Mateus Gonçalves Silva

Digital Design of a Forward Error Correction System for IEEE 802.15.7

Belo Horizonte

2020

Mateus Gonçalves Silva

Digital Design of a Forward Error Correction System for IEEE 802.15.7

Dissertation submitted to the Graduate Program in Electrical Engineering at Universidade Federal de Minas Gerais, in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

Supervisor: Prof. Dr. Ricardo de Oliveira Duarte

Belo Horizonte

2020

Silva, Mateus Gonçalves.
S586d Digital design of a forward error correction system for IEEE 802.15.7 [recurso eletrônico] / Mateus Gonçalves Silva. – 2020.
1 recurso online (149 f. : il., color.) : pdf.

Orientador: Ricardo de Oliveira Duarte.

Dissertação (mestrado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Anexos: f. 127-149.

Bibliografia: f. 121-126.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Sistemas digitais - Teses. 3. Propriedade intelectual - Teses. I. Duarte, Ricardo de Oliveira. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica: Biblioteca Profº Mário Werneck, Escola de Engenharia da UFMG.

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Mateus Gonçalves Silva

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 15 de dezembro de 2020.

Por:

Prof. Dr. Ricardo de Oliveira Duarte - Orientador
DELT (UFMG)

Prof. Dr. Davies William de Lima Monteiro
DEE (UFMG)

Prof. Dr. Frank Sill Torres
DELT (UFMG)

Prof. Dr. Dalton Martini Colombo
DEE (UFMG)

Agradecimentos

Esta dissertação de mestrado é um resultado de um processo de formação acadêmica e cidadã fornecido pela Universidade Federal de Minas Gerais (UFMG), iniciado nos tempos de graduação. Por isso sou grato à UFMG pelo suporte dado durante esses anos e aos seus membros que acreditam em um ensino público, de qualidade e inclusivo. Em especial, agradeço ao Prof. Dr. Ricardo de Oliveira Duarte, que me acompanhou durante a maior parte deste processo como professor, orientador e amigo. Também agradeço aos alunos de iniciação científica Elisa Santos Bacelar e Gabriel Ricardo Thomaz de Araújo pelo esforço e dedicação empenhados na pesquisa e pela contribuição fundamental para este trabalho.

Estendo a minha gratidão aos professores do OptMA Lab, principalmente aos Prof. Dr. Davies William de Lima Monteiro, Prof. Dr. Frank Sill Torres, e Prof. Dr. Dalton Martini Colombo que aceitaram compor a banca de avaliação deste trabalho. Agradeço também à Cadence Design Systems pelo suporte dado a mim durante este período de pesquisa.

Agradeço o suporte de todos os meus familiares e amigos próximos. Principalmente, da minha amável companheira Lívia Teixeira Duarte, por seu cuidado e incentivo dispensados a mim durante esta jornada. Dedico esta dissertação a minha mãe (Célia Regina Gonçalves) e ao meu pai (Jairo da Silva), que sempre me ampararam com muito amor e resiliência.

Acknowledgements

This master thesis is a result of a process for academic and citizen formation carried out by the Universidade Federal de Minas Gerais (UFMG), started at the undergraduate period. For that reason, I am grateful to UFMG for the support provided over these years and to its members who believe in a public, high quality and inclusive education. In particular, I thank the Prof. Dr. Ricardo de Oliveira Duarte, who guided me during most of this process as a teacher, advisor and friend. I also thank the undergraduate students Elisa Santos Bacelar and Gabriel Ricardo Thomaz de Araújo for their effort and dedication committed to the research and for their fundamental contribution to this work.

I extend my gratitude to the professors at OptMA Lab, especially to Prof. Dr. Davies William de Lima Monteiro, Prof. Dr. Frank Sill Torres, and Prof. Dr. Dalton Martini Colombo who agreed to serve on my master's thesis evaluation committee. I also thank Cadence Design Systems for the support provided to me during this research period.

I appreciate the support of all my family and close friends – especially from my lovely partner Lívia Teixeira Duarte, for the affection and encouragement she gave me during this journey. I dedicate this master thesis to my mother (Célia Regina Gonçalves) and my father (Jairo da Silva), who always supported me with love and resilience.

“Cuando uno busca algo, no debe ni soñar en encontrarlo por azar,
por lo menos dentro de un plazo determinado. Porque uno de los
tantos chistes del azar es, justamente, escondernos lo que buscamos,
y hacernos encontrar lo que no buscamos, o que ya no buscamos.”
(El sótano, Mario Levrero)

Abstract

Visible Light Communication (VLC) is an emerging field that has attracted attention in recent times, and it has been proposed as a complement or even an alternative to the conventional RF systems. The motivation for it is that the latter is suffering from a phenomenon known as RF spectrum crunch - due to the overuse of wireless communication in user-end applications. The first global attempt to standardize VLC was promoted by IEEE 802.15.7, which specifies the Physical (PHY) and Media Access Control (MAC) layers for short range VLC. It has three PHY layers (I, II, and III) with thirty operating modes suitable for a wide range of noisy channel conditions. The main element of them is the Forward Error Correction (FEC) component, which defines a set of error control techniques Reed Solomon (RS) codes, Interleaving, and Convolutional Codes (CC) employed to improve the capacity of the transmission channel. The goal of this master thesis is to propose a digital system that implements a FEC compliant with IEEE 802.15.7. The main outcome of this work is an open access Intellectual Property (IP) Core for the FEC, followed by a comprehensive explanation of its related Register Transfer Level (RTL) architecture. Most attempts for implementing a IEEE 802.15.7 compliant system is targeted to prototype applications in embedded platforms, whereas dedicated digital devices are more appropriate for the hardware realization of PHY layers. Moreover, the availability of reliable IP Cores for communication such as the FEC and its base blocks is scarce. These facts corroborate the demand for the intended work. Verification and synthesis of the resulting IP Core are carried out by both Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC) flows, and their results for size, timing and power consumption are analyzed and cross validated. IEEE 802.15.7 requirements for throughput and latency are also checked for the FEC IP Core, and they are fulfilled for most operating modes at the target device technologies. Improvements for the digital design architecture and methodology of the FEC IP Core are discussed at the end of this thesis, enabling opportunities for future academic and development projects.

Keywords: Digital Systems, Error correction code, IEEE 802.15.7, Intellectual Property, Visible Light Communication

Resumo

Comunicação por luz visível (VLC Visible Light Communication) é um campo emergente que tem chamado atenção nos últimos tempos, sendo proposto como um complemento ou mesmo uma alternativa aos sistemas de Rádio Frequência (RF) convencionais. A motivação para isso é que estes últimos sofrem de um fenômeno conhecido como RF spectrum crunch devido ao uso exagerado de comunicação sem fio em aplicações de usuário final. A primeira tentativa global de normatizar o VLC foi promovida pelo IEEE 802.15.7, que detalha as camadas física (PHY - Physical) e de Controle de Acesso ao Meio (MAC Media Access Control) para o VLC de alcance curto. O padrão especifica três camadas PHY (I, II, and III) com trinta modos de operação que são adequados para uma ampla gama de cenários de canal ruidoso. O principal elemento dessas camadas é o Forward Error Correction (FEC), que define um conjunto de técnicas de controle de erro códigos Reed-Solomon (RS), Interleaving, Códigos Convolucionais (CC) utilizado para melhorar a capacidade de transmissão do canal. O objetivo desta dissertação de mestrado é propor um sistema digital que implementa um FEC compatível com o IEEE 802.15.7. O principal resultado deste trabalho é uma propriedade intelectual (IP Intellectual Property) de livre acesso para o FEC, acompanhada de uma explicação detalhada da sua arquitetura Register Transfer Level (RTL). Na literatura, a maioria dos esforços para implementar uma camada PHY compatível com o IEEE 802.15.7 é visada para aplicações de prototipação em plataformas embarcadas, enquanto um sistema digital dedicado é mais apropriado para a implementação de camadas PHY. Além do mais, a disponibilidade de IPs confiáveis para a área de comunicação, tais como o FEC e seus blocos básicos, é escassa. Estas constatações corroboram a necessidade de um trabalho como este. A verificação e a síntese do IP resultante é feita tanto no fluxo Field-Programmable Gate Array (FPGA) quanto no Application-Specific Integrated Circuit (ASIC), e os resultados para tamanho, atraso, consumo energético são analisados juntamente com uma validação cruzada dos mesmos. Os requisitos do IEEE 802.15.7 para a taxa de transferência efetiva e a latência também são conferidos para o IP, e eles são satisfeitos para a maioria dos modos de operação, dado os dispositivos selecionados. Melhorias para a arquitetura e a metodologia do projeto digital do FEC IP são discutidas no final desta dissertação, o que habilita oportunidades para futuros projetos acadêmicos e de desenvolvimento.

Palavras-chave: Sistemas Digitais, Código de Correção de Erro, IEEE 802.15.7, Propriedade Intelectual, Comunicação por luz Visível

Contents

List of Figures

List of Tables

Acronyms

ABV Assertion-Based Verification.

AC Alternating Current.

ACS Add-Compare-Select.

ADC Analog-to-Digital Converter.

ALM Adaptative Logic Module.

ALUT Adaptive Look-up Table.

ASIC Application Specific Integrate Circuits.

BCH Bose and Ray-Chaudhuri.

BER Bit Error Rate.

BM Branch Metric.

CAGR Compound Annual Growth Rate.

CC Convolutional Codes.

CCA Clear Channel Assessment.

CD Compact Disk.

CDR Clock and Data Recovery.

CEx Counter Example.

CMOS Complementary Metal Oxide Semiconductor.

CoI Cone of Influence.

CONN Connectivity.

COTS Commercial Off-The-Shelf.

CPU Central Processing Unit.

CRC Cycle Redundancy Check.

CS Compensation Symbols.

CSK Color Shift Keying.

CSV Comma-Separated Values.

DAC Digital-to-Analog Converter.

DC Direct Current.

DDR Double Data Rate.

DME Device Management Entity.

DS Data Subframe.

DSP Digital Signal Processor.

DUT Design Under Test.

DVD Digital Versatile Disk.

$Eb/N0$ Energy per Bit to Noise Power Spectral Density Ratio.

ECC Error Correcting Code.

EDA Electronic Design Automation.

FEC Forward Error Correction.

FLP Fast Locking Pattern.

FPGA Field Programmable Gate Array.

FSM Finite State Machine.

FSO Free-Space Optical.

FV Formal Verification.

G Groups.

GF Galois Field.

GP Generator Polynomial.

GPIO General Purpose Input/Output.

GPL General Public License.

GSC Generic Standard Cell.

HCS Header-Check Sequence.

HDL Hardware Description Language.

HPF High-Pass Filter.

I/O Input/Output.

IC Integrated Circuit.

ID Identifier.

IoT Internet of Things.

IP Intellectual Property.

ISO International Organization for Standardization.

IWLS International Workshop on Logic and Synthesis.

JEITA Japan Electronics and Information Technology Industries Association.

LAB Logic Array Block.

LD Laser Diodes.

LDPC Low-density parity-check.

LED Light-Emitting Diode.

LiFi Light-Fidelity.

LLC Logic Link Control.

LoS Line of Sight.

LPDDR Low-Power Double Data Rate.

LUT Look-up Table.

LVDS Low-voltage Differential Signaling.

MAC Medium Access Control.

MCS Modulation and Coding Scheme.

MIT Massachusetts Institute of Technology.

MLAB Memory Logic Array Block.

MTC Machine Type Communication.

NMOS N-type Metal-Oxide-Semiconductor.

OCC Optical Camera Communication.

OFDM Orthogonal Frequency Division Multiplexing.

OOK On-Off Keying.

OSI Open System Interconnection.

OWC Optical Wireless Communications.

P/S Parallel/Serial.

PCB Printed Circuit Board.

PCI Peripheral Component Interconnect.

PD Physical-layer Data.

PHR Physical-layer Header.

PHY Physical.

PIB Physical-layer personal-area-network Information Base.

PLME Physical-layer Management Entity.

PM Path Metric.

PMOS P-type Metal-Oxide-Semiconductor.

PPDU Physical-layer Data Unit.

PS Parallel-to-Serial.

PSDU PHY Service Data Unit.

RAM Random-Access Memory.

RF Radio Frequency.

RIFS Reduced Interframe Space.

RISC Reduced Instruction Set Computer.

RLL Run-Length Limit.

ROM Read-Only Memory.

RS Reed-Solomon.

RTL Register Transfer Level.

S/P Serial/Parallel.

SAP Service Access Point.

SDC Synopsis Design Constrains.

SDR Software Defined Radio.

SHR Synchronization Header.

SIFS Short Interframe Space.

SNR Signal to Noise Ratio.

SoC System-on-Chip.

SP Serial-to-Parallel.

SPI Serial Peripheral Interface.

SRAM Static Random-Access Memory.

SSCS Service-Specific Convergence Sublayer.

STA Static Timing Analysis.

SVA System Verilog Assertions.

TDP Topology Dependent Pattern.

TIA Trans-Impedance Amplifier.

TSMC Taiwan Semiconductor Manufacturing Company.

UPF Unified Power Format.

USB Universal Serial Bus.

USRP Universal Software Radio Peripheral.

VCD Value Change Dump.

VLC Visible Light Communication.

$VPAN$ Visible-light communication Personal Area Network.

$VPPM$ Variable Pulse Position.

$WiFi$ Wireless Fidelity.

$WPAN$ Wireless Personal Area Network.

WQI Wavelength Quality Indication.

1 Introduction

The demand for wireless communication has been imposed by the increasing number of mobile devices which require Internet connection and the strict data throughput requirements of their latest applications. As proof of that, a study carried out in 2017 [1] estimated that the Compound Annual Growth Rate (CAGR) of wireless traffic has been 60% during the last ten years. Moreover, if the same pace is maintained for the next two decades, the bandwidth request would rise by 12000 times if the same spectrum efficiency is assumed [2]. The expectations regarding Internet of Things (IoT) and Machine Type Communication (MTC) technologies indicate that such trend is realistic. Radio Frequency (RF) spectrum used by Wireless Fidelity (WiFi) has become unable to address the next mobile data traffic demands [3].

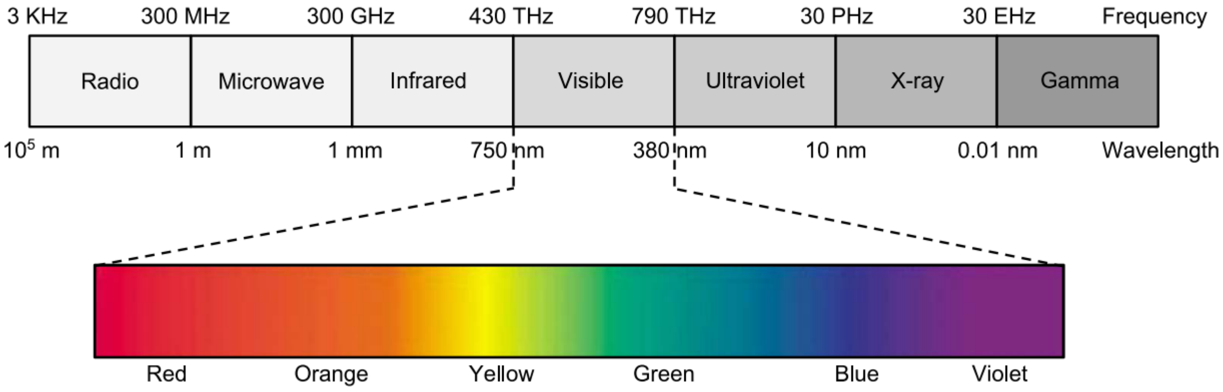
The aggressive exploration of spatial reuse has been the main approach adopted to mitigate the spectrum shortage issue, and it caused a massive shrinkage of cell sizes in cellular communication. For instance, the cell radius in 2G systems was 35 km, whereas it is expected 25 m for 5G to achieve higher data densities. However, this strategy is bound by infrastructure costs for connecting all access points to the core network and the augment of intra-cell interference with shorter cell radius [2]. Hence, the exploration of high frequency spectra will play a major role to meet the next requirements of mobile industry. Visible light spectrum (Fig. 1) has gained attention by the academic community since its use as communication medium - known as Visible Light Communication (Visible Light Communication (VLC)) - could abruptly increase data traffic capacity. Speeds around 100 Gbps have been demonstrated by academic researches [4, 5]. VLC poses as a reasonable candidate for being part of next generations of mobile services due several aspects:

- Energy-efficient Light-Emitting Diode (LED) light bulbs have been widely adopted to replace incandescent and fluorescent lamps. They could easily provide illumination and data communication functions.
- Visible light spectrum is license-free.
- Improvement in data densities around three orders of magnitude in comparison to current RF technologies [6].
- Unlike RF, visible light cannot pass through walls, and this intrinsic feature enhances physical layer security.
- It is allowed to be used in environments where RF is banned (i.e petrochemical plants and oil platforms) [2].

- It can be employed as a complement to WiFi technologies without major issues [7].

Despite of several advantages, VLC faces some challenges for its wide deployment in wireless communication systems. According to the Friis free space equation, higher frequencies mean path loss raises in electromagnetic wave propagation [8]. Hence, VLC systems require enhancements in the probability of Line of Sight (LoS). Moreover, visible light source used for communication are subjected to interference from sunlight and other light sources. Therefore, VLC transceivers must be designed with coding techniques that are able to mitigate possible errors occurred in the communication channel.

Figure 1 – Visible light in electromagnetic spectrum scale ([9], p. 2048)



1.1 VLC: Related Terminologies and Standards

Technically, VLC categorizes any type of communication where data is transmitted via visible spectrum, which contains wavelengths ranging from 380 nm to 750 nm. A requirement present in VLC is that data transfer should not modify light source aspects perceived by human vision such as brightness and flickering. There are other terminologies that are similar to VLC, but they have some differences:

- Optical Wireless Communications (OWC): It encompasses the whole optical band - infrared, visible, and ultraviolet - for wireless communication [10]. Then, VLC is a subset of OWC.

- Free-Space Optical (FSO) Communication: It is considered as a specialization of OWC in which it focuses on high data rate communication between two fixed points over distances up to several kilometers. Transmission occurs in a medium without barriers and very narrow laser beams are used in FSO systems [11].

- Light-Fidelity (LiFi): It is not restricted to visible light, and it also accepts infrared and ultraviolet. LiFi provides a complete proposal of a wireless networking system, which includes a bi-directional multiuser communication with high data rate and multiple access points to enable full user mobility [3]. This term arises to contrast with original stardards and systems elaborated for VLC, which has been conceived as a point-to-point data communication technique with lower speeds.

The standardization process for VLC is in continuous progress. The first initiative was carried out in 2003 by the VLC Consortium in Japan. It resulted in two standards for VLC systems included in Japan Electronics and Information Technology Industries Association (JEITA): CP-1221 and CP-1222 [12]. In 2011, IEEE 802.15.7 [13] (Short-Range Wireless Optical Communication Using Visible Light) was released, and it specifies Physical (PHY) and Medium Access Control (MAC) layers for VLC. This first IEEE standard for VLC encompasses thirty operating modes among three PHY layers with On-Off Keying (OOK), Variable Pulse Position (VPPM), and Color Shift Keying (CSK) as modulation scheme options. In order to cope with LED Identification (LED-ID), Optical Camera Communication (OCC) and High Rate Power Distance Communication, IEEE 802.15.7m was launched in 2014 with three new PHY modes with several modulation schemes [14, 15]. After that, the standardization for High Rate Power Distance Communications has been moved to IEEE 802.15.7 (Multi-Gigabit/s Optical Wireless Communications), which added two new Orthogonal Frequency Division Multiplexing (OFDM)-based PHY modes. Recently, IEEE 802.11bb has been created to deal with light-based systems, aiming to define a single ecosystem for chipset vendors, specify LiFi network infrastucture, and establish a cooperation between LiFi with WiFi standards [16].

1.2 Forward Error Correction (FEC)

FEC defines a set of error control techniques targeted to improve the capacity of a transmission channel by incorporating to the source data some designed redundant information [17]. It is employed in applications in which re-transmission is costly or impossible such as satellite and broadcast communications. The FEC encoding process builds an output codeword that is characterized by some specific rule which restricts their possible values. As a consequence, the FEC decoding process relies on such rule to extract the most probably correct codeword from the received data. The methods applied for FEC fall into two major categories:

- Block codes: Data stream is broken up into a series of fixed-size packets of n bits in length. If k bits are allocated to the original source data where $k < n$, then the code rate is defined by n/k, which is commonly represented by the notation (n, k). Block codes are implemented by hard-decision algorithms since their produced results are based on data input with strict values (‘0’ or ‘1’ in binary notation) [18]. Examples of this category are Hamming, Bose and Ray-Chaudhuri (BCH), Reed-Solomon (RS) and Low-density parity-check (LDPC) codes.

- Convolutional Codes (CC): Data stream is treat as a string of consecutive bits constrained by a maximum length of K bits [19]. The encoder scheme is conceived by m shift registers, and the output is generated by a linear combination of selected shift register outputs which maps to a generator polynomial [17]. They are characterized by a coding rate of k/n, which means that n output bits will be produced for each k input bits. Viterbi algorithm is the most common decoding methodology. It is based on the maximum likelihood decoding technique and operates with hard or soft-decision methods [18]. Soft decision algorithms use a multi-bit quantization to represent the probability of a single data bit be ‘0’ or ‘1’.

Conventional Error Correcting Code (ECC) algorithms may be used in VLC systems. IEEE 802.15.7 prescribes three operational ECC blocks for its FEC system which have been widely used in other applications: RS codes, Interleaver, and CC. The main particularity of VLC is that the light source used to transmit data should not flicker or change its intensity; however, these ECC methods employed in the FEC system does not guarantee an even distribution of 0’s and 1’s for the encoded frame [20]. Hence, it may cause disturbances in the light source that are perceived by human vision depending on the chosen modulation scheme. IEEE 802.15.7 mitigates them by using Compensation Symbols (CS) within the encoded frame and Run-Lenght Limit (Run-Length Limit (RLL)) coding strategies that balances the distribution of 0s and 1s within the encoded bitstream.

1.3 Motivation and Goals

VLC is a topic that has gained attention in academic and corporate fields because of the incipient RF spectrum crunch in wireless communication. Many startups like pure LiFi, Lucibel, vlncomm, OLEDCOM, and others have been established to develop dedicated VLC and LiFi systems [21]. Large companies such as Signify, General Electric and Panasonic have also invested in this area. Most frameworks for VLC are implemented in software and installed into embedded platforms. They are more prototypes than final user solutions since PHY layers should have a dedicated implementation in hardware to meet throughput requirements. For IEEE 802.15.7, there are very few works on developing a digital system for PHY layer.

The goal of this master thesis is to explore the PHY layer of IEEE 802.15.7 by developing a digital system for its FEC, which is the most complex block of that layer.

In order to contextualize the FEC for IEEE 802.15.7, the main relevant aspects of the standard for it are covered in this document and related works are analyzed. The deliverable of this work is an open access Register Transfer Level (RTL) Intellectual Property (IP) of the FEC specified in IEEE 802.15.7.

IEEE 802.15.7 is the starting point to understand basic premises of a PHY layer for VLC systems. That is reason for sticking to this standard instead of exploring the latest updates for VLC standards, which still require some level of maturation. Moreover, the operational blocks of the FEC system implement well known algorithms used in other applications or standards. They were designed to be parameterizable, which enables using them in other contexts that require configuration setups other than the ones specified in IEEE 802.15.7.

This work covers implementation steps that are common in every project of a RTL IP. The resulting Hardware Description Language (HDL) files adopted certain coding style, signal name convention, and sub-block reuse. The verification methodology of the developed IP has also been detailed in this thesis. Performance metrics of the FEC system are collected for Field Programmable Gate Array (FPGA) and Application Specific Integrate Circuits (ASIC) flows and analyzed in this work. Such results are important to assess the proposed IP and certify that there is not any architectural flaws in its design.

The result of this master thesis is the first step to enable an open IP for IEEE 802.15.7 PHY layer, which is something that is not available to the academic community yet. The FEC is the most complex system within this PHY layer, then the effort required to obtain an end-to-end solution for IEEE.802.15.7 will be significantly lower after this work. VLC is an emerging research field that lacks an adequate digital framework for its PHY layer, which is a prerequisite for any feasible network connection. Several subsequent projects are enabled by this master thesis, and they present different levels of complexity and activity types.

1.4 Thesis Roadmap

This master thesis is organized as follows: Chapter 2 elucidates key aspects of the IEEE 802.15.7 PHY layer that are required to comprehend the developments of this work; Chapter 3 is literature review related works, which provides a better picture about the novel contribution of this work; Chapter 4 carries out a detailed analysis of the RTL architecture of the proposed digital system to perform FEC; Chapter 5 analyzes the results the implemented RTL IP; and Chapter 6 delineates the future works and presents the conclusions of this master thesis.

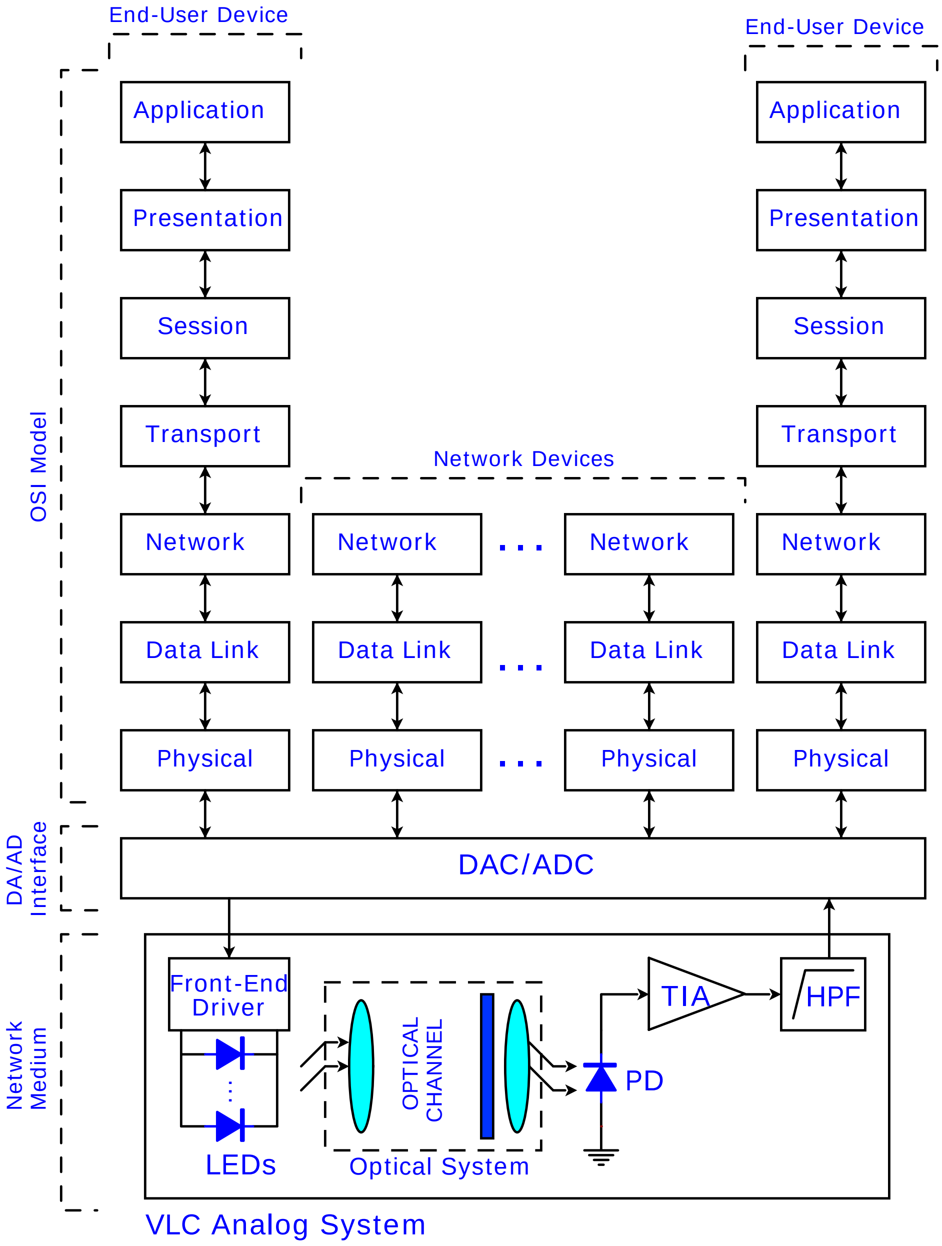
2 IEEE 802.15.7: Guided Review

This chapter reviews aspects of IEEE 802.15.7 that are relevant for understanding architectural implications for PHY layer design and implementation details of the FEC specified by the standard. It provides a guided overview of IEEE 802.15.7 based on what is required to understand the development of this master thesis.

Before exploring the key aspects of IEEE 802.15.7, it is relevant to cover the general process to exchange data via VLC in order to understand where the standard fits in. Fig. 2 depicts all communication blocks required to interconnect to end-user devices in the network using VLC. The Open System Interconnection (OSI) model is a International Organization for Standardization (ISO) standard released in 1979 responsible for establishing a unified communication architecture based on seven well-defined layers. Such layered model is a reference until today because it simplifies the understand of the network architecture, creates behavioral domains that facilitate implementation and troubleshooting of the network and guarantees better consistency in functions and protocols [22]. Hubert Zimmermann, member of the OSI standardization group, provides an overview of the seven layers of the OSI model in [23]:

- *Application* serves the end user by providing protocols appropriate to application management tasks (e.g initiate, maintain, and terminate connections for data transfer).
- *Presentation* provides the set of services for the management, display, and control of structured data used by *Application* layer.
- *Session* establishes an administration service mechanism to (un)bind presentation entities and controls their data exchange.
- *Transport* provides an universal transport service in a reliable and cost-effective way.
- *Network* allows functional and procedural means to exchange network service data units considering routing and switching aspects of the network connection.
- *Data Link* provides means to establish, maintain and release data links between network entities.
- *Physical* offers mechanical, electrical, functional, and procedural characteristics to establish maintain, and release physical connections between data link entities.

Figure 2 – Communication blocks required to transfer data between two end-user devices.



IEEE 802.15.7 covers *Data Link* and *Physical* layers of the OSI model as they are tied to the characteristics of the network medium used for data transfer. A Digital-to-Analog Converter (DAC) or a Analog-to-Digital Converter (ADC) are required to bind the digital blocks that implement the OSI model and the analog transducer system. The VLC analog system requires LEDs, controlled by a front-end driver circuit that implements a transconductance amplifier, to transmit data and a Physical-layer Data (PD) to detect the light source, which demands a Trans-Impedance Amplifier (TIA) and a High-Pass Filter (HPF) to amplify its voltage and remove low-frequency components of the received signal.

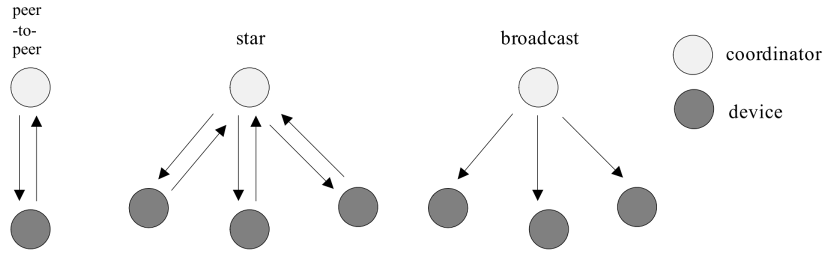
Also, an optical system (lens and filters) may required to adjust the light beam in order to enhance optical communication performance. Between the end-user devices there might be some network devices (e.g repeaters, hubs, switches, and routers) used to transport data transferred by the end-user devices. They are devices that implements at most the three lowest layers of the OSI model and may require other network mediums (e.g optical fiber) that are omitted in Fig. 2.

A key aspect in VLC is that the data transmitted by LEDs and Laser Diodes (LD) cannot cause any light source disturbances that may be detected by the human eye. Hence, IEEE 802.15.7 introduces the concept of Visible-light communication Personal Area Network (VPAN), which is a branch of Wireless Personal Area Network (WPAN). This specialization is required because it aims to merge lighting and data communication functions on the same device, which brings up new requirements such as dimming and visibility support that impact the specification of *Data Link* and *Physical* layers for VLC.

2.1 Topologies Types

VPAN defines three topologies: peer-to-peer, star, and broadcast (Fig. 3). There is always a VLC device in the network which assumes the coordinator role, which manages the VPAN and is usually always powered on. Each device has a unique 64-bit address, which may be shortened to 16-bit address when it is associated with a coordinator. In peer-to-peer topology, the coordinator role is assigned to the first device that starts to transmit data. Networks structured in a star topology are identified by a VPAN number not used by any other surrounding VPAN networks. This Identifier (ID) is held by the coordinator, which allows newcomer devices to join the network. Broadcast topology does not require a network formation, and transmissions of unidirectional signals may be captured by any device in their range.

Figure 3 – Network topologies in IEEE 802.15.7 ([13], p. 6)



2.2 Architectural Aspects

The system architecture for IEEE 802.15.7 is organized in layers and sublayers that offers services to higher layers (Fig. 4), and such arrangement is compliant with the OSI model. The standard specifies the PHY layer and the MAC sublayer with its accessor Service-Specific Convergence Sublayer (SSCS). Logic Link Control (LLC) layer is defined by IEEE 802.2, and the upper layers are not covered by IEEE 802.15.7. The Device Management Entity (DME) is a control interface that sets up dimmer related attributes in MAC and PHY layers such as the selection of optical light sources and photodetectors. These devices are identified as cells, and they communicate with the PHY switch via OPTICAL Service Access Point (SAP) interface, which is not specified by the standard.

The implementation of the VPAN architecture should consist of a hardware-firmware co-design. PHY layer is usually implemented in hardware since it deals with time-critical signal processing functions. MAC layer should be implemented as a firmware, though, because it implements a complex structure of protocol logics, and a software approach facilitates its realization. However, it is required to implement a MAC driver that communicates to the PHY layer using the interfaces provided by PD-SAP and Physical-layer Management Entity (PLME)-SAP. As DME interacts with many layers of the network model, it should be implemented similarly to MAC. Finally, a DAC/ADC must be placed between the PHY switch and the OPTICAL-SAP because this is the boundary between analog and digital worlds.

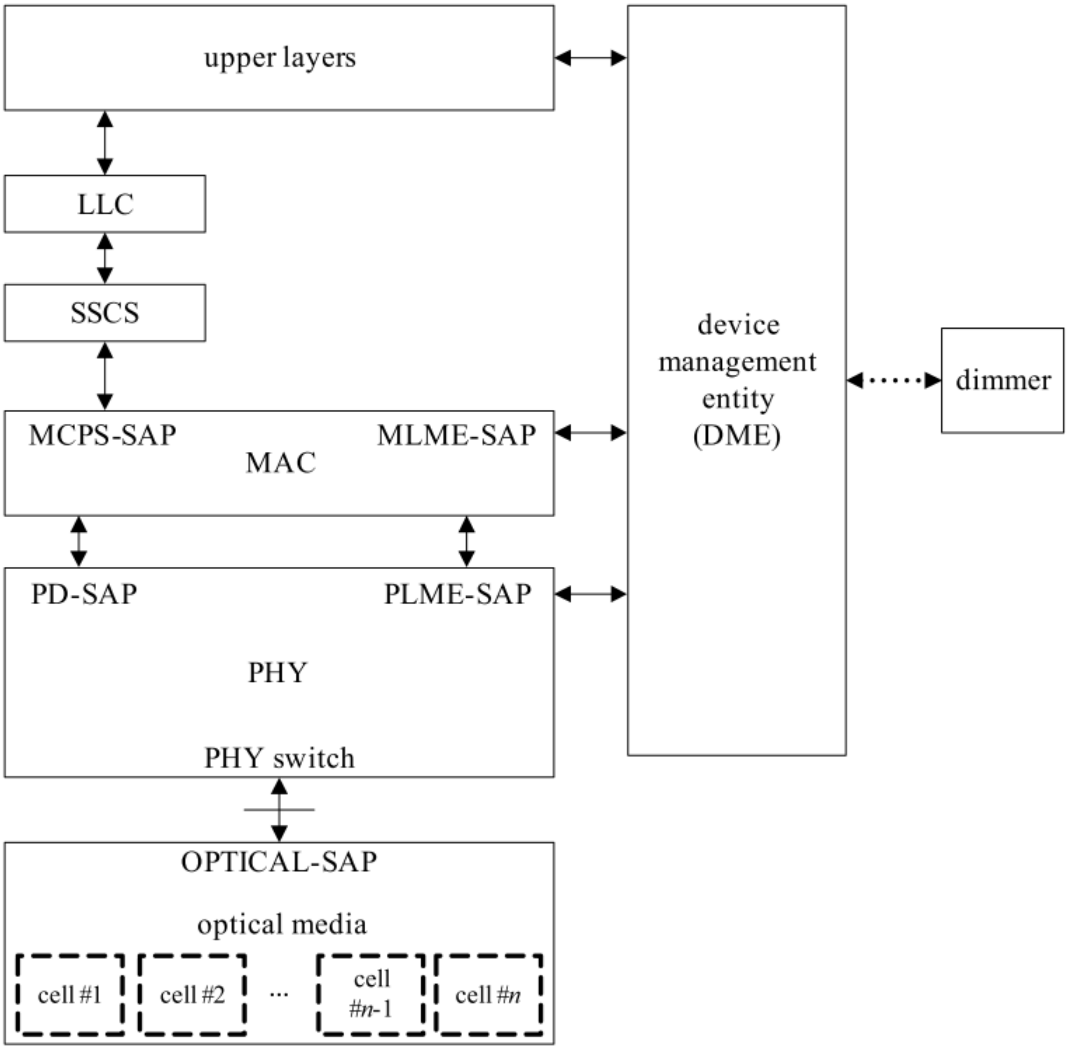
2.3 PHY Layer

The following functions are assigned to PHY layer:

- Activation and deactivation of the VLC transceiver
- Wavelength Quality Indication (WQI) for received frames
- Channel Selection
- Data transmission and reception
- Error Correction
- Frame Synchronization

PHY layer in IEEE 802.15.7 supports three PHY layers, which are characterized by their defined operating modes (section 2.3.1). Each one is focused on a different communication scenario:

Figure 4 – VPAN device architecture ([13], p.8).



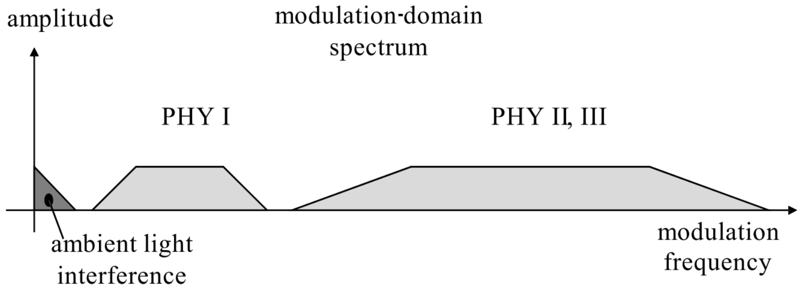
- PHY I: Intended for outdoor usage with low data rate applications (hundreds of kb/s).

- PHY II: Intended for indoor usage with moderate data rate applications (tens of Mb/s).

- PHY III: Indicated for applications using color-shift keying (CSK) that have multiple sources and detectors (tens of Mb/s).

As highlighted by Fig. 5, PHY I and PHY II occupy different spectral regions in the modulation-domain spectrum, and Frequency Division Multiplexing (FDM) mechanism may be used as a coexistence mechanism for both PHY layers. The same scenario is applicable for PHY I and PHY III. For PHY II and PHY III, their modulation frequencies overlap, then PHY III is able to detect PHY II communication to avoid such prohibitive overlap.

Figure 5 – Frequency spectrum occupied by each PHY layer ([13], p.9).



2.3.1 Operation Modes

PHY I, PHY II and PHY III specify nine, fourteen, and seven operating modes with data rates ranging from 11.67 kb/s to 96 Mb/s. Three modulation schemes are used: OOK, VPPM, and CSK. OOK is the simplest modulation method, where light source in ‘ON’ and ‘OFF’ state represents ‘1’ and ‘0’ logic values (Fig. 6a). VPPM models logic values by the position of the pulse within the optical clock cycle, and its variable pulse provides an inherent dimming control mechanism (Fig. 6b). CSK codes information bits into different color wavelengths and requires three light sources to emit red, green, blue colors. IEEE 802.15.7 specifies seven band ranges to encode data (Fig. 6c) with constellations of sizes 4, 8, and 16 symbols.

The RLL techniques are applied in PHY I and PHY II and aim to bound the length of repeated bits in a stream, and it eases clock recovery at the receiver. They also help in Direct Current (DC) balance and light flicker mitigation. Manchester mode encodes logic values into transitions ‘0’ to ‘1’ and ‘1’ to ‘0’, being a self-clocking signal without a DC component (Fig. 7a). Because of that, data rate is reduced by two and light source brightness is constant at 50%. CSs are required then to adjust the source light brightness to the desired intensity. 4B6B mode converts 4 to 6 bits, where the most uneven bit value distributions for 6-bit words are excluded (Fig. 7b). 8B10B uses the same approach of 4B6B, but it converts 8 to 10 bits.

Multiple optical clock rates are provided to allow the application of a broader class of optical transmitters. MAC layer defines an optical clock-rate selection procedure during device discovery phase. The clock rate does not change during the transmission of different portions of the PHY frame: the preamble, used to synchronize the receiver with the clock rate required by the incoming message, the header, which contains fields required to configure the PHY device to receive the message, and the payload, which is the

Figure 6 – Modulation schemes present in IEEE 802.15.7 PHY layers: OOK (a), VPPM (b), and CSK (c) ([24], p. 42 for (a) and (b) [13], p. 254 for (c))

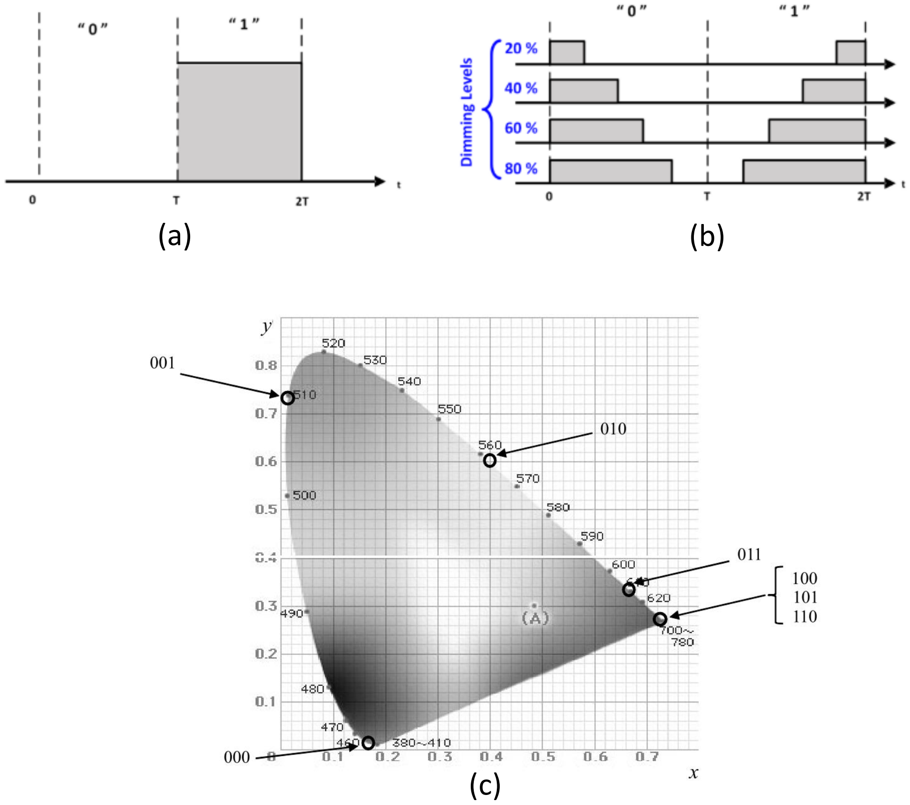
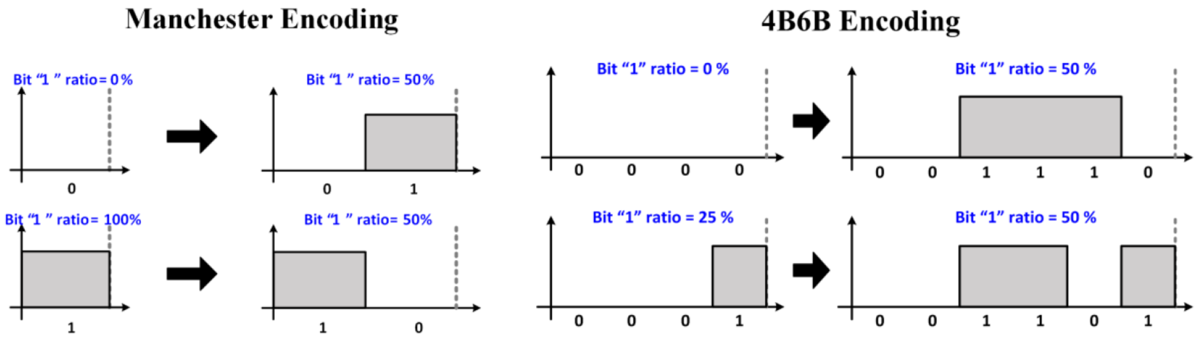


Figure 7 – IEEE 802.15.7 RLL coding conversion examples ([24], p. 41).



message itself that is processed by the MAC layer. However, header should be sent at the lowest available data rate of the selected optical clock rate. Also, the preamble does not require any data encoding. The preamble and the header shall be transmitted using OOK.

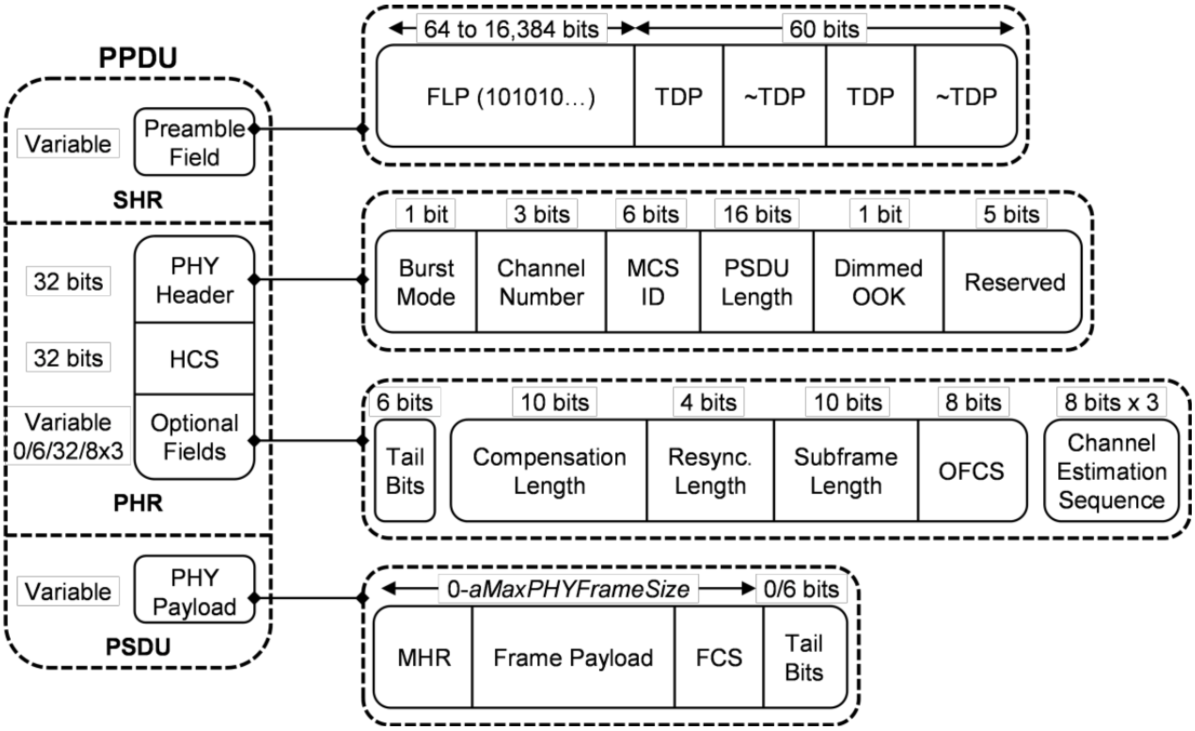
Annex A summarizes the modulation and the RLLs techniques required by each operating mode in PHY I, II, and III (Fig. 56, 57, and 58). There are ten possible arrangements for FEC specified by the thirty operating modes. RS codes are used in all

PHY layers, and PHY I also counts on CC since it is potentially more exposed to channel interference. More explanation about RS and CC will be provided in the next chapters. The goal of this master thesis involves providing a digital system for IEEE 802.15.7 FEC that is able to harmonize all possible combinations for FEC present in the standard. The optical clock rate - the frequency at which the data is clocked out to the optical source - of each operating mode is a constrain which defines the minimum data rate for each arrangement required for the IEEE 802.15.7 FEC.

2.3.2 Frame Structure

All fields of PHY frame are presented in Fig.8. There are 5 major fields, and each composes a group of subfields.

Figure 8 – Physical-layer Data Unit (PPDU) strucuture ([25], p. 9).



2.3.2.1 Synchronization Header (SHR)

This field used for synchronization of the optical clock at the receiver end. Fast Locking Pattern (FLP) is a variable sequence of alternate ones and zeros which is identified by the Clock and Data Recovery (CDR) circuit - the entry point for the incoming message at the receiver. Topology Dependent Pattern (TDP) is a pattern of 15 bits that identifies the network topology: only visibility, peer-to-peer, star, and broadcast. Four repetitions of this pattern are sent, and every subsequent repetition is inverted to provide DC balance.

2.3.2.2 Physical-layer Header (PHR)

This field represents the PHY Header, which provides information to configure the VLC PHY device that is receiving the message. The first bit informs if ‘Burst mode’ is enabled. The subsequent three bits represent the ‘Channel number’ used in PHY III for CSK modulation. The subfield ‘MCS ID’ is a 6-bits representation for each Modulation and Coding Scheme (MCS) operating mode specified in Fig. 56, 57, and 58. The ‘PSDU Length’ is informed by a 16-bit word since the payload - PHY Service Data Unit (PSDU) - may assume maximum values (in bytes) of 1023 for PHY I or 65535 for PHY II and PHY III. The next bit informs if ‘Dimmed OOK extention’ is enabled, which basically determines the existence of the optional field. The 5 last bits are reserved for future use.

2.3.2.3 Header-Check Sequence (HCS)

Every PHY header shall be protected with a 2 octet Cycle Redundancy Check (CRC).

2.3.2.4 Optional Fields

As illustrated by Fig. 8, there are three possibilities for optional field. Tail bits shall be transmitted at the end of the PHY header whenever the selected operating mode uses CC in its FEC scheme. If ‘Dimmed OOK extension’ is enabled, then four more fields are added to set up the dimming mitigation support for OOK modulation. More information about them may be found in section 8.6.4 of IEEE 802.15.7. And ‘Channel Estimation Sequence’ is used by PHY III and discussed in section 12.9 of IEEE 802.15.7.

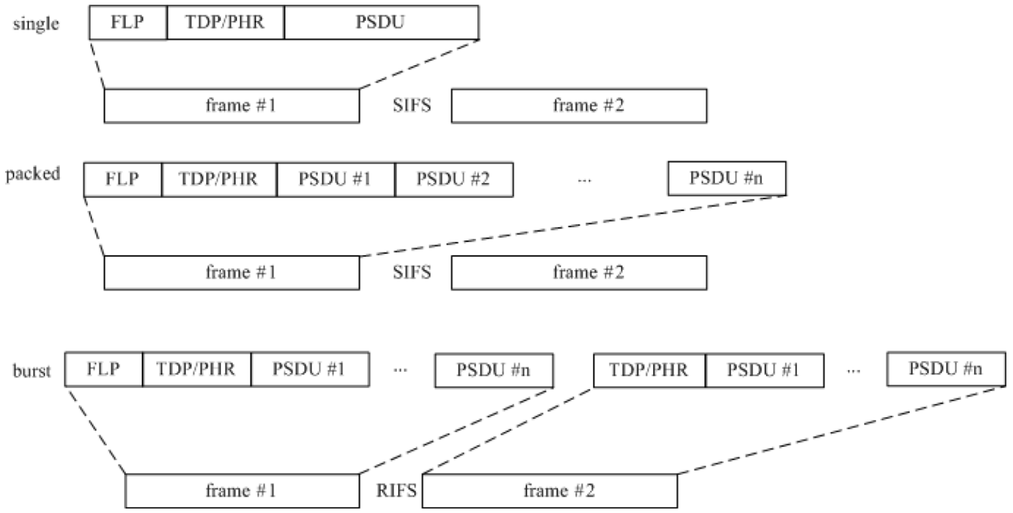
2.3.2.5 PHY Service Data Unit (PSDU)

The PSDU field represents the payload of the PHY frame. Six tail bits are added at the end of this field whenever the selected operating mode uses CC in its FEC scheme.

2.4 Data Modes

As depicted by Fig.9, IEEE 802.15.7 specifies three data transmission modes: single, packed and burst. The single mode sends one PSDU per frame and is applied for short data communication such as acknowledgments, association, and others. The packed mode contains multiple PSDU per frame to enable higher throughput. The burst mode is similar to packed mode, but it only uses FLP in the first frame and has a shorter interval between frames - Reduced Interframe Space (RIFS) instead of Short Interframe Space (SIFS). Minimum values for RIFS and SIFS are 40 and 120 optical clocks respectively, and it determines the latency of each FEC arrangement in the IP by this master thesis.

Figure 9 – Data transmission modes in PHY layer ([13], p. 218).



2.5 Dimming and Flicker Mitigation Support

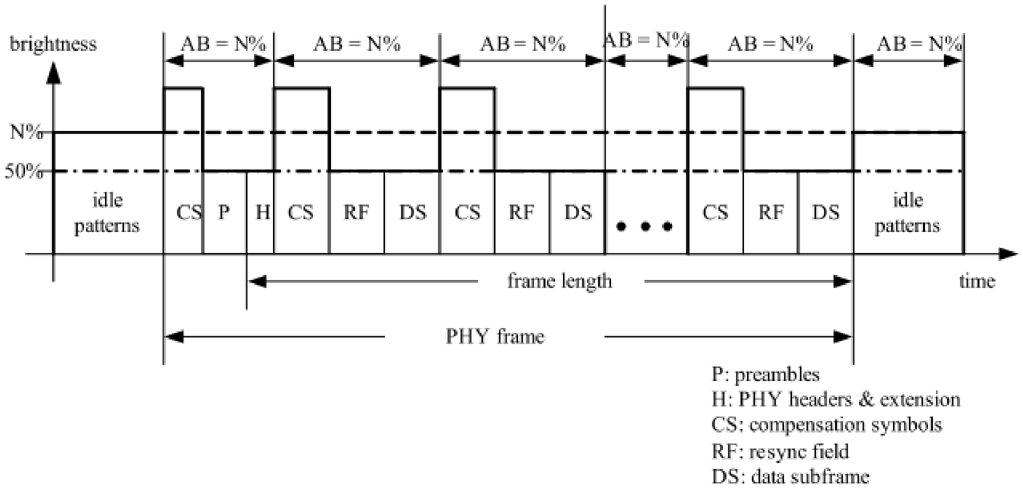
A basic requirement in IEEE 802.15.7 is to guarantee that the light source is at the desired level of brightness and does not flicker during frame transmission. The VLC system shall provide a visibility pattern during idle time. Section 8.5.1.2 in IEEE 802.15.7 highlights that patterns based on 8B10B code provides a brightness resolution of 10%, and it also explains an algorithm to achieves 0.1% dimming resolution. The only modulation scheme that is not able to adjust itself to reach the desired brightness is OOK. Then, CS must be included to control brightness as illustrated by Fig.10. Between CS and Data Subframe (DS), it is always mandatory a Resync. Field with FLP content, then the receiver can be re-synchronized to receive the next DS, which is the next message slice. With the 'Optional Fields' described by Fig. 8, it is possible to adjust the length of CS, Resync. Field and DS.

Flicker is another disturbance that shall be prevented. OOK modulation presents intra-frame flicker - which happens during data frame transmission -, and it is mitigated by using dimmed OOK mode. Inter-frame flicker happens during the transition between transmission and idle periods. Visibility pattern during idle periods are used to control brightness regardless of the VLC transceiver state.

2.6 PHY Management Service

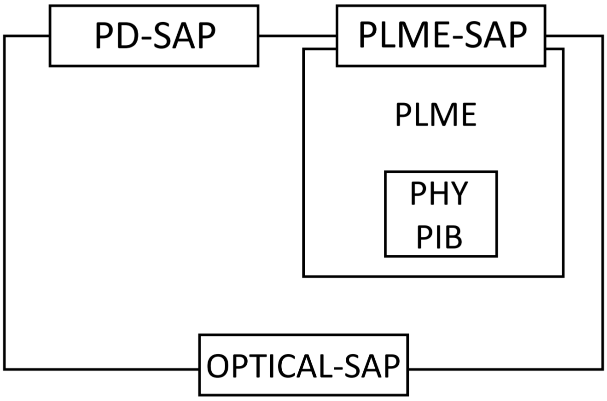
IEEE 802.15.7 defines PHY layer access interfaces for MAC sublayer and physical optical channel (Fig.11). The PHY interface with upper layer levels are split into PLME-SAP and PD-SAP. PLME-SAP provides layer management functions, whereas PD-SAP is

Figure 10 – OOK dimming mechanism in IEEE 802.15.7 PHY layer ([13], p. 14).



responsible for frame reception and transmission. OPTICAL-SAP is not specified by the standards and shall be defined by the IP developer.

Figure 11 – SAP points in PHY layer ([13], p. 231).



2.6.1 PHY Management Service

PLME-SAP defines five functions using ‘request’ and ‘confirm’ primitives:

- PLME-CCA: Performs Clear Channel Assessment (CCA) on ‘PLME-CCA.request’ and reports its results on ‘PLME-CCA.confirm’.
- PLME-GET: Receives ‘PLME-GET.request’ to obtain a PHY Physical-layer personal-area-network Information Base (PIB) attribute and returns its value on ‘PLME-GET.confirm’.

- PLME-SET: Receives ‘PLME-SET.request’ to modify a PHY PIB attribute detailed by section 9.5.2 in IEEE 802.15.7 and reports change status on ‘PLME-SET.confirm’.
- PLME-SET-TRX-STATE: Receives ‘PLME-SET-TRX-STATE.request’ to enable transmitter or receiver and disable transceiver and reports change status on ‘PLME-SET-TRX-STATE.confirm’.
- PLME-SWITCH: Uses a bit vector determined by DME to turn on/off optical cells on ‘PLME-SWITCH.request’ and reports change status on ‘PLME-SWITCH.confirm’.

2.6.2 PHY Data Service

PD-SAP defines a single function (PD-DATA) for frame transport and has 3 primitives:

- PD-DATA.request: Data transfer of PSDU from MAC sublayer to PHY to be transmitted by the transceiver.
- PD-DATA.confirm: Reports status informing the end of the transmission.
- PD-DATA.indication: Indicates data transfer from PHY to MAC sublayer.

3 Related Works

The literature presents some works dedicated to demonstrate a VLC system for IEEE 802.15.7 PHY layer. However, none of it reports a complete architectural proposal of a digital system for it. Even the FEC subsystem is not completely devised for PHY I, PHY II and PHY III in previous works. Most effort has been made to deliver prototypes for IEEE 802.15.7 using existing hardware platforms. As PHY layers are very tied to the analog transducer devices which transmit or receive a PHY frame, they should be performed by ASIC devices in order to meet throughput requirements. However, it has been found only a single work [26] describing a ASIC design of a IEEE 802.15.7 PHY I transceiver.

3.1 Prototypes for IEEE 802.15.7 PHY layer

3.1.1 Gavrincea et al. (2014)

The concept of Software Defined Radio (SDR) is employed to implement a rapid prototyping for IEEE-802.15.7 PHY I and PHY II [27]. Then, signal processing algorithms are conceived in a software layer, which requires less effort than crafting a specialized hardware for it. For this case, a hardaware subsystem is only required for ADC/DAC functions, front-end circuits and transducers. A configurable software application has been developed, where it is possible to set up all aspects of frame transmission for PHY I and PHY II. The ADC/DAC is performed by a Universal Software Radio Peripheral (USRP) device from Ettus Reserach$^{\mathrm{TM}}$ due to its superior trade-off between cost and performance. The proposed VLC system is validated with a successful video transmission and chat application over a distance of 1.5m and 4.0m, but the complete performance analysis is carried out by another complementary work [28]. This paper demonstrated that the execution of the PHY frame decoding requires only 25% of the computer processing power.Bit Error Rate (BER) is analyzed for LED-photodetector distances ranging from 0.5m to 3.5m for all PHY I OOK operating modes, and its results confirm the expectation of the more robust the FEC scheme the better the BER performance is. Also, effective data rate at 2m distance is measured, and the results are much inferior to what the standards specifies for each operating mode.

3.1.2 Hussain et al. (2015)

A VLC system for IEEE 802.15.7 PHY I is developed using SDR approach in a USRP 2920 platform [29] using system architecture similar to [27]. The main difference between them is that National Instruments$^{\text{TM}}$ LabVIEW is employed to implement the software subsystem. LabVIEW already has the required functions for the ECC methods specified by the standard, is compatible with USRPs, and provide a suitable user interface. A Luminous$^{\text{TM}}$ SST-50 white LED and a Thorlabs$^{\text{TM}}$ PDA36A-EC phodectector are used as transceivers of this VLC system. A performance analysis is carried out comparing BER against illuminance with transmitter-receiver distance of 1m. As expected, BER is reduced when illuminance increases. However, it does not mention the effective data rate for each PHY I operating mode.

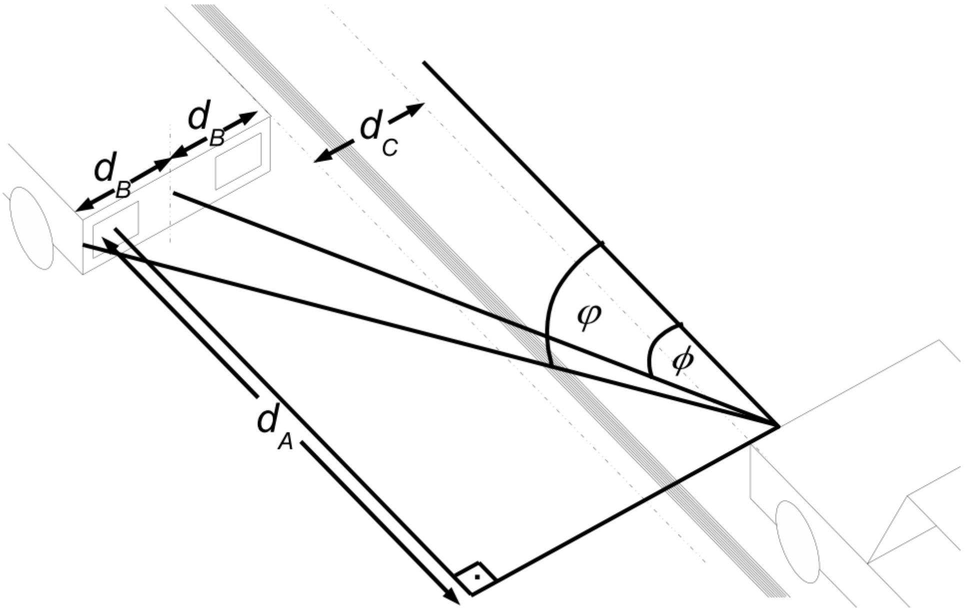
3.1.3 Turan et al. (2017)

Following a system design approach similar to [27] and [29], the feasibility of PHY I for LoS vehicle-to-vehicle VLC is evaluated using a prototype [30]. A Commercial Off-The-Shelf (COTS) automotive LED fog light is used as transmitter and Thorlabs $^{\text{TM}}$ PDA36A is the receptor positioned in the car bumper. In the first experiment, two vehicles are alighted to communicate with each other via LoS, and the distance between them is varied to analyze BER. Operating modes with data rate of 11.67 and 24.44 kbps allowed robust communication around 12m. All PHY I operating modes does not presented any error below 6m. In the second experiment, VLC is evaluated for vehicles traveling towards each other (Fig. 12). Critical angles to reach one (ϕ) or two lights (φ) of the opposite vehicle determines the opening angle of the light source. Considering vehicle width of 2m ($d_B = 1$m) and lane separation (d_C) of 40cm, the maximum value for d_A is 6m using Commercial Off-The-Shelf (COTS) LED fog lights.

3.1.4 Namonta et al. (2017)

A real time vital sign transmission system using the operating modes of IEEE 802.15.7 PHY I is proposed [31]. It enhances previous developed VLC system prototype [32] implemented in a arduino Mega2560, which connects to a computer via RS-232 serial communication. Its front-end circuit uses the relay driver ULN2803, which lights up the LED, and SH203P is the driver circuit for the phodectector. Coding algorithms are implemented using a SDR approach, and the arduino Mega2560 operates as ADC/DAC and interconnects the front-end circuit. As an experimental analysis of the proposed system, the maximum distance between transmitter and receiver to fulfill data rate requirements of PHY I operating modes with OOK modulation is calculated. The maximum distance obtained was around 130cm. In order to increase this number, a VLC repeater is used, and it increased the maximum transmitter-receiver distance to 590cm for the target application.

Figure 12 – Critical angles ϕ and φ for vehicle-to-vehicle VLC in opposite directions based on the vehicle and lane separation width ($2d_B$ and d_C). These angles determine the longitudinal distance between the vehicles for successful communication using visible light ([30], p. 4).



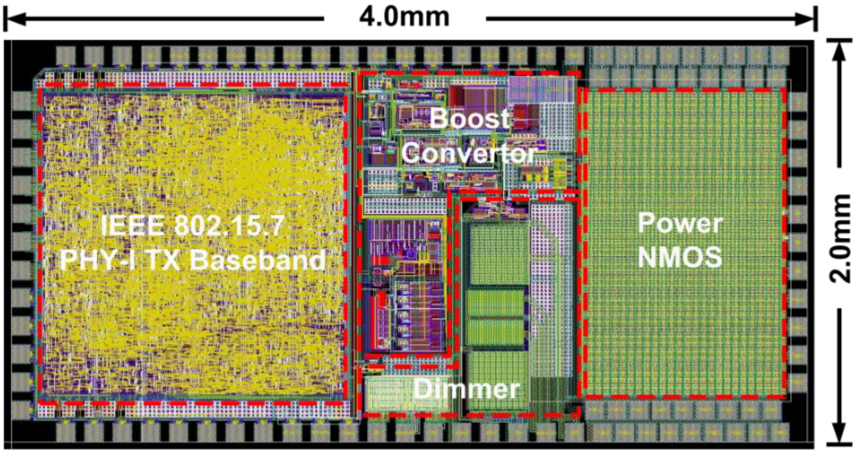
3.1.5 Hosein et al. (2019)

A low-cost fixed-point DSP chipset (TMS320F2812) is used to perform IEEE 802.15.7 PHY I operating modes with data rates of 24.44 kb/s and 73.3 kb/s [33]. The main limitation of this VLC system is that it is not able to decode data in real-time without adding concatenated coding to compensate the timing overhead introduced by the Viterbi decoder. Because of that, the operating mode that requires Viterbi decoder has a data rate approximately 6 times below to what is required by the standard (24.44 Kb/s). This result corroborates that a custom hardware design is the appropriate approach for implementing IEEE 802.15.7 PHY layer.

3.2 ASIC Design for IEEE 802.15.7 PHY layer

The Hong Kong University pioneers the development of ASIC design for IEEE 802.15.7 PHY, and the related publications [26,34] are results of F. Che's and B. Hussain's master theses, supervised by professors C. P. Yue and L. Wu. This collaborative effort at Hong Kong University ended up with an extensive end-to-end solution of a transceiver for IEEE 802.15.7 PHY I. F. Che's master thesis ("VLC Transceiver System Design using LED for IEEE 802.15.7" [25]) focuses on the specification of transceiver front-end circuit and covers the complete mixed signal design flow for the tape-out of the VLC transmitter, including its System-on-Chip (SoC) design, synthesis and Printed Circuit Board (PCB) project. It was fabricated using AMS 0.35 μm Complementary Metal Oxide Semiconductor (CMOS) process, and the SoC area is 2mm×4mm (Fig.13). The receiver, though, is implemented in a FPGA (Xilinx$^{\text{TM}}$ Virtex5 FPGA development board).

Figure 13 – Layout of transmitter SoC ([25], p. 47).



B. Hussain's master thesis ("Design and Implementaion of Visible Light Communication Systems" [24]) is complementary to F. Che's work. However, it emphasizes optical wireless link budget analysis, characterization of LEDs and photodetectors, and has more details about the digital design architecture of the VLC system, which is the major point to be recapitulated in this master thesis. At the receiver (Fig. 14), FEC, RLL and modulation are performed sequentially, and every function block in the data path has a buffer to accommodate input data. The "MCS Controller" commands the processing blocks and uses clock dividers to obtain the optical clock from the system clock (4 MHz). It is important to highlight that only the 'digital modulator' should require the optical clock to drive the VLC signal using the proper frequency, and other blocks may use the system clock. For such case, there will be a clock domain crossing in the 'digital modulator', and it should be handled by existing synchronization techniques [35].

The receiver has a system architecture (Fig. 15) very similar to the transmitter, but data flow goes in the opposite direction throughout the counterpart functions of the encoding procedure. It has a CDR circuit to detect the FLP and does not require any clock frequency transformation. MCS Controller has a 'PHR Decoder', which extracts information from the header to configure the data path to decode the PSDU. Beyond the parsing of the PHR, MCS Controller also coordinates high level operation of each data path block. According to the flowchart in [24] that presents the state transition diagram of MCS Controller, one processing step at a time is executed in the data path. However,

Figure 14 – System block diagram of the transmitter ([25], p. 21).

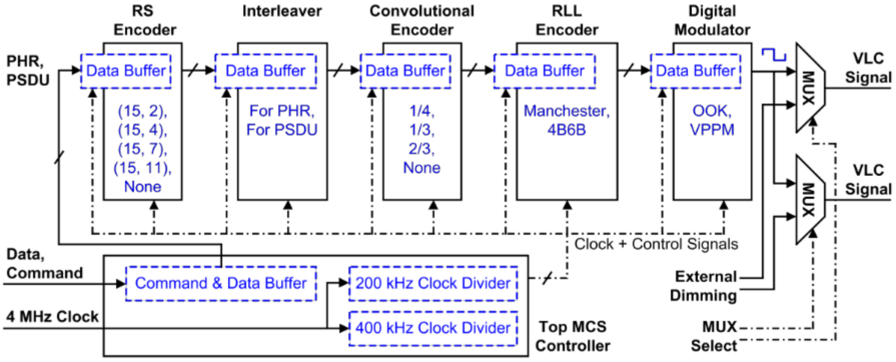
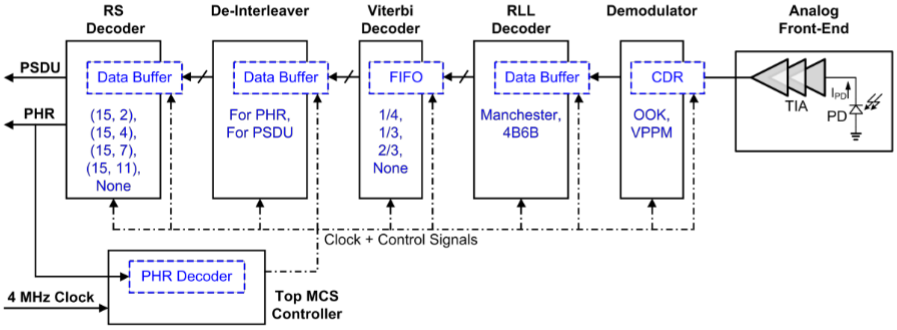


Figure 15 – System block diagram of the receiver ([25], p. 48).



some of these processing blocks might operate concomitantly, which would reduce the system latency. Another relevant architectural aspect is the use of an input buffer in every processing block, which impacts overall chip area, latency, and consumption. Some of them might not be needed depending on architectural choices, and this is investigated in the present work.

Since the target of this master thesis is narrowed down to the RTL implementation of a FEC compliant with IEEE 802.15.7, it is possible to explore lower level details of the system design that have not been covered by [25] or [24] due to their broader subject scope. For instance, interface definitions, architectural choices in hardware design, RTL system verification, and critical path and throughput analysis are aspects depicted by this study and present relevant outcomes to the academic community. Also, these works only explore FEC arrangements for PHY I, whereas this master thesis also covers FEC configurations for PHY II and III. Moreover, it is intended to provide open access to the IPs produced by this work as the design of a compliant IEEE 802.15.7 PHY layer is an extensive project that demands academic cooperation.

4 Forward Error Correction for IEEE 802.15.7

According to the operating modes enumerated in Fig. 56, 57 and 58 multiple arrangements of FEC are proposed for PHY I, II, and III IEEE 802.15.7 (Tables 56, 57 and 58). Each FEC arrangement results in a specific error correction capacity and data rate, and these two performance requirement measures are inversely proportional to each other. This chapter covers the architectural aspects of the proposed IP to implement a RTL model of a FEC compliant with all the thirty operating modes specified by IEEE 802.15.7. This standard only specifies the FEC functional blocks that a given operating mode must contain and does not define any implementation aspect at FEC block level.

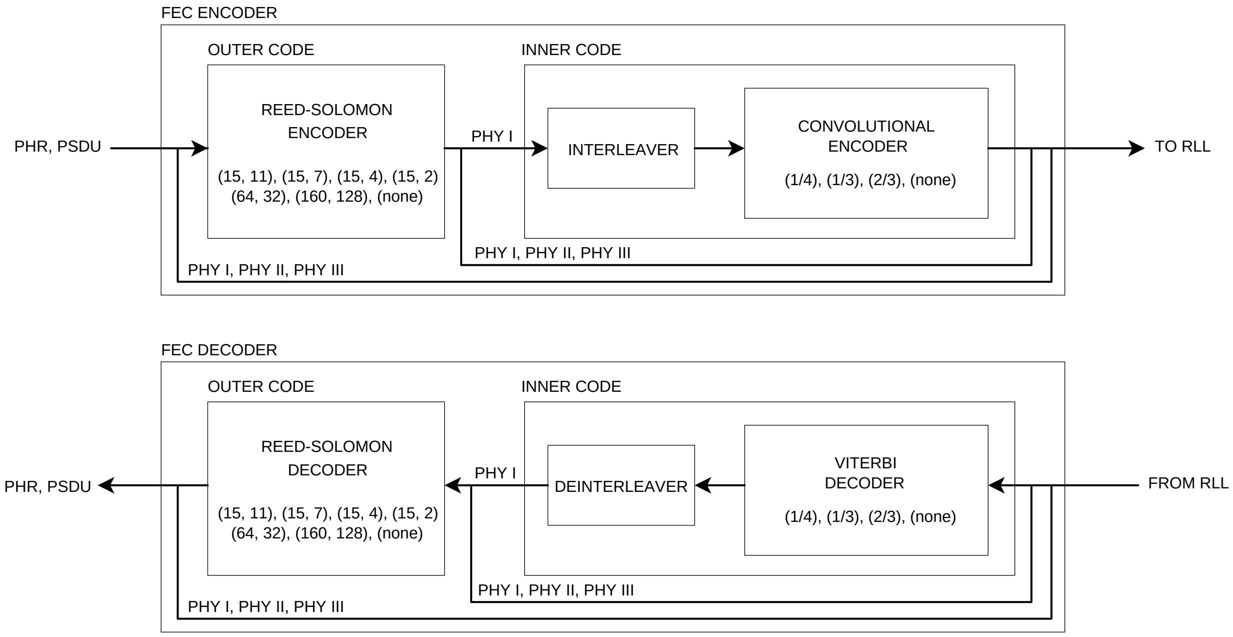
Fig. 16 summarizes all coexisting FEC configurations with the required base coding blocks: RS codec, Interleaver and CC. IEEE 802.15.7 states that PHR (section 2.3.2.2) and PSDU (section 2.3.2.5) must go through FEC encoder, and the subsequent step in the PHY data path is the RLL encoder (Fig. 7). For FEC decoder, the data flow takes the opposite direction and contains the counterparts of the encoder procedure. With the exception of the operating modes that do not require any coding handling in FEC, RS codec is prescribed for all other cases in PHY I, II and III, and there are six possible parametric configurations for it - explained by section 4.4.4. As PHY I has been designed for outdoor use with shorter frames, some of its operating modes also have the Interleaver and CC in their data path to enhance error correction capacity. CC has three different parametric configurations which are covered by sections 4.4.6 and 4.4.7.

The first aspect covered in this chapter is the FEC IP interface and its basic use. Then, the behavioral characteristics of RS, Interleaver, and CC are described to understand the role of each of them in a FEC. The interface of these blocks is also in this chapter to help understanding some architectural implications for upper levels of the FEC IP; however, detailed implementation analysis of them are covered by their related reference papers or user manuals. Finally, after exploring the basic FEC building blocks, it is possible to step into the architectural details of the FEC IP core using a top-down approach.

4.1 Generic data protocol interface for IP development

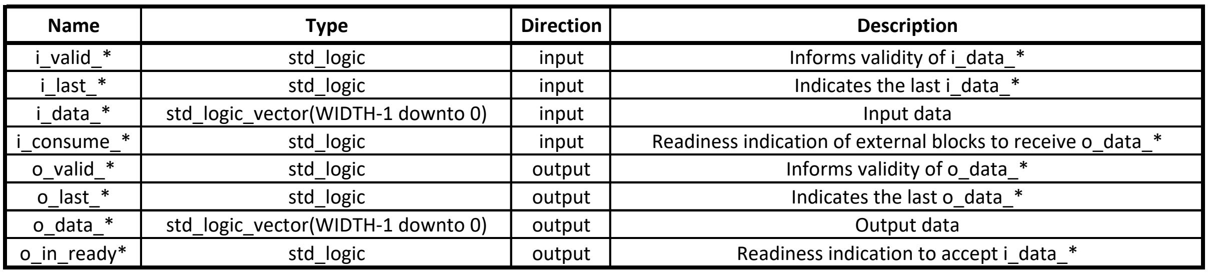
Before getting into the implementation aspects of the FEC IP itself, it is worth to introduce the data protocol interface used throughout the development of this project. This interface follows some conventions seen in commercial IPs [36] and is functionally similar AXI4-Stream protocol [37]. Fig. 17 describes the signal names that are employed in the interface of the RTL blocks. Input and output ports have "*i_*" and "*o_*" prefixes. The ‘*’ character is a place holder for custom meaningful suffixes that are chosen according to

Figure 16 – High-level diagram of FEC encoder and decoder for IEEE 802.15.7



the block functionality.

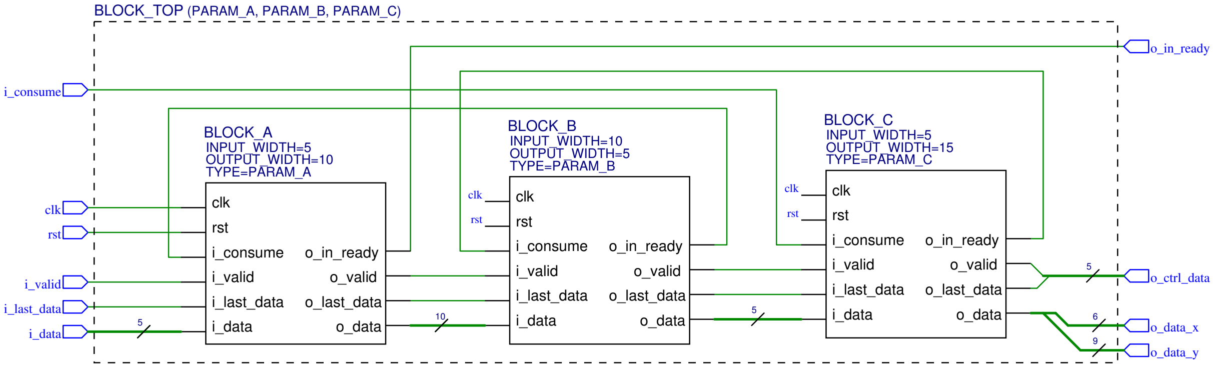
Figure 17 – Generic data protocol interface used in this master thesis.



Name	Type	Direction	Description
i_valid_*	std_logic	input	Informs validity of i_data_*
i_last_*	std_logic	input	Indicates the last i_data_*
i_data_*	std_logic_vector(WIDTH-1 downto 0)	input	Input data
i_consume_*	std_logic	input	Readiness indication of external blocks to receive o_data_*
o_valid_*	std_logic	output	Informs validity of o_data_*
o_last_*	std_logic	output	Indicates the last o_data_*
o_data_*	std_logic_vector(WIDTH-1 downto 0)	output	Output data
o_in_ready*	std_logic	output	Readiness indication to accept i_data_*

Fig. 18 illustrates how blocks are interconnected using the generic data protocol interface. There are three blocks in a serial connection. Data signals have a straight-forward connection, and output data ports (*o_data*) always feed input data ports (*i_data*) of the subsequent block in the path. The connection *o_in_ready*-*i_consume* between two blocks indicates whether the subsequent block is ready or not to receive incoming data from the predecessor one. This diagram has some symbolic representations that will appear in the next figures in this master thesis: generic parameters are placed bellow the block name with their instantiation values; all buses have their size annotated and are represented with a thicker width; and signal aliases are included to avoid many wire crossings.

Figure 18 – Block interconnect example using generic data protocol.

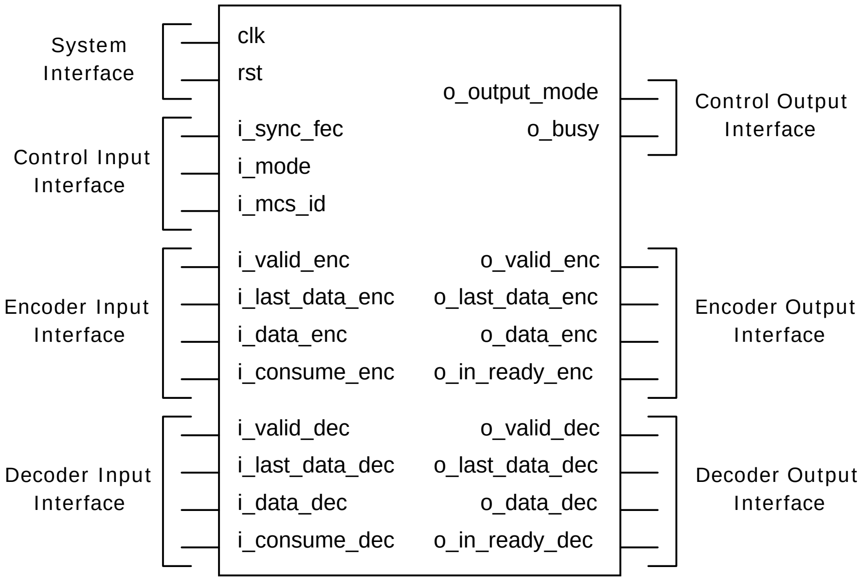


4.2 FEC IP Core Interface

The port interface used in the FEC IP for IEEE 802.15.7 (*VLC_PHY_FEC*) is depicted by Fig.19, and Fig. 20 describes the purpose of each port and displays its type and direction. The input frame must be serialized to the block since its functional sub-blocks are based on fixed-size slices (RS and Interleaver) or even single bits (CC). Also, there are specific ports for encoding and decoding input data (*i_data_enc* and *i_data_dec*), as input data is driven by a different unit in each case. Whenever an external block requires to use a FEC scheme for an input frame, the IP must be configured with the MCS ID (*i_mcs_id*), which is 6-bit word specified by IEEE 802.15.7 that maps all operating modes (Fig. 21). There is an internal ID representation in bits for selecting RS and CC modes, which directly correlates with the desired MCS ID. The FEC mode (encoder or decoder) must also be provided as well using *i_mode*. Such control information is synchronized by *i_sync_fec*, which configures the FEC to receive a new frame.

The effective width of input or output data signals varies according to the selected MCS ID. It turns out that the different configurations of RS and CC blocks used in FEC expects different input and output data width as also described by Fig. 21. External blocks that interacts with FEC IP are responsible to fill and extract the correct slices of input and output data ports, placed in their lowest significant bits. For FEC input data, the predecessor block - a Parallel-to-Serial (PS)/Serial-to-Parallel (SP) converter - should be able to serialize the input frame in 8, 4, or 1 bits depending on the MCS ID. And for FEC output data, the expected data input width in the successor block (RLL) matches FEC

Figure 19 – Port interface of VLC_PHY_FEC ports.



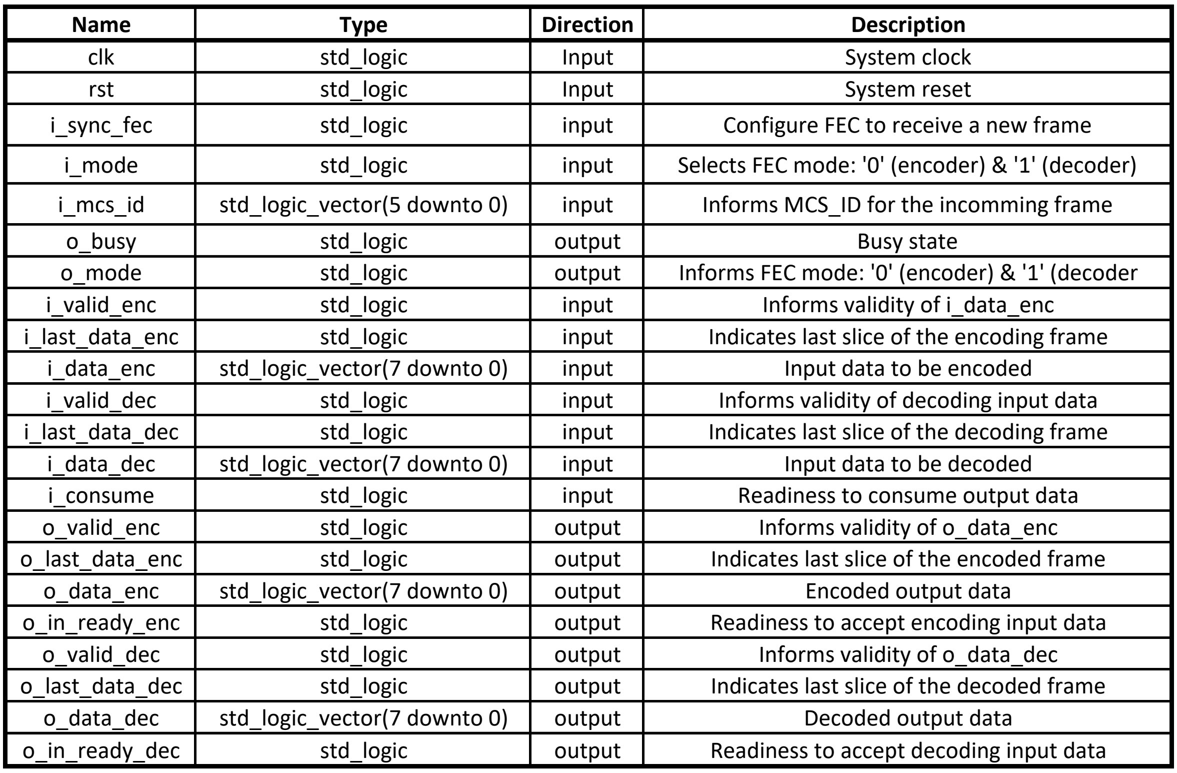
output data width for a given MCS ID, then throughput is not affected since not data conversion should be needed.

A waveform with a typical scenario for *VLC_PHY_FEC* is presented by Fig. 22. The procedure for reset is active high and takes 1 cycle. Whenever the FEC is not busy (*o_busy* = '0'), the FEC may be synchronized (*i_sync_fec* = '1') to encode (*i_mode* = '0') or decode (*i_mode* = '0') a frame for the desired operating mode (*i_mcs_id*). In the cycle after the synchronization, *o_busy* is assigned to '1', and the FEC is able to receive the data frame. There are specific data input and output ports for encoding and decoding flows.

Input data (*i_data_enc* or *i_data_dec*) is taken into account by the FEC only if the validity data indicator (*i_valid_enc* or *i_valid_dec*) is asserted. The last portion of the input frame is informed by *i_last_data_enc* or *i_last_data_dec*. Analogously, output data ports (*o_data_enc* or *o_data_dec*) are also validated by data indicators (*o_valid_enc* or *o_valid_dec*), and the last portion of the output frame is referred by *o_last_data_enc* or *o_last_data_dec*.

Blocks which drive input data frames to the FEC must also read *o_in_ready_enc* or *o_in_ready _dec*, which inform whether the FEC is ready or not to receive a new data input. The current data input should be hold until the FEC is ready to receive it. The port *i_consume* is used by the blocks which receive data output of the encoded or decoded frame to inform the FEC whether it is ready or not to consume the data. The FEC holds the value of output data ports whenever the respective "consumption" port is de-asserted. Only after the transmission of the last portion of the decoded or encoded frame, *o_busy*

Figure 20 – Description of *VLC_PHY_FEC* ports



Name	Type	Direction	Description
clk	std_logic	Input	System clock
rst	std_logic	Input	System reset
i_sync_fec	std_logic	input	Configure FEC to receive a new frame
i_mode	std_logic	input	Selects FEC mode: '0' (encoder) & '1' (decoder)
i_mcs_id	std_logic_vector(5 downto 0)	input	Informs MCS_ID for the incomming frame
o_busy	std_logic	output	Busy state
o_mode	std_logic	output	Informs FEC mode: '0' (encoder) & '1' (decoder
i_valid_enc	std_logic	input	Informs validity of i_data_enc
i_last_data_enc	std_logic	input	Indicates last slice of the encoding frame
i_data_enc	std_logic_vector(7 downto 0)	input	Input data to be encoded
i_valid_dec	std_logic	input	Informs validity of decoding input data
i_last_data_dec	std_logic	input	Indicates last slice of the decoding frame
i_data_dec	std_logic_vector(7 downto 0)	input	Input data to be decoded
i_consume	std_logic	input	Readiness to consume output data
o_valid_enc	std_logic	output	Informs validity of o_data_enc
o_last_data_enc	std_logic	output	Indicates last slice of the encoded frame
o_data_enc	std_logic_vector(7 downto 0)	output	Encoded output data
o_in_ready_enc	std_logic	output	Readiness to accept encoding input data
o_valid_dec	std_logic	output	Informs validity of o_data_dec
o_last_data_dec	std_logic	output	Indicates last slice of the decoded frame
o_data_dec	std_logic_vector(7 downto 0)	output	Decoded output data
o_in_ready_dec	std_logic	output	Readiness to accept decoding input data

is deasserted and the FEC may process a new incoming frame. The lack of multi-frame support affects latency, and this aspect is covered in chapter 5.

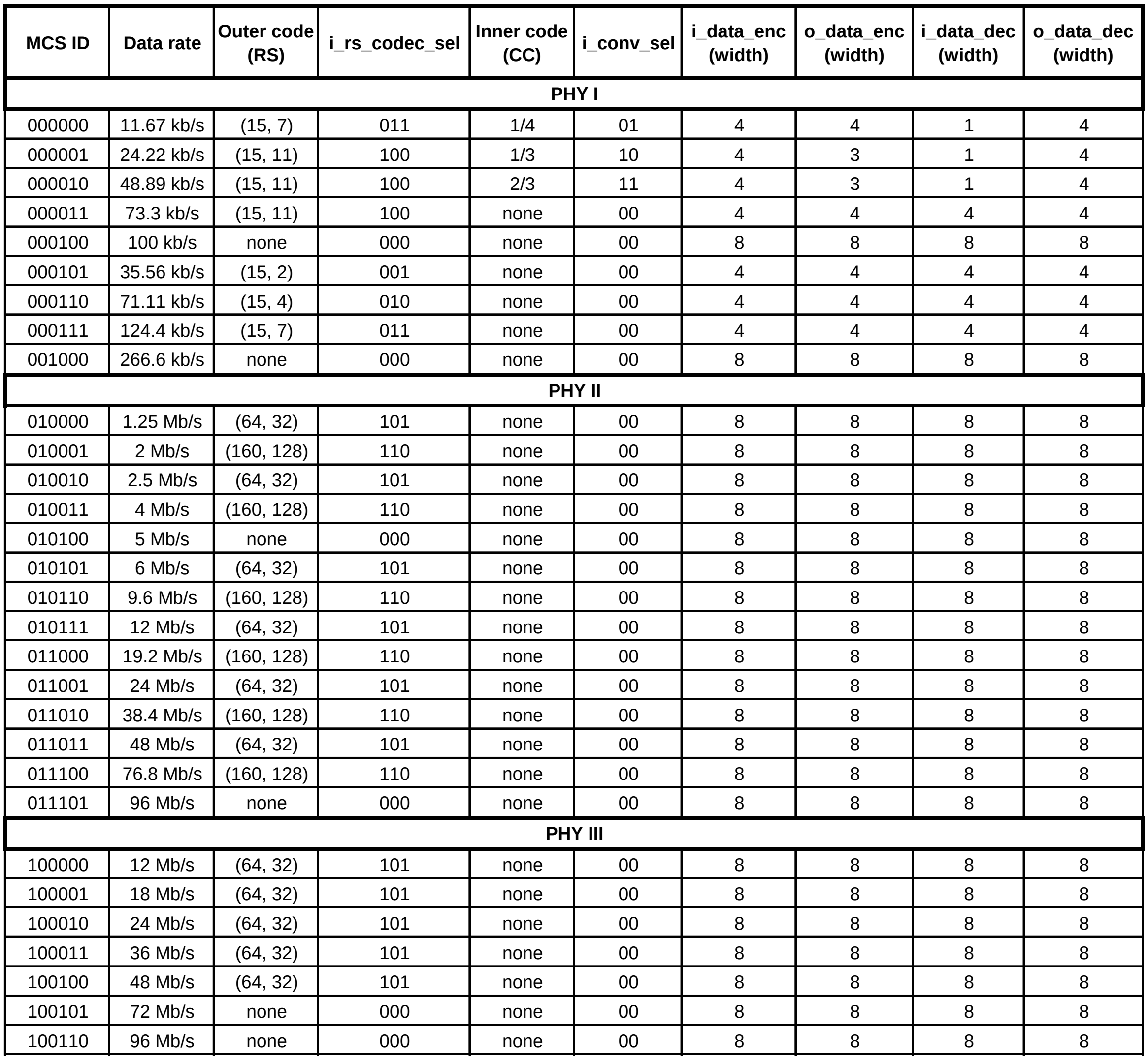
4.3 Basic building blocks in FEC

4.3.1 RS Codes

RS codes are one of the most popular ECC methods that use a block-by-block basis to correct burst errors and erasures in data. The adequate compromise between effectiveness and implementation complexity guarantees the widespread use of RS codes in many applications until today. The paper entitled "Polynomial Codes over Certain Finite Fields" [38] published in June 1960 was the first formal publication of RS codes. Many technologies and standards have adopted RS codes, and the Compact Disk (CD) system was the first consumer mass application to employ it [39], and since then engineering solutions such as satellite and mobile systems, Digital Versatile Disk (DVD) and barcodes, and digital television have used RS codes.

Theoretically, they are simply sets of algebraic curves defined by polynomials with a limited range of degrees. These curves when graphed are a set of discrete points - the abscissas and ordinates are values in a Galois Field (GF). The degree limitation allows

Figure 21 – Architectural mapping of MCS ID in FEC IP

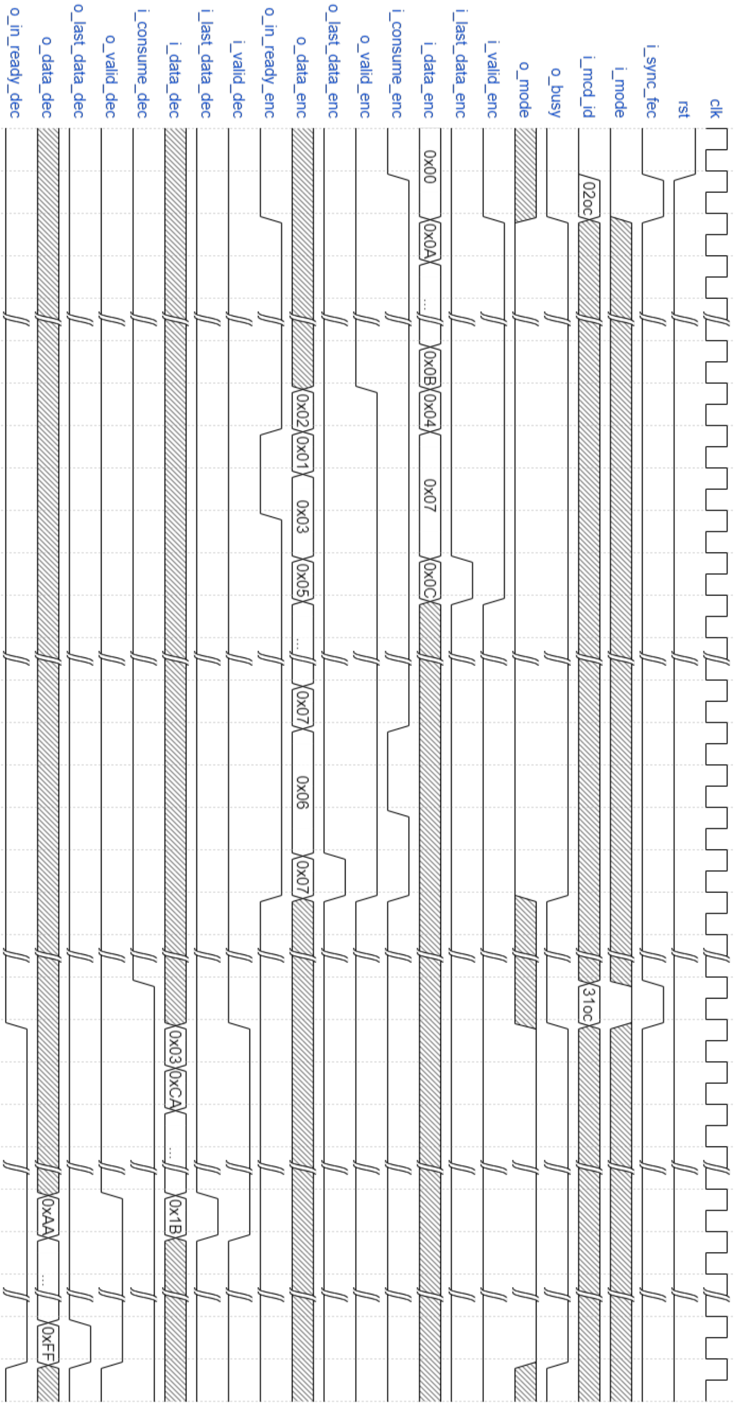


MCS ID	Data rate	Outer code (RS)	i_rs_codec_sel	Inner code (CC)	i_conv_sel	i_data_enc (width)	o_data_enc (width)	i_data_dec (width)	o_data_dec (width)
PHY I									
000000	11.67 kb/s	(15, 7)	011	1/4	01	4	4	1	4
000001	24.22 kb/s	(15, 11)	100	1/3	10	4	3	1	4
000010	48.89 kb/s	(15, 11)	100	2/3	11	4	3	1	4
000011	73.3 kb/s	(15, 11)	100	none	00	4	4	4	4
000100	100 kb/s	none	000	none	00	8	8	8	8
000101	35.56 kb/s	(15, 2)	001	none	00	4	4	4	4
000110	71.11 kb/s	(15, 4)	010	none	00	4	4	4	4
000111	124.4 kb/s	(15, 7)	011	none	00	4	4	4	4
001000	266.6 kb/s	none	000	none	00	8	8	8	8
PHY II									
010000	1.25 Mb/s	(64, 32)	101	none	00	8	8	8	8
010001	2 Mb/s	(160, 128)	110	none	00	8	8	8	8
010010	2.5 Mb/s	(64, 32)	101	none	00	8	8	8	8
010011	4 Mb/s	(160, 128)	110	none	00	8	8	8	8
010100	5 Mb/s	none	000	none	00	8	8	8	8
010101	6 Mb/s	(64, 32)	101	none	00	8	8	8	8
010110	9.6 Mb/s	(160, 128)	110	none	00	8	8	8	8
010111	12 Mb/s	(64, 32)	101	none	00	8	8	8	8
011000	19.2 Mb/s	(160, 128)	110	none	00	8	8	8	8
011001	24 Mb/s	(64, 32)	101	none	00	8	8	8	8
011010	38.4 Mb/s	(160, 128)	110	none	00	8	8	8	8
011011	48 Mb/s	(64, 32)	101	none	00	8	8	8	8
011100	76.8 Mb/s	(160, 128)	110	none	00	8	8	8	8
011101	96 Mb/s	none	000	none	00	8	8	8	8
PHY III									
100000	12 Mb/s	(64, 32)	101	none	00	8	8	8	8
100001	18 Mb/s	(64, 32)	101	none	00	8	8	8	8
100010	24 Mb/s	(64, 32)	101	none	00	8	8	8	8
100011	36 Mb/s	(64, 32)	101	none	00	8	8	8	8
100100	48 Mb/s	(64, 32)	101	none	00	8	8	8	8
100101	72 Mb/s	none	000	none	00	8	8	8	8
100110	96 Mb/s	none	000	none	00	8	8	8	8

recovery of the complete curve even when the graph is assumed to be smudged and erased at many points [40]. It turns out that these discrete points oversample the curve delimited by the polynomial resulting from the input message, then it is possible to find out the original curve for cases where the number of points with errors after transmission is less than the maximum recovery capacity, which is determined by number of oversampling points.

RS codes are represented as a set of length n vectors (codewords) where their elements (symbols) consist of m binary digits imposed by the GF order (2^m) (Fig. 23). The original message takes k vector positions, then there are $n-k$ parity symbols. The maximum capacity of error correction (t) of a RS codec corresponds to $t=(n-k)/2$. A RS codec configuration is usually referred by the notation $RS(n,k)$. IEEE 802.15.7 defines six possible RS codec configurations: RS(15,2), RS(15,4), RS(15,7), and RS(15,11) over a

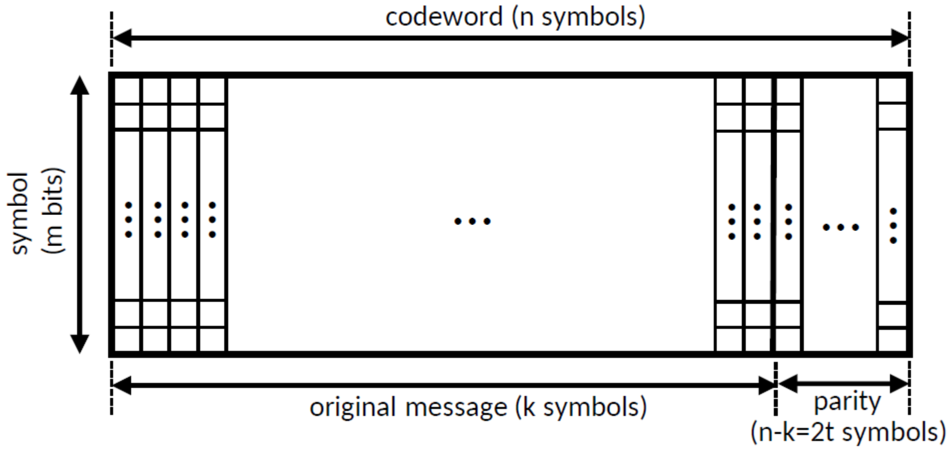
Figure 22 – Waveform example for FEC IP utilization



GF order of 2^4 in PHY I and RS(64,32) and RS(160,132) over a GF order of 2^8 in PHY II and PHY III.

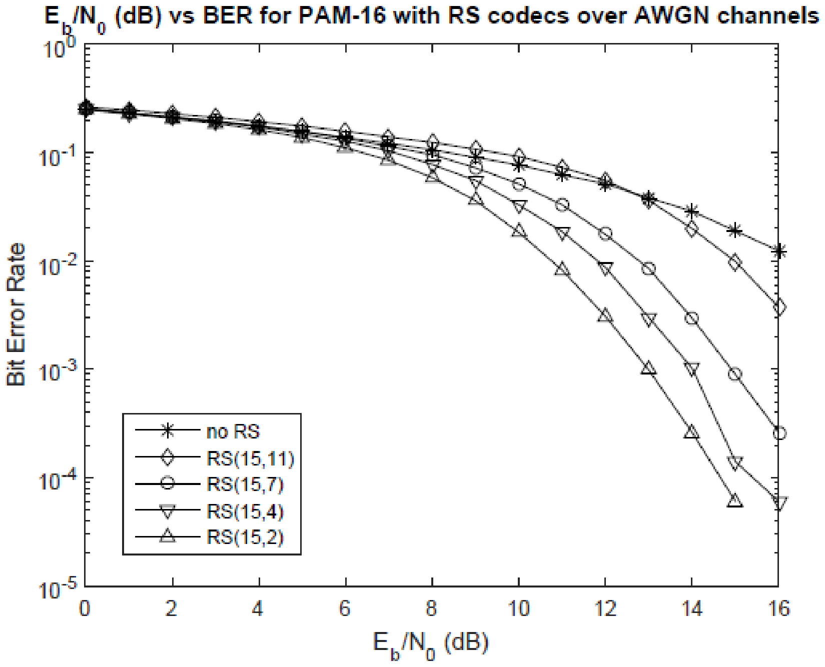
Performance in RS codecs can be measured comparing Energy per Bit to Noise Power Spectral Density Ratio (Eb/N0) with BER, as Eb/N0 allows comparing different coding schemes because it normalizes Signal to Noise Ratio (SNR) per bit. To illustrate

Figure 23 – Generic parametric configuration of RS codes ([41], p. 2)



how the number of parity symbols ($n - k$) affects BER performance, Fig. 3 shows different configurations of RS codes from IEEE 802.15.7 where n = 15. As expected, BER increases when k is closer to n.

Figure 24 – Transmission efficiency for Pulse-Amplitude Modulation with 16 discrete levels (PAM-16) over Additive White Gaussian Noise (AMGN) channels using RS codecs with n = 15 and k = 11, 7, 4, and 2 with Eb/N0 varying from 0 to 16 in a MATLAB® simulation of 10^6 sampling transmissions [42].



RS codecs use a Generator Polynomial (GP) as keycode to encode the input sequence, and it is going to result in a restricted polynomial of order n that may be discoverable by its decoder counterpart if the number of errors during the transmission does not exceed t. GP is characterized by a polynomial of order $2t$, where its roots are usually the first $2t$ elements of the defined GP.

The encoding process transmits k message symbols and $n - k$ parity symbols per encoding iteration. Parity symbols corresponds to the remaining of the division between the polynomial formed by the input sequence of k symbols and the GP. The decoding process is based on a key equation solver that finds the error locator and the evaluator polynomials, which informs which symbol is incorrect and which value should be added to it to cancel the introduced error.

The use of GF as arithmetic framework for codeword encoding and decoding processes guarantees that RS codes are linear - every codeword comes from the sum of all other codewords - and cyclic - a shift rotated codeword of any number of symbols represents another codeword [41]. It enables the application of Linear Feedback Shifter Registers (LFSR) for RS codec implementation, which has a convenient hardware realization [43].

4.3.2 IP Core for RS Codec

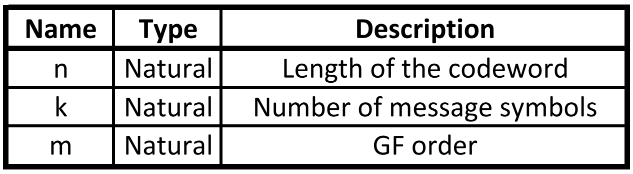
It has not been found any open IP solution that encompasses all required RS configurations specified by IEEE 802.15.7. Most of the available IPs are crafted for a specific configuration of RS codes. As result of this master thesis, an open IP core for a parameterizable RS Codec has been developed to be employed in this project. Its generic and port interfaces are covered by Fig. 25 and 26. Detailed explanation about its architecture, verification and synthesis can be found in the following publication [42]:

Mateus G. Silva, Guilherme L. Silvano, and Ricardo O. Duarte

RTL Development of a Parameterizable Reed-Solomon Codec

ISSN 1751-8601 doi: pending www.ietdl.org

Figure 25 – Parameter interface of the RS Codec IP [42].

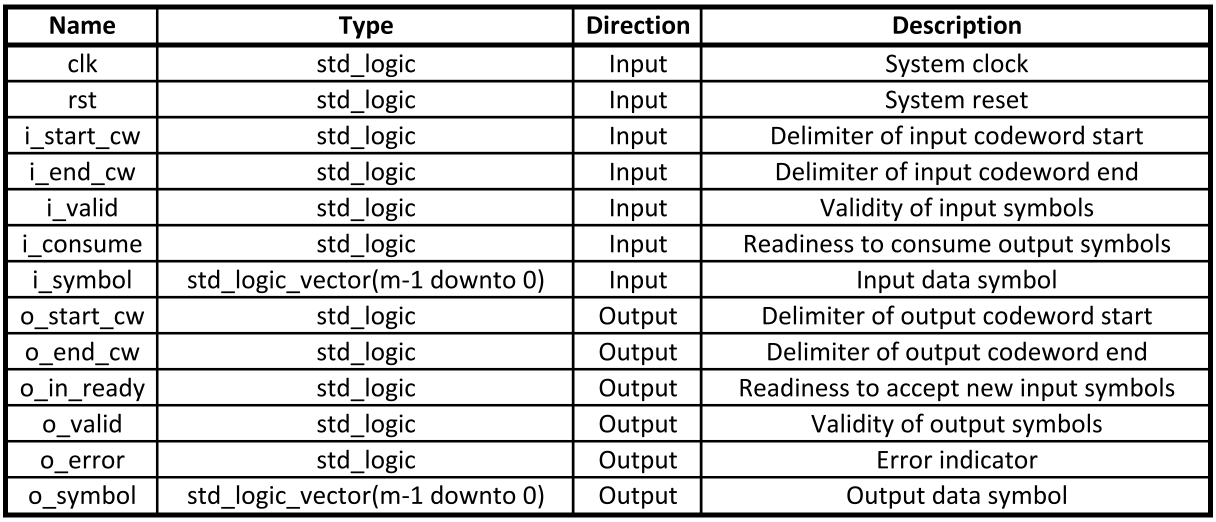


Name	Type	Description
n	Natural	Length of the codeword
k	Natural	Number of message symbols
m	Natural	GF order

4.3.3 Block Interleaver

Block interleavers are employed to mitigate burst errors occurred during data exchange in a communication channel. These errors are harmful because they may easily

Figure 26 – Port interface of the RS Codec IP [42].



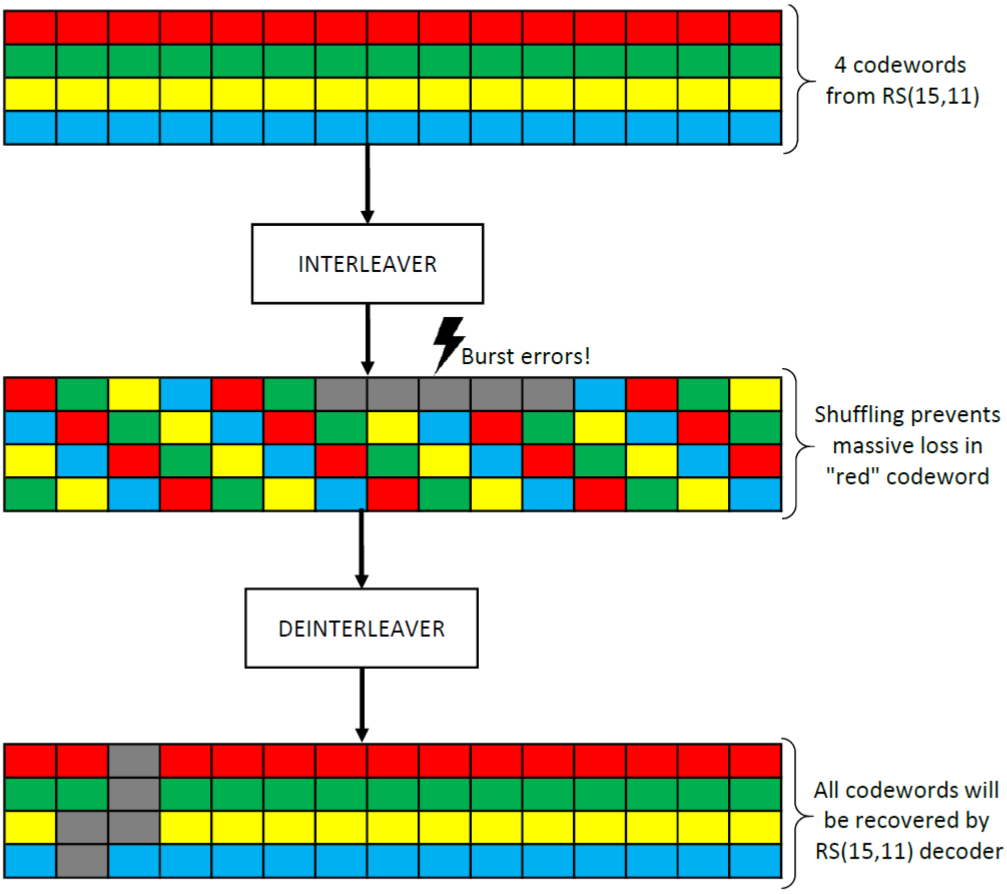
Name	Type	Direction	Description
clk	std_logic	Input	System clock
rst	std_logic	Input	System reset
i_start_cw	std_logic	Input	Delimiter of input codeword start
i_end_cw	std_logic	Input	Delimiter of input codeword end
i_valid	std_logic	Input	Validity of input symbols
i_consume	std_logic	Input	Readiness to consume output symbols
i_symbol	std_logic_vector(m-1 downto 0)	Input	Input data symbol
o_start_cw	std_logic	Output	Delimiter of output codeword start
o_end_cw	std_logic	Output	Delimiter of output codeword end
o_in_ready	std_logic	Output	Readiness to accept new input symbols
o_valid	std_logic	Output	Validity of output symbols
o_error	std_logic	Output	Error indicator
o_symbol	std_logic_vector(m-1 downto 0)	Output	Output data symbol

violate the maximum capacity of ECC methods employed in communication systems, such as the RS codes used in IEEE 802.15.7. The fundamental principle of a interleaver is to reorder a set of data through a certain permutation pattern [44]. By doing so, eventual burst errors are spread throughout the data stream, which leverages the effectiveness of ECCs in correcting them [45].

Fig. 27 is an illustration of the mitigation for burst errors offered by the interleaver. Consider a frame of 22 bytes where a RS(15, 11) over a GF(2^4) is used as ECC - each square is a symbol of 4 bits and there are 4 parity symbols. Then, there will be four resulting codewords from the RS encoding process which are represented by different colors. In this example, the block interleaver shuffles the data stream using a matrix structure, in which input data is written row-wise and output data is read column-wise. Due to the use of an interleaver, a burst error of five symbols did not affect the recovery of any codeword because after the deinterleaving process the number of errors is within the maximum capacity of error capacity of RS(15, 11), which is two. Nonetheless, the "red" codeword would be lost if the interleaving procedure has not been not used.

The provided example mimics what is described by Equation (4.1), which defines the permutation rule of the block interleaver specified by IEEE 802.15.7. The number of rows (n) has a fixed value (15), whereas the interleaver depth (D) represents a variable number of columns filled after the writing process, limited by the maximum number of elements defined by the internal memory size. The dimension of the block (S_{block}) is given by $n * D$, and p is the number of missing elements to complete the last column of the block. The interleaved indexes are provided by $l(i)$, while the location of the indexes to be

Figure 27 – Interleaver against burst errors. Each color represents a codeword output from RS Encoder.



punctured are obtained by $z(t)$.

$$\begin{aligned} l(i) &= (i \, mod \, D) * n + \left\lfloor \frac{i}{D} \right\rfloor; \, i = 0, 1, ..., (S_{block} - 1) \\ z(t) &= (n - p + 1) * D + t * D - 1; \, t = 0, 1, ..., (p - 1) \end{aligned} \tag{4.1}$$

4.3.4 IP Core for Block Interleaver

Similar to RS codec, an open IP solution of a block interleaver compliant with IEEE 802.15.7 has not been found. Hence, an open IP for a parameterizable block interleaver has been developed to be used in this project as well. Its IP interface is covered by Fig.

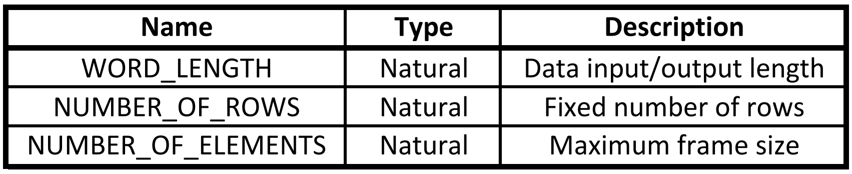
28 and 29. Detailed explanation about its architecture, verification and synthesis can be found in the following publication [46]:

Mateus G. Silva, Gabriel R. T. Araújo, Elisa S. Bacelar, Ricardo O. Duarte

Design and Verification of a Parameterizable Interleaver and Deinterleaver for Visible Light Communication

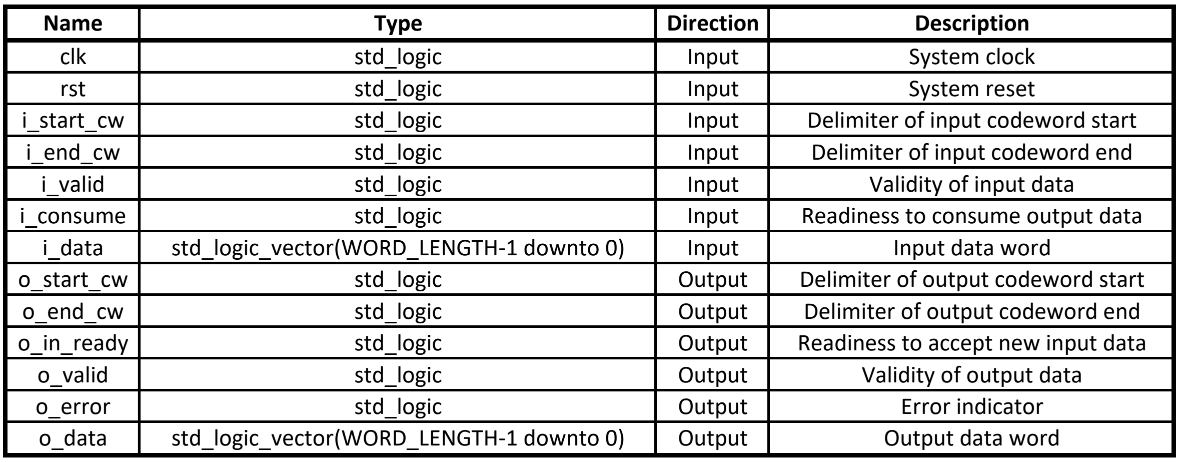
WCAS 2020, August 24–28, 2020, Campinas, SP, Brazil

Figure 28 – Parameter interface of the Interleaver [46].



Name	Type	Description
WORD_LENGTH	Natural	Data input/output length
NUMBER_OF_ROWS	Natural	Fixed number of rows
NUMBER_OF_ELEMENTS	Natural	Maximum frame size

Figure 29 – Port interface of the Interleaver [46].



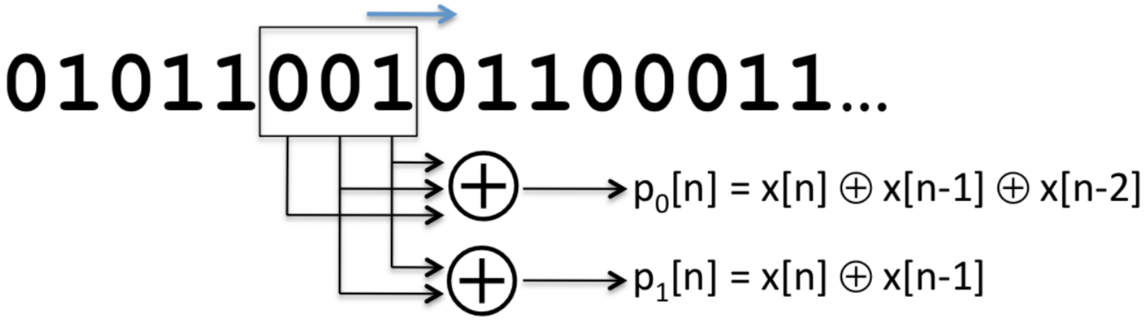
Name	Type	Direction	Description
clk	std_logic	Input	System clock
rst	std_logic	Input	System reset
i_start_cw	std_logic	Input	Delimiter of input codeword start
i_end_cw	std_logic	Input	Delimiter of input codeword end
i_valid	std_logic	Input	Validity of input data
i_consume	std_logic	Input	Readiness to consume output data
i_data	std_logic_vector(WORD_LENGTH-1 downto 0)	Input	Input data word
o_start_cw	std_logic	Output	Delimiter of output codeword start
o_end_cw	std_logic	Output	Delimiter of output codeword end
o_in_ready	std_logic	Output	Readiness to accept new input data
o_valid	std_logic	Output	Validity of output data
o_error	std_logic	Output	Error indicator
o_data	std_logic_vector(WORD_LENGTH-1 downto 0)	Output	Output data word

4.3.5 Convolutional Codes

Convolutional Codes (CC) are another widespread technique present in many applications, including IEEE 802.11 (WiFi) standards. IEEE 802.15.7 also prescribes CC for three operating modes in PHY I (Table 56), which are the most reliable options to transmit a VLC frame at the cost of lower data rates. Massachusetts Institute of Technology (MIT) has very comprehensive lectures on CC [47] for the course of "Digital Communication Systems", and a brief explanation of CC will be provided based on them.

The encoding process uses a sliding window to compute the parity symbols of an input bitstream as described by Fig. 30. This is equivalent to the mathematical operation of convolution - and that is the reason for name of the method. The size of the window is known as constraint length (L) and the number of output parity bits (r) defines the rate ($1/r$) of the CC. Each parity bit output is determined by a GP, which selects the bits of the window that are summed up - performed by reduce-XOR operation - to result in a parity bit. Unlike RS codes, the input bitstream is totally converted into parity bits and the message is not split into independent blocks. The larger L and r, the greater is the resilience against bit errors, albeit processing time in the decoder and bandwidth demand are higher.

Figure 30 – Illustration of CC with sliding window of $L = 3$ and $r = 2$ ([47], p. 66).


01011001011000 11...
$p_0[n] = x[n] \oplus x[n-1] \oplus x[n-2]$
$p_1[n] = x[n] \oplus x[n-1]$

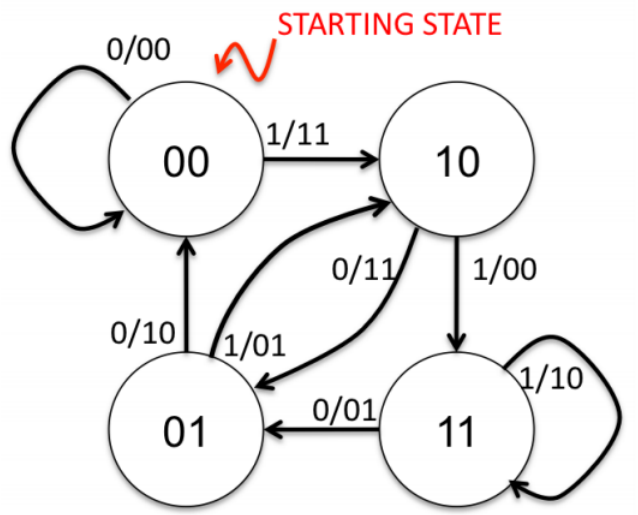
A Finite State Machine (FSM) representation for CC (Fig. 31) is an alternative view of Fig. 30 and also helps to understand its decoding process. The number of states S is given by 2^{L-1} since $L - 1$ corresponds to the number of bits that the window held from the previous iteration. Each state has 2 possible transitions determined by the incoming message bit of the current iteration n. Then, states and arcs are labeled as "$x[n-1], x[n-2]...x[n-L+1]$" and "$x[n]/p_0p_1$" respectively.

The decoding process aims to infer the most likely sequence sent (c) given a received sequence r. It basically maximizes the probability $P(r|c)$, and it is known as "maximum likelihood decoder". That corresponds to the smallest Hamming distance between r and all possible sequences of c. The brute force algorithm of checking Hamming distance for all c possibilities does not scale since the combinatorial number of sequences is given by 2^l, where l is the length of c. For example, IEEE 802.15.7 allows frames up to 8184 bits for PHY I. Therefore, there must be a more efficient algorithm to decode CCs.

Viterbi algorithm [48] provides a feasible method to decode CCs, and it is accomplished by finding the maximum likelihood path in a trellis structure (Fig. 32). Such representation displays the unrolling of all possible state transitions of a FSM for CC over time. The smallest Hamming distance between the initial and last states in the trellis structure provides the most likely c. This algorithm has time complexity of O(lS^2) and space complexity of O(lS).

The implementation of the Viterbi algorithm must calculate the Branch Metric

Figure 31 – State machine representation of CC for $L = 3$ and $r = 2$ ([47], p. 69).



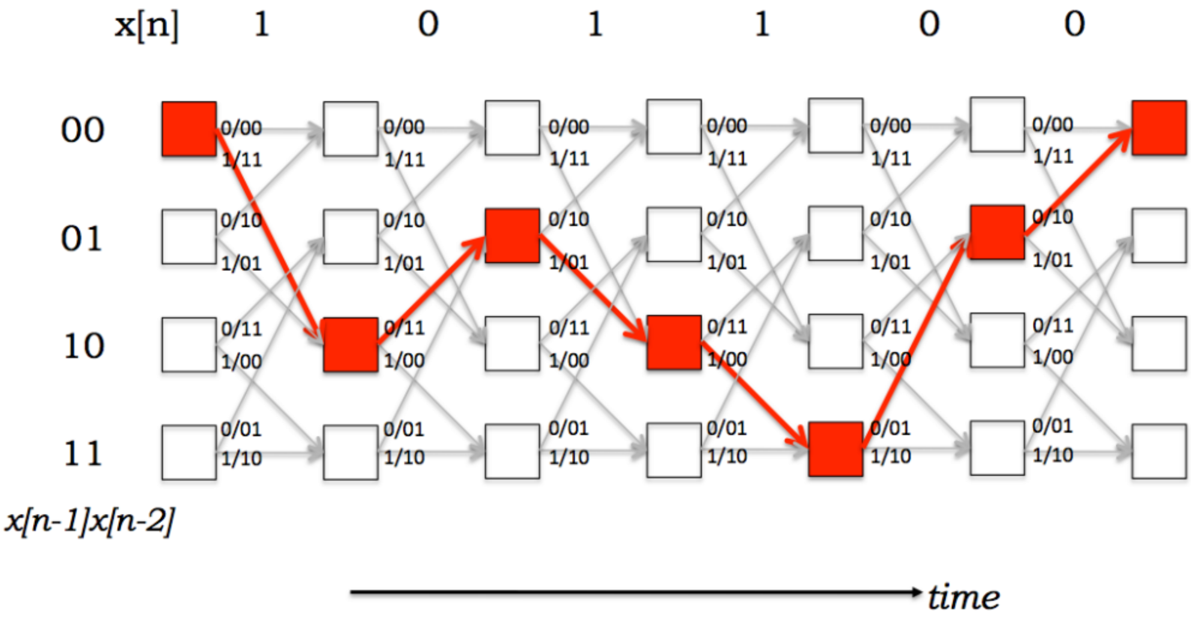
(BM) of the received parity bits (Fig. 33), which is the Hamming distance relative to all possible FSM transitions. In every iteration the Path Metric (PM) - the smallest cumulative Hamming distance of the path - is updated for each state using an *add-compare-select* procedure: *add* BM to the previous PM to calculate the two possible PMs for a state, *compare* these results, and *select* the smallest one to update PM. At the end of this process, the lowest PM value among the states indicates maximum likelihood path within the trellis structure, and it determines the decoded bit sequence.

The previous explanation of the decoding process for CCs relies on a hard decision algorithm since the input data values are already assigned to logical values '0' or '1'. However, it turns out that the analog to digital conversion of the input data stream might introduce inaccurate values. This conversion process relies on a comparison between the received analog signal and a threshold value, and, if they are very close to each other, it could be stated that the resulting digital value is undetermined even though the ADC assigns some value for it. Then, the soft decision mode of the Viterbi algorithm attempts to avoid such premature bit digitalization and uses quantized version of a bit based on its original analog value. The only difference between hard and soft decision algorithms is that BM and PM are positive real numbers since BM is calculated with the square of the difference between the received and the expected threshold values.

4.3.5.1 IP core for Convolutional Encoder

IEEE 802.15.7 specifies a rate-1/3 convolutional encoder with $L = 7$ and GPs corresponding to $GP_0 = 133_8$, $GP_1 = 171_8$, and $GP_2 = 165_8$. There is not any available open IP core for convolutional encoder with such specification. Then, a parameterizable

Figure 32 – The trellis illustration for the decoding task and the timing unrolling the FSM that represents the number of states hold by the convolutional code ($x[n-1]x[n-2]$). In red, it is represented the most likely path given the received parity bits, which assemblies the decoded message ($x[n]$). In each state, the top and the bottom arcs are translated into '0' and '1'. For each state, PM is updated by the BM, and the lowest PM among the states is selected as the decoded sequence at the last time iteration. ([47], p. 72)



convolutional encoder has been implemented from scratch, where rate, L, and GPs are configurable. This block has a straight-forward RTL architecture, which has already been explored by the literature [49, 50]. The parameter and port interface of the convolutional encoder IP is described by Fig. 34 and 35. The input data is a single bit, whereas the output data width matches the chosen rate value.

4.3.5.2 IP Core for Viterbi Decoder

It has been found a suitable open IP core for Viterbi Decoder under General Public License (GPL) license which is compliant with IEEE 802.15.7. This IP core has been maintained by Creonic GmbH, which is an ISO 9001:2015 certified provider of IP cores for communication applications. Its user guide [51] explains the block interface, provides waveform examples, and also presents a very detailed performance analysis with multiple communication standards which require CCs with different parametric variations. Table 36 refers to the parameter interface of the IP core. Beyond the usual parameters for rate (*NUMBER_PARITY_BITS*) and GPs (*PARITY_POLYNOMIALS*), there are other parameters which impact performance and area of the resulting synthesis of the block: *BW_LLT_INPUT* determines the input data length, which indicates the number of

Figure 33 – The BM for hard decision decoding. In this example, the receiver gets the parity bits 00, and each transition receives a score (in red) based on the hamming distance between parity bits represented by the transition and the received bits. The PM of the states in the next iteration $(i+1)$ is updated with the lowest possible value. ([47], p. 77)

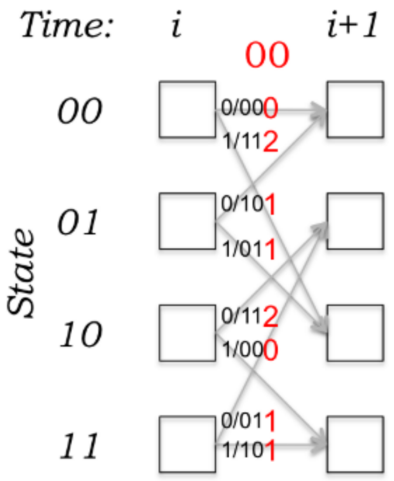
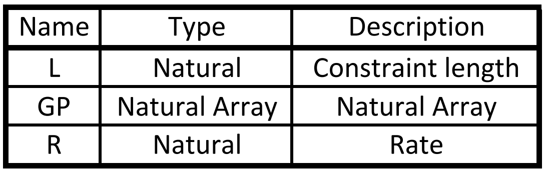


Figure 34 – Parameter interface of the Convolutional Encoder

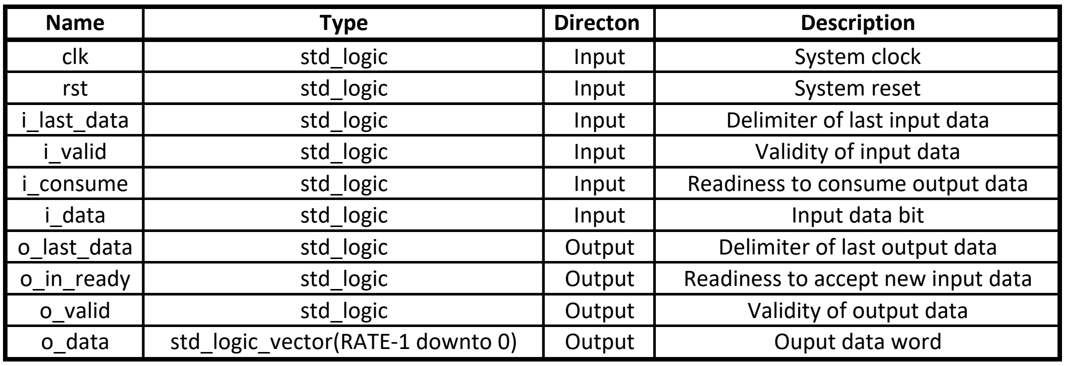


Name	Type	Description
L	Natural	Constraint length
GP	Natural Array	Natural Array
R	Natural	Rate

quantization levels for a single bit; *MAX_WINDOW_LENGTH* specficies the maximum value for window size and acquisition length, and it bounds the size of the RAM components used by the IP core; and *DISTRIBUIED_RAM* decides whether a distributed or a blcok Random-Access Memory (RAM) is used in the block. Also, the user can enable a recursive convoultional code approach [52] with *FEEDBACK_POLYNOMIAL*.

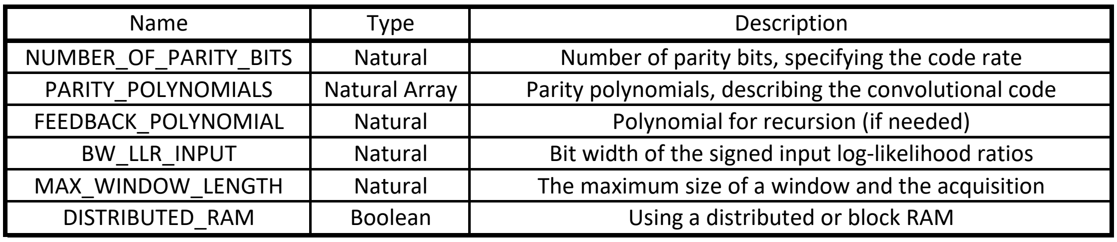
The port interface of the IP core for Viterbi decoder (Fig. 37) uses AXI4-Stream protocol [37], and most signals have direct translation to the generic data protocol interface used by this thesis, covered by section 4.1. Then, a block has been created by this work to wrap the original IP core (Fig.38). Only ports belonging to control interface are handled inside the wrapper, and the IP core is configured to receive the next frame after the transmission of the last decoded bit of the current frame. The port *s_axis_ctrl_tdata* has width of 32 bits, which is used to setup window length (first 16 bits) and acquisition length (last 16 bits) in run time during the configuration step.

Figure 35 – Port interface of the Convolutional Encoder



Name	Type	Directon	Description
clk	std_logic	Input	System clock
rst	std_logic	Input	System reset
i_last_data	std_logic	Input	Delimiter of last input data
i_valid	std_logic	Input	Validity of input data
i_consume	std_logic	Input	Readiness to consume output data
i_data	std_logic	Input	Input data bit
o_last_data	std_logic	Output	Delimiter of last output data
o_in_ready	std_logic	Output	Readiness to accept new input data
o_valid	std_logic	Output	Validity of output data
o_data	std_logic_vector(RATE-1 downto 0)	Output	Ouput data word

Figure 36 – Parameter interface of Viterbi Decoder [51].



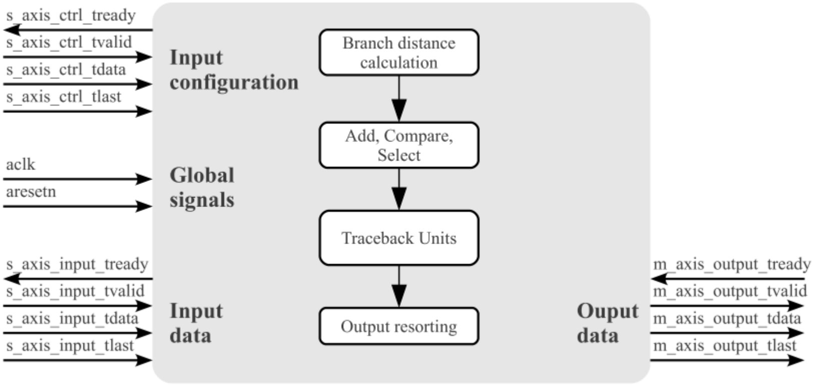
Name	Type	Description
NUMBER_OF_PARITY_BITS	Natural	Number of parity bits, specifying the code rate
PARITY_POLYNOMIALS	Natural Array	Parity polynomials, describing the convolutional code
FEEDBACK_POLYNOMIAL	Natural	Polynomial for recursion (if needed)
BW_LLR_INPUT	Natural	Bit width of the signed input log-likelihood ratios
MAX_WINDOW_LENGTH	Natural	The maximum size of a window and the acquisition
DISTRIBUTED_RAM	Boolean	Using a distributed or block RAM

The window length determines the size of data chunks of the frame processed in each iteration. It is also the upper bound the acquisition length, which determines how many inputs bits must be processed prior to generate a valid output. The IP core manual recommends $6L$ for them in order to avoid performance downgrade in BER. In IEEE 802.15.7, it corresponds to 42. Fig. 37 also highlights the main steps of Viterbi algorithm, and ‘Traceback Unit’ and ‘Output resorting’ were not mentioned in the previous algorithm explanation. ‘Traceback Unit’ is responsible for performing a backwards traversal in the trellis structure to find out the maximum likelihood path, whereas ‘Output resorting’ rearranges the resulting decoded frame from ‘Traceback Unit’, which is in reverse order.

4.4 Top-down architecture view of FEC

After explaining the functional aspects of the core blocks (RS, Interleaver, and CC) used in the FEC IP core, this section describes how these blocks are arranged to enable all the thirty operating modes specified by IEEE 802.15.7. The RTL architecture proposed for FEC is analyzed using a top-down approach, which covers the system level decisions taken in this project. The top level block (*VLC_PHY_FEC*) has a controller and the units responsible for FEC encoding and decoding functions (Fig. 39). The FEC is only able to receive input data after the configuration process, which asserts *o_busy*.

Figure 37 – Port interface and main sub-blocks of Viterbi decoder IP core ([51], p. 4)



This signal allows *i_valid* ports of encoder and decoder units to receive the logic value '1'. Also, encoding and decoding operations are mutually exclusive since a device will never be receiving and transmitting a frame at the same time - section 9.2.7 of IEEE 802.15.7 states that the transceiver might be at either transceiver disabled (*TRX_OFF*), transmitter enabled (*TX_ON*) or received enable (*RX_ON*) states.

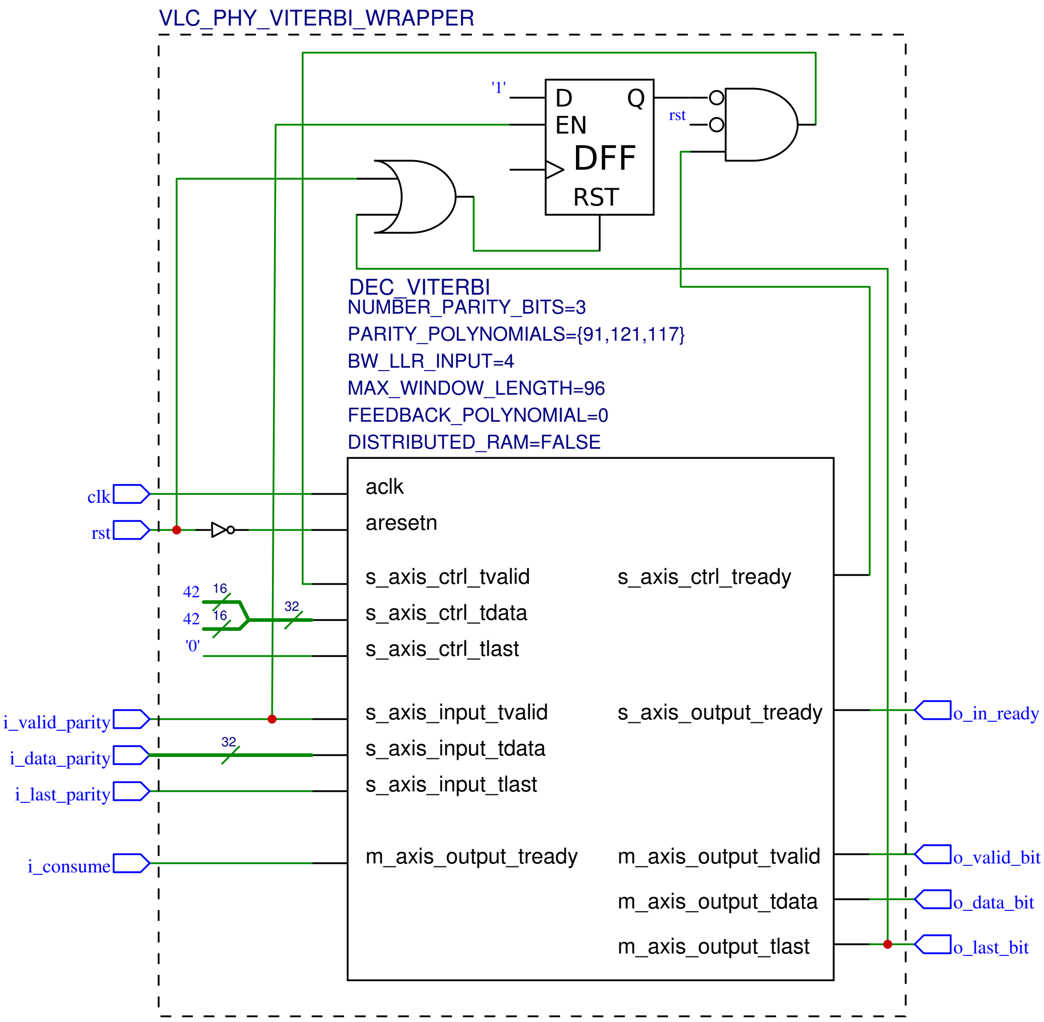
4.4.1 FEC Controller

The FEC Controller (*VLC_PHY_FEC_CONTROLLER*) has a simple interlock logic responsible for setting up the FEC with the provided MCS ID (*i_mcs_id*) and the functional mode (*i_mode*), which are stored into two configuration registers (Fig. 59). Once the FEC is busy, it is only released after the transmission of the last slice of the decoded or encoded frame. It means that the FEC is able to process just one frame at a time, and it is serialized according to the input/output data width specified by 21) for each operating mode. The block *VLC_PHY_MCS_ID_DEC* is a lookup table which converts the MCS ID into two signals (*o_rs_codec_sel* and *o_conv_sel*) responsible for selecting the proper RS and CC modes for encoding and decoding procedures.

4.4.2 FEC Encoder

The FEC Encoder (*VLC_PHY_FEC_ENCODER*) performs data transformation on the input frame according to selected operating mode (Fig. 60). It interprets *i_conv_sel* and *i_rs_codec*, driven by *VLC_PHY_FEC_CONTROLLER*, to select the data path with the expected FEC arrangement using multiplexers and AND gates. In summary, output data signals may be driven either by the convolutional encoder, RS encoder, or even input data signals when FEC is inactive for a given operating mode.

Figure 38 – Wrapper for Viterbi Decoder IP core.



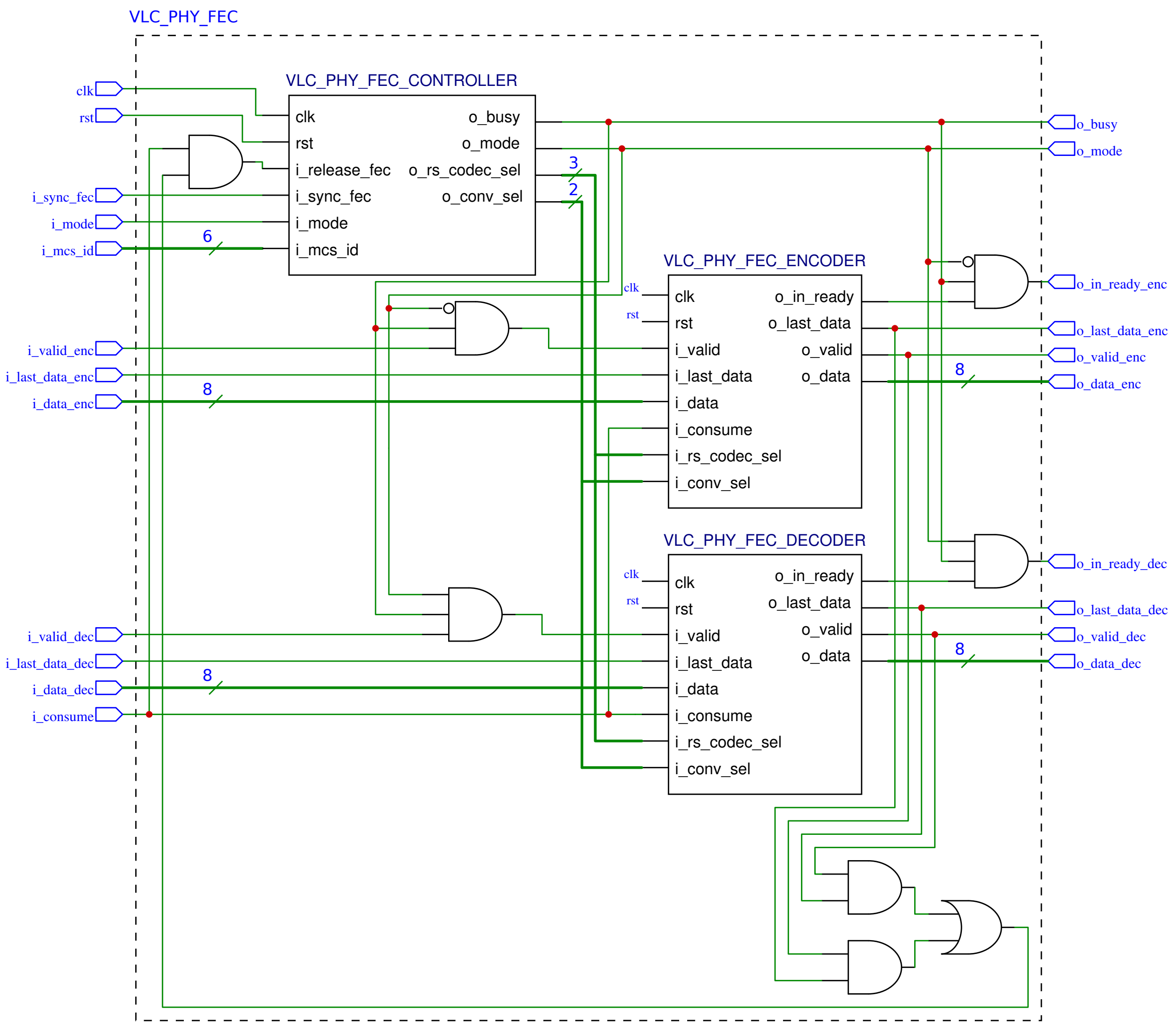
4.4.3 FEC Decoder

The FEC Decoder (*VLC_PHY_FEC_DECODER*) has the counterpart of the sub-blocks present in FEC Encoder arranged in the opposite order (Fig. 61). Similar to *VLC_PHY_FEC_ENCODER*, *i_conv_sel* and *i_rs_codec* are used to define the correct data path of input data signals. The main difference is the position of the switching logic in the FEC Decoder that requires the selection the FEC arrangement in the reverse order compared to *VLC_PHY_FEC_ENCODER*.

4.4.4 RS Codec for FEC

Since IEEE 802.15.7 specifies RS codecs with six different parametric configurations, it is required to create one instance of the IP core for RS codec (section 4.3.1) per each configuration, and the *VLC_PHY_RS_CODEC* is responsible for such arrangement (Fig.

Figure 39 – Architecture of *VLC_PHY_FEC*



62). This block is present in both *VLC_PHY_ENCODER* and *VLC_PHY_DECODER*, as it has a parametric entry which selects its mode: ENCODER or DECODER. The input port *i_rs_codec_sel* selects one of the RS codec instances using a demultiplexer for *i_valid* and a multiplexer for the outputs of them. Since RS codecs operate on codewords and has signals to delimit the starting and ending of them (*i_start_cw* and *i_end_cw*), there is a control mechanism using a counter based on the k - for RS Encoders - and N - for RS Decoders - parameters, which corresponds to the length of the expected input codeword for each case. Then, the only difference between *VLC_PHY_RS_CODEC* versions for ENCODER and DECODER is the constants used to delimit the input codeword and drive *i_start_cw* and *i_end_cw* ports (represented by dashed rectangle boxes in Fig.62).

4.4.5 Block Interleaver for FEC

The block interleaver for FEC is implemented by *VLC_PHY_INTERLEAVER*, which basically instantiates the interleaver (or deinterleaver) IP core with the parameter values specified by IEEE 802.15.7, as defined in Fig. 63. The value of 4382 for *NUMBER_OF_ELEMENTS* is based on the largest possible PHY I frame that the interleaver might receive. IEEE 802.15.7 defines in section 9.5.1 that the maximum PSDU size (*aMaxPHYFrameSize*) for PHY I is 1023 Bytes (2046 Nibbles), and the RS(15,7) - configuration that adds more parity symbols used with the interleaver - makes the largest (de)interleaver input frame be 3485 ($(15 \times 2046)/2$) Nibbles. Beyond the instantiation of the IP core for (de)interleaver, there is a simple logic responsible for determining the first Nibble of the frame.

4.4.6 Convolutional Encoder for FEC

The Convolutional Encoder for FEC is implemented by *VLC_PHY_CONV_ ENCODER*, as illustrated by Fig. 64. Beyond the instantiation of the IP core for Convolutional Encoder with the parameter specification in agreement with IEEE 802.15.7, it is required a PS converter and a puncturing block. The output data from the block interleaver is serialized bit-to-bit by the PS converter for the convoultional encoder. Also, the puncturing block receives the 1/3-rate parity outputs from the convolutional encoder, and it is able to convert them into 2/3-rate or 1/4-rate parity outputs if required.

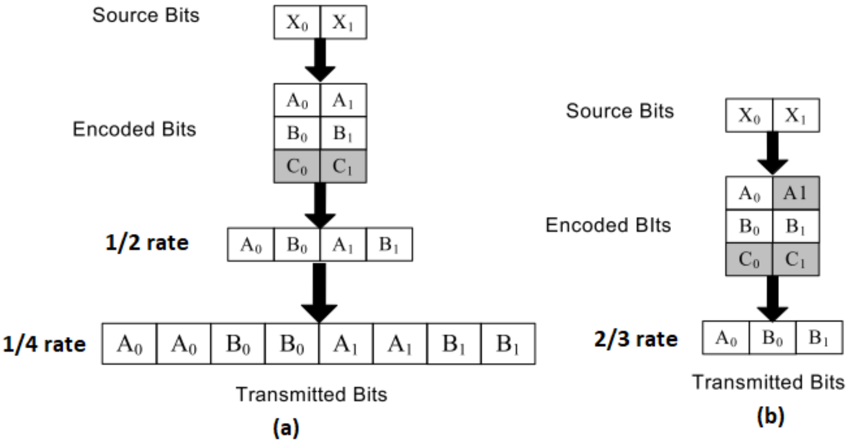
4.4.7 Viterbi Decoder for FEC

The Viterbi Decoder for FEC (*VLC_PHY_VITERBI_DECODER*) has an arrangement that is similar to *VLC_PHY_CONVOLUTIONAL_ENCODER*, but sub-blocks are placed in the opposite order (Fig. 65). The block receives a single bit from the external input data interface (Table 21), then it goes first to the depuncturing block, which is responsible for assembling the parity inputs for the Viterbi Decoder according to the selected rate. The IP core for Viterbi Decoder is instantiated according to IEEE 802.15.7 specification, and has a single bit data output. Since the deinterleaver block expects an input data of 4-bit width, a Serial/Parallel (S/P) converter is needed for the outputs of the Viterbi Decoder.

4.4.7.1 Puncturing and Depuncturing for rates 1/4 and 2/3

As seen in section 2.3.1, either the convolutional encoder or the viterbi decoder are configured with a rate of 1/3. However, there are operating modes in PHY I that require rates of 1/4 and 2/3 as depicted by Fig. 56. According to IEEE 802.15.2, the puncturing mechanism is employed to convert the 1/3-rate outputs into other rates. It adjusts parity outputs from 1/3-rate CC blocks by removing or replicating their bits as depicted by Fig. 40. For instance, 1/3-rate may be converted to 1/4-rate by removing the third bit of the 1/3-rate parity output and replicating the two remaining bits. The conversion from 1/3 to 2/3 rate is more complex since its puncturing procedure alternates the number of bits that are removed from 1/3-rate output - one or two bits - in each iteration.

Figure 40 – Puncturing of parity bits for rate conversion from 1/3 to 1/4 and 2/3. The 1/4 puncturing (a) removes the third bit of the 1/3 parity output and duplicates the remaining two bits. The 2/3 puncturing (b) performs a interleaved removal of one and two bits of the 1/3 parity output, and the remaining bits are transmitted.



The architectural design of the puncturing procedure performs bit-removal and replication operations on the outputs of the 1/3-rate convolutional encoder according to the selected rate. In order to avoid alternating output widths for 2/3-rate, there is an internal mechanism to always store the 2 bits resulted from 1-bit removal and send it together with the input bits of the next iteration, which collects just 1 bit from the 1/3-rate output. Then, the output width for 2/3-rate mode is always 3. Fig. 66 depicts the RTL architecture for the puncturing mechanism. The input *i_conv_sel* selects the output rate mode according to the Fig. 21. For 1/3-rate, this block just forward its inputs to outputs. For 1/4-rate, the input data conversion is carried out according to Fig.40; however, control signals are forward to their respective outputs similarly to 1/3-rate. Most complexity of the puncturing block is related to 2/3-rate mechanism. It requires two flops: one stores the punctured input data of the first iteration responsible for 1-bit removal, and the other controls *o_valid* and *o_in_ready* signals since it waits two valid inputs to transmit a single output data of 3-bit width.

The depuncturing counterpart presents a more complex RTL implementation (Fig.67) as the input frame is serialized bit-to-bit, then the block must assembly the output parity according to the selected rate. This is carried out by a shift register which accumulates input bits until it obtains a complete parity symbol, which is controlled by a counter and depends on the selected rate. The alternating input width of 2/3-rate is also managed by the block which expects 1 or 2 input data iterations each time to provide a valid output. This is the reason for accept only a single data bit, as it is easier to handle such alternating characteristic of 2/3-rate. However, such approach increases latency and worsens data rate as depicted by section 5.4, since it could receive the number of bits needed to feed the Viterbi decoder.

There are three separate sub-blocks for each supported rate which are responsible for assigning 'X' to parity bits that are unknown due to puncturing during the encoding process. It also converts logic values ('1','0', and 'x') into their corresponding quantized value because the IP core for adopted Viterbi Decoder IP uses a soft decision approach. Given that 1/4-rate parity has bit replication, its possible to compare A1 with A0 and B0 with B1 (Fig. 40), and they should be equal. If they are different, an 'X' value it is assigned already to the resulting parity output.

4.5 Summary

This chapter covered all the architectural aspects of the proposed FEC IP. It has been implemented according to IEEE 802.15.7, which specifies the functional ECC blocks that each operating block must contain. However, it does not define any details of the RTL architecture, which has been designed by this master thesis. Also, fundamental ECC blocks (RS codec and Interleaver) have been developed in HDL by this master thesis. Moreover, these resulting IPs are devised to be parameterizable, reliable and open source.

5 Results and Discussion

This chapter aims to characterize the FEC IP Core according to the following aspects: design size, timing, and power consumption. Related data is extracted from RTL synthesis tools for both FPGA and ASIC flows, and results are cross validated. Verification for the FEC IP is also covered in this chapter, and its methodology and related results are detailed. Throughput and latency are calculated for the FEC IP using the selected devices, and resulting values are checked against IEEE 802.15.7 requirements for those attributes. Finally, a closing discussion wraps up all results, compares them against previous works, highlights weaknesses and strengths of the proposed IP, and suggests improvements for it.

5.1 General Aspects of the FEC IP

The FEC IP has been implemented in VHDL-2008 according to the architecture described in Chapter 4. Table 1 displays some generic metrics of the IP. It has been extracted from Cadence® JasperGold® 2019.12v, using the Tcl command '*get_design_info*'. The number of lines, entities, instances and packages confirms the design complexity required to address multiple operating modes using outer and inner code mechanisms. The values obtained for FSMs and counters indicate that the IP has a decentralized controlling mechanism and corroborates the designed sub-IP block independence through well-defined port interfaces.

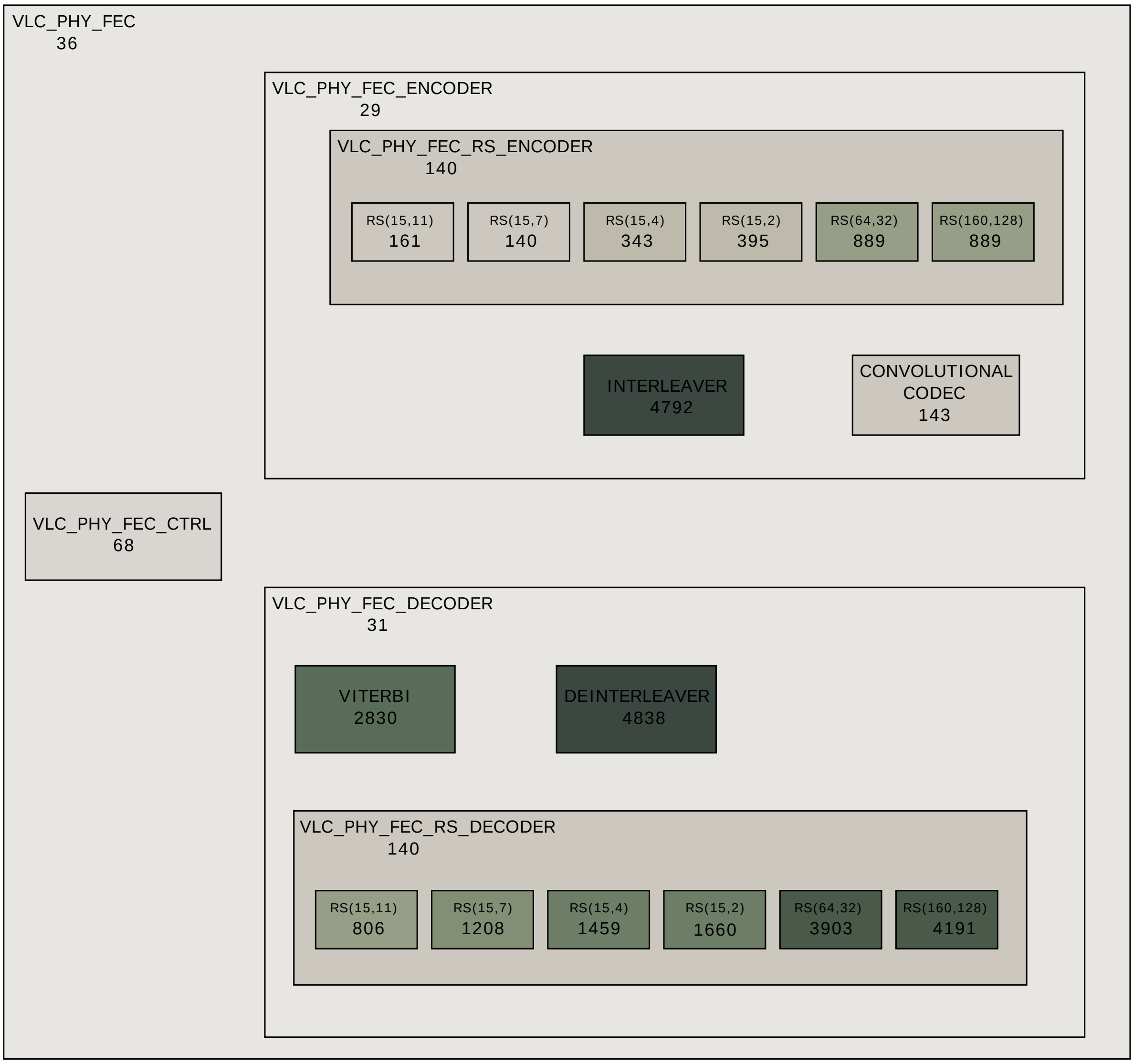
The number of signals contained in a IP block may be used as a rough estimate for design complexity. The FEC IP has almost 30K signals, and Fig. 41 split this number among its main sub-blocks. It clearly shows that decoding blocks such as RS codec and Viterbi has significantly more signals than their encoding counterparts. The Interleaver and the Deinterleaver blocks have close numbers as their functional mechanisms are very

Table 1 – General FEC IP Core metrics

Attribute	Value
Number of lines	34618
Packages	16
Entities	89
Instances	2336
Instance Levels	9
FSMs	33
Counters	101
Signals	29092

similar, and they instantiate memories to hold the whole input frame - and that is why there are more signals for them. Upper hierarchy instances have few signals since they are only responsible for the IP integration, but their functionality defines the switching mechanism in the data path, which enables the support for multiple operating modes. For that reason, verification of signal connectivity and control operation have high relevance for integrated IP Cores such as the one presented by this work.

Figure 41 – Main *VLC_PHY_FEC* instances with their number of signals



5.2 Verification

Chapter 4 presented the basic building blocks required by the IEEE 802.15.7 FEC: RS Codec, Interleaver, and CC. The verification for these sub-IPs is already done and described by the related publications [42, 46, 51]. The FEC IP instantiates them to implement the 30 operating modes specified by IEEE 802.15.7. Then, the verification task is to evaluate the the FEC control mechanism for all operating mode options. Such effort is mapped by 3 verification steps that are described in the next sub-sections: connectivity checking, controller inspection, and end-to-end examination.

5.2.1 Connectivity Checking

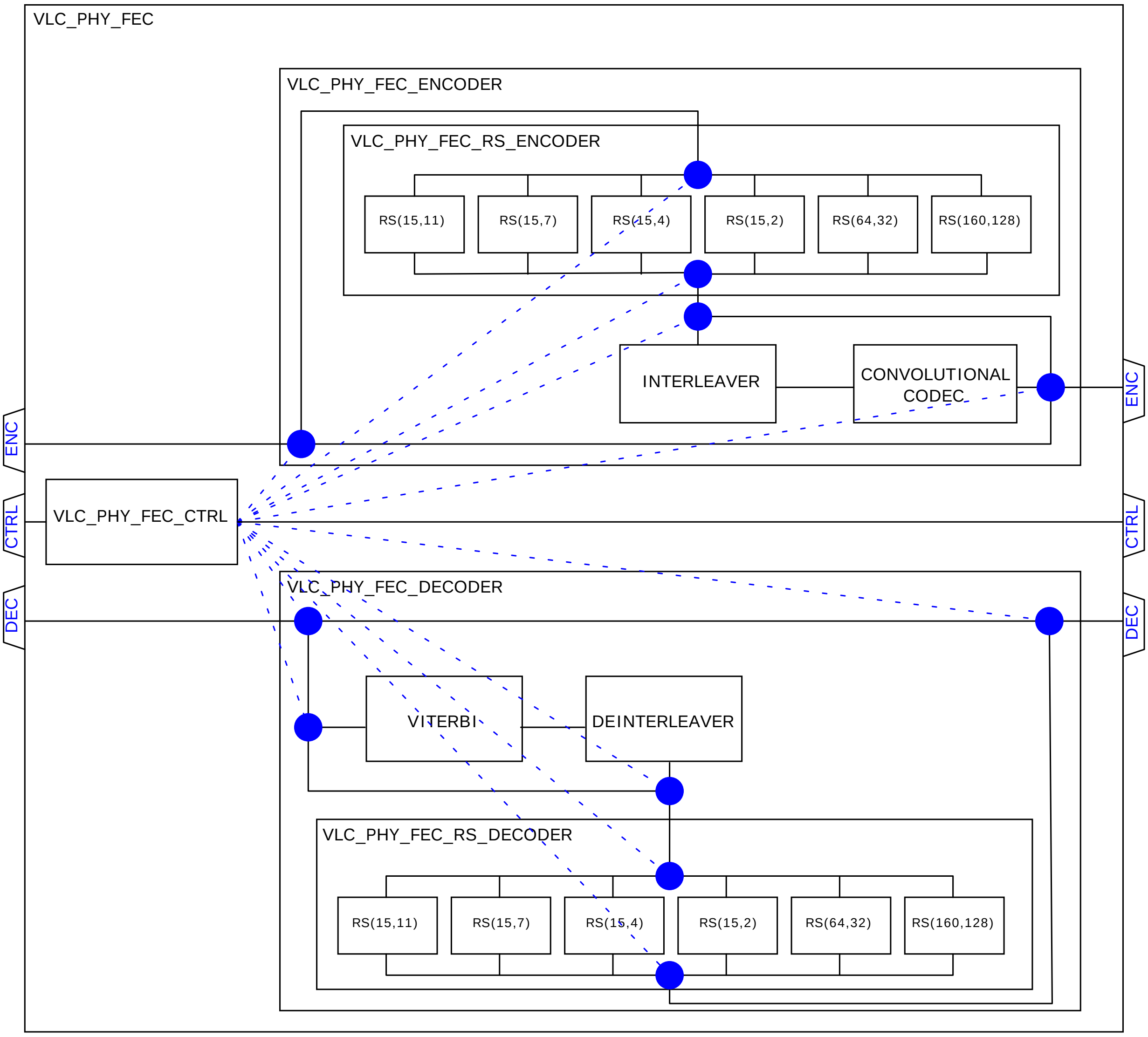
The first aspect that must be tested in an integrated IP Core is to certify that its sub-blocks are connected properly. Fig. 42 is a high level diagram that shows how the encoder and decoder data path should be wired in order to implement the required operating modes. The blue circles represent the nodes where switching control is required, which are controlled by *VLC_PHY_FEC_CONTROLLER* according to the configured *MCS_ID*. Hence, the *MCS_ID* is the precondition to attest the the data path connectivity for a specific operating mode.

Some Electronic Design Automation (EDA) tools already provide a solution for checking signal connectivity, and this thesis adopted the Connectivity (CONN) App provided by Cadence® JasperGold® 2019.12v. As illustrated by Fig. 43, it proves that a connection between source and destination signals exist if they are structurally connected - the source is part of the Cone of Influence (CoI) of the destination -, functional equivalent - the same values are assigned to both when the required precondition is met -, and toggle capable - source may have all possible value transitions.

The input for the CONN App is a Comma-Separated Values (CSV) file that has all target connections with source and destination signals and the condition expression required to attest the signal connectivity. Since generating a CSV file listing all required connections is a laborious task, the CONN App provides the reverse connectivity mechanism, which automatically generates an raw CSV file made based on a structural design analysis using a list of target source and destination instances. It has been generated one CSV file for each possible *MCS_ID*, and these csv files were manually scanned to check the completeness of the generated spec and the correctness of the condition expressions. After that, these CSV files were used as inputs by the tool to create the connection checks, and, finally, prove them. Fig. 44 summarizes the described verification flow, and it can be inferred that the "CSV Spec Generation" step only needs to be done when there is some change in the CONN specification of the FEC IP.

Table 2 displays the number of connections created for each *MCS_ID* for the encoder and decoder data paths - which have evenly numbers since they have analogous architectures. The *MCS_ID*s 0, 1, and 2 present more connections because they include connections of inner instances of CC (Parallel/Serial (P/S), S/P, puncturing and depuncturing), the (De)Interleaver, and the RS Codec in the analysis. The *MCS_ID*s 4, 8, 20, 29, 37, 38 have only the connections that bound the *VLC_PHY_FEC_ENCODER* and

Figure 42 – High level diagram of *VLC_PHY_FEC* with the wire connections between its main sub-blocks. Blue circles represent multiplexing elements controlled by *VLC_PHY_FEC_CTRL* according to the selected *MCS_ID*.



VLC_PHY_FEC_DECODER inputs to the outputs since they do not have any FEC treatment. The remaining *MCS_ID*s have few additional connections because of they need RS Codecs in the path. All specified connections (576) could be proven at the end of the design development without any issues.

5.2.2 Controller Inspection

As described by section 4.4.1, the *VLC_PHY_FEC_CONTROLLER* is a simple block responsible for configuring the FEC IP according to the values provided by the input control interface. This block is very suitable to Formal Verification (FV), which is

Figure 43 – Checks required for signal connectivity assessment: structural (a), functional (b), toggle (c) checks.

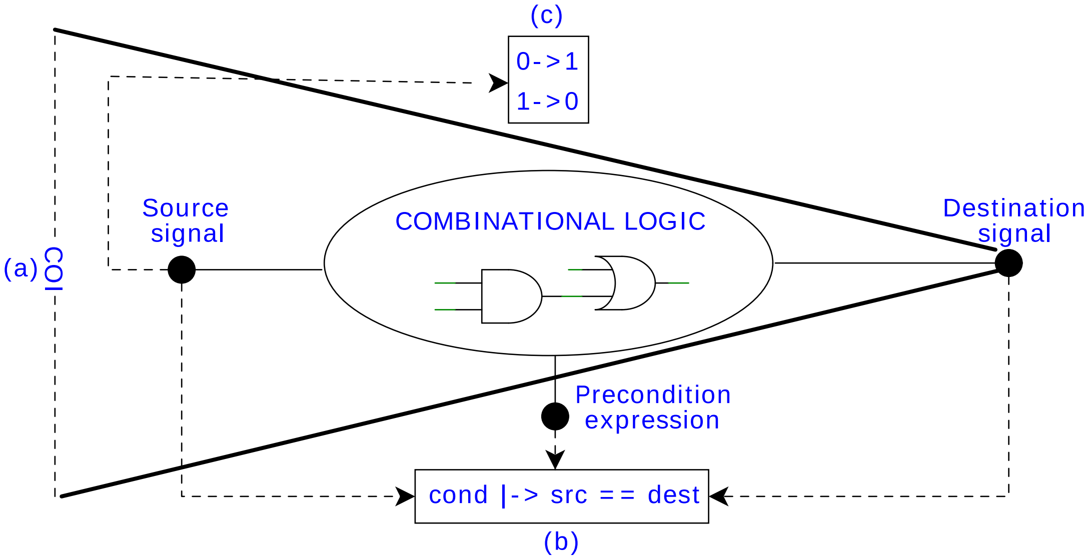
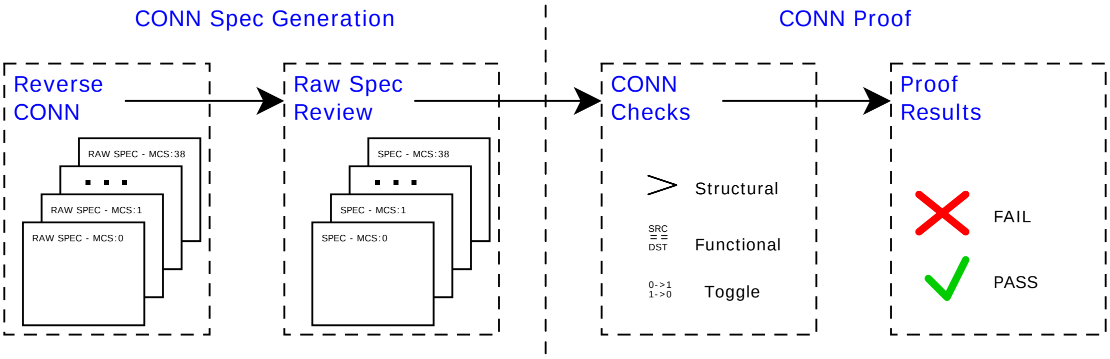


Figure 44 – Verification flow for data path signal connectivity using CONN App.



a method that mathematically analyzes the state space of a Design Under Test (DUT), rather than computing results for particular values as seen in Dynamic Assertion-Based Verification (ABV) techniques such as directed tests, constrained-random simulation [53]. System requirements are expressed in System Verilog Assertions (SVA) properties, either assertions - to capture a design behavior to be proven - or covers - to monitor an expected design functionality. In order to restrict the set of acceptable input value sequences to the scenarios of interest, assume properties are also used to constrain functional behaviors in the DUT. These properties are processed by the Formal Verification (FV) tool, which provides Counter Example (CEx) traces or proven status for assertions and rechability traces or unreachable status for covers.

Table 2 – Total number of data path connections (encoder + decoder) for each *MCS_ID*

MCS_IDs	Connections
0, 1, 2	64
4, 8, 20, 29, 37, 38	8
3, 5, 6, 7, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26 ,27, 28, 32, 33, 34, 35, 36	16

Table 3 – System requirements for *VLC_PHY_FEC_CONTROLLER*

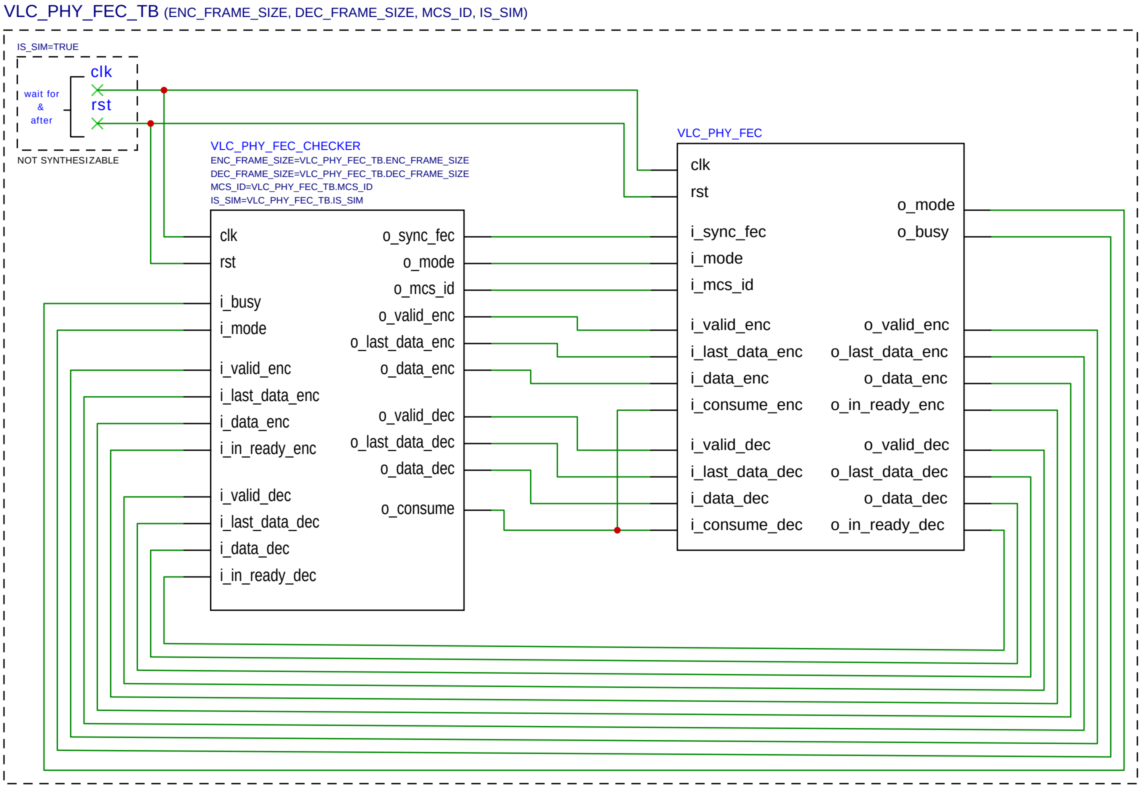
System Requirement	Description	Property Expression
Locking mechanism	FEC may be configured by asserting *i_sync_fec* whenever *o_busy* is de-asserted. When it happens, *o_busy* is asserted one cycle after, and *o_mode* assumes the past value of *i_mode*.	*i_sync_fec and not o_busy and not i_release_fec |=> o_busy and o_mode = $past(i_mode)*
Release mechanism	If FEC finishes encoding or decoding frame transmission, *i_release_fec* is asserted. FEC is released (*o_busy = 0*) whenever *i_release_fec* is asserted and *i_sync_fec* is de-asserted.	*i_sync_fec and not o_busy and not i_release_fec |=> o_busy and o_mode = $past(i_mode)*
MCS_ID conversion	*MCS_ID* is converted to *o_rs_codec_sel* and *o_conv_sel* according to Table 21, and it is enabled whenever the FEC is locked.	*i_sync_fec and not o_busy and i_mcs_id = ${mcs_id} |=> o_rs_codec_sel = ${rs_ref} and o_conv_sel = ${conv_ref}*

Table 3 describes the verification requirements that encompasses the block functionality. It also shows the SVA property expressions that are bound to their related requirements. Assertions and Covers were created using these expressions. Cadence® JasperGold® 2019.12v has been used to attest these properties and all of them could be proven by the tool in few seconds.

5.2.3 End-to-End Examination

In order to exercise the data path of the FEC IP Core - for both encoder and decoder portions -, a synthesizable test bench has been created. Fig. 45 illustrates the high level aspects of the "End-to-End" (*VLC_PHY_FEC_TB*) test bench. The internal signals *clk* and *rst* represent the clock and reset of the system and are driven either by non-synthesizable VHDL test bench constructs (*wait for* and *after*) in simulation tools, or by tool constrains (assumptions) in formal tools. All other DUT signals are connected to *VLC_PHY_FEC_CHECKER*, which is responsible for running the following actions: configures FEC as encoder, feeds the original frame to the encoder interface, stores the encoded frame, configures FEC as decoder, feeds the encoded frame to the decoder interface, stores the decoded frame and compares it against the original frame. In this way, the complete FEC data path is exercised.

Figure 45 – High level view of the the test bench *VLC_PHY_FEC_TB* responsible for handling the "End-to-End" examination.



The test bench has four parameters: *ENC_FRAME_SIZE*, the size of the encoding (original) frame; *DEC_FRAME_SIZE*, the size of the decoding frame; *MCS_ID*, definition for the target operating mode; and *IS_SIM*, setting for adding non-synthesizable constructs used by the simulation environment. All these parameters are used by *VLC_PHY_FEC_CHECKER*, and its RTL architecture is described by Fig. 46. The entity *VLC_PHY_FEC_CHECKER_CTRL* implements a FSM responsible for configuring the FEC, enabling validity signal flags, and also increment addresses of the memories, which store the encoding and decoding frames. The Read-Only Memory (ROM) memory (*SINGLE_PORT_LINEAR_ROM*) holds the encoding frame, whereas the RAM memory (*SINGLE_PORT_LINEAR_RAM*) keeps track of the stored decoding frame produced by the FEC itself. Their addresses are indexed by counters, which are controlled by a simple combinational logic, which increments the memory addresses in the following scenarios:

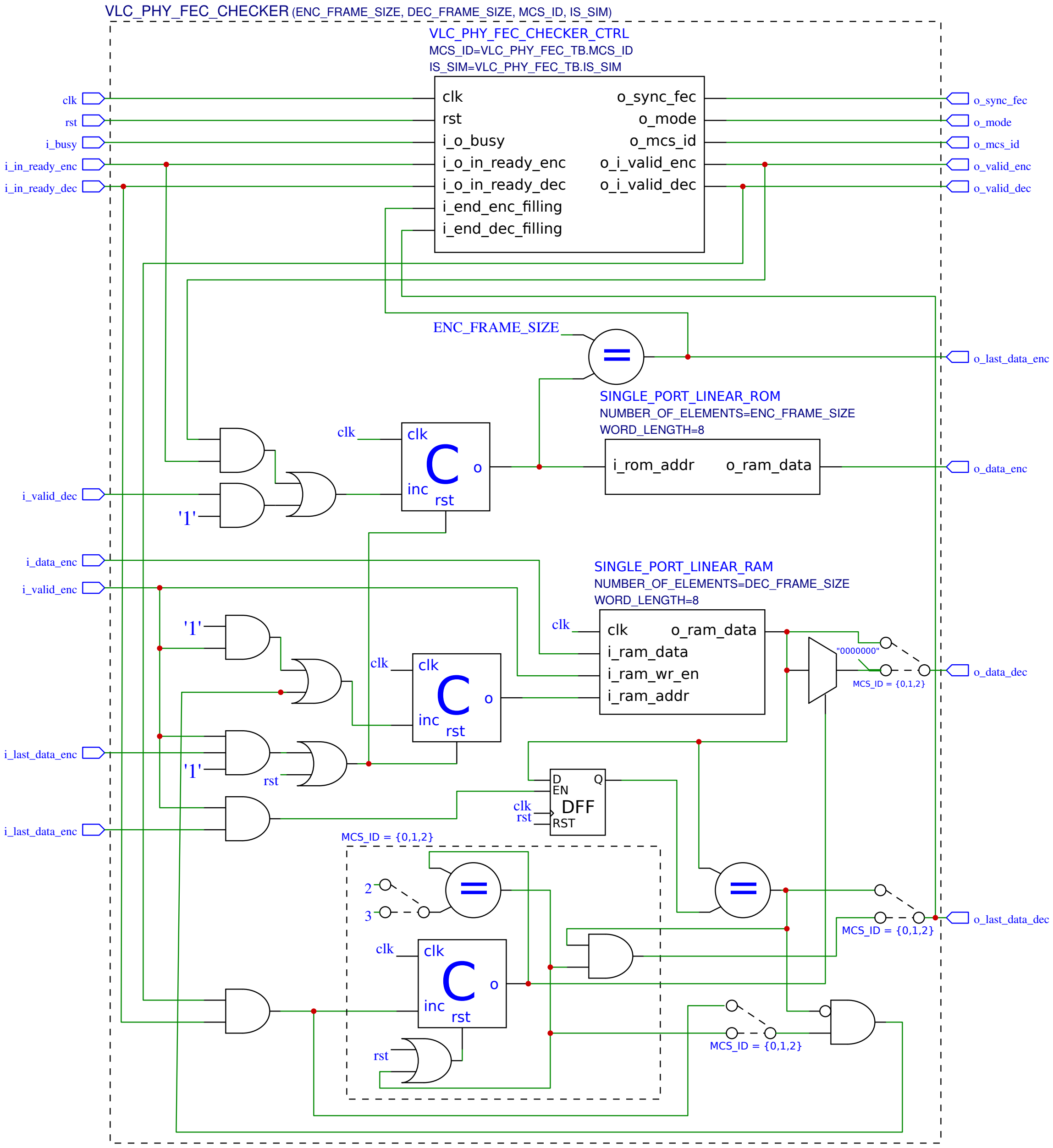
- To feed the FEC encoder (*o_i_valid_enc* = '1') with the ROM whenever it is ready to receive data (*i_o_in_ready_enc* = '1').
- To store the encoded frame (*i_o_valid_enc* = '1') into the RAM.
- To feed the FEC decoder (*o_i_valid_enc* = '1') with the RAM whenever it is ready to receive data (*i_o_in_ready_dec* = '1').
- To retrieve the original frame dato from the ROM to compare it against the decoded frame from FEC (*i_o_valid_dec* = '1')

There are also some constant '1' values driving the referred combinational logic. They represent the FEC input *i_consume*, which is always '1' for a matter of simplicity. It must be taken into account then its value is dynamic since address counters cannot be incremented when it is '0'. In order to determine the last frame data, there are two comparators - for the encoding e decoding frames -, and it is required a register to store the size of the encoded frame to be used as reference when it is delivered to the decoder portion of the FEC.

When the selected operating mode has inner code (*MCS_ID*s = 0,1, and 2), an additional logic is required to deal with the misalignment between the effective data slices of the FEC encoder output (4 or 3 bits) and the FEC decoder input (1 bit) as displayed by Table 21. In this case, there is an additional counter which indexes the output data of the RAM memory bit-to-bit, and it is sent to the FEC decoder interface. Whenever it reaches the last position of the effective slice of the RAM memory data output, this auxiliary counter is reset, and the RAM address counter is incremented.

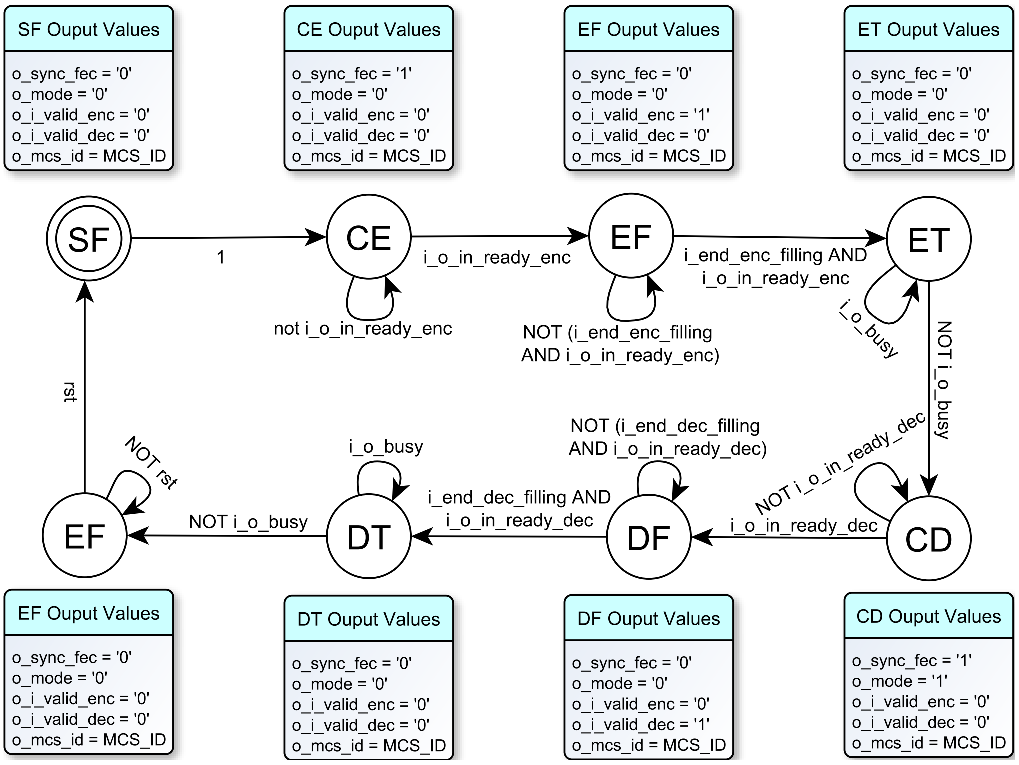
The FSM implemented by *VLC_PHY_FEC_CHECKER_CTRL* is depicted by Fig. 47. The initial state is *START_FLOW* (SF), which is reached when reset procedure is done. Then, it moves to *CONFIG_ENCODER* (CE), which configures the FEC for the encoding mode using the parameter MCS_ID. Once the FEC is ready to receive data, *ENC_FILLING* (EF) is reached, and it fills the FEC with the data stored into the ROM. As the complete frame is delivered to the FEC, *ENC_TX*(ET) waits the end of the transmission of the encoded frame, which is stored into the RAM. An analogous process is carried out for the decoder portion of the FEC with the states *CONFIG_DECODER* (CD), *DEC_FILLING*(DF), and *DEC_TX*(DT). The process finishes at *END_FLOW* (EF), and the FSM remains stuck at this state unless the test bench is reset.

Figure 46 – RTL architecture of *VLC_PHY_FEC_CHECKER*.



The goals of the "End-to-End" test bench are to certify that *END_FLOW* state is reached and guarantee that the output frame of the FEC decoder is equivalent to the original frame stored into the ROM for each MCS ID. Cadence® JasperGold® 2019.12v has also been used in this task, and these described checks were mapped into SVA assertions. However, due to the design complexity and the bound required to complete the whole encoding and decoding process, the memory required by the tool during formal analysis overcomes the maximum available resources (16 Gb). Hence, a simulation-based approach fits this activity since all input signals of the FEC is constrained by the test bench.

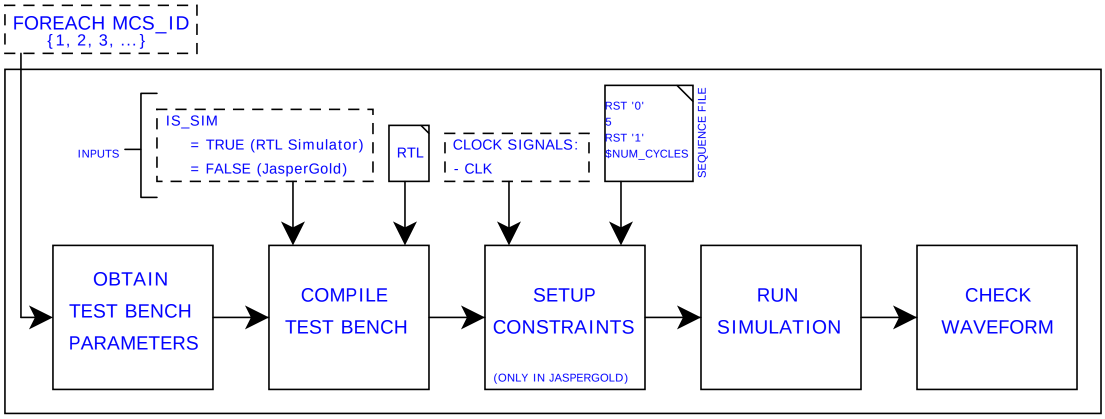
Figure 47 – FSM implemented by *VLC_PHY_FEC_CHECKER_CTRL*.



Even though JasperGold® is formal verification tool, it was possible to implement - in Tcl - a simulation-based flow that is acceptable by either JasperGold or any other RTL simulator, as illustrated by Fig. 48. For each MCS ID, test bench parameters (*ENC_FRAME_SIZE*, *DEC_FRAME_SIZE*, and *MCS_ID*) are obtained based on the original frame size. This frame is stored into the ROM memory of the test bench, and their values are informed by a VHDL package *VLC_PHY_FEC_CONSTANTS*, whose a python script generated it using random values. Then, the test bench is compiled with intended parameters and *IS_SIM* is selected according to the target tool. A step specific to Jaspergold is required to setup the clock and reset constraints, whereas it is described in the test bench itself for RTL simulators. The sequence file informs the reset to the tool and does the trick that enables using JasperGold as a RTL simulator. This file provides "signal & values" and the number of cycles that it holds, and it is used during the "Reset Analysis" - a simulation step required to obtain the initial design state for checking design properties. Jaspergold provides a way to open such reset trace in its waveform viewer with the regular debug capabilities, and that is why this work stuck to it. Finally, the last two steps are responsible for running the simulation and checking whether the resulting waveform has the expected functional behavior or not.

Short (63B) and long (1023B)frames were used in this verification step. It takes ~5hrs to run this verification routine for all MCS IDs with the available computational

Figure 48 – Routine used to assess the *VLC_PHY_FEC* for each MCS ID using the "End-to-End" test bench



resources (Intel® Core® i7-7700 3.6 GHz with a RAM memory of 16 GB). Some defects have been caught, and most of them are related to the logic outside the base IP blocks (RS Codec, Interleaver, Convolutional Codec or Viterbi) and have been fixed. However, an exception was a defect caught inside the Viterbi IP Core, which is probably related to its memory controller. As it is a third-party open IP [51], the solution for it could not be figured out on time for this thesis. However, a workaround has been found for it, which was to increase the window length to the size of the input frame. It was decided to maintain the original value assigned to *MAX_WINDOW_LENGTH* (96) in the parametric (design-time) configuration of the Viterbi IP. The reason for that is because such value has already been used in its related test benches and at least the short frames could be validated for the FEC IP by changing the run-time configuration for the window length from 42 to 63 in the control interface of the Viterbi IP. However, it is expected a considerable design area penalty since memory has twice more elements.

Despite being successful to guarantee core functional behaviors of the FEC IP, there are some enhancements in this flow that could improve the confidence level of the IP and catch more corner case defects. For instance, *o_in_consume* is constant ‘1’, and it could be dynamically altered by the test bench to stimulate other scenarios. However, the FEC signal *i_consume* has direct connections to the base IP blocks, which have already been verified standalone. Then, the connectivity check for i_consume - described by section 5.2.1 - is at least a sanity check for its related functional behavior. In the same way, the validity signals of the *VLC_PHY_FEC_CHECKER* (*o_i_valid_end* & *o_i_valid_dec*) have contiguous values, and they could assume other value combinations. Furthermore, the frame store into the test bench ROM could also be dynamically changed in each run since *VLC_PHY_FEC_CONSTANTS* can be regenerated. Finally, a regression mechanism with all these proposed enhancements could be implemented to run continuously and more malfunctions could be caught. Nonetheless, it requires dedicated computational resources, and for the purpose of this thesis, which is to obtain a certain level of confidence that guarantees that the proposed *VLC_PHY_FEC* architecture will be stable, the employed verification effort is enough as typical use cases could be covered by it.

5.3 Synthesis Results

In order to typify the proposed FEC IP, design size, timing and power consumption are the main attributes used to qualify a given RTL design applied to a target device technology. They can be estimated during the synthesis step of either ASIC or FPGA development flows [54]. In this work, both were used to collected synthesis results and assess the FEC IP Core.

5.3.1 Device Technologies

Synthesis results are provided by EDA software tools based on a specified device technology. For the FPGA synthesis flow, the device 5CGXFC5C6F27C7N from Altera® Cyclone® V family was selected and results were extracted from Quartus® Prime Lite Edition 18.1.0, whereas TCB018GBWP7T stardard cell library from Taiwan Semiconductor Manufacturing Company (TSMC)® 0.18μm CMOS process was adopted for ASIC synthesis flow and results were extracted from Cadence® Genus™ *17.20-p003_1v*.

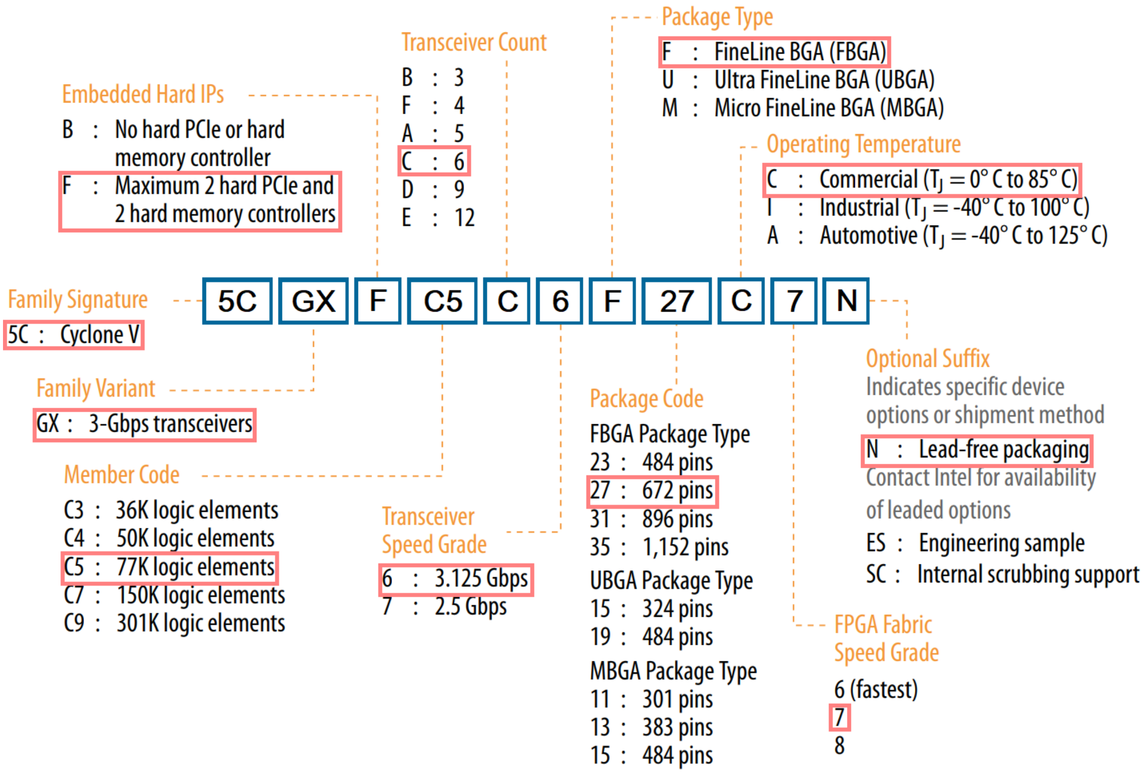
5.3.1.1 Altera® Cyclone® V - 5CGXFC5C6F27C7N

The family Cyclone® V has been conceived for small-form-factor applications that are cost- and power-sensitive, and its main features among others are [55]:

- TSMC's 28 nm low power process technology with core voltage of 1.1V
- 614 Mbps to 5.0 Gbps integrated transceiver speed
- 550 MHz global clock network
- Includes hard IP blocks for embedded transceiver Input/Output (I/O), variable-precision Digital Signal Processor (DSP), and memory controllers (Double Data Rate (DDR)3, DDR2, Low-Power Double Data Rate (LPDDR), and LPDDR2)
- General Purpose Input/Output (GPIO)s with 875 Mbps Low-voltage Differential Signaling (LVDS) receiver and 840 Mbps LVDS transmitter
- 8-input Adaptative Logic Module (ALM) with four registers

The device name (5CGXFC5C6F27C7N) defines some device options with its nomenclature as illustrated by Fig.49. One of the main aspects is the "Family Variant", which impacts the overall architecture of the device. In this case, the "GX" variant has a design optimized for the lowest cost and power requirements, and it is compliant with transceiver applications. The other terms mainly impact capacity, timing and operation constrains of the device.

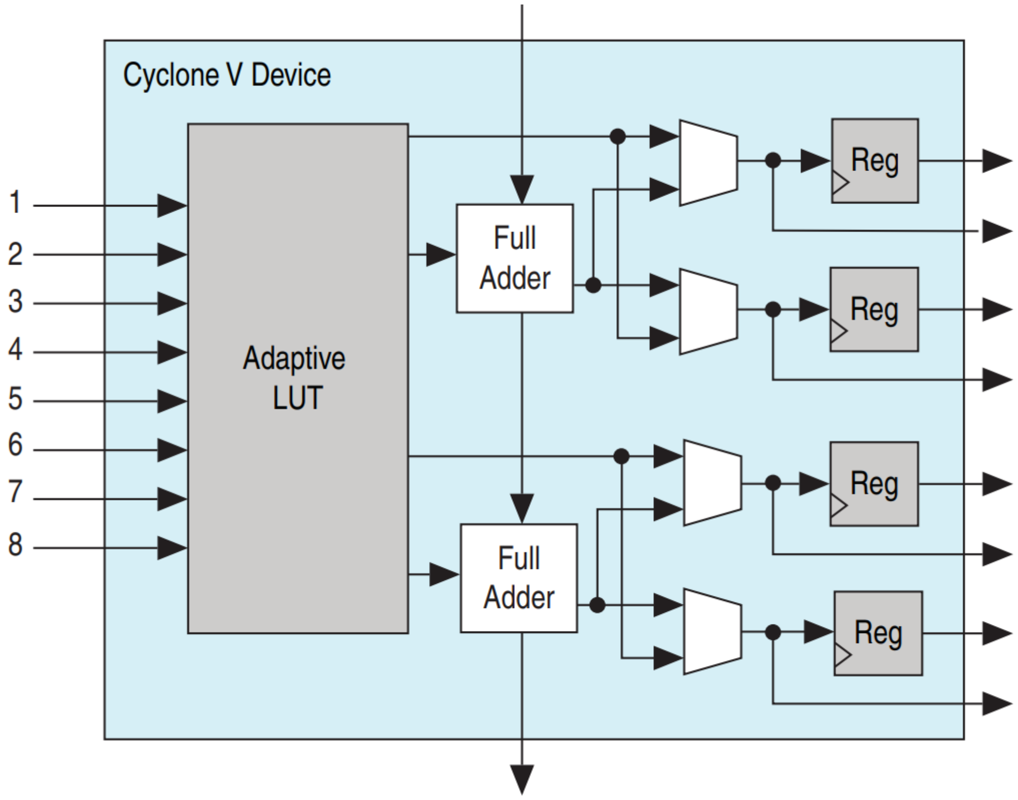
Figure 49 – Nomenclature explanation for 5CGXFC5C6F27C7N. (figure adapted from [56])



The ALM is the basic building block of the FPGA device. It consists of a Adaptive Look-up Table (ALUT) - which models combinatorial logic -, 4 registers and 2 adders as shown in Fig. 50. ALMs are usually grouped into Logic Array Block (LAB)s, which have a restricted number of signals that may drive it. 5CGXFC5C6F27C7N has 56,415 ALMs, and they are configured and interconnected accordingly to implement the user-input RTL design.

Cyclone® V GX is suitable for VLC applications since it is targeted for IoT communication applications. It also has a transceiver that could be used to drive the VLC front-end circuit used to supply the photodetectors and LEDs as its speed (Mbps) is greater than the Maximum Optical Rate specified by IEEE 802.15.7 - 120 MHz (or 120 Mbps) for OOK and VPPM modulation schemes. Moreover, 5CGXFC5C6F27C7N has enough ALMs to implement the FEC IP Core.

Figure 50 – ALM architecture in Cyclone® V [55]. Reports for resource usage in FPGA from Quartus® Prime are in terms of ALMs.



5.3.1.2 TCB018GBWP7T stardard cell library from TSMC® $0.18\mu m$ CMOS

TCB018GBWP7T is a standard cell library of $0.18\mu m$ node technology which contains vast set of combinational and sequential logic cells for digital designs. Annex C describes the core cell types that maps most RTL constructs described in a HDL language. The manual for TCB018GBWP7T [57] details its DC and Alternating Current (AC) characteristics (e.g operating conditions, propagation delay, power dissipation, and etc) and the datasheet for each individual cell. This information is mapped into a Liberty library format ($\dot{l}ib$), which is used by the synthesis tool to report area, timing and power consumption of a given RTL design.

Many IoT applications require low fabrication costs and less demanding system requirements, and older technology nodes, such as $0.18\mu m$, might be feasible for them. The literature already contains successful Integrated Circuit (IC) developments of wireless transceivers for IoT devices using $0.18\mu m$ node [58, 59]. Since VLC has IoT as one of its main targets, collecting synthesis results for FEC IP Core based on TCB018GBWP7T provides an early estimate regarding the feasibility such node technology.

Table 4 – Resource usage summary for the FEC IP in 5CGXFC5C6F27C7N

Resource	Usage
Logic utilization (ALMs needed / total ALMs on device) [=A-B+C]	48,001 / 56,480 (85%)
[A] ALMs used in final placement [=a+b+c+d]	50,935 / 56,480 (90%)
[a] ALMs used for Look-up Table (LUT) logic and registers	11,726 / 56,480 (21%)
[b] ALMs used for LUT logic	17,228 / 56,480 (31%)
[c] ALMs used for registers	21,981 / 56,480 (39%)
[d] ALMs used for memory	0 / 56,480 (0%)
[B] Estimate of ALMs recoverable by dense packing	9,123 / 56,480 (16%)
[C] Estimate of ALMs unavailable	6,189 / 56,480 (11%)
Total combinational ALUT usage for logic	44,976
Combinational ALUT usage for route-throughs	15,134
Dedicated logic registers	75,450

5.3.2 Design size

For a FPGA device, design size means how many ALMs are required to implemented a given RTL design. Table 4 displays the resource Usage summary for the FEC IP in 5CGXFC5C6F27C7N using the ‘Normal’ compilation flow - even balance between area and timing performance. It spent 48,001 (85%) ALMs of the target device, which implemented 44,976 ALUTs for combinational logic and 75,450 registers. Nonetheless, 15,134 ALUTs (34% of the total usage) are used for "route-throughs", required when the drivers of a register are external to the ALM that it belongs to.

ALMs are evenly distributed for combinational and sequential logic: 39% for registers, 31% for LUT logic and 21% for both. No ALMs or Memory Logic Array Block (MLAB)s are dedicated for memory because the setting "Auto ROM/RAM Replacement" is disabled, then memories are mapped into regular design LUT and registers. The motivation for that is related to timing issues when it is enabled - explained in section 5.3.3. Moreover, this behavior also helps the cross analysis between FPGA and ASIC results for design size as the latter also maps memories in plain liberty cells.

Some ALMs are required not because of the implemented RTL design, but due to the synthesis step of ‘Place & Route’ itself. For instance, 9,123 are considered recoverable because they were only used for timing optimization, but they may be removed as the design grows or when the aggressive packing mode is enabled. Around 6,189 ALMs are unavailable due to design and device constraints, and most of them are in this situation due the "LAB-wide signal conflicts". It means that these populated ALMs use all control inputs (e.g clock, sload, and etc) available for their related LABs, then logic that requires other control inputs cannot be added. Hence, these LABs are constrained by the number of available inputs and not by the ALMs utilization. There are other reasons for unavailable ALMs, which may be found in the Quartus® help page for compilation reports [60].

Table 5 – Standard cell report summary for the FEC IP using TCB018GBWP7T

Type	Instances	Area (μm^2)
sequential	75,074	3,982,745 (68%)
inverter	9,299	128,730 (2%)
buffer	1,871	36,339 (1%)
logic	111,316	1,675,027 (29%)

Table 6 – Standard cell report summary for the FEC IP using GSC 3.0v library

Type	Instances	Area (μm^2)
sequential	75,074	11,022,936 (55%)
inverter	13,325	331,237 (2%)
buffer	3,836	114,204 (1%)
logic	173,977	8,451,421 (42%)

The reconfigurable capacity of FPGA devices comes with the penalty of an increased resource usage, whereas ASIC devices do not present the referred overheads due to its customizable characteristics. Design size results were also collected for TCB018GBWP7T standard cell library, and results are summarized in Table 5. The total number of sequential instance cells (75,074) is comparable to the total number of dedicated registers reported for the adopted FPGA device (75,450). The same comparison cannot be precisely done for combinational logic (inverter, buffer, and logic types) since it is implemented by LUTs in FPGA - which accepts up to 7 inputs and build up a truth table for a boolean expression, whereas TCB018GBWP7T only provides basic gate cells (e.g AND, OR, XOR, and etc) with at most 4 inputs. Then, if ALUTs used for "route-through" purpose are excluded, the average rate of standard combinational cells per ALUT is 4.1x. The area results are the total sum of the cell instance area specified by an attribute in the library cell definition.

In order to obtain a comparative analysis for the design scale of the FEC IP among other open-access IPs, the International Workshop on Logic and Synthesis (IWLS) 2005 benchmark data - an initiative from Cadence Research Laboratories at Berkeley to compare synthesis results of Open Access IPs using the 0.18μm Generic Standard Cell (GSC) 3.0v library - were analyzed [61]. Then, synthesis results for the FEC IP were obtained using the same standard library cell (GSC 3.0v) adopted by the benchmark initiative. As depicted by Table 6, this library requires more combinational cells than TCB018GBWP7T to synthesize the same digital circuit and the total area is increased by more than 4x. The reason for that is because GSC 3.0 has a more elementary set of cells compared to TCB018GBWP7T. For more detailed information, section C.3 and C.4 in annex C provides a detailed report of cell usage for both libraries.

Tables 7 and 8 situate the FEC IP among other selected RTL designs from the benchmark (8 out of 84). The amount of resource required by FEC IP is considerably

Table 7 – Description of the selected IPs for the comparison analysis

Design	Source	function
usb_phy	OpenCores®	Universal Serial Bus (USB) 1.1 PHY layer
spi	OpenCores®	Serial Peripheral Interface (SPI) protocol
pci_bridge32	OpenCores®	Peripheral Component Interconnect (PCI) bus
dsp	Faraday® Technology	16-bit DSP
ethernet	OpenCores®	Ethernet IP Core
risc	Faraday® Technology	32-bit Reduced Instruction Set Computer (RISC) Central Processing Unit (CPU)
vlc_fec_ip	**This master thesis**	FEC compliant with IEEE 802.15.7
leon3mp	Cobham Gaisler®	Leon ® 3 radiation-tolerant 32-bit CPU
leon2	Cobham Gaisler®	Leon ® 2 radiation-tolerant 32-bit CPU

Table 8 – Standard cell report summary for the selected IPs using GSC 3.0v library

Design	Sequential	Inverter	Buffer	Logic	Total
usb_phy	98	118	17	313	546
spi	229	462	49	2,487	3,227
pci_bridge32	3,359	3,095	100	10,262	16,816
dsp	3,611	5,258	42	23,523	32,436
ethernet	10,544	3,404	234	32,557	46,739
risc	7,599	7,370	126	44,872	59,974
vlc_fec_ip	75,074	13,325	3,836	173,977	266,212
leon3mp	149,381	104,393	14,964	511,665	780,403
leon2	108,839	87,122	3,303	346,539	545,803

higher than the ones referred by OpenCores® and Faraday® Technology Coorporation IPs, but it is still behind the Leon® processor for aerospace applications developed by Cobham Gaisler®. In order to understand the culprits that makes FEC IP larger than other communication IPs such as *ethernet*, a resource utilization view based on the RTL design hierarchy is useful for such analysis. Table 9 and 10 display design size data for the main FEC IP sub-blocks based on synthesis results from from both FPGA (5CGXFC5C6F27C7N) and ASIC (TCB018GBWP7T) synthesis tools.

As expected, design size results by entity from 5CGXFC5C6F27C7N and TCB018GB WP7T are comparable. Table 11 refers to percentage of each block occupied by memory blocks (FIFOs or SRAMs), and it is clear that memory dominates the resource utilization of the FEC IP: 83% for ALM utilization in 5CGXFC5C6F27C7N and 75% of the IC area in TCB018GBWP7T. If memory were excluded from this analysis and treated as an external device that connects to the FEC IP, the results for design size would be in the same order of magnitude as the Ethernet IP from OpenCores®. From the results observed in Table 9 and 10, the following enhancements can reduce the design size of the FEC IP:

Table 9 – Resource usage by instance for the FEC IP in 5CGXFC5C6F27C7N (percentage number in parenthesis)

Instance	ALMs	ALUTs	Registers
vlc_phy_rs_encoder_15_11	40 (0.08%)	62 (0.14%)	33 (0.04%)
vlc_phy_rs_encoder_15_7	50 (0.10%)	77 (0.17%)	49 (0.06%)
vlc_phy_rs_encoder_15_4	56 (0.12%)	92 (0.20%)	61 (0.08%)
vlc_phy_rs_encoder_15_2	58 (0.12%)	97 (0.22%)	69 (0.09%)
vlc_phy_rs_encoder_64_32	184 (0.38%)	348 (0.77%)	279 (0.37%)
vlc_phy_rs_encoder_160_128	185 (0.38%)	354 (0.79%)	281 (0.37%)
vlc_phy_fec_interleaver	12184 (23.38%)	10419 (23.17%)	17648 (23.36%)
vlc_phy_convolutional_codec	20 (0.04%)	35 (0.08%)	21 (0.03%)
vlc_phy_rs_decoder_15_11	293 (0.61%)	450 (1.00%)	448 (0.59%)
vlc_phy_rs_decoder_15_7	369 (0.77%)	553 (1.23%)	592 (0.78%)
vlc_phy_rs_decoder_15_4	412 (0.86%)	594 (1.32%)	703 (0.93%)
vlc_phy_rs_decoder_15_2	471 (0.98%)	676 (1.50%)	766 (1.01%)
vlc_phy_rs_decoder_64_32	2802 (5.84%)	3653 (5.60%)	4199 (5.56%)
vlc_phy_rs_decoder_160_128	3890 (8.10%)	4806 (8.12%)	6522 (8.63%)
vlc_phy_fec_deinterleaver	12876 (26.83%)	10436 (23.20%)	17626 (23.33%)
vlc_phy_viterbi	13934 (29.03%)	12033 (26.75%)	26059 (34.46%)
others	176 (0.52%)	291 (0.65%)	184 (0.08%)
vlc_phy_fec (total)	48000 (100.00%)	44976 (100.00%)	75540 (100.00%)

- RS codec could be improved to accept N and K configurations in run time instead of being static parameters in its generic interface. Then only a single instance of a RS codec would be required, and its size would be around the worst case (RS(160, 128)). It represents $\sim$ 9% of ALM count reduction (Table 9) in 5CGXFC5C6F27C7N and $\sim$ 13% of area reduction in TCB018GBWP7T (Table 10) for the FEC IP.

- If the premise that the FEC IP never encodes or decodes data simultaneously is true, then a single Static Random-Access Memory (SRAM) memory could be shared between the Interleaver and the De-interleaver. Given that more than 99% of the (De)Interleaver is composed by the memory instance (Table 11), such memory sharing mechanism would reduce ALM utilization (5CGXFC5C6F27C7N) and area (TCB018GBWP7T) in more than 20% for the FEC IP.

- For the Viterbi IP, *MAX_WINDOW_LENGTH* parameter value (96) is greater than the required for it (42). It is directly related to the size of its internal memories. Then, the appropriate parametric value reduces the design size of *vlc_phy_viterbi* by almost half as memory utilization represents more than 80% of such block. Then, it would reduce ALM utilization (5CGXFC5C6F27C7N) and area (TCB018GBWP7T) by $\sim$ 15%. However, the default parameter for *MAX_WINDOW_LENGTH* could not be change because of a bug reported in section 5.2.3.

Table 10 – Area by instance for the FEC IP in TCB018GBWP7T (percentage number in parenthesis)

Instance	Cells	Cell Area (μm^2)	Routing Area (μm^2)	Total (μm^2)
vlc_phy_rs_encoder_15_11	188 (0.10%)	3679 (0.06%)	1702 (0.08%)	5381 (0.07%)
vlc_phy_rs_encoder_15_7	247 (0.13%)	5251 (0.09%)	2194 (0.10%)	7445 (0.09%)
vlc_phy_rs_encoder_15_4	315 (0.16%)	6875 (0.12%)	2788 (0.13%)	9663 (0.12%)
vlc_phy_rs_encoder_15_2	325 (0.16%)	7361 (0.13%)	2843 (0.14%)	10203 (0.13%)
vlc_phy_rs_encoder_64_32	1294 (0.65%)	34792 (0.60%)	11748 (0.56%)	46540 (0.59%)
vlc_phy_rs_encoder_160_128	1329 (0.67%)	34851 (0.60%)	12130 (0.58%)	46981 (0.59%)
vlc_phy_fec_interleaver	40167 (20.33%)	1235542 (21.22%)	430923 (20.61%)	1666465 (21.06%)
vlc_phy_convolutional_codec	91 (0.05%)	2022 (0.03%)	772 (0.04%)	2794 (0.04%)
vlc_phy_rs_decoder_15_11	1685 (0.85%)	36794 (0.63%)	17194 (0.82%)	53987 (0.68%)
vlc_phy_rs_decoder_15_7	2161 (1.09%)	49497 (0.85%)	21604 (1.03%)	71102 (0.90%)
vlc_phy_rs_decoder_15_4	2492 (1.26%)	58225 (1.00%)	24700 (1.12%)	82926 (1.05%)
vlc_phy_rs_decoder_15_2	2760 (1.40%)	64842 (1.11%)	27039 (1.29%)	91881 (1.16%)
vlc_phy_rs_decoder_64_32	23648 (11.97%)	535269 (9.19%)	225575 (10.79%)	760844 (9.61%)
vlc_phy_rs_decoder_160_128	30809 (15.59%)	703750 (12.09%)	306422 (14.65%)	1010173 (12.76%)
vlc_phy_fec_deinterleaver	36718 (18.59%)	1221978 (20.99%)	401908 (19.22%)	1623886 (20.52%)
vlc_phy_viterbi	52724 (26.69%)	1809668 (31.08%)	592415 (28.33%)	2402089 (30.35%)
others	607 (0.31%)	12447 (0.21%)	9083 (0.43%)	21523 (0.27%)
vlc_phy_fec(total)	197560 (100.00%)	5822843 (100.00%)	2091040 (100.00%)	7913883 (100.00%)

Table 11 – Memory utilization percentage for the FEC IP in 5CGXFC5C6F27C7N and TCB018GBWP7T

Instance	5CGXFC5C6F27C7N	TCB018GBWP7T
vlc_phy_rs_encoder_15_11	0.00%	0.00%
vlc_phy_rs_encoder_15_7	0.00%	0.00%
vlc_phy_rs_encoder_15_4	0.00%	0.00%
vlc_phy_rs_encoder_15_2	0.00%	0.00%
vlc_phy_rs_encoder_64_32	0.00%	0.00%
vlc_phy_rs_encoder_160_128	0.00%	0.00%
vlc_phy_fec_interleaver	99.51%	99.35%
vlc_phy_convolutional_codec	0.00%	0.00%
vlc_phy_rs_decoder_15_11	43.34%	45.00%
vlc_phy_rs_decoder_15_7	34.15%	34.23%
vlc_phy_rs_decoder_15_4	31.55%	29.35%
vlc_phy_rs_decoder_15_2	27.60%	26.49%
vlc_phy_rs_decoder_64_32	26.37%	21.81%
vlc_phy_rs_decoder_160_128	46.48%	40.93%
vlc_phy_fec_deinterleaver	99.46%	99.19%
vlc_phy_viterbi	86.29%	83.32%
others	0.00%	0.00%
vlc_phy_fec (total)	83.36%	75.00%

5.3.3 Timing Analysis

Every digital circuit has timing constrains that defines its maximum operating frequency (F_{max}), and they are mostly affected by the setup and hold times, the amount of time a signal at the synchronous input pin must be stable before and after the active edge of clock, and the cell delays. Static Timing Analysis (STA) tools validate all possible design paths to report timing violations and worst-case scenarios. The maximum circuit delay is determined by the longest sensitizable - activable by primary inputs - combinational path that results in the worst setup or hold time case, known as critical path. Such aspect is fundamental to determine latency and throughput of the application devised by the implemented RTL design.

The critical path of the FEC IP has been analyzed for both FPGA (5CGXFC5C6F -27C7N) and ASIC (TCB018GBWP7T) synthesis flows, and both pointed to the same path in RS(160,128) already confirmed by [42]: the combinational data path that integrates the lower to the upper logic portions of the Berlekamp-Massey unit, used as the key equation solver of the RS decoding mechanism. 80 Mhz and 250 Mhz were the F_{max} obtained for the FEC IP by FPGA and ASIC synthesis tools respectively, and the performance superiority showed by the ASIC flow is expected due to the customizable characteristics of its device technology. Fig. 51 shows the critical path for 5CGXFC5C6F27C7N using the post-fitting "Technology Map Viewer". Each cell - mapped in ALMs - has the delay inside itself and the interconnect (wire) delay, and the critical is determined by summing up all delays together. For TCB018GBWP7T, the section C.5 of Annex C shows the timing report for the critical path found in the FEC IP. The critical path has 24 different cells with 54 instances, and most of them are used to implement the adders and full multipliers required by the Berlekamp-Massey unit.

Figure 51 – Critical path of the FEC IP for 5CGXFC5C6F27C7N reported by Quartus® Technology Map Viewer.

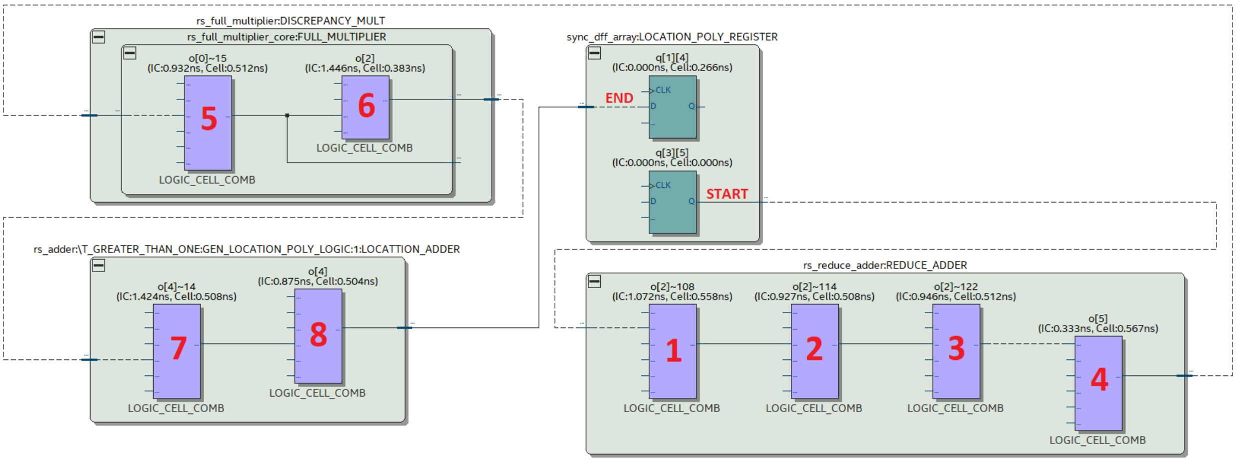

rs_full_multiplier:DISCREPANCY_MULT
rs_full_multiplier_core:FULL_MULTIPLIER
o[0]~15
(IC:0.932ns, Cell:0.512ns)
5
LOGIC_CELL_COMB
o[2]
(IC:1.446ns, Cell:0.383ns)
6
LOGIC_CELL_COMB
sync_dff_array:LOCATION_POLY_REGISTER
q[1][4]
(IC:0.000ns, Cell:0.266ns)
END
CLK
D
Q
q[3][5]
(IC:0.000ns, Cell:0.000ns)
CLK
D
Q
START
rs_adder:\T_GREATER_THAN_ONE:GEN_LOCATION_POLY_LOGIC:1:LOCATTION_ADDER
o[4]~14
(IC:1.424ns, Cell:0.508ns)
7
LOGIC_CELL_COMB
o[4]
(IC:0.875ns, Cell:0.504ns)
8
LOGIC_CELL_COMB
rs_reduce_adder:REDUCE_ADDER
o[2]~108
(IC:1.072ns, Cell:0.558ns)
1
LOGIC_CELL_COMB
o[2]~114
(IC:0.927ns, Cell:0.508ns)
2
LOGIC_CELL_COMB
o[2]~122
(IC:0.946ns, Cell:0.512ns)
3
LOGIC_CELL_COMB
o[5]
(IC:0.333ns, Cell:0.567ns)
4
LOGIC_CELL_COMB

Table 12 – 12 topmost critical path for the FEC IP in 5CGXFC5C6F27C7N

End Point Instance	Location	Delay (ns)	F_{max} (MHz)
vlc_phy_rs_decoder_160_128	Key equation solver in BM	11.75	85.1
vlc_phy_rs_decoder_64_32	Key equation solver in BM	11.60	86.2
vlc_phy_fec_deinterleaver	Read enable logic+SRAM indexing mechanism	11.11	90.0
vlc_phy_fec_interleaver	SRAM indexing mechanism	10.74	93.0
vlc_phy_viterbi	SRAM indexing mechanism	10.30	97.0
vlc_phy_viterbi	ACS unit	8.83	113.2
vlc_phy_fec_deinterleaver	SRAM indexing mechanism	8.83	113.2
vlc_phy_fec_deinterleaver	2D address generator	8.81	113.5
vlc_phy_fec_interleaver	2D address generator	8.81	113.5
vlc_phy_rs_decoder_64_32	MCS ID decoding + RS syndrome data path	8.64	115.7
vlc_phy_rs_decoder_15_7	Key equation solver in BM	8.25	121.2
vlc_phy_rs_decoder_15_2	Key equation solver in BM	8.22	121.6

Table 12 and 13 lists the 12 topmost critical paths in FEC IP for 5CGXFC5C6F27C7N and TCB018GBWP7T. They only inform the worst delay for a given path, and its other structurally equivalent paths with lower delays are omitted. For 5CGXFC5C6F27C7N, the five first positions are related to the worst critical path and memory access data paths for reading in the Interleaver, Deinterleaver, and Viterbi blocks. After these paths there is a timing gap occupied by path replications of the 5 topmost positions with lower delay values - the critical path has wide signals and the synthesis tools process a bit wise timing analysis. After these repetitions, a critical path inside Add-Compare-Select (ACS) Unit from Viterbi IP is reached. The remaining paths are related to the data path processing units of RS codec and (De)Interleaver, with the exception of the path 10th, which is a merge between the *VLC_PHY_FEC_CTRL* and the RS_decoder. As expected, most paths are related to decoding blocks, which are more complex than the encoding related blocks indeed.

For TCB018GBWP7T, most paths are in common to 5CGXFC5C6F27C7Nl; however, they are in a different order since the 9 first path are related to core data path processing units of the base blocks - with the same exceptions related to the merging with *VLC_PHY_FEC_CTRL*. Then, memory related paths occupies the bottommost positions. Also, it is worth to highlight that the delta timing between the paths is lower in comparison to TCB018GBWP7T. The reason for that is related to the fact that two connected ALMs might not be as close to each other as is could be due to lack of resources, whereas such situation does not occur for TCB018GBWP7T, which is able to conceive an optimized placement for the gate level design of the FEC IP. Since most paths are related to the base IPs, the architecture of each one [42, 46, 51] must be investigated in order to find out possible timing optimization. The only timing enhancement for the FEC IP level is related to the decoding mechanism of the MCS ID, which could have another register right after the *VLC_PHY_MCS_ID_DEC* to exclude such entity from any possible critical path.

Table 13 – 12 topmost critical path for the FEC IP in TCB018GBWP7T

End Point Instance	Location	Delay (ns)	F_{max} (MHz)
vlc_phy_rs_decoder_160_128	Key equation solver in BM	3.93	254.8
vlc_phy_rs_decoder_64_32	Key equation solver in BM	3.93	254.8
vlc_phy_rs_decoder_15_2	MCS ID decoding + RS syndrome data path	3.59	278.6
vlc_phy_viterbi	Branch Distance Unit + ACS	3.57	280.1
vlc_phy_rs_decoder_15_11	Busy status control + RS syndrome data path	3.55	280.5
vlc_phy_rs_decoder_15_11	MCS ID decoding + RS syndrome data path	3.55	280.5
vlc_phy_rs_decoder_15_4	Key equation solver in BM	3.55	281.5
vlc_phy_convolutional_codec	MCS ID decoding + CC data path	3.52	284.2
vlc_phy_rs_decoder_15_7	Key equation solver in BM	3.49	286.3
vlc_phy_fec_deinterleaver	Read enable logic+SRAM indexing mechanism	3.48	287.0
vlc_phy_viterbi	SRAM indexing mechanism	3.48	287.1
vlc_phy_fec_interleaver	Read enable logic+SRAM indexing mechanism	3.47	288.0

5.3.4 Power Consumption

Power consumption is a critical design consideration for communication applications targeted to IoT devices - such as VLC systems. The total power consumption of a IC device is the sum of the following components [62]: Dynamic Power (P_{dy}), the power dissipated due to signal switching activity (eq. 5.1); Short-Circuit Power (P_{sh}), the power dissipated by the path between the supply and the ground during the switching of a CMOS gate (eq. 5.2); and the Leakage Power (P_{lek}), power dissipated by the leakage currents between source and drain (subthreshold), the diffusion layers and the substrate (reverse), and the channel and the gate. The sum of all these components is the total power dissipation (P_{total}) in CMOS digital circuits (eq. 5.3).

$$P_{dy} = \alpha_{\Gamma} C_{load} V_{DD}^2 F_{clk} \tag{5.1}$$

where:

α_{Γ} is the Node Transition Factor

C_{load} is the Output Node Load Capacitance

V_{DD} is the Supply Voltage

F_{clk} is the Clock Frequency

$$P_{sc} = \frac{1}{12} k \tau F_{clk} (V_{DD} - 2V_T)^3 \tag{5.2}$$

where:

k is the Process Transconductance Parameter

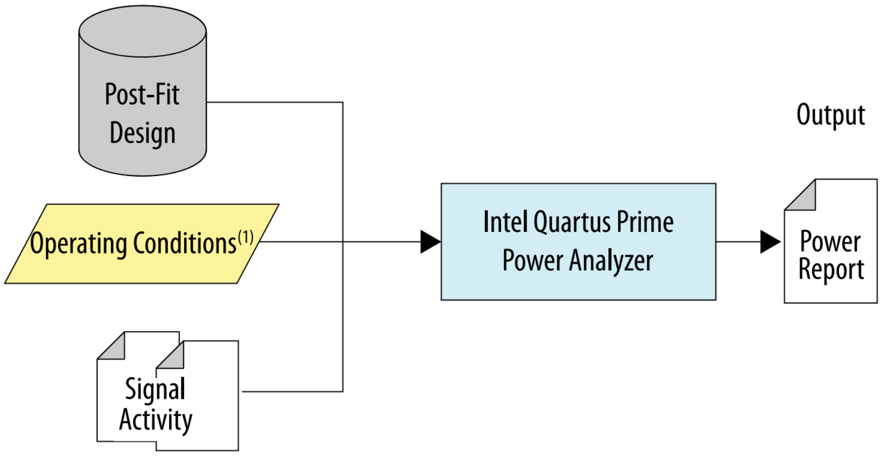
τ is the Short Circuit Duration

V_T is the Threshold N-type Metal-Oxide-Semiconductor (NMOS) or P-type Metal-Oxide-Semiconductor (PMOS) Voltage

$$P_{total} = \alpha_\Gamma C_{load} V_{DD}^2 F_{clk} + \frac{1}{12} k\tau F_{clk} (V_{DD} - 2V_T)^3 + \upsilon_{DD} I_{leakage} \tag{5.3}$$

Quartus ® Prime offers the Power Analyzer tool [63] that informs power consumption in a FPGA device based on the inputs highlighted in Fig. 52. The Post-Fit Design is the factual implementation of the input RTL design on the target FPGA device; the operating conditions are a set of attribute values for voltage and temperature which is specific to the FPGA device; and the signal activity informs the toggling pattern of the the design nets and pins, using a default toggle rate or providing a input signal activity file (e.g Value Change Dump (VCD) format). The Synopsis Design Constrains (SDC) file generated during Timing Analysis step is also considered as input for the Power Analyzer tool. Its reports provide dynamic and short circuit power dissipated for routing and logic implementation (P_{dy+sc}), leakage power (P_{lek}), and also the power dissipated by I/O pins $P_{I/O}$. The total power (P_{total}) is the sum of all these components.

Figure 52 – Inputs required for Power Analyzer tool in Quartus ® Prime [63]



(1)Operating condition specifications are available for only some device families

Genus™ has the "report_power" command, which provides the power estimates. It takes almost the same inputs as required by Quartus® Prime. The only difference is that the operating conditions is provided by the "lib." file which contains the characterization of the standard liberty cell used to implement the RTL. Its reports includes leakage power (P_{lek}) and dynamic power - sum between the "Internal Power" and "Net Power" - estimates. The component "Internal Power" (P_{int}) is defined as the power dissipation of the cell, which are affected by short circuit events and switching activities, whereas "Net Power" (P_{net}) is characterized by the power dissipation due to external wires and pins driven by the circuit cells [64].

In order to understand the overall power consumption profile of the FEC IP, power estimates have been collected for 5CGXFC5C6F27C7N and TCB018GBWP7T by varying the average toggle rate of the circuit signals. Logic intensive digital designs have an average toggle rate around 12.5%, and it usually does not exceed 20% [65]. Then, a range between 2% and 20% for average toggle rate has been used to assess power dissipation on the FEC IP. The clock frequency used in this experiment is the F_{max} reported by section 5.3.3 (80 MHz for 5CGXFC5C6F27C7N and 254 MHz for TCB018GBWP7T). Also, V_{DD} is 1.1 V for 5CGXFC5C6F27C7N and 1.8 V for TCB018GBWP7T.

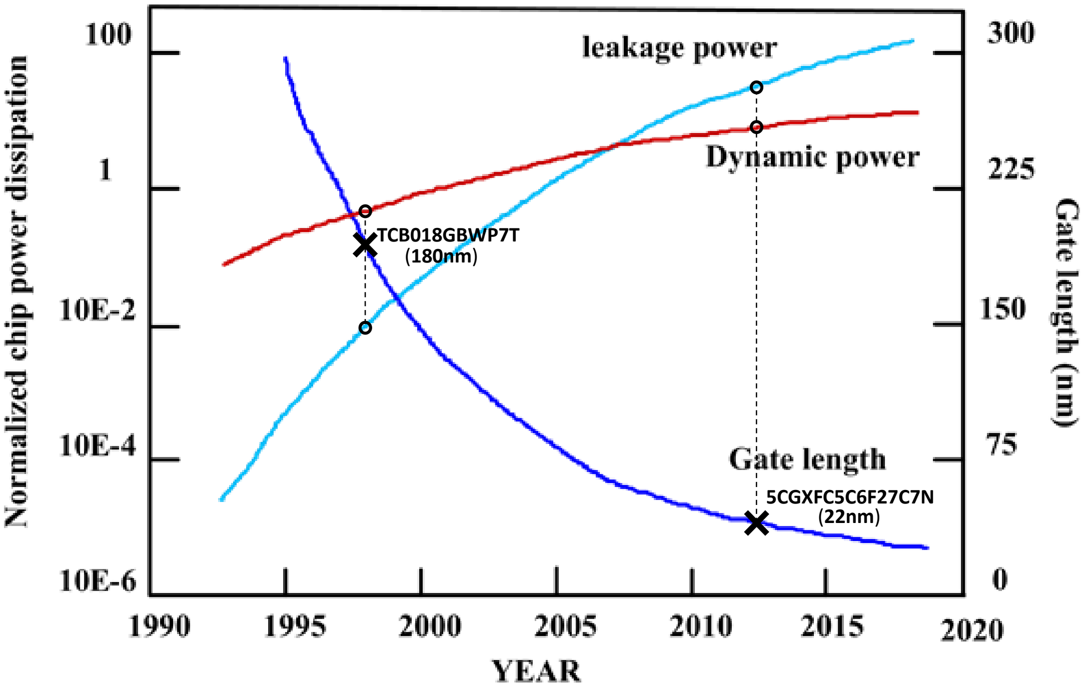
Quartus® Prime and Genus™ provide different ways measure power consumption by varying the toggle rate. In Quartus® Prime, there is a field in Power Analyze GUI to inform the default toggle rate percentage for ether input I/O signals or all signals, which is relative to the clock domain which governs a given logic node. Toggle rates have been defined for all signals, and optimizations related to signal value propagation (vectorless estimation) was turned off since it demonstrated to be inaccurate when using only the default toggle rate as input. Power Analyzer settings have not been changed for the experiments - ‘Board thermal model’ is ‘CONSERVATIVE’ and ‘Device Power Characteristics’ is ‘TYPICAL’.

In Genus™, it is required to set up the attributes *lp_asserted_toggle_rate*, which defines the toggle count in terms of a time unit, and *lp_asserted_probability*, which specifies the static probability of a signal being high. The former was set to the clock period established by F_{max}, and the later was measured for the ranges between 1% and 10% and 99% and 90%. Note that both cases end up with an average toggle rate between 2% and 20%. However, not all signals are assigned to those probability because Genus™ propagates the for combinational logic according to its truth table, then overall signal activity might differ according to the most likely logic value (high or low). Therefore, power consumption results might be different for those two ranges, and that is the reason it was measured for both cases. The value propagation analysis could be turned off, but it was kept enabled because it provided more accurate results when they are compared to the ones obtained by the experiment using VCD activity - described at the end of this section.

Table 14 shows the results for 5CGXFC5C6F27C7N, and P_{lek} is around 350 mW and P_{dy+sc} has a delta of $\sim 700mW$ for the reference toggle rate range. Also, P_{lek} plays a significant role in the overall power consumption. On the other hand, as depicted by Table 15, P_{lek} is almost neglectable for TCB018GBWP7T (0.02 mW), as P_{int} and P_{net} dominates the total power consumption. These different power profiling characteristics for the analyzed device technologies are in agreement with the power trend for technology nodes as depicted by Fig. 53 - P_{lek} only becomes an important component below 75 nm. In both cases, there is a high initial dynamic power consumption even though the signal activity is low. It indicates that the clock tree of the FEC IP is reason for such finding because the signal "clk" has a static toggle rate of 50%. Another aspect related to TCB018GBWP7T results is that static probability for the range between 99% and 90% - stuck at logic "1" - consumes less power than the range between 1% and 10% - stuck at logic "0".

As F_{clk} assumes different values for 5CGXFC5C6F27C7N and TCB018GBWP7T, their results must be normalized to each other in order to make them comparable - and there is an additional column in Table 15 that. After the normalization, power consumption results for TCB018GBWP7T are closer to the ones obtained for 5CGXFC5C6F27C7N, but they are still higher. However, it is worth to highlight that the different power supplies values are used for 5CGXFC5C6F27C7N and TCB018GBWP7T (1.1 V versus 1.9 V), and V_{DD} is a quadratic factor in eq. 5.1.

Figure 53 – Power dissipation for dynamic and leakage rates according to the technology node [66]



Using default probability attributes to model switching activity in a digital circuit to measure power consumption is very inaccurate and might be far away from a real case. In other to obtain more precise values for power consumption a typical waveform in VCD format has been obtained for all IEEE 802.15.7 operation modes covered by the FEC IP behaving as either encoder or decoder. To do so, a simulation using the test bench proposed in section 5.2.3 was carried out for the gate level design version of the FEC IP provided by the synthesis tools. The frame length used was 63B. Tables 17, 18, 19, and 20

Table 14 – Power consumption (in mW) for 5CGXFC5C6F27C7N by varying the default toggle rate of all FEC IP signals

Toggle Rate (%)	P_{dy+sc}	P_{lek}	$P_{I/O}$	P_{total}
0	319.79	351.52	7.12	678.43
2	390.28	351.99	8.36	750.63
4	460.78	352.46	9.60	822.84
6	531.27	352.94	10.83	895.05
8	601.77	353.43	12.07	967.27
10	672.26	353.92	13.31	1039.49
12	742.76	354.42	14.55	1111.72
14	813.25	354.92	15.78	1183.95
16	883.75	355.42	17.02	1256.19
18	954.24	355.93	18.26	1328.43
20	1024.74	356.44	19.50	1400.68

Table 15 – Power consumption (in mW) for TCB018GBWP7T by varying the static probability of FEC IP signals

Static Probability (%)	Related Toggle Rate (%)	P_{lek}	P_{dy}	P_{sc}	P_{total}	P_{total_norm}
1	2	0.02	3139.87	632.49	3772.36	1188.15
2	4	0.02	3152.59	637.26	3789.85	1193.65
3	6	0.02	3170.29	643.94	3814.23	1201.33
4	8	0.02	3182.61	648.92	3831.53	1206.78
5	10	0.02	3202.86	655.84	3858.70	1215.34
6	12	0.02	3216.38	659.25	3875.63	1220.67
7	14	0.02	3240.34	670.68	3911.02	1231.82
8	16	0.02	3289.58	689.43	3979.01	1253.23
9	18	0.02	3333.87	707.20	4041.07	1272.78
10	20	0.02	3361.37	722.05	4083.41	1286.11
90	20	0.02	2907.39	521.43	3428.84	1079.95
91	18	0.02	2856.22	506.24	3362.48	1059.05
92	16	0.02	2836.06	496.37	3332.45	1049.59
93	14	0.02	2817.43	488.56	3306.00	1041.26
94	12	0.02	2788.15	480.05	3268.22	1029.36
95	10	0.02	2784.09	476.95	3261.06	1027.10
96	8	0.02	2754.19	466.08	3220.29	1014.26
97	6	0.02	2745.80	461.58	3207.40	1010.20
98	4	0.02	2751.77	463.05	3214.84	1012.55
99	2	0.02	2728.41	455.16	3183.59	1002.71

Table 16 – Group IDs and their related MCS IDs and waveform length for the original frame of 63B (in cycles)

Group ID	MCS ID	Encoder Trace Length	Decoder Trace Length
G_1	{0}	694	2415
G_2	{1}	454	1271
G_3	{2}	454	740
G_4	{3}	96	93
G_5	{4, 8, 20, 29, 37, 38}	71	66
G_6	{5}	488	1017
G_7	{6}	248	298
G_8	{7}	144	150
G_9	{16, 18, 21, 23, 25, 28, 32, 33, 34, 35, 36}	136	379
G_10	{17, 19, 22, 24, 26, 28}	104	410

refer to the results obtained for the dynamic power consumption at the target technology devices for the main FEC IP sub-blocks functioning as either encoder or decoder. The columns labeled as Groups (G) ID represent the operating modes that requires the same set of base sub-blocks as displayed by table 16, with the required number of cycles for the encoding and decoding phase for the original frame of 63B.

The first consideration about the obtained results is that the total power consumption of the FEC IP is below the value collected in Tables 14 and 15 for their minimum default toggle rate (2%). It means that the switching activity for a typical waveform is reduced in comparison to the overall toggle rate potential present in FEC IP. This is expected for a complex digital design with many sub-blocks. First of all, signal belonging to control interface are usually constant for most of the time. Also, data path signals usually do not switch all at same time even including certain level of architectural pipelining. Moreover, there are sub-blocks - with their power consumption results 'greyed out' - that are not required for a given operating mode.

Another aspect that validates even the functional behavior of the FEC IP is that the power consumption results for the blocks required by the groups (in 'bold') has an expected slope in comparison to the cases that the same blocks are not used by other groups. However, there are some outliers that identify a possible flaw in the proposed architecture in terms of power consumption. The Interleaver, for instance, actually consumes more power when it is not used and some RS Encoder is required - G5 does not report such behavior. It is consistent for both 5CGXFC5C6F27C7N (Table 17) and TCB018GBWP7T (Table 19) results. It turns out that the validity signal for the Interleaver is accordingly multiplexed to disallow the block operation when it is not required, but output data of RS Encoder continues to drive the Interleaver since there is not any functional side effect as validity bit is disable. However, as observer in power consumption results, it causes an undesired power consumption into the SRAM memory that the Interleaver holds. Moreover, it is even greater in G4, G6, G7, G8, G9, G10 because the Interleaver data input density in their waveforms are higher since RS Encoder is always driving data, as opposed to G1, G2, and G3 which has an additional step for the Convolutional Codec where RS Encoder is idle. Another fact that indicates that such analysis is correct is that the De-interleaver does not present the same power consumption anomaly because the block that drives data to it - Viterbi - is never enabled when the De-interlever is disabled.

The results also demonstrate two points that are in conflict between the power consumption profiles for 5CGXFC5C6F27C7N and TCB018GBWP7T. The first one is related to G5. Surprisingly, it has one of the worst power consumption for encoding and decoding modes in 5CGXFC5C6F27C7N (Tables 17 and 18), and it does not require any error correction block. For 5CGXFC5C6F27C7N there is a considerably increase for 'others', whereas it is not identified in TCB018GBWP7T. As 5CGXFC5C6F27C7N is constrained by resources, the routing path between the input and the output might include many ALMs that becomes a hot spot for G5, as it only drives the inputs to the outputs. However, if power consumption is normalized by the data throughput, G5 will certainly have one of the lowest values for it. The second one is the unexpected power consumption for Viterbi in 5CGXFC5C6F27C7N (Table 18) for G4 to G10, which does not happen for TCB018GBWP7T. The reason for such increase is probably the same as reported for the interleaver, as viterbi is the data input signal for the decoding mode (i_data_dec). However, it requires more investigation to understand why the same behavior was not observed in TCB018GBWP7T power consumption results.

This analysis of dynamic power consumption for sub-blocks that are not used by a given Group ID could be avoided if all control and data signals were included together with the validity data signal. Then, clock signal is another important signal that causes dynamic power consumption in registers. There are many low power techniques for clock gating described by the literature [67] that aim to restrict the clock ticks to save power and synthesis tools, such as Quartus ® Prime and Genus ®, already provide automated flow for clock gating insertion. Such approach is appropriate for technologies such as TCB018GBWP7T, as it only focus on dynamic power, which is dominant in 180 nm node. For lower technology nodes, leakage power becomes a major component, and other techniques are used to reduce it such as the power shutoff. It basically disconnects an unused block from the supply source using power switches. In order to enable it, the power intent model for the RTL design must be defined, usually specified by the Unified Power Format (UPF) format [68].

Table 17 – Dynamic power consumption (in mW) for 5CGXFC5C6F27C7N based on a typical VCD activity of each operating mode group using vlc_phy_fec as encoder.

Instance	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
rs_encoder_15_11	0.16	**0.23**	**0.23**	**0.52**	0.18	0.17	0.17	0.18	0.17	0.17
rs_encoder_15_7	**0.28**	0.21	0.21	0.24	0.24	0.21	0.22	**0.63**	0.22	0.23
rs_encoder_15_4	0.24	0.24	0.24	0.26	0.25	0.25	**0.74**	0.25	0.25	0.25
rs_encoder_15_2	0.25	0.25	0.25	0.27	0.27	**0.77**	0.26	0.26	0.26	0.26
rs_encoder_64_32	0.82	0.82	0.82	0.87	0.88	0.82	0.83	0.85	**3.88**	0.86
rs_encoder_160_128	0.80	0.80	0.80	0.85	0.86	0.81	0.82	0.83	0.83	**4.28**
interleaver	**48.03**	**47.40**	**47.40**	63.83	32.49	67.41	64.13	64.60	64.14	63.72
convolutional_codec	**0.43**	**0.42**	**0.44**	0.05	0.05	0.05	0.05	0.05	0.05	0.05
rs_decoder_15_11	0.99	1.00	1.00	1.25	1.11	2.15	1.70	1.32	1.33	1.31
rs_decoder_15_7	1.32	1.32	1.32	1.46	1.43	1.73	1.57	1.52	1.47	1.48
rs_decoder_15_4	1.41	1.42	1.42	1.59	1.52	2.03	1.91	1.61	1.62	1.61
rs_decoder_15_2	1.61	1.61	1.61	1.79	1.72	2.37	2.00	1.81	1.82	1.81
rs_decoder_64_32	7.36	7.39	7.39	8.01	7.87	9.32	8.56	8.03	8.65	8.07
rs_decoder_160_128	10.10	10.18	10.18	11.26	11.29	12.24	11.52	11.11	11.15	11.57
deinterleaver	32.70	32.70	32.70	32.69	32.71	32.70	32.71	32.69	32.65	32.74
viterbi	31.57	31.96	32.34	32.29	32.78	31.55	31.47	31.83	31.85	32.23
other	**79.54**	**77.69**	**76.87**	**78.57**	**126.00**	**78.89**	**78.58**	**78.48**	**81.24**	**81.25**
vlc_phy_fec (total)	217.61	215.64	215.22	235.80	251.65	243.47	237.24	236.05	241.58	241.89
vlc_phy_fec (required)	**128.28**	**125.74**	**124.94**	**79.09**	**126.00**	**79.66**	**79.32**	**79.11**	**85.12**	**85.53**

Table 18 – Dynamic power consumption (in mW) for 5CGXFC5C6F27C7N based on a typical VCD activity of each operating mode group using vlc_phy_fec as decoder.

Instance	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
rs_encoder_15_11	0.16	0.16	0.15	0.17	0.18	0.16	0.16	0.16	0.16	0.16
rs_encoder_15_7	0.20	0.20	0.19	0.22	0.25	0.20	0.21	0.21	0.20	0.20
rs_encoder_15_4	0.24	0.24	0.23	0.24	0.25	0.24	0.24	0.24	0.24	0.24
rs_encoder_15_2	0.24	0.24	0.24	0.26	0.27	0.25	0.25	0.25	0.25	0.25
rs_encoder_64_32	0.81	0.81	0.78	0.85	0.88	0.81	0.82	0.84	0.81	0.82
rs_encoder_160_128	0.79	0.79	0.77	0.83	0.86	0.80	0.81	0.82	0.80	0.80
interleaver	31.47	31.48	30.44	31.51	31.36	31.46	31.48	31.52	31.49	31.47
convolutional_codec	0.05	0.05	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05
rs_decoder_15_11	0.98	**0.98**	**0.95**	**4.75**	0.98	1.82	2.30	2.54	1.56	1.38
rs_decoder_15_7	**1.31**	1.31	1.26	1.90	1.30	1.60	1.78	**3.68**	1.51	1.45
rs_decoder_15_4	1.40	1.40	1.35	2.30	1.39	1.85	**4.05**	2.26	1.71	1.62
rs_decoder_15_2	1.59	1.59	1.54	2.55	1.59	**3.93**	2.37	2.51	1.93	1.83
rs_decoder_64_32	7.30	7.30	7.06	10.11	7.27	8.72	9.56	9.98	**11.53**	8.52
rs_decoder_160_128	9.97	9.97	9.64	13.01	9.93	11.49	12.40	12.86	12.16	**15.04**
deinterleaver	**32.69**	**32.69**	**31.62**	32.73	32.57	32.68	32.70	32.74	32.71	32.69
viterbi	**30.90**	**30.91**	**29.94**	44.92	44.88	42.59	41.56	43.76	35.77	34.38
other	**71.90**	**71.90**	**69.63**	**76.95**	**87.09**	**75.54**	**74.51**	**75.76**	**73.37**	**73.97**
vlc_phy_fec (total)	192.00	192.02	185.83	223.35	221.10	214.19	215.25	220.18	206.25	204.87
vlc_phy_fec (required)	**136.80**	**136.48**	**132.14**	**81.70**	**87.09**	**79.47**	**78.56**	**79.44**	**84.90**	**89.01**

Table 19 – Dynamic power consumption (in mW) for TCB018GBWP7T based on a typical VCD activity of each operating mode group using vlc_phy_fec as encoder.

Instance	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
rs_encoder_15_11	1.77	**1.79**	**1.79**	**2.19**	1.78	1.77	1.77	1.78	1.80	1.79
rs_encoder_15_7	**2.55**	2.38	2.38	2.39	2.43	2.37	2.38	**2.96**	2.41	2.40
rs_encoder_15_4	2.71	2.71	2.71	2.72	2.73	2.68	**3.41**	2.70	2.71	2.71
rs_encoder_15_2	3.08	3.09	3.09	3.10	3.11	**3.70**	3.08	3.07	3.07	3.09
rs_encoder_64_32	11.56	11.57	11.57	11.57	11.61	11.54	11.55	11.55	**11.23**	11.59
rs_encoder_160_128	11.32	11.32	11.32	11.25	11.27	11.19	11.19	11.22	17.78	**18.51**
interleaver	**510.16**	**509.93**	**509.93**	599.49	488.55	613.78	601.04	604.27	602.49	600.28
convolutional_codec	**0.96**	**0.95**	**0.95**	0.58	0.59	0.57	0.58	0.58	0.58	0.58
rs_decoder_15_11	20.36	20.25	20.25	20.05	19.88	20.23	20.22	20.17	20.12	20.08
rs_decoder_15_7	24.75	24.67	24.67	24.46	24.38	24.69	24.64	24.63	24.53	24.50
rs_decoder_15_4	27.58	27.49	27.49	27.34	27.19	27.50	27.44	27.46	27.36	27.32
rs_decoder_15_2	29.74	29.62	29.62	29.48	29.39	29.65	29.62	29.60	29.49	29.51
rs_decoder_64_32	169.78	169.20	169.20	168.13	167.38	169.22	168.97	168.74	167.99	168.08
rs_decoder_160_128	307.00	305.55	305.55	302.25	299.66	305.71	304.90	304.25	302.45	302.06
deinterleaver	483.49	483.49	483.49	484.20	485.14	483.48	483.49	483.49	483.64	484.34
viterbi	727.02	727.04	727.04	729.55	731.04	727.06	727.44	728.17	728.24	729.49
other	**189.21**	**189.21**	**189.28**	**189.71**	**189.64**	**189.67**	**189.68**	**189.59**	**189.72**	**189.84**
vlc_phy_fec (total)	2523.37	2520.59	2520.66	2608.78	2496.07	2625.12	2611.72	2614.56	2615.94	2616.48
vlc_phy_fec (required)	**702.94**	**701.94**	**702.01**	**191.92**	**189.64**	**193.39**	**193.11**	**192.58**	**200.97**	**208.37**

Table 20 – Dynamic power consumption (in mW) for TCB018GBWP7T based on a typical VCD activity of each operating mode group using vlc_phy_fec as decoder.

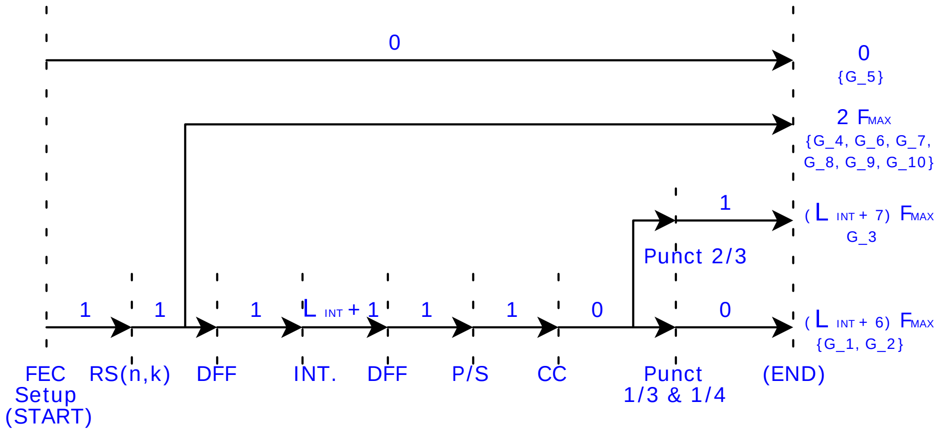
Instance	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
rs_encoder_15_11	1.75	1.78	1.77	1.80	1.83	1.77	1.77	1.78	1.77	1.78
rs_encoder_15_7	2.35	2.36	2.43	2.46	2.46	2.37	2.39	2.37	2.38	2.38
rs_encoder_15_4	2.68	2.68	2.68	2.71	2.75	2.75	2.69	2.70	2.68	2.68
rs_encoder_15_2	3.04	3.04	3.04	3.08	3.15	3.05	3.05	3.07	3.05	3.05
rs_encoder_64_32	11.53	11.52	11.52	11.58	11.66	11.66	11.56	11.58	11.54	11.57
rs_encoder_160_128	11.16	11.17	11.17	11.22	11.29	11.18	11.18	11.20	11.17	11.18
interleaver	483.24	483.24	483.25	486.79	483.25	483.25	484.21	484.22	483.74	483.73
convolutional_codec	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
rs_decoder_15_11	20.41	**20.42**	**20.49**	**22.79**	20.33	20.33	20.69	20.77	20.53	20.49
rs_decoder_15_7	**24.86**	24.83	24.85	25.64	24.79	25.50	25.50	**28.49**	25.10	25.02
rs_decoder_15_4	27.60	27.65	27.69	28.85	27.58	27.58	**32.39**	28.79	28.05	27.90
rs_decoder_15_2	29.76	29.81	29.85	31.24	29.73	**31.95**	30.95	31.16	30.26	30.10
rs_decoder_64_32	169.89	169.97	170.16	174.86	169.80	169.80	173.91	174.64	**177.39**	171.70
rs_decoder_160_128	307.61	307.70	307.88	312.57	307.48	311.64	311.64	312.34	310.37	**314.41**
deinterleaver	**487.23**	**489.55**	**496.90**	482.75	482.75	482.76	482.76	482.76	482.75	482.74
viterbi	**799.84**	**821.08**	**870.62**	749.09	747.39	747.39	745.31	749.80	735.48	733.20
other	**188.92**	**188.90**	**188.94**	**189.42**	**190.01**	**189.07**	**189.14**	**189.30**	**189.08**	**189.08**
vlc_phy_fec (total)	2572.77	2596.59	2654.07	2537.71	2517.14	2523.04	2530.01	2535.88	2516.23	2511.91
vlc_phy_fec (required)	**1500.91**	**1520.01**	**1577.02**	**212.23**	**190.01**	**221.04**	**221.55**	**217.80**	**366.49**	**503.51**

5.4 Communication Performance

IEEE 802.15.7 supports multiple optical clock rates (Tables 56, 57, and 58) that are bound to the frequency at which the data is transmitted by the optical source. Hence, they establish the output data rate required for each operating mode. Also, latency is covered by the data transmission modes (single, packed, and burst) specified for the PHY layer (Fig. 9), and SIFS and RIFS (40 and 120 optical clocks) determine the latency requirements. Then, latency, output data rate, and throughput are calculated in this section in order to determine whether the FEC IP on target technology devices (5CGXFC5C6F27C7N and TCB018GBWP7T) fulfills or not the standard requirements. The documentation related to the sub-blocks used in this master thesis [42, 46, 51] has all required information to determine the referred communication performance metrics.

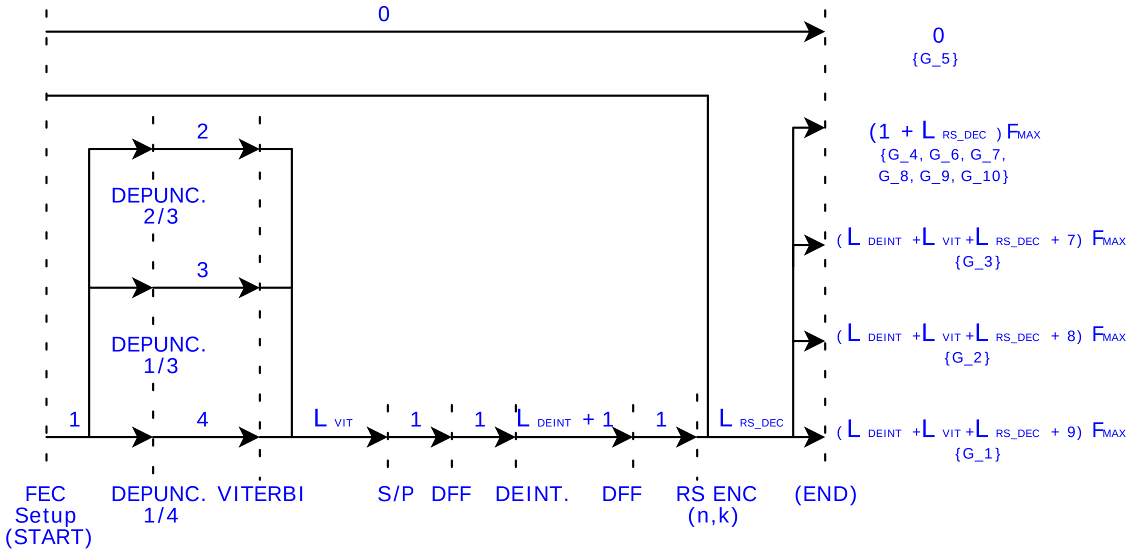
Fig. 54 and 55 are timing diagrams which illustrate the latency required for each Group ID (Table 16). The values over the edge represents the latency required by the source sub-block, indicated by the vertical dotted lines. The latency on the (De)Interleaver is related to the number of cycles required to fill it (L_{INT} and L_{DEINT}), which depends on the size of the original frame after the processing step of the RS Encoder (S_{INT}), described by eq. 5.4. For L_{INT} (eq. 5.6), the latency (in cycles) is the same as S_{INT}, since the output of the RS Encoder is contiguous and matches the the data width size of the Interleaver (4). For L_{DEINT} (eq. 5.7), it is required a S/P (1 to 4) block to convert the data width of the Viterbi output, which is not contiguous as well. It happens because of the depuncturing mechanism might take 4 (G_0), 3 (G_1) or 2 (G_0) cycles to transmit a single input data to the Viterbi block, whereas the latter transmits one output data per cycle. So, the Viterbi throughput has a penalty (eq. 5.5) due to the depuncturing mechanism, which affects L_{DEINT} as well. The latency on the Viterbi block (eq. 5.8) in FEC IP ($VLC\_PHY\_VITERBI$) is the latency reported by its manual [51], normalized by the latency overhead caused by the depuncturing mechanism (K_{DEPUN}). And the latency on the RS Decoder (L_{RS_DEC}) for its common case - input codeword with n symbols - is described by eq. 5.9 [42]. With all these equations at hand, it is possible to calculate the latency of FEC IP for each Group IP with the sum of all its latency nodes multiplied by F_{clk}.

Figure 54 – Timing diagram for the encoder portion of FEC IP



The output data rate of the FEC IP depends on the last sub-block that drives the output data signal. For the encoder portion of the FEC IP, there might be three options: the CC with its puncturing (VLC_PHY_CC) blocks (eq. 5.10), the RS Encoder (eq. 5.11), or directly driven by the input data of FEC IP (eq. 5.12). For the encoder portion of the FEC IP, output data could be driven by RS Decoder (eq. 5.13) or its input data (eq. 5.12). The throughput is basically the obtained output data rate normalized by the amount of redundant data added by the coding techniques. The only data rate equations that take

Figure 55 – Timing diagram for the decoder portion of FEC IP



redundant data into account are eq. 5.10 and 5.11, and their related throughput equations are multiplied by the R_{CC} and $R_{RS_{E}NC}$ to convert them to the original frame size (eq. 5.14 and eq. 5.15). Other equations for output data rate are related to either the decoder side of the FEC IP - which does not have redundant data in their outputs - or the operating modes that bypasses the FEC IP, and throughput equations for them (eq. 5.12 and 5.13) are equivalent to their data rate equations.

$$S_{INT} = \left\lceil \frac{nS_{FRAME}}{k} \right\rceil \tag{5.4}$$

$$P_{DEPUN} = \frac{1}{(2 - K_{DEPUN})} \tag{5.5}$$

where:

S_{INT} is the size of the frame input for the Interleaver block.

S_{FRAME} is the size of the original frame.

k is the number of parity symbols used in the RS Encoder.

n is the number of symbols in the codeword of the RS Encoder.

P_{DEPUN} is the latency penalty due to the puncturing mechanism.

K_{DEPUN} depuncturing penalty factor for latency (G_1-> $K_{DEPUN} = 1/4$,G_2-> $K_{DEPUN} = 1/3$, and G_2-> $K_{DEPUN} = 1/2$).

$$L_{INT} = S_{INT} \tag{5.6}$$

$$L_{DEINT} = P_{DEPUN}\frac{S_{INT}}{4} \tag{5.7}$$

$$L_{VIT\_FEC} = K_{DEPUN}(2(WL + AL) + 6) \tag{5.8}$$

$$L_{RS\_DEC} = n + 3\frac{n-k}{2} + 4 \tag{5.9}$$

where:

L_{INT} is the latency for the Interleaver.

L_{DEINT} is the latency for the Deinterleaver.

L_{VIT_FEC} is the latency for the Viterbi in FEC IP - which includes the depuncturing mechanism.

L_{RS_DEC} is the latency for the RS Decoder.

$$DR_{CC\_FEC} = K_{PUN}W_{FEC\_slice}Fclk \tag{5.10}$$

$$DR_{RS\_ENC} = m * F_{clk} \tag{5.11}$$

$$DR_{NO\_FEC} = W_{FEC\_slice}F_{clk} \tag{5.12}$$

$$DR_{RS\_DEC} = m\frac{n-k}{2^m}F_{clk} \tag{5.13}$$

where:

DR_{CC_FEC} is the data rate output for the CC in FEC IP - which includes the puncturing mechanism.

K_{PUNC} is the puncturing penalty factor for output data rate (G_1-> $K_{PUNC} = 1$ and G_2-> $K_{PUNC} = 1$, G_2-> $K_{PUNC} = 1/2$).

W_{FEC_slice} is the useful slice of the data output pin (described by Fig. 21).

DR_{RS_ENC} is the data rate output for the RS Encoder.

m is the size of the GF field for the RS Encoder.

DR_{NO_FEC} is the data rate output when FEC IP is bypassed in both encoder and decoder portions (G_5).

DR_{RS_DEC} is the data rate output for the RS Decoder.

$$TR_{CC\_FEC} = R_{CC} R_{RS\_ENC} DR_{CC\_FEC} \tag{5.14}$$

$$TR_{RS\_ENC} = R_{RS\_ENC} DR_{RS\_ENC} \tag{5.15}$$

$$TR_{NO\_FEC} = DR_{NO\_FEC} \tag{5.16}$$

$$TR_{RS\_DEC} = DR_{RS\_DEC} \tag{5.17}$$

where:

TR_{CC_FEC} is the data rate output for the CC in FEC IP - which includes the puncturing mechanism.

R_{CC} is the redundant data deduction factor for CC (G_1-> $R_{CC} = 1/4$ and G_2-> $R_{CC} = 1/3$, G_2-> $R_{CC} = 2/3$).

R_{RS_ENC} is the redundant data deduction factor for RS Encoder ($R_{CC} = k/n$)

RR_{RS_ENC} is the throughput for the RS Encoder.

TR_{NO_FEC} is the throughput when FEC IP is bypassed in both encoder and decoder portions (G_5).

TR_{RS_DEC} is the throughput for the RS Decoder.

Annex D displays detailed results for latency, data rate and throughput of all IEEE 802.15.7 operating modes for encoder and decoding portions considering the F_{max} obtained for 5CGXFC5C6F27C7N ($80MHz$) and TCB018GBWP7T ($254MHz$). The requirements for data rate and latency for the MCS IDs were also calculated based on the their optical clocks and the maximum allowed SIFS (120) and RIFS (40). Latency was calculated for the worst case where the size of the original frame has the highest possible value (1023B). Results demonstrate that the FEC IP is compliant with all operating modes for PHY I, and it was expected due to the low optical clock frequencies specified for PHY I. However,

Table 21 – Latency violations for TCB018GBWP7T (ASIC)

MCS_ID	Opt. Clock (Mhz)/ Data Rate (Mbps)	SIFS Maximum Latency (us)	RIFS Maximum Latency (us)	Latency Decoder (us) ASIC
26	60	2.00	0.67	0.84
27	60	1.00	0.33	0.46
28	60	1.00	0.33	0.84

Table 22 – Latency violations for 5CGXFC5C6F27C7N (FPGA)

MCS_ID	Opt. Clock (Mhz)/ Data Rate (Mbps)	SIFS Maximum Latency (us)	RIFS Maximum Latency (us)	Latency Decoder (us) FPGA
23	30	4.00	1.33	1.46
24	30	4.00	1.33	2.66
25	60	2.00	0.67	1.46
26	60	2.00	0.67	2.66
27	120	1.00	0.33	1.46
28	120	1.00	0.33	2.66

it is worth to state that latency for the MCS IDs 0, 1, and 2 are relatively high ($> 10\mu s$ up to $\sim 125\mu s$) because of the Interleaver that requires to receive the complete frame before starting transmitting data due to its inherent functional characteristics.

Nonetheless, latency requirements for some PHY II operating modes could not be fulfilled. TCB018GBWP7T met all SIFS latency requirements; however, its decoder latency values is over the ones specified for RIFS latency in the MCS IDs 26, 27, and 28 (Table 21). 5CGXFC5C6F27C7N violates latency constrains for the MCS IDs between 23 and 28 (Table 22), and data rate requirements are not met for MCS IDs 27 and 28 (120 Mbps is required whereas 80 Mbps was the obtained value). It means that 5CGXFC5C6F27C7N cannot use some MCS IDs for single and packed data transmission modes (Fig 9), whereas TCB018GBWP7T has no restriction for it as it is only disallowed for some MCS IDs in the burst data transmission mode. These results clear elucidates that the critical path found in RS Decoder - analyzed by section 5.3.3 - must be addressed in order to overcome those limitations, as all violations are directly related to operating modes that requires such block. Also, it was clear that the customizable characteristics of the ASIC device (TCB018GBWP7T) could make the difference as it covered more requirements than 5CGXFC5C6F27C7N.

5.5 Closing Discussion

The experimental results for design size, timing analysis, power consumption, and communication performance provided all aspects for the characterization of the proposed FEC IP using technology devices for both ASIC (TCB018GBWP7T) and FPGA (5CGXFC5C6F27C7N) flows. The verification effort was enough to assess the core functionality of the block and guarantee the robustness of the collected synthesis results. However, evaluation of design corner case scenarios and coverage analysis for verification sign-off are required prior to any effort to carry out a tapeout process for the FEC IP.

On design size, it was confirmed a high resource utilization on 5CGXFC5C6F27C7N ($\sim$ 85%) and a reasonable amount of area in TCB018GBWP7T (around $7mm^2$), and RAM blocks are responsible for a large portion of it. On timing analysis, the expected critical path in the RS Decoder has been certified, but other closer timing bottlenecks have been found as well. It indicates that analyzing only the worst critical path is not enough, and a more comprehensive timing analysis is required to improve F_{max} in a complex IP. Also, F_{max} obtained for TCB018GBWP7T (254 MHz) was considerably higher than for 5CGXFC5C6F27C7N (80 MHz), which corroborates the expected performance advantages in the ASIC flow even using an older technology node (180 nm). On power consumption, early synthesis results in a RTL design flow is useful to expose some design flaws that do not affect the functionality of the project but impacts power consumption. It also confirmed the need for a power aware version of the FEC IP if it is targeted to IoT applications, which have power consumption constraints. Moreover, it was possible to understand the different profiles for power consumption in the technology nodes adopted by 5CGXFC5C6F27C7N (22 nm) and TCB018GBWP7T (180 nm). Finally, communication performance metrics show the compliance of the FEC IP with PHY I and III. However, it has some limitations - mainly in latency - for PHY II, which is the high data rate PHY mode in IEEE 802.15.7.

The analysis of synthesis results and the verification effort also indicated some possible enhancements for the architectural design of the FEC IP. First of all, it is required to understand the defect found in the Viterbi sub-block in order to fix a functional behavior and also use the correct parametric values for 'MAX_WINDOW_LENGTH' (42 instead of 96), which would result in a reduction of $\sim$ 15% in design size. Power consumption would also be decreased by $\sim$ 11% in 5CGXFC5C6F27C7N and $\sim$ 14% with such parameter change. This bug may affect the F_{max} for 'SRAM indexing mechanism' (Tables 12 and 13); however timing analysis has not been run for '$MAX\_WINDOW\_LENGTH' = 42$ and such critical path is not among the worst cases.

Another aspect is related to operating modes that use the (De)Interleaver (MCS IDs 0, 1 and 2). One of the main limitations that restrict the FEC IP to process only a single frame at a time - increasing the overall latency - is that the (De)Interleaver is not able to receive a new frame when it is in the transmission mode. A naive way to resolve it is to have a twin (De)Interleaver that is able to receive a new frame, when the other block is in transmission block. However, it clearly impacts design size and power consumption, and a more efficient option would be to have (de)interleaving address mechanism that is able to store the new frame in the RAM spaces released by the transmission the current store frame. Fortunately, this change is not required for IEEE 802.15.7 as the current architecture already meets the latency and data rate requirements for the MCS IDs 0, 1 and 2; however, the overall latency for these modes are still high and other applications might required a decrease of it.

As depicted by Table 23, latency requirements for some of the MCS IDs in PHY II are violated. The RS Decoder (RS(64, 32) and RS(160, 128)) is the main component used by these MCS IDs. The main point is that the FEC currently blocks the reception of any data from the next frame until the it becomes idle. However, if the next frame has the same MCS ID as the current one being processed, the RS Decoder would be able to start processing the next frame before finishing the current one. No changes are needed in RS Decoder for that; however, the restriction for multi-frame processing for these cases happens at the FEC IP level, and it requires some architectural changes to enable it (i.e handling of multiple delimiters of last frame input indicator - *i_last_data_**). If it is needed to reconcile multiple frames with different MCS IDs, a more careful analysis must be conducted since they will use data paths with different data rates in the FEC. Then, a handling in the data output of the FEC is needed to maintain the correct order of the input frames.

As RAM memories occupies a considerable potion of the implemented design, an option is to extract the main - and largest - RAM memories from the FEC IP and replace it by a memory interface that is able to access external RAM memories. It might be useful because these external RAM memories will not only be used by the FEC IP but also by other non-related blocks. Another useful feature is to include a parametric configuration to enable the data path instantiation of a single PHY layer - or even specific operating modes. For instance, PHY I and PHY II have conflicting requirements, as the former requires low power consumption and the latter pursues operating modes with high data rate. Then, if the application requires only one of the PHY layer modes, such parametric configuration would be relevant. However, if it is required both PHY layers, it is recommended to split them into two clock domains in order to use different clock frequencies. For this approach, data synchronization between the domain crossing of multiple clocks are required to overcome metastability issues [69].

Regarding the adopted multi-width data input and output, which depends on the chosen operating mode (Fig. 21), it showed that serializing input or output to the shortest required data width might not be a proper choice. The reason for that is because it affects throughput negatively if the blocks that connect to the FEC IP are able to drive or receive data at the original data width required or delivered by inner FEC sub-blocks. Furthermore, if it is required a regular data input or output width, a simple wrapper block could be implemented over the FEC IP to serialize input or output data in a regular width. Another aspect related to multi-width data I/O is the lack of such support for the depuncturing unit - only accepts a single bit due to design simplification reasons, whereas it could accept a wider input width in some cases. It negatively affects the overall communication performance of the block for the MCS IDs 0, 1 and 2.

Chapter 3 describes a similar work [26, 34] that might be used for comparison reasons. Despite being a very complete work as it is an end-to-end development of a PHY layer transceiver, including elaboration of analog circuits and chip tapeout, it does not offer a detailed RTL design description to be compared against this work. Also, it implements the complete PHY I IEEE 802.15.7 layer, whereas this work proposes an IP that implements all required FEC mechanisms for PHY I, II, and III. Moreover, the technology devices (AMS 0.35 um CMOS and Xilinx Virtex-5 xc5v1x110t) used at their work are very different from the ones used in this master thesis (TSMC 0.18um CMOS and Cyclone 5CGXFC5C6F27C7N). Therefore, results from [26, 34] are not quantitatively comparable even though some high level aspects are very similar, as explained in section 3.2

6 Conclusion

6.1 Future Works

This master thesis enables many future works at different levels of complexity and domain areas. The following list provides possible ramifications of this work:

- Investigation of Viterbi Decoder: As this work used a third party open IP, the in-depth analysis of the architectural aspects of this block has not been done. It is a relevant activity since it uses a considerable amount of resources in the FEC IP. Moreover, it has been found a possible malfunction in its functional behavior and because of that memory sizes used in the instantiated block are larger than what is really required for IEEE 802.15.7 - and this is another motivation for such investigation.

- Improvements in IP Verification: FEC IP is data-path centric design with a considerable level of complexity, and advanced verification flows might be needed to cover corner-case scenarios. For instance, traditional approaches in formal verification are not feasible for FEC IP due its high number of design states for the available computing resources. Assume-guarantee [70], case split [71] and formal bug hunting [72] techniques can be employed to lighten the problem treated by verification tools and use computational resources in a more reasonable way.

- RS Codec with "k" and "n" defined in run-time: The IP used for RS Coded requires definition of "k" and "n" in its parameter interface. As IEEE 802.15.7 requires six different combinations of "k" and "n", six instances of the RS Coded were needed. In order to avoid logic duplication by the multiple instantiations of the same IP, it could accept "k" and "n" as configuration in its port interface. Such improvement would save many hardware resources.

- Multi-frame FEC: As discussed in section 5.5, enabling multi-frame processing in FEC would improve overall latency and throughput, and some changes at the architectural level of the FEC IP and some sub-blocks (i.e Interleaver) are needed to enable it.

- Make FEC IP power aware: As depicted by section 5.3.4, a given operating mode does not require all sub-blocks of the FEC IP. To save power consumption, a power intent model for the FEC IP must be created (usually in UPF), and power saving techniques have to be applied on the design level (e.g clock gating).

- Hardware-in-the-loop infrastructure for testing: In order to assess the FEC IP in a real scenario, a hardware test platform must be created. This project would involve the use of FPGA devices and the implementation of the front-end analog circuit to control the LEDs and photodetectors used to transmit and receive data. With such infrastructure, it is possible not only validate the FEC IP, but also evaluate BER performance at different communication channel conditions.

- Tapeout of FEC IP: This work analyzes some preliminary synthesis results for the proposed IP, but this master thesis is still faraway from the fabrication of such IP as it requires other steps in the ASIC flow.

6.2 Final Remarks

This master thesis presented a complete development flow of a communication IP at RTL level from the specification analysis (IEEE 802.15.7) to its implementation results. The FEC IP is a complex hardware block with many options and requires multiple ECC techniques, and its synthesis results provide a large information set that was analyzed in detail. Data collection for both FPGA and ASIC flows helped to understand the pros and cons of each design technology. For instance, FPGA devices are applicable to VLC applications that only require PHY I; however, when data rate requirements become more challenging, ASIC devices might be the only viable option for it. However, the results indicate that there is room for improvements in both flows as the proposed FEC IP is a first workable solution. Overall, the proposed RTL architecture for IEEE 802.15.7 FEC is very suitable to PHY I operating modes. However, more demanding operating modes in PHY II require some architectural enhancements in the proposed IP, which is able to cope with an optical clock rate up to 15 Mbps - and 30, 60, and 120 Mbps are not covered mostly because it fails the latency requirements.

There are many options detailed by future works (6.1), but the development of test infrastructure that is able to validate FEC in a real scenario is one of the most relevant subsequent projects. With such framework, it is possible to measure BER performance - an important metric in communication systems - and improve confidence in the developed IP. To do so, it is needed to implement the modulation step and the analog circuit that controls the LEDs and photodetectors. The exploration of these other areas of IEEE 802.15.7 also helps to uncover other possible communication performance bottlenecks, which determines if further enhancements of the RTL architecture proposed in this work to improve communication performance metrics pays off or not. At the other end of the proposed future works, the next steps of the ASIC development flow could be explored to understand the effort to obtain the tape-out of the FEC IP.

It is worth to state that the developed IP core is open to the community, and it can be accessed in `https://github.com/mateusgs/FEC_IEEE.802.15.7`. By this work, it is understandable that joint effort in the development of IPs is a requirement to address the short time windows expected by the society as whole. Moreover, the base IP blocks used by FEC (RS Codec, Interleaver, and CC) perform general ECC functions, and it could be useful for other projects not related to IEEE 802.15.7. Open-sourcing for hardware systems has become more popular due to the variety and complexity of the contemporary devices and applications, and this work also takes this trend, being a more effective contribution for the communication systems area.

Bibliography

[1] Winzer, Peter J and David T Neilson: *From scaling disparities to integrated parallelism: A decathlon for a decade.* Journal of Lightwave Technology, 35(5):1099–1115, 2017.

[2] Haas, Harald: *LiFi is a paradigm-shifting 5G technology.* Reviews in Physics, 3:26–31, 2018.

[3] Haas, Harald, Liang Yin, Yunlu Wang, and Cheng Chen: *What is lifi?* Journal of lightwave technology, 34(6):1533–1544, 2015.

[4] Azhar, Ahmad Helmi, Tuan Anh Tran, and Dominic O'Brien: *A gigabit/s indoor wireless transmission using MIMO-OFDM visible-light communications.* IEEE photonics technology letters, 25(2):171–174, 2012.

[5] Gomez, Ariel, Kai Shi, Crisanto Quintana, Masaki Sato, Grahame Faulkner, Benn C Thomsen, and Dominic O'Brien: *Beyond 100-Gb/s indoor wide field-of-view optical wireless communications.* IEEE Photonics Technology Letters, 27(4):367–370, 2014.

[6] Stefan, Irina, Harald Burchardt, and Harald Haas: *Area spectral efficiency performance comparison between VLC and RF femtocell networks.* In *2013 IEEE international conference on communications (ICC)*, pages 3825–3829. IEEE, 2013.

[7] Ayyash, Moussa, Hany Elgala, Abdallah Khreishah, Volker Jungnickel, Thomas Little, Sihua Shao, Michael Rahaim, Dominic Schulz, Jonas Hilt, and Ronald Freund: *Coexistence of WiFi and LiFi toward 5G: concepts, opportunities, and challenges.* IEEE Communications Magazine, 54(2):64–71, 2016.

[8] Friis, Harald T: *A note on a simple transmission formula.* Proceedings of the IRE, 34(5):254–256, 1946.

[9] Pathak, Parth H, Xiaotao Feng, Pengfei Hu, and Prasant Mohapatra: *Visible light communication, networking, and sensing: A survey, potential and challenges.* IEEE communications surveys & tutorials, 17(4):2047–2077, 2015.

[10] Uysal, Murat and Hatef Nouri: *Optical wireless communications—An emerging technology.* In *2014 16th International Conference on Transparent Optical Networks (ICTON)*, pages 1–7. IEEE, 2014.

[11] Khalighi, Mohammad Ali and Murat Uysal: *Survey on free space optical communication: A communication theory perspective.* IEEE communications surveys & tutorials, 16(4):2231–2258, 2014.

[12] Haruyama, Shinichiro, V Chairman, and J Yokohama: *Japan's Visible Light Communications Consortium and its standardization activities.* Presentation at IEEE, 802, 2008.

[13] 7, IEEE Std. 802.15.: *IEEE Standard for Local and Metropolitan Area Networks—Part 15.7: Short-Range Wireless Optical Communication Using Visible Light.* 2011.

[14] Mariappan, Vinayagam and Jaesang Cha: *IEEE802. 15.7 m OWC PHY Specification Overview.* IEE E COMSOC MMTC Communications-Frontier s, 13(8):25–28, 2018.

[15] Hikmatyarsyah, Sasono Rahardjo and Juliati Junde: *Light Fidelity (Li-Fi): Security and Market Sector.* 2019.

[16] Haas, Harald and Tezcan Cogalan: *LiFi Opportunities and Challenges.* In *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, pages 361–366. IEEE, 2019.

[17] Herman, R. M. and A. Asgharian. *J. Mol. Spectrosc.*, 19:305, 1966.

[18] Herman, R. M. and A. Asgharian. *J. Mol. Spectrosc.*, 19:305, 1966.

[19] Herman, R. M. and A. Asgharian. *J. Mol. Spectrosc.*, 19:305, 1966.

[20] Kim, Sunghwan and Sung Yoon Jung: *Novel FEC coding scheme for dimmable visible light communication based on the modified Reed–Muller codes.* IEEE photonics technology letters, 23(20):1514–1516, 2011.

[21] Jungnickel, V, M Hinrichs, KL Bober, C Kottke, AA Corici, M Emmelmann, J Rufo, P B Bök, D Behnke, M Riege, *et al.*: *Enhance lighting for the internet of things.* In *2019 Global LIFI Congress (GLC)*, pages 1–6. IEEE, 2019.

[22] Alani, Mohammed M: *Guide to OSI and TCP/IP models.* 2014.

[23] Zimmermann, Hubert: *OSI reference model-the ISO model of architecture for open systems interconnection.* IEEE Transactions on communications, 28(4):425–432, 1980.

[24] Hussain, Babar: *Design and implementation of visible light communication systems.* PhD thesis, Hong Kong University of Science and Technology, 2015.

[25] Che, Fengyu: *VLC Transceiver System Design Using LED for IEEE 802.15. 7 Standard.* PhD thesis, Hong Kong University of Science and Technology, 2015.

[26] Che, Fengyu, Babar Hussain, Liang Wu, and C Patrick Yue: *Design and implementation of IEEE 802.15. 7 VLC PHY-I transceiver.* In *2014 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pages 1–4. IEEE, 2014.

[27] Gavrincea, Ciprian G, Jorge Baranda, and Pol Henarejos: *Rapid prototyping of standard-compliant visible light communications system.* IEEE Communications Magazine, 52(7):80–87, 2014.

[28] Baranda, Jorge, Pol Henarejos, and Ciprian G Gavrincea: *An SDR implementation of a visible light communication system based on the IEEE 802.15. 7 standard.* In *ICT 2013*, pages 1–5. IEEE, 2013.

[29] Hussain, Waqas, H Fatih Ugurdag, and Murat Uysal: *Software defined VLC system: Implementation and performance evaluation.* In *2015 4th International Workshop on Optical Wireless Communications (IWOW)*, pages 117–121. IEEE, 2015.

[30] Turan, Bugra, Omer Narmanlioglu, Sinem Coleri Ergen, and Murat Uysal: *Physical layer implementation of standard compliant vehicular VLC.* In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, 2016.

[31] Namonta, Pornchanok and Panarat Cherntanomwong: *Real time vital sign transmission using IEEE 802.15. 7 VLC PHY-I transceiver.* In *2017 International Electrical Engineering Congress (iEECON)*, pages 1–4. IEEE, 2017.

[32] Namonta, Pornchanok and Panarat Cherntanomwong: *The improvement of repeater system for visible light communication.* In *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6. IEEE, 2016.

[33] Ahmadinejad, Hosein, Abolfazl Falahati, and Ebrahim Shafiee: *Design and Implementation of a Visible Light Communication and Coding System Based on IEEE802. 15.7 Standard.* In *2019 2nd West Asian Colloquium on Optical Wireless Communications (WACOWC)*, pages 106–110. IEEE, 2019.

[34] Che, Fengyu, Liang Wu, Babar Hussain, Xianbo Li, and C Patrick Yue: *A fully integrated IEEE 802.15. 7 visible light communication transmitter with on-chip 8-W 85% efficiency boost LED driver.* Journal of Lightwave Technology, 34(10):2419–2430, 2016.

[35] Cummings, Clifford E: *Clock domain crossing (cdc) design & verification techniques using systemverilog.* SNUG-2008, Boston, 2008.

[36] Cooporation, Altera: *Reed-Solomon II IP Core User Guide*, 2016. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_rsii.pdf`.

[37] AMBA, ARM: *4 AXI4-Stream Protocol*, 2010.

[38] Reed, Irving S and Gustave Solomon: *Polynomial codes over certain finite fields.* Journal of the society for industrial and applied mathematics, 8(2):300–304, 1960.

[39] Immink, KA Schouhamer: *Reed-Solomon codes and the compact disc.* In *Reed-Solomon codes and their applications*, pages 41–59. IEEE Press, 1994.

[40] Reed, Irving S: *Reed–Solomon codes: A historical overview.* New York: IEEE Press, 1991.

[41] Clarke, CKP: *Reed-Solomon error correction. White Paper WHP 031.* British Broadcasting Corporation Research and Development, 2002.

[42] Silva, Mateus, Guilherme Silvano, and Ricardo Duarte: *RTL Development of a Parameterizable Reed-Solomon Codec.* IET Computers & Digital Techniques, 15(1):xxx–xxx, 2021.

[43] Geisel, William A: *Tutorial on Reed-Solomon error correction coding.* 1990.

[44] Asghar, Rizwan: *Flexible Interleaving Sub–systems for FEC in Baseband Processors.* PhD thesis, Linköping University Electronic Press, 2010.

[45] Shi, Yun Q, Xi Min Zhang, Zhi Cheng Ni, and Nirwan Ansari: *Interleaving for combating bursts of errors.* IEEE circuits and systems magazine, 4(1):29–42, 2004.

[46] Silva, Mateus and Arajo, Gabriel, and Bacelar, Elisa and Duarte, Ricardo: *Design and Verification of a Parameterizable Interleaver and Deinterleaver for Visible Light Communication.* In *10th Workshop on Circuits and Systems Design (WCAS)*, pages 48–52. Unicamp, 2020.

[47] Balakrishnan, Hari, Christopher J Terman, and George C Verghese: *Bits, Signals, and Packets.*

[48] Viterbi, Andrew: *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm.* IEEE transactions on Information Theory, 13(2):260–269, 1967.

[49] Tulasi, J, T Venkata Lakshmi, and M Kamaraju: *FPGA Implementation of Convolutional Encoder And Hard Decision Viterbi Decoder.* International Journal of Computer & Communication Technology (IJCCT), 3(4), 2012.

[50] Suganya, GS and G Kavya: *RTL design and VLSI implementation of an efficient convolutional encoder and adaptive Viterbi decoder.* In *2013 International Conference on Communication and Signal Processing*, pages 494–498. IEEE, 2013.

[51] Gmbh, Creonic: *Viterbi Decoder User Guide - Version 1.0.1*, 2019. `https://www.creonic.com/wp-content/uploads/UG_creonic_viterbi_decoder.pdf`.

[52] Benedetto, Sergio and Guido Montorsi: *Role of recursive convolutional codes in turbo codes.* Electronics Letters, 31(11):858–859, 1995.

[53] Seligman, Erik, Tom Schubert, and MV Achutha Kiran Kumar: *Formal verification: an essential toolkit for modern VLSI design.* Morgan Kaufmann, 2015.

[54] Coorporation, Intel: *AN 311: Standard Cell ASIC to FPGA Design Methodology and Guidelines*, 2009. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/an/an311.pdf`.

[55] Cooporation, Altera: *Cyclone V device Handbook*, 2012. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_5v1.pdf`.

[56] Cooporation, Intel: *Cyclone V device Overview*, 2018. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_51001.pdf`.

[57] TSMC: *TCB018GBWP7T 0.18um Standard Cell Library Databook - Version 270a*, May 2019.

[58] Song, Zheng, Xiliang Liu, Zongming Jin, Xiaokun Zhao, Qiongbing Liu, Yun Yin, and Baoyong Chi: *A 180nm CMOS wireless transceiver by utilizing guard band for narrowband IoT applications.* In *2015 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pages 1–4. IEEE, 2015.

[59] Song, Zheng and Baoyong Chi: *Two 180nm CMOS wireless transceivers for IoT applications.* In *2016 URSI Asia-Pacific Radio Science Conference (URSI AP-RASC)*, pages 1000–1002. IEEE, 2016.

[60] Coorporation, Intel: *Quartus Compilation Reports*, 2017. `https://www.intel.com/content/www/us/en/programmable/quartushelp/17.0/report/rpt/rpt_list_format.htm`.

[61] Albrecht, Christoph: *IWLS 2005 Benchmarks*, 2005. `https://iwls.org/iwls2005/benchmarks.html`.

[62] Kang, Sung Mo and Yusuf Leblebici: *CMOS digital integrated circuits.* Tata McGraw-Hill Education, 2003.

[63] Cooporation, Intel: *Intel® Quartus® Prime Pro Edition User Guide: Power Analysis and Optimization*, 2020. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qpp-power.pdf`.

[64] Systens, Cadence Design: *Genus Low Power for Legacy UI*, 2017.

[65] Wang, Xinsheng and Mingyan Yu: *Power research of JPEG circuits in FPGA.* In *2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 206–208. IEEE, 2011.

[66] Bikki, Pavankumar, Pitchai Karuppanan, *et al.*: *SRAM cell leakage control techniques for ultra low power application: A Survey.* Circuits and systems, 8(02):23, 2017.

[67] Wu, Qing, Massound Pedram, and Xunwei Wu: *Clock-gating and its application to low power design of sequential circuits.* IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 47(3):415–420, 2000.

[68] Karmann, Juergen and Wolfgang Ecker: *The semantic of the power intent format UPF: Consistent power modeling from system level to implementation.* In *2013 23rd international workshop on power and timing modeling, optimization and simulation (PATMOS)*, pages 45–50. IEEE, 2013.

[69] Cooporation, Intel: *Understanding Metastability in FPGAs*, 2009. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01082-quartus-ii-metastability.pdf`.

[70] McMillan, Kenneth: *Modular specification and verification of a cache-coherent interface.* In *2016 Formal Methods in Computer-Aided Design (FMCAD)*, pages 109–116. IEEE, 2016.

[71] Devarajegowda, Keerthikumara, Lorenzo Servadei, Zhao Han, Michael Werner, and Wolfgang Ecker: *Formal verification methodology in an industrial setup.* In *2019 22nd Euromicro Conference on Digital System Design (DSD)*, pages 610–614. IEEE, 2019.

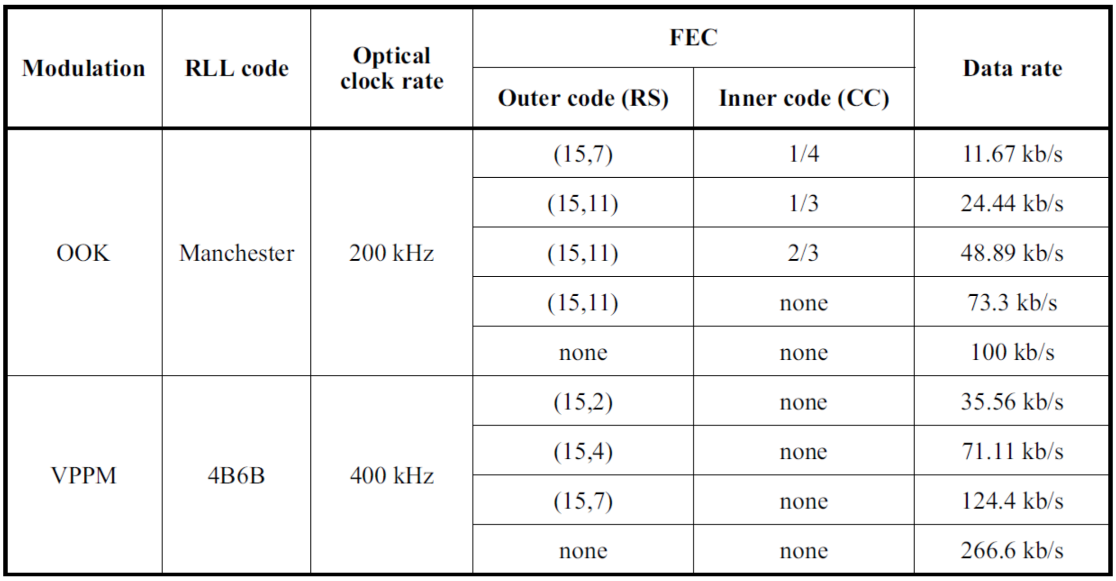
[72] Eslinger, Mark and P Yeung: *Formal Bug Hunting with "River Fishing" Techniques.* In *Design and Verification Conference and Exhibition (DVCon)*, 2019.

Annex

ANNEX A – Operating modes in IEEE 802.15.7

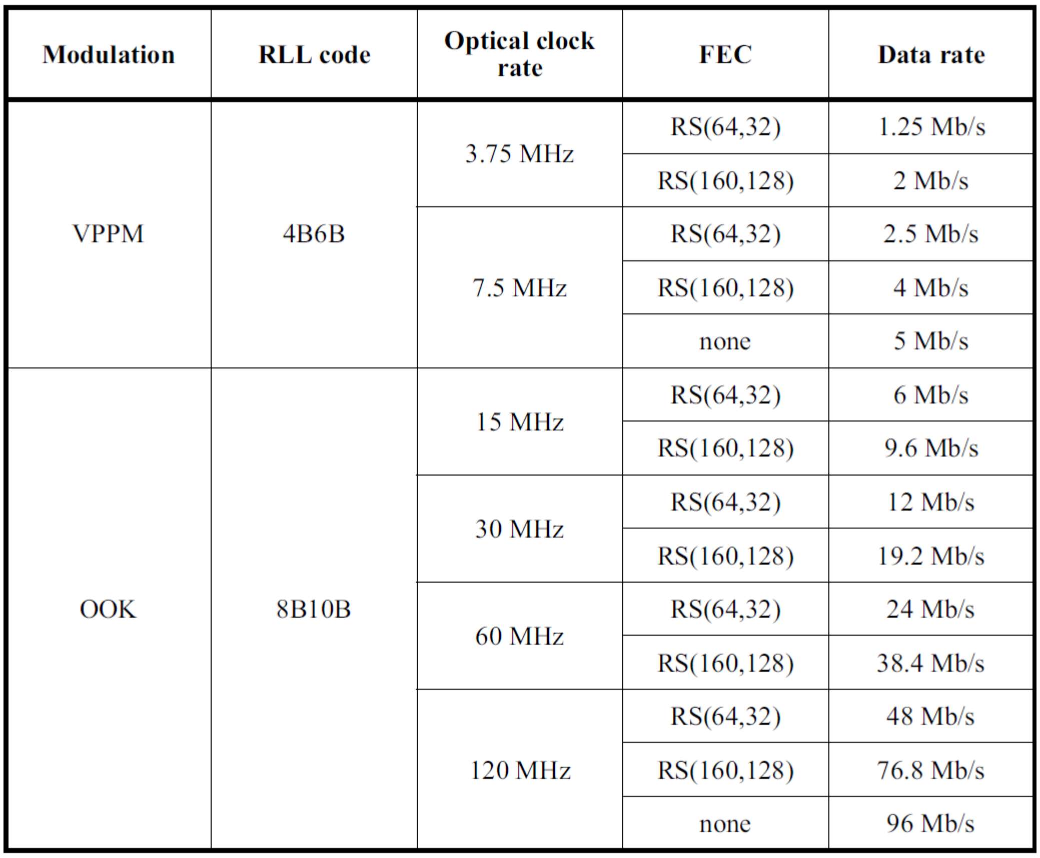
Fig. 56, 57, and 58 have the complete information about all operating modes implemented by PHY I, II, and III.

Figure 56 – PHY I operating modes ([13], p. 213).



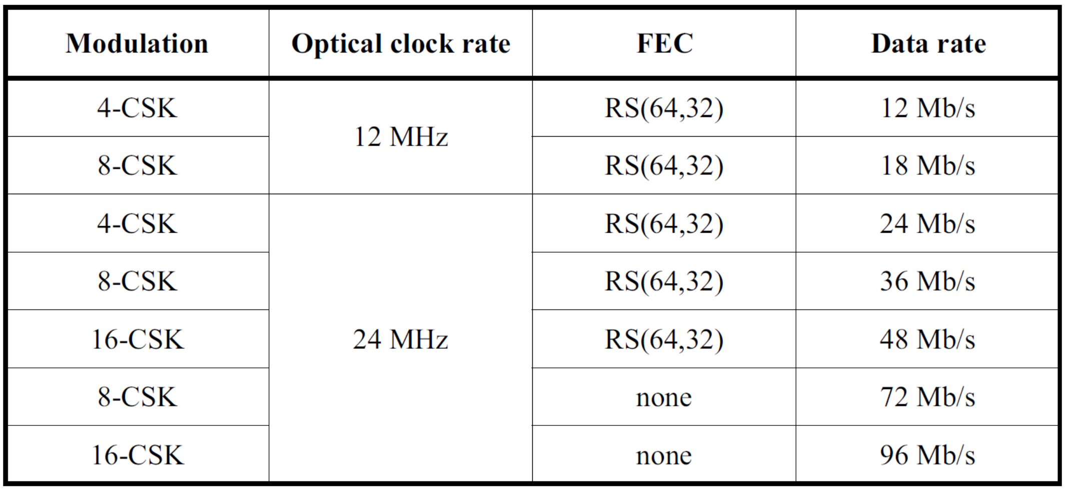
Modulation	RLL code	Optical clock rate	FEC		Data rate
			Outer code (RS)	Inner code (CC)	
OOK	Manchester	200 kHz	(15,7)	1/4	11.67 kb/s
			(15,11)	1/3	24.44 kb/s
			(15,11)	2/3	48.89 kb/s
			(15,11)	none	73.3 kb/s
			none	none	100 kb/s
VPPM	4B6B	400 kHz	(15,2)	none	35.56 kb/s
			(15,4)	none	71.11 kb/s
			(15,7)	none	124.4 kb/s
			none	none	266.6 kb/s

Figure 57 – PHY II operating modes ([13], p. 213).



Modulation	RLL code	Optical clock rate	FEC	Data rate
VPPM	4B6B	3.75 MHz	RS(64,32)	1.25 Mb/s
			RS(160,128)	2 Mb/s
		7.5 MHz	RS(64,32)	2.5 Mb/s
			RS(160,128)	4 Mb/s
			none	5 Mb/s
OOK	8B10B	15 MHz	RS(64,32)	6 Mb/s
			RS(160,128)	9.6 Mb/s
		30 MHz	RS(64,32)	12 Mb/s
			RS(160,128)	19.2 Mb/s
		60 MHz	RS(64,32)	24 Mb/s
			RS(160,128)	38.4 Mb/s
		120 MHz	RS(64,32)	48 Mb/s
			RS(160,128)	76.8 Mb/s
			none	96 Mb/s

Figure 58 – PHY III operating modes ([13], p. 214).



Modulation	Optical clock rate	FEC	Data rate
4-CSK	12 MHz	RS(64,32)	12 Mb/s
8-CSK		RS(64,32)	18 Mb/s
4-CSK	24 MHz	RS(64,32)	24 Mb/s
8-CSK		RS(64,32)	36 Mb/s
16-CSK		RS(64,32)	48 Mb/s
8-CSK		none	72 Mb/s
16-CSK		none	96 Mb/s

ANNEX B – RTL schematics of FEC IP

Fig. 59, 60, 61, 62, 63, 64, 65, 66, and 67 are the RTL schematics of the blocks described in section 4.4.

Figure 59 – Architecture of *VLC_PHY_FEC_CONTROLLER* described in section 4.4.1.

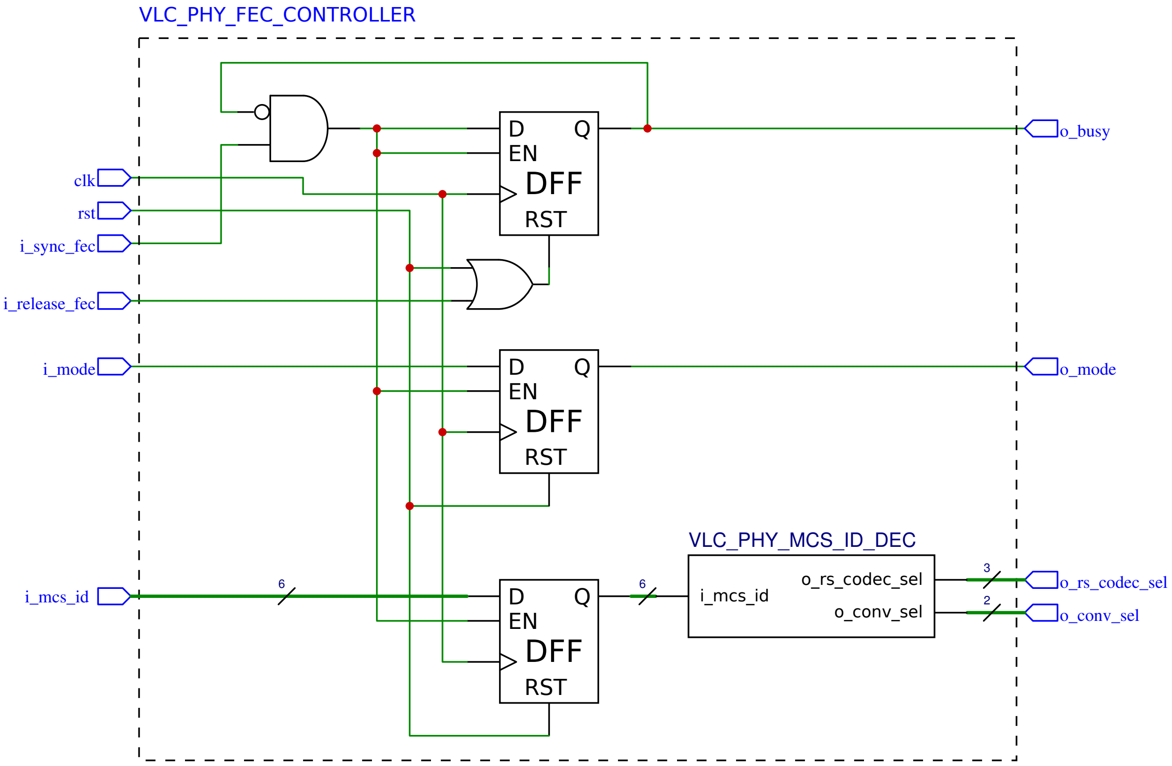


Figure 60 – Architecture of *VLC_PHY_FEC_ENCODER* described in section 4.4.2.

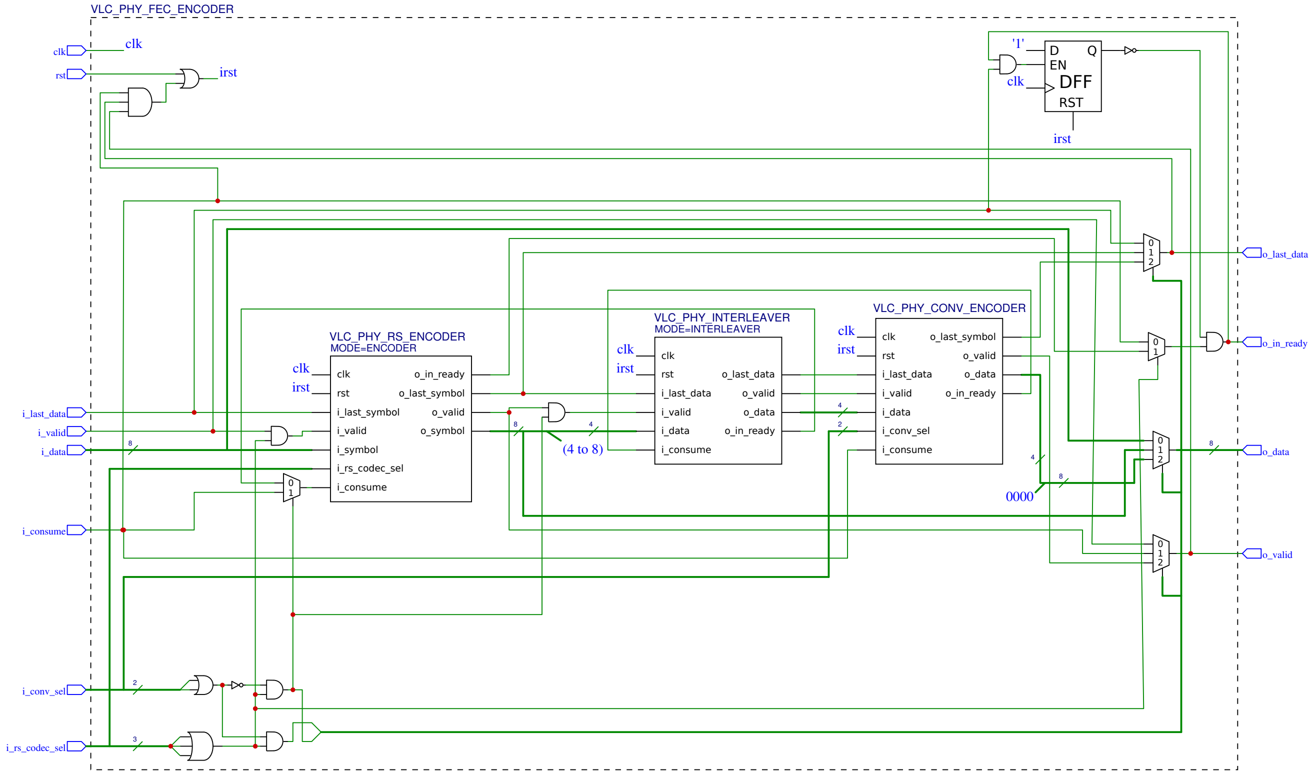


Figure 61 – Architecture of *VLC_PHY_FEC_DECODER* described in section 4.4.3.

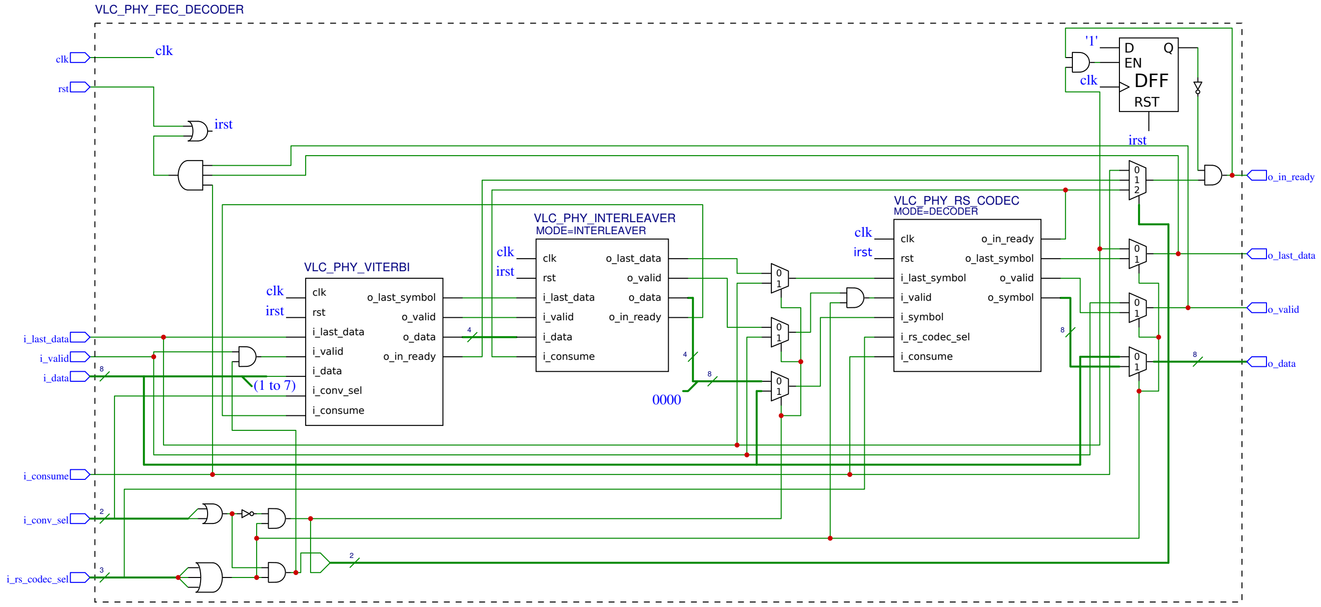


Figure 62 – Architecture of *VLC_PHY_RS_CODEC* described in section 4.4.4.

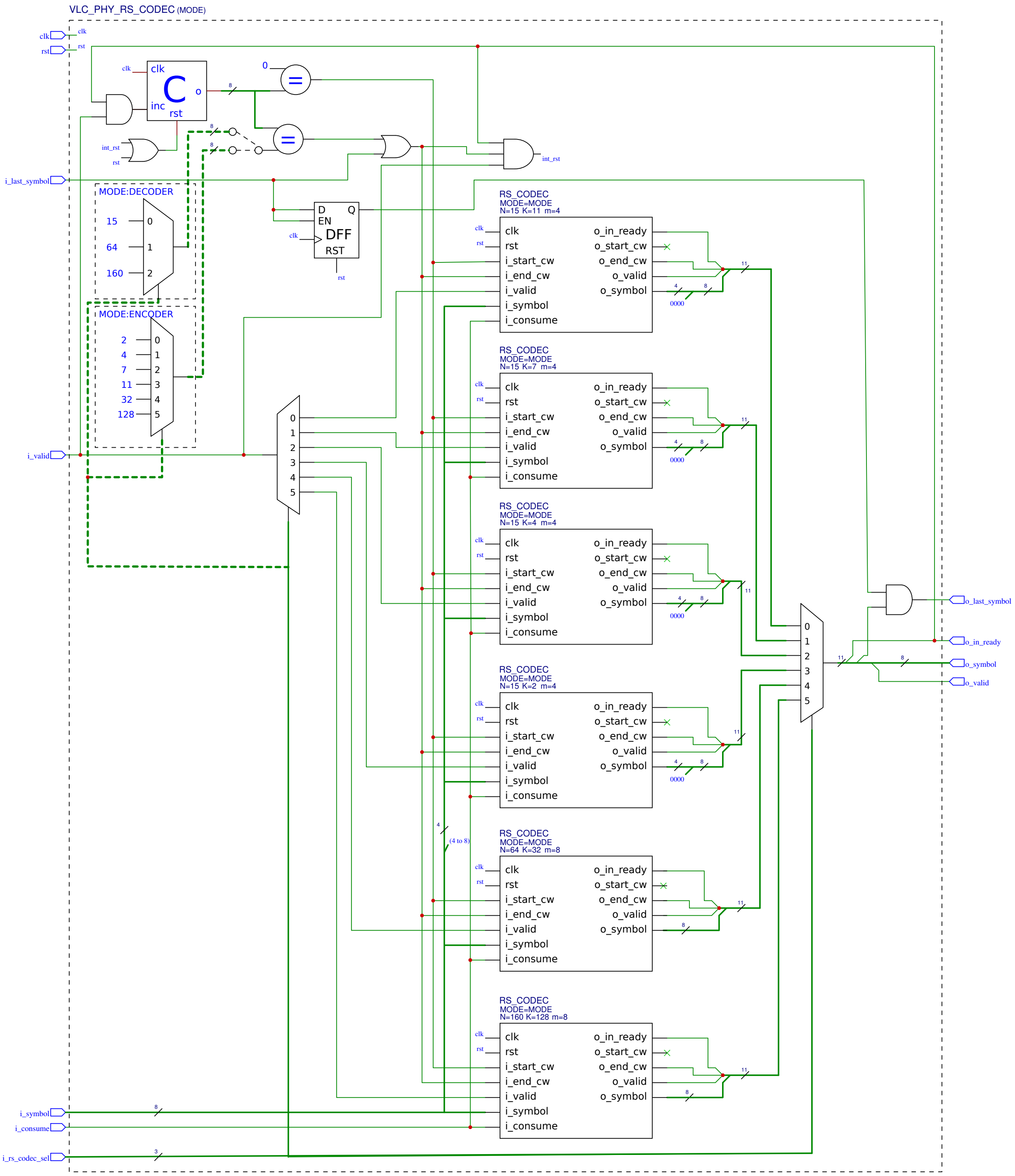


Figure 63 – Architecture of *VLC_PHY_INTERLEAVER* described in section 4.4.5.

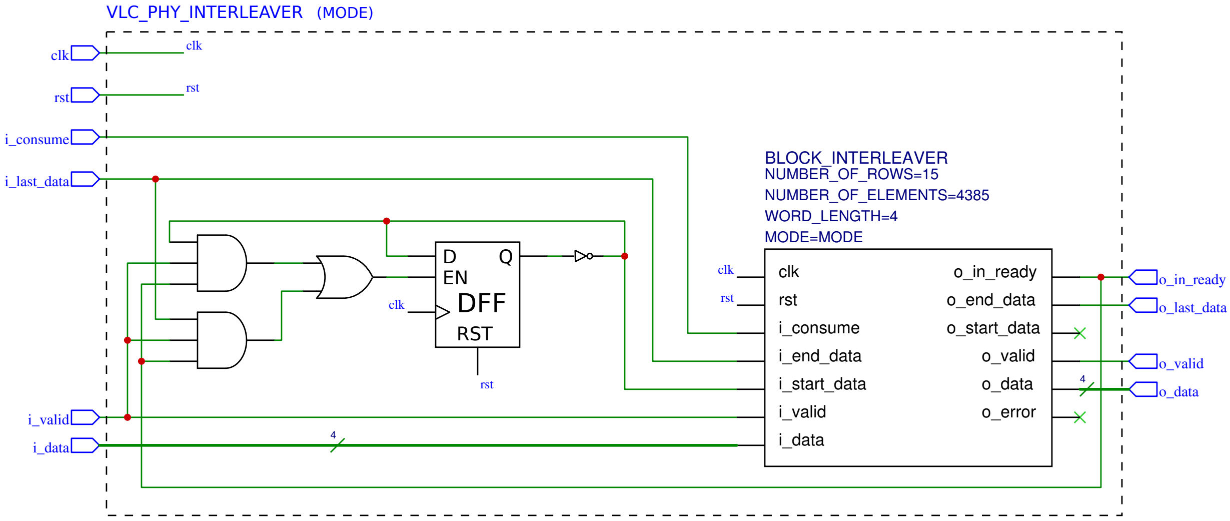


Figure 64 – Architecture of *VLC_PHY_CONVOLUTIONAL_ENCODER* described in section 4.4.6.

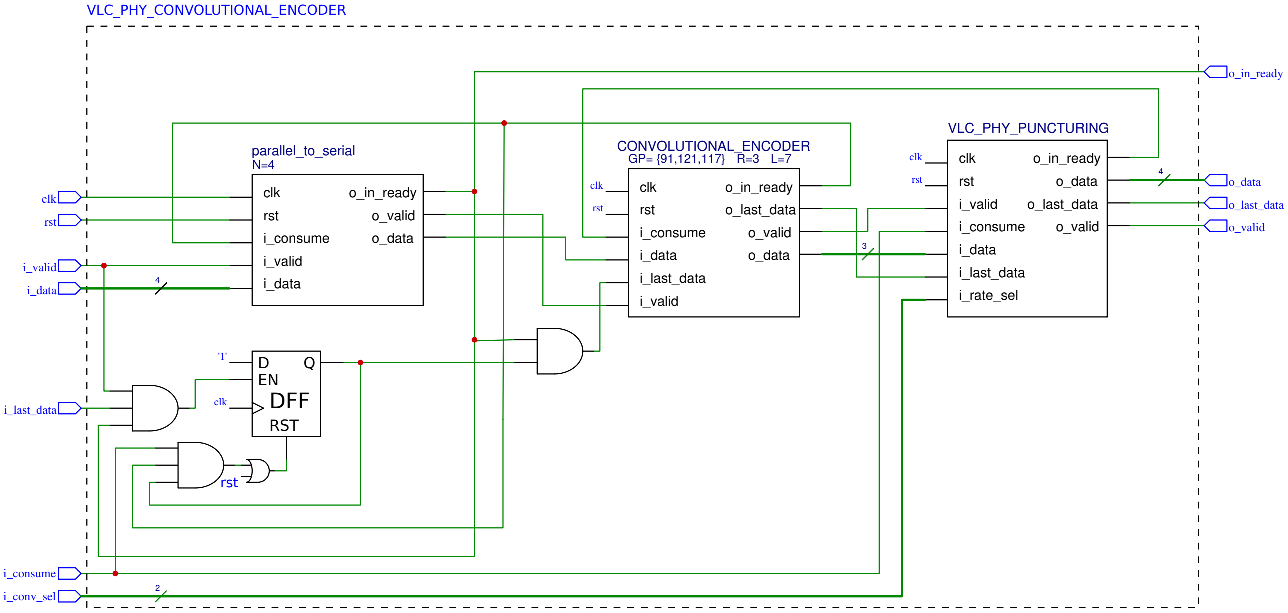


Figure 65 – Architecture of *VLC_PHY_VITERBI_DECODER* described in section 4.4.7.

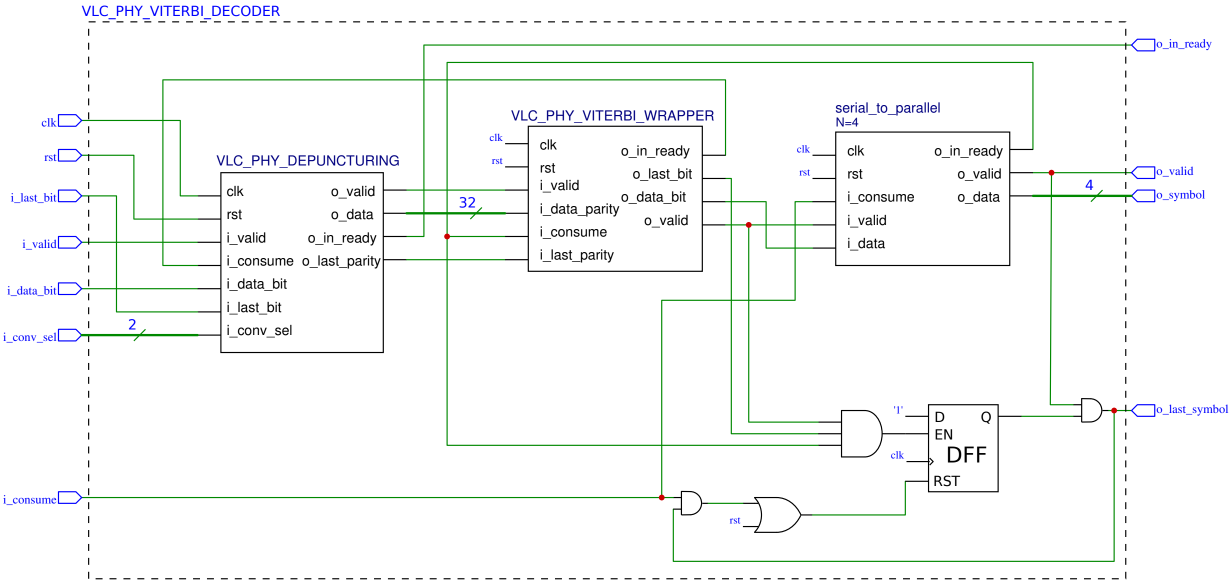


Figure 66 – Puncturing of parity bits for rate conversion from 1/3 to 1/4 and 2/3 described in section 4.4.7.1.

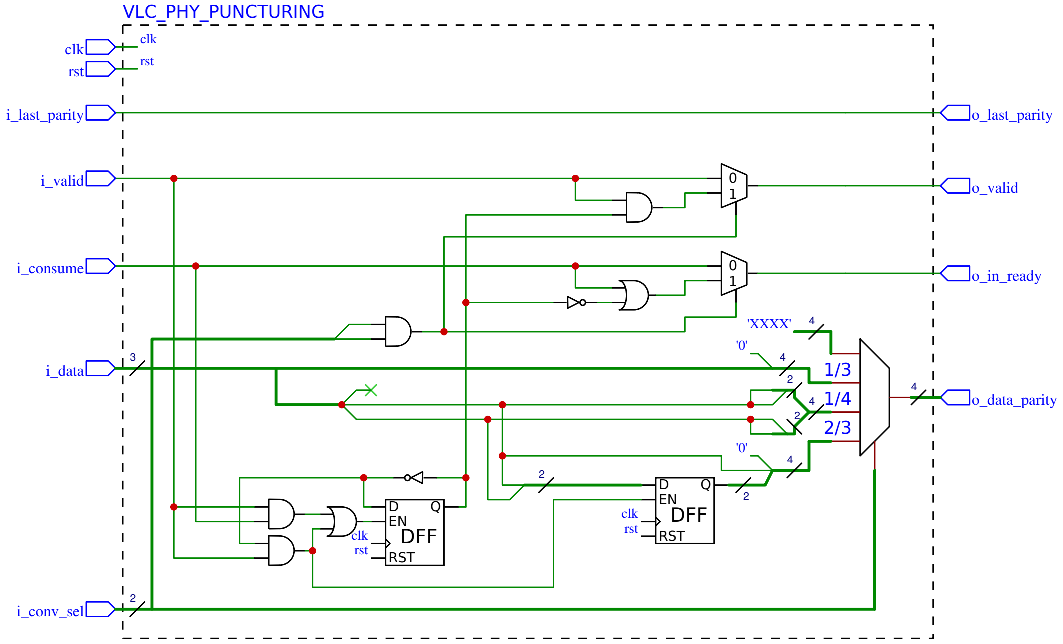
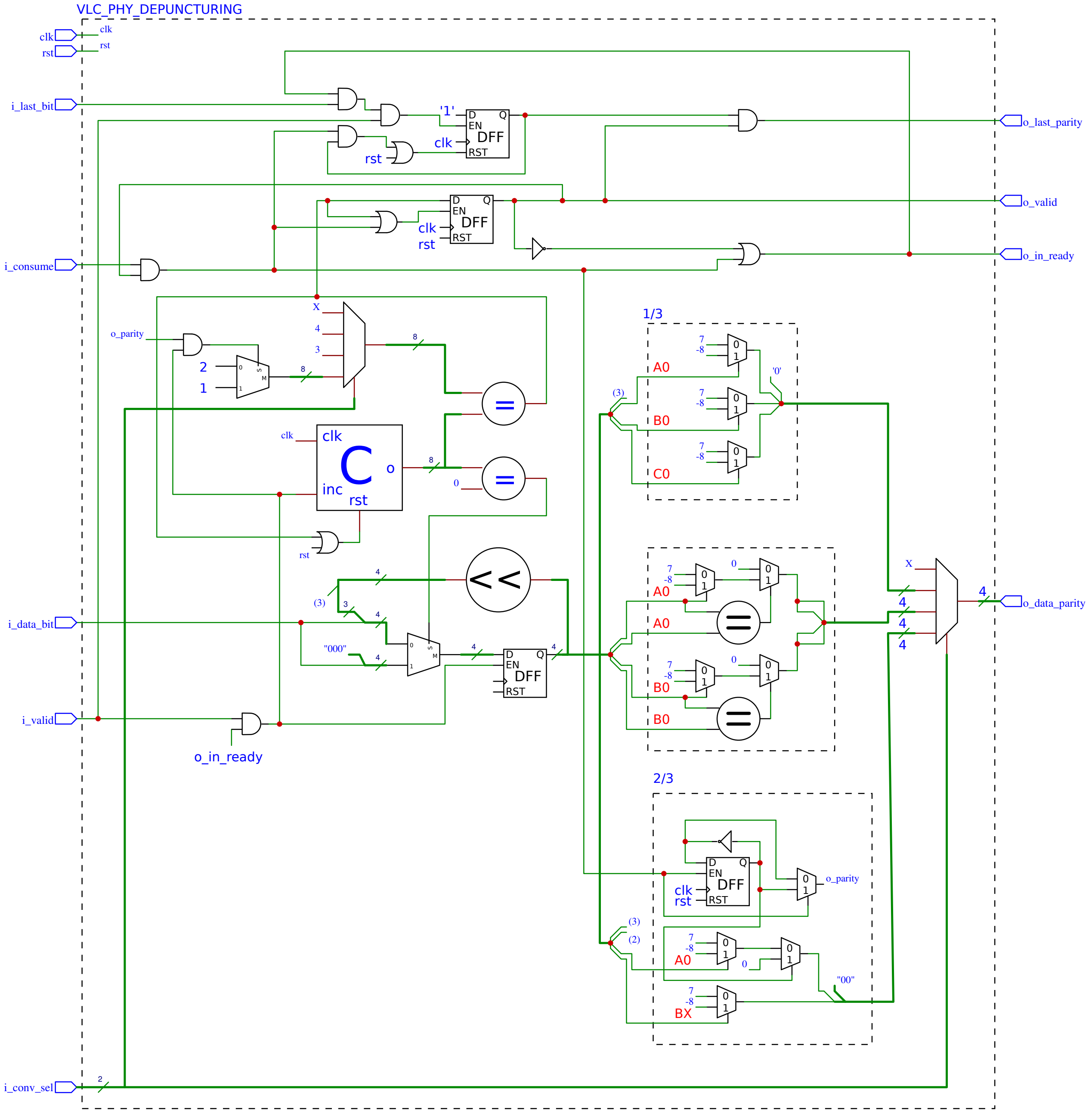


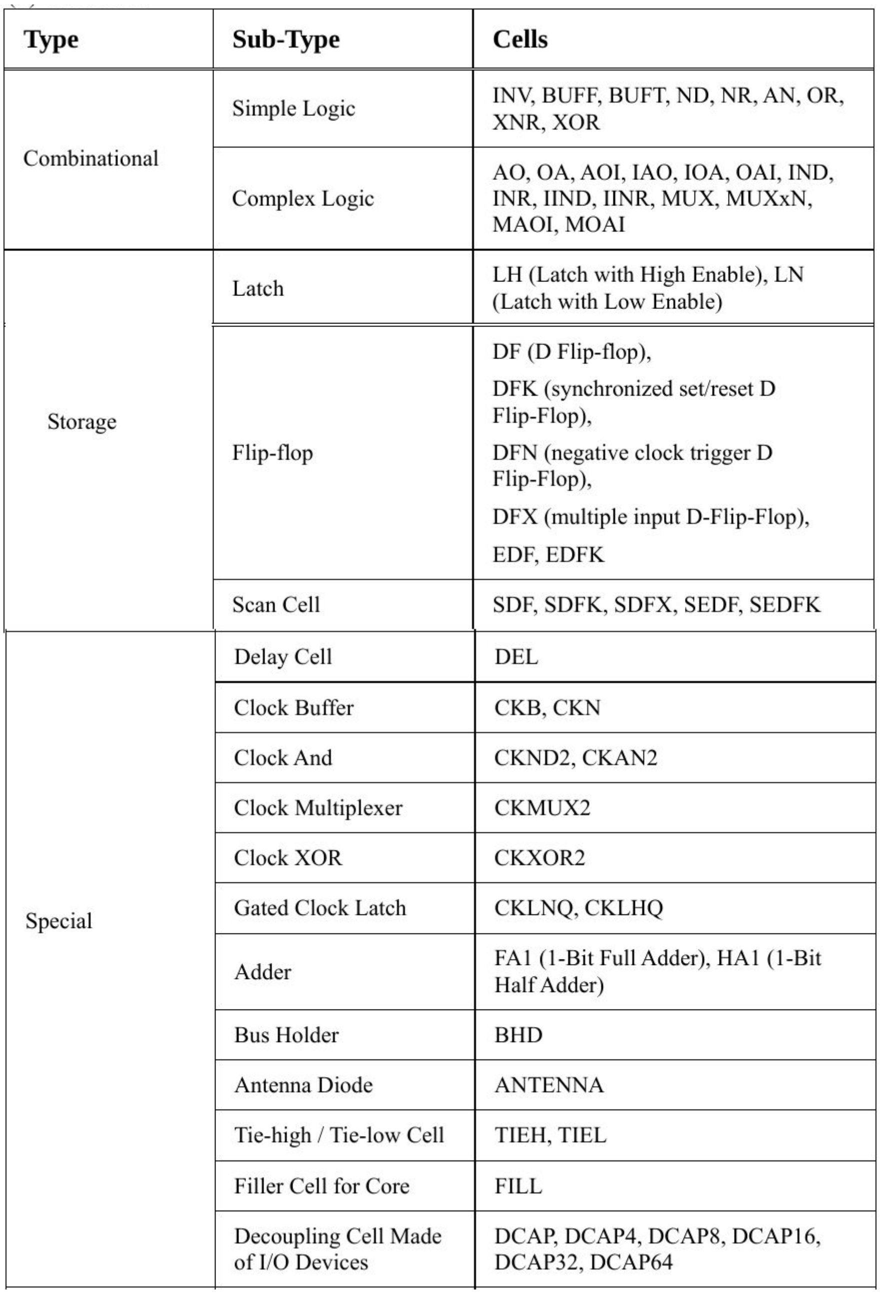
Figure 67 – Depuncturing of parity bits for rate conversion from 1/3 to 1/4 and 2/3 described in section 4.4.7.1.



ANNEX C – Standard Library Cell Details

C.1 Cell Types for TCB018GBWP7T

Figure 68 – Core cell types in TCB018GBWP7T [57].

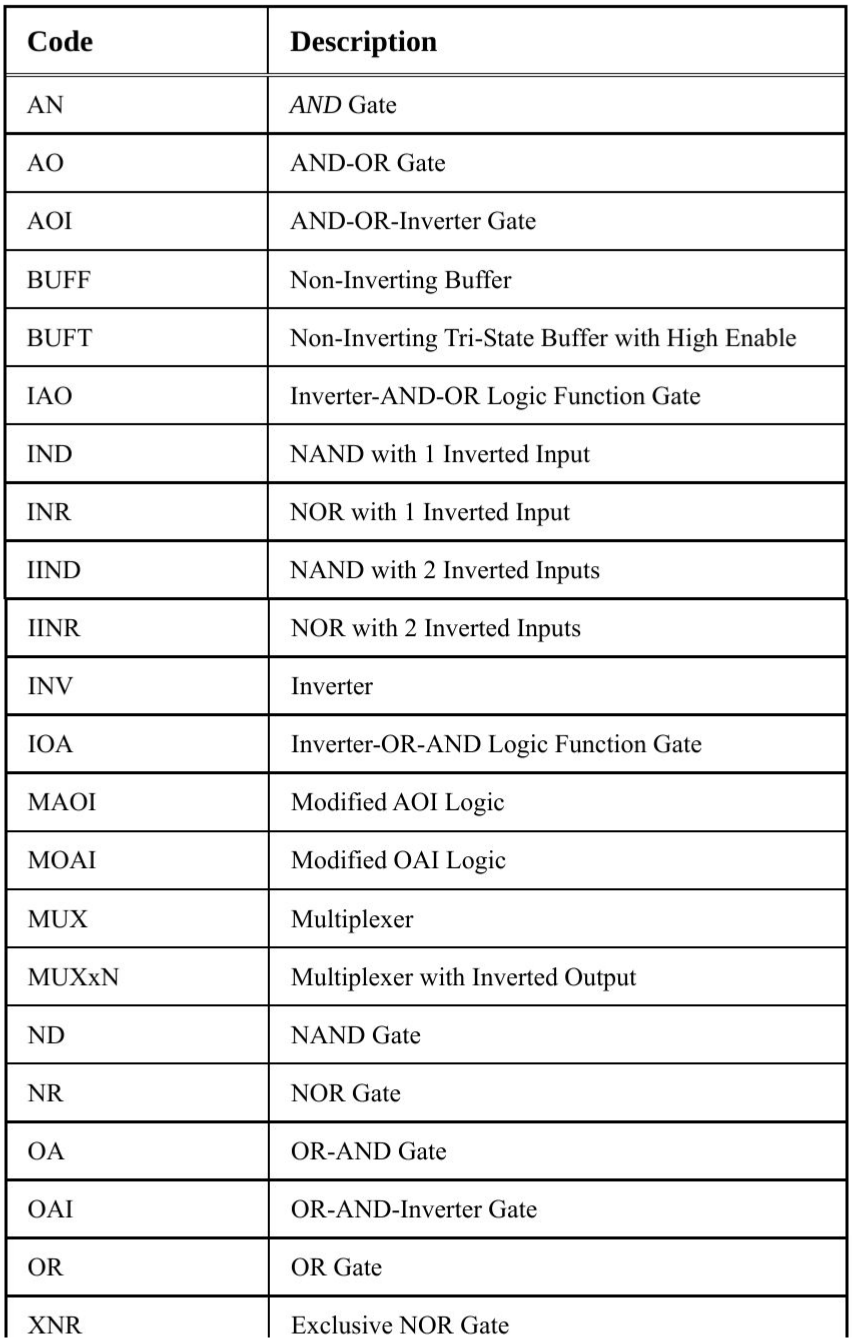


Type	Sub-Type	Cells
Combinational	Simple Logic	INV, BUFF, BUFT, ND, NR, AN, OR, XNR, XOR
	Complex Logic	AO, OA, AOI, IAO, IOA, OAI, IND, INR, IIND, IINR, MUX, MUXxN, MAOI, MOAI
Storage	Latch	LH (Latch with High Enable), LN (Latch with Low Enable)
	Flip-flop	DF (D Flip-flop), DFK (synchronized set/reset D Flip-Flop), DFN (negative clock trigger D Flip-Flop), DFX (multiple input D-Flip-Flop), EDF, EDFK
	Scan Cell	SDF, SDFK, SDFX, SEDF, SEDFK
Special	Delay Cell	DEL
	Clock Buffer	CKB, CKN
	Clock And	CKND2, CKAN2
	Clock Multiplexer	CKMUX2
	Clock XOR	CKXOR2
	Gated Clock Latch	CKLNQ, CKLHQ
	Adder	FA1 (1-Bit Full Adder), HA1 (1-Bit Half Adder)
	Bus Holder	BHD
	Antenna Diode	ANTENNA
	Tie-high / Tie-low Cell	TIEH, TIEL
	Filler Cell for Core	FILL
	Decoupling Cell Made of I/O Devices	DCAP, DCAP4, DCAP8, DCAP16, DCAP32, DCAP64

C.2 Glossary of Cell Prefix Code for TCB018GBWP7T

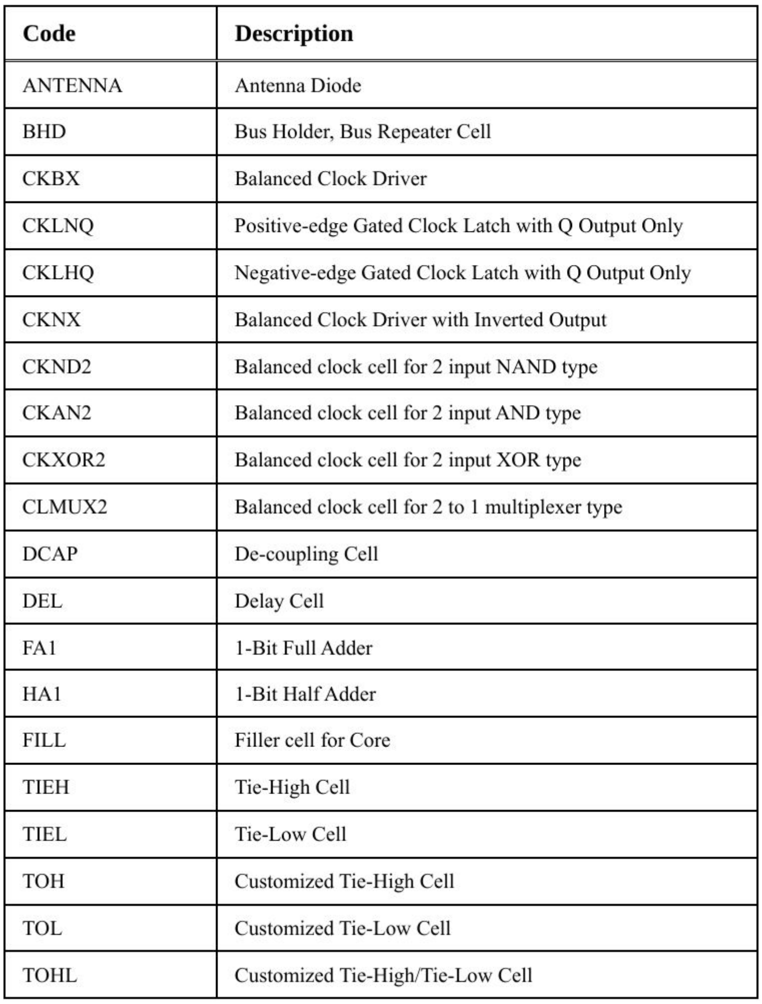
Fig. 69 and 70 describes all prefix code of TCB018GBWP7T cells.

Figure 69 – Prefix Code Descriptions of Combinational Elements [57].



Code	Description
AN	*AND* Gate
AO	AND-OR Gate
AOI	AND-OR-Inverter Gate
BUFF	Non-Inverting Buffer
BUFT	Non-Inverting Tri-State Buffer with High Enable
IAO	Inverter-AND-OR Logic Function Gate
IND	NAND with 1 Inverted Input
INR	NOR with 1 Inverted Input
IIND	NAND with 2 Inverted Inputs
IINR	NOR with 2 Inverted Inputs
INV	Inverter
IOA	Inverter-OR-AND Logic Function Gate
MAOI	Modified AOI Logic
MOAI	Modified OAI Logic
MUX	Multiplexer
MUXxN	Multiplexer with Inverted Output
ND	NAND Gate
NR	NOR Gate
OA	OR-AND Gate
OAI	OR-AND-Inverter Gate
OR	OR Gate
XNR	Exclusive NOR Gate

Figure 70 – Prefix Code Descriptions of Special Function Cells [57].



Code	Description
ANTENNA	Antenna Diode
BHD	Bus Holder, Bus Repeater Cell
CKBX	Balanced Clock Driver
CKLNQ	Positive-edge Gated Clock Latch with Q Output Only
CKLHQ	Negative-edge Gated Clock Latch with Q Output Only
CKNX	Balanced Clock Driver with Inverted Output
CKND2	Balanced clock cell for 2 input NAND type
CKAN2	Balanced clock cell for 2 input AND type
CKXOR2	Balanced clock cell for 2 input XOR type
CLMUX2	Balanced clock cell for 2 to 1 multiplexer type
DCAP	De-coupling Cell
DEL	Delay Cell
FA1	1-Bit Full Adder
HA1	1-Bit Half Adder
FILL	Filler cell for Core
TIEH	Tie-High Cell
TIEL	Tie-Low Cell
TOH	Customized Tie-High Cell
TOL	Customized Tie-Low Cell
TOHL	Customized Tie-High/Tie-Low Cell

C.3 Detailed Report of TCB018GBWP7T Utilization for the FEC IP

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.20-p003_1
  Generated on:           Nov 01 2020  02:23:57 pm
  Module:                 vlc_phy_fec_PA_MODE0
  Technology libraries:   tcb018gbwp7ttc 270
                          physical_cells
  Operating conditions:   NCCOM
  Interconnect mode:      global
  Area mode:              physical library
============================================================


    Gate       Instances     Area          Library
---------------------------------------------------------
AN2D0BWP7T             5       54.880    tcb018gbwp7ttc
AN2D1BWP7T          1225    13445.600    tcb018gbwp7ttc
AN2D2BWP7T           802    12323.853    tcb018gbwp7ttc
```

AN2D4BWP7T	124	3266.458	tcb018gbwp7ttc
AN2XD1BWP7T	33	434.650	tcb018gbwp7ttc
AN3D0BWP7T	9	118.541	tcb018gbwp7ttc
AN3D1BWP7T	267	3516.710	tcb018gbwp7ttc
AN3D2BWP7T	3	52.685	tcb018gbwp7ttc
AN3D4BWP7T	1	32.928	tcb018gbwp7ttc
AN3XD1BWP7T	1	15.366	tcb018gbwp7ttc
AN4D0BWP7T	381	5854.598	tcb018gbwp7ttc
AN4D1BWP7T	498	7652.467	tcb018gbwp7ttc
AN4D2BWP7T	1	19.757	tcb018gbwp7ttc
AN4XD1BWP7T	40	702.464	tcb018gbwp7ttc
AO211D0BWP7T	70	1075.648	tcb018gbwp7ttc
AO211D1BWP7T	1	17.562	tcb018gbwp7ttc
AO21D0BWP7T	189	2489.357	tcb018gbwp7ttc
AO21D1BWP7T	61	937.350	tcb018gbwp7ttc
AO21D2BWP7T	2	35.123	tcb018gbwp7ttc
AO221D0BWP7T	262	5176.282	tcb018gbwp7ttc
AO221D1BWP7T	1	21.952	tcb018gbwp7ttc
AO222D0BWP7T	854	20621.709	tcb018gbwp7ttc
AO222D1BWP7T	7	184.397	tcb018gbwp7ttc
AO222D2BWP7T	4	114.150	tcb018gbwp7ttc
AO22D0BWP7T	1364	23954.022	tcb018gbwp7ttc
AO22D1BWP7T	2	39.514	tcb018gbwp7ttc
AO22D2BWP7T	4	87.808	tcb018gbwp7ttc
AO31D1BWP7T	52	913.203	tcb018gbwp7ttc
AO32D1BWP7T	21	460.992	tcb018gbwp7ttc
AO33D0BWP7T	8	175.616	tcb018gbwp7ttc
AO33D1BWP7T	1	24.147	tcb018gbwp7ttc
AOI211D0BWP7T	22	289.766	tcb018gbwp7ttc
AOI211D1BWP7T	67	882.470	tcb018gbwp7ttc
AOI211XD0BWP7T	278	3661.594	tcb018gbwp7ttc
AOI21D0BWP7T	387	4247.712	tcb018gbwp7ttc
AOI21D1BWP7T	70	921.984	tcb018gbwp7ttc
AOI21D2BWP7T	3	59.270	tcb018gbwp7ttc
AOI221D0BWP7T	630	11063.808	tcb018gbwp7ttc
AOI221D1BWP7T	48	948.326	tcb018gbwp7ttc
AOI222D0BWP7T	1242	24537.946	tcb018gbwp7ttc
AOI222D1BWP7T	428	9395.456	tcb018gbwp7ttc
AOI22D0BWP7T	29378	386943.514	tcb018gbwp7ttc
AOI22D1BWP7T	218	3349.875	tcb018gbwp7ttc
AOI22D2BWP7T	4	105.370	tcb018gbwp7ttc
AOI31D0BWP7T	161	2120.563	tcb018gbwp7ttc
AOI31D1BWP7T	30	460.992	tcb018gbwp7ttc
AOI32D0BWP7T	4	70.246	tcb018gbwp7ttc
AOI32D1BWP7T	34	597.094	tcb018gbwp7ttc
AOI33D1BWP7T	24	474.163	tcb018gbwp7ttc
BUFFD10BWP7T	158	6589.990	tcb018gbwp7ttc
BUFFD12BWP7T	84	4056.730	tcb018gbwp7ttc
BUFFD1BWP7T	5	43.904	tcb018gbwp7ttc
BUFFD1P5BWP7T	60	658.560	tcb018gbwp7ttc
BUFFD2BWP7T	116	1527.859	tcb018gbwp7ttc
BUFFD2P5BWP7T	5	76.832	tcb018gbwp7ttc
BUFFD3BWP7T	150	2304.960	tcb018gbwp7ttc
BUFFD4BWP7T	215	4247.712	tcb018gbwp7ttc
BUFFD5BWP7T	124	2994.253	tcb018gbwp7ttc
BUFFD6BWP7T	64	1685.914	tcb018gbwp7ttc
BUFFD8BWP7T	107	3758.182	tcb018gbwp7ttc
CKAN2D0BWP7T	1	10.976	tcb018gbwp7ttc
CKAN2D1BWP7T	2525	27714.400	tcb018gbwp7ttc
CKAN2D2BWP7T	14	215.130	tcb018gbwp7ttc

CKAN2D4BWP7T	5	131.712	tcb018gbwp7ttc
CKAN2D8BWP7T	6	237.082	tcb018gbwp7ttc
CKBD10BWP7T	4	184.397	tcb018gbwp7ttc
CKBD12BWP7T	4	193.178	tcb018gbwp7ttc
CKBD1BWP7T	5	43.904	tcb018gbwp7ttc
CKBD2BWP7T	13	171.226	tcb018gbwp7ttc
CKBD3BWP7T	31	544.410	tcb018gbwp7ttc
CKBD4BWP7T	39	770.515	tcb018gbwp7ttc
CKBD6BWP7T	7	199.763	tcb018gbwp7ttc
CKBD8BWP7T	12	421.478	tcb018gbwp7ttc
CKMUX2D1BWP7T	735	14521.248	tcb018gbwp7ttc
CKND0BWP7T	6	39.514	tcb018gbwp7ttc
CKND10BWP7T	175	4994.080	tcb018gbwp7ttc
CKND12BWP7T	169	5935.821	tcb018gbwp7ttc
CKND1BWP7T	422	2779.123	tcb018gbwp7ttc
CKND2BWP7T	400	3512.320	tcb018gbwp7ttc
CKND2D0BWP7T	13	114.150	tcb018gbwp7ttc
CKND2D1BWP7T	922	8095.898	tcb018gbwp7ttc
CKND2D2BWP7T	153	2351.059	tcb018gbwp7ttc
CKND2D3BWP7T	167	3299.386	tcb018gbwp7ttc
CKND2D4BWP7T	126	3319.142	tcb018gbwp7ttc
CKND2D8BWP7T	143	6906.099	tcb018gbwp7ttc
CKND3BWP7T	173	2278.618	tcb018gbwp7ttc
CKND4BWP7T	182	2796.685	tcb018gbwp7ttc
CKND6BWP7T	158	3121.574	tcb018gbwp7ttc
CKND8BWP7T	211	5558.246	tcb018gbwp7ttc
CKXOR2D0BWP7T	579	11439.187	tcb018gbwp7ttc
CKXOR2D1BWP7T	3128	61799.270	tcb018gbwp7ttc
CKXOR2D2BWP7T	13	313.914	tcb018gbwp7ttc
CKXOR2D4BWP7T	209	10093.530	tcb018gbwp7ttc
DEL015BWP7T	1	8.781	tcb018gbwp7ttc
DEL01BWP7T	667	5856.794	tcb018gbwp7ttc
DFCNQD1BWP7T	2162	104412.493	tcb018gbwp7ttc
DFCNQD2BWP7T	46	2322.522	tcb018gbwp7ttc
DFD0BWP7T	12	526.848	tcb018gbwp7ttc
DFD1BWP7T	14	614.656	tcb018gbwp7ttc
DFD2BWP7T	7	353.427	tcb018gbwp7ttc
DFKCND0BWP7T	1	48.294	tcb018gbwp7ttc
DFKCND1BWP7T	2	96.589	tcb018gbwp7ttc
DFKCNQD1BWP7T	328	14400.512	tcb018gbwp7ttc
DFKCNQD2BWP7T	514	23694.989	tcb018gbwp7ttc
DFKSND1BWP7T	192	10115.482	tcb018gbwp7ttc
DFQD0BWP7T	274	11428.211	tcb018gbwp7ttc
DFQD1BWP7T	8743	364660.038	tcb018gbwp7ttc
DFQD2BWP7T	17	746.368	tcb018gbwp7ttc
DFSNQD1BWP7T	6	302.938	tcb018gbwp7ttc
DFXD1BWP7T	3	171.226	tcb018gbwp7ttc
DFXQD1BWP7T	18	987.840	tcb018gbwp7ttc
DFXQD2BWP7T	2	114.150	tcb018gbwp7ttc
EDFCNQD1BWP7T	10	614.656	tcb018gbwp7ttc
EDFKCND1BWP7T	6	368.794	tcb018gbwp7ttc
EDFKCND2BWP7T	2	136.102	tcb018gbwp7ttc
EDFKCNQD1BWP7T	952	56425.421	tcb018gbwp7ttc
EDFKCNQD2BWP7T	4	254.643	tcb018gbwp7ttc
EDFQD0BWP7T	23810	1306692.800	tcb018gbwp7ttc
EDFQD1BWP7T	37849	2077153.120	tcb018gbwp7ttc
FA1D0BWP7T	295	14894.432	tcb018gbwp7ttc
FA1D1BWP7T	114	7007.078	tcb018gbwp7ttc
FA1D2BWP7T	10	636.608	tcb018gbwp7ttc
HA1D0BWP7T	191	6289.248	tcb018gbwp7ttc

HA1D1BWP7T	6	210.739	tcb018gbwp7ttc
HA1D2BWP7T	7	291.962	tcb018gbwp7ttc
IAO21D0BWP7T	59	777.101	tcb018gbwp7ttc
IAO21D1BWP7T	11	169.030	tcb018gbwp7ttc
IAO21D2BWP7T	4	79.027	tcb018gbwp7ttc
IAO22D0BWP7T	8	140.493	tcb018gbwp7ttc
IAO22D1BWP7T	4	79.027	tcb018gbwp7ttc
IIND4D0BWP7T	8	158.054	tcb018gbwp7ttc
IINR4D0BWP7T	196	3872.333	tcb018gbwp7ttc
IND2D0BWP7T	35	384.160	tcb018gbwp7ttc
IND2D1BWP7T	1309	14367.584	tcb018gbwp7ttc
IND2D2BWP7T	1338	23497.421	tcb018gbwp7ttc
IND2D4BWP7T	1202	39579.456	tcb018gbwp7ttc
IND3D0BWP7T	12	158.054	tcb018gbwp7ttc
IND3D1BWP7T	328	4320.154	tcb018gbwp7ttc
IND3D2BWP7T	24	632.218	tcb018gbwp7ttc
IND4D0BWP7T	331	5086.278	tcb018gbwp7ttc
IND4D1BWP7T	27	474.163	tcb018gbwp7ttc
INR2D0BWP7T	46	504.896	tcb018gbwp7ttc
INR2D1BWP7T	490	5378.240	tcb018gbwp7ttc
INR2D2BWP7T	48	842.957	tcb018gbwp7ttc
INR2D4BWP7T	25	823.200	tcb018gbwp7ttc
INR2XD0BWP7T	422	4631.872	tcb018gbwp7ttc
INR2XD1BWP7T	27	474.163	tcb018gbwp7ttc
INR2XD2BWP7T	128	3090.842	tcb018gbwp7ttc
INR2XD4BWP7T	22	1110.771	tcb018gbwp7ttc
INR3D0BWP7T	125	1920.800	tcb018gbwp7ttc
INR3D1BWP7T	2	39.514	tcb018gbwp7ttc
INR4D0BWP7T	15	263.424	tcb018gbwp7ttc
INVD0BWP7T	331	2179.834	tcb018gbwp7ttc
INVD10BWP7T	18	592.704	tcb018gbwp7ttc
INVD12BWP7T	149	5560.442	tcb018gbwp7ttc
INVD1BWP7T	1838	12104.333	tcb018gbwp7ttc
INVD1P5BWP7T	106	930.765	tcb018gbwp7ttc
INVD2BWP7T	1401	12301.901	tcb018gbwp7ttc
INVD2P5BWP7T	290	3183.040	tcb018gbwp7ttc
INVD3BWP7T	790	10405.248	tcb018gbwp7ttc
INVD4BWP7T	837	12861.677	tcb018gbwp7ttc
INVD5BWP7T	268	4706.509	tcb018gbwp7ttc
INVD6BWP7T	506	9996.941	tcb018gbwp7ttc
INVD8BWP7T	869	22891.546	tcb018gbwp7ttc
IOA21D0BWP7T	6	79.027	tcb018gbwp7ttc
IOA21D1BWP7T	1010	13302.912	tcb018gbwp7ttc
IOA21D2BWP7T	31	612.461	tcb018gbwp7ttc
IOA22D2BWP7T	1	24.147	tcb018gbwp7ttc
MAOI222D1BWP7T	184	3231.334	tcb018gbwp7ttc
MAOI22D0BWP7T	32	491.725	tcb018gbwp7ttc
MAOI22D1BWP7T	6	105.370	tcb018gbwp7ttc
MAOI22D2BWP7T	3	72.442	tcb018gbwp7ttc
MOAI22D0BWP7T	6696	102893.414	tcb018gbwp7ttc
MOAI22D1BWP7T	26	456.602	tcb018gbwp7ttc
MOAI22D2BWP7T	1	24.147	tcb018gbwp7ttc
MUX2D0BWP7T	17	335.866	tcb018gbwp7ttc
MUX2D1BWP7T	2082	41133.658	tcb018gbwp7ttc
MUX2D2BWP7T	13	285.376	tcb018gbwp7ttc
MUX2ND0BWP7T	39	684.902	tcb018gbwp7ttc
MUX3D1BWP7T	122	4017.216	tcb018gbwp7ttc
MUX4D0BWP7T	1	43.904	tcb018gbwp7ttc
MUX4D1BWP7T	2	87.808	tcb018gbwp7ttc
MUX4ND0BWP7T	15	592.704	tcb018gbwp7ttc

```
MUX4ND1BWP7T            2       105.370    tcb018gbwp7ttc
ND2D0BWP7T            319      2801.075    tcb018gbwp7ttc
ND2D1BWP7T           5756     50542.285    tcb018gbwp7ttc
ND2D1P5BWP7T          348      5347.507    tcb018gbwp7ttc
ND2D2BWP7T           3068     47144.115    tcb018gbwp7ttc
ND2D2P5BWP7T          209      4129.171    tcb018gbwp7ttc
ND2D3BWP7T           2327     45974.074    tcb018gbwp7ttc
ND2D4BWP7T           1854     48838.810    tcb018gbwp7ttc
ND2D5BWP7T            534     17583.552    tcb018gbwp7ttc
ND2D6BWP7T            401     14964.678    tcb018gbwp7ttc
ND2D8BWP7T            171      8258.342    tcb018gbwp7ttc
ND3D0BWP7T            302      3314.752    tcb018gbwp7ttc
ND3D1BWP7T            240      3161.088    tcb018gbwp7ttc
ND3D2BWP7T            511     10095.725    tcb018gbwp7ttc
ND3D3BWP7T            365     10416.224    tcb018gbwp7ttc
ND3D4BWP7T             57      2252.275    tcb018gbwp7ttc
ND4D0BWP7T           6435     84756.672    tcb018gbwp7ttc
ND4D1BWP7T            100      1536.640    tcb018gbwp7ttc
ND4D2BWP7T              3        85.613    tcb018gbwp7ttc
ND4D3BWP7T              2        74.637    tcb018gbwp7ttc
NR2D0BWP7T            133      1167.846    tcb018gbwp7ttc
NR2D1BWP7T           5192     45589.914    tcb018gbwp7ttc
NR2D1P5BWP7T            8       122.931    tcb018gbwp7ttc
NR2D2BWP7T            621      9542.534    tcb018gbwp7ttc
NR2D2P5BWP7T           28       553.190    tcb018gbwp7ttc
NR2D3BWP7T            145      2864.736    tcb018gbwp7ttc
NR2D4BWP7T             53      1396.147    tcb018gbwp7ttc
NR2D5BWP7T             12       395.136    tcb018gbwp7ttc
NR2D6BWP7T             11       410.502    tcb018gbwp7ttc
NR2XD0BWP7T          9265     81354.112    tcb018gbwp7ttc
NR2XD1BWP7T            70      1075.648    tcb018gbwp7ttc
NR2XD2BWP7T            49      1290.778    tcb018gbwp7ttc
NR2XD3BWP7T            13       428.064    tcb018gbwp7ttc
NR2XD4BWP7T            15       658.560    tcb018gbwp7ttc
NR2XD8BWP7T             5       384.160    tcb018gbwp7ttc
NR3D0BWP7T            193      2118.368    tcb018gbwp7ttc
NR3D1BWP7T              7       122.931    tcb018gbwp7ttc
NR4D0BWP7T           1200     15805.440    tcb018gbwp7ttc
NR4D1BWP7T              5       109.760    tcb018gbwp7ttc
OA211D0BWP7T           47       825.395    tcb018gbwp7ttc
OA211D1BWP7T           17       335.866    tcb018gbwp7ttc
OA21D0BWP7T           285      3753.792    tcb018gbwp7ttc
OA21D1BWP7T             6        92.198    tcb018gbwp7ttc
OA21D2BWP7T             2        35.123    tcb018gbwp7ttc
OA221D0BWP7T           34       671.731    tcb018gbwp7ttc
OA221D1BWP7T           19       417.088    tcb018gbwp7ttc
OA221D2BWP7T            1        26.342    tcb018gbwp7ttc
OA222D0BWP7T            7       169.030    tcb018gbwp7ttc
OA222D1BWP7T            5       131.712    tcb018gbwp7ttc
OA22D0BWP7T            39       684.902    tcb018gbwp7ttc
OA22D1BWP7T             6       118.541    tcb018gbwp7ttc
OA31D1BWP7T            10       175.616    tcb018gbwp7ttc
OA32D1BWP7T             8       175.616    tcb018gbwp7ttc
OA33D0BWP7T             1        21.952    tcb018gbwp7ttc
OAI211D0BWP7T          31       408.307    tcb018gbwp7ttc
OAI211D1BWP7T         470      6190.464    tcb018gbwp7ttc
OAI211D2BWP7T         305      7364.896    tcb018gbwp7ttc
OAI21D0BWP7T          436      4785.536    tcb018gbwp7ttc
OAI21D1BWP7T          170      2239.104    tcb018gbwp7ttc
OAI21D2BWP7T           62      1224.922    tcb018gbwp7ttc
```

```
OAI221D0BWP7T         191     3354.266    tcb018gbwp7ttc
OAI221D1BWP7T          30      592.704    tcb018gbwp7ttc
OAI222D0BWP7T          30      592.704    tcb018gbwp7ttc
OAI222D1BWP7T           1       21.952    tcb018gbwp7ttc
OAI22D0BWP7T          130     1712.256    tcb018gbwp7ttc
OAI22D1BWP7T           10      153.664    tcb018gbwp7ttc
OAI22D2BWP7T          254     6690.970    tcb018gbwp7ttc
OAI31D0BWP7T           65      856.128    tcb018gbwp7ttc
OAI31D1BWP7T           18      276.595    tcb018gbwp7ttc
OAI32D0BWP7T            4       70.246    tcb018gbwp7ttc
OAI32D1BWP7T           35      614.656    tcb018gbwp7ttc
OAI33D0BWP7T            1       19.757    tcb018gbwp7ttc
OAI33D1BWP7T            4       79.027    tcb018gbwp7ttc
OR2D1BWP7T            845     9274.720    tcb018gbwp7ttc
OR2D2BWP7T            142     2182.029    tcb018gbwp7ttc
OR2D4BWP7T             68     1791.283    tcb018gbwp7ttc
OR2D8BWP7T              3      125.126    tcb018gbwp7ttc
OR2XD1BWP7T            19      250.253    tcb018gbwp7ttc
OR3D0BWP7T              6       92.198    tcb018gbwp7ttc
OR3D1BWP7T             20      307.328    tcb018gbwp7ttc
OR3D2BWP7T              8      140.493    tcb018gbwp7ttc
OR3XD1BWP7T             5       76.832    tcb018gbwp7ttc
OR4D0BWP7T              2       35.123    tcb018gbwp7ttc
OR4D1BWP7T             33      579.533    tcb018gbwp7ttc
OR4XD1BWP7T             2       35.123    tcb018gbwp7ttc
SDFCNQD1BWP7T           7      430.259    tcb018gbwp7ttc
SDFKCNQD0BWP7T          1       61.466    tcb018gbwp7ttc
SDFKCNQD1BWP7T         28     1721.037    tcb018gbwp7ttc
SDFKCNQD2BWP7T          1       63.661    tcb018gbwp7ttc
SDFKSNQD1BWP7T         33     2173.248    tcb018gbwp7ttc
SDFKSNQD2BWP7T          2      136.102    tcb018gbwp7ttc
SDFQD0BWP7T            26     1369.805    tcb018gbwp7ttc
SDFXQD1BWP7T            1       74.637    tcb018gbwp7ttc
SEDFQD2BWP7T            1       72.442    tcb018gbwp7ttc
XNR2D1BWP7T           432     8534.938    tcb018gbwp7ttc
XNR2D2BWP7T            56     1475.174    tcb018gbwp7ttc
XNR3D0BWP7T            31     1156.870    tcb018gbwp7ttc
XNR3D1BWP7T           157     5858.989    tcb018gbwp7ttc
XNR3D2BWP7T            12      474.163    tcb018gbwp7ttc
XNR4D0BWP7T            82     4140.147    tcb018gbwp7ttc
XNR4D1BWP7T            14      706.854    tcb018gbwp7ttc
XOR2D0BWP7T            32      702.464    tcb018gbwp7ttc
XOR2D1BWP7T            87     1909.824    tcb018gbwp7ttc
XOR2D2BWP7T           188     4952.371    tcb018gbwp7ttc
XOR3D0BWP7T           184     6866.586    tcb018gbwp7ttc
XOR3D1BWP7T           571    21308.806    tcb018gbwp7ttc
XOR3D2BWP7T           168     6638.285    tcb018gbwp7ttc
XOR4D0BWP7T           129     6513.158    tcb018gbwp7ttc
XOR4D1BWP7T            41     2070.074    tcb018gbwp7ttc
---------------------------------------------------------
total              197560  5822842.637
```

C.4 Detailed Report of GSC 3.0v Cell Utilization for the FEC IP

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.20-p003_1
  Generated on:           Oct 31 2020  11:51:56 am
  Module:                 vlc_phy_fec_PA_MODE0
  Technology libraries:   gsclib
                          physical_cells
  Operating conditions:   typical
  Interconnect mode:      global
  Area mode:              physical library
============================================================


  Gate    Instances      Area      Library
--------------------------------------------
ADDFX1          772    149309.741    gsclib
ADDHX1          646     87796.051    gsclib
AND2X1         4134    129655.469    gsclib
AOI21X1        3204    100487.693    gsclib
AOI22X1       39461   1650164.314    gsclib
BUFX3          1748     54822.874    gsclib
CLKBUFX1       1168     30526.848    gsclib
CLKBUFX2         16       501.811    gsclib
CLKBUFX3        904     28352.333    gsclib
DFFSRX1        2231    361518.379    gsclib
DFFX1         72843  10661418.029    gsclib
INVX1          8588    179564.774    gsclib
INVX2          1058     27651.888    gsclib
INVX4          3266    102432.211    gsclib
INVX8           413     21588.336    gsclib
MX2X1         65127   4085182.253    gsclib
NAND2X1       17309    452388.024    gsclib
NAND2X2         343     12550.507    gsclib
NAND3X1        1641     60044.846    gsclib
NAND4X1        8887    371633.011    gsclib
NOR2X1        13360    349176.960    gsclib
NOR3X1          188     11792.563    gsclib
NOR4X1           35      3476.088    gsclib
OAI21X1        4525    189224.640    gsclib
OAI22X1         167     10475.309    gsclib
OAI33X1          63      5269.018    gsclib
OR2X1          1857     58241.462    gsclib
OR4X1          1258     92061.446    gsclib
XOR2X1        11000    632491.200    gsclib
--------------------------------------------
total        266212  19919798.078
```

C.5 Timing Report of the FEC IP Critical Path using TCB018GBWP7T

```
[
   fontsize=\small, %or \small or \footnotesize etc.
]
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.20-p003_1
  Generated on:           Nov 01 2020  02:23:57 pm
  Module:                 vlc_phy_fec_PA_MODE0
  Operating conditions:   NCCOM
  Interconnect mode:      global
  Area mode:              physical library
============================================================


Path 1: Setup Check with Pin .../RS_BM_INST/LOCATION_POLY_REGISTER/q_reg[7][4]/CP->D
```

```
          Group: clk
     Startpoint: (R) .../RS_BM_INST/SYNDROME_SHIFTER/r_array_reg[11][6]/CP
          Clock: (R) clk
       Endpoint: (F) .../RS_BM_INST/LOCATION_POLY_REGISTER/q_reg[7][4]/D
          Clock: (R) clk

                     Capture       Launch
        Clock Edge:+    4000            0
       Src Latency:+       0            0
       Net Latency:+       0 (I)        0 (I)
           Arrival:=    4000            0

             Setup:-     136
     Required Time:=    3464
      Launch Clock:-       0
         Data Path:-    3788
             Slack:=    +76

#---------------------------------------------------------------------------------------
#                  Timing Point                         Cell     Delay(ps) Arrival(ps)
#---------------------------------------------------------------------------------------
  .../RS_BM_INST/SYNDROME_SHIFTER/r_array_reg[11][6]/CP (arrival)       -       0
  .../RS_BM_INST/SYNDROME_SHIFTER/r_array_reg[11][6]/Q  DFKCNQD2BWP7T   267    267
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2439/ZN           INVD5BWP7T      86     352
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2434/ZN           INVD4BWP7T      57     409
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2292/ZN           OAI211D2BWP7T   94     504
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2240/ZN           ND2D3BWP7T      62     566
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2229/ZN           INVD2BWP7T      58     624
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2183/ZN           ND2D3BWP7T      54     678
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2150/ZN           ND2D3BWP7T      56     734
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2138/ZN           ND2D4BWP7T      47     782
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2125/ZN           ND2D6BWP7T      67     849
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2119/ZN           CKND6BWP7T      42     891
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2098/ZN           ND2D2BWP7T      65     956
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2080/ZN           ND2D4BWP7T      53     1009
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2074/ZN           CKND2BWP7T      56     1065
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2060/ZN           ND2D4BWP7T      45     1110
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2054/ZN           ND2D2P5BWP7T    64     1174
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2524/ZN           IND2D4BWP7T     47     1220
  .../RS_BM_INST/.../FULL_MULTIPLIER/g2049/ZN           ND2D5BWP7T      52     1273
  .../RS_BM_INST/REDUCE_ADDER/g3645/ZN                  IND2D4BWP7T     101    1374
  .../RS_BM_INST/REDUCE_ADDER/g3315/ZN                  ND2D6BWP7T      51     1424
  .../RS_BM_INST/REDUCE_ADDER/g3611/ZN                  IND2D4BWP7T     112    1536
  .../RS_BM_INST/REDUCE_ADDER/g3139/ZN                  ND2D8BWP7T      49     1585
  .../RS_BM_INST/REDUCE_ADDER/g3016/ZN                  ND3D4BWP7T      83     1668
  .../RS_BM_INST/REDUCE_ADDER/g3011/ZN                  ND2D5BWP7T      73     1741
  .../RS_BM_INST/REDUCE_ADDER/g3002/ZN                  ND2D6BWP7T      45     1786
  .../RS_BM_INST/REDUCE_ADDER/g2988/ZN                  ND2D6BWP7T      58     1844
  .../RS_BM_INST/g10362/ZN                              ND2D6BWP7T      53     1896
  .../RS_BM_INST/g10356/ZN                              INVD12BWP7T     89     1986
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3654/ZN           ND2D3BWP7T      55     2040
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3764/ZN           IND2D4BWP7T     112    2152
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3564/ZN           ND2D5BWP7T      69     2222
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3758/ZN           IND2D4BWP7T     110    2332
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3434/ZN           ND2D5BWP7T      45     2377
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3400/ZN           ND2D5BWP7T      48     2425
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3388/ZN           ND2D6BWP7T      44     2469
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3380/ZN           INVD4BWP7T      45     2514
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3368/ZN           ND2D5BWP7T      44     2558
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3352/ZN           CKND2D8BWP7T    92     2650
  .../RS_BM_INST/fopt10424/ZN                          CKND8BWP7T      72     2723
  .../RS_BM_INST/fopt10422/ZN                          INVD8BWP7T      89     2812
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3343/ZN          ND2D3BWP7T      63     2875
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3236/ZN          ND2D1BWP7T      128    3003
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3170/ZN          ND2D3BWP7T      58     3060
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3360/ZN          IND2D4BWP7T     105    3166
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3128/ZN          ND2D4BWP7T      62     3227
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3120/ZN          CKND4BWP7T      36     3263
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3090/ZN          ND2D2BWP7T      56     3319
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3075/ZN          ND2D3BWP7T      46     3365
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3061/Z           CKXOR2D4BWP7T   199    3563
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3353/ZN          IND2D4BWP7T     97     3660
  .../RS_BM_INST/.../FULL_MULTIPLIER/g3021/ZN          ND2D6BWP7T      41     3702
  .../RS_BM_INST/..LOCATTION_ADDER/g135/ZN             INVD3BWP7T      39     3741
  .../RS_BM_INST/..LOCATTION_ADDER/g113/ZN             OAI22D2BWP7T    47     3788
  .../RS_BM_INST/LOCATION_POLY_REGISTER/q_reg[7][4]/D   DFKCNQD2BWP7T   0      3788
#-----------------------------------------------------------------------------------------
```

ANNEX D – Communication performance data of FEC IP

Table 23 – Latency, data rate and throughput of all IEEE 802.15.7 operating modes for encoder and decoder portions on 5CGXFC5C6F27C7N (FPGA) and TCB018GBWP7T (ASIC)

MCS_ID	Opt. Clock (Mhz)/ Data Rate (Mbps)	SIFS Maximum Latency (us)	RIFS Maximum Latency (us)	Latency Encoder (us) FPGA	Latency Encoder (us) ASIC	Latency Decoder (us) FPGA	Latency Decoder (us) ASIC	Data Rate Encoder (Mbps) FPGA	Data Rate Encoder (Mbps) ASIC	Data Rate Decoder (Mbps) FPGA	Data Rate Decoder (Mbps) ASIC	Throughput Encoder (Mbps) FPGA	Throughput Encoder (Mbps) ASIC	Throughput Decoder (Mbps) FPGA	Throughput Decoder (Mbps) ASIC
0	0.2	600.00	200.00	54.88	17.28	126.30	39.78	320.00	1016.00	160.00	508.00	37.33	118.53	160.00	508.00
1	0.2	600.00	200.00	34.95	11.01	84.66	26.66	240.00	762.00	80.00	254.00	58.67	186.27	80.00	254.00
2	0.2	600.00	200.00	34.96	11.01	93.94	29.59	120.00	381.00	80.00	254.00	58.67	186.27	80.00	254.00
3	0.2	600.00	200.00	0.03	0.01	0.33	0.10	320.00	1016.00	80.00	254.00	234.67	745.07	80.00	254.00
4	0.2	600.00	200.00	0.00	0.00	0.00	0.00	640.00	2032.00	640.00	2032.00	640.00	2032.00	640.00	2032.00
5	0.4	300.00	100.00	0.03	0.01	0.49	0.16	320.00	1016.00	260.00	825.50	42.67	135.47	260.00	825.50
6	0.4	300.00	100.00	0.03	0.01	0.46	0.14	320.00	1016.00	220.00	698.50	85.33	270.93	220.00	698.50
7	0.4	300.00	100.00	0.03	0.01	0.40	0.13	320.00	1016.00	160.00	508.00	149.33	474.13	160.00	508.00
8	0.4	300.00	100.00	0.00	0.00	0.00	0.00	640.00	2032.00	640.00	2032.00	640.00	2032.00	640.00	2032.00
16	3.75	32.00	10.67	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
17	3.75	32.00	10.67	0.03	0.01	2.66	0.84	640.00	2032.00	80.00	254.00	512.00	1625.60	80.00	254.00
18	7.5	16.00	5.33	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
19	7.5	16.00	5.33	0.03	0.01	2.66	0.84	640.00	2032.00	80.00	254.00	512.00	1625.60	80.00	254.00
20	7.5	16.00	5.33	0.00	0.00	0.00	0.00	640.00	2032.00	640.00	2032.00	640.00	2032.00	640.00	2032.00
21	15	8.00	2.67	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
22	15	8.00	2.67	0.03	0.01	2.66	0.84	640.00	2032.00	80.00	254.00	512.00	1625.60	80.00	254.00
23	30	4.00	1.33	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
24	30	4.00	1.33	0.03	0.01	2.66	0.84	640.00	2032.00	80.00	254.00	512.00	1625.60	80.00	254.00
25	60	2.00	0.67	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
26	60	2.00	0.67	0.03	0.01	2.66	0.84	640.00	2032.00	80.00	254.00	512.00	1625.60	80.00	254.00
27	120	1.00	0.33	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
28	120	1.00	0.33	0.03	0.01	2.66	0.84	640.00	2032.00	80.00	254.00	512.00	1625.60	80.00	254.00
29	120	1.00	0.33	0.00	0.00	0.00	0.00	640.00	2032.00	640.00	2032.00	640.00	2032.00	640.00	2032.00
32	12	10.00	3.33	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
33	12	10.00	3.33	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
34	24	5.00	1.67	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
35	24	5.00	1.67	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
36	24	5.00	1.67	0.03	0.01	1.46	0.46	640.00	2032.00	80.00	254.00	320.00	1016.00	80.00	254.00
37	24	5.00	1.67	0.00	0.00	0.00	0.00	640.00	2032.00	640.00	2032.00	640.00	2032.00	640.00	2032.00
38	24	5.00	1.67	0.00	0.00	0.00	0.00	640.00	2032.00	640.00	2032.00	640.00	2032.00	640.00	2032.00