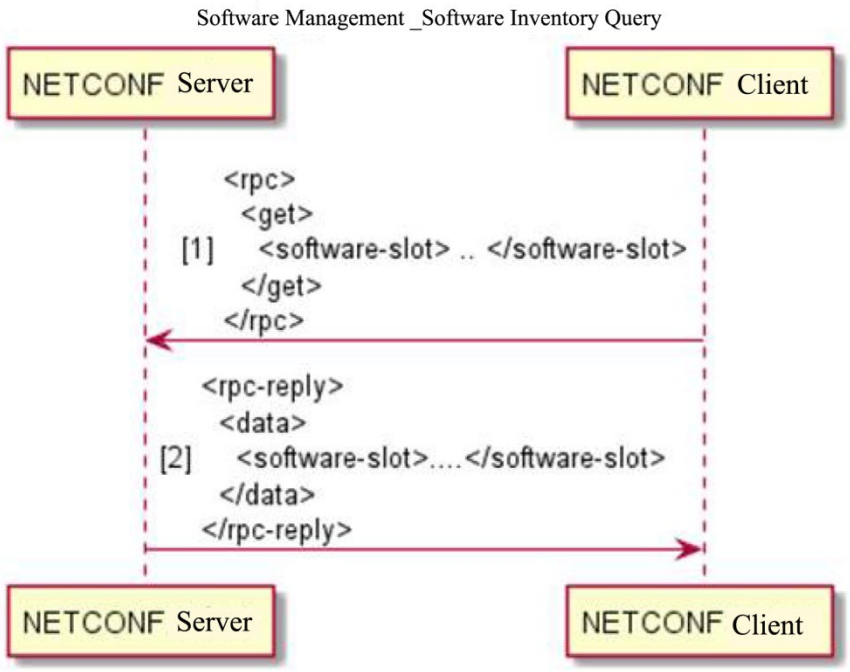


Figure Annex3- 6 Example of software download procedure

1) NETCONF client send software-download operation to NETCONF server.
2) The NETCONF server replies to the NETCONF client with software-download state.
3) O-RU$_8$ transports software files by using sFTP protocol.
4) The NETCONF server sends a download-event notification to the NETCONF client.
5) If there are multiple software files to be downloaded, the above will be repeated until all of files have been downloaded.

Notes:

- The NETCONF client determines the software file to be downloaded according to hardware information of the O-RU$_8$ (vendor number, product number, product name, etc.) and the manifest.xml file of the software package.
- The software download sFTP transmission supports the following types of authentication: password-based O-RU$_8$ authentication and public key (DSA/RSA) list-based sFTP server authentication.

A3.6.5 Software Installation Procedure

Pre-conditions:

- M-plane NETCONF session established.
- At least one software slot with a status of active::False and running::False in O-RU$_8$.
- Software Download has been completed successfully and files are available in O-RU$_8$.

Post-condition:

- O-RU$_8$ software is installed in the specified target software-slot.

Basic procedure:

The software installation procedure is shown in FIG. 32.

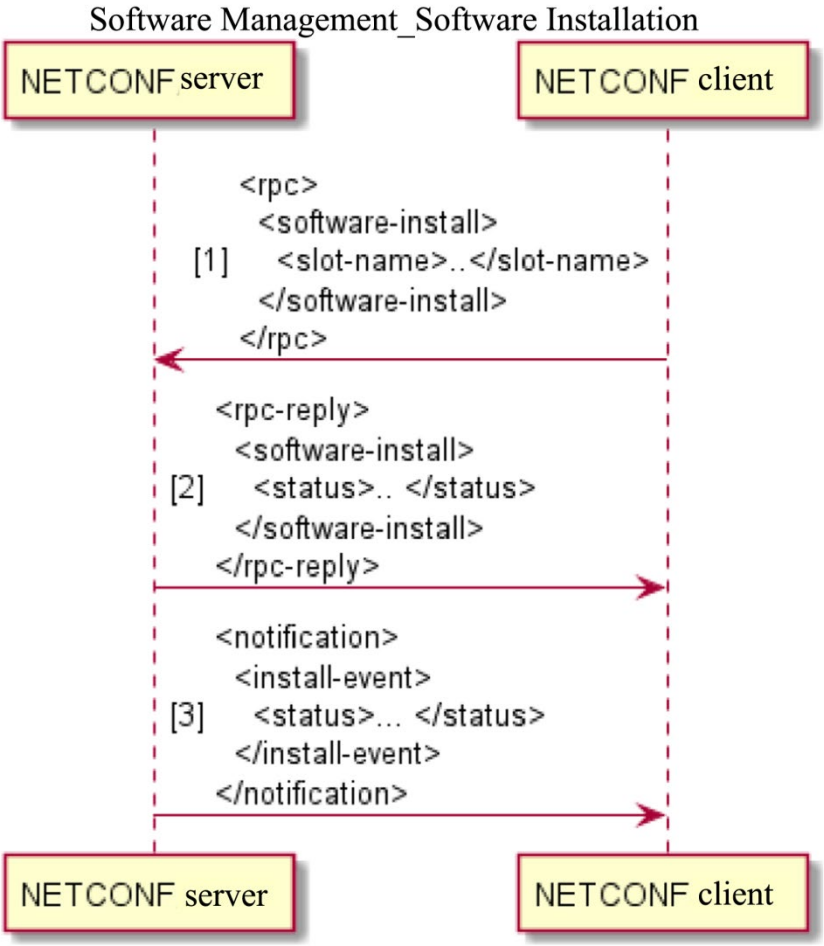


Figure Annex3- 7

NETCONF **software-install rpc** is used to install the previously downloaded software (all files provided in the package) to the specified target **software-slot** on O-RU$_8$. This slot must have status active::False and running::False.

The O-RU$_8$ shall send an immediate rpc-reply message with one of following statuses:

- STARTED: software install operation has been started;
- FAILED: software install operation could not be proceeded, reason for failure in error-message.

When O-RU$_8$ completes the software install or software install procedure fails, the O-RU$_8$ will send NETCONF **install-event** notification with one of the following statuses:

- COMPLETED;
- FILE_ERROR: operation on the file resulted in in error, disk failure, not enough disk space, incompatible file format;
- INTEGRITY_ERROR: file is corrupted;
- APPLICATION_ERROR: operation failed due to internal reason.

When the software install commences, the O-RU$_8$ shall set the slot status to INVALID. After the install procedure finishes, the O-RU$_8$ shall change the slot status to its appropriate status. This operation avoids reporting of inaccurate status when the install procedure is in operation or when it is interrupted (e.g., by spurious reset operation).

A3.6.6 Software Activation Procedure

Precondition:

- M-Plane NETCONF session established.
- Software slot to be activated has status VALID.

Post-conditions:

- O-RU$_8$ software activates to the version of software slot.

Basic procedure:

The software activation procedure is shown in FIG. 33.

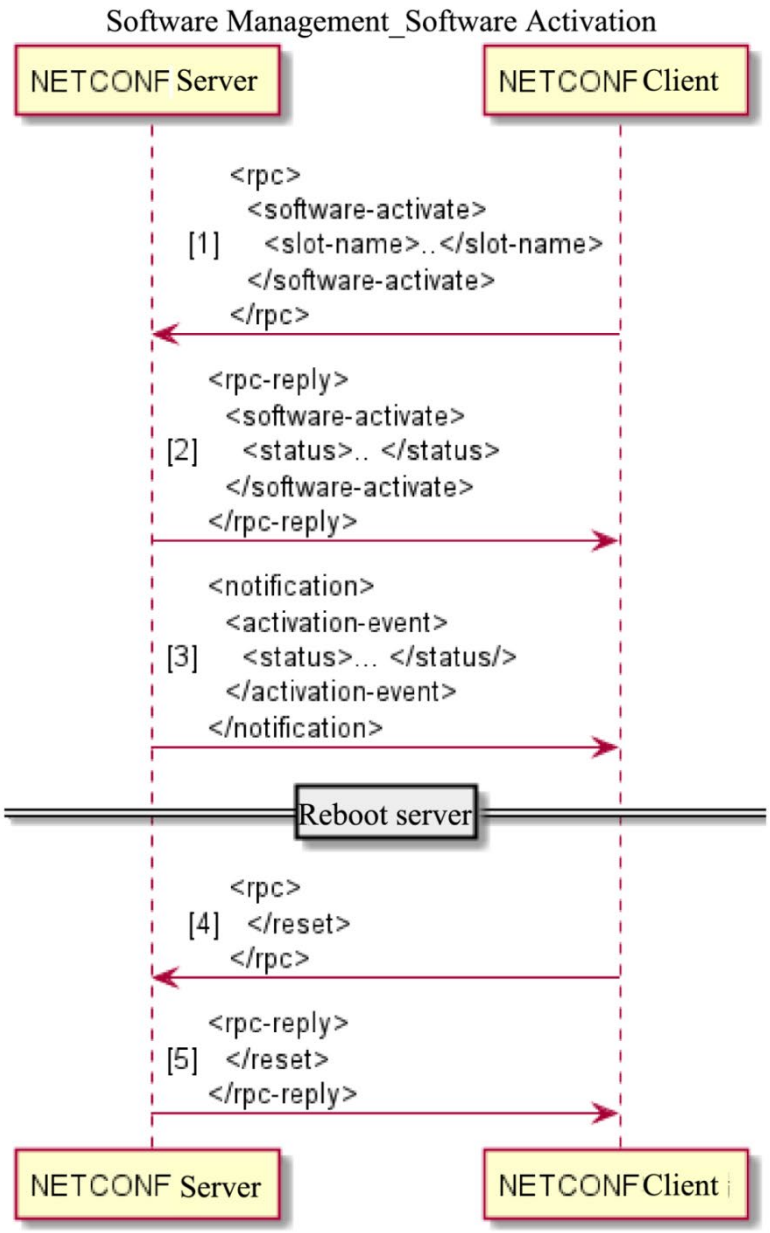


Figure Annex3- 8

NETCONF **software-activate** rpc is used to activate the software. The name of the software-slot is specified in the activate request.

The O-RU$_8$ shall send an immediate rpc-reply message with one of following statuses:

- STARTED: software activation operation has been started;
- FAILED: software activation operation could not be proceeded, reason for failure in **error-message**.

When the activation is completed, the O-RU$_8$ shall send the NETCONF **activation-event** notification with the status of activation. The following status is returned in the NETCONF **activation-event** notifications:

- COMPLETED: activation procedure is successfully completed. O-RU$_8$ must be restarted via NETCONF reset rpc for the new software activated.
- APPLICATION_ERROR: operation fails due to internal reason.

Only one software slot can be active at one time. Thus, **software-activate** command will set active::True to the slot that was provided in the rpc and automatically set active::False to the previously activated slot.

NETCONF **reset** rpc shall be sent to O-RU$_8$ to activate the version of the software slot. O-RU$_8$ restarts and performs startup procedure as described in Chapter 3 as regular startup with new software version running.

A3.6.7 Software Update Procedure

Basic Software Update Procedure is as following:

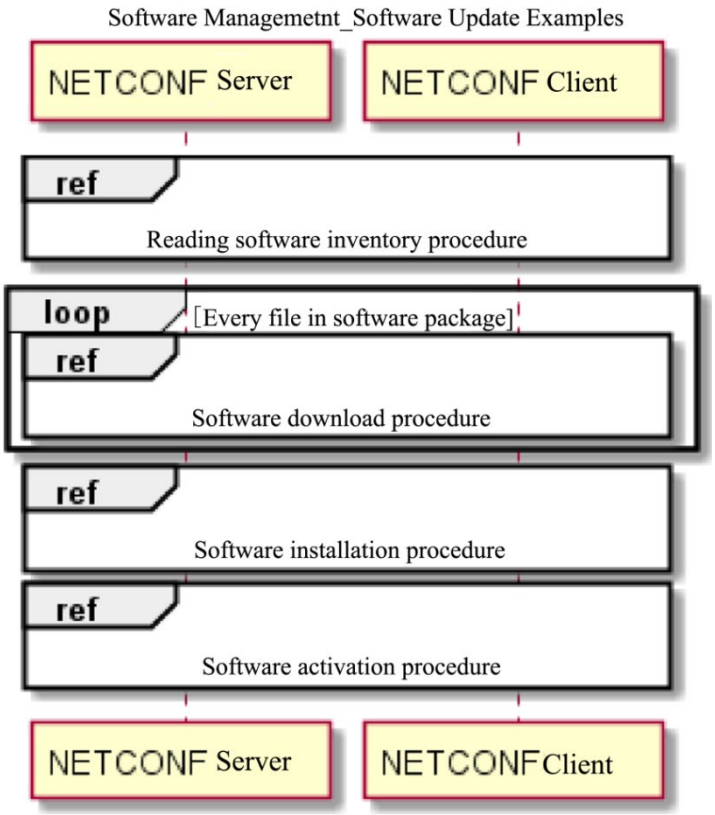


Figure Annex3- 9 Basic procedure of software update

1. NETCONF client performs a software inventory operation and determines that a new software

package is available and can be installed on the O-RU$_8$.

2. NETCONF client using the **software-download** rpc requests the O-RU$_8$ to download a software package (if the software package contains several files, steps 2-4 need to be performed repeatedly until all files have been downloaded).
3. O-RU$_8$ sends rpc response that download was started.
4. O-RU$_8$ finishes downloading the file(s) and reports this by sending the **download-event** notification.
5. NETCONF client requests installation of the software using **software-install** rpc, and provides the slot name where the software needs to be installed along with a list of filenames to be installed (if the software package contains only one file, the list will contain only one entry).
6. O-RU$_8$ sends rpc response that installation was started.
7. O-RU$_8$ sets installation slot status to INVALID.
8. O-RU$_8$ installs the software and after successful installation (with checksum control) changes status of the slot to VALID.
9. O-RU$_8$ notifies the NETCONF client that the installation is finished using **install-event** notification.
10. NETCONF client requests the O-RU$_8$ to activate the newly installed software using the **software-activate** rpc.
11. O-RU$_8$ sends rpc response that activation was started.
12. For requested slot, O-RU$_8$ changes active::True and at the same time sets active::False on previously active slot.
13. O-RU$_8$ notifies NETCONF client about activation finished using the **activation-event** notification.
14. NETCONF client restarts the O-RU$_8$ forcing it to use the newly installed and activated software. The O-RU$_8$ restarts as regular startup with new software version running.

A3.6.8 Factory Reset

O-RU$_8$ can be reset to the factory default software by activating the software-slot containing the factory default software and initiating NETCONF reset rpc. O-RU$_8$ may clear persistent memory data during factory reset as vendor implementation option. This function is optional.

A3.7 Fault Management

A3.7.1 Specific Conventions

This chapter refers to Chapter 8 and Appendix A of the O-RAN M Plane Specification [3]. Because of the general nature of alarm management, this chapter basically follows the original specification in terms of functions, procedures and options.

In order to improve the readability of the Chinese version, the content of the original text has been modified and the fault levels described in Appendix A have been moved to Section 4.5.2.6. In

addition, in order to maintain the consistency of the revised text, some notes have been added to help reader to understand.

A3.7.2 Overview of Fault

A3.7.2.1 Overall situation

The O-RAN fault management mechanism is based on O-RAN NETCONF management framework and the specification covers alarm management of O-RU$_8$. The mechanism includes two types of operation body, an alarm server and an alarm client which are simplified as Server and Client below in this section.

Operation body:

1) Server: NETCONF server, as defined in Basic Conventions of Chapter 1.

2) Client: NETCONF client, as defined in Basic Conventions of Chapter 1.

The server is the main body for maintaining fault information. It maintains an “active-alarm-list” (including Critical, Major and Minor levels, and excluding WARNING level) and sends “alarm notification” to the subscribing body when the alarms change. The alarm notification contains only changed portion of alarms, not all alarms.

Main operation:

The main functions provided by the server are “Reading active-alarm-list”, “Notifying”, “Subscribing” and “cancel subscription”.

Reading active-alarm-list: the client can use this feature to get all current activity alarm information.

Subscribe/cancel subscription: the client initiates to the server with subscription notification or cancel subscription notification.

Notification： the server sends an alarm notification to the client when the alarm information changes.

A3.7.2.2 Key definition and Description

A3.7.2.3 Granularity and identification of alarm message

The alarm messages are presented at the level of a single alarm.

Each alarm message is identified by the following three tuples: fault-id, fault-source and fault-severity. If the three tuples are the same, it identifies the same alarm; if there is any different tuple, it represents a different alarm.

A3.7.2.4 Fault-id

Fault numbers are divided into two categories: “public” and “manufacturer defined”. Appendix A defines common fault numbers and reserves more numbers for future expansion (Note: Common fault numbers range from 0 to 999). Vendor defined fault number range is [1000 ... 65535].

A3.7.2.5 Fault-source

Alarm notifications reported by NETCONF Server contain element “fault-source” which indicates the origin of an alarm. In general values of “fault-source” are based on names defined as YANG leafs:

- Source (in O-RU$_8$, Examples: fan, module, PA, port)

indicates that fault sources within the O-RU$_8$. Value of “fault-source” is set as name of YANG leafs.

*Note: In case the NETCONF Server reports an “unknown” fault-source, the NETCONF Client can discard the alarm notification.

- Source (outside O-RU$_8$)

Value of fault-source may be empty or may identify the most likely external candidate; for example, antenna.

A3.7.2.6 Fault-severity

Fault levels are defined as three types: critical, major, and minor, without warning, with the following meaning:

Severe: the scope specified in the alarm is no longer available.

Significant: the scope specified in the alarm is available but performance is degraded.

Normal: there are alarms, but no performance is impacted.

A3.7.2.7 Failure source and affected object

Both are concepts of functional body in the context of alarm. The former emphasizes the “source” and the latter (affected-objects) emphasizes the affected object. For example, if failure source is a

quartz crystal oscillator (OCXO), the affected objects may be many of the components defined in the O-RU$_8$.

The English version of the O-RAN specification emphasizes that when the failure source is in the O-RU$_8$, the failure source should not be placed in the affected-objects.

A3.7.3 Reading Alarm Procedure

The server is responsible for maintaining an active-alarm-list. This list does not contain any "warning" level information. When an alarm is detected, it is added into the list; when the cause of the alarm disappears, the alarm will be cleared, i.e., removed from the active-alarm list. Further, if this alarm source is the fault-source of an existing alarm, then all associated alarms are removed from the current fault list.

Operation mode:

The client operates "get" to read the active alarms list by NETCONF rpc, see FIG. 11-1 for procedure.

Reading NETCONF rpc operation is an operation for the full table, it has a strong initialization nature, and it is mainly used in an establishment phase of the alarm management relationship to get the initial situation of the alarms. During the operation, this mode can also be used periodically in low-frequency, to realize periodical synchronization of full tables of alarm management information for both parties, in case of long time step out of the alarm information due to unexpected situation.

For specific use of this method, this specification does not stipulate.

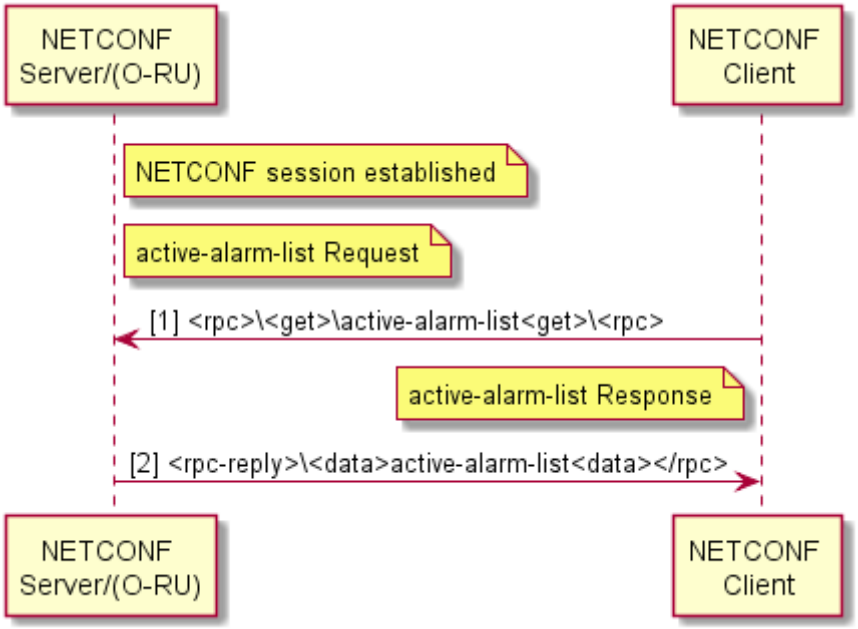


Figure Annex3- 10 Procedure of reading active alarm list

In the O-RAN alarm management system, NETCONF server provides alarm initial information in the form of full table by "obtaining active-alarm-list" operation, which is the "static" part. The client subscribes alarm notifications (with filtering conditions) to the server, the server (based on conditional filtering) finds the alarm changes and updates the local active-alarm-list, and then notifies the client of the alarm changes in form of alarm notifications. A complete alarm notification system is formed by a static part and a dynamic part.

The alarm notification subscription is initiated by the client and responded by the server, which includes two cases, with or without filtering conditions. Definitions of message formats of subscription with and without filtering conditions are given in chapter 4.5.3.2 and chapter 4.5.3.3, respectively. In high-level view, the NETCONF alarm subscription has a brief process shown in FIG. 19.

A3.7.3.1 Subscription message format

The NETCONF Client can "subscribe" to Fault Management Element by sending "create-subscription", RFC5277 [21], to NETCONF Server.

RFC5277 allows "create-subscription" below:

<netconf:rpc netconf:message-id="101"

xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">

```
<create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <filter netconf:type="subtree">
    <event xmlns="http://example.com/event/1.0">
      <eventClass>fault</eventClass>
      <severity>critical</severity>
    </event>
    <event xmlns="http://example.com/event/1.0">
      <eventClass>fault</eventClass>
      <severity>major</severity>
    </event>
    <event xmlns="http://example.com/event/1.0">
      <eventClass>fault</eventClass>
      <severity>minor</severity>
    </event>
  </filter>
</create-subscription>
</netconf:rpc>
```

Note: the NETCONF Client can disable/enable alarm sending only for all the alarms with same severity, not for a single alarm.

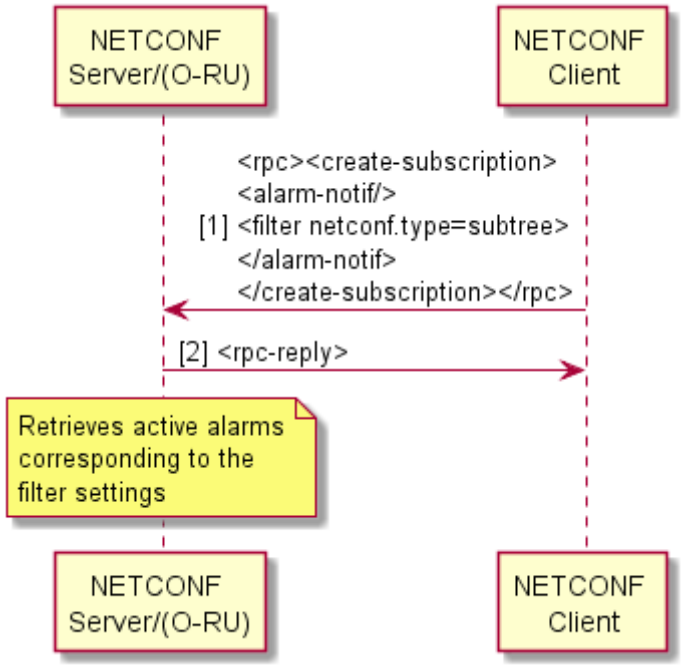


Figure Annex3- 11 Procedure of alarm subscription

A3.7.3.2 Case 1: Subscription request with filter conditions

NETCONF client subscribes “alarm-notif” filtering fault-severity: CRITICAL, MAJOR and MINOR, and “measurement-result-stats” filtering “transceiver-stats” and “rx-window-stats” which measurement-object is RX_ON_TIME only:

```
<rpc xmlns:netconf=“urn:ietf:params:xml:ns:netconf:base:1.0” message-id=“101”>
  <create-subscription
      xmlns=“urn:ietf:params:xml:ns:netconf:notification:1.0”>
    <filter netconf:type=“subtree”>
      <alarm-notif xmlns=“urn:o-ran:fm:1.0”>
        <fault-severity>CRITICAL</fault-severity>
      </alarm-notif>
      <alarm-notif xmlns=“urn:o-ran:fm:1.0”>
        <fault-severity>MAJOR</fault-severity>
      </alarm-notif>
      <alarm-notif xmlns=“urn:o-ran:fm:1.0”>
        <fault-severity>MINOR</fault-severity>
```

```
        </alarm-notif>
        <measurement-result-stats xmlns="urn:o-ran:performance-management:1.0">
            <transceiver-stats/>
        </measurement-result-stats>
        <measurement-result-stats xmlns="urn:o-ran:performance-management:1.0">
          <rx-window-stats>
            <measurement-object>RX_ON_TIME</measurement-object>
          </rx-window-stats>
        </measurement-result-stats>
      </filter>
    </create-subscription>
  </rpc>
```

A3.7.3.3 Case 2: Subscription request without filter conditions

NETCONF client subscribes default event stream NETCONF to receive all notifications defined in O-RAN YANG modules:

```
<rpc xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>NETCONF</stream>
  </create-subscription>
</rpc>
```

A3.7.4 Procedure of reporting alarm

After the NETCONF client subscribes to an alarm notification, the NETCONF server is responsible for sending alarm notifications (alarm-notif) to the client, which involves two cases:

1) a new alarm is detected (note: this may be an alarm the same as the existing alarm with different fault source);

2) an alarm is removed from the active alarm list.

Alarm removed from the active alarm list caused by fault source detection is considered as alarm clearing, which will result in an alarm notification being sent to the NETCONF client. This is applicable for alarms that are explicitly associated with detected fault source. This is to avoid step loss of information between NETCONF clients that could result from a particular client detecting corresponding cell.

The NETCONF server only reports new alarms or alarms that have just disappeared in the alarm notification, not all active alarms every time.

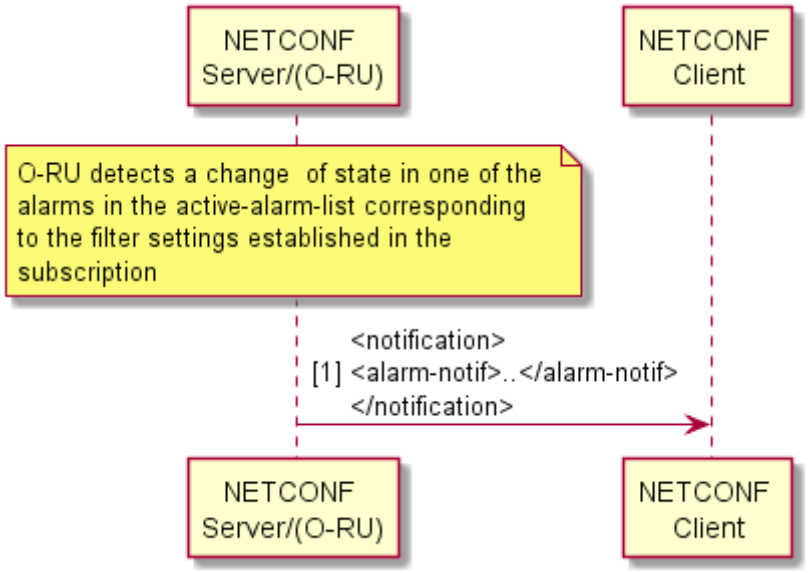


Figure Annex3- 12 procedure of an alarm notification

In short:

Alarms are reported in form of alarm notifications. According to filtering conditions, the server sends alarm notifications to the subscriber (client) when an alarm change required to be notified is detected. The notifications are at the level of a single alarm, not all alarm information is sent.

The alarm updates (the alarm information identification is a ternary group: fault number, fault source, and fault level). When the alarm level needs to be updated, the server sends a new alarm notification first, and then sends the old alarm clearing notification. Example:

When an alarm of fault ID with the same fault source is reported in different fault levels, the alarm level will be upgraded or downgraded, specifically: NETCONF server reports an alarm of “the same fault number, the same fault source, a new fault level, and the is-cleared field of “FALSE”, and clear the previous alarm by reporting “fault ID, fault source, the original fault level, and is-cleared of ‘TRUE” (note: alarm update is achieved by adding new alarms first and then removing the old alarms).

A3.7.5 Procedure of terminating an alarm subscription

If the subscription is to be terminated, the client sends "<close-session>" operation in a subscription session. See FIG. 11-4 for a flowchart.

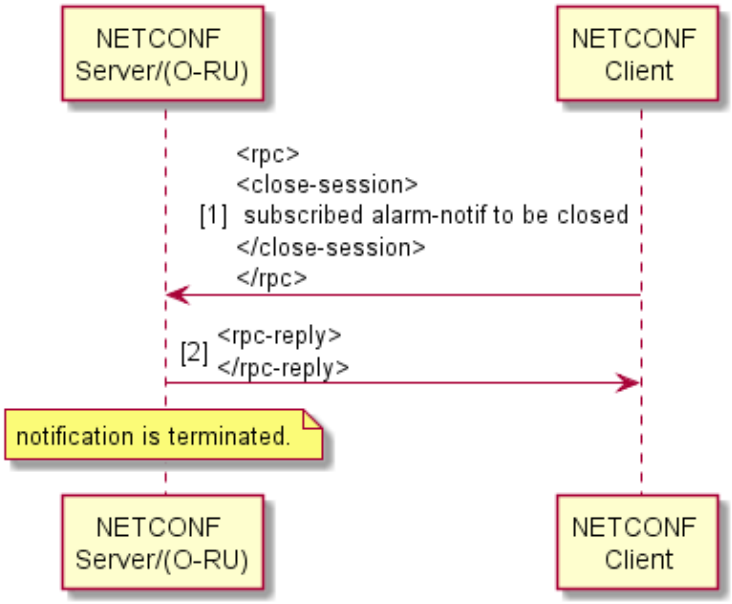


Figure Annex3- 13 Procedure of terminating an alarm subscription

A3.8 File Management

File transfers are done with sFTP. sFTP authenticates with account and password, which is compared with the present specification as following:

Table Annex3- 11

	O-RAN M-plane Specification	Modification herein
Configuration files	a) account and password for authentication b) certificate for authentication	only account and password for authentication

A3.8.1 File System Structure

The file System structure of the O-RU$_8$ is represented as a logical structure that is used by the file management procedures. If the O-RU$_8$'s physical file structure differs from the logical file structure defined below, the O-RU$_8$ is responsible for performing the mapping between the two structures.

A3.8.1.1 Directory definition

The O-RU$_8$ shall support the standardized file directory. The following are definitions of the file directory:

O-RAN/log/

O-RAN/PM/

O-RAN/transceiver/

A3.8.1.2 Specification comparison

Table Annex3- 12

	O-RAN M-plane Specification	Modifications herein
Configuration files	Supporting O-RAN/beamforming/	No supporting O-RAN/beamforming/

A3.8.2 File list retrieve procedure

This subsection describes file retrieve method which the O-DU$_8$/NMS retrieves the file list from the O-RU$_8$. One or multiple files' information can be retrieved by one retrieve file list operation (use of wildcard is allowed). The O-DU$_8$/NMS triggers the retrieve file list operation from the O-RU$_8$.

The following rpc is used for retrieve file list operation:

-rpc:retrieve-file-list

- input

  - logical path: the logical path of files to be retrieved, wild-card is allowed;

  - file-name & file-name-filter: file name or file name filter, * is the wild-card;

- output

  - status: whether O-RU$_8$ accepted or rejected the retrieve file list request;

  - reject-reason: the reason why O-RU$_8$ rejects the request (only applicable if status is rejected);

- file list: file list retrieved;

The following shows retrieve file list sequence diagram:

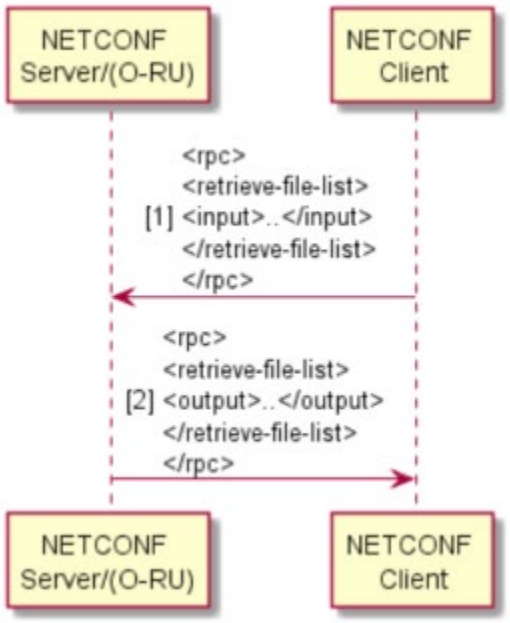


Figure Annex3- 14

A3.8.3 File upload procedure

This subsection describes file upload method from O-RU$_8$ to O-DU/NMS. sFTP is used as upload means, and one file can be uploaded by one upload operation. The O-DU$_8$/NMS triggers file upload operation to O-RU$_8$.

Simultaneous multiple file upload operations can be supported under the same sFTP connection between O-RU$_8$ to O-DU$_8$/NMS. Following rpc parameters is used for upload operation:

-rpc: file-upload

- input

  - local-logical-file-path: the path of file to be uploaded locally (* is allowed as wild-card);

  - remote-file-path: URI of file on the O-DU$_8$/NMS;

- output

  - status: whether O-RU$_8$ accepted or rejected the upload request;

  - reject-reason: the reason why O-RU$_8$ rejects the request (only applicable if status is rejected)

The following shows the file upload sequence diagram:

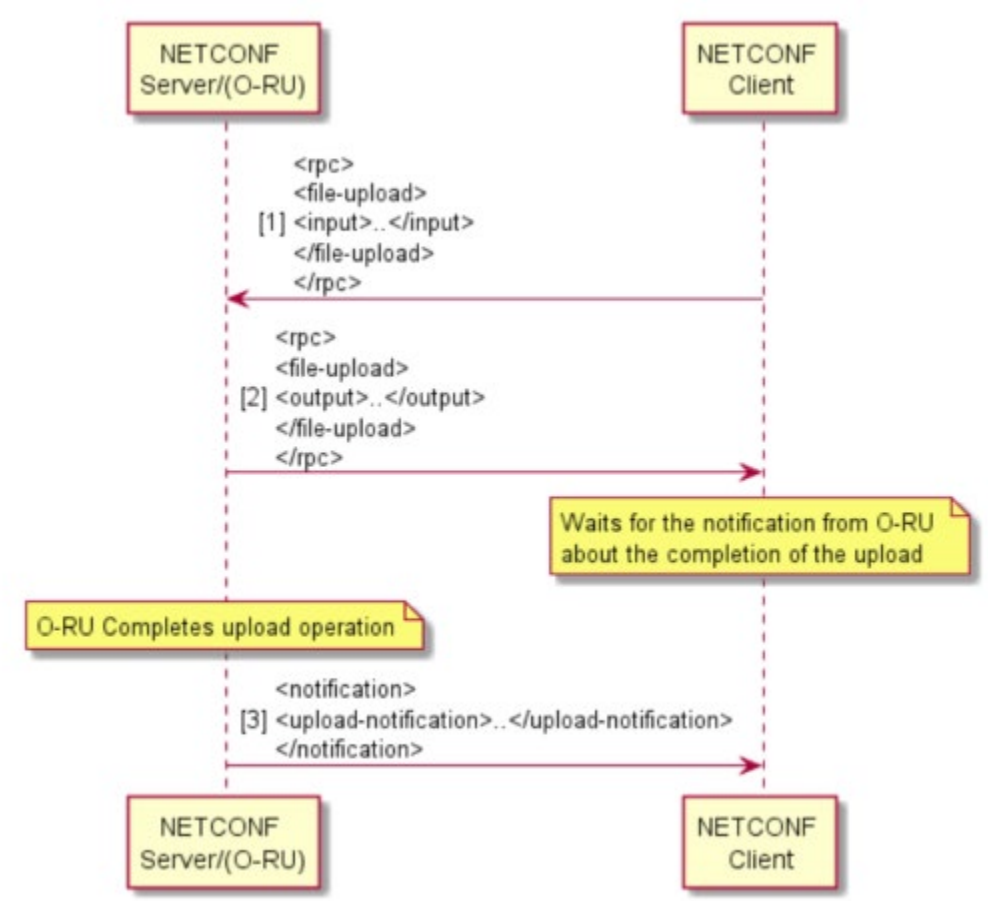


Figure Annex3- 15

A3.8.4 File download procedure

This section describes file download method from O-RU$_8$ to O-DU$_8$/NMS. sFTP is used as download means, and one file can be downloaded by one download operation. The O-DU$_8$/NMS triggers file download operation.

Multiple file download operations can be supported under one sFTP connection between O-RU$_8$ to O-DU$_8$/NMS.

Following rpc parameters is used for download operation:

-rpc: file-download

- input

  - local-logical-file-path: the path of file to be downloaded (* is allowed as wild-card);;

  - remote-file-path: URI of file on the O-DU$_8$/NMS;

- output

  - status: whether O-RU$_8$ accepted or rejected the download request;

  - reject-reason: the reason why O-RU$_8$ rejects the request (only applicable if status is rejected)

The following shows the file download sequence diagram:

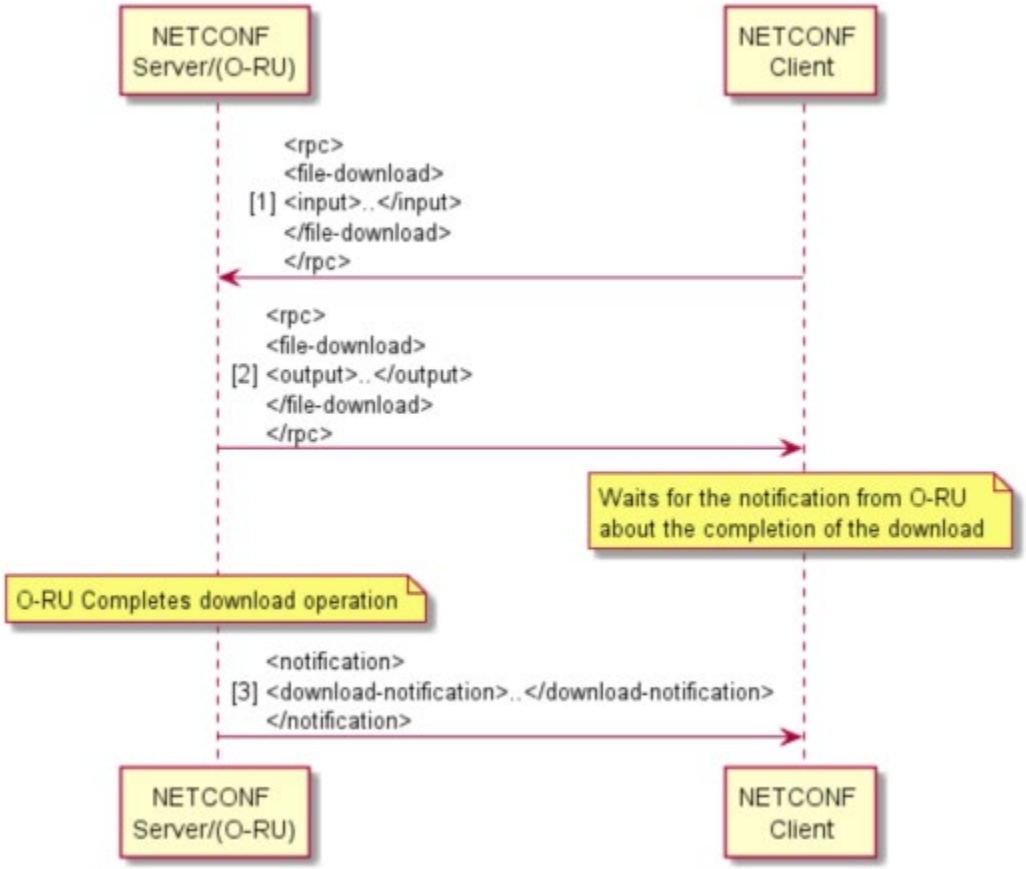


Figure Annex3- 16

A3.9 Synchronization plane management

The O-RU$_8$ is responsible for managing its synchronization status, to select one or more synchronization input source(s) (based on vendor specific implementation) and assure that the resulting accuracy meets that required by the Radio Access Technology being implemented.

A3.9.1 CPRI SYNC

A3.9.1.1 CPRI SYNC configuration

The O-RU$_8$ obtains associated clock from CPRI without configuration of O-DU$_8$.

A3.9.1.2 CPRI SYNC status

CPRI SYNC Status container is used to collect status information about the CPRI as a synchronization source. This information can be used to represent operational information and is useful for the operator to locate faults. This information is not used to obtain alarms automatically by the O-DU$_8$. If the O-DU$_8$ is interested in CPRI SYNC status, it may send NETCONF <subscribe-notification> to the NETCONF Server in the O-RU$_8$. Notifications are reported merely when the lock-state is changed. Before obtaining CPRI status information, O-DU$_8$ needs to make sure that the CPRI is supported by the O-RU$_8$ by querying supported-timing-reference-types. The following table contains related parameters of this container.

reporting-period

This parameter is used to define minimum time interval in seconds, sent by NETCONF Server.

default: 10

lock-state

This parameter is used to indicate whether the O-RU$_8$'s clock system is locked in a line up state of a reference clock and a CPRI link recovered by a CPRI associated clock. The locked and unlocked states are defined by a specific implementation.

- LOCKED:

a) the clock system is locked in the reference clock recovered by the CPRI associated clock, i.e. hardware PLL locked

b) CPRI links line up (or in a state that is considered to indicate that the CPRI links can be operated properly, F-status indication)

note: LOCKED when both a) and b) are satisfied

- UNLOCKED:

a) the clock system isn't locked in the reference clock recovered by the CPRI associated clock, i.e. hardware PLL unlocked

b) CPRI links line up (or in a state that is considered to indicate that the CPRI links can be operated properly, F-status indication)

note: UNLOCKED when either a) or b) occurs. When a) occurs, alarm item is 18 synchronization error; When b) occurs, alarm item is 17 no sync source.

sources

This parameter describes a port selection of a clock source, controlled by the O-RU$_8$.

Refer to o-ran-sync-cpri.yang YANG Model for details.

A3.10 O-RU$_8$ U-plane Configuration

This chapter refers specifically to the U-plane configuration of the O-RU$_8$ based on CPRI Option8 split, i.e., the U-plane configuration of the O-RU$_8$ according to the architectural model chapter.

A3.10.1 Specific Conventions

A3.10.1.1 YANG model

O-RU$_8$ U-plane under CPRI Option8 split is modeled with reference to O-RU$_{7\text{-}2}$ U-plane modeling under the condition of eCPRI Option7-2 split, o-ran-uplane-conf.yang and newly defined o-ran-uplane-conf-option8.yang. Main relations and differences between the two are compared in the following table.

Table Annex3- 13 U-plane comparison of Option7-2 and Option8

eCPRI option7-2	CPRI option8	Main changes
static-low-level-tx-endpoint	static-tx-endpoint	new definition with reference;
static-low-level-rx-endpoint	static-rx-endpoint	new definition with reference;
low-level-tx-endpoint	tx-endpoint	new definition with reference;
low-level-rx-endpoint	rx-endpoint	new definition with reference;
low-level-tx-link	tx-link	new definition with reference;
low-level-rx-link	rx-link	new definition with reference;
endpoint-types	endpoint-types-option8	new definition with reference; original parameters seem be not applicable
endpoint-capacity-sharing-groups	not support	
endpoint-prach-group	not support	
tx-array-carriers	tx-array-carriers	reuse without supporting laa;
rx-array-carriers	rx-array-carriers	reuse
tx-arrays	tx-arrays	reuse

rx-arrays	rx-arrays	reuse
relations	relations	reuse
eaxc-id-group-configuration	not support	
general-config	not support	

A3.10.1.2 Comparison of Specification Requirement

The content of this chapter is newly written and is not relevant to the O-RAN specification.

However, portions of this specification refer to the U-plane configuration under the O-RAN specification eCPRI Option 7-2 split.

A3.10.2 Format of eaxc-id

eaxc-id definition consistent with the O-RAN specification is used, ORAN-WG4.CUS.0-v03.00[2] may be referred for the specific format.

A3.10.3 U-plane Endpoint Addressing

For CPRI protocol-based connections, Option8 is used to split the O-RU_8, and eaxc-id is used for resource mapping and U-plane endpoint addressing.

This method is applicable to end-to-end CPRI Option8 architecture.

This method is also applicable to an architecture converting from eCPRI Option7-2 to CPRI Option8, where O-DU_8 and $FHGW_{7\text{-}2\rightarrow 8}$ are connected based on an eCPRI Option7-2, $FHGW_{7\text{-}2\rightarrow 8}$ and O-RU_8 are connected CPRI Option8.

Different eaxc-ids correspond to different logical functions, such as cell, carrier, and space division dataflow. The end-to-end dataflow from a processing end of the O-DU_8 to an antenna end of the O-RU_8 are identified.

During the configuration procedure:

O-DU_8 configures eaxc-id for the O-RU_8 U-plane endpoint, and also specifies a physical antenna, see tx-endpoint and rx-endpoint in o-ran-upline-conf-option8.yang which realizes mapping between logical functions and O-RU_8 physical resources.

O-DU$_8$ configures a relationship between U-plane endpoints and CPRI dataflow for the O-RU$_8$, which realizes the addressing of CPRI dataflow and O-RU$_8$ U-plane endpoint.

During data transmission, in combination with CPRI U-plane dataflow model of 4.3.1.2:

In a downstream direction, the O-RU$_8$ obtains CPRI dataflow from the CPRI interface, and finds corresponding U-plane transmitting endpoint tx-endpoint according to eaxc-id, and transmits the processed data to corresponding physical antenna.

In an upstream direction, the O-RU$_8$ receives data from the physical antenna, and finds corresponding U-plane receiving endpoint rx-endpoint according to eaxc-id, and transmits the processed data to the CPRI dataflow on corresponding CPRI interface.

The following is rules for configuring the eaxc-id by O-DU$_8$:

- For the same O-RU$_8$, only one eaxc-id is configured for each dataflow in the same direction (e.g., upstream or downstream), and different eaxc-ids are configured for different dataflow. eaxc-ids can be the same for dataflow in different directions (upstream and downstream) for the same O-RU$_8$.

- For the same upstream U-plane data stream,

  - ✓ lower-level-tx-endpoint in FHM$_8$ is configured with the same eaxc-id with the sending endpoint tx-endpoint in O-RU$_8$.

- For the same downstream U-plane data stream.

  - ✓ Low-level-rx-endpoint in the FHM$_8$ with the same eaxc-id configured with the receiving endpoint rx-endpoint in the O-RU$_8$.

- For the same user-facing data stream for the same shared cell.

  - ✓ Sending endpoint tx-endpoint in different O-RU$_8$s configured with the same eaxc-id.
  - ✓ Receive endpoint rx-endpoint in different O-RU$_8$s configured with the same eaxc-id.

- For the same O-RU$_8$, only one eaxc-id is configured for each dataflow in the same direction (e.g., upstream or downstream), and different eaxc-ids are configured for different dataflow. The eaxc-ids can be the same for dataflow in different directions (upstream and downstream) for the same O-RU$_8$.
- For the same upstream U-plane dataflow,
  - ✓ lower-level-tx-endpoint in FHM$_8$ is configured with the same eaxc-id as the tx-endpoint in O-RU$_8$;
- For the same downstream U-plane dataflow,
  - ✓ low-level-rx-endpoint in FHM$_8$ is configured with the same eaxc-id as the rx-endpoint in O-RU$_8$;
- For the same U-plane dataflow for the same shared cell,
  - ✓ tx-endpoints in different O-RU$_8$ are configured with the same eaxc-id;

- ✓ rx-endpoints in different O-RU$_8$ are configured with the same eaxc-id;

O-RU$_8$ shall reject a configuration request that violates the above rules.

A3.10.4 U-plane Configuration

O-RU$_8$ U-plane routing configuration under CPRI Option 8 split is shown below.

For more information, refer to o-ran-uplane-conf-option8.yang.

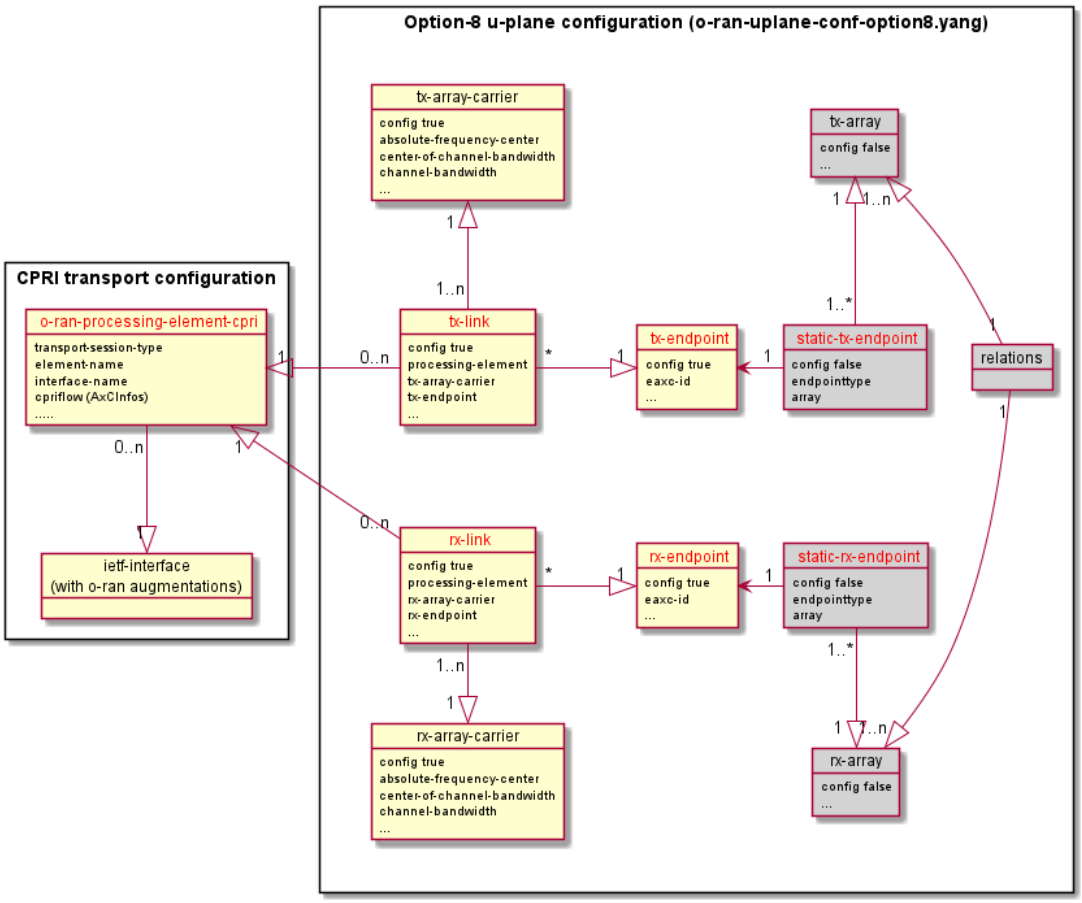


Figure Annex3- 17 O-RU$_8$ U-plane routing configuration under Option8 split

Pre-condition:

M-plane connection between the NETCONF client and the O-RU$_8$ is established.

Post-condition:

U-plane routing configuration is completed.

Basic procedure:

1) NETCONF client gets the intrinsic properties of the O-RU$_8$ by reading the following parameters via RPC edit-config:
- o-ran-uplane-conf-option8.yang
  - ✓ tx-arrays
  - ✓ rx-arrays
  - ✓ static-tx-endpoints list
  - ✓ static-rx-endpoints list
- o-ran-interfaces-cpri.yang
  - ✓ interface element

2) NETCONF client checks relationships between the elements read in step 1):
   - ✓ static-tx-endpoint element and tx-array element;
   - ✓ static-rx-endpoint element and rx-array element;
   - ✓ static-tx-endpoint element and interface element;
   - ✓ static-rx-endpoint element and interface element;
   - ✓ tx-arrays, rx-arrays and element thereof.
3) NETCONF client checks whether the following configurations meet the user cell configuration requirements, e.g., being connected with a specific antenna array and able to support the target service type, etc.
   - ✓ tx-endpoint
   - ✓ rx-endpoint
4) NETCONF client assigns eaxc-id for tx-endpoint/rx-endpoint. The assignment rules see U-plane endpoint addressing, and configures the eaxc-id to a terminal by RPC edit-config.
   - ✓ tx-endpoint
   - ✓ rx-endpoint
5) NETCONF client creates a carrier by RPC edit-config
   - ✓ tx-array-carriers
   - ✓ rx-array-carriers
6) NETCONF client configures the U-Plane interface for the northbound CPRI interface via RPC edit-config
   - ✓ processing-elements
   - ✓ AxC-information
7) NETCONF client establishes U-plane routing for the O-RU$_8$ by creating and configuring the following parameters via RPC edit-config, and thus associating the CPRI northbound interface, the transceiver terminal, and the carrier.
   - ✓ tx-links
   - ✓ rx-links
8) NETCONF client activates the carrier via RPC edit-config
   - ✓ tx-array-carriers.active = ACTIVE
   - ✓ rx-array-carriers.active = ACTIVE

Example of U-plane configuration:

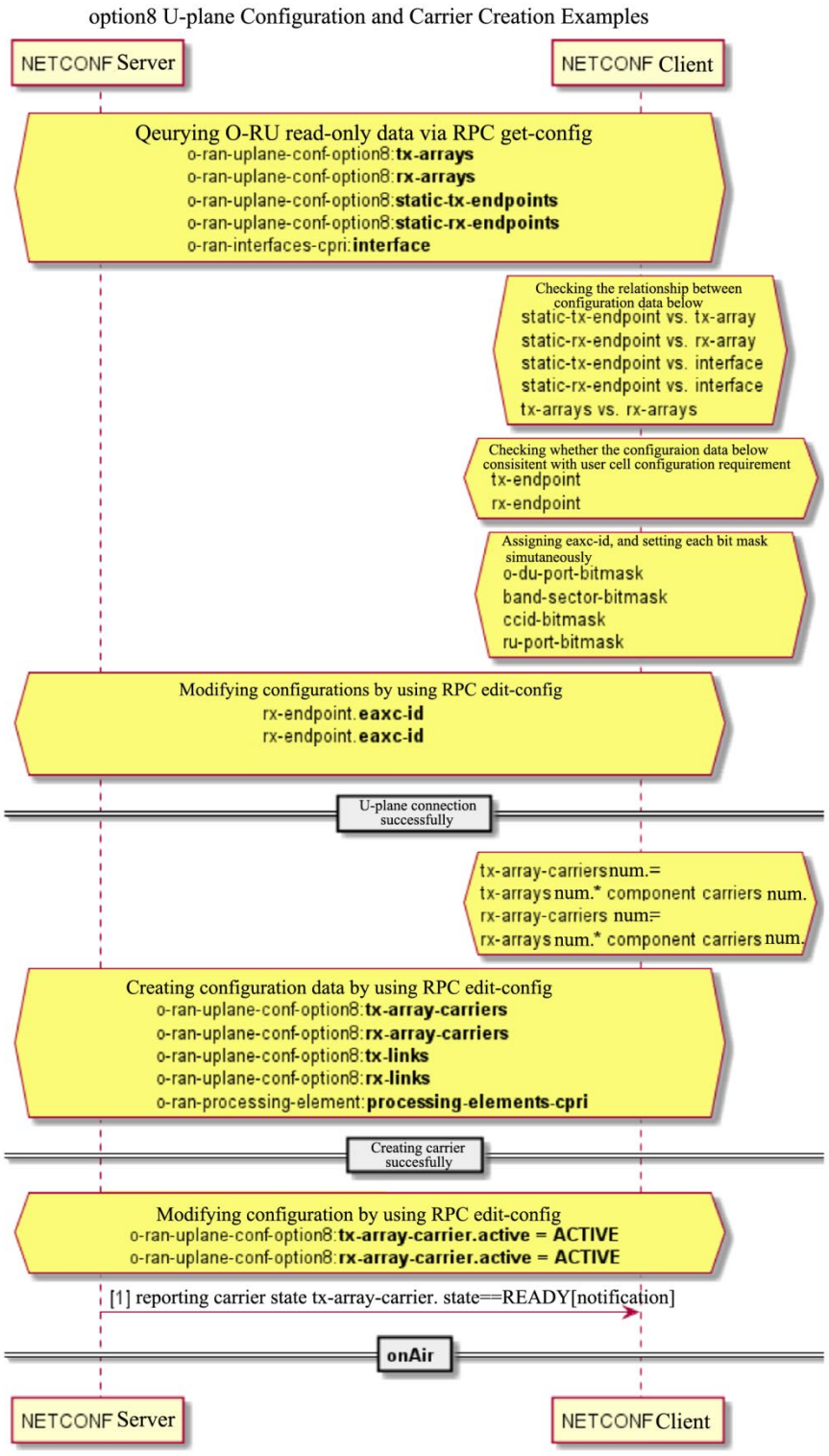


Figure Annex3- 18

A3.10.5 Carrier State Management

The state management of carrier is consistent with that described in the O-RAN specification, unless otherwise specified. The following table describes a comparison of value range supported by this specification and the O-RAN specification.

Table Annex3- 14 Comparison of carrier

	O-RAN Specification	This specification
Value range of carrier active, configurable parameter	INACTIVE （default）	the same as O-RAN specification
	ACTIVE	the same as O-RAN specification
	SLEEP	the same as O-RAN specification
Value range of carrier state, read-only parameter	DISABLED	the same as O-RAN specification
	BUSY	no supported
	READY	the same as O-RAN specification

A3.11 Delay Measurement and Management

A3.11.1 Convention

This specification is direct at indoor distribution scenario, in which current O-RU$_8$s are all at single-level leaf stage, and do not involve cascade or multiple uplink. Therefore, this specification has the following scenario conventions:

O-RU$_8$ cascade, not involved.

O-RU$_8$ multiple uplink, not involved.

O-RU$_8$ multiple uplink and cascade, not involved.

A3.11.2 Delay Model

CPRI protocol based on the Option 8 physical layer functional split is used for a fronthual interface between the FHM_8 and the $O\text{-}RU_8$. The model presented in this section supports both $O\text{-}RU_8$ in indoor distribution that docks the FHM_8, and $O\text{-}RU_8$ in outdoor macro station in the future.

The CPRI delay management model is holistic in nature and often needs to be described from DU south-port to air port for complete representation. FIG. 4.9-1 lists the delay management model given in CPRI V7.0 Section 4.2.9.

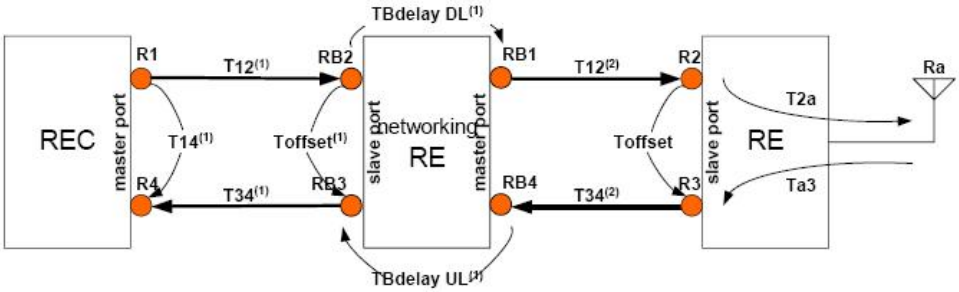


Figure Annex3- 19 CPRI delay management model (cascade)

CPRI start point of the system in which the $O\text{-}RU_8$ located may be $O\text{-}DU_8$ (corresponding) that provides CPRI interface, or it may be CPRI interface FHM_8 for converting from 7-2x to 8, so the traditional REC in the CPRI model needs to be generalized to the CPRI Master to achieve an independent and complete description of the $O\text{-}RU_8$ model, as shown in FIG. 4.9-2 for details.

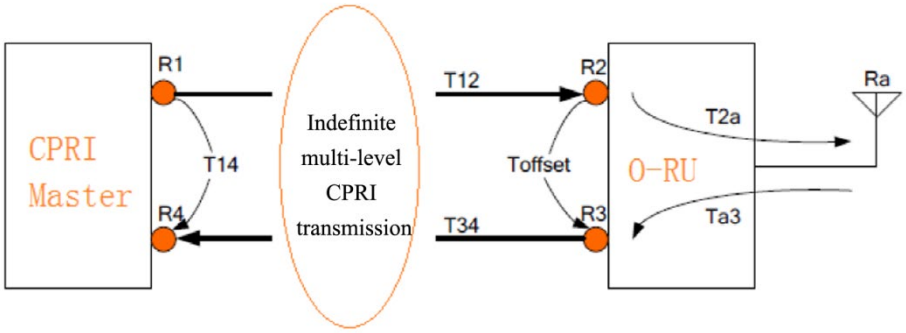


Figure Annex3- 20 Relationship between $O\text{-}RU_8$ and CPRI Master

The elliptical region in the figure may be a cable, an FHM_8 of a primary CPRI interface, a multi-level CPRI FHM_8 cascade or an indefinite multi-level CPRI $O\text{-}RU_8$ and so on. In either above cases, the delay relationship between a given $O\text{-}RU_8$ and the CPRI Master is limited to the four elements, R1, R4, T12, and T34, which specifically are:

point R1: relationship between a CPRI frame header and absolute time; relationship between IQ data and CPRI frame header;

point R4: no need particular attention.

T12: TX's (generalized) transmission delay.

T34: RX's (generalized) transmission latency.

In order to further enhance the expressiveness of the model, it may be useful to define:

Offset of the CPRI frame header of the point R1 relative to absolute time: **TR1_Offset** in Tc, which is used as a descriptive reference for time of the CPRI system. Each R1 point has one. Optionally, this time reference can be conventionally specified as absolute time by default when it is not required.

Relative position of the IQ data at point R1 and the CPRI frame header: Dl_Offset in Tc with reference to IR protocol, which describes time offset relationship between the IQ data and the CPRI frame.

A3.11.2.1 Measurement Model

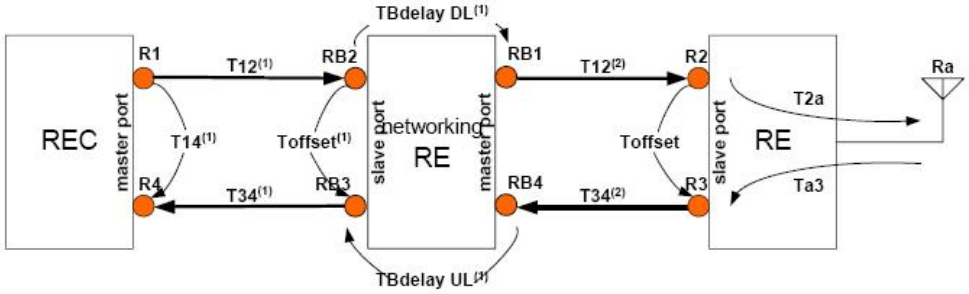


Figure Annex3- 21 Delay measurement after the O-RU$_8$ M-plane channel is normally established:

after receiving a delay measurement request message, the O-RU$_8$ reports its own delay measurements to the BBU via a delay measurement response message, which should include the following delays:

- Toffset
- T2a
- Ta3
- TBdelay DL
- TBdelay UL

N(unit:basic frame, the difference of the frame position at RB3 relative to RB4

N is a specific concept of cascade RU, which is directly 0 in non-cascade mode.

A3.11.2.2 Timing model

Independence of the O-RU$_8$ model can be improved by detaching time (in frame number form) from delay configuration model. Based on TR1_Offset and T12, O-RU$_8$ can obtain global time in frame number form.

A3.11.2.3 Dataflow delay configuration model

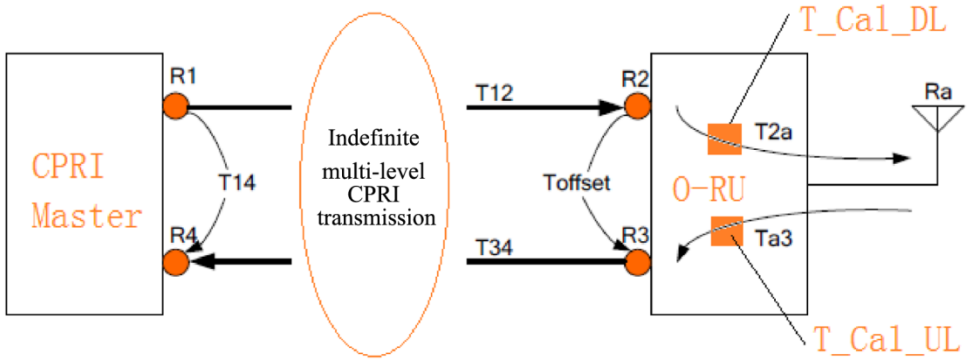


Figure Annex3- 22 Schematic diagram of O-RU$_8$ side delay configuration compensation point

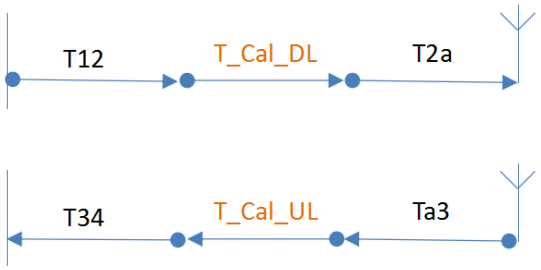


Figure Annex3- 23 Schematic diagram of O-RU$_8$ side delay configuration

Under this model, the O-RU$_8$ can take on all of delay compensation requirements independently, or it can take on part of the delay configuration requirements under the control of the CPRI Master.

A3.11.3 Delay Parameter Modeling

Table Annex3- 15 O-RU$_8$ delay parameters

Type of delay parameter	Name of delay parameter	**Attribution level modeling of parameter**	Description
RU delay attribution parameter, reported by RU, DU read-	Toffset	RU level parameter	one value per RU, in unit of ns
	T2a	carrier specification parameter	one parameter per carrier specification, the carrier specification includes: 1. carrier format NR/LTE

only			2. spectrum bandwidth 3. subcarrier spacing the same value is still used when multiple CPRI uplink ports of RU are involved in unit of ns
	Ta3	the same as above	the same as above
	TBdelayDl	RU level parameter	same value when multipath CPRI cascades are involved (very unlikely to exist) in unit of ns
	TBdelayUl	the same as above	the same as above
	T_Cal_DL_MAX	carrier specification parameter	the parameter belongs to a digital domain and is not affected by the frequency band, with reference to T2a, Ta3 modeling in unit of ns
	T_Cal_UL_MAX	the same as above	the same as above
	N	Each pair of cascade ports	in unit of Tc (basic frame period); pass-by data introduced by cascade, frame header alignment compensation.
RU delay configuration parameter, DU readable	T_Cal_DL	carrier antenna level parameter	one value per carrier antenna, the carrier antenna includes: 1. carrier 2. physical antenna For example, a 4T4R device single-carrier

and writable			has four T_Cal_DL configuration points; a 4T4R device dual-carrier has eight T_Cal_DL configuration points in unit of ns
	T_Cal_UL	the same as above	the same as above
	T12	RU.OptPort level parameter	CPRI T12, in unit of ns
	T34	RU.OptPort level parameter	CPRI T34，in unit of ns
	DL_Offset	carrier antenna level parameter	Offset relationship between IQ data and CPRI frame header, in unit of Tc
	TR1_Offset	RU.OptPort level parameter	Offset of frame number of point R1 relative to absolute time, in unit of Tc. Optional parameter.

A3.11.3.1 Scenario involved TR1_Offset

In the model given in FIG. 4.9.2.3-1, the frame number at the start point R1 of the CPRI reference time, includes two cases.

Case 1: FDD

In LTE FDD and similar systems that do not require alignment of 1PPS to work, the CPRI frame number at CPRI master exit is not required.

Case 2: TDD

Under LTE TDD, NR systems, gap frame header has an exact relationship to 1PPS. The CPRI frame number at the CPRI master exit is aligned to 1PPS by default.

The CPRI frame number at the CPRI master exit may not be aligned to 1PPS in Case 2 under certain usages. Defining parameter TR1_Offset in this case helps to supplement the expression

capability of delay configuration system, and RU can still recover the CPRI frame number aligned with 1PPS by using TR1_Offset, t12 in this case.

A3.11.3.2 Parameter grouping

Parameters to be measured at this level:

Toffset, one value for each RU, regardless of port.

T2a, Ta3, T_Cal_DL_MAX, T_Cal_UL_MAX, related to format, carrier bandwidth, subcarrier spacing

Parameters required to be measured for cascade (this specification does not refer to RU cascade, TBD):

TBdelayDl, TBdelayUl, N

CPRI link-related configuration parameters:

T12, T34, TR1_Offset

IQ data flow-related configuration parameters:

T_Cal_DL, DL_Offset, T_Cal_UL

A3.11.3.3 Modeling after grouping

yang file: o-ran-delay-management-ru-cpri.yang

Measurement parameter required for this level: ru-delay-profile

Measurement parameter required for cascade: ru-delay-profile-cascade-related

CPRI link related configuration parameter: ru-delay-config-cpri-link-related

IQ dataflow related configuration parameter: ru-delay-config-data-flow-related

A3.11.3.3.1 Modeling description of ru-delay-profile/ru-delay-profile-list-radio-related

The reference factors for O-RAN's existing delay management modeling are carrier bandwidth and subcarrier spacing, and neither format, nor frequency point is included.

In CPRI class device practice, CPRI RU delay is mainly influenced by the format, carrier, and subcarrier spacing, with the format having the greatest influence and the latter two having relatively little influence. Factors such as frequency point have a small impact and are usually self-digested within the device without exposing the differences externally.

Therefore, this part of the parameter is modeled as follows:

Format Bandwidth Subcarrier spacing T2a Ta3 T_Cal_DL_MAX T_Cal_UL_MAX

The three underlined elements are used as table lookup keys to uniquely specify a set of relevant experimental parameters within the current RU. In terms of value, formats such as COMMON, LTE, NR, etc. are defined; frequency is expressed in kHz; subcarrier spacing is expressed in Hz. The format COMMON means wildcard, and frequency and subcarrier interval of 0 means wildcard.

The model has the following usage:

case a: O-RU$_8$ only have one set of delay paras,.

COMMON 0 0 t2a ta3 ...

case b: LTE use one set of delay paras, NR use another set of paras.

COMMON 0 0 t2a ta3 ...

case c: LTE NR use 2 set of delay paras...

LTE 5000 15000 ...

LTE 10000 15000 ...

NR 100000 30000 ...

...

Recommended usage:

Table Annex3- 16

LTE single mode single-carrier	COMMON 0 0 t2a ta3 T_Cal_DL_MAX T_Cal_UL_MAX
LTE single mode multi-carrier	COMMON 0 0 t2a ta3 T_Cal_DL_MAX T_Cal_UL_MAX
NR single mode single-carrier	COMMON 0 0 t2a ta3 T_Cal_DL_MAX T_Cal_UL_MAX
NR single mode multi-carrier	COMMON 0 0 t2a ta3 T_Cal_DL_MAX T_Cal_UL_MAX
LTE/NR bimode	LTE 0 0 t2a ta3 T_Cal_DL_MAX T_Cal_UL_MAX NR 0 0 t2a ta3 T_Cal_DL_MAX T_Cal_UL_MAX

A3.11.3.3.2 Modeling description of ru-delay-config-data-flow-related

Modeling perspective:

The CPRI RU delay configuration tends to be passive, i.e., by means of DU measuring, calculating, and sending pending configurations to RU.

Modeling approach:

In the RU perspective, a configuration model can be built based on its configuration points. However, this approach is not easy to use externally because the configuration points within the RU may be complex in multi-carrier and multi-format scenarios.

Mode selection:

From an external point of view, link between the DU/RU IQ dataflow is the eaxc-id. “Information pair” of eaxc-id & delay is used as the basic element herein. On the one hand, DU is easy to express, and on the other hand, RU is easy to use.

In detail, because the same eaxc-id or different eaxc-ids may be used for upstream and downstream of the same carrier antenna, it is not convenient to describe both DL and UL under one eaxc-id. For this reason, DL and UL are modeled separately:

dl_cal_ru_list：eaxc-id and t_cal_dl, DL_Offset

ul_cal_ru_list：eaxc-id and t_cal_ul

A3.11.4 Delay Management Procedure

The following figure shows a delay management procedure.

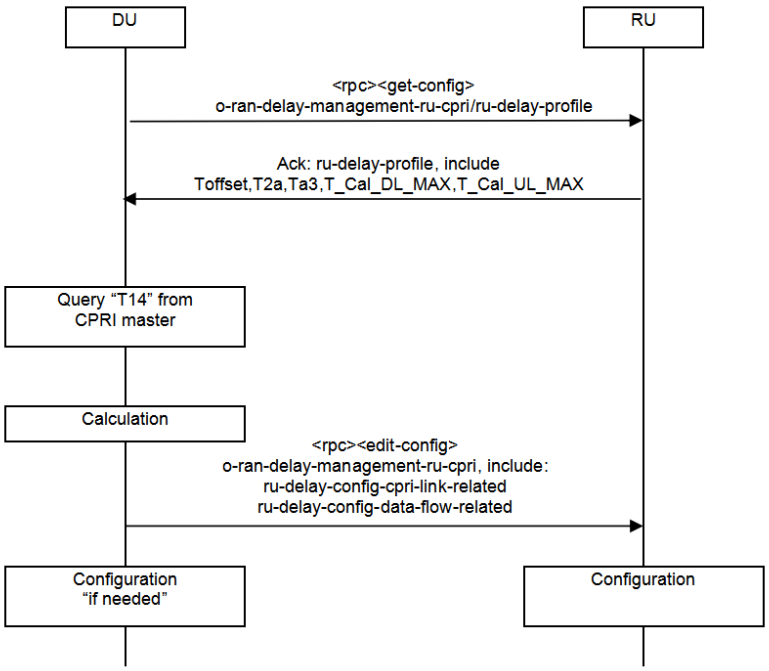


Figure Annex3- 24 Delay management procedure

A3.12Log Management

A3.12.1 Specific Conventions

A3.12.1.1 YANG model

o-ran-troubleshooting.yang: defining O-RU$_8$'s fault location log

o-ran-trace.yang: defining O-RU$_8$'s tracking log

A3.12.1.2 Comparison of specification requirements

Comparison of the differences between this chapter and the O-RAN Management Plane Specification [1] is shown in the following table:

Table Annex3- 17 Comparison of differences of log management related specification

	O-RAN Management Plane Specification	Requirements of the present specification
Compression format of Log file	• .gz (DEFLATE) • .lz4 (LZ4)	gz

	• .xz (LZMA2 - xz utils) • .zip (DEFLATE - zlib library)	
Log file size	not defined (recommending that the upload time be controlled within 3 minutes)	the same as O-RAN specification upper limit for log files is defined by vender, file can be renamed and dated

A3.12.2 Fault Location Log

A log used to locate a fault after system startup. It is recommended that the logs be graded and that only errors and logs that affect the operation of the system be introduced to default location log.

A3.12.2.1 Overview of YANG model

o-ran-troubleshooting.yang: defining fault location log of O-RU[8]

namespace: urn:o-ran:troubleshooting:1.0

RPC-reply:

Table Annex3- 18

Parameter name	Parameter definition	Parameter type	Parameter value	Importance
status	RPC post-condition	enumeration	SUCCESS FAILURE	mandatory
failure-reason	reason for failure (coming into effect when status takes a value of FAILURE)	string	“ ”	mandatory

RPC:

<start-troubleshooting-logs> starts to create troubleshooting Log file, logging time starts from system startup to the time when this RPC is received

<stop-troubleshooting-logs> stops creating troubleshooting Log file, and ignores this RPC after sending out Notification (since the file has already been created)

Notification:

<troubleshooting-logs-generated> notifies that O-RU$_8$ controller that the troubleshooting Log file has been created, and carries Log filename.

A3.12.2.2 Procedure requirements

After the O-RU$_8$ software system starts, it automatically starts to collect Logs and store them in the memory, and writes them to the file when it receives creating file command.

(1) Triggering the O-RU$_8$ with <start-troubleshooting-logs> to create a troubleshooting logs file, and the O-RU$_8$ replies rpc-reply to indicate whether the operation is successful or not.

(2) If rpc-reply shows the operation is success, the O-RU$_8$ starts to create fault location log file, and sends <troubleshooting-logs-generated> after creating all Log files, to notify the O-RU$_8$ that controller file has been created and is waiting to be collected, with all the URLs of the log files carried therein. See File Management for file collection method.

(3) The O-RU$_8$ controller may send <stop-troubleshooting-logs> to stop the log file creation procedure if the rpc-reply indicates fail or if it is a long time before receiving the <troubleshooting-logs-generated> notification.

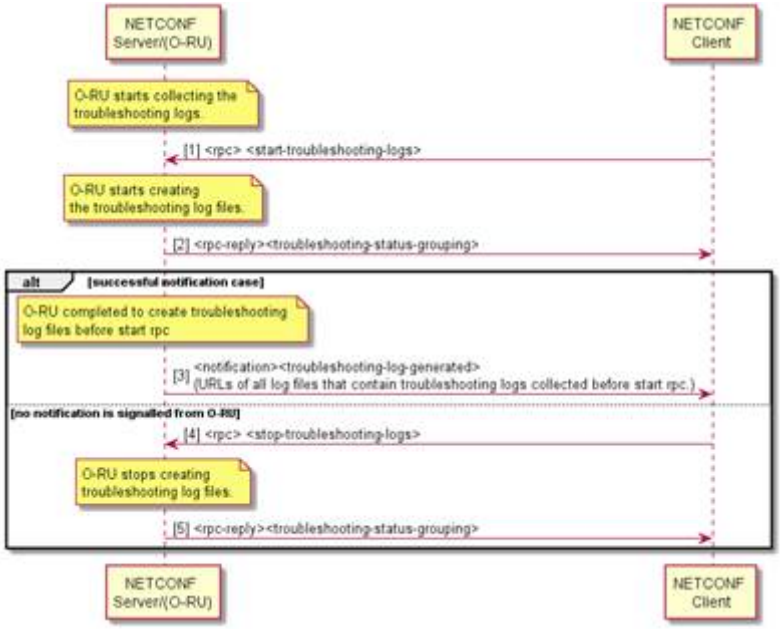


Figure Annex3- 25 Fault location log procedure

A3.12.3 Trace Log

Trace log is used to record a complete log of the system over a period of time and is controlled by the O-RU$_8$ Controller.

A3.12.3.1 YANG model overview

o-ran-trace.yang: defining Trace Log of O-RU$_8$.

namespace: urn:o-ran:trace:1.0

RPC-reply:

Table Annex3- 19

Parameter name	Parameter definition	Parameter type	Parameter value	Importance
status	RPC post-condition	enumeration	SUCCESS FAILURE	mandatory
failure-reason	reason for failure (coming into effect when status takes a value of FAILURE)	string	“ ”	mandatory

RPC:

<start-trace-logs> starts to create trace log file, logging time starts from receiving RPC to receiving <stop-trace-logs>

<stop-trace-logs> stops creating the trace log file

Notification:

<trace-log-generated> notifies that O-RU$_8$ controller trace log file has been created, and carries Log filename.

A3.12.3.2 Procedure requirements

(1) The O-RU$_8$ is triggered with <start-trace-logs> to start collecting trace logs, and the O-RU$_8$ replies rpc-reply to indicate whether the operation is successful.

(2) The O-RU$_8$ starts to create log files in a loop and issues a <trace-log-generated> notification, and the URL includes all created Log files.

(3) When the O-RU$_8$ controller needs to stop collecting the trace logs, it issues <stop-trace-logs>. The O-RU$_8$ immediately stops collecting the trace logs after receiving the <stop-trace-logs>, and indicates in rpc-reply that is-notification-last is TRUE.

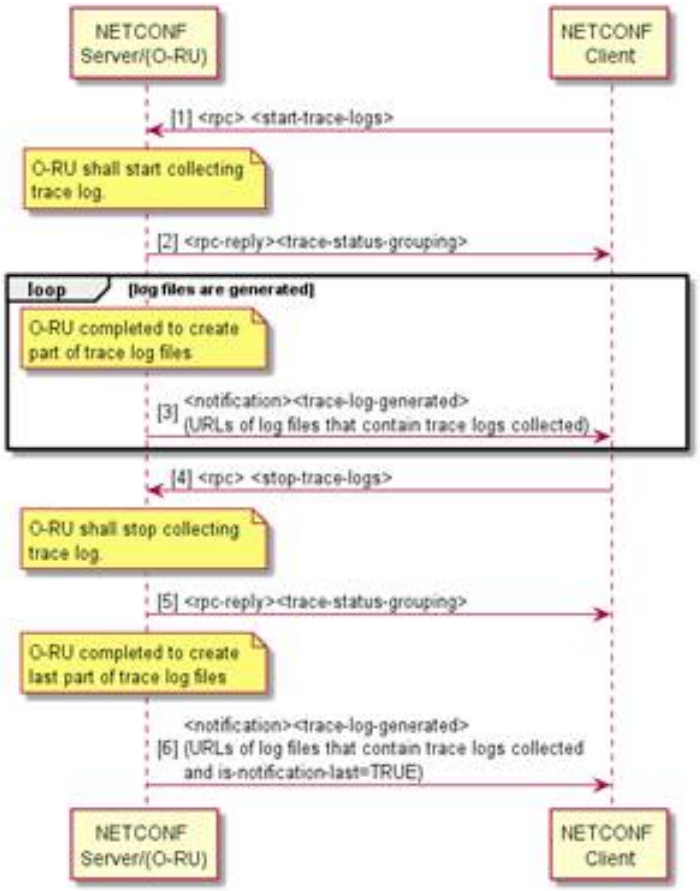


Figure Annex3- 26 Procedure of tracing log

A3.13 Startup procedure

This subsection describes a typical startup process for Option8 O-RU$_8$, with following main differences from 7-2x:

1. CPRI L1 and L2 connections may be established in two main ways, default matching and self-adaption. The Option8 O-RU$_8$ supports line rates of 10137.6Mbps and 24330.24Mbps in the currently discussed context. The currently selected deployment mode is that the O-RU$_8$ and its superior CPRI device are matched by default (optical or electrical aspect, rate and error correcting code option, etc.) and are directly connectable, i.e., no CPRI link self-adaptive function is introduced in the current period. The self-adaptive function will be discussed later.

2. CPRI clock does not require additional configurations for locking, but only needs to be monitored by alarms. Therefore, there is no clock synchronization attempt or configuration process in the startup procedure.

3. Considering the convenience of device management in configuration scenarios without carrier, the alarm management is placed in front of U-plane configuration.

4. Topology discovery is a behavior of CPRI Master, and O-RU$_8$ cascade scenario is not considered in the current period, it is not necessary to pay attention to it in the O-RU$_8$ context.

A3.13.1 Examples of O-RU$_8$ startup procedure

Table Annex3- 20 Option8 O-RU$_8$ startup procedure

Startup procedure	Description	Reference chapter
Initialization of the O-RU$_8$ northbound CPRI port	The CPRI link synchronization involves: rate (matching to determine the encoding format), error correcting code, and P-value selection. Once the above parameters are determined, the O-RU$_8$ can perform data interaction with the outside world at the MAC layer if the physical links and the upstream devices are normal. The following mode may be chose herein: Mode 1: O-RU$_8$ matches by default with the upstream device in options such as rate and error correcting code, which is directly connectable. Mode 2: By introducing self-adaption, and the O-RU$_8$ interacts with the upstream device, and each matches with an appropriate rate and error correcting code options according to a predetermined adaptation action; the P-value is extracted from the CPRI control word. Note: O-RU$_8$ provides a globally unique MAC address for each CPRI uplink port	none

Initialization of the M-plane network layer connection	By DHCP, O-RU$_8$ obtains its own M-plane IP address as the basis for external interaction	Chapter 4.2.3
M-plane Call Home	O-RU$_8$ initiates NETCONF Call Home to O-DU$_8$	Chapter 4.2.4.1
establishing SSH connection	O-DU$_8$ establish SSH connections with O-RU$_8$; performing NETCONF certification	Chapter 4.2.4
NETCONF capacity discovery	performing NETCONF capacity discovery by O-RU$_8$ and O-DU$_8$	Chapter 4.2.5
NETCONF connection monitoring	performing NETCONF connection monitoring by O-RU$_8$ and O-DU$_8$	Chapter 4.2.5
Read O-RU$_8$ parameter	obtaining O-RU$_8$ parameter	
O-RU$_8$ software management	O-DU$_8$ verifies and manages the O-RU$_8$ software, ensures that the O-RU$_8$ software is as expected	Chapter 4.4
O-RU$_8$ fault management	Initializing O-RU$_8$ fault management to achieve state control of the O-RU$_8$. Note: Fault management is placed in front of the U-plane configuration so that the fault management of O-RU$_8$ can be performed normally even when there is no cell configuration.	Chapter 4.5
O-RU$_8$ U-plane management	Carrier antenna	Chapter 4.8.4
Delay configuration	Delay configuration	Chapter 4.9.4
Carrier status management	Carrier status management	Chapter 4.8.5
Fault management supplement	Supplementing corresponding fault management arrangement after the carrier being activated	Chapter 4.5

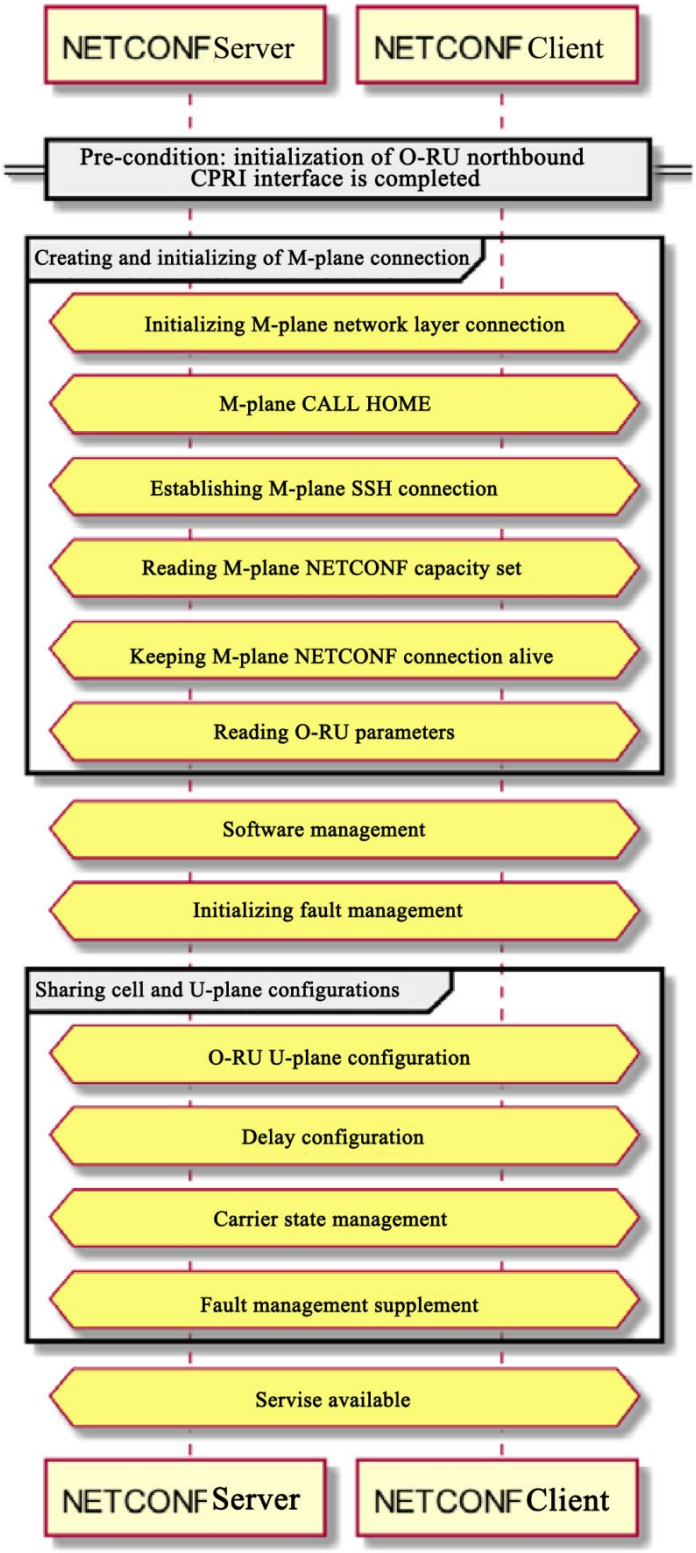


Figure Annex3- 27 Examples of O-RU8 startup procedure

A3.14 Appendix - YANG Model Tree Diagram

This appendix lists YANG model tree diagrams that are newly defined or modified by this specification; refer to the O-RAN specification for other YANG models.

A3.14.1 A.1 o-ran-uplane-conf-option8 Module

The format for the uplane conf option8 module is provided below

```
module: o-ran-uplane-conf-option8
 +--rw user-plane-configuration
   +--rw tx-links* [name]
   | +--rw name                string
   | +--rw processing-element    -> /o-ran-pe:processing-elements/ru-elements/name
   | +--rw tx-array-carrier      -> /user-plane-configuration/tx-array-carriers/name
   | +--rw tx-endpoint           -> /user-plane-configuration/tx-endpoints/name
   +--rw rx-links* [name]
   | +--rw name                string
   | +--rw processing-element    -> /o-ran-pe:processing-elements/ru-elements/name
   | +--rw rx-array-carrier      -> /user-plane-configuration/rx-array-carriers/name
   | +--rw rx-endpoint           -> /user-plane-configuration/rx-endpoints/name
   +--ro endpoint-types-option8* [id]
   | +--ro id    uint16
   +--ro static-tx-endpoints* [name]
   | +--ro name                  string
   | +--ro restricted-interfaces*   -> /if:interfaces/interface/name
   | +--ro array                 -> /user-plane-configuration/tx-arrays/name
   | +--ro endpoint-type?          -> ../../endpoint-types-option8/id
   +--ro static-rx-endpoints* [name]
   | +--ro name                  string
```

```
| +--ro restricted-interfaces*   -> /if:interfaces/interface/name
| +--ro array                    -> /user-plane-configuration/rx-arrays/name
| +--ro endpoint-type?           -> ../../endpoint-types-option8/id
+--rw tx-endpoints* [name]
| +--rw name          -> /user-plane-configuration/static-tx-endpoints/name
| +--rw compression!
| +--rw e-axcid
|   +--rw o-du-port-bitmask     uint16
|   +--rw band-sector-bitmask   uint16
|   +--rw ccid-bitmask          uint16
|   +--rw ru-port-bitmask       uint16
|   +--rw eaxc-id               uint16
+--rw rx-endpoints* [name]
| +--rw name                        -> /user-plane-configuration/static-rx-endpoints/name
| +--rw compression
| +--rw e-axcid
| | +--rw o-du-port-bitmask     uint16
| | +--rw band-sector-bitmask   uint16
| | +--rw ccid-bitmask          uint16
| | +--rw ru-port-bitmask       uint16
| | +--rw eaxc-id               uint16
| +--rw non-time-managed-delay-enabled?   boolean
+--rw tx-array-carriers* [name]
| +--rw name                        string
| +--rw absolute-frequency-center     uint32
| +--rw center-of-channel-bandwidth   uint64
| +--rw channel-bandwidth             uint64
```

```
| +--rw active?                    enumeration
| +--ro state                      enumeration
| +--rw type?                      enumeration
| +--ro duplex-scheme?             enumeration
| +--rw rw-duplex-scheme?          -> /user-plane-configuration/tx-array-carriers[name=current()/../name]/duplex-scheme
| +--rw rw-type?                   -> /user-plane-configuration/tx-array-carriers[name=current()/../name]/type
| +--rw band-number?               -> /mcap:module-capability/band-capabilities/band-number {mcap:LAA}?
| +--rw lte-tdd-frame
| | +--rw subframe-assignment      enumeration
| | +--rw special-subframe-pattern enumeration
| +--rw gain                       decimal64
| +--rw downlink-radio-frame-offset  uint32
| +--rw downlink-sfn-offset        int16
+--rw rx-array-carriers* [name]
| +--rw name                       string
| +--rw absolute-frequency-center  uint32
| +--rw center-of-channel-bandwidth  uint64
| +--rw channel-bandwidth          uint64
| +--rw active?                    enumeration
| +--ro state                      enumeration
| +--rw type?                      enumeration
| +--ro duplex-scheme?             enumeration
| +--rw downlink-radio-frame-offset  uint32
| +--rw downlink-sfn-offset        int16
| +--rw gain-correction            decimal64
| +--rw n-ta-offset                uint32
+--ro tx-arrays* [name]
```

```
| +--ro name                              string
| +--ro number-of-rows                    uint16
| +--ro number-of-columns                 uint16
| +--ro number-of-array-layers            uint8
| +--ro horizontal-spacing?               decimal64
| +--ro vertical-spacing?                 decimal64
| +--ro normal-vector-direction
| | +--ro azimuth-angle?   decimal64
| | +--ro zenith-angle?    decimal64
| +--ro leftmost-bottom-array-element-position
| | +--ro x?   decimal64
| | +--ro y?   decimal64
| | +--ro z?   decimal64
| +--ro polarisations* [p]
| | +--ro p              uint8
| | +--ro polarisation    polarisation_type
| +--ro band-number                       -> /mcap:module-capability/band-capabilities/band-number
| +--ro max-gain                          decimal64
| +--ro independent-power-budget          boolean
| +--ro capabilities* []
|   +--ro max-supported-frequency-dl?   uint64
|   +--ro min-supported-frequency-dl?   uint64
|   +--ro max-supported-bandwidth-dl?   uint64
|   +--ro max-num-carriers-dl?          uint32
|   +--ro max-carrier-bandwidth-dl?     uint64
|   +--ro min-carrier-bandwidth-dl?     uint64
|   +--ro supported-technology-dl*      enumeration
```

```
+--ro rx-arrays* [name]
| +--ro name                                string
| +--ro number-of-rows                      uint16
| +--ro number-of-columns                   uint16
| +--ro number-of-array-layers              uint8
| +--ro horizontal-spacing?                 decimal64
| +--ro vertical-spacing?                   decimal64
| +--ro normal-vector-direction
| | +--ro azimuth-angle?   decimal64
| | +--ro zenith-angle?    decimal64
| +--ro leftmost-bottom-array-element-position
| | +--ro x?   decimal64
| | +--ro y?   decimal64
| | +--ro z?   decimal64
| +--ro polarisations* [p]
| | +--ro p              uint8
| | +--ro polarisation   polarisation_type
| +--ro band-number                         -> /mcap:module-capability/band-capabilities/band-number
| +--ro gain-correction-range
| | +--ro max    decimal64
| | +--ro min    decimal64
| +--ro capabilities* []
|   +--ro max-supported-frequency-ul?   uint64
|   +--ro min-supported-frequency-ul?   uint64
|   +--ro max-supported-bandwidth-ul?   uint64
|   +--ro max-num-carriers-ul?          uint32
|   +--ro max-carrier-bandwidth-ul?     uint64
```

```
|    +--ro min-carrier-bandwidth-ul?     uint64
|    +--ro supported-technology-ul*      enumeration
+--ro relations* [entity]
| +--ro entity    uint16
| +--ro array1
| | +--ro (antenna-type)?
| |    +--:(tx)
| |    | +--ro tx-array-name?   -> /o-ran-uplane-conf:user-plane-configuration/tx-arrays/name
| |    +--:(rx)
| |      +--ro rx-array-name?   -> /o-ran-uplane-conf:user-plane-configuration/rx-arrays/name
| +--ro array2
| | +--ro (antenna-type)?
| |    +--:(tx)
| |    | +--ro tx-array-name?   -> /o-ran-uplane-conf:user-plane-configuration/tx-arrays/name
| |    +--:(rx)
| |      +--ro rx-array-name?   -> /o-ran-uplane-conf:user-plane-configuration/rx-arrays/name
| +--ro types* [relation-type]
|    +--ro relation-type    enumeration
|    +--ro pairs* [element-array1]
|      +--ro element-array1    uint16
|      +--ro element-array2?   uint16
+--rw eaxc-id-group-configuration {mcap:EAXC-ID-GROUP-SUPPORTED}?
   +--rw max-num-tx-eaxc-id-groups?      -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-capabilities/max-num-tx-eaxc-id-groups
   +--rw  max-num-tx-eaxc-ids-per-group?      ->  /mcap:module-capability/ru-capabilities/eaxcid-grouping-capabilities/max-num-tx-eaxc-ids-per-
group
   +--rw max-num-rx-eaxc-id-groups?      -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-capabilities/max-num-rx-eaxc-id-groups
   +--rw  max-num-rx-eaxc-ids-per-group?      ->  /mcap:module-capability/ru-capabilities/eaxcid-grouping-capabilities/max-num-rx-eaxc-ids-per-
group
```

```
      +--rw tx-eaxc-id-group* [representative-tx-eaxc-id]
      | +--rw representative-tx-eaxc-id    uint16
      | +--rw member-tx-eaxc-id*           uint16
      +--rw rx-eaxc-id-group* [representative-rx-eaxc-id]
        +--rw representative-rx-eaxc-id    uint16
        +--rw member-rx-eaxc-id*           uint16


notifications:
 +---n tx-array-carriers-state-change
 | +--ro tx-array-carriers* [name]
 |    +--ro name     -> /user-plane-configuration/tx-array-carriers/name
 |    +--ro state?   -> /user-plane-configuration/tx-array-carriers/state
 +---n rx-array-carriers-state-change
    +--ro rx-array-carriers* [name]
       +--ro name     -> /user-plane-configuration/rx-array-carriers/name
       +--ro state?   -> /user-plane-configuration/rx-array-carriers/state
```

A3.14.2 A.2. o-ran-uplane-conf-hub Module

```
module: o-ran-uplane-conf-hub
 +--rw eaxc-id-group-configuration {mcap:EAXC-ID-GROUP-SUPPORTED}?
 | +--rw max-num-tx-eaxc-id-groups?      -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-capabilities/max-num-tx-eaxc-id-groups
 | +--rw max-num-tx-eaxc-ids-per-group?  -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-capabilities/max-num-tx-eaxc-ids-per-group
 | +--rw max-num-rx-eaxc-id-groups?      -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-capabilities/max-num-rx-eaxc-id-groups
 | +--rw max-num-rx-eaxc-ids-per-group?  -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-capabilities/max-num-rx-eaxc-ids-per-group
 | +--rw tx-eaxc-id-group* [representative-tx-eaxc-id]
 | | +--rw representative-tx-eaxc-id    uint16
```

```
| | +--rw member-tx-eaxc-id*          uint16
| +--rw rx-eaxc-id-group* [representative-rx-eaxc-id]
|    +--rw representative-rx-eaxc-id    uint16
|    +--rw member-rx-eaxc-id*          uint16
+--rw user-plane-configuration
  +--rw low-level-tx-links* [name]
  | +--rw name                    string
  | +--rw processing-element      -> /o-ran-pe:processing-elements/ru-elements/name
  | +--rw low-level-tx-endpoint   -> /user-plane-configuration/low-level-tx-endpoints/name
  +--rw low-level-rx-links* [name]
  | +--rw name                    string
  | +--rw processing-element      -> /o-ran-pe:processing-elements/ru-elements/name
  | +--rw low-level-rx-endpoint   -> /user-plane-configuration/low-level-rx-endpoints/name
  | +--rw user-plane-uplink-marking?   -> /o-ran-pe:processing-elements/enhanced-uplane-mapping/uplane-mapping/up-marking-name
  +--ro endpoint-types* [id]
  | +--ro id                      uint16
  | +--ro supported-section-types* [section-type]
  | | +--ro section-type          uint8
  | | +--ro supported-section-extensions*   uint8
  | +--ro supported-frame-structures*       uint8
  | +--ro managed-delay-support?            enumeration
  | +--ro multiple-numerology-supported?    boolean
  | +--ro max-numerology-change-duration?   uint16
  | +--ro max-control-sections-per-data-section?   uint8
  | +--ro max-sections-per-symbol?          uint16
  | +--ro max-sections-per-slot?            uint16
  | +--ro max-remasks-per-section-id?       uint8
```

```
| +--ro max-beams-per-symbol?                  uint16
| +--ro max-beams-per-slot?                    uint16
| +--ro max-prb-per-symbol?                    uint16
| +--ro prb-capacity-allocation-granularity*   uint16
| +--ro max-numerologies-per-symbol?           uint16
+--ro endpoint-capacity-sharing-groups* [id]
| +--ro id                                     uint16
| +--ro max-control-sections-per-data-section? uint8
| +--ro max-sections-per-symbol?               uint16
| +--ro max-sections-per-slot?                 uint16
| +--ro max-remasks-per-section-id?            uint8
| +--ro max-beams-per-symbol?                  uint16
| +--ro max-beams-per-slot?                    uint16
| +--ro max-prb-per-symbol?                    uint16
| +--ro max-numerologies-per-symbol?           uint16
| +--ro max-endpoints?                         uint16
| +--ro max-managed-delay-endpoints?           uint16
| +--ro max-non-managed-delay-endpoints?       uint16
+--ro endpoint-prach-group* [id]
| +--ro id                                     uint16
| +--ro supported-prach-preamble-formats*      prach-preamble-format
+--ro static-low-level-tx-endpoints* [name]
| +--ro name                                   string
| +--ro restricted-interfaces*                 -> /if:interfaces/interface/name
| +--ro endpoint-type?                         -> ../../endpoint-types/id
| +--ro capacity-sharing-groups*               -> ../../endpoint-capacity-sharing-groups/id
+--ro static-low-level-rx-endpoints* [name]
```

```
| +--ro name                    string
| +--ro restricted-interfaces*     -> /if:interfaces/interface/name
| +--ro endpoint-type?             -> ../../endpoint-types/id
| +--ro capacity-sharing-groups*   -> ../../endpoint-capacity-sharing-groups/id
| +--ro prach-group?               -> ../../endpoint-prach-group/id
+--rw low-level-tx-endpoints* [name]
| +--rw name                          -> /user-plane-configuration/static-low-level-tx-endpoints/name
| +--rw compression!
| | +--rw iq-bitwidth?                uint8
| | +--rw compression-type            enumeration
| | x--rw bitwidth?                   uint8
| | +--rw (compression-format)?
| |    +--:(no-compresison)
| |    +--:(block-floating-point)
| |    | +--rw exponent?              uint8
| |    +--:(block-floating-point-selective-re-sending)
| |    | +--rw sres-exponent?         uint8
| |    +--:(block-scaling)
| |    | +--rw block-scalar?          uint8
| |    +--:(u-law)
| |    | +--rw comp-bit-width?        uint8
| |    | +--rw comp-shift?            uint8
| |    +--:(beam-space-compression)
| |    | +--rw active-beam-space-coeficient-mask*   uint8
| |    | +--rw block-scaler?          uint8
| |    +--:(modulation-compression)
| |    | +--rw csf?                   uint8
```

```
| |   | +--rw mod-comp-scaler?                  uint16
| |   +--:(modulation-compression-selective-re-sending)
| |      +--rw sres-csf?                       uint8
| |      +--rw sres-mod-comp-scaler?           uint16
| +--rw frame-structure?                       uint8
| +--rw cp-type?                               enumeration
| +--rw cp-length                              uint16
| +--rw cp-length-other                        uint16
| +--rw offset-to-absolute-frequency-center    int32
| +--rw number-of-prb-per-scs* [scs]
| | +--rw scs             mcap:scs-config-type
| | +--rw number-of-prb   uint16
| +--rw e-axcid
| | +--rw o-du-port-bitmask     uint16
| | +--rw band-sector-bitmask   uint16
| | +--rw ccid-bitmask          uint16
| | +--rw ru-port-bitmask       uint16
| | +--rw eaxc-id               uint16
| +--rw downlink-radio-frame-offset          uint32
| +--rw downlink-sfn-offset                  int16
+--rw low-level-rx-endpoints* [name]
| +--rw name                                 -> /user-plane-configuration/static-low-level-rx-endpoints/name
| +--rw compression
| | +--rw iq-bitwidth?                       uint8
| | +--rw compression-type                   enumeration
| | x--rw bitwidth?                          uint8
| | +--rw (compression-format)?
```

```
| |    +--:(no-compresison)
| |    +--:(block-floating-point)
| |    | +--rw exponent?                        uint8
| |    +--:(block-floating-point-selective-re-sending)
| |    | +--rw sres-exponent?                   uint8
| |    +--:(block-scaling)
| |    | +--rw block-scalar?                    uint8
| |    +--:(u-law)
| |    | +--rw comp-bit-width?                  uint8
| |    | +--rw comp-shift?                      uint8
| |    +--:(beam-space-compression)
| |    | +--rw active-beam-space-coeficient-mask*   uint8
| |    | +--rw block-scaler?                    uint8
| |    +--:(modulation-compression)
| |    | +--rw csf?                             uint8
| |    | +--rw mod-comp-scaler?                 uint16
| |    +--:(modulation-compression-selective-re-sending)
| |      +--rw sres-csf?                        uint8
| |      +--rw sres-mod-comp-scaler?            uint16
| +--rw frame-structure?                       uint8
| +--rw cp-type?                               enumeration
| +--rw cp-length                              uint16
| +--rw cp-length-other                        uint16
| +--rw offset-to-absolute-frequency-center    int32
| +--rw number-of-prb-per-scs* [scs]
| | +--rw scs            mcap:scs-config-type
| | +--rw number-of-prb    uint16
```

```
  | +--rw ul-fft-sampling-offsets* [scs]
  | | +--rw scs                    mcap:scs-config-type
  | | +--rw ul-fft-sampling-offset?   uint16
  | +--rw e-axcid
  | | +--rw o-du-port-bitmask      uint16
  | | +--rw band-sector-bitmask    uint16
  | | +--rw ccid-bitmask           uint16
  | | +--rw ru-port-bitmask        uint16
  | | +--rw eaxc-id                uint16
  | +--rw non-time-managed-delay-enabled?       boolean
  | +--rw downlink-radio-frame-offset           uint32
  | +--rw downlink-sfn-offset                   int16
  | +--rw n-ta-offset                           uint32
  +--rw general-config
    +--rw regularization-factor-se-configured?   boolean
    +--rw little-endian-byte-order?              boolean
```

A3.14.3 A.3. o-ran-shared-cell-ecpri7-to-cpri8 Module

```
module: o-ran-shared-cell-ecpri7-to-cpri8
 +--rw shared-cell-cpri7-to-cpri8
   +--ro shared-cell-module-cap-ecpri7-to-cpri8
   | +--ro max-number-node-copy-and-combine    uint8
   | +--ro max-number-eaxcid-copy              uint8
   | +--ro max-number-eaxcid-combine           uint8
   | +--ro compression-method-supported* [] {FHM-eCPRI7-to-CPRI8}?
   +--rw shared-cell-config
     +--rw (shared-cell-copy-combine-mode)?
       +--:(COMMON)
```

```
|  +--rw shared-cell-copy-entities* [name]
|  |  +--rw name                          string
|  |  +--rw north-node-processing-element?    -> /o-ran-pe:processing-elements/ru-elements/name
|  |  +--rw south-node-processing-elements*   -> /o-ran-pe:processing-elements/ru-elements/name
|  |  +--rw shared-cell-copy-uplane-config {FHM}?
|  |     +--rw tx-eaxc-id* [eaxc-id]
|  |     |  +--rw eaxc-id    uint16
|  |     +--rw rx-eaxc-id* [eaxc-id]
|  |     |  +--rw eaxc-id    uint16
|  |     +--rw downlink-radio-frame-offset    uint32
|  |     +--rw downlink-sfn-offset            int16
|  +--rw shared-cell-combine-entities* [name]
|     +--rw name                           string
|     +--rw north-node-processing-element?      -> /o-ran-pe:processing-elements/ru-elements/name
|     +--rw south-node-processing-elements*     -> /o-ran-pe:processing-elements/ru-elements/name
|     +--rw ta3-prime-max?                      uint32
|     +--rw shared-cell-combine-uplane-config {FHM}?
|        +--rw rx-eaxc-id* [eaxc-id]
|        |  +--rw eaxc-id             uint16
|        |  +--rw comression-method
|        |     +--rw iq-bitwidth?                uint8
|        |     +--rw compression-type            enumeration
|        |     x--rw bitwidth?                   uint8
|        |     +--rw (compression-format)?
|        |        +--:(no-compresison)
|        |        +--:(block-floating-point)
|        |           +--rw exponent?             uint8
```

```
|      |      +--:(block-floating-point-selective-re-sending)
|      |      | +--rw sres-exponent?                  uint8
|      |      +--:(block-scaling)
|      |      | +--rw block-scalar?                   uint8
|      |      +--:(u-law)
|      |      | +--rw comp-bit-width?                 uint8
|      |      | +--rw comp-shift?                     uint8
|      |      +--:(beam-space-compression)
|      |      | +--rw active-beam-space-coeficient-mask*   uint8
|      |      | +--rw block-scaler?                   uint8
|      |      +--:(modulation-compression)
|      |      | +--rw csf?                            uint8
|      |      | +--rw mod-comp-scaler?                uint16
|      |      +--:(modulation-compression-selective-re-sending)
|      |        +--rw sres-csf?                       uint8
|      |        +--rw sres-mod-comp-scaler?           uint16
|      +--rw downlink-radio-frame-offset    uint32
|      +--rw downlink-sfn-offset           int16
|      +--rw n-ta-offset                   uint32
|      +--rw number-of-prb                 uint16
+--:(SELECTIVE)

augment /sc:shared-cell/sc:shared-cell-config/sc:shared-cell-copy-combine-mode:
 +--:(eCPRI7-to-CPRI8)
   +--rw shared-cell-copy-entities-ecpri7-to-cpri8* [name]
   | +--rw name                              string
   | +--rw low-level-tx-endpoint?            -> /uc-hub:user-plane-configuration/low-level-tx-endpoints/name
```

```
  | +--rw south-node-processing-elements-cpri*   -> /o-ran-pe:processing-elements/ru-elements/name
  +--rw shared-cell-combine-entities-ecpri7-to-cpri8* [name]
    +--rw name                              string
    +--rw low-level-rx-endpoint?                -> /uc-hub:user-plane-configuration/low-level-tx-endpoints/name
    +--rw south-node-processing-elements-cpri*   -> /o-ran-pe:processing-elements/ru-elements/name
```

A3.14.4 A.4 o-ran-interfaces-cpri Module

```
module: o-ran-interfaces-cpri
 +--ro cpri-status
   +--ro cpri-startup-status
   | +--ro cpri-startup-state?   enumeration
   +--ro cpri-link-status
     +--ro cpri-link-state?   enumeration

 augment /if:interfaces/if:interface:
  +--rw cpri-interface
    +--rw mac-address?      yang:mac-address
    +--rw port-reference
    | +--rw port-name?     -> /hw:hardware/component/name
    | +--rw port-number?   uint8
    | +--rw cpri-config {CPRI-INTERFACE}?
    |    +--rw port-mode?         enumeration {CPRI-INTERFACE}?
    |    +--rw cpri-version?      string
    |    +--rw line-bit-rate?     decimal64
    |    +--rw ethernet-pointer?   uint8
    |    +--rw samplewidth?       uint8
    |    +--rw linecoding?        enumeration
    +--ro ber-counts?       string
```

```
rpcs:
  +---x reset-interface-counters

notifications:
  +---n cpri-startup-state-change
  |  +--ro cpri-startup-state?   -> /cpri-status/cpri-startup-status/cpri-startup-state
  +---n cpri-link-state-change
     +--ro cpri-link-state?   -> /cpri-status/cpri-link-status/cpri-link-state
```

A3.14.5 A.5 o-ran-processing-element-cpri Module

```
module: o-ran-processing-element-cpri
  augment /o-ran-pe:processing-elements/o-ran-pe:ru-elements/o-ran-pe:transport-flow:
    +--rw cpri-flow
       +--rw compression-flag?              boolean
       +--rw iq-start-bit?                  uint16
       +--rw iq-end-bit?                    uint16
       +--rw inband-compression-start-bit?  uint16
       +--rw inband-compression-end-bit?    uint16
       +--rw agc-start-bit?                 uint16
       +--rw agc-end-bit?                   uint16
```

A3.14.6 A.6 o-ran-delay-management-ru-cpri Module

```
module: o-ran-delay-management-ru-cpri
  +--r ru-delay-profile
     +--r ru-delay-profile-list-radio-related
```

```
   +--ru-delay-profile-common
  +--rw ru-delay-config-cpri-link-related
 +--rw ru-delay-config-data-flow-related
```

A3.14.7 A.7. o-ran-system-time Module

```
module: o-ran-system-time
  +--rw system-time
    +--rw current-datetime?                ietf-yang-types:date-and-time
```

A3.14.8 A.8. o-ran-sync-cpri Module

```
module: o-ran-sync-cpri
  +--rw sync-status
    +--rw reporting-period?     uint32
    +--ro sync-state?          enumeration
    +--ro error-message?       string
    +--ro source-port-number?   -> if:interfaces/interface/o-ran-int-cpri:cpri-interface/port-reference/port-number

  notifications:
   +---n cpri-state-change
      +--ro cpri-sync-state?   -> /sync-status/sync-state
```

Annex ZZZ O-RAN Adopter License Agreement

BY DOWNLOADING, USING OR OTHERWISE ACCESSING ANY O-RAN SPECIFICATION, ADOPTER AGREES TO THE TERMS OF THIS AGREEMENT.

This O-RAN Adopter License Agreement (the "Agreement") is made by and between the O-RAN Alliance and the entity that downloads, uses or otherwise accesses any O-RAN Specification, including its Affiliates (the "Adopter").

This is a license agreement for entities who wish to adopt any O-RAN Specification.

Section 1: DEFINITIONS

1.1 "Affiliate" means an entity that directly or indirectly controls, is controlled by, or is under common control with another entity, so long as such control exists. For the purpose of this Section, "Control" means beneficial ownership of fifty (50%) percent or more of the voting stock or equity in an entity.

1.2 "Compliant Implementation" means any system, device, method or operation (whether implemented in hardware, software or combinations thereof) that fully conforms to a Final Specification.

1.3 "Adopter(s)" means all entities, who are not Members, Contributors or Academic Contributors, including their Affiliates, who wish to download, use or otherwise access O-RAN Specifications.

1.4 "Minor Update" means an update or revision to an O-RAN Specification published by O-RAN Alliance that does not add any significant new features or functionality and remains interoperable with the prior version of an O-RAN Specification. The term "O-RAN Specifications" includes Minor Updates.

1.5 "Necessary Claims" means those claims of all present and future patents and patent applications, other than design patents and design registrations, throughout the world, which (i) are owned or otherwise licensable by a Member, Contributor or Academic Contributor during the term of its Member, Contributor or Academic Contributorship; (ii) such Member, Contributor or Academic Contributor has the right to grant a license without the payment of consideration to a third party; and (iii) are necessarily infringed by a Compliant Implementation (without considering any Contributions not included in the Final Specification). A claim is necessarily infringed only when it is not possible on technical (but not commercial) grounds, taking into account normal technical practice and the state of the art generally available at the date any Final Specification was published by the O-RAN Alliance or the date the patent claim first came into existence, whichever last occurred, to make, sell, lease, otherwise dispose of, repair, use or operate a Compliant Implementation without infringing that claim. For the avoidance of doubt in exceptional cases where a Final Specification can only be implemented by technical solutions, all of which infringe patent claims, all such patent claims shall be considered Necessary Claims.

1.6 "Defensive Suspension" means for the purposes of any license grant pursuant to Section 3, Member, Contributor, Academic Contributor, Adopter, or any of their Affiliates, may have the discretion to include in their license a term allowing the licensor to suspend the license against a licensee who brings a patent infringement suit against the licensing Member, Contributor, Academic Contributor, Adopter, or any of their Affiliates.

Section 2: COPYRIGHT LICENSE

2.1 Subject to the terms and conditions of this Agreement, O-RAN Alliance hereby grants to Adopter a nonexclusive, nontransferable, irrevocable, non-sublicensable, worldwide copyright license to obtain, use and modify O-RAN Specifications, but not to further distribute such O-RAN Specification in any modified or unmodified way, solely in furtherance of implementations of an O-RAN

Specification.

2.2 Adopter shall not use O-RAN Specifications except as expressly set forth in this Agreement or in a separate written agreement with O-RAN Alliance.

Section 3: FRAND LICENSE

3.1 Members, Contributors and Academic Contributors and their Affiliates are prepared to grant based on a separate Patent License Agreement to each Adopter under Fair Reasonable And Non- Discriminatory (FRAND) terms and conditions with or without compensation (royalties) a nonexclusive, non-transferable, irrevocable (but subject to Defensive Suspension), non-sublicensable, worldwide patent license under their Necessary Claims to make, have made, use, import, offer to sell, lease, sell and otherwise distribute Compliant Implementations; provided, however, that such license shall not extend: (a) to any part or function of a product in which a Compliant Implementation is incorporated that is not itself part of the Compliant Implementation; or (b) to any Adopter if that Adopter is not making a reciprocal grant to Members, Contributors and Academic Contributors, as set forth in Section 3.3. For the avoidance of doubt, the foregoing licensing commitment includes the distribution by the Adopter's distributors and the use by the Adopter's customers of such licensed Compliant Implementations.

3.2 Notwithstanding the above, if any Member, Contributor or Academic Contributor, Adopter or their Affiliates has reserved the right to charge a FRAND royalty or other fee for its license of Necessary Claims to Adopter, then Adopter is entitled to charge a FRAND royalty or other fee to such Member, Contributor or Academic Contributor, Adopter and its Affiliates for its license of Necessary Claims to its licensees.

3.3 Adopter, on behalf of itself and its Affiliates, shall be prepared to grant based on a separate Patent License Agreement to each Members, Contributors, Academic Contributors, Adopters and their Affiliates under Fair Reasonable And Non-Discriminatory (FRAND) terms and conditions with or without compensation (royalties) a nonexclusive, non-transferable, irrevocable (but subject to Defensive Suspension), non-sublicensable, worldwide patent license under their Necessary Claims to make, have made, use, import, offer to sell, lease, sell and otherwise distribute Compliant Implementations; provided, however, that such license will not extend: (a) to any part or function of a product in which a Compliant Implementation is incorporated that is not itself part of the Compliant Implementation; or (b) to any Members, Contributors, Academic Contributors, Adopters and their Affiliates that is not making a reciprocal grant to Adopter, as set forth in Section 3.1. For the avoidance of doubt, the foregoing licensing commitment includes the distribution by the Members', Contributors', Academic Contributors', Adopters' and their Affiliates' distributors and the use by the Members', Contributors', Academic Contributors', Adopters' and their Affiliates' customers of such licensed Compliant Implementations.

Section 4: TERM AND TERMINATION

4.1 This Agreement shall remain in force, unless early terminated according to this Section 4.

4.2 O-RAN Alliance on behalf of its Members, Contributors and Academic Contributors may terminate this Agreement if Adopter materially breaches this Agreement and does not cure or is not capable of curing such breach within thirty (30) days after being given notice specifying the breach.

4.3 Sections 1, 3, 5 - 11 of this Agreement shall survive any termination of this Agreement. Under surviving Section 3, after termination of this Agreement, Adopter will continue to grant licenses (a) to entities who become Adopters after the date of termination; and (b) for future versions of ORAN Specifications that are backwards compatible with the version that was current as of the date of termination.

Section 5: CONFIDENTIALITY

Adopter will use the same care and discretion to avoid disclosure, publication, and dissemination of O-RAN Specifications to third parties, as Adopter employs with its own confidential information, but no less than reasonable care. Any disclosure by Adopter to its Affiliates, contractors and consultants should be subject to an obligation of confidentiality at least as restrictive as those contained in this Section. The foregoing obligation shall not apply to any information which is: (1) rightfully known by Adopter without any limitation on use or disclosure prior to disclosure; (2) publicly available through no fault of Adopter; (3) rightfully received without a duty of confidentiality; (4) disclosed by O-RAN Alliance or a Member, Contributor or Academic Contributor to a third party without a duty of confidentiality on such third party; (5) independently developed by Adopter; (6) disclosed pursuant to the order of a court or other authorized governmental body, or as required by law, provided that Adopter provides reasonable prior written notice to O-RAN Alliance, and cooperates with O-RAN Alliance and/or the applicable Member, Contributor or Academic Contributor to have the opportunity to oppose any such order; or (7) disclosed by Adopter with O-RAN Alliance's prior written approval.

Section 6: INDEMNIFICATION

Adopter shall indemnify, defend, and hold harmless the O-RAN Alliance, its Members, Contributors or Academic Contributors, and their employees, and agents and their respective successors, heirs and assigns (the "Indemnitees"), against any liability, damage, loss, or expense (including reasonable attorneys' fees and expenses) incurred by or imposed upon any of the Indemnitees in connection with any claims, suits, investigations, actions, demands or judgments arising out of Adopter's use of the licensed O-RAN Specifications or Adopter's commercialization of products that comply with O-RAN Specifications.

Section 7: LIMITATIONS ON LIABILITY; NO WARRANTY

EXCEPT FOR BREACH OF CONFIDENTIALITY, ADOPTER'S BREACH OF SECTION 3, AND ADOPTER'S INDEMNIFICATION OBLIGATIONS, IN NO EVENT SHALL ANY PARTY BE LIABLE TO ANY OTHER

PARTY OR THIRD PARTY FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RESULTING FROM ITS PERFORMANCE OR NON-PERFORMANCE UNDER THIS AGREEMENT, IN EACH CASE WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, AND WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. O-RAN SPECIFICATIONS ARE PROVIDED "AS IS" WITH NO WARRANTIES OR CONDITIONS WHATSOEVER, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. THE O-RAN ALLIANCE AND THE MEMBERS, CONTRIBUTORS OR ACADEMIC CONTRIBUTORS EXPRESSLY DISCLAIM ANY WARRANTY OR CONDITION OF MERCHANTABILITY, SECURITY, SATISFACTORY QUALITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, ERROR-FREE OPERATION, OR ANY WARRANTY OR CONDITION FOR O-RAN SPECIFICATIONS.

Section 8: ASSIGNMENT

Adopter may not assign the Agreement or any of its rights or obligations under this Agreement or make any grants or other sublicenses to this Agreement, except as expressly authorized hereunder, without having first received the prior, written consent of the O-RAN Alliance, which consent may be withheld in O-RAN Alliance's sole discretion. O-RAN Alliance may freely assign this Agreement.

Section 9: THIRD-PARTY BENEFICIARY RIGHTS

Adopter acknowledges and agrees that Members, Contributors and Academic Contributors (including future Members, Contributors and Academic Contributors) are entitled to rights as a third-party beneficiary under this Agreement, including as licensees under Section 3.

Section 10: BINDING ON AFFILIATES

Execution of this Agreement by Adopter in its capacity as a legal entity or association constitutes that legal entity's or association's agreement that its Affiliates are likewise bound to the obligations that are applicable to Adopter hereunder and are also entitled to the benefits of the rights of Adopter hereunder.

Section 11: GENERAL

This Agreement is governed by the laws of Germany without regard to its conflict or choice of law provisions.

This Agreement constitutes the entire agreement between the parties as to its express subject matter and expressly supersedes and replaces any prior or contemporaneous agreements between the parties, whether written or oral, relating to the subject matter of this Agreement.

Adopter, on behalf of itself and its Affiliates, agrees to comply at all times with all applicable laws, rules and regulations with respect to its and its Affiliates' performance under this Agreement, including without limitation, export control and antitrust laws. Without limiting the generality of the foregoing, Adopter acknowledges that this Agreement prohibits any communication that would violate the antitrust laws.

By execution hereof, no form of any partnership, joint venture or other special relationship is created between Adopter, or O-RAN Alliance or its Members, Contributors or Academic Contributors. Except as expressly set forth in this Agreement, no party is authorized to make any commitment on behalf of Adopter, or O-RAN Alliance or its Members, Contributors or Academic Contributors.

In the event that any provision of this Agreement conflicts with governing law or if any provision is held to be null, void or otherwise ineffective or invalid by a court of competent jurisdiction, (i) such provisions will be deemed stricken from the contract, and (ii) the remaining terms, provisions, covenants and restrictions of this Agreement will remain in full force and effect.