



**PESQUISA E DESENVOLVIMENTO DA
UNIDADE DE RÁDIO (O-RU) 5G**

A2.3 – Desenvolvimento de lógica programável

OpenRAN@Brasil – Fase 2

SUMÁRIO

1	INTRODUÇÃO	9
2	LEVANTAMENTO DO ESTADO DA ARTE.....	11
2.1	TECNOLOGIA 5G.....	11
2.2	ARQUITETURA OPENRAN.....	11
3	BASE TEÓRICA.....	13
3.1	5G E OPENRAN.....	13
3.1.1	<i>Camada Física 5g.....</i>	<i>13</i>
3.1.1.1	Forma de onda.....	13
3.1.1.2	Modulação.....	14
3.1.1.3	Codificação de Canal.....	14
3.1.1.4	Recursos Físicos.....	15
3.1.1.5	Canais Físicos.....	15
3.1.1.6	Sinais Físicos.....	16
3.1.2	<i>OpenRAN.....</i>	<i>17</i>
3.1.2.1	Divisão Funcional 7.2X.....	19
4	ESTUDOS DA TECNOLOGIA E PLATAFORMA DE DESENVOLVIMENTO.....	25
4.1	SISTEMAS EMBARCADOS.....	25
4.2	SISTEMAS EMBARCADOS USANDO FERRAMENTAS PÉTALINUX (SOFTWARE).....	26
5	ANÁLISE E SIMULAÇÕES SEM PLACA.....	27
5.2.1	<i>Plano do Usuário.....</i>	<i>30</i>
5.2.2	<i>Plano de Controle.....</i>	<i>35</i>
6	AVALIAÇÃO DO COMPORTAMENTO DA LÓGICA PROGRAMÁVEL EM SIMULAÇÃO.....	38
6.1	SIMULAÇÃO REDE ETHERNET.....	41
6.2	SIMULAÇÃO DFE OFDM.....	46
6.3	RESULTADOS OBTIDOS COM AS SIMULAÇÕES.....	49
7	SETUP DE LABORATÓRIO.....	50
7.1	EQUIPAMENTOS UTILIZADOS.....	50
7.2.1	<i>Keysight PathWave Signal Generation.....</i>	<i>51</i>
7.2.2	<i>Keysight Open RAN Studio.....</i>	<i>54</i>
7.2.3	<i>Keysight PathWave 89600 VSA.....</i>	<i>57</i>
7.3	TESTES EM LOOPBACK.....	58
7.4	BRING-UP DO KIT DE DESENVOLVIMENTO.....	62
7.4.1	<i>Preparação de Placas de Desenvolvimento AMD modelo ZCU670.....</i>	<i>62</i>
7.4.2	<i>A Importância da Placa de Desenvolvimento:.....</i>	<i>63</i>
7.4.3	<i>Desafios na Preparação da Placa de Desenvolvimento:.....</i>	<i>63</i>
7.4.4	<i>Equipamentos e softwares utilizados na preparação da placa ZCU670:.....</i>	<i>63</i>
7.4.5	<i>Detalhamento da execução da preparação da placa ZCU670.....</i>	<i>64</i>
7.4.6	<i>Desafios com relação à preparação da placa de desenvolvimento.....</i>	<i>70</i>
8	CONCLUSÃO.....	72
9	REFERÊNCIAS BIBLIOGRÁFICAS.....	73
10	HISTÓRICO DE ALTERAÇÕES DO DOCUMENTO CONSOLIDADO.....	75
11	EXECUÇÃO E APROVAÇÃO.....	76

Figuras

Figura 1 - Matrix de Alocação de recursos.....	15
Figura 2 - Diagrama da cadeia de processamento 5G e fragmentação[26]	19
Figura 3 - Descrição do Rádio Split 7.2A Downlink DU-RU[27].....	20
Figura 4 - Descrição do Rádio Split 7.2B Downlink DU-RU[27].....	21
Figura 5 - Descrição do Rádio Split 7.2 UpLink DU-RU[27].....	21
Figura 6 - Arquitetura de referência de uma O-RU[27]	22
Figura 7 - Diagrama simplificado de blocos O-RAN da AMD	27
Figura 8 - Leitura dos requisitos	28
Figura 9 - Campos que compõem o protocolo do Plano U	31
Figura 10 – Esquemático Open RAN IP Example Design	38
Figura 11 – Simulação gerada com os dados da Keysight Open RAN Studio equipamento emulando uma O-DU.....	40
Figura 12 – Subsistema rede Ethernet.....	41
Figura 13 - Ambiente de Simulação	43
Figura 14 - Drivers	44
Figura 15 – Simulação do processo de configuração do modulo TIMERSS0	44
Figura 16 – Simulação do processo de configuração do modulo ORANSS0	45
Figura 17 - Simulação de Downlink.....	46
Figura 18 - arquitetura DFE OFDM	47
Figura 19 - Mensagens de saída em console do QEMU	48
Figura 20 - XSim Vivado Atividade de escrita/leitura do AXI Lite	48
Figura 21 - Xsim Vivado Protocolo AXI Stream	48
Figura 22 - Xsim Vivado Máquinas de estado.....	49
Figura 23 - Painel frontal do S5040A	50
Figura 24 - Chassi PXI com os módulos para geração e análise de sinais RF	51
Figura 25 - Controle de configuração de portadora do software Pathwave Signal generation	52
Figura 26 - Configuração básica de exemplo do BWP	52
Figura 27 - Configuração do Canal. A esquerda bloco de configuração do SS /PBCH e a direita do bloco DL-SCH.....	53
Figura 28 - Painel de configuração do eAxC.....	55
Figura 29- Painel de configuração de compressão.....	55
Figura 30 - Painel de configuração dos parâmetros de temporização.....	56
Figura 31 - Painel de configuração da opção de vLAN.....	57
Figura 32 - Adaptador para loopback (esquerda) e o mesmo sendo utilizado no equipamento S5040	58
Figura 33 - Painel de monitoramento do tráfego nas portas óticas do equipamento	59
Figura 34 - Painel de análise dos sinais trafegando no fronthaul	60
Figura 35 - Chassi PXI com módulo RF conectado em loopback.....	60

Figura 36 - Sinal 100MHz, demodulado com estatísticas da qualidade de sinal.....	62
Figura 37 – Setup de preparação da placa ZCU670 montado no laboratório do Instituto Eldorado ..	66
Figura 38 – M9410A e S5040A montados no rack no laboratório do Instituto Eldorado	66
Figura 39 – Conexões das placas AMD XM755 e ZCU670.....	67

Glossário

Beam: Padrão de radiação sem restrição ao lóbulo principal.

O-CU: Unidade de Controle O-RAN – um nó lógico que hospeda PDCP, RRC, SDAP e outras funções de controle

O-DU: Unidade Distribuída O-RAN: um nó lógico que hospeda camadas RLC/MAC/High-PHY com base em uma camada funcional inferior.

O-RU: Unidade de rádio O-RAN: um nó lógico que hospeda a camada Low-PHY e o processamento de RF baseado em uma camada funcional inferior.

PRB: Bloco de recursos físicos. Um grupo de 12 subportadoras consecutivas de um símbolo OFDM.

RE: Elemento de recurso. Uma subportadora de um símbolo OFDM.

Slot: Define uma duração de tempo.

Fronthaul: Um conexão lógica entre a O-DU e a O-RU.

eAxC: Portadora de antena estendida: um fluxo de dados para uma única antena (ou fluxo espacial) para uma única portadora em um único setor.

High-PHY: Partes do processamento da camada PHY no lado O-DU da interface fronthaul.

Low-PHY: Partes do processamento da camada PHY no lado O-RU da interface fronthaul.

NR: New Radio, uma tecnologia de acesso por rádio 5G.

Abreviações

- IQ:** In-phase e Quadrature-phase.
- DL:** Downlink.
- PHY:** Physical Layer.
- PLL:** Phase Locked Loop.
- MMCM:** Módulo Multi-Clock Manager.
- PTP:** Precision Time Protocol.
- TOD:** Time of Day.
- RX:** Receiver.
- TX:** Transmitter.
- UL:** Uplink.
- RB:** Resource Block.
- RE:** Resource Element.
- eAxC_ID:** Identificador eAxC.
- CC:** Component carrier (Portadora).
- IP/UDP:** Internet Protocol/User Datagram Protocol.
- BBU:** Baseband Unit.
- SSB:** Synchronization Signal Block.
- PSS:** Primary Synchronization Signal.
- SSS:** Secondary Synchronization Signal.
- PBCH:** Physical Broadcast Channel.
- PRACH:** Physical Random-Access Channel.
- VLAN:** Virtual Local Area Network.
- IFFT:** Inverse Fast Fourier Transformation.
- FFT:** Fast Fourier Transformation.
- OFDM:** Orthogonal Frequency Division Multiplexing.
- TRD:** Target Referency Design.
- RLC:** Radio Link Control.
- MAC:** Media Access Control.
- PDCP:** Packet Data Convergence Protocol (Protocolo de Convergência de Dados de Pacotes).
- RRC:** Radio Resource Control (Controle de Recursos de Rádio).
- SDAP:** Service Data Adaptation Protocol (Protocolo de Adaptação de Dados de Serviço).
- eCPRI:** evolved Common Public Radio Interface.
- FPGA:** Field Programmable Gate Array.

gNB: Next generation NodeB (applies to NR).

O-DU: ORAN Distributed Unit.

O-RU: ORAN Radio Unit.

RF: Radio Frequency.

RFSoc: Radio Frequency System on Chip (Sistema de Radiofrequência em Chip).

MPSoc: Multiprocessor System on Chip (Sistema de Multiprocessamento em Chip).

Resumo

Este relatório contém uma descrição das atividades realizadas até o presente momento, referente ao desenvolvimento de lógica programável, que tem como objetivo o desenvolvimento do projeto funcional na ferramenta correspondente ao fabricante de FPGA escolhido. Esse projeto será demonstrado em kits de desenvolvimento disponíveis e, não necessariamente, no hardware de processamento em banda base desenvolvido na atividade 2.2 desse projeto. As atividades descritas nesse relatório englobam desde o levantamento do estado da arte, necessárias para apropriação do conhecimento e aprendizado, até análises e simulações, estudo de setup laboratorial, realização de treinamentos e documentação dos resultados preliminares obtidos.

1 Introdução

A atividade relacionada ao desenvolvimento de lógica programável tem como principal objetivo o desenvolvimento ou integração das funções lógicas do fronthaul em blocos internos da FPGA, divididos em interface fronthaul e *Low-PHY*.

Para a Interface *fronthaul*, a especificação da interface de rádio pública comum aprimorada (eCPRI, do inglês: *evolved Common Public Radio Interface*) tem o objetivo de disponibilizar publicamente especificações de unidades remotas interfaces da cabeça de rádio remoto (RRH, do inglês: *Remote Radio Head*) ou equipamento de rádio, e a unidade estação base (BBU, do inglês: *Base Band Unit*) ou Equipamento de controle de Rádio (REC, do inglês: *Radio Equipment Controller*) via o que é chamado de rede de transporte fronthaul. No caso da especificação eCPRI, é possível dizer que é adequada ao 5G e aos requisitos do Open RAN, que suporta uma maior flexibilidade no posicionamento da partição funcional dentro da camada física de uma estação base celular. Neste caso, a camada física é dividida em *low-PHY* e *high-PHY*. Destas funcionalidades para serem implementadas em um componente de Arranjo de Porta Programável em Campo (FPGA, do inglês: *Field Programmable Gate Array*), temos:

- Plano de Controle (Plano-C, *C-Plane* ou *Control Plane*, em inglês) e Plano de Usuário (Plano-U, *U-Plane* ou *User Plane*, em inglês) que correspondem aos símbolos IQ no domínio da frequência;
- Plano de Gerenciamento (Plano-M, *M-Plane* ou *Manager Plane*, em inglês) que correspondem a mensagens de escrita e leitura para o controle e a gerência;
- Plano de Sincronização (Plano-S, *S-Plane* ou *Synchronization Plane*, em inglês) que correspondem a mensagens periódicas, transmitem a informação da base de tempo.

Para fins de sincronização plena da operação da Unidade de Distribuída ORAN (O-DU) e a Unidade de Radio ORAN (O-RU) elas necessitam realizar o compartilhamento de dados relacionados entre o Plano-U, Plano-C, Plano-M e Plano-S através da interface de *fronthaul*. Devido à crescente demanda por dados e pelo aumento da complexidade de futuras redes de comunicação, a quantidade de dados que deve ser transmitida pelo *fronthaul* tem aumentado significativamente. Esta atividade visa desenvolver algoritmos de compressão e de descompressão para o *fronthaul* da O-RU.

Para a parte referente a *Low-PHY*, o objetivo principal é a implementação dos blocos necessários para o processamento de sinais de tempo discreto para implementação da parte baixa da camada física da pilha de protocolos do Projeto de Parceria da 3ª Geração (3GPP, do inglês: *3rd Generation Partnership Project*), tais como:

- *Downlink* (DL): Pré-codificação, Mapeamento da Matrix de Alocação de Recursos (RE Map, do inglês: *Resource Element Mapping*), Filtragem espacial (do inglês: *Beamforming*), realização da Transformada de Fourier Inversa Rápida (IFFT, do inglês: *Inverse Fast Fourier Transform*), e inserção de um prefixo cíclico (CP, do inglês: *cyclic prefix*).
- *Uplink* (UL): Detecção de sinais do canal físico de acesso aleatório (PRACH, do inglês: *Physical Random-Access Channel*), realização da Transformada Rápida de Fourier (FFT, do inglês: *Fast Fourier Transform*), Funcionalidade de pré-filtragem ou redução de portas, e a remoção do CP.
- *Front-end* digital: processamento de sinais de tempo discreto para implementações de funções adicionais tais como: interpolação/decimação fracionária, linearização de amplificadores de potência de rádio frequência (RF) utilizando pré-distorção digital (DPD, em inglês: *Digital Pre-Distortion*), compensação de imperfeições dos circuitos analógicos de conversão de frequências e, opcionalmente, redução de redução da relação de potência pico/média (PAPR, do inglês: *Peak to Average Power Ratio*).

Com isso, é esperado ter, ao final dessa atividade, o projeto funcional na ferramenta correspondente ao fabricante de FPGA escolhido. Esse projeto poderá ser demonstrado em Kits de desenvolvimento disponíveis, e, não necessariamente, no hardware de processamento em banda base da Atividade 2.2. Estão previstas duas versões de cada entregável, uma ao final do primeiro e outra ao final do segundo ano. Esse relatório tem como objetivo principal apresentar as atividades já realizadas e os resultados preliminares obtidos.

2 Levantamento do estado da arte

A indústria de telecomunicações está passando por uma mudança transformadora com a chegada da tecnologia 5G e o surgimento das arquiteturas de Rede de Acesso por Rádio Aberto (OpenRAN).

2.1 Tecnologia 5g

O 5G representa a quinta geração de tecnologia sem fio, prometendo melhorias significativas em taxas de dados, latência, confiabilidade e conectividade. Características-chave do 5G incluem:

- Banda Larga Móvel Aprimorada (eMBB): Fornecendo velocidades de dados ultrarápidas e aumento de capacidade para suportar streaming de vídeo de alta definição, realidade virtual e aplicativos de jogos imersivos.
- Comunicação Ultraconfiável de Baixa Latência (URLLC): Possibilitando aplicações críticas como veículos autônomos, automação industrial e cirurgias remotas com latência extremamente baixa e alta confiabilidade.
- Comunicação Massiva entre Máquinas (mMTC): Apoiando a implantação massiva de dispositivos Internet das Coisas (IoT), sensores e dispositivos inteligentes com uso eficiente de recursos de rede.
- Fragmentação de Rede: Permitindo a criação de múltiplas redes virtuais em uma infraestrutura física compartilhada, adaptadas às necessidades específicas de diferentes casos de uso e indústrias.

2.2 Arquitetura OpenRAN

OpenRAN é uma abordagem fragmentada para a construção de redes de acesso por rádio, baseada em interfaces abertas, interoperabilidade e princípios de redes definidas por software. Componentes-chave do OpenRAN incluem:

- Unidade Centralizada (CU): responsável por funções de processamento de camada superior, como processamento de banda base, escalonamento e coordenação.
- Unidade Distribuída (DU): lida com funções de camadas inferiores, como processamento de radiofrequência (RF), modulação e codificação.
- Unidade de Rádio (RU): interface com a antena e converte sinais digitais em ondas de rádio para transmissão e recepção.
- Interfaces Abertas: definidas por organizações como a O-RAN Alliance, permitindo interoperabilidade entre componentes de diferentes fornecedores e promovendo inovação por meio de um ecossistema diversificado.

O estado da arte para o 5G e a OpenRAN é caracterizado por vários avanços e inovações notáveis:

- Antenas MIMO Massivas (Múltiplas Entradas e Múltiplas Saídas): aumentando a eficiência espectral e a capacidade ao empregar um grande número de elementos de antena para formação de feixes e multiplexação espacial.
- Arquitetura Nativa de Nuvem: alavancando virtualização, containerização e tecnologias de orquestração para permitir implantações de rede flexíveis, escaláveis e econômicas.
- Inteligência Artificial (IA) e Aprendizado de Máquina (AM): otimizando o desempenho da rede, alocação de recursos e manutenção preditiva por meio de algoritmos e análises inteligentes.
- Software de Código Aberto: impulsionando o desenvolvimento de projetos de código aberto como a Plataforma de Automação de Rede Aberta (ONAP) e OpenAirInterface (OAI), promovendo colaboração e inovação na comunidade de telecomunicações.

3 Base Teórica

Para a implementação da unidade de rádio e o entendimento das funcionalidades da plataforma de referência fornecida pelo fabricante AMD, foi necessário tanto o estudo de redes 5G e das especificações OpenRAN quanto o treinamento na plataforma de desenvolvimento da lógica digital.

3.1 5G e OpenRAN

3.1.1 Camada Física 5g

3.1.1.1 Forma de onda

O novo radio 5G (NR, *New Radio*, em inglês) utiliza CP-OFDM (*Cyclic Prefix - Orthogonal Frequency Division Multiplexing*, em inglês) tanto para *Downlink* quanto para *Uplink*, com espaçamento variável entre sub-portadoras, o espaçamento é definido como $15 \cdot 2^\mu$ kHz, onde μ é denominado numerologia. Como resultado do espaçamento variável das sub-portadoras, a largura de banda do sinal também varia.

A Tabela 1 apresenta as características do frame 5G NR. Cada frame possui uma duração de 10 milissegundos e é subdividido em sub-frames de 1 ms. Por sua vez, cada sub-frame é dividido em um número variável de slots, dependendo da numerologia escolhida. Cada slot contém 14 símbolos OFDM. O número máximo de portadoras ativas em um símbolo OFDM é igual a 3300, portanto a largura de banda máxima disponível para cada μ é $3300 \cdot 15 \cdot 2^\mu$ KHz, exceto para $\mu = 4$, onde o número máximo de portadoras ativas é igual a 1644 para não exceder a largura de banda máxima permitida de 400 MHz.

Tabela 1.- Características do Frame 5G NR

Numerologia (μ)	Espaçamento da sub-portadora (KHz)	Largura de Banda Máxima (MHz)	Nro Slots (1 Sub-frame)	Nro Slots (1 frame)	Símbolos (1 Frame)
0	15	49.5	1	10	140
1	30	99	2	20	280
2	60	198	4	40	560
3	120	396	8	80	1120
4	240	394.56	16	160	2240

5G NR pode operar em modo TDD (*time division duplexing*, em inglês) ou FDD (*Frequency division Duplexing*, FDD), a escolha de um ou o outro modo depende da frequência FR (Frequency Range) de operação. Para frequências abaixo de 7GHz, chamadas de FR1 [23] o modo pode ser TDD ou FDD, já para a faixa de frequência FR2 (> 24 GHz) o modo é

exclusivamente TDD. A Tabela 2 mostra algumas das de faixas de operação do GnB, a tabela completa pode ser acessada em [23]. Na Tabela 3Tabela 3.- Definições de Faixas de Frequências, as frequências de operação para FR1 e FR2 [23].

Tabela 2.- Modo Duplex para várias frequências de operação

Faixa Frequência	Banda	Modo Duplex	Uplink (MHz)	Downlink (MHz)
FR1	N25	FDD	1850– 1915	1930– 1995
	N38	TDD	2570– 2620	
	N78	TDD	3300– 3800	
FR2	N257	TDD	26500– 29500	
	N258	TDD	24250 – 27500	
	N259	TDD	39500– 43500	

Tabela 3.- Definições de Faixas de Frequências

Faixa de Frequência	Range de frequência Correspondente (MHz)
FR1	410– 7125
FR2	24250– 52600

3.1.1.2 Modulação

5G NR suporta modulações de QPSK (*quadrature phase shift keying*, em inglês) e 16/64/256 QAM (*Quadrature Amplitude Modulation*, em inglês) tanto para *Downlink* quanto para *Uplink* [4]. Adicionalmente, $\pi/2$ -BPSK (*Binary Phase Shift Keying*) é suportado para *Uplink*.

3.1.1.3 Codificação de Canal

5G NR utiliza dois tipos de códigos avançados para garantir transmissão fluida e confiável: códigos LDPC (*Low Density Parity Check*) para dados de banda larga móvel (MBB) e códigos polares para sinalização de controle. Diferentemente de outras tecnologias sem fio, o NR usa uma estrutura LDPC "adaptável à taxa", permitindo taxas de transmissão flexíveis e uma técnica chamada HARQ para redundância incremental, garantindo robustez na transmissão [23].

Para sinalização de controle, onde os pacotes de informação são menores, a NR usa códigos polares. Combinando esses códigos com outros e aplicando técnicas avançadas de decodificação, a NR garante alta performance mesmo com blocos de dados curtos. Para os menores pacotes, códigos Reed-Muller entram em cena.

3.1.1.4 Recursos Físicos

Recursos físicos de tempo-frequência são definidos dependendo da numerologia e a configuração do rádio usada. Uma matriz de alocação de recursos é definida, onde cada elemento, chamado RE (*Resource Element*, em inglês), desta matriz representa uma portadora do símbolo OFDM. A Figura 1 - Matrix de Alocação de recursos, mostra como são distribuídos os recursos de tempo-frequência. No tempo, 14 símbolos OFDM formam um slot, um número variável de slots um sub-frame (lembrando que o número de slots varia com a numerologia), e 10 sub-frames um frame. Na frequência, 12 sub-portadoras de um símbolo OFDM formam um RB (*Resource Block*, em inglês), e no máximo 275 RBs podem ser alocados para um símbolo ($275 \times 12 = 3300$ Sub-portadoras).

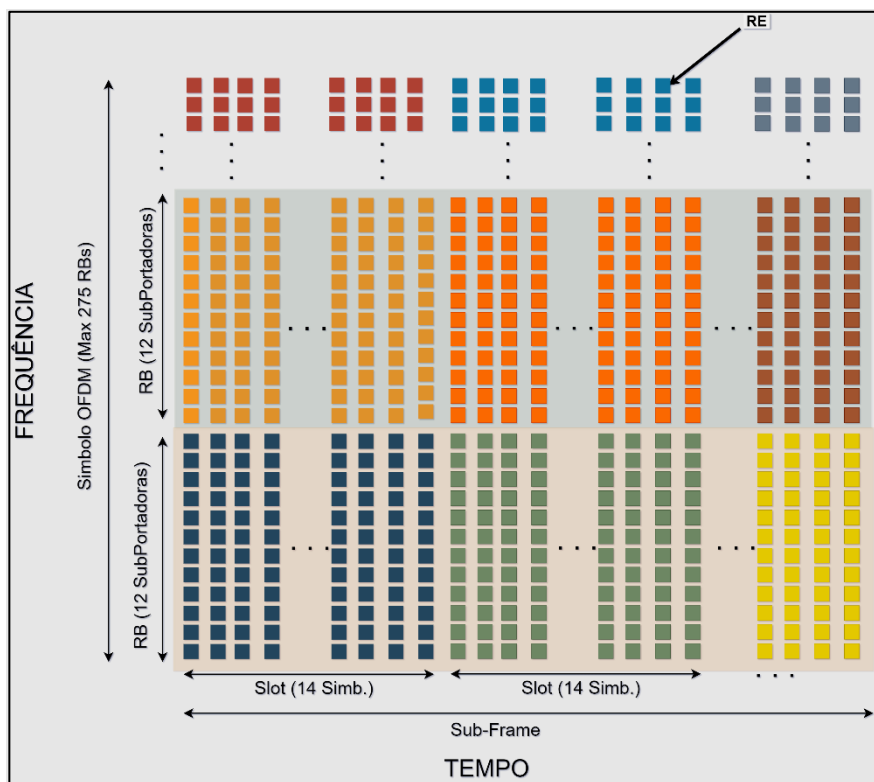


Figura 1 - Matrix de Alocação de recursos

3.1.1.5 Canais Físicos

Toda informação originada em camadas superiores à camada física e transportada pela camada física é chamada de *canal físico* [23]. Os canais físicos podem ser classificados em canais físicos de *Uplink* (UL) e canais físicos de *Downlink* (DL):

- Canais Físicos de DL

- Canal compartilhado de DL PDSCH (*Physical Downlink Shared channel*, em inglês): Usado para o envio de dados.
- Canal de controle de DL PDCCH (*Physical Downlink Control Channel*): Para o envio de informação de controle como agendamentos dos dados dos canais de PDSCH e de canais de UL.
- Canal de difusão PBCH (*Physical Broadcast Channel*): Usado para a difusão de informações do rádio necessárias pelo UE para estabelecer e manter conexão.
- Canais Físicos de UL
 - Canal compartilhada de UL PUSCH (*Physical Uplink Shared Channel*): Canal de envio de dados do UE para o GnB.
 - Canal de controle de UL PUCCH (*Physical Uplink Control Channel*): Canal de informação de controle o que inclui sinais de confirmação de recepção informação, petições de agendamento de recursos do sistema, e solicitações de informações do estado dos canais de DL.
 - Canal de acesso aleatório físico PRACH (*Physical Random Access Channel*): O PRACH permite que os UE se conectem na GnB sem necessidade de sincronização previa.

3.1.1.6 Sinais Físicos

Toda informação transmitida não proveniente de camadas superiores à camada física é chamada de sinal físico. Como no caso dos canais físicos os sinais físicos podem ser classificados em sinais de DL e UL:

- Sinais Físicos de DL
 - Sinal de referência de demodulação DM-RS (*Demodulation reference signal*, em inglês): Sinal usado para estimação do canal para demodulação. Este sinal é próprio de cada UE, não periódico, e só transmitido quando necessário. Este sinal muda para cenários de baixa ou alta velocidade, para cenários de alta velocidade a densidade do sinal é mais alta, permitindo o rastreamento de mudanças muito rápidas no sinal, para o caso contrário a densidade é baixa.
 - Sinal de referência de sincronização de fase PT-RS (*Phase Tracker Reference Signal*, em inglês): Permite a compensação de ruído de fase na recepção. A natureza da degradação devido ao ruído de fase pode ser

constante para todas as portadoras do símbolo OFDM, por isto o sinal de referência de sincronização PT-RS é projetado para ser disperso no domínio da frequência e denso no domínio do tempo.

- Sinal de referência de informação do estado do canal CSI-RS (*Channel State Information Referency signal*, em inglês): É um sinal projetado para ser usado em diferentes casos. Para aquisição da qualidade do estado do canal CQI (*Channel Quality Indicator*, em inglês), indicadores de classe RI (rank indicator) e parâmetros de *beamforming*, como PMI (*precoding matrix indicator*, em inglês). O CSI-RS também pode ser usado como medida de interferência, e é um sinal de potência zero. Para gerenciamento de feixes quando *beamforming* é usado, o sinal de CSI é usado para avaliar possíveis candidatos de feixes, medindo a potência recebida do sinal. Finalmente, o sinal CSI-RS pode ser usado como sincronização e rastreamento TRS (*Tracking reference signal*, em inglês) e pode ser utilizado para sincronização fina de tempo e frequência.
- Sinais de sincronização Primária e Secundária PSS/SSS (Primary/Secondary Synchronization Signal, em inglês): Os sinais de sincronizações primários e secundários são usadas para sincronização de frame pelo UE e são enviadas em um bloco chamado de SSB em conjunto com o canal de difusão PBCH.
- Sinais Físicos de UL
 - Sinal de referência de som SRS (*Sound Reference Signal*, em inglês): É o sinal usado para extrair informações do canal, quando informação do UE são enviadas até a estação base.
 - Sinal de referência de demodulação DM-RS: o mesmo sinal de referência de Downlink, neste caso para a transmissão de dados do UE até o Gnb.
 - Sinal de referência de sincronização de fase PT-RS: Mesmo propósito do sinal PT-RS no DL, desta vez para a transmissão UL.

Os protocolos de conexão entre UE e gNB estão sendo estudados e as normas 3GPP que descrevem estes protocolos são [24] e [25].

3.1.2 OpenRAN

OpenRAN (Open Radio Access Networks, em inglês) é uma aliança que visa a padronização de redes 3GPP. Através da fragmentação da arquitetura da rede, a OpenRAN desenvolve especificações que permitem aos fabricantes desenvolverem partes individuais, em vez de soluções completas.

O desenvolvimento das especificações ORAN é dividido em grupos de trabalho (WG, *Working Groups*, em inglês):

- WG1 Casos de uso e Arquitetura: Desenvolve requerimentos de sistema e aplicações com o intuito de demonstrar os produtos ao mercado.
- WG2 Controlador de RAN Inteligente em tempo no real (nonRT-RIC, *non-real time* (RT) RAN *Intelligent Controller* (RIC), em inglês): Dentre as tarefas do WG2 estão o controle do gerenciamento do rádio (RRM, *Radio Resource Management*), otimização de diretrizes RAN e aplicações de Inteligência artificial e modelos de aprendizado de máquinas. O nonRT-RIC executa tarefas em laços de controle maiores do que 1 segundo.
- WG3 RIC em tempo quase real (nearRT-RIC, *near Real Time* RIC) e Interface E2: Define as especificações da interface E2, interface que comunica o nearRT-RIC com a unidade de controle central (CU, *Control Unit*, em inglês).
- WG4 Interface de Fronthaul: Define a especificação do protocolo de comunicação entre a unidade distribuída de ORAN (O-DU, *desentralized Unit*, em inglês) e a unidade ORAN de rádio (O-RU, *radio Unit*, em inglês). O *fronthaul* usa Ethernet como meio de transporte e o protocolo eCPRI ou IEEE1914.3 para transportar os pacotes do Plano-C, Plano-U, e de gerenciamento (Plano-M) e os pacotes de sincronização das unidades de Rádio (Plano-S). Este protocolo é descrito em detalhe nas próximas seções deste documento.
- WG5 Interfaces Abertas F1/W1/E1/X2/Xn: Descreve interface O1, interface que realiza conexão entre O-DU e o gerenciador de serviços e orquestração (SMO, *service management and orchestration*, em inglês). Este último define inicialização, configuração como também gerenciamento de software, gerenciamento de performance e de falhas da O-DU.
- WG6 Computação na Nuvem e Orquestração: Este grupo identifica casos nos quais é possível demonstrar as vantagens de migração para computação na nuvem. De maneira mais precisa, o WG6 identifica casos de uso nos quais a migração para software traz benefícios. Também define cenários de implementação, requerimentos e projetos de referência, virtualização de soluções na nuvem.
- WG7 Hardware de caixa branca: Define arquiteturas de referência para a DU e RU. As empresas colaboradoras são principalmente companhias de hardware que desenvolvem soluções e exemplificam como pode ser feita a implementação.
- WG8 Arquitetura de referência de pilha: Igual ao WG7 para arquitetura de software.

- WG9: *X-Haul* Abertos: Define arquiteturas da camada física para as interfaces de *BackHaul* (Conexão O-CU e SMO), *MidHaul* (Conexão CU-DU) e *fronthaul* (DU-RU).

Dado que o objetivo é projetar uma O-RU, o foco principal do estudo se concentrou nos WGs 4 e 7, responsáveis pela interface de *fronthaul* e pelo hardware de referência.

Para compreendermos a fundo a arquitetura da ORU e o protocolo de comunicação entre a O-RU e a O-DU, é fundamental situá-las dentro das especificações do rádio 5G da 3GPP.

3.1.2.1 Divisão Funcional 7.2X

A fragmentação da rede é chamada de divisão funcional (*Functional Split N*, em inglês) e várias propostas têm sido feitas durante o desenvolvimento das especificações de 5G. A Figura 2 - Diagrama da cadeia de processamento 5G e fragmentação[26]**Error! Reference source not found.**, mostra, de modo geral, a cadeia de processamento dos dados para 5G e uma fragmentação da rede em três unidades principais:

- Uma O-CU que concentra o controle de recursos do rádio (RRC, *Radio Resource Control*) e o Protocolo Unificado de Dados por Pacotes (PDCP, *Packet Data Convergence Protocol*, em inglês). A O-CU encontra-se na nuvem e é responsável por tarefas de controle e configuração de tempo não real.
- A O-DU que concentra tarefas das camadas de controle de enlace (*Radio Link Control*, em inglês), (MAC, *Media Access Control* em inglês) e parte do processamento da camada física (PHY, *Physical Layer*, em inglês), também chamada de parte alta da camada física (*High-PHY*, em inglês). Estas três camadas segmentam e reagrupam dados, corrigem erros e realizam a modulação e demodulação dos dados digitalmente.
- Finalmente a O-RU que termina o processo da camada física, a parte baixa (*Low-PHY*, em inglês) que envolve o tratamento dos dados para serem transformados em sinais analógicos.

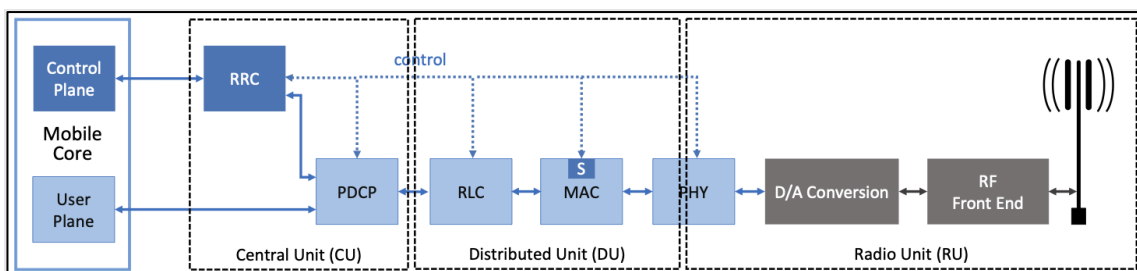


Figura 2 - Diagrama da cadeia de processamento 5G e fragmentação[26]

A ORAN Alliance propõe o Split 7.2X (ORAN ALLIANCE, 2022), com duas variações para o DL, o Split 7.2 categoria A, mostrado na Figura 3, onde a pré-codificação dos dados acontece na O-DU (ORAN, *Decentralized Unit*, em inglês) e a variação B, mostrado na Figura 4 - Descrição do Rádio Split 7.2B Downlink DU-RU[27], onde a pré-codificação é feita no RU.

Na Figura 3 - Descrição do Rádio Split 7.2A Downlink DU-RU[27] e Figura 4 - Descrição do Rádio Split 7.2B Downlink DU-RU[27] é possível também ver uma descrição do processamento feito em cada unidade. Na abordagem 7.2X a unidade de DU inclui todo o processamento de dados como pré-codificação de *beamforming* (No caso do Split 7.2A), codificação de canal, modulação dos dados binários e o embaralhamento dos dados (*Scrambling*, em inglês). Além disso, no caso de DL, a O-DU mapeia os REs nos recursos físicos de frequência e canais de camadas superiores nos canais físicos.

Quando atuando no caso de UL, a O-DU faz o caminho inverso dos dados, demodulando, desmapeando e desembaralhando os dados. Na UL O-DU também são feitas as funções de acondicionamento dos sinais para o processamento correto dos sinais recebidos. Equalização, estimação do canal e processamento dos sinais de PRACH para agendamento dos UEs são processados na UL O-DU.

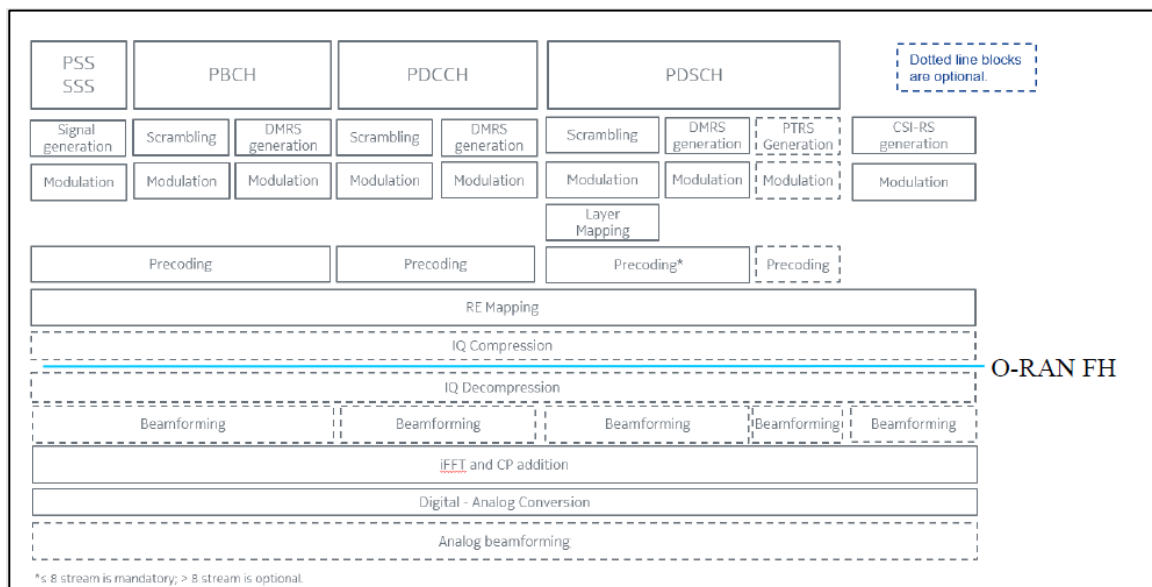


Figura 3 - Descrição do Rádio Split 7.2A Downlink DU-RU[27]

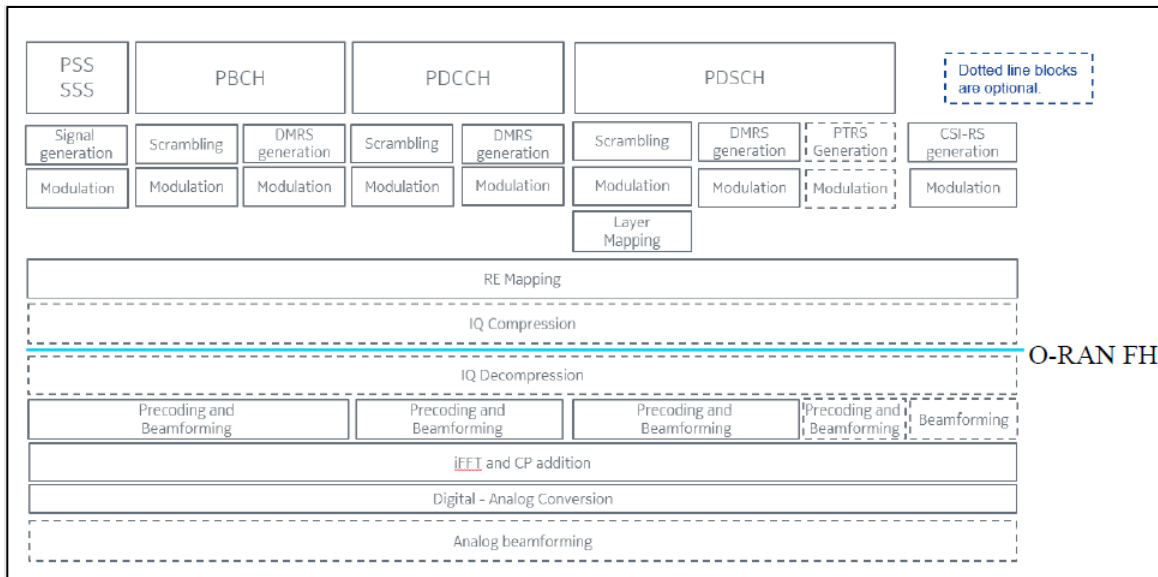


Figura 4 - Descrição do Rádio Split 7.2B Downlink DU-RU[27]

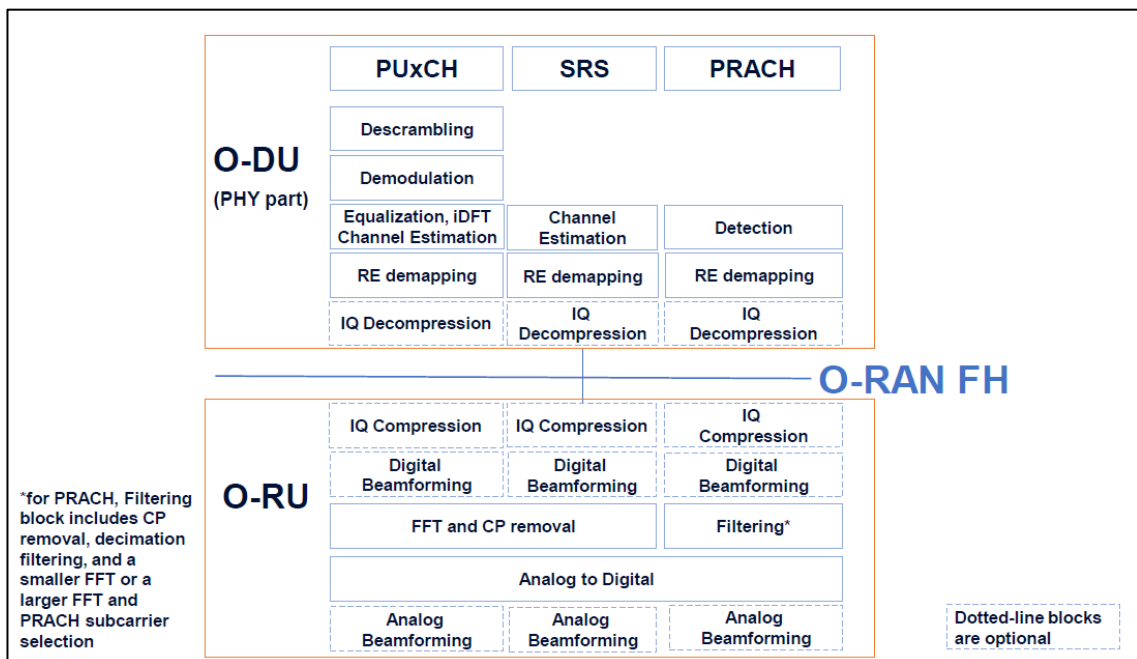


Figura 5 - Descrição do Rádio Split 7.2 UpLink DU-RU[27]

O Split 7.2X simplifica a O-RU para consumir a menor quantidade possível de potência. É na O-RU que é feito o tratamento dos dados e acondicionamento pré-conversão para enviar os dados para a cadeia RF. Como mostra a Figura 5 - Descrição do Rádio Split 7.2 UpLink DU-RU[27], [Error! Reference source not found.](#) no DL O-RU é feita a conversão frequência tempo, o *pre-coding* (no caso da variação B do Split 7.2) e opcionalmente o *beamforming*. Mais detalhes do processamento na O-RU é descrito na seção ORAN-RU.

Uma outra característica do Split 7.2X é o suporte à compressão de dados IQ, o que permite atingir altas velocidades de transmissão. A comunicação entre O-RU e O-DU é especificado pelo chamado fronthaul (FH). A comunicação O-DU O-RU é feita via ethernet, que além de carregar dados dos planos de controle (Plano-C) e plano de usuário (Plano-U) no seu campo de dados, carrega também dados dos planos de sincronização (Plano-S) e dados do plano de gerenciamento (Plano-M). A encapsulação dos planos O-CU é eCPRI ou IEEE 1914.3.

3.1.2.1.1 ORAN-RU

Como descrito anteriormente, o WG7 propõe arquiteturas para O-DU e O-RU que podem ser usadas como referência para implementações. A Figura 6 - Arquitetura de referência de uma O-RU[27], mostra a arquitetura de referência proposta pelo WG7.

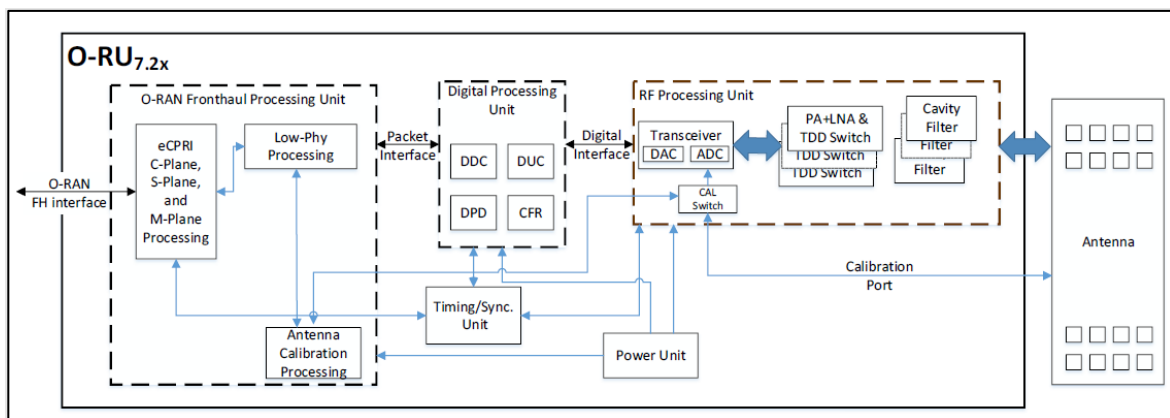


Figura 6 - Arquitetura de referência de uma O-RU[27]

A arquitetura de referência é dividida em três unidades principais:

- A unidade de processamento de FrontHaul

Como descrito em seções anteriores a comunicação O-DU/O-RU é feita pela interface fronthaul. No DL, vários planos estão carregando as informações que são recebidas pela O-RU. Então é necessário como primeiro passo fazer a filtragem dos planos como mostrado na Figura 6 - Arquitetura de referência de uma O-RU[27]. O filtro de mensagens deve extrair as informações necessárias e repassá-las para as outras unidades do sistema. Portanto, devem ser reconhecidos e extraídos os cabeçalhos dos protocolos eCPRI ou IEEE 1914.3, assim como as informações de controle contidas no Plano-C e Plano-M e os dados IQ contidos no Plano-U.

Os dados IQ extraídos do Plano-U no domínio da frequência são processados na subunidade Low-PHY do fronthaul. Como descrito anteriormente o processamento

da Low-PHY no Split 7.2X na O-RU faz a transformação dos dados do domínio da frequência para o domínio do tempo e compõe o símbolo OFDM, a informação de como compor este símbolo é recebida pelo Plano-C. A informação de beamforming¹, também é recebida pelo Plano-C e é processada na subunidade de calibração de antenas. Que repassa esta informação para o circuito de calibração analógico. As informações de sincronização do Plano-S são repassadas para a unidade de sincronização que ajusta os temporizadores e os relógios no sistema.

- Unidade de processamento digital

Os sinais complexos IQ agora no domínio do tempo precisam ser passar por uma reamostragem. No DL os sinais passam pelo conversor digital de frequência ascendente (DUC, *Digital Up-Converter*, em inglês) que aumenta a taxa de amostragem do sinal. No UL os sinais recebidos em taxas altas, precisam ser reamostrados para taxas menores, para eles conseguirem ser processados. O conversor digital de frequência descendente (DDC, *Digital Down-Converter*, em inglês) diminui a taxa de amostragem dos sinais.

Para melhorar a eficiência do amplificador de potência (PA, *Power Amplifier*, em inglês) no circuito analógico de RF, o sinal digital precisa ser previamente condicionado. Duas subunidades atuam no caminho do sinal digital durante a cadeia de DL para otimizá-lo antes da amplificação:

1. Redutor da Razão Pico-Médio (CFR, *Crest Factor Reduction*, em inglês): O CFR diminui a diferença entre o pico e a média do sinal digital, reduzindo o fator de crista. Essa redução limita a amplitude do sinal, evitando distorções e otimizando a linearidade do PA. O CFR contribui para melhorar a eficiência energética do PA, diminuindo o consumo de energia e o calor gerado.
2. Unidade de Pré-distorção Digital (DPD, *digital pre-distortion*, em inglês): A pré-distorção compensa as distorções não lineares introduzidas pelo PA. Essa compensação lineariza o sinal amplificado, aumentando a qualidade do sinal transmitido. A pré-distorção melhora a performance do PA, permitindo maior potência de saída e melhor relação sinal-ruído (SNR).

¹ Dado que o beamforming é opcional ((ORAN Alliance; WG4, 2022)) e não forma parte da primeira versão da O-RU, o seu estudo não foi contemplado nesta etapa do projeto.

- Unidade de processamento de RF

Finalmente, no DL, o sinal digital otimizado é convertido em um sinal analógico e amplificado pelo PA. No UL o sinal recebido passa pelo amplificador de baixo ruído (LNA, *Low Noise Amplifier*, em inglês) e pela filtragem analógica para minimizar ruído não desejado.

A arquitetura mostrada é uma base de referência mínima necessária para uma O-RU funcional, muitas outras unidades de processamento tanto digitais como analógicas podem ser encontradas na implementação final de uma O-RU. A arquitetura final dependerá das adaptações feitas por cada fabricante.

4 Estudos da tecnologia e plataforma de desenvolvimento

O projeto de referência da O-RU fornecido pela AMD é um sistema embarcado no qual hardware e software interagem para executar as funções de uma estação base 5G. Além dos principais componentes do sistema, como o processador e o componente de hardware de lógica programável, o sistema embarcado é composto por diversos periféricos, como uma interface assíncrona universal de transmissão e recepção (UART, *Universal asynchronous receiver/transmitter*, do inglês), ou uma interface periférica serial (SPI, *serial peripheral interface*, do inglês), uma Quad-SPI (QPSI) para fazer interface com memória flash e núcleos de gerenciamento de potência.

A metodologia e os processos de desenvolvimento, assim como as arquiteturas da placa de desenvolvimento, são específicos de cada fabricante, o que torna necessário o estudo das ferramentas de desenvolvimento fornecidas.

Foram feitos treinamentos com foco em algumas áreas específicas para que houvesse um entendimento inicial e familiarização com a plataforma de desenvolvimento selecionada.

4.1 Sistemas Embarcados

Este treinamento mostrou de modo geral e genérico as metodologias de desenvolvimento para um sistema embarcado usando as ferramentas da AMD.

- A primeira etapa do treinamento apresentou uma visão geral do processo de desenvolvimento de hardware, é um passo a passo do fluxo de desenvolvimento passando brevemente pelas ferramentas usadas, fazendo ênfase na geração e integração de modelos já prontos fornecidos pela AMD chamados de modelos de propriedade intelectual (IP, *Intellectual Property*, em inglês).
- Uma segunda parte do treinamento introduziu o software em sistemas embarcados da AMD. Dentre os tópicos do curso foram abordados: uso da ferramenta da AMD Vitis para o gerenciamento de projetos de software embarcado, processo de inicialização do sistema e geração da imagem de inicialização com o sistema operacional.
- O treinamento continuou com as interfaces AXI. Esta interface define um protocolo para comunicar IP-IP e IP – PS. Nesta seção, além da descrição do protocolo, foram mostradas as diferentes variações do AXI. Simulações reforçaram e ajudaram a entender o uso do protocolo.
- A quarta parte do treinamento apresentou algumas das diferentes plataformas de *hardware* disponíveis pela AMD. Dentre elas estão a Zynq UltraScale

MPSoC/RFSoc, dois sistemas em um circuito integrado (SOC, *system on chip*, em inglês), que são as tecnologias usadas na nossa plataforma de desenvolvimento. Foi apresentado também o Microblaze, um processador inteiramente feito em lógica programável assim como o SOC Zynq7000.

- Finalmente, o treinamento encerrou com o aprendizado do uso de depuração com o acionamento cruzado, onde sinais do PL podem disparar ações no PS e vice-versa.

4.2 Sistemas embarcados usando ferramentas Petalinux (Software)

Este treinamento abrange o desenvolvimento de software e o gerenciamento do PS em sistemas embarcados da AMD. O foco principal está no sistema operacional que opera no processador do SOC e nos drivers que comunicam com os periféricos. O conteúdo do curso inclui os seguintes módulos:

- Visão geral do sistema operacional Linux: Abordagem da estrutura do Linux e sua versão para sistemas embarcados.
- Petalinux: Descrição da versão customizada do Linux para os sistemas embarcados da AMD, incluindo o fluxo de desenvolvimento com as ferramentas de software da AMD Vitis.
- Criação de aplicações: Detalhes sobre a criação de aplicações dentro do Petalinux, ferramentas e processos de depuração.
- Inicialização do sistema: o processo de inicialização do Linux.
- Integração do OS com o hardware e o PL: Descreve o processo de integração da lógica digital, periféricos e o sistema operacional.
- Desenvolvimento de drivers: este modulo descreve o processo de desenvolvimento de controladores de hardware como aplicações de software operando dentro do sistema operacional.

5 Análise e Simulações sem placa

5.1 Descrição Geral

Devido à complexidade do sistema, tanto em circuito lógico programável, quanto em tamanho, faz-se a necessidade de estudos específicos em vários âmbitos do projeto. Um estudo teórico para adquirir um domínio técnico e as verificações das conformidades dos requisitos do projeto com os IP's adquiridos que foram desenvolvidos pela AMD é uma das primeiras demandas. O diagrama de blocos ilustrado na Figura 7 - Diagrama simplificado de blocos O-RAN da AMD, evidência os principais IP's, agora módulos internos que compõem todo o sistema O-RAN de forma simplificada. O bloco identificado como ZYNCSS0 compreende um microcontrolador programável com núcleos de processamento da família ARM, formando uma plataforma RFSoc, que controla e configura toda a lógica programável do sistema por meio de uma camada de aplicação. É possível remover o bloco ZYNCSS0 e dividir toda a lógica programável do sistema em duas grandes partes para simplificar o entendimento e as simulações. A primeira parte é responsável pelo controle e processamento de pacotes de redes, enquanto que a segunda pelo processamento de sinais, que troca informações com o meio externo através dos filtros digitais, conversores ADC (*Analog Digital Converter*) e DAC (*Digital Analog Converter*).

Rede Ethernet: ETHSS0, TIMERSS0, CLKSS0 e ORANSS0 e
Processamento de sinais: CHAIN0 (DFE OFDM), RFSS0 e VECSS0

As simulações para o processamento de sinais apenas ocorreram para o módulo interno da CHAIN0 (DFE OFDM).

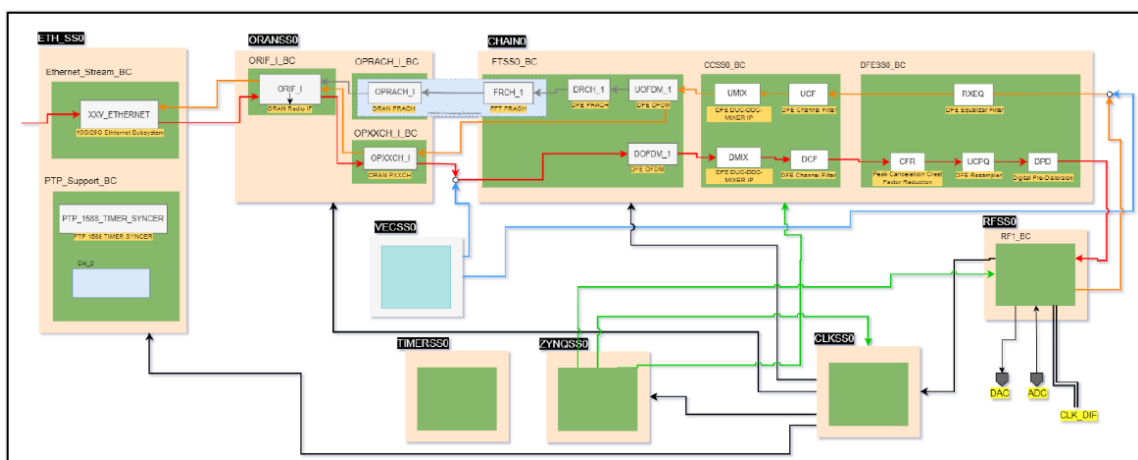


Figura 7 - Diagrama simplificado de blocos O-RAN da AMD

5.2 Estudo dos Requisitos

Com um entendimento inicial e superficial do sistema, uma compreensão mais profunda das funcionalidades principalmente da lógica programável foi necessária. Portanto uma verificação entre as especificações dos blocos com os requisitos do projeto foi elaborada. A Figura 8 - Leitura dos requisitos, em conjunto com a Tabela 4 - Status do Sistema, define como interpretar a Tabela 5 – Conformidade com os requisitos, que mostra quais os requisitos solicitados pelo projeto estão sendo contemplados e a prioridade deles no sistema. Outras informações importantes como as normativas e especificações dos módulos também são reveladas.

			Referência ORAN/3GPP	Referência TRD		
Item	Descrição	Bloco 1	Ref. ORAN/3GPP	Ref. Xilinx TRD	Prioridade	Status
FFT & CP remov e	Aplicar a FFT e remover o prefixo	-	3GPP TS 38.201 [5] 3GPP TS 38.202 [6] O-RAN.WG7 [3]	XAPP1382 OFDMIP [7]	Mandatário	OK
			normativa			

Figura 8 - Leitura dos requisitos

O entendimento da Tabela 5 – Conformidade com os requisitos, segue a interpretação da Figura 8 - Leitura dos requisitos, de leitura dos requisitos e o código de cores mostrado na Tabela 4 - Status do Sistema, dos possíveis status do sistema.

Tabela 4 - Status do Sistema

Status	Significado
OK	O subsistema ou IP do TRD atende todos os requisitos da descrição RNP como também os requisitos ORAN/5G.
NTS	Não tem suporte ou o suporte no TRD é limitado. Estes itens têm que ser desenvolvidos ou propor uma solução
“ICT”	Responsabilidade de outras ICTs, aparecem na tabela porque Eldorado ainda presta suporte para a integração na lógica digital.

Tabela 5 – Conformidade com os requisitos

Item	Descrição	Bloco ¹	Ref. ORAN/3GPP	Ref. AMD TRD	Prioridade	Status
FFT & CP remove	Aplicar a FFT e remover o prefixo cíclico do sinal para o processamento de Uplink	CHAIN0-FTSS0	3GPP TS 38.201 [7] 3GPP TS 38.202 [8]	XAPP1382_OFDM IP [9]	Mandatório	OK
FFT & CP Add	Aplicar a IFFT e adiciona o prefixo cíclico ao sinal para o processamento de Downlink	CHAIN0-FTSS0	3GPP TS 38.201 [7] 3GPP TS 38.202 [8]	XAPP1382_OFDM IP [9]	Mandatório	OK
Front End PRACH	Processamento Front End (canal de acesso aleatório físico)	CHAIN0-FTSS0	3GPP TS 38.211 [10]	XAPP1386_PRACTH SUBS [11]	Mandatório	OK
DDC	Conversor de descida (down) digital de componentes de portadora.	CHAIN0-CCSS0	O-RAN.WG7.OMAC-HRD [3]	PG393 DFE DCX_MIXER [12]	Mandatório	OK
DUC	Conversor de subida (up) digital de componentes de portadora.	CHAIN0-CCSS0	O-RAN.WG7.OMAC-HRD [3]	PG393 DFE DCX_MIXER [12]	Mandatório	OK
DPD	Pré-distorção digital.	CHAIN0-DFESS0	O-RAN.WG7.OMAC-HRD [3]	PG076 DPD IP [13]	Opcional	inatel
CFR	Redução do fator de crista (pico).	CHAIN0-DFESS0	O-RAN.WG7.OMAC-HRD [3]	PG097 CFR IP [14]	Opcional	inatel
Planos C/U/M	Processamento do plano de controle, usuário e gerenciamento.	ORANS0-ORAN_I	O-RAN.WG4.CUS [2]	PG370 ORANIF [15]	Mandatório	OK
Protocolo 1588	Protocolo de sincronização IEEE 1588.	ETHSS0-ETHPTPSUBS	IEEE1588PTPProtocol [16]	PG210 Ethernet Subs - PTP 1588 Timer Syncer [17]	Mandatório	OK
eCPRI	Evolved Common Public Radio Interface.	ORANS0-ORAN_I	ECPRI Standard [18] O-RAN.WG4.CUS [2]	PG370 ORANIF [15]	Mandatório	OK
Compressão de dados QI	Compressão/descompressão de dados QI para redução do tráfego na interface fronthaul.	ORANS0-ORAN_I	O-RAN.WG4.CUS [2]	PG370 ORANIF [15]	Opcional	OK
Beamforming	Processo de conformação de feixes.				Opcional	NTS
Pré-codificação	Pré-codificação	ORANS0-ORAN_I	O-RAN.WG4.CUS [2]		Opcional	Obs ²
Mapeamento de REs	Mapeamento dos elementos de recurso de rádio.	CHAIN0-FTSS0		XAPP1382_OFDM IP [9]	Opcional	OK

(1) Faz referência ao diagrama de blocos da Figura 1 - Matrix de Alocação de recursos

(2) Apesar de não ter a função de *precoding* interna, a O-RU é capaz de processar os Extensions Type 3 e disponibilizar as informações relevantes para um módulo de *precoding* externo, que eventualmente pode ser implementado.

O estudo dos requisitos em conjunto com as especificações foi importante e conclusivo, uma vez indicou por meio da documentação poder atender as condições de tráfego de uma rede O-RAN. Alguns requisitos são opcionais, portanto, a rede deve funcionar normalmente. Como por exemplo na compressão/descompressão de dados que é opcional para a rede O-RAN mais foi implementado. E no caso do *beamforming* que é também opcional e não foi implementada pela AMD não será problema.

Para o caso da necessidade de implementar ou modificar um modulo e integrá-lo ao sistema, existe a demanda de um entendimento de como cada modulo do sistema está trabalhando. Portanto um estudo sobre a norma O-RAN.WG4 [2] foi necessária, pois os protocolos de comunicação dos planos usuário e de controle são detalhados e seu entendimento é de vital importância para começar compreender o que cada modulo tem a incumbência de fazer, para assim chegar ao funcionamento do sistema como um todo.

5.2.1 Plano do Usuário

O primeiro protocolo a ser compreendido foi o *plano do usuário (plano u)* devido a sua peculiaridade de ser responsável por carregar os dados úteis, amostras em sua estrutura. A Figura 9 - Campos que compõem o protocolo do Plano U, mostra os campos presentes no protocolo do Plano-U, cada cor destaca, os cabeçalhos, que trazem informações para qualificar e quantificar a informação do sistema de forma adequada. Como existem campos que são opcionais, os tamanhos em octetos de K e M variam, gerando tamanhos diferentes de elemento de recursos (REs).

(msb)	1	2	3	4	5	6	(lsb)	# of bytes	
CABEÇALHO DA CAMADA DE TRANSPORTE								8	Octeto 1
dataDirection	payloadVersion			filterIndex				1	Octeto 9
subframeId			frameId		slotId			1	Octeto 10
slotId			startSymbolId					1	Octeto 11
sectionId			rb		symInc		startPrbu	1	Octeto 13
startPrbu			numPrbu		udCompHdr (cabeçalho de compactação de dados do usuário) (nem sempre presente)			0/1	Octeto 14
reserved (nem sempre presente)			udCompLen (nem sempre presente)		udCompParam (nem sempre presente)			0/1	Octeto 15
iSample (1st RE in the PRB)			qSample (1st RE in the PRB)		iSample (12th RE in the PRB)			0/2	Octeto 16
qSample (1st RE in the PRB)			iSample (12th RE in the PRB)		qSample (12th RE in the PRB)			0/1/2	Octeto 17/19/21
iSample (12th RE in the PRB)			qSample (12th RE in the PRB)		udCompParam (nem sempre presente)			1	K= 17/19/20/21/23
qSample (12th RE in the PRB)			iSample (1st RE in the PRB)		qSample (1st RE in the PRB)			1	K+1
iSample (12th RE in the PRB)			qSample (12th RE in the PRB)		udCompParam (nem sempre presente)			1	K+22
qSample (12th RE in the PRB)			iSample (1st RE in the PRB)		qSample (1st RE in the PRB)			1	K+23
udCompParam (nem sempre presente)			iSample (1st RE in the PRB)		qSample (1st RE in the PRB)			0/1/2	K+24
iSample (1st RE in the PRB)			qSample (1st RE in the PRB)		iSample (12th RE in the PRB)			1	K+24/25/26
qSample (1st RE in the PRB)			iSample (12th RE in the PRB)		qSample (12th RE in the PRB)			1	K+25/26/27
iSample (12th RE in the PRB)			qSample (12th RE in the PRB)		iSample (12th RE in the PRB)			1	K+46/47/48
qSample (12th RE in the PRB)			iSample (12th RE in the PRB)		qSample (12th RE in the PRB)			1	K+47/48/49
sectionId			rb		symInc		STARTPRBU	1	Octet M
startPrbu			numPrbu		udCompHdr (nem sempre presente)			1	M + 1
reserved (nem sempre presente)			udCompLen (nem sempre presente)		udCompParam (nem sempre presente)			0/1	M + 2
iSample (1st RE in the PRB)			qSample (1st RE in the PRB)		iSample (12th RE in the PRB)			0/2	M + 3
qSample (1st RE in the PRB)			iSample (12th RE in the PRB)		qSample (12th RE in the PRB)			0/1/2	M + 4
iSample (12th RE in the PRB)			qSample (12th RE in the PRB)		udCompParam (nem sempre presente)			1	M + 5
qSample (12th RE in the PRB)			iSample (1st RE in the PRB)		qSample (1st RE in the PRB)			0/1/2	M+4/6
udCompParam (nem sempre presente)			iSample (1st RE in the PRB)		qSample (1st RE in the PRB)			1	M+4/6/8
iSample (1st RE in the PRB)			qSample (1st RE in the PRB)		iSample (12th RE in the PRB)			1	K=M+4/6/7/8/10
qSample (1st RE in the PRB)			iSample (12th RE in the PRB)		qSample (12th RE in the PRB)			1	K+1
iSample (12th RE in the PRB)			qSample (12th RE in the PRB)		udCompParam (nem sempre presente)			1	K+22
qSample (12th RE in the PRB)			iSample (1st RE in the PRB)		qSample (1st RE in the PRB)			1	K+23
udCompParam (nem sempre presente)			iSample (12th RE in the PRB)		qSample (12th RE in the PRB)			0/1/2	K+24
iSample (1st RE in the PRB)			qSample (1st RE in the PRB)		iSample (12th RE in the PRB)			1	K+24/25/26
qSample (1st RE in the PRB)			iSample (12th RE in the PRB)		qSample (12th RE in the PRB)			1	K+25/26/27
iSample (12th RE in the PRB)			qSample (12th RE in the PRB)		iSample (12th RE in the PRB)			1	K+46/47/48
qSample (12th RE in the PRB)			iSample (12th RE in the PRB)		qSample (12th RE in the PRB)			1	K+47/48/49

OBS 1: Sombreamento: amarelo é o cabeçalho de transporte, verde escuro é o cabeçalho da aplicação de rádio.

OBS 2: Contagem de octetos fornecidas quando iqWidth = 8 mas outros valores de iqWidth são possíveis; a contagem de octetos assume que todos os REs estão presentes, o que talvez não seja o caso quando o campo sREsMask esta presente em udCompParam.

Figura 9 - Campos que compõem o protocolo do Plano U

A tecnologia O-RAN permite o uso de dois cabeçalhos de transporte diferentes dentro do pacote Ethernet, com o intuito de detalhar o tratamento dos dados da aplicação nos planos *Control* e *User*. Cada cabeçalho tem 8 bytes e oferece recursos básicos de roteamento de dados, como:

- Tipo de fluxo de dados.
- Identificadores de portas de envio e recepção.
- Capacidade de concatenar várias mensagens da aplicação em um único pacote Ethernet.
- Numeração sequencial.

Os possíveis cabeçalhos de transportes utilizados são o eCPRI [18] e o IEEE 1914.3 [19].

O cabeçalho de transporte eCPRI, mostrado na Tabela 6 – Definições do cabeçalho de transporte eCPRI, tem internamente campos com tamanhos diferentes. Os campos ecpriRtcid/ecpriPcid são subdivididos em subcampos ocupando dois octetos assim como acontece como o ecpriSeqid e entre outros.

Tabela 6 – Definições do cabeçalho de transporte eCPRI

Section Type: any											
(msb)	1	2	3	4	5	6	(lsb)	# de bytes			
ecpriVersion			ecpriReserved			ecpriConcat Enation		1	Octeto 1		
ecpriMessage											
ecpriPayload											
ecpriPayload											
ecpriRtcid / ecpriPcid											
DU_Port_ID				BandSector_ID						1	Octeto 5
CC_ID			RU_Port_ID							1	Octeto 6
ecpriSeqid											
Sequence ID											
E bit	Subsequence ID								1	Octeto 8	

Os campos do cabeçalho de transporte eCPRI com uso prático, podem ser descrito resumidamente como:

- **EcpriVersion:** Este campo indica a versão do protocolo eCPRI, pode ter o valor em binário 0001 para as versões 1.0, 1.1, 1.2 e 2.0 e o valor 0000 acima de 2.0.
- **EcpriConcatenation:** indica quando se esta utilizando o concatenação para permitir multiplas mensagem eCPRI em um simples Ethernet *payload*.
- **EcpriMessage:** Indica o tipo de serviço transmitido pelo tipo de mensagem. Atualmente apenas os valores 00000000, 00000010 e 00000101 em binários são usados, indicando uma mensagem IQ, uma mensagem de controle de tempo real ou uma mensagem de medição de atraso de rede de transporte respectivamente.
- **ecpriPayload:** É o tamanho em bytes da parte útil da mensagem eCPRI correspondente. Não incluindo quaisquer bytes da parte de preenchimento após a mensagem eCPRI. O tamanho máximo suportado é $2^{16} - 1$, mas o tamanho real ainda pode ser limitado pelo tamanho máximo de dados útil da rede de transporte subjacente.
- **ecpriRtcid/ecpriPcid:** São subcampos que definem um parâmetro identificador eAxC (eAxC_ID: Identificador de fluxo de dados em redes O-RAN) que identifica o fluxo de dados específico, associado a cada Mensagem Plano-C (ecpriRtcid) ou Plano-U (ecpriPcid).
- **EcpriSeqid:** Subcampos que definem como identificar e ordenar mensagens eAxC em diferentes níveis, para garantir a entrega correta e na ordem certa, mesmo em cenários com fragmentação.

O cabeçalho de transporte IEEE 1914.3 detalhado na Tabela 7 – Definições do cabeçalho de transporte IEEE 1914.3 é uma alternativa ao cabeçalho eCPRI, então alguns campos

como o RoOrderInfo são equivalentes ao ecpriRtcid/ecpriPcid e o ecpriSeqid juntos. Havendo uma compatibilidade de função entre eles, portanto já explicada nos campos similares referentes ao cabeçalho de transporte do eCPRI.

Tabela 7 – Definições do cabeçalho de transporte IEEE 1914.3

Section Type: any									
(msb)	1	2	3	4	5	6	(lsb)	# of bytes	
RoEsubType								1	Octeto 1
RoEflowId								1	Octeto 2
RoElength								1	Octeto 3
RoElength								1	Octeto 4
RoOrderInfo								4	Octeto 4/5/6/7
DU_Port_ID				BandSector_ID				1	Octeto 4
CC_ID				RU_Port_ID				1	Octeto 5
Sequence ID								1	Octeto 6
E bit	Subsequence ID							1	Octeto 7

Os campos do cabeçalho de transporte IEEE 1914.3 com uso prático e não explicado ainda são apenas 2.

- **RoEsubType:** Define o tipo de dado útil dentro da faixa de subtipos RoE (Radio over Ethernet) do padrão IEEE 1914.3 para encapsulamentos e mapeamentos RoE. Eles podem ser 128 para Plano-U ou 130 para Plano-C ambos sem concatenação ou 129 e 131 para Plano-U e Plano-C com concatenação respectivamente. Outros valores são ignorados ou não mapeados para o O-RAN.
- **RoElength:** É o tamanho em bytes da parte útil dos dados da mensagem. O valor do campo do tamanho dos dados útil deve ser o número total de octetos seguindo o cabeçalho comum O-RAN. Não inclui o Ethernet FCS(4 Bytes) ou os seguintes bytes.

O cabeçalho de aplicação é composto principalmente pelo campo sectionId que serve para relacionar partes específicas de dados do Plano-U com as definições correspondentes no Plano-C. Ele permite lidar com situações em que uma seção de dados do plano U pode conter informações combinadas de diferentes tipos de dados do plano C.

- **dataDirection:** Direção do dado UL = 1 ou DL = 0.
- **FilterVersion:** Este parâmetro define um índice para o filtro do canal a ser utilizado entre os dados IQ e a interface aérea, tanto em DL e UL.
- **frameId:** É um contador de 0 até 256 ciclos para gerar 10ms de intervalos.
- **SubframeId:** Este campo é um contador para subintervalos de 1 ms em um intervalo de 10 ms.

- **SlotId:** É o número do slot dentro de um subframe de 1 ms. Todos os slots em um subframe serão contados por este parâmetro, de 0 a $N^{\text{slot}} - 1$. No presente documento, o N slot máximo = 16, todos os outros valores dos 6 bits não são usados.
- **StartSymbolId:** Identifica o número mais antigo do símbolo (dentro de um slot), ao qual o as informações desta mensagem são aplicáveis.
- **SectionId:** Indica o número de descrições de seções de dados (citações separadas do ID da seção, mesmo para citações múltiplas do mesmo sectionId) incluídas nesta mensagem do Plano-C.
- **Rb:** Este campo Indica se cada RB é usado ou todos os outros RB são usados. O RB inicial é definido por startPrbc e o número total de RBs usados é definido por numPrbc. Por exemplo se RB=1, startPrbc=1, numPrbc=3, então os PRBs usados são 1, 3 e 5.
- **SymInc:** Este campo define como especificar quais símbolos de dados estão associados a cada seção dentro de uma mensagem Plano-C, permitindo o manuseio de vários símbolos em uma única mensagem. Enquanto o Plano-C permite flexibilidade para lidar com vários símbolos, o Plano-U *foca na transmissão de dados* de um único símbolo por mensagem, tornando o uso de symInc=1 desnecessário e potencialmente prejudicial.
- **StartPrbc:** Normalmente o startPrbc transmite o primeiro PRB (frequência mais baixa) descrito na descrição da seção.
- **StartPrbu:** Representa o primeiro PRB (frequência mais baixa) da seção de dados, serve para garantir a alocação eficiente de recursos de frequência para o Plano-U, evitando sobreposição e garantindo a correspondência entre as seções Plano-C e Plano-U. Ele também considera diferentes métodos de acoplamento e cenários de fragmentação.
- **NumPrbu:** Representa o número total de PRBs (blocos de 12 elementos de recursos consecutivos em frequência) no Plano-U na seção de dados. A Idea também é a alocação eficiente de recursos de frequência para o Plano-U evitando sobreposição e garantindo a correspondência entre as seções Plano-C e Plano-U. O valor de rb determina se os PRBs são consecutivos ou alternados em frequência:

rb = 0: Os PRBs serão numerados consecutivamente.

rb = 1: Os PRBs serão alternados, determinado no momento.

- **UdCompHdr:** Define o método de compressão o que habilita a otimização de recursos e a adaptação dinâmica da largura de bits IQ em seções de dados Plano-U, tanto no *Downlink* quanto no *Uplink*, com o controle adequado via Plano-C ou Plano-M.
- **UdCompParam:** Este campo se aplica a qualquer método de compactação usado para o PRB (configurado estaticamente via M-Plano ou especificado no campo **udCompHdr** da seção de dados que contém o PRB).
- **UdCompLen:** Especifica o número total de octetos na seção atual. Isso inclui preenchimento de bits (realizado por PRB após a última parte da última amostra de QI do PRB) em todos os PRBs na seção atual. O comprimento máximo do campo PRB suportado é $2^{16} - 1$, mas o tamanho real pode ser ainda mais limitado pelo tamanho máximo de tráfego de dados da rede de transporte subjacente.

5.2.2 Plano de Controle

Alguns campos do plano de controle Plano-C são os mesmos utilizados no Plano-U, estes campos que são comuns a ambos no protocolo, carregam as informações pertinentes a comunicação do rádio, enquanto os campos próprios do Plano-U carregam os dados úteis a informação do dispositivo a ser transmitida na comunicação. O Plano-C tem vários formatos de protocolo que é identificado pelo campo Section Type. Existem 9 tipos de formato, sendo que o 2 e o 9 são reservadas para futuros uso. Os Section Type 4, 5, 6, 7 e 8 não são obrigatórios e não são suportados. As Section Type 0, 1 e 3 são as únicas suportadas pelo sistema O-RAN utilizado, as outras possibilidades são ignoradas. A Tabela 8 – Campos que compõem o protocolo do plano de controle é a descrição de um protocolo Section Type 1, o qual é visível os campos similares ao Plano-U, cabeçalho de transporte de aplicação de rádio com os campos do sectionId.

Tabela 8 – Campos que compõem o protocolo do plano de controle

(msb)	1	2	3	4	5	6	(lsb)	# de bytes	
CABEÇALHO DA CAMADA DE TRANSPORTE								8	Octeto 1
dataDirection	payloadVersion			filterIndex				1	Octeto 9
frameId								1	Octeto 10
subframeId				slotId				1	Octeto 11
slotId		startSymbolId						1	Octeto 12

numberOfsections				1	Octeto 13
SectionType = 1				1	Octeto 14
udCompHdr (cabeçalho de compactação de dados do usuário) (nem sempre presente)				1	Octeto 15
reserved (nem sempre presente)				1	Octeto 16
sectionId				1	Octeto 17
sectionId	rb	symInc	startPrbu	1	Octeto 18
startPrbc				1	Octeto 19
numPrbc				1	Octeto 20
ReMask[11:4]				1	Octeto 21
ReMask[3:0]		numSymbol		1	Octeto 22
ef	BeamId[14:8]			1	Octeto 23
BeamId[7:0]				1	Octeto 24
sectionId				1	Octeto N
sectionId	rb	symInc	startPrbu	1	N+1
startPrbc				1	N+2
numPrbc				1	N+3
ReMask[11:4]				1	N+4
ReMask[3:0]		numSymbol		1	N+5

Os campos pertinentes a ambos já descritos, não serão referenciados, apenas os com utilização prática na aplicação O-RAN. Na camada de aplicação teremos:

- **filterIndex:** Define um índice para o filtro do canal a ser utilizado entre os dados IQ e a interface aérea, tanto em DL e UL. O index pode ser escolhido pela DU para a RU, Porém, não é obrigatório comandar filtros especiais, e o índice de filtro = 0000 em binário também é permitido para PRACH.
- **NumberOfsections:** Indica o número de descrições de seções de dados (citações separadas do ID da seção, mesmo para citações múltiplas da mesma sectionId) incluídas nesta mensagem do Plano-C.
- **SectionType:** Determina as características dos dados do Plano-U a serem transferidos ou recebidos de um feixe com um ID de padrão.

- **ReMask:** Define a máscara do RE (Resource Element) dentro de um PRB. Cada configuração de bit no reMask indica se o controle de seção é aplicável ao RE enviado em mensagens Plano-U (0 = não aplicável, 1 = aplicável). O bit mais significativo indica o valor do RE de menor frequência em um PRB. O número máximo de valores reMask diferentes que podem ser aplicados a um PRB é uma característica da O-RU que é transportado do O-RU para o O-DU através do M-Plane. Se qualquer RE em um PRB nunca for apontado por um reMask (mas por outros REs nesse PRB são), o RE "ausente" deve ser definido como zero no Plano-U, e nenhum ID de formação de feixe ou outro processamento deve ser aplicado ao RE "ausente". Um RE não deve ser referenciado mais de uma vez em uma seção de dados.
- **ef:** Incumbido de indicar a presença de parâmetros adicionais que se aplicam à seção, além daqueles incluídos no cabeçalho principal. Esses parâmetros adicionais são fornecidos através de extensões de seção.

extType: Campo de 7 bits que define o tipo específico da extensão.

extLen: Campo de 1 ou 2 bytes que indica o número de palavras de 4 bytes contidas na extensão (mínimo de 1, incluindo os próprios campos "extType" e "extLen").

O objetivo dessa abordagem é permitir a inclusão de novos parâmetros em seções de dados sem precisar redefinir completamente o cabeçalho da seção ou criar novos tipos de seção para cada necessidade específica. Em resumo, o ef (extension flag) fornece uma maneira flexível de estender as seções de dados do Plano-C para acomodar novos requisitos das especificações de fronthaul no futuro.

- **BeamId:** Define o padrão *beam* a ser aplicado aos dados do Plano-U. Como o sistema não tem a implementação do *beamforming*, seu valor é sempre, beamId = 0. Portanto, o sinal é transmitido igualmente por todas as antenas.

6 Avaliação do Comportamento da Lógica Programável em Simulação

Após a fase de estudos para compreensão de como o protocolo e as especificações dos módulos do sistema funcionam. Uma abordagem ainda teórica, porém, mais prática foi adotada. Com simulação, partes específicas do sistema foram verificadas, utilizando o Open IP Example Design sistema fornecido pela AMD no software vivado. A Figura 10 – Esquemático Open RAN IP Example Design, o esquemático do Open IP Example Design fornecido pela AMD para entendimentos do funcionamento da O-RU através de verificação. O bloco ORIF_I_0_block, que corresponde ao ORANSS0, é módulo mais importante e é excitado e monitorado por outros módulos periféricos. Sendo o torwave_0 incumbido de enviar os dados, lidos de uma memória, para o ORIF_I_0_block e o modulo roe_radio_top_0 de monitorar a saída e verificar a sanidade dos dados. As diretrizes são enviadas para todos os blocos configuráveis, por meio do modulo jtag_axi e distribuída pelo axi4_connect. Cada modulo recebe suas configurações por meio de seu endereço. Os blocos clk_wlz_0 e reset_i criam o clock (relógio) e o resete do sistema de forma sincronizada, enquanto o radio_star_recover_v_0 é responsável pela temporização da janela de tempo do protocolo O-RAN.

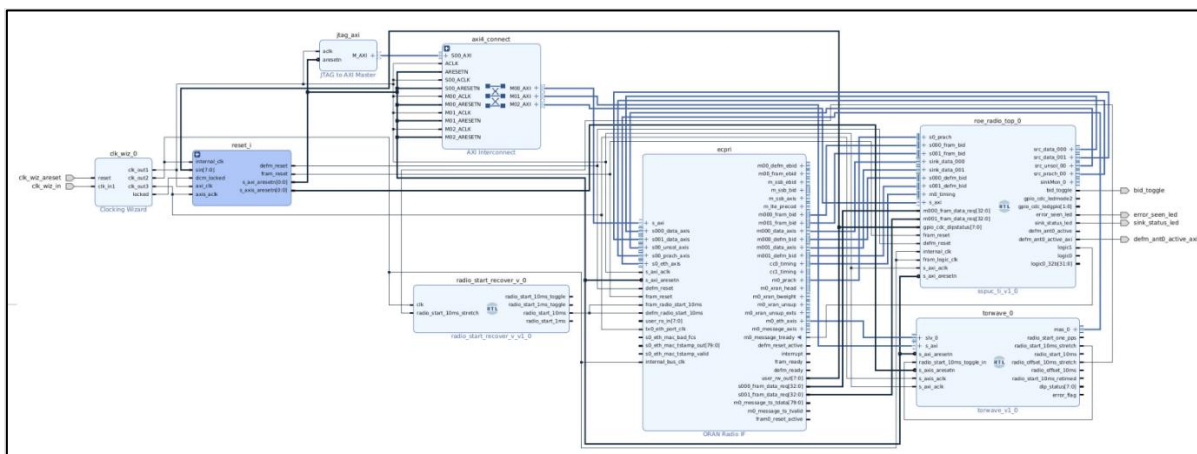


Figura 10 – Esquemático Open RAN IP Example Design

A verificação disponibilizada teve os parâmetros de configuração alterados para funcionar com a taxa de 25G bits/s. Dentro do modulo torwave_0 existem memórias onde estão salvas as informações dos protocolos que são enviadas para o ORIF_I_0_block. A Tabela 9 – Pacote Plano-C gerado pela Open IP Example, mostra os valores de uma parte do primeiro pacote do Plano-C, criado pelo Open IP Example Design e reutilizados nas verificações. Os valores em hexadecimal do pacote Plano-C e seus respectivos significados estão representados lado a lado.

Tabela 9 – Pacote Plano-C gerado pela Open IP Example

(msb)	1	2	3	4	5	6	(lsb)	Valores práticos
CABEÇALHO DA CAMADA DE TRANSPORTE								FE AE
ecpriVersion				ecpriReserved		ecpriConcat Enation		10
ecpriMessage								02
ecpriPayload								00
ecpriPayload								5C
DU_Port_ID				BandSector_ID				00
CC_ID				RU_Port_ID				00
Sequence ID								00
E bit	Subsequence ID							80
dataDirection	payloadVersion			filterIndex				90
frameId								00
subframeId				slotId				00
slotId	startSymbolId							00
numberOfsections								0A
SctionType = 1								01
udCompHdr (cabeçalho de compactação de dados do usuário) (nem sempre presente)								00
reserved (nem sempre presente)								00
sectionId								00
sectionId			rb	symlnc	startPrbu			00
startPrbc								00
numPrbc								02
ReMask[11:4]								FF
ReMask[3:0]				numSymbol				F7
ef	BeamId[14:8]							01
BeamId[7:0]								01
sectionId								00
sectionId			rb	symlnc	startPrbu			10
startPrbc								02
numPrbc								02
ReMask[11:4]								FF
ReMask[3:0]				numSymbol				F7
ef	BeamId[14:8]							01
BeamId[7:0]								01

Os frames são enviados sequencialmente, um frame Plano-C seguidos por vários frames Plano-U, dependendo da quantidade de amostras IQ a serem enviadas. Entre os valores do pacote pode se destacar o FEAE do cabeçalho eCPRI da camada de transporte o campo SctionType = 1 e o ef = 0 avisando que não existe adicionais parâmetros.

A simulação foi realizada com dados realísticos gerados pelo software da Keysight OpenRAN Studio e salvos no formato PCAP. A partir deste formato o software Wireshark converte o conteúdo para arquivos hexadecimais com a extensão .mem, que é gravada na memória do bloco de teste torwave_0. Então o sistema de teste envia os dados no protocolo AXI4-Stream que é reconhecido pelo modulo ORIF_I_0_block. O resultado da simulação obtidos pelo simulador podem ser vistos nos sinais mostrados na Figura 11 – Simulação gerada com os dados da Keysight Open RAN Studio equipamento emulando uma O-DU, onde é visível o reconhecimento dos cabeçalhos pelo sinal **m0_radio_app_head_valid** e dentro da janela de comunicação **m0_packet_in_window** conforme o exemplo disponibilizado na página da AMD [22].

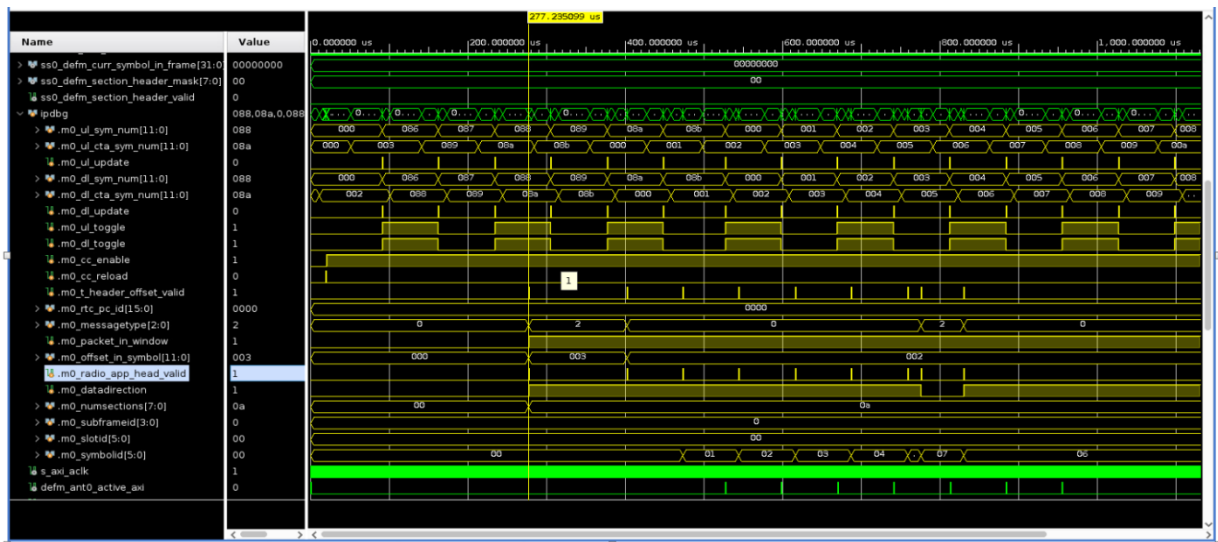


Figura 11 – Simulação gerada com os dados da Keysight Open RAN Studio equipamento emulando uma O-DU

O sinal **m0_radio_app_head_valid** indica que um cabeçalho válido foi detectado pelo IP ORAN. Portanto os pacotes chegaram na janela de recepção. O sinal **m0_packet_in_window** indica se os pacotes chegaram ou não dentro da janela de recepção. Se **m0_packet_in_window** não estiver em nível lógico zero, significa que há um problema de tempo nesta simulação. E seria necessário alterar o horário de chegada da mensagem do Plano-C / Plano-U para corrigir isso.

As verificações validaram algumas configurações que estavam previstas pelos estudos realizados nas especificações. Estas validações foram realizadas com diferentes pacotes, gerados no Keysight Open RAN Studio para excitar o sistema.

- Pacotes Plano-C / Plano-U sem VLAN, dados uteis IQ de 16 bits sem compressão;
- Pacotes Plano-C / Plano-U com VLAN, dados uteis IQ de 16 bits sem compressão;

- Pacotes Plano-C / Plano-U com VLAN, dados uteis IQ de 14 bits sem compressão;
- Pacotes Plano-C / Plano-U com VLAN, dados uteis IQ de 14 bits com compressão.

6.1 Simulação Rede Ethernet

A próxima etapa foi simular a lógica programável, mas devido ao problema da sua complexidade, se faz necessário um microcontrolador para configurar todos os blocos presentes no sistema. Um microcontrolado com um software embarcado funcionando na simulação é inviável. Então a solução foi utilizar a divisão proposta inicialmente, separando a parte do processamento de sinal da parte de rede Ethernet. Com esta nova abordagem é possível configurar os poucos módulos por meio de um *Testebench* e de forma inovadora na metodologia utilizar o software da Cadence SimVision para as verificações. Uma vez que devido à complexidade e do tamanho da lógica programável resulte em simulações demoradas, o SimVision consegue ter uma melhor agilidade. A Figura 12 – Subsistema rede Ethernet, mostra os módulos que são responsáveis pelo subsistema chamado rede Ethernet, os módulos também são compostos por vários submódulos, formando uma hierarquia.

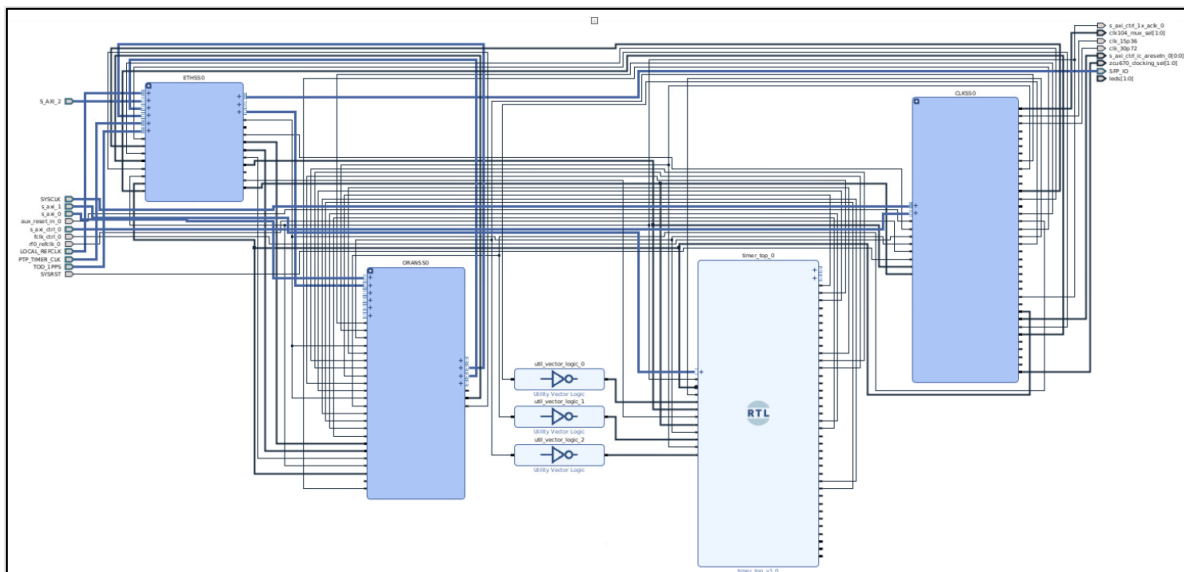


Figura 12 – Subsistema rede Ethernet

Os principais módulos, do subsistema, que compreende a rede Ethernet tem como algumas características técnicas:

- **ETHSS0:** Projetado para atender aos requisitos de Ethernet de 10/25 Gb/s especificados pelas normas IEEE 802.3 Clausula 49 [20], IEEE 802.3by [21] pelo

Consórcio Ethernet 25G. Responsável pela marcação de data/hora pela norma IEEE 1588[16].

- **ORANSS0**: Este modulo implementa apenas a função O-RU (unidade de rádio O-RAN) e tem como configuração:
 1. Duas portas Ethernet de 25 Gb/s.
 2. Duas portadoras (CCs), com temporizadores independentes de *Uplink* e *Downlink*.
 3. Suporta resolução de até um bloco de recursos (RB) por mensagem de seção.
 4. Suporte Plano-U para até 20 streams espaciais para *Downlink*, 16 para *Uplink*, um fluxo adicional separado para o tráfego de SSB (PSS, SSS, PBCH), PRACH e quatro fluxos de dados não solicitados compartilhados (para uso em SRS).
 5. Uma porta de dados PRACH *Uplink*.
 6. Implementa a interface de *fronthaul* de rádio, gerenciando os planos de controle e usuário, implementando os avanços de temporização apropriados e compensando as variações de atraso dos pacotes provenientes de diferentes unidades de banda base (BBUs).
 7. Suporta eCPRI e 1914.3 sobre pacotes Ethernet IEEE 802.1 padrão, opcionalmente incluindo adendos VLAN, bem como UDP sobre IPv4 ou IPv6.
 8. Cada campo do cabeçalho Ethernet e IP/UDP programável.
 9. Alinhamento a um pulso externo de 10 ms de Início de Frame de Rádio, habilitando a sincronização IEEE 1588.
- **CLKSS0**: Tem a funcionalidade de receber os principais *clocks* de referência em conjunto com resetes e redistribui-los de forma síncrona para todo o sistema novamente. Ele faz isto por meio de uso de PLL's ou MMCM's. Portanto este modulo tem a função de ser um relógio do sistema para gerar e distribuir uma árvore de *clock* e resete para o os módulos síncronos.
- **TIMER_TOP(TIMERSS0)**: Modulo temporizador responsável em gerar pulsos de 10 ms de forma independentes de *Uplink* e *Downlink*.

Cada modulo tem seus respectivos registros de configuração que são configurados por meio do protocolo AXI4-Lite, portanto cada modulo tem um “driver” individual com os endereços e conteúdos específicos para obter a configuração adequada. Os módulos gerenciais ETHSS0 e ORANSS0 são supridos de sincronização por meio dos módulos, TIMER e CLKSS0. O CLKSS0 fornece o sincronismo para a lógica do sistema funcionar e trocar informações de forma harmoniosa enquanto o Timer está relacionado com os sincronismos da normativa de Comunicação. Em sistemas de comunicação sem fio, como o O-RAN, existem normativas de comunicação que definem a estrutura das mensagens e a sincronização entre transmissores e receptores. O Timer está envolvido em informações sobre a duração das janelas de recepção e como os dispositivos ajustam sua sincronização.

O subsistema é instanciado formando um DUT (*Design Under Test*) o qual é composto de “drivers”, que individualmente configura cada modulo, através dos seus respectivos registros. A Figura 13 - Ambiente de Simulação, ilustra superficialmente a ideia do

Testbench, onde os estímulos são configurações e amostras enviadas. As amostras são as mesmas utilizadas no Open IP Example Design da AMD da Figura 10, enquanto as configurações são enviadas em formato de protocolo AXI4-Lite. O protocolo AXI4 acessa cada registro interno de configuração dos módulos ou submódulos por meio de endereçamento, portanto cada registro tem uma faixa de endereço específico.

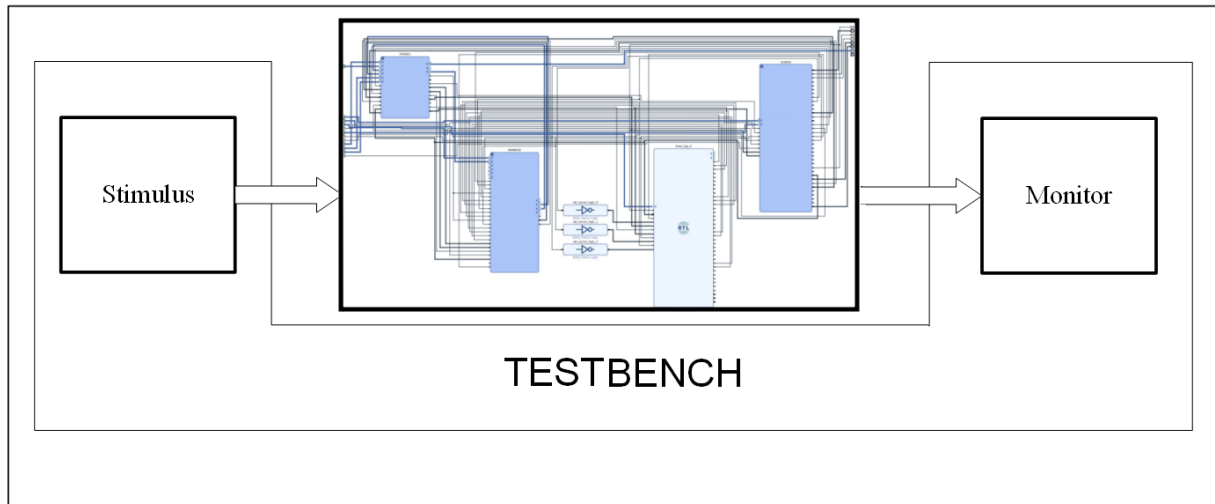


Figura 13 - Ambiente de Simulação

Os estímulos gerados por meio dos drivers são memórias ou banco de registros (*reg_bank*), como ilustra a Figura 14 - Drivers, que armazenam o endereço e o valor a ser escrito pelo ambiente de simulação. Estes registros são lidos sequencialmente e sincronizados por ciclos de relógio. Uma máquina de estado em cada driver controla a transmissão ou recepção dos dados para a DUT obedecendo as diretrizes do protocolo AXI4-Lite e as individualidades técnicas de cada bloco do sistema.

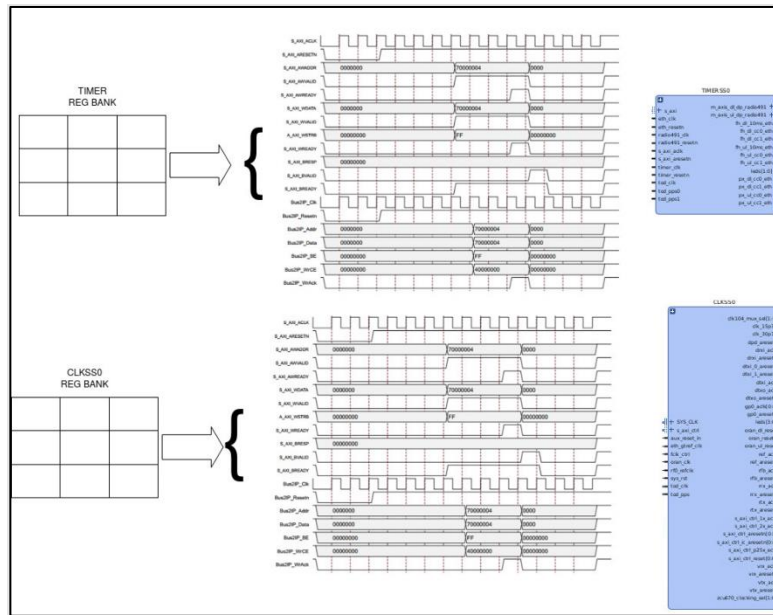


Figura 14 - Drivers

Inicialmente quando o sistema é retirando do resete, o ambiente de simulação envia o endereço do modulo CLKSS0 e em seguida os valores a serem configurados. Resumidamente é habilitado alguns resetes para que *clocks* comessem a oscilar. O próximo modulo a ser configurado é o TIMERSS0, que tem temporizadores independentes para cada portadora (CCs). Entretanto é opcional esta escolha, pois existe um temporizador de 10ms para UL e outro para DL, que pode ser utilizado para gerar um sinal de referência para todas as portadoras utilizadas. Para cada portadora e temporizadores, *Uplink* e *Downlink*, de 10 ms existem quatro registros, um de configuração, um de atraso e dois de períodos. Portanto é mais prático configurar apenas os registros de 10ms e utilizá-lo com referência para todas as portadoras. A Figura 15 – Simulação do processo de configuração do modulo TIMERSS0, mostra a configuração dos registros de endereço 0000, 0004 e 0008 em hexadecimal para os valores 00000021 e 00000011 e 927C0000 também em hexadecimal. O valor 927C0000 corresponde a configuração para se obter um período de 10 ms para um *clock* de 245,76 MHz.

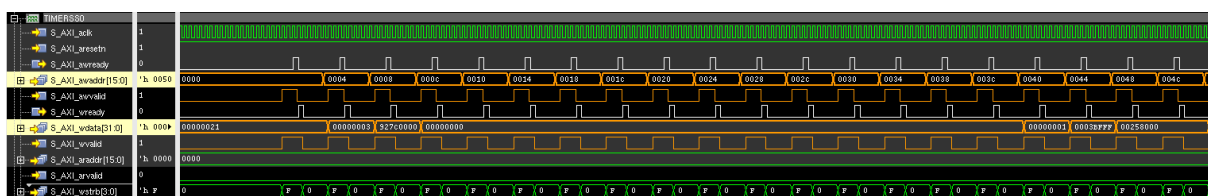


Figura 15 – Simulação do processo de configuração do modulo TIMERSS0

O módulo ORANSS0 foi concebido para apoiar a gestão dos planos de controle, usuário e de sincronização, funcionando como uma interface de rede inteligente e adaptável como um submódulo dentro de uma implementação para uma O-RU 5G. Os registros de configurações são mais complicados devido ao controle de numerologia que ele deve processar. Os dados de entradas devem corresponder a configuração dos registros internos conforme a numerologia utilizada. O valor utilizados são os mesmos gerados e utilizados pelo Open IP Example Design da AMD, pois os registros de configuração são muitos. A Figura 16 – Simulação do processo de configuração do módulo ORANSS0, mostra a simulação do processo de configuração por meio do protocolo AXI4-Lite, destacando os sinais de endereço e de dados. O ambiente de teste por meio de um driver, envia o endereço específico dos registros com os valores a ser escrito e aguarda os sinais de `axi_awready` e `axi_wready` irem para o nível lógico um, confirmando a escrita.

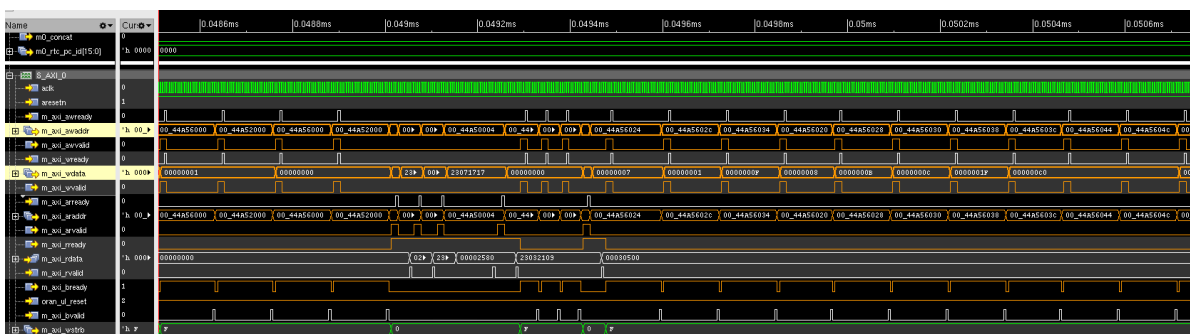


Figura 16 – Simulação do processo de configuração do módulo ORANSS0

O sistema Ethernet junto com o O-RAN forma o chamado *fronthaul*, neste contexto o IP Ethernet é incumbido pela interface com a O-DU por meio de cabos ethernet ou fibra ótica. O IP ETHSS0 tem como submódulo o PTP_1588_timer_sync responsável pela implementação do protocolo de Tempo (PTP), definido pelo IEEE 1588, que é usado para sincronizar relógios em dispositivos conectados a uma rede. Ele é frequentemente usado em cenários onde a sincronização de tempo precisa é fundamental. Em uma rede PTP 1588, há um único relógio mestre que serve como a referência de tempo para a rede. O relógio mestre é geralmente selecionado usando um algoritmo de "melhor relógio mestre". Este relógio mestre é o ponto de referência para a sincronização de tempo em toda a rede. Periodicamente, o relógio mestre realiza uma amostragem do seu contador de tempo de referência do sistema e transmite esse valor amostrado em toda a rede usando pacotes definidos. Essa amostra de tempo é chamada de "*timestamp*". O sistema está implementado

para funcionar com o formato Tod (*Time-of-Day*). Que consiste em um campo de segundos de 48 bits e um campo de nanossegundos de 32 bits. Os registros de configurações de controle do Ethernet têm como uns dos conteúdos a numerologia que ele deve processar e os valores de tempo para o submodulo PTP_1588.

Depois que todos os módulos foram configurados, os dados são disponibilizados na entrada do modulo ORANSS0 por meio do ambiente de teste, igual a simulação da Open IP Example Design da AMD. A Figura 17 - Simulação de Downlink, destaca o mesmo sinal **m0_radio_app_head_valid** sendo posto no estado de nível lógico um avisando que o cabeçalho é valido. Os sinais com sufixo **s_axis_eth_** se referem as entradas dos dados através do protocolo AXI4-Stream, como se os dados estivessem vindo da O-DU para a O-RU.

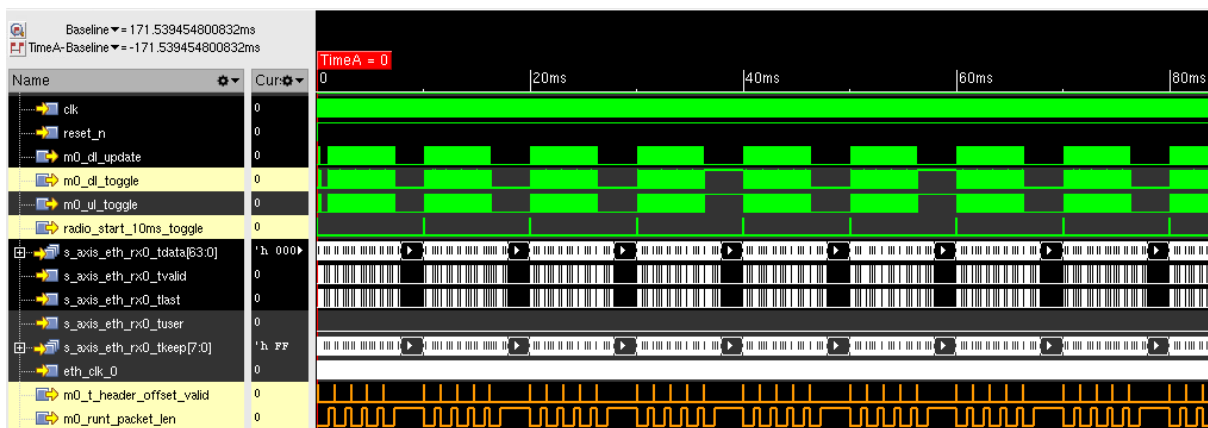


Figura 17 - Simulação de Downlink

6.2 Simulação DFE OFDM.

As simulações da parte do processamento de sinais foram realizadas apenas para o modulo interno DFE OFDM que faz parte do CHAINSS0. Este modulo transforma sinais do domínio do tempo em sinais no domínio da frequência quando no modo UL e frequência-tempo quando em DL. Quando no modo DL, os sinais complexos no domínio da frequência provenientes do *fronthaul* são convertidos em dados no domínio do tempo, no caso de DL são os dados recebidos do *down-converter mixer*. O diagrama da Figura 18 - arquitetura DFE OFDM, mostra a arquitetura interna do DFE OFDM.

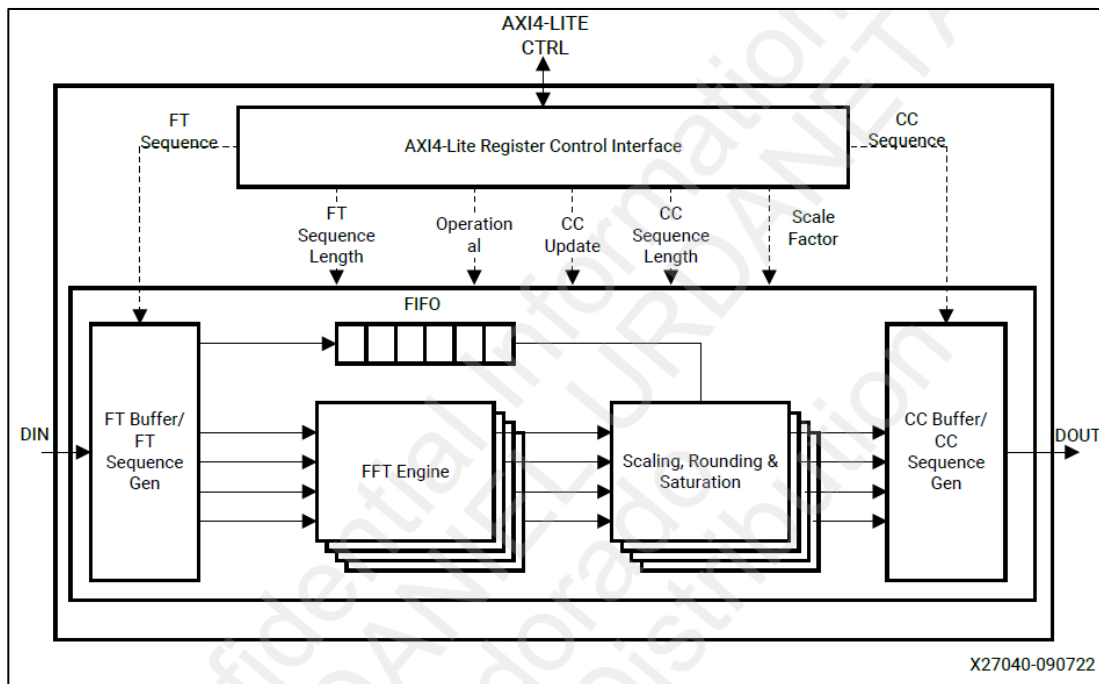


Figura 18 - arquitetura DFE OFDM

Como no caso de outros IPS da AMD, este IP recebe as configurações do PS pelo Axi-Lite. As configurações da sequência de FFT, o tamanho, o fator de *scaling* feito dentro da FFT, se a operação do bloco é DL ou UL é escrito no conjunto de registradores do AXI-Lite. Como mostra a figura o conjunto de amostrar complexas é bufferizada, depois é feita a FFT e por último é feito o *scaling* do dado na saída da FFT para evitar overflow.

A simulação do DFE OFDM do projeto de exemplo gerado pelo Vivado é uma co-simulação entre um emulador de processador chamado QEMU e o simulador xsim da xilinx. A operação de escrita feita pelo PS é feita neste caso pelo QEMU, a Figura 19 - Mensagens de saída em console do QEMU mostra as mensagens do QEMU quando é rodada a simulação. A Figura 20 - XSim Vivado Atividade de escrita/leitura do AXI Lite, mostra o processo de escrita nos registradores internos do IP pelo QEMU. A Figura 21 - Xsim Vivado Protocolo AXI Stream mostram a atividade do protocolo AXI-Stream (a interface de dados do IP) e nela os sinais de AXI_ *valid*, que validam que os sinais de entrada e saída do IP estão certos, e na Figura 22 - Xsim Vivado Máquinas de estado, são mostradas as atividades das máquinas de estados internas do bloco, que mostra as transições nos estados do cálculo do armazenamento de dados nos buffers e o cálculo da FFT.

```

Start Simulation with the following environment variable set:
setenv COSIM_MACHINE_TCPIP_ADDRESS tcp:127.0.0.1:9028
PMU Firmware 2019.1      May 9 2019   05:56:58
PMU ROM Version: xpbr-v8.1.0-0
main.c
exdes.c

OFDM Example Design - Start
SW Version: Major 1, Minor 0

Active: 1Added CCID 3
SetNextCCCFgAndTrigger Successful
OFDM Example Design - Complete
DFE OFDM Example Design Run Succeeded
    
```

Figura 19 - Mensagens de saída em console do QEMU

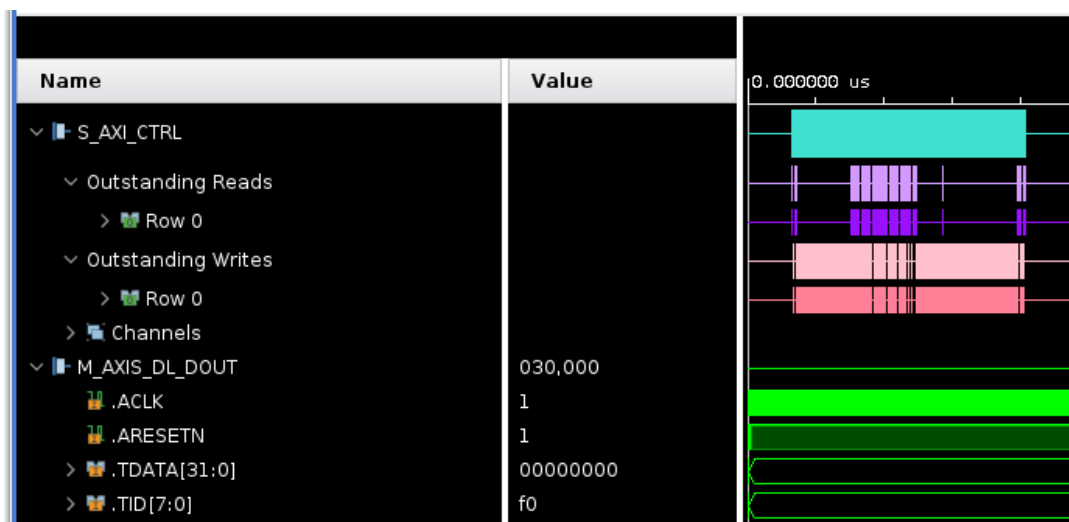


Figura 20 - XSim Vivado Atividade de escrita/leitura do AXI Lite

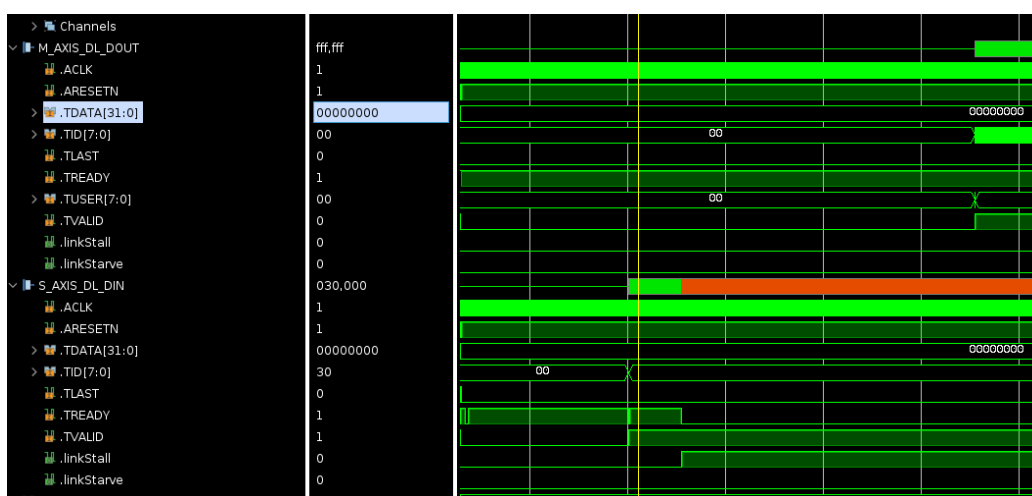


Figura 21 - Xsim Vivado Protocolo AXI Stream

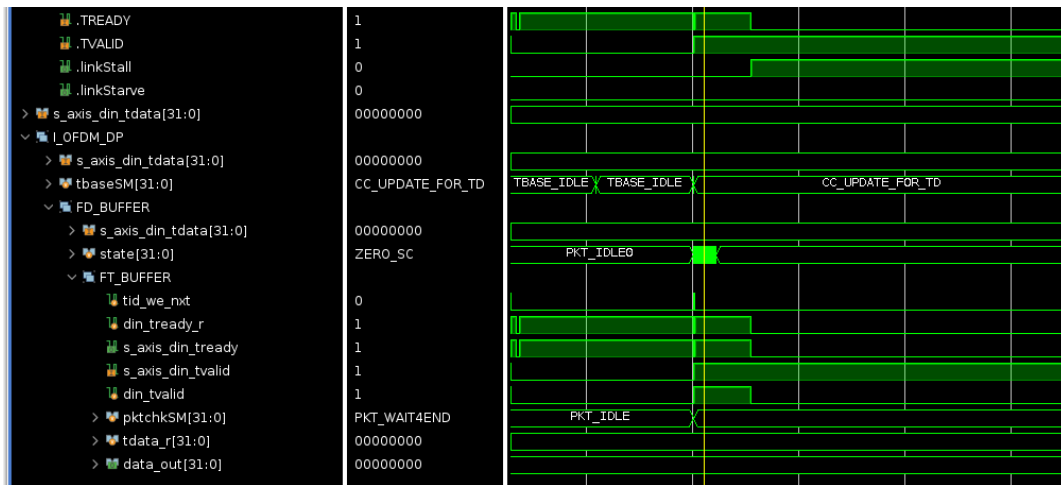


Figura 22 - Xsim Vivado Máquinas de estado

6.3 Resultados obtidos com as Simulações

As configurações individuais e os envios de dados foram e são um processo desafiador para o projeto. Pois as complexidades individuais de todos os blocos junto com as sequências de ativação dos resetes e relógios de sincronismos de cada bloco, tornam o sistema difícil de simular. Portanto o resultado da Figura 17 - Simulação de Downlink e Figura 22 - Xsim Vivado Máquinas de estado, foi uma conquista importante que foi alcançada depois de toda uma sequência de estudos e testes descritos até o momento. Resultando em uma metodologia inovadora de abordagem para o circuito O-RAN, pois dividi o sistema em duas partes, que são verificadas independentemente, com um outro software mais rápido chamado SimVision, para o caso dos blocos da rede Ethernet. Uma das vantagens da simulação foi a possibilidade de entender, por meio de visualizações de forma de ondas, o comportamento interno e externo de estruturas lógicas do sistema. Portanto os resultados da simulação mostram a viabilidade do sistema frente a um protocolo de rede OpenRAN ainda em fase de crescimento no mercado.

7 Setup de Laboratório

Para que sejam realizados testes de verificação das funcionalidades fornecidas pelo projeto de referência da Xilinx assim como validar as eventuais alterações feitas a esse projeto pelo Eldorado é necessário um conjunto de equipamentos de laboratório para que esse trabalho seja realizado. Tais equipamentos devem ser capazes de fornecer duas funcionalidades diferentes:

- Gerar e analisar a atividade e tráfego de dados que ocorre na interface de *fronthaul*;
- Gerar sinais RF compatíveis com padrões de *Uplink* de uma RU e ser capaz de receber e analisar sinais RF compatíveis com os padrões de *Downlink* de uma RU.

7.1 Equipamentos utilizados

O primeiro equipamento selecionado para realizar essas funções é o Keysight S5040A *Open RAN Studio Player and Capture Appliance*, apresentado na Figura 23 mostrando seu painel frontal com a conexão para *fronthaul* e conexões auxiliares de *trigger* e relógio.



Figura 23 - Painel frontal do S5040A

O segundo equipamento necessário é um conjunto de módulos montado a um chassi PXI (*PCI eXtensions for Instrumentation*) que permite realizar a geração, recepção e análise dos sinais de rádio frequência (RF). Na Figura 24 - Chassi PXI com os módulos para geração e análise de sinais RF, é possível ver o chassi PXI contendo os módulos M9300A que funciona como um gerador de frequência de referência para o módulo M9410A que é um Transceptor Vetorial, combinando a capacidade de gerar e analisar sinais em uma banda de até 1.2GHz nas frequências entre 0.38GHz – 6.0GHz, cobrindo a faixa FR1 contemplada no projeto. Por fim tem-se o módulo M9037A que é um computador que age como controlador do sistema através de softwares que interagem com os outros módulos instalados no chassi.

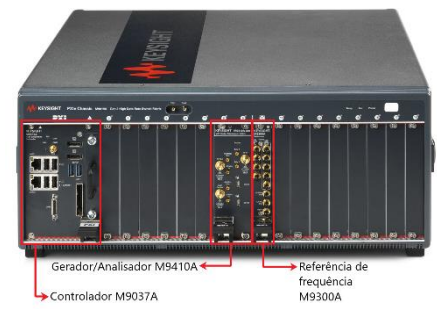


Figura 24 - Chassi PXI com os módulos para geração e análise de sinais RF

7.2 Softwares utilizados

Para que os equipamentos sejam controlados adequadamente de acordo com as necessidades do projeto foi necessário um período de estudo e entendimento dos *softwares* de controle envolvidos no processo de geração e análise de sinais. Os *softwares* envolvidos no processo e que precisaram ser estudados são:

- Keysight PathWave Signal Generation
- Keysight Open RAN Studio
- Keysight PathWave 89600 VSA

A seguir é detalhado o processo envolvido para a utilização de cada um dos *softwares* que veio a ser entendido a partir do estudo realizado para sua utilização efetiva.

7.2.1 Keysight PathWave Signal Generation

O Software PathWave Signal Generation é o responsável dentro do conjunto de softwares por criar a configuração do sinal banda base para 5G ORAN permitindo uma grande gama de opções sobre o tipo de sinal que é gerado. Este software permite gerar tanto configurações de sinais de *Downlink* e *Uplink* assim como um sinal de PRACH que é um tipo específico de sinal de *Uplink* vindo da UE. Foi realizado um estudo inicial sobre o software para entender de maneira geral sobre os três tipos de sinais que poderiam ser gerados e qual grau de controle existe sobre a geração.

No caso dos sinais de *Downlink* e *Uplink* as configurações são separadas nas sessões de Portadora (em inglês: *Carrier*), Sessão de banda (BWP, em inglês: *Bandwidth Part*) e de Canal (em inglês: *Channel*). Na sessão de Portadora são controladas as características gerais do sinal a ser gerado. Na Figura 25 - Controle de configuração de portadora do software Pathwave Signal generation, é apresentada a tela de configuração mostrando os parâmetros que podem ser configurados como por exemplo o deslocamento de frequência

Por fim tem-se a seção de configuração do Canal onde efetivamente é configurado os quais tipos de dados serão transmitidos, suas formatações e em alguns casos e qual o conteúdo deles. Para uma portadora do tipo *Downlink* é possível configurar os seguintes tipos de blocos de sinais:

- Sinal de Sincronismo (SS, do inglês: *Synchronization Signal*) e Canal de Físico de Transmissão (PBCH, do inglês: *Physical Broadcast Chanel*)
- Informação de controle de *Downlink* (DCI, do inglês: *Downlink Control Information*)
- Canal Compartilhado de *Downlink* (DL-SCH, do inglês: *Downlink Shared Channel*)
- Sinal de Referência de Informação do Estado do Canal (CSI-RS, do inglês: *Channel-State Information Reference Signal*)
- Sinal de Referência de Posicionamento (PRS, do inglês: *Positioning Reference Signals*)
- Sinal de Referência de Gerenciamento de Interferência Remota (RIM-RS, do inglês: *Remote Interference Management Reference Signal*)
- Coexistencia com LTE

Dentre esses blocos de informação o estudo foi focado no SS/PBCH e no DL-SCH uma vez que são os blocos de informação mais básicos a serem utilizados em uma comunicação do tipo Downlink real. Na Figura 27 - Configuração do Canal. A esquerda bloco de configuração do SS /PBCH e a direita do bloco DL-SCH, são mostrados do lado esquerdo os parâmetros possíveis de serem configurados no bloco SS/PBCH enquanto na esquerda e apresentado as configurações do bloco de DL-SCH.

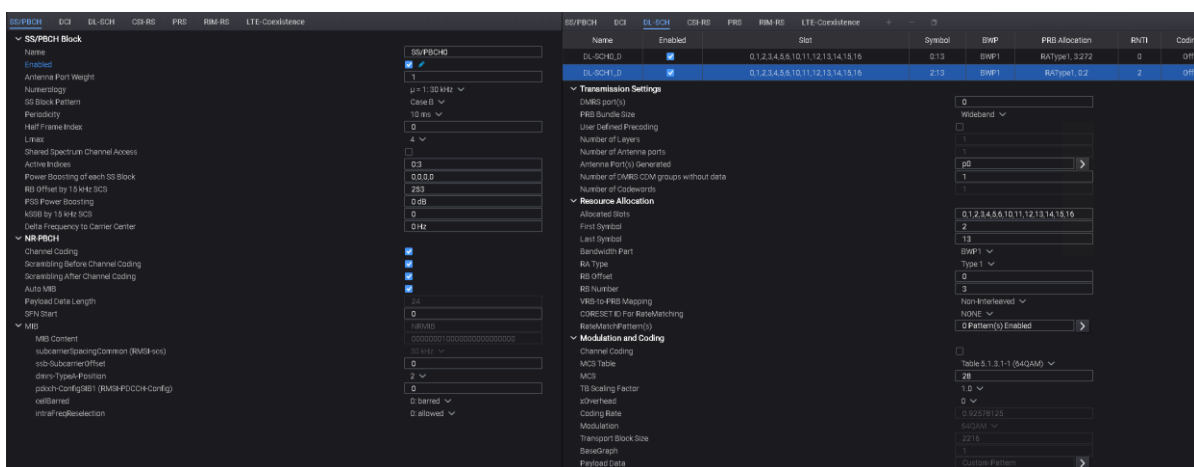


Figura 27 - Configuração do Canal. A esquerda bloco de configuração do SS /PBCH e a direita do bloco DL-SCH

Para uma portadora do tipo *Uplink* os blocos de sinais possíveis de serem configurados são:

- Canal de Controle de Informações (UCI, do inglês: *Control Chanel Information*)
- Canal Compartilhado de *Uplink* (DL-SCH, do inglês: *Uplink Shared Channel*)
- (SRS, do inglês: *Sounding Reference Signals*)

Tendo os sinais configurados dois tipos de ações podem ser realizadas. Ou a configuração do sinal pode ser exportada para o *software Keysight Open RAN Studio* onde será gerado o tráfego ethernet da interface *fronthaul* ou então é possível gerar o sinal RF correspondente a configuração gerada.

Durante o aprendizado do uso do software foi visto que é possível realizar as configurações dos sinais de duas maneiras principais. Uma utilizando configurações padrões disponibilizadas dentro do software e outra fazendo a entrada manual de cada parâmetro. As configurações pré-definidas que já existem dentro do software são ou configurações com a máxima ocupação de banda ou então são configurações criadas para atender testes pré-estabelecidos pelas normas da 3GPP [28] [29]. Durante todos os testes iniciais e para auxiliar no entendimento de algumas funcionalidades do software foram utilizadas essas configurações pré-definidas, principalmente a de ocupação máxima de canal.

7.2.2 Keysight Open RAN Studio

O próximo *software* que foi estudado é o *Keysight Open RAN Studio* cuja função é a de emular uma DU, criando o tráfego de dados na interface do *fronthaul* assim como agir como Mestre PTP, servindo para sincronizar temporalmente os dispositivos conectados. Para a utilização desse software foi necessário realizar o estudo de algumas características de uma DU assim como da interação entre a DU e a RU.

Tendo carregado o arquivo de configuração gerado pelo PathWave Signal Generation, o primeiro ponto estudado foi em relação as configurações de extensão de Portadora-Antena (eAxC, do inglês: *Extended Antenna-Carrier*). Esta é uma configuração que representa, para cada fluxo de dados sendo enviado para uma RU, identificações de fonte e destino para o pacote de dados que está sendo transmitido. Duas configurações são necessárias de serem realizadas nesse passo. A Figura 28 - Painel de configuração do eAxC, apresenta os campos de configuração, sendo eles a largura de bits de cada um dos quatro campos do eAxC assim como os valores de cada um dos campos. Esses valores devem ser compatíveis com a RU sob teste que está sendo utilizada.

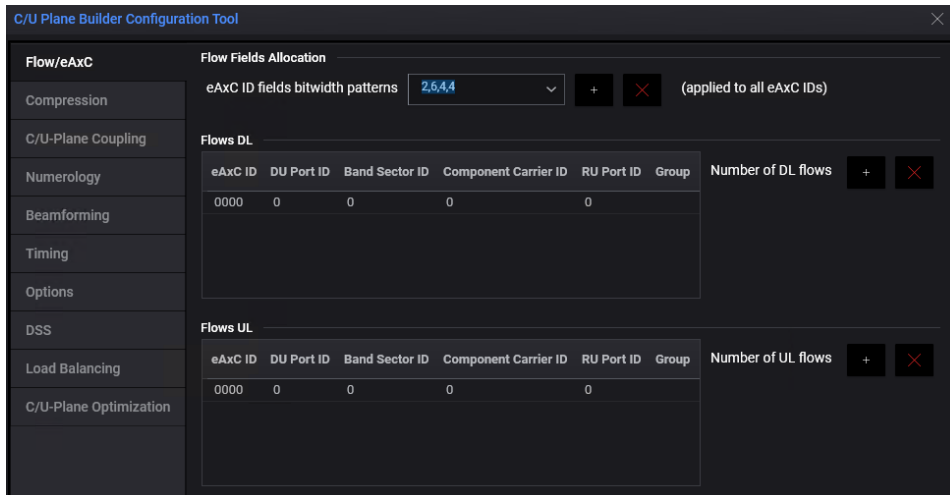


Figura 28 - Painel de configuração do eAxC

Uma segunda configuração necessária é relacionada ao tipo de compressão que está sendo utilizada pela RU. Essa configuração é mostrada na Figura 29- Painel de configuração de compressão, onde pode-se ver que é necessário configurar a compressão individualmente para cada eAxC criado, assim como definir qual o tipo de compressão, a largura de bits I/Q que está sendo utilizado e se a compressão é estática ou não.

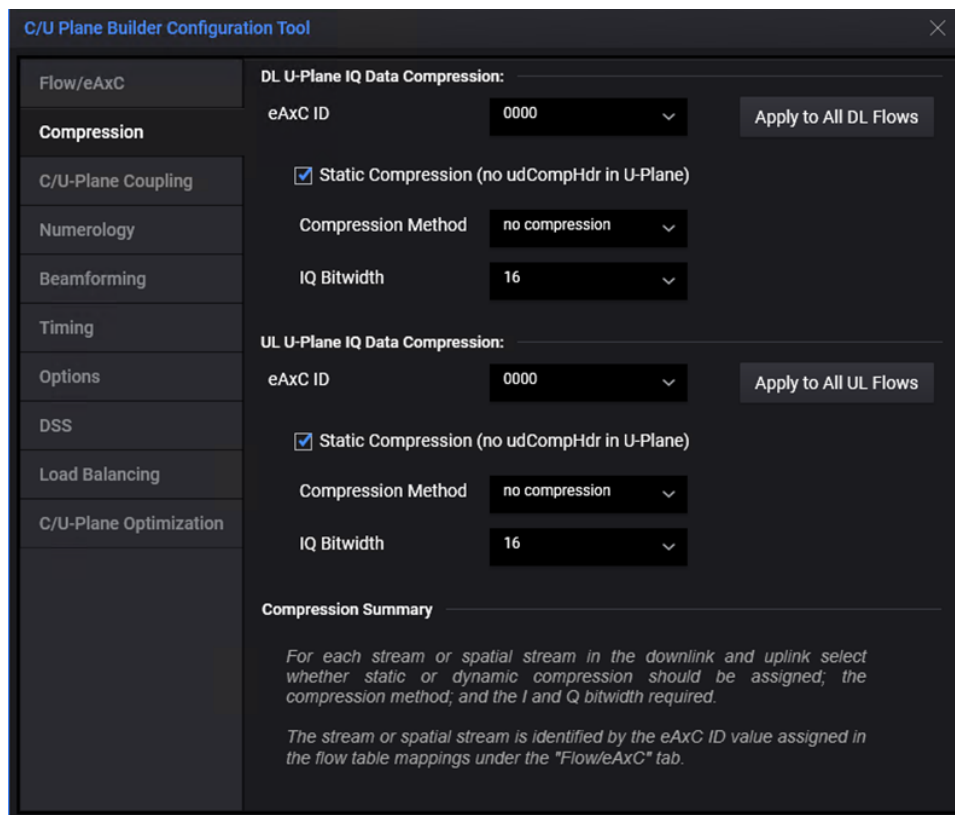


Figura 29- Painel de configuração de compressão

Em relação a transmissão de mensagens entre a DU e a RU existem janelas de tempo que devem ser respeitadas conforme as definições apresentadas na documentação da Open RAN Alliance [30]. Esta janela deve ser definida de acordo com alguns parâmetros físicos de tempo de processamento da RU que está sendo testada nas operações de *Downlink* e *Uplink* assim como o atraso inserido meio físico que conecta RU à DU. Estas janelas definem o tempo relativo de recepção entre mensagens do Plano C e mensagens do Plano U de modo que haja tempo hábil para que tanto a RU possa processar as mensagens que chegam até ela e realizar a transmissão do sinal assim como na DU ela receber e processar as mensagens recebidas da RU. A Figura 30 - Painel de configuração dos parâmetros de temporização, apresenta o painel de configuração dessas temporizações tanto para os sinais na direção de *Downlink* quanto de *Uplink*.

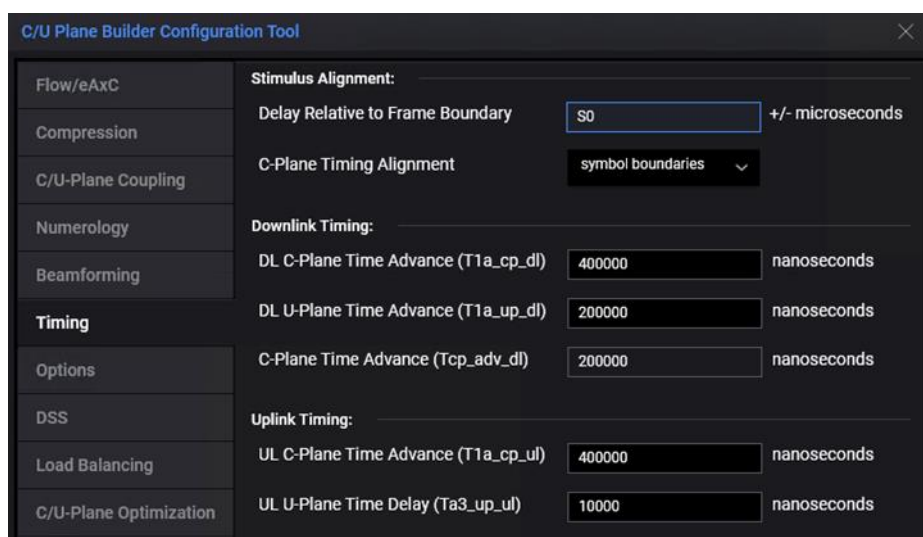


Figura 30 - Painel de configuração dos parâmetros de temporização

Uma última opção de configuração relevante observada no estudo inicial foi a respeito do uso de etiquetas de vLAN. Esse é um identificador que está presente nos cabeçalhos de transporte dos pacotes e que deve estar habilitado ou não a depender das configurações realizadas na RU conectada. A Figura 31 - Painel de configuração da opção de vLAN, apresenta o painel de configuração dessa opção dentro do software, caso não seja utilizado vLAN nenhuma configuração extra se faz necessária, porém caso seja utilizada, também é necessário, para cada eAxC utilizado associar um identificador de qual número de vLAN será utilizado, ainda que o mesmo possa ser utilizado para múltiplos eAxC. Também é possível definir o código do ponto de prioridade (PCP, do inglês: Priority Code Point), ou seja, a prioridade de tratamento dos dados que estarão chegando, para os planos C e U.

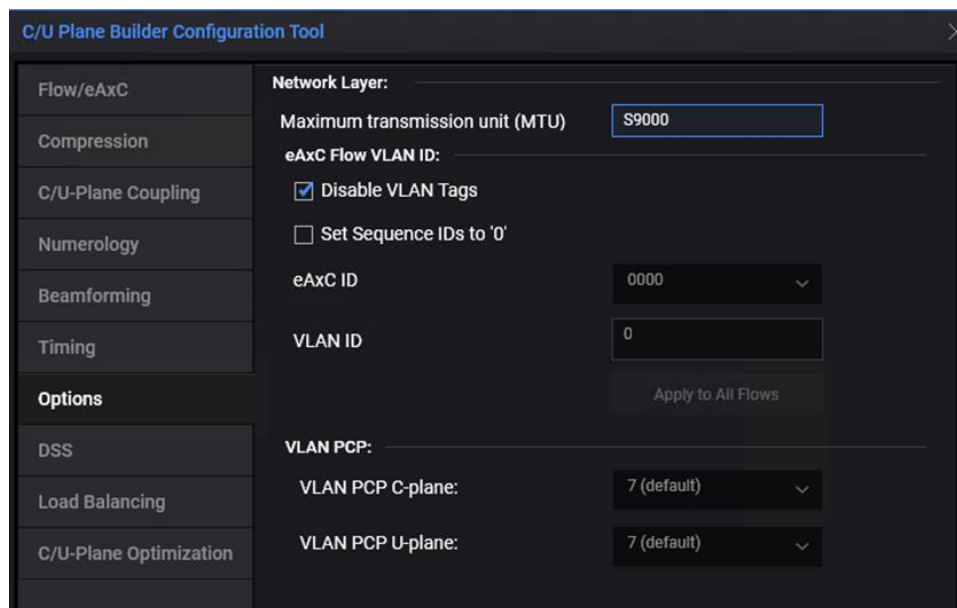


Figura 31 - Painel de configuração da opção de vLAN

Realizadas as configurações correspondentes as características da DU e RU o estímulo do *fronthaul* é gerado como um arquivo com extensão tipo “.pcap”, que é então recarregado para o próprio software e então é possível fazer reprodução do sinal em uma porta de saída do tipo QSFP28 presente no equipamento. Alternativamente o equipamento também pode ser utilizado como um gravador e analisador de sinais do *fronthaul* podendo assim também ser utilizado para verificar o funcionamento dos sinais de *Uplink* de uma RU.

7.2.3 Keysight PathWave 89600 VSA

O último software utilizado no fluxo de testes que foi necessário realizar um estudo foi o Keysight PathWave 89600 VSA, este software é responsável por controlar o analisador de sinais presente no chassi PXI de modo que possível realizar a demodulação do sinal RF e a análise da qualidade do sinal recebido. Este é um software que já vinha sendo utilizado no Instituto de Pesquisas Eldorado, porém a extensão utilizada para análise de sinais 5G-Open RAN foi uma nova aquisição. A configuração do software para realizar a demodulação do sinal RF pode ser feita de duas maneiras, tanto usando um arquivo de pré-configuração que pode ser gerado pelo Keysight PathWave Signal Generation, quanto realizando a configuração manualmente de todos os parâmetros relacionados a configuração do sinal para que ele seja demodulado. Foi realizado um entendimento de como acessar manualmente todas as configurações necessárias para a correta demodulação do sinal para caso seja necessário fazer pequenas alterações na configuração do sinal. Porém optou-se por de maneira geral utilizar o arquivo de configuração exportado pelo PathWave Signal

velocidade de tráfego, um na direção de *Downlink* e outro na de *Uplink*, sendo que ambos mostram a mesma velocidade conforme o esperado já que o sistema está em *loopback*. Também é mostrado nos painéis a esquerda as estatísticas dos pacotes transmitidos com o número de pacotes e bytes transmitidos e recebidos no plano U e C assim como o status do plano S com as informações dos pacotes PTP enviados e recebidos. Por fim há também um painel informativo com os dados das portas QSFP.

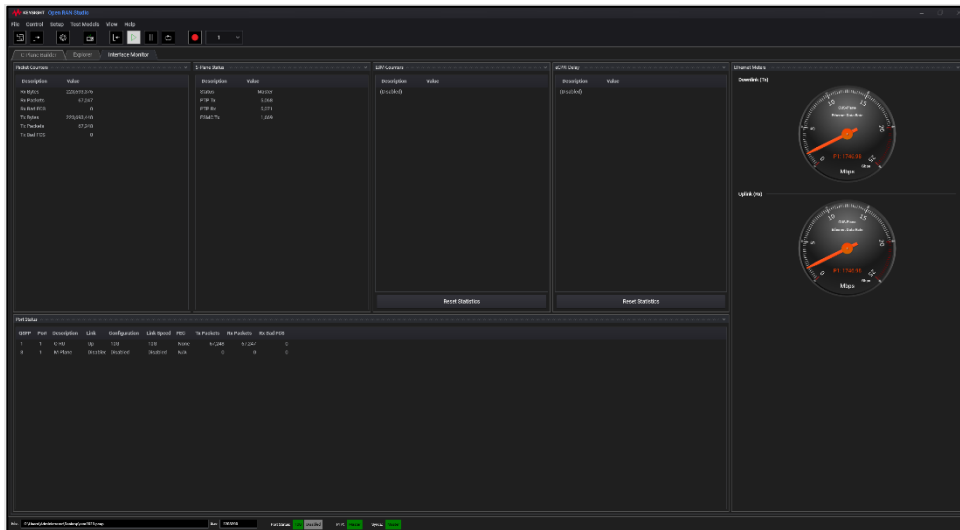


Figura 33 - Painel de monitoramento do tráfego nas portas óticas do equipamento

Um segundo teste realizado para verificar a utilização da aba *Explorer*. Neste teste foi utilizada a mesma configuração descrita anteriormente, porém em vez de usar apenas uma portadora a mesma configuração foi replicada para um total de 10 portadoras. Esse teste foi realizado para se entender quais são as informações que podem ser obtidas dos pacotes que estão sendo transmitidos. É possível carregar o mesmo arquivo que é utilizado para a reprodução do sinal do *fronthaul* na porta SFP, e sem reproduzi-lo fazer uma análise dele. Na Figura 34 - Painel de análise dos sinais trafegando no fronthaul,

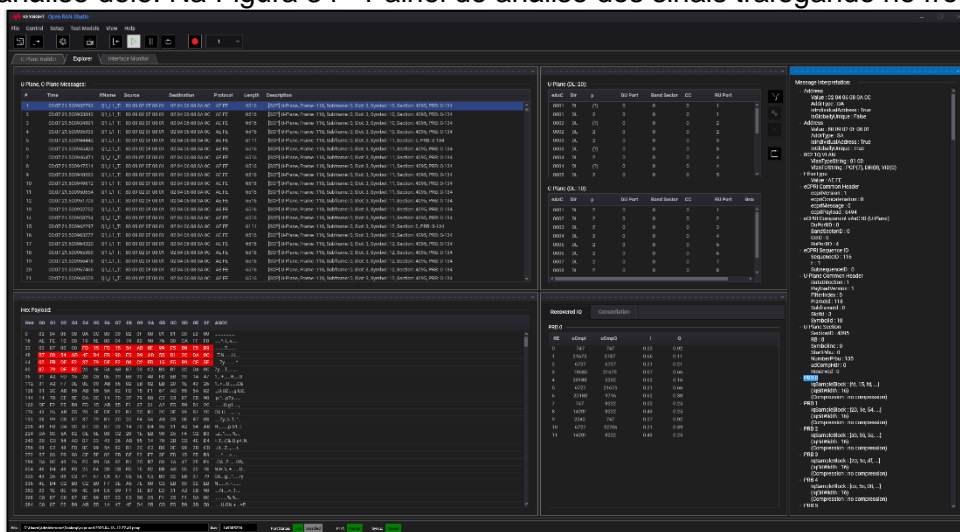


Figura 34 é possível ver que nesse modo pode-se visualizar todas as mensagens do plano U e C, e selecionando uma mensagem um painel de interpretação da de todos os campos

da mensagem, sendo assim possível utilizar a o software para depurar possíveis problemas com os pacotes trafegando no *fronthaul*.

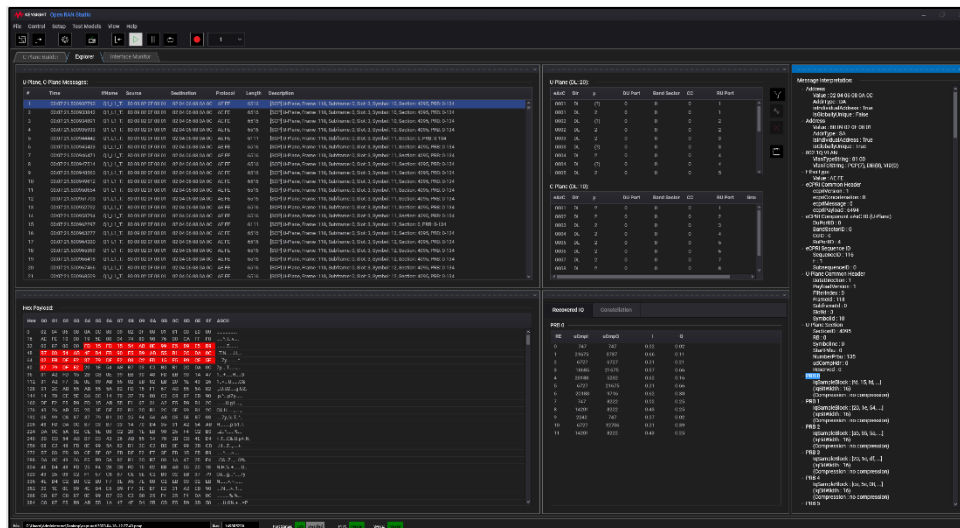


Figura 34 - Painel de análise dos sinais trafegando no fronthaul

Para os testes em *loopback* do equipamento de geração e recepção de sinais RF é possível ver na Figura 35 - Chassi PXI com módulo RF conectado em *loopback*, que é feita uma conexão através de um cabo SMA no painel frontal do módulo M9410A entre as portas de saída e a de entrada de sinal. Com essa conexão realizada, utilizou-se o software Pathwave Signal Studio tanto para gerar a configuração do sinal RF a ser enviada para o gerador como para gerar configuração de demodulação a ser carregada no software analisador VSA.



Figura 35 - Chassi PXI com módulo RF conectado em loopback

A configuração de sinal escolhida para esse teste em *loopback* foi de uma portadora de 100MHz, com numerologia 1, modulação 1024 QAM, apenas o bloco de DL-SCH ativado e o conteúdo da mensagem sendo transmitida foi configurado para ser uma sequência repetitiva facilmente identificável conforme mostrado abaixo:

0000000011111111000011110011001101010101 1010101011001100111100001111111100000000

Em relação aos parâmetros físicos do sinal, foi escolhido usar uma frequência central de 3500MHz com uma potência de -30dBm . Com todas as configurações realizadas, o sinal é carregado no gerador e a geração do sinal de saída é ativada. No software de análise é carregado o arquivo de configuração de demodulação de modo que na o sinal é recebido e demodulado. Na Figura 36 - Sinal 100MHz, demodulado com estatísticas da qualidade de sinal, é mostrado o resultado da analisa do sinal recebido onde é possível ver 6 painéis de resultado com as seguintes informações:

- Medida OFDM, todas as camadas: Constelação do sinal recebido mostrando uma constelação 1024QAM em pontos amarelos conforme esperado assim como uma constelação QPSK em azul correspondente a um sinal de sincronismo existente junto ao bloco DL-SCH (painel superior esquerdo)
- Símbolos decodificados: mostrando os bits recebidos depois de terem passado pelo processo de demodulação e decodificação, é possível identificar exatamente a mesma sequência dos bits configurados para a transmissão (painel superior central)
- Resumo de informações: mostrando um resumo de métricas de qualidade do sinal que está sendo recebido como por exemplo a potência do sinal, a EVM total e de pico do sinal, erros de magnitude, fase e frequência assim como o erro de borda de relógio do alinhamento dos símbolos OFDM (painel superior direito)
- Espectro: apresentando a visualização do espectro do sinal recebido onde dentro da banda de análise escolhida (painel inferior direito)
- Erro vetorial do espectro: mostrando a EVM de cada subportadora individual do sinal recebido, é possível identificar três informações diferentes nesse painel, a média da EVM por sub-portadora mostrada em branco enquanto em amarelo são mostrados a EVM instantânea das sub-portadoras de dados e em azul as de sincronismo (painel inferior central)
- Resumo do quadro: apresenta as informações básicas da qualidade do quadro de sinal recebido separado pelos tipos de sinais existentes, nesse caso o sinal de dados

utilizados, desafios enfrentados e estratégias para otimização do ambiente de desenvolvimento.

7.4.2 A Importância da Placa de Desenvolvimento:

A placa ZCU670 serve como a referência onde as ideias concebidas pela equipe começam a tomar forma. É o ponto de convergência entre o conceitual e o tangível, onde as funcionalidades projetadas são implementadas e testadas. Portanto, a preparação adequada desta plataforma é crucial para assegurar que os desenvolvedores possam explorar todo o potencial do hardware, identificar possíveis desafios e, finalmente, otimizar o desempenho do produto.

7.4.3 Desafios na Preparação da Placa de Desenvolvimento:

Apesar da importância da preparação da placa, muitas equipes enfrentam desafios significativos nessa fase. A integração de diferentes componentes e softwares, a interface com o ambiente de depuração, a conexão com os instrumentos de análise, entre outros fatores, são elementos que demandam atenção minuciosa. Neste contexto, a abordagem metodológica e a aplicação de práticas eficientes tornam-se cruciais para superar esses desafios de maneira eficaz.

Ainda existem desafios técnicos a serem resolvidos no software embarcado, mas que no atual estágio de desenvolvimento não se mostram obstáculos para que a equipe realize as etapas futuras do projeto.

Nos próximos itens detalha-se cada um desses elementos, fornecendo uma base valiosa para a placa ZCU670 ser utilizada adequadamente pela equipe. Deste modo contribuiu-se para a eficiência e sucesso da equipe, promovendo avanços significativos no desenvolvimento do projeto.

7.4.4 Equipamentos e softwares utilizados na preparação da placa ZCU670:

Os pré-requisitos para a preparação da placa de desenvolvimento são instalar e configurar os equipamentos, sistemas e softwares necessários à operação da placa. Estes são:

Equipamentos:

- Fonte de Alimentação (fornecida com a placa ZCU670);
- Transceptor Vetorial tipo VXT modelo M9410A da Keysight; (software de análise vetorial *PathWave 89600 VSA*);
- Gerador de tráfego de Fronthaul 5G ORAN modelo *S5040A* da Keysight;

- Computador usado para rodar o software de depuração remota de hardware via USB/JTAG.
- Placa de interligação de RF modelo XM755 da AMD (fornecida com a placa ZCU670);
- Softwares:
- Software de análise vetorial modelo *PathWave 89600 VSA* da Keysight;
- Software para geração da configuração do sinal 5G ORAN modelo *PathWave Signal Generation* da Keysight;
- Software para geração de tráfego de DU 5G ORAN modelo OpenRAN Studio da Keysight;
- IDE de lógica programável modelo *Vivado* da AMD;
- IDE de software embarcado para os RFSoc modelo *Vitis* da AMD;
- Servidor de depuração remota de hardware via USB/JTAG da AMD;
- Ambiente para criação de imagens de Linux embarcado *Petalinux* da AMD.

Os IDEs *Vitis* e *Vivado* da AMD foram instalados com as respectivas licenças em um servidor Linux na rede do Instituto Eldorado (doravante denominado por Servidor Vivado) por serem utilizados por diferentes times e são acessados via conexão remota.

O ambiente para criação de imagens de Linux embarcado *Petalinux* foi instalado em um servidor dedicado (doravante denominado por Servidor Petalinux) por necessitar de muita CPU, memória e espaço em disco para realizar a compilação e criar as imagens em tempo hábil (aprox. 1h). Foi utilizada uma máquina virtual x86_64 com 64GBytes de memória RAM, 48 vCPUS (virtual CPUs) e 250GBytes de disco virtual do providos pelo servidor de armazenamento.

Os softwares da Keysight estão todos instalados do equipamento S5040A da Keysight com as respectivas licenças necessárias. O servidor de depuração remota foi instalado em um computador alocado especificamente para esta função. Todos os equipamentos e computadores descritos, além da placa ZCU670, foram instalados em um rack de 19" para facilitar a interligação entre eles.

7.4.5 Detalhamento da execução da preparação da placa ZCU670.

Com a aquisição das licenças do software Vivado, Petalinux e dos IPs da AMD e sua respectiva instalação no Servidor Vivado no mês de Agosto de 2023 iniciou-se o processo de preparação do software embarcado para a placa ZCU670. Foi usado como referência o documento UG1530 da AMD, fornecido juntamente com o TRD versão 2022.2, para o processo de preparação da placa.

Como passo inicial instalou-se os arquivos do projeto de referência referentes à lógica programável no Servidor Vivado e o software embarcado disponibilizados pela AMD em um computador desktop disponível no laboratório do Instituto Eldorado.

Seguiu-se então à construção da imagem do software embarcado conforme as orientações constantes no capítulo 4 – *Software Design and Build* nesse computador desktop.

Durante esta etapa verificou-se que a compilação da imagem do software embarcado, composto pelo sistema operacional Linux (baseado no Yocto), *device drivers* para acesso à lógica programável em conjunto com a respectiva API e a software de aplicação necessitavam de um computador com especificações mais robustas, pois nos testes iniciais no computador desktop que tinha como especificações básicas 8GB de memória DDR4 e processador i5 de 8a. Geração o processo demorou cerca de 10hrs, o que acarretaria muitos atrasos caso fosse necessário recompilar a imagem. Foi então solicitado o uso de um servidor dedicado com especificações de 48 vCPUs, 32GB de memória DDR4 e 250 GB de disco virtual alocado em armazenamento RAID com discos SSD de alto desempenho, e com isso a imagem pode ser gerada em torno de 40 minutos, um tempo que não geraria impacto durante o desenvolvimento do produto.

Após estar com o software embarcado devidamente compilado e sem erros aguardamos a chegadas das placas, que ocorreu em outubro de 2023, para iniciarmos os testes com a imagem gerada.

Com a chegada das placas, foi montado um setup de preparação, ilustrado na Figura 37 – Setup de preparação da placa ZCU670 montado no laboratório do Instituto Eldorado. O setup é composto por:

- Placa ZCU670 como elemento central do setup;
- Transceptor Vetorial M9410A, com a entrada e saída de RF do transceptor conectadas respectivamente aos conectores de saída e entrada da placa de distribuição de sinais de RF AMD XM755;
- Placa XM755 conectada através do conector do barramento de RF à placa AMD ZCU670;
- Servidor de depuração conectado à rede do Instituto Eldorado via Ethernet e à placa ZCU670 via cabo USB, onde é efetuada a comunicação JTAG e Serial com a placa;
- Fonte de Alimentação conectada à placa ZCU670 e à rede de energia do laboratório.
- Gerador de tráfego S5040A conectado à placa ZCU670 através de cabo de fibra ótica com transceptores SFP de 25Gbps e à rede do Instituto Eldorado via Ethernet.

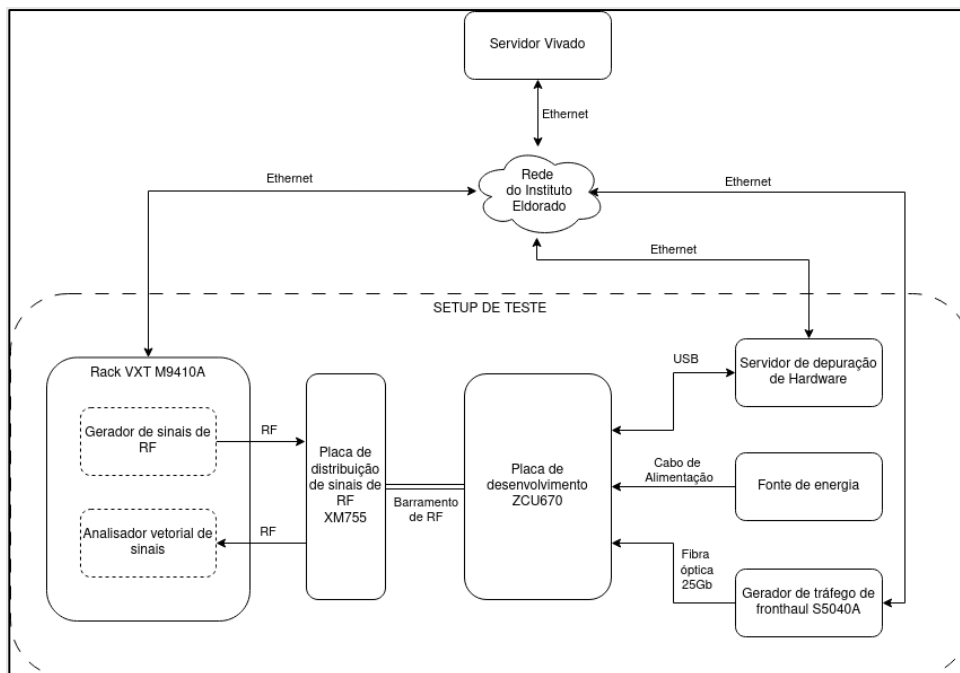


Figura 37 – Setup de preparação da placa ZCU670 montado no laboratório do Instituto Eldorado

Na Figura 38 – M9410A e S5040A montados no rack no laboratório do Instituto Eldorado, temos uma foto de parte do setup descrito acima, onde podem ser vistos o M9410A na parte superior o S5040A logo abaixo montados no rack.

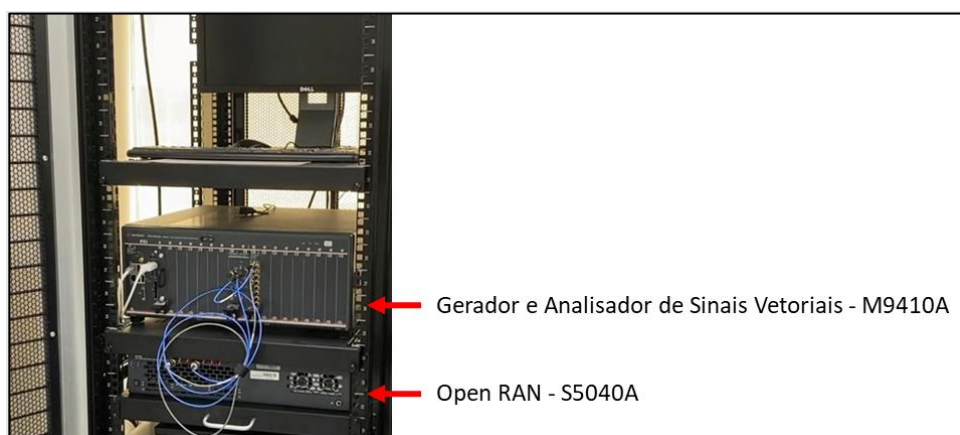


Figura 38 – M9410A e S5040A montados no rack no laboratório do Instituto Eldorado

Na Figura 39 – Conexões das placas AMD XM755 e ZCU670, temos as placas XM755 e ZCU670, já interconectadas, com as conexões da ZCU670 de fibra óptica, fonte de

alimentação e USB do lado direito (de cima para baixo) e as conexões de entrada e saída de RF da XM755 do lado esquerdo (de cima para baixo).

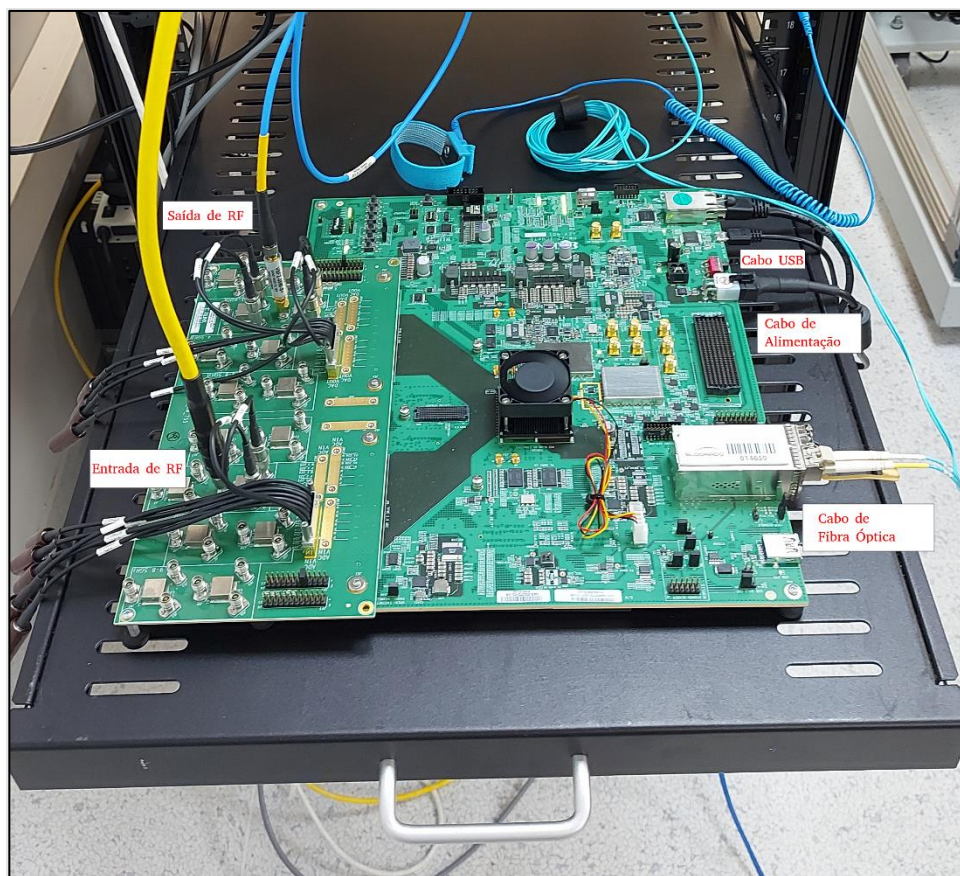


Figura 39 – Conexões das placas AMD XM755 e ZCU670

Para o teste inicial dos 3 conjuntos de placas ZCU670 + XM755 adquiridos o procedimento utilizado foi:

- Montagem das placas no setup
- Verificação do estado dos jumpers e chaves de configuração da placa, que devem estar conforme indicados no UG1530.
- Verificação do autodiagnostico da placa, onde os LED 1234 devem permanecer acesos, o LED 5 ficar aceso constantemente em verde e os LEDs indicadores do estado de funcionamento de todas as fontes de alimentação internas da placa devem estar acessos continuamente.
- Gravação da imagem na placa ZCU670, usando a conexão USB-JTAG pelo servidor de depuração remoto, do software embarcado gerado no servidor Petalinux conforme o procedimento descrito no UG1530 para uso em testes iniciais, pois a inicialização via USB-JTAG fica toda em RAM.

- Gravação da imagem do software embarcada em cartão microSD usando os scripts fornecidos pelo TRD para uso em ambiente de desenvolvimento pois permite salvar dados no cartão após a inicialização.
- Verificação da inicialização correta através da porta USB-Serial conectada ao servidor de depuração remoto, onde o Linux deve iniciar e após o procedimento de inicialização apresentar o prompt de login, tanto usando a inicialização via USB-JTAG quando via cartão microSD.

Durante os testes iniciais, uma das placas ZCU670 apresentou problemas na alimentação do PLL externo, indicado pelo apagamento do LED de alimentação do respectivo circuito e pela ausência de tensão (0V) no ponto de teste da saída de alimentação do circuito do PLL externo. Foi solicitada suporte à AMD para resolução deste problema, e após vários e-mails com solicitações de diversos procedimentos de diagnósticos e fotografias das medições e estados dos LEDs a AMD considerou a placa defeituosa e aprovou o procedimento de troca dela. Este procedimento está em andamento no momento da escrita deste relatório, na fase de tramites alfandegários. Ao todo o processo de diagnóstico, verificação e troca de mensagens faladas e escritas com a AMD durou cerca de 2 semanas, onde pausaram-se os testes com as outras placas.

Após a sinalização pela AMD da aprovação do reparo da placa, retomou-se os testes com as outras 2 placas no início de dezembro de 2023. Fora realizado, em paralelo a análise do problema da placa acima citado, um estudo dos softwares de teste disponibilizados pela imagem fornecida pela AMD, e verificou-se que não existiam ferramentas amigáveis de configuração dos blocos da lógica programável responsáveis pela cadeia de sinais ORAN, apenas as bibliotecas dos controladores de dispositivo ("*device drivers*") estavam disponíveis, criando um desafio adicional para os testes iniciais da placa.

O time de desenvolvedores verificou que no final do terceiro trimestre de 2023 a AMD disponibilizou uma nova versão do TRD, denominada 2023.1. Realizou-se o download do mesmo e fez-se o procedimento de geração da imagem conforme o UG1530, já atualizado para a versão 2023.1, e nesta versão já se encontravam softwares de configuração e teste mais amigáveis, mas sem documentação escrita, apenas as mensagens de ajuda da própria aplicação e os comentários no código fonte. Estas aplicações são o *raft.py*, *xorif.py* e *menu-app.py*. Destas a *menu-app.py* é a mais amigável e tem um conjunto de opções de configuração tal que torna possível a configuração os blocos da lógica programável de forma transparente, baseado apenas nas configurações realizadas para a criação dos pacotes trafegados pela interface óptica de 25GB conectada a uma O-DU.

Usando a aplicação *menu-app.py* foi possível configuração os blocos da lógica programável para decodificar corretamente os pacotes de teste gerados pelo gerador de tráfego de fronthaul S5040A, pacotes estes devidamente configurado pelo software *PathWave Signal Generation* e criados com o software *OpenRAN Studio*. A configuração de teste usada foi a mesma do último teste de loopback do item 4.3, i.e, uma portadora de 100MHz, com numerologia 1, modulação 1024 QAM, apenas o bloco de DL-SCH ativado e sinal em 0dBFS.

Um ponto que ainda não estava correto era que apesar do software *menu-app.py* indicar que os pacotes recebidos pela interface optica estavam sendo corretamente decodificados no analisador vetorial o sinal gerado pelo DAC da placa ZCU670 não era apresentado com a intensidade configurada, e sim um sinal com poucos dBs acima do piso de ruído do DAC. Inicialmente houve suspeita que pudesse ser um problema de hardware, mas o mesmo comportamento era visto nas duas placas que estavam funcionando.

Iniciou-se uma linha de investigação para determinar a causa do problema. Para retirar a suspeita da causa ser no hardware ou na placa, instalou-se o software de configuração e teste do ADC/DAC do RFSoc da AMD. Esta instalação demandou alguns dias de esforço da equipe responsável pela preparação da placa pois foi necessário um computador separado com toda a instalação do Software Vivado e das bibliotecas de Runtime do Matlab R2022 para evitar problemas de compatibilidade. Após dias de estudo conseguiu-se criar uma portadora com a banda de 100MHz centrada em 3500MHz e exportá-la para o software de teste, e assim pode-se observar a portadora com a intensidade de aprox. -50dBm , compatível com o esperado pela simulação fornecida pelo software de teste do ADC/DAC para o mesmo sinal. Deste modo descartou-se a possibilidade de ser um problema de hardware e iniciou-se uma linha de investigação do problema no software de teste.

Em uma exaustiva análise dos códigos fonte da aplicação e de todas as bibliotecas e API implementadas no software embarcado, realizada ao longe de 1 semana pela equipe responsável pela preparação da placa, verificou-se que o modulo responsável pelo cálculo da compensação do ganho do bloco de FFT do sinal estava sendo desviado por uma definição de teste no código e em seu lugar uma constante era introduzida. Comentando a definição no código fonte o cálculo do ganho é realizado e assim a portadora com potência de canal esperada de -40dBm foi obtida na saída no DAC. Infelizmente não encontramos a razão pela qual esse calcula estava sendo desviado no código pois nenhum comentário sobre a necessidade dele estava no código, levando a suspeita de ser um teste de implementação do desenvolvedor do modulo que foi esquecido de ser removido na versão final.

Durante os testes do DAC para diferentes configurações de modulação, onde era necessário transferir a imagem do software embarcado para a placa diversas vezes para assegurar que o teste era sobre uma versão não modificada do software, uma das placas apresentou problema no acesso à memória DDR4 do processador, o que impossibilitou o uso da mesma. Foram feitos diversos testes para assegurar que o problema não era no hardware, incluindo a criação de uma imagem software embarcado de teste de memória de processador, porém o resultado indicava um problema de hardware. A AMD foi novamente contactada e foram solicitados diversos testes, aberturas de chamados de suporte e muitas trocas de e-mails com diversos funcionários da AMD até que se confirma-se o problema na placa e se aprova-se o reparo em garantia. Este procedimento está em andamento no momento da escrita deste relatório na fase de solicitação do retorno para a AMD.

Um outro problema detectado durante a fase de preparação das placas foi na geração da imagem para cartão microSD. Usando o procedimento indicado pela AMD no UG1530 da versão 2023.1 para criar a imagem do cartão microSD, a mesma apresenta problemas durante a inicialização da PLL externa e o kernel do Linux não consegue configurar o dispositivo corretamente. Esse problema impede a imagem do cartão microSD ser usada para fins de desenvolvimento pois não é possível ter persistência das alterações efetuadas na imagem durante o uso na placa, obrigando o uso da imagem para inicialização via USB-JTAG, que tem a mesma característica de não ter persistência das alterações efetuadas.

Na documentação há uma indicação que o problema está na ordem de boot do sistema ser diferente na opção com microSD, sem, no entanto, indicar um caminho para a solução. Por 2 semanas investigou-se a causa do problema, e até o momento se identificou que o erro está no driver do kernel do Linux responsável por carregar o firmware do periférico via interface selecionada. O driver procura o arquivo com o firmware nos caminhos pré-definidos, porém o arquivo não é encontrado e um erro é propagado até a função que originou a chamada. No estágio atual a linha de pesquisa é investigar o porquê de apesar do arquivo estar no caminho de pesquisa este não estar sendo encontrado pelo driver do kernel.

7.4.6 Desafios com relação à preparação da placa de desenvolvimento

Um dos desafios é a resolução do problema da criação de imagem de inicialização para microSD com persistência das alterações realizadas durante a operação do sistema. Um indicativo que o problema deve ter uma solução é que a imagem dos componentes de inicialização do Linux (arquivo BOOT.BIN) fornecido pela AMD para cartão microSD inicializa corretamente a PLL, porém não existem nenhuma documentação de como foi criado esse

arquivo. Outra questão é que esta imagem tem a mesma característica de falta de persistência que a imagem para USB-JTAG.

Outro desafio para a próxima fase é a decodificação dos sinais de RF recebidos pela ZCU670, que ainda não pode ser concluída adequadamente, pois não foi possível analisar se o sinal de RF gerado estava aderente à configuração realizada para a decodificação pela ZCU670. Temos duas linhas de investigação, uma sendo a análise da aderência do sinal gerado à configuração criada, e outra na análise do código fonte do software embarcado para verificar se está configurando os blocos da lógica programável corretamente. Para tal serão necessários criar blocos de depuração na lógica programável para verificar o correto funcionamento dos blocos. Uma vez se conseguindo decodificar o sinal de RF de referência teremos finalizado tanto a cadeia de dados O-RU para O-DU quanto a Cadeia O-DU para O-RU.

Também será um desafio técnico a integração do software de controle do plano de gestão (M-Plane) que está em desenvolvimento pelo CPqD com os controladores de dispositivo da lógica programável.

8 Conclusão

O desenvolvimento da O-RU é um grande desafio técnico científico, uma vez que o sistema envolve uma sinergia complexa entre software, hardware e RF, bem como um trabalho integrado entre as equipes de cada instituição. A simulação da lógica programável teve que ter uma abordagem inovadora, dividindo e utilizando ferramentas diferenciadas para obter um melhor desempenho.

Enquanto a emulação da rede utilizou equipamentos complexos, inovadores e de alto desempenho, que exigem expertises em áreas como redes, RF, sistemas operacionais entre outras para plena operação e uso deles. Apesar das dificuldades as conquistas foram promissoras tanto nas simulações da lógica programável quanto nas emulações de rede no kit de desenvolvimento.

Para a próxima etapa existem grandes desafios, entre eles: a implementação, desenvolvimento e integração das comunicação do hardware com o a camada de software do Plano-M (esta camada em desenvolvimento pelo CPqD); reestruturação de alguns módulos da lógica programável de forma a atender novos requisitos (ativação das portas *TDD Real Time Ports* dos ADCs e DACs, que devem ser habilitadas para cada canal; implementação do bloco de controle das porta TDD; inserção de um modulo AGC (*Automatic Gain Control*) para trabalhar em conjunto com o *Digital Front End*; simulação do caminho de uplink para correta avaliação do sistemas implementados; entre outros.

Todos estes desafios exigirão grandes estudos e soluções inovadoras por parte dos pesquisadores envolvidos para a entrega do objeto do projeto, uma O-RU operacional e compatível com os requisitos propostos.

9 Referências bibliográficas

- [1] RNP, “A2.1 Documento de requisitos Técnicos para desenvolvimento de uma unidade remota Open RAN, OpenRAN@Brasil – Fase 2 (GDrive > documentos > Entregas Oficiais > M01 a M04 MCTI),” 3 abril 2023. [Online]. Disponível: https://drive.google.com/file/d/1yiXfVisgnavzWTSN4oZojNc1QI1WS5Ea/view?usp=drive_link.
- [2] ORAN WG4: Open Fronthaul Interfaces Workgroup, “O-RAN Control, User and Synchronization Plane Technical Specification v09.00, O-RAN.WG4.CUS.0-R003,” 31 março 2022. [Online]. Disponível: <https://orandownloadsweb.azurewebsites.net/specifications>.
- [3] ORAN WG7: White-box Hardware Workgroup, “O-RAN Working Group 7 Whitebox Hardware Reference Design Specification for Outdoor Macrocell with Split Architecture Option 7.2 O-RAN-WG7.OMAC.HRD.0-R003-v03.00,” 16 fevereiro 2022. [Online]. Disponível: <https://orandownloadsweb.azurewebsites.net/specifications>.
- [4] 3GPP, “3GPP Specification series 38,” [Online]. Disponível: <https://www.3gpp.org/dynareport?code=38-series.htm>.
- [5] AMD, “Zynq RFSoc DFE Advance Tools Access Secure Site V2022.2,” [Online]. Disponível: https://www.xilinx.com/member/rfsoc-dfe-tools.html#2022_2.
- [6] AMD, “Zynq UltraScale+ RFSoc DFE Targeted Reference Design User Guide UG1530 (v2.0),” 7 abril 2023. [Online]. Disponível: https://www.xilinx.com/content/dam/xilinx/member/rfsoc-dfe-tools/ip-cores-2022-2/ug1530_rfsoc-dfe-trd_v2.0_WtMkX.pdf.
- [7] 3GPP, “ETSI TS 138 201 V15R15 Technical Specification 5G NR, Physical layer General description,” 09 2018. [Online]. Disponível: https://www.etsi.org/deliver/etsi_ts/138200_138299/138201/15.00.00_60/ts_138201v150000p.pdf.
- [8] 3GPP, “ETSI TS 138 202 Technical Specification V16R15, 5G NR Services provided by the physical layer,” 07 2020. [Online]. Disponível: https://www.etsi.org/deliver/etsi_ts/138200_138299/138202/16.01.00_60/ts_138202v160100p.pdf.
- [9] AMD, “OFDM Reference IP, Application Note XAPP1382 (V1.0) : Zynq UltraScale+ RFSoc DFE Devices,” 26 outubro 2022. [Online]. Disponível: https://www.xilinx.com/content/dam/xilinx/member/rfsoc-dfe-tools/ip-cores2022-2/xapp1382-ofdm_WtMkX.pdf.
- [10] 3GPP, “ETSI TS 138 211 V15.2.0 Technical Specification, 5G NR Physical channels and modulation,” 07 2018. [Online]. Disponível: https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/15.02.00_60/ts_138211v150200p.pdf.
- [11] AMD, “PRACH Processing Subsystem, Application Note XAPP1386 (v1.0). Zynq UltraScale+ RFSoc DFE Devices,” novembro 16, 2022. [Online]. Disponível: https://www.xilinx.com/content/dam/xilinx/member/rfsoc-dfe-tools/ip-cores-2022-2/xapp1386-prach_WtMkX.pdf.
- [12] AMD, “RFSoc DFE DUC-DDC Mixer, PG391 (v2.0) LogiCORE IP Product Guide,” 10 maio 2023. [Online]. Disponível: https://www.xilinx.com/content/dam/xilinx/member/rfsoc-dfe-documents/2023-1/pg393-rfsoc-dfe-duc-ddc-mixer_WtMkX.pdf.
- [13] AMD, “Digital Pre-Distortion V14.0 LogiCORE IP Product Guide PG076,” 10 maio 2023. [Online]. Disponível: https://www.xilinx.com/member/dpd_eval.html.
- [14] AMD, “Peak Cancellation Crest Factor Reduction v7.2 LogiCORE IP Product Guide PG097 (v7.2),” 19 outubro 2022. [Online]. Disponível: https://www.xilinx.com/content/dam/xilinx/member/pc_cfr_eval/v7_2/pg097-pc-cfr_WtMkX.pdf.
- [15] AMD, “O-RAN Radio Interface, LogiCORE IP Product Guide PG370 v2.3,” [Online]. Disponível: https://www.xilinx.com/content/dam/xilinx/member/oran-radio-if/v2_3/pg370-oran-radio-if.pdf.

- [16] IEEE, "IEEE 1588 - Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," 2008. [Online]. Disponível: <https://standards.ieee.org/ieee/1588/4355/>.
- [17] AMD, "10G/25G High Speed Ethernet Subsystem Product Guide PG210, PTP 1588 Timer Syncer Block," [Online]. Disponível: <https://docs.xilinx.com/r/en-US/pg210-25g-ethernet/PTP-1588-Timer-Syncer-Block>.
- [18] Ericsson AB, Huawei Technologies Co. Ltd, NEC Corporation and Nokia, "Common Public Radio Interface eCPRI Specification V2.0," [Online]. Disponível: http://www.cpri.info/downloads/eCPRI_v_2.0_2019_05_10c.pdf.
- [19] IEEE, "IEEE 1914.3-2018 - IEEE Standard for Radio over Ethernet Encapsulations and Mappings," 2018. [Online]. Disponível: <https://ieeexplore.ieee.org/document/8486937>.
- [20] IEEE, "IEEE Std 802.3-2015 - IEEE Standard for Ethernet," 2015. [Online]. Disponível: <https://standards.ieee.org/findstds/standard/802.3-2015.html>.
- [21] IEEE, "IEEE Std 802.3by - IEEE Standard for Ethernet - Amendment 2: Media Access Control Parameters, Physical Layers, and Management Parameters for 25 Gb/s Operation," 2016. [Online]. Disponível: <http://standards.ieee.org/findstds/standard/802.3by-2016.html>.
- [22] AMD, "ORAN simulation example design - How to use the IP example design to evaluate behavior with custom O-DU data packets" [Online]. Disponível: https://support.xilinx.com/s/article/ORAN-IP-Example?language=en_US.
- [23] Zaidi, Ali; Athley, Fredrik ; Medbo, Jonas ; Gustavsson, Ulf ; Durisi, Giuseppe ; Chen, Xiaoming; 5G Physical Layer Principles, Models and Technology Components, 2018.
- [24] 38-213, 3GPP; "NR Physical layer procedures for control"; 2023.
- [25] 38-214, 3GPP; "NR Physical layer procedures for data"; 2023.
- [26] Peterson, Larry; 2022.
- [27] ORAN Alliance; WG4; "Working Group 4 (Open Fronthaul Interfaces WG) Control, User and Synchronization Plane Specification"; 2022.
- [28] 38-141, 3GPP; "NR; Base Station (BS) conformance testing Part 1: Conducted conformance testing"; v17.5.0 e v15.6; Disponível em: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3367>.
- [29] 38-521, 3GPP; "NR; User Equipment (UE) conformance specification; Radio transmission and reception"; v15.3.0 e v15.1.0; Disponível em: <https://www.3gpp.org/dynareport?code=38-series.htm>
- [30] O-RAN ALLIANCE; "O-RAN ALLIANCE Introduces 53 New Specifications Released Since"; 2022; Disponível em: <https://www.o-ran.org/blog/o-ran-alliance-introduces-53-new-specifications-released-since-july-2022>.

10 Histórico de alterações do documento consolidado

Data de emissão	Versão	Descrições das alterações realizadas
28/02/24	AC	Versão final
25/02/24	AB	Revisão do documento, inserindo novas contribuições, etc.
23/02/24	AA	Versão inicial

11 Execução e aprovação

Elaborado por:

Revisado por:

Aprovado por:

Data da emissão: 28/fev/2024