VLSI IMPLEMENTATION OF POLAR CODES USING VERILOG HDL

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering

by

KOLAVENNU VISWANADH VENKATA SAI(37130200)
K. LAXMI KANTHA RAO(37130203)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KOLAVENNU VISWANADH VENKATA SAI** (Register No: 37130200) and **K. LAXMI KANTHA RAO** (Register No: 37130203) who carried out the project entitled "**VLSI IMPLEMENTATION OF POLAR CODES USING VERILOG HDL**" under our supervision from December 2020 to April 2020.

Internal Guide
Dr. M.SUGADEV, M.Tech., Ph.D.,

Head of the Department
Dr. T.RAVI, M.E., Ph.D.,

Submitted for Viva voce examination held on _____

Internal Examiner **External Examiner**

DECLARATION

We **KOLAVENNU VISWANADH VENKATA SAI** (37130200) and **K. LAXMI KANTHA RAO** (37130203) hereby declare that the Project Report entitled "**VLSI IMPLEMENTATION OF POLAR CODES USING VERILOG HDL**", done by us under the guidance of **Dr. M.Sugadev, M.Tech., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

PLACE:

SIGNATURE OF CANDIDATES

1.

2.

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D. Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E., Ph.D. Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. M.SUGADEV, M.Tech., Ph.D., Assistant Professor, Department of Electronics and Communication Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

We wish to express our thanks to all teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

We express our gratitude to our parents for their constant encouragement and support for the completion of the project.

ABSTRACT

Polar codes are error correction codes that are widely used in communication systems. Polar codes exhibits high error correction capability as compared with other error correction codes. This paper proposes a Very Large Scale Integration (VLSI) architecture for the implementation of Polar decoder. Soft-in-soft out decoders, interleavers and deinterleavers is used in the decoder side which employs Maximum-a- Posteriori (MAP) algorithm. The number of iterations required to decode the information bits being transmitted is reduced by the use of MAP algorithm. For the encoder part, thispaper uses a system which contains two Recursive convolutional encoders along with pseudorandom interleaver in encoder side. The proposed system has designed by usingVerilog HDL and this design was simulated in Xilinx ISE 13.2. 5G peak mobile broadband data rates are expected to be around 20Gbps. Beyond-5G systems are expected to operate at Terabit/s data rates. Today's most advanced polar code implementations currently deliver only around 5Gbps. Therefore, Turbo codes and LDPC codes that played key enablers in 3G and 4G systems are already unproven for many new 5G applications. Polar code is believed as prominent breakthrough in 5G. It guarantees apical performance for 5G scenarios and hence it is considered as a promising candidate for the 5G New Radio. This work accentuates on the suitability of polar codes for the 5G scenarios.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

VLSI	VERY LARGE SCALE INTEGRATION
TTL	TRANSISTOR TRANSISTOR LOGIC
IC	INTEGRATED CIRCUIT
SSI	SMALL SCALE INTEGRATION
TTL	TRANSISTOR-TRANSISTOR LOGIC
ULSI	ULTRA LARGE SCALE INTEGRATION
ASIC	APPLICATION SPECIFIC INTEGRATED CIRCUIT
MSI	MEDIUM SCALE INTEGRATION
LSI	LARGE SCALE INTEGRATION
SRAM	STATIC RANDOM ACCESS MEMORY
GPU	GRAPHIC PROCESSING UNIT
HDL	HARDWARE DESCRIPTION LANGUAGE
CMOS	COMPLEMENTARY METAL OXIDE SEMICONDUCTOR
ASSP	APPLICATION SPECIFIC STANDARD PRODUCTS
SOC	SYSTEM ON A CHIP
FPGA	FIELD PROGRAMABLE GATE ARRAY
RSC	RECURSIVE CONVOLUTIONAL ENCODER
SISO	SERIAL IN SERIAL OUT
EDA	ELECTRONIC DESIGN AUTOMATION

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION OF VLSI:

VLSI Design presents state-of-the-art papers in VLSI design, computer-aided design, design analysis, design implementation, simulation and testing. Its scope also includes papers that address technical trends, pressing issues, and educational aspects in VLSI Design. The Journal provides a dynamic high-quality international forum for original papers and tutorials by academic, industrial, and other scholarly contributors in VLSI Design.

The development of microelectronics spans a time which is even lesser than the average life expectancy of a human, and yet it has seen as many as four generations. Early 60's saw the low density fabrication processes classified under Small Scale Integration (SSI) in which transistor count was limited to about 10. This rapidly gave way to Medium Scale Integration in the late 60's when around 100 transistors could be placed on a single chip.

It was the time when the cost of research began to decline and private firms started entering the competition in contrast to the earlier years where the main burden was borne by the military. Transistor-Transistor logic (TTL) offering higher integration densities outlasted other IC families like ECL and became the basis of the first integrated circuit revolution. It was the production of this family that gave impetus to semiconductor giants like Texas Instruments, Fairchild and National Semiconductors. Early seventies marked the growth of transistor count to about 1000 per chip called the Large Scale Integration.

By mid-eighties, the transistor count on a single chip had already exceeded 1000 and hence came the age of Very Large Scale Integration or VLSI. Though many improvements have been made and the transistor count is still rising, further names of generations like ULSI are generally avoided. It was during this time when TTL lost the

battle to MOS family owing to the same problems that had pushed vacuum tubes into negligence, power dissipation and the limit it imposed on the number of gates that could be placed on a single die.

The second age of Integrated Circuits revolution started with the introduction of thefirst microprocessor, the 4004 by Intel in 1972 and the 8080 in 1974. Today many companies like Texas Instruments, Infineon, Alliance Semiconductors, Cadence, Synopsys, Celox Networks, Cisco, Micron Tech, National Semiconductors, STMicroelectronics, Qualcomm, Lucent, Mentor Graphics, Analog Devices, Intel, Philips, Motorola and many other firms have been established and are dedicated to the various fields in "VLSI" like Programmable Logic Devices, Hardware Descriptive Languages, Design tools, Embedded Systems etc.

In 1980s, hold-over from outdated taxonomy for integration levels obviouslyinfluenced from frequency bands, i.e. HF, VHF, and UHF. Sources disagree on what is measured (gates or transistors)

SSI – Small-Scale Integration (0-102)

MSI – Medium-Scale Integration (102 -103)

LSI – Large-Scale Integration (103 -105)

VLSI – Very Large-Scale Integration (105 - 107)

ULSI – Ultra Large-Scale Integration (>= 107)

VLSI Technology, Inc. was a company which designed and manufactured custom and semi-custom ICs. The company was based in Silicon Valley, with headquarters at 1109 McKay Drive in San Jose, California. Along with LSI Logic, VLSI Technology defined the leading edge of the application-specific integrated circuit (ASIC) business, which accelerated the push of powerful embedded systems into affordable products. The company was founded in 1979 by a trio from Fairchild Semiconductor by way of Synertek - Jack Balletto, Dan Floyd, and Gunnar Wetlesen - and by Doug Fairbairn of Xerox PARC and Lambda (later VLSI Design) magazine.

Alfred J. Stein became the CEO of the company in 1982. Subsequently VLSI built its first fab in San Jose; eventually a second fab was built in San Antonio, Texas. VLSI had its initial public offering in 1983, and was listed on the stock market as (NASDAQ: VLSI). The company was later acquired by Philips and survives to this day as part of NXP Semiconductors.

The first semiconductor chips held two transistors each. Subsequent advances added more and more transistors, and, as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now retrospectively as small-scale integration (SSI), improvements in technique led to devices with hundreds of logic gates, known as medium-scale integration (MSI). Further improvements led to large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and billions of individual transistors.

At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like ultra-large-scale integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use.

As of early 2008, billion-transistor processors are commercially available. This is expected to become more commonplace as semiconductor fabrication moves from the current generation of 65 nm processes to the next 45 nm generations (while experiencing new challenges such as increased variation across process corners). A notable example is NVidia’s 280 series GPU. This GPU is unique in the fact that almost all of its 1.4 billion transistors are used for logic, in contrast to the Itanium, whose large transistor count is largely due to its 24 MB L3 cache. Current designs, as opposed to the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM (Static Random Access Memory) cell, however,

are still designed by hand to ensure the highest efficiency (sometimes by bending or breaking established design rules to obtain the last bit of performance by trading stability)[citation needed]. VLSI technology is moving towards radical level miniaturization with introduction of NEMS technology. A lot of problems need to be sorted out before the transition is actually made.

1.2 WHY VLSI?

Integration improves the design, lowers the parasites, which means higher speed and lower power consumption and physically smaller. The Integration reduces manufacturing cost - (almost) no manual assembly.

The course will cover basic theory and techniques of digital VLSI design in CMOS technology. Topics include: CMOS devices and circuits, fabrication processes, static and dynamic logic structures, chip layout, simulation and testing, low power techniques, design tools and methodologies, VLSI architecture. We use full-custom techniques to design basic cells and regular structures such as data-path and memory. There is an emphasis on modern design issues in interconnect and clocking. We will also use several case-studies to explore recent real-world VLSI designs (e.g. Pentium, Alpha, PowerPC Strong ARM, etc.) and papers from the recent research literature. On-campus students will design small test circuits using various CAD tools. Circuits will be verified and analyzed for performance with various simulators. Some final project designs will be fabricated and returned to students the following semester for testing.

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors.

The first semiconductor chips held one transistor each. Subsequent advances added more and more transistors, and, as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known retrospectively as "small-scale integration" (SSI), improvements in technique led to devices with hundreds of logic gates, known as large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and hundreds of millions of individual transistors.

At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like Ultra-large-scale Integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use. Even VLSI is now somewhat quaint, given the common assumption that all microprocessors are VLSI or better.

As of early 2008, billion-transistor processors are commercially available, an example of which is Intel's Montecito Itanium chip. This is expected to become more commonplace as semiconductor fabrication moves from the current generation of 65 nm processes to the next 45 nm generations (while experiencing new challenges such as increased variation across process corners). Another notable example is NVIDIA’s 280 series GPU. This microprocessor is unique in the fact that its 1.4 Billion transistor count, capable of a teraflop of performance, is almost entirely dedicated to logic (Itanium's transistor count is largely due to the 24MB L3 cache). Current designs, as opposed to the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM cell, however, are still designed by hand to ensure the highest efficiency (sometimes by bending or breaking established design rules to obtain the last bit of performance by trading stability).

The original business plan was to be a contract wafer fabrication company, but the venture investors wanted the company to develop IC (Integrated Circuit) design tools to help fill the foundry.

Thanks to its Caltech and UC Berkeley students, VLSI was an important pioneer in the electronic design automation industry. It offered a sophisticated package of tools, originally based on the 'lambda-based' design style advocated by Carver Mead and Lynn Conway.VLSI became an early vendor of standard cell (cell-based technology) to the merchant market in the early 80s where the other ASIC-focused company, LSI Logic, was a leader in gate arrays. Prior to VLSI's cell-based offering, the technology had been primarily available only within large vertically integrated companies with semiconductor units such as AT&T and IBM.

VLSI's design tools eventually included not only design entry and simulation but eventually cell-based routing (chip compiler), a data-path compiler, SRAM and ROM compilers and a state machine compiler. The tools were an integrated design solution for IC design and not just point tools, or more general purpose system tools. A designer could edit transistor-level polygons and/or logic schematics, then run DRC and LVS, extract parasites from the layout and run Spice simulation, then back-annotate the timing or gate size changes into the logic schematic database. Characterization tools were integrated to generate Frame Maker Data Sheets for Libraries. VLSI eventually spun off the CAD and Library operation into Compass Design Automation but it never reached IPO before it was purchased by Avanti Corp.

VLSI's physical design tools were critical not only to its ASIC business, but also in setting the bar for the commercial EDA industry. When VLSI and its main ASIC competitor, LSI Logic, were establishing the ASIC industry, commercially-available tools could not deliver the productivity necessary to support the physical design of hundreds of ASIC designs each year without the deployment of a substantial number of layout engineers. The companies' development of automated layout tools was a rational "make because there's nothing to buy" decision. The EDA industry finally caught up in the late 1980s when Tangent Systems released its Tan-Cell and Tan-Gate products. In 1989, Tangent was acquired by Cadence Design Systems (founded in 1988).

Unfortunately, for all VLSI's initial competence in design tools, they were not leaders in semiconductor manufacturing technology. VLSI had not been timely in developing a 1.0 µm manufacturing process as the rest of the industry moved to that geometry in the late 80s. VLSI entered a long-term technology partnership with Hitachi and finally released a 1.0 µm process and cell library (actually more of a 1.2 µm library with a 1.0 µm gate).

As VLSI struggled to gain parity with the rest of the industry in semiconductor technology, the design flow was moving rapidly to a Verilog HDL and synthesis flow. Cadence acquired Gateway, the leader in Verilog hardware design language (HDL) and Synopsys was dominating the exploding field of design synthesis. As VLSI's tools were being eclipsed, VLSI waited too long to open the tools up to other fabrications and Compass Design Automation was never a viable competitor to industry leaders. Meanwhile, VLSI entered the merchant high speed static RAM (SRAM) market as they needed a product to drive the semiconductor process technology development. All the large semiconductor companies built high speed SRAMs with cost structures VLSI could never match. VLSI withdrew once it was clear that the Hitachi process technology partnership was working.

ARM Ltd was formed in 1990 as a semiconductor intellectual property licensor, backed by Acorn, Apple and VLSI. VLSI became a licensee of the powerful ARM processor and ARM finally funded processor tools. Initial adoption of the ARM processor was slow. Few applications could justify the overhead of an embedded 32 bit processor. In fact, despite the addition of further licensees, the ARM processor enjoyed little market success until they developed the novel 'thumb' extensions. Ericsson adopted the ARM processor in a VLSI chipset for its GSM handset designs in the early 1990s. It was the GSM boost that is the foundation of ARM the company/technology that it is today.

Only in PC chipsets, did VLSI dominate in the early 90s. This product was developed by five engineers using the 'Mega cells" in the VLSI library that led to a business unit at VLSI that almost equaled its ASIC business in revenue. VLSI eventually ceded the market to Intel because Intel was able to package-sell its processors, chipsets, and even board level products together. VLSI also had an early partnership with PMC, a

design group that had been nurtured of British Columbia Bell. When PMC wanted to divest its semiconductor intellectual property venture, VLSI's bid was beaten by a creative deal by Sierra Semiconductor. The telecom business unit management at VLSI opted to go it alone. PMC Sierra became one of the most important telecom ASSP vendors.

Scientists and innovations from the 'design technology' part of VLSI found their way to Cadence Design Systems (by way of Redwood Design Automation). Compass Design Automation (VLSI's CAD and Library spin-off) was sold to Avant! Corporation, which itself was acquired by Synopsys.

1.3 STRUCTURED DESIGN

Structured VLSI design is a modular methodology originated by Carver Mead and Lynn Conway for saving microchip area by minimizing the interconnect fabrics area. This is obtained by repetitive arrangement of rectangular macro blocks which can be interconnected using wiring by abutment. An example is partitioning the layout of an adder into a row of equal bit slices cells. In complex designs this structuring may be achieved by hierarchical nesting.

Structured VLSI design had been popular in the early 1980s, but lost its popularity later because of the advent of placement and routing tools wasting a lot of area by routing, which is tolerated because of the progress of Moore's Law. When introducing the hardware description language KARL in the mid' 1970s, Reiner Hartenstein coined the term "structured VLSI design" (originally as "structured LSI design"), echoing Edsger Dijkstra's structured programming approach by procedure nesting to avoid chaotic spaghetti-structured programs.

1.4 WHAT IS VLSI?

VLSI stands for "Very Large Scale Integration". This is the field which involves packing more and more logic devices into smaller and smaller areas.

- Simply we say Integrated circuit is many transistors on one chip.

- Design/manufacturing of extremely small, complex circuitry using modified semiconductor material
- Integrated circuit (IC) may contain millions of transistors, each a few mm in size
- Applications wide ranging: most electronic logic devices

In olden days, when huge computers made of vacuum tubes could occupy an entire dedicated rooms and could do about 360 multiplications of 10 digit numbers in a second. Modern day computers are getting smaller, faster, and cheaper and more power efficient for every progressing second. The electronic miniaturizing started when the occurrence of semiconductor transistor by Bardeen (1947-48) and then the Bipolar Transistor by Shockley (1949) in the Bell Laboratory.

The first IC (Integrated Circuit) was invented by Jack Kilby in 1958, in the form of a Flip Flop our ability to pack more and more transistors onto a single chip has doubled roughly every 18 months, in accordance with the Moore's Law. Such exponential or increasing development had never been seen in any other field and still it is continuing in major areas of research work.

1.5 HISTORY & EVOLUTION

The development of microelectronics spans a time which is even lesser than the average life expectancy of a human, and yet it has seen as many as four generations. In early 60's, the low density fabrication processes classified under Small Scale Integration (SSI) in which transistor count was limited to about 10. This rapidly gave way to Medium Scale Integration in the late 60's when around 100 transistors could be placed on a single chip. It was the time when the cost of research began to decline and private firms started entering the competition. Transistor-Transistor logic (TTL) offering higher integration densities than other IC families like ECL and became the basis of the first integrated circuit revolution. Early seventies marked the growth of transistor count to about 1000 perchip called the Large Scale Integration. By mid-eighties, the transistor count on a single

chip had already exceeded 1000 and hence came the age of Very Large Scale Integration or VLSI. Though many improvements have been made and the transistor count is still rising, further names of generations like ULSI are generally avoided.

The second age of Integrated Circuits revolution started with the introduction of the first microprocessor, the 4004 by Intel in 1972 and the 8080 in 1974. Devices, Intel, Philips, Motorola and many other firms have been established and are dedicated to the various fields in "VLSI" like Programmable Logic Devices, Hardware Descriptive Languages, Design tools, Embedded Systems etc.

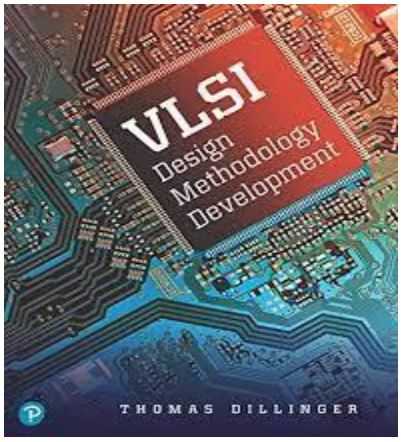
1.6 FUTURE OF VLSI

Generally, VLSI technology is used in the devices like computers, cell phones, digital cameras and any electronic gadget. There are certain key issues that serve as active areas of research and are constantly improving as the field continues to mature. VLSI is dominated by the CMOS technology and much like other logic families, this too has its limitations which have been battled and improved upon since years. By taking the example of a processor, the process technology has rapidly shrunk from 180 nm in 1999 to 60nm in 2008 and now it stands at 45nm and attempts are being made to reduce it for 32nm. As the number of transistors increase, the power dissipation is increasing and also the noise. Heat is generated per unit area. New alternatives like Gallium Arsenide technology are becoming an active area of research, future of VLSI seems to change for every little moment.

History of Scale Integration

- Late 40s Transistor invented at Bell Labs
- Late 50s First IC (JK-FF by Jack Kilby at TI)
- Early 60s Small Scale Integration (SSI)

- 10s of transistors on a chip
- Late 60s Medium Scale Integration (MSI)
- 100s of transistors on a chip
- Early 70s Large Scale Integration (LSI)
- 1000s of transistor on a chip
- Early 80s VLSI 10,000s of transistors on a Chip (later 100,000s & now 1,000,000s)
- Ultra LSI is sometimes used for 1,000,000s
- SSI - Small-Scale Integration (0-102)
- MSI - Medium-Scale Integration (102-103)
- LSI - Large-Scale Integration (103-105)
- VLSI - Very Large-Scale Integration (105-107)



ULSI - Ultra Large-Scale Integration (>=107)

1.6.1 System Design

Create a high-level (Behavioral) representation of your

systemTools: Verilog, VHDL, System C

- Synthesizable (PLD's and/or ASIC)
- Non-synthesizable
- Moraine future lectures

1.6.2 Switches

Digital equipment is largely composed of switches are

1. Switches can be built from many technologies

2. Relays (from which the earliest computers were built)

3. Thermionic valves

4. Transistors

The perfect digital switch would have the following

1. Switch instantly

2. Use no power

3. Have an infinite resistance when off and zero resistance when on

1.7 SEMICONDUCTORS AND DOPING

By adding trace amounts of certain materials to semiconductors alters the crystal structure and can change their electrical properties in particular it can change the number of free electrons or holes. N-Type semiconductor has free electrons and dopant is (typically) phosphorus, arsenic, antimony. P-Type semiconductor has free holes dopant is (typically) boron, indium, and gallium. Dopants are usually implanted into the semiconductor using Implant Technology, followed by thermal process to diffuse the dopants. IC Technology is mainly used for Speed / Power performance of available technologies and the microelectronics evolution and SIA Roadmap and Semiconductor Manufacturers 2001 Ranking.

Metal-Oxide-Semiconductor(MOS) and related VLSI technology PMOS, NMOS, CMOS, Bi CMOS, GaAs.

Basic MOS Transistors are implemented as minimum line width, transistor cross section, Charge inversion channel, Source connected to substrate, Enhancement vs. Depletion mode devices, PMOS are 2.5 time slower than NMOS due to electron and holemotilities.

1.7.1 Fabrication Technology

It is Silicon of extremely high purity and chemically purified then grown into large crystals. Wafer is type of crystal which is sliced into substrates, and its diameter is currently 150mm, 200mm, 300mm and wafer thickness <1mm and also surface is polished to optical smoothness. Wafer is then ready for processing. Each wafer will yield many chips and the chip die size varies from about 5mmx5mm to 15mmx15mm.A whole

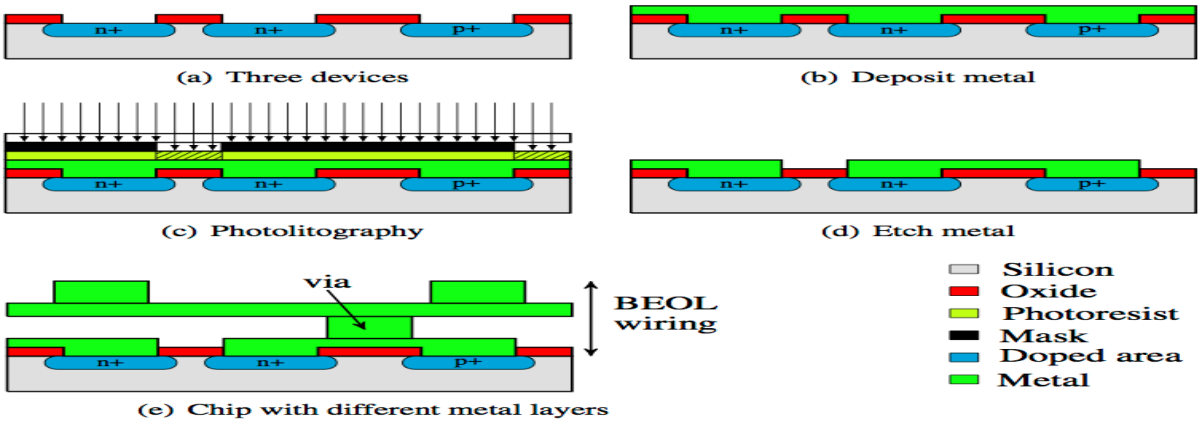


Fig 1.7.1(a) fabrication

wafer is processed at a time. Different parts of each die will be made P-type or N-type (small amount of other atoms intentionally introduced - doping -implant). Interconnections are made with metal insulation and material used is typically SiO2, SiN. New materials are being investigated (low-k dielectrics).In CMOS Fabrication p-well process, n-well process and twin-tub process all the devices on the wafer are made at the same time. After the circuitry has been placed on the chip, the chip is over-glassed (with a passivation layer) to protect it only those areas which connect to the outside world will be left uncovered (the pads). The wafer finally passes to a test station test probes send test signal patterns to the chip and monitor the output of the chip. The yield of a process is the percentage of die which pass this testing,the wafer is then scribed and separated up into the individual chips. These are then packaged and Chips are 'binned' according to their performance.

1.7.2 Conventional Approach to Digital Design

Digital ICs of SSI and MSI types have become universally standardized and have been accepted for use. Whenever a designer has to realize a digital function, he uses a standard set of ICs along with a minimal set of additional discrete circuitry.

1.7.3 Design of VLSI

The complexity of VLSI is being designed and used today, which makes the manual approach to be impractical. Design automation is the order of the day. With the rapid technological developments in the last two decades, the status of VLSI technology

is characterized by the following

1. A steady increase in the size and hence the functionality of the ICs.

2. A steady reduction in feature size and hence increase in the speed of operation as well as gate or transistor density.

3. A steady improvement in the predictability of circuit behavior.

4. A steady increase in the variety and size of software tools for VLSI design.

The above developments have resulted in a proliferation of approaches to VLSI design. We briefly describe the procedure of automated design flow. The aim is more to bring out the role of a Hardware Description Language (HDL) in the design process. An abstraction based model is the basis of the automated design.

The model divides the whole design cycle into various domains. With such an abstraction through a division process the design is carried out in different layers. The designer at one layer can function without bothering about the layers above or below. The thick horizontal lines separating the layers in the figure signify the compartmentalization. As an example, let us consider design at the gate level. The circuit to be designed would be described in terms of truth-tables and state tables. With these as available inputs, he has to express them as Boolean logic equations and realize them in terms of gates and flip-flops. In turn, these form the inputs to the layer immediately below. Compartmentalization of the approach to design in the manner described here is the essence of abstraction; it is the basis for development and use of CAD tools in VLSI design at various levels.

The design methods at different levels use the respective aids such as Boolean equations, truth tables, state transition table, etc. But the aids play only a small role in the process. To complete a design, one may have to switch from one tool to another, raising the issues of tool compatibility and learning new environments.

1.8 VLSI AND SYSTEMS

These advantages of integrated circuits translate into advantages at the system level are

- **Smaller physical size**: Smallness is often an advantage in itself-consider portable televisions or handheld cellular telephones.
- **Lower power consumption**: Replacing a handful of standard parts with a single chip reduces total power consumption. Reducing power consumption has a ripple effect on the rest of the system: a smaller, cheaper power supply can be used; since less power consumption means less heat, a fan may no longer be necessary; a simpler cabinet with less shielding for electromagnetic shielding may be feasible, too.
- **Reduced cost:** Reducing the number of components, the power supply requirements, cabinet costs, and so on, will inevitably reduce system cost. The ripple effect of integration is such that the cost of a system built from custom ICs can be less, even though the individual ICs cost more than the standard parts they replace.

Understanding why integrated circuit technology has such profound influence on the design of digital systems requires understanding both the technology of IC manufacturing and the economics of ICs and digital systems.

1.9 APPLICATIONS OF VLSI

VLSI having applications in various domains such as electronic systems, medical, communication, digital signal processing. Some of them are listed as below

1.9.1 Electronic system in cars

Electronic systems now perform a wide variety of tasks in daily life. Electronic systems in some cases have replaced mechanisms that operated mechanically, hydraulically, or by other means; electronics are usually smaller, more flexible, and easier to service. In other cases electronic systems have created totally new applications. Electronic systems perform a variety of tasks, some of them are visible while some are hidden. Personal entertainment systems such as portable MP3 players and DVD players perform sophisticated algorithms with remarkably little energy.

1.9.2 Digital electronics control VCRs

Digital electronics compress and decompress video, even at high-definition data rates, on-the-fly in consumer electronics. Low-cost terminals for Web browsing still requiresophisticated electronics, despite their dedicated function.

1.9.3 Transaction processing system, ATM

Electronic systems in cars operate stereo systems and displays; they also control fuel injection systems, adjust suspensions to varying terrain, and perform the control functions required for anti-lock braking systems.

1.9.4 Personal computers and Workstations

Personal computers and workstations provide word-processing, financial analysis, and games. Computers include both central processing units and special-purpose hardware for disk access, faster screen display, etc.

1.9. 5 Medical electronic systems

Medical electronic systems measure bodily functions and perform complex processing algorithms to warn about unusual conditions. The availability of these complex systems, far from overwhelming consumers, only creates demand for even more complex systems.

1.20 ASIC

An Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use. For example, a chip designed solely to run a cell phone is an ASIC. Intermediate between ASICs and industry standard integrated circuits, like the 7400 or the 4000 series, are application specific standard products (ASSPs).

As feature sizes have shrunk and design tools improved over the years, the maximum complexity (and hence functionality) possible in an ASIC has grown from 5,000 gates to over 100 million. Modern ASICs often include entire 32-bit processors, memory blocks including ROM, RAM, EEPROM, Flash and other large building blocks. Such an ASIC is often termed a SOC (system-on-a-chip). Designers of digital ASICs use a

hardware description language (HDL), such as Verilog or VHDL, to describe the functionality of ASICs.

Field-programmable gate arrays (FPGA) are the modern-day technology for building a breadboard or prototype from standard parts; programmable logic blocks and programmable interconnects allow the same FPGA to be used in many different applications. For smaller designs and/or lower production volumes, FPGAs may be more cost effective than an ASIC design even in production.

1.20.1 ASIC Design flow

As with any other technical activity, development of an ASIC starts with an idea and takes tangible shape through the stages of development as shown in Fig 1 and in Fig 2. The first step in this process is to expand the idea in terms of behavior of the target circuit.

The design is tested through a simulation process; it is to check, verify, and ensure that what is wanted is what is described. Simulation is carried out through dedicated tools. With every simulation run, the simulation results are studied to identify errors in the design description. The errors are corrected and another simulation run carried out. Simulation and changes to design description together form a cyclic iterative process, repeated until an error-free design is evolved.

Design description is an activity independent of the target technology or manufacturer. It results in a description of the digital circuit. To translate it into a tangible circuit, one goes through the physical design process. The same constitutes a set of activities closely linked to the manufacturer and the target technology.

CHAPTER 2

LITERATURE SURVEY

Technology has been changing rapidly in our day-to-day life. The improvement of microelectronics traverses a period which is much lesser than the normal future of a human, but it has seen upwards of four ages. Mid 60's saw the low thickness creation measures ordered under Small Scale Integration (SSI) in which semiconductor tally was restricted to around 10. This quickly offered approach to Medium Scale Integration in the last part of the 60's when around 100 semiconductors could be set on a solitary chip.

Xin-Yu Shih et al. (2016) has presented a paper on LEGO based VLSI design and implementation of polar codes encoder architecture with radix-2 processing engines. This paper has proposed work on polar codes become another channel coding, which will be regular to apply for cutting edge remote MIMO correspondence frameworks. In this work, we propose LEGO-based VLSI equipment plan and execution of the Polar encoder utilizing radix-2 preparing motors, which highlights low territory cost, low force dispersal, high velocity, and high throughput by means of sequential calculation. Under TSMC 90nm CMOS innovation, the 16384-point LEGO-based radix-2 Polar encoder chip (LB-R2-PE) is planned and orchestrated with complete space of 0.244mm 2 and force scattering of 366.6mW, working at greatest recurrence of 2.0GHz. In the APR chip execution perspective, the 16384-point LB-R2-PE chip just involves 0.305mm 2 and devours 357.8mW with greatest working recurrence of 1.61GHz, conveying complete throughput of 1.61Gbps.

Rahul Shrestha er al. (2019) has presented a paper on High-Throughput and High-Speed Polar-Decoder VLSI-Architecture for 5G New Radio. This paper has proposed a work on a new technique for computing logarithmic-likelihood-ratio (LLR) messages in the processing element (PE) unit of belief-propagation polar decoder that is based on two's complement representation of LLR values. In addition, a new PE-architecture corresponding to this technique has been presented that consumes lesser hardware and has shorter critical-path delay resulting in higher

clock frequency as well as throughput. We have incorporated these PE units in the design of single-column based unidirectional belief-propagation polar-decoder using the round-trip scheduling for decoding the polar code of 1024 code-length and 1/2 code rate. Performance analysis of such decoding algorithm in AWGN channel environment has been carried out and it delivered an adequate bit-error-rate of 10^{-4} at 4.2 dB of signal-to-noise ratio. VLSI architecture of the suggested polar-decoder is ASIC synthesized and post-layout simulated in 90 nm and 65 nm CMOS processes using industry standard EDA tools.

Kavi Priya et al. (2019) has presented a paper on High Speed Polar Encoder Architecture For next Generation 5G Applications Using Radix-k Processing Engine. This paper has proposed a work on the various encoding techniques which is effective, polar encoding is found to be the best because of its unique characteristics of channel achieving property. The coding scheme chosen by Third Generation Partnership Project (3GPP) is polar code since it has attracted the attention towards the 5 th generation wireless system. The emerging research is going on the polar codes in the application of MIMO system. Polar encoder is mainly designed because of its high processing speed as compared with other techniques. The VLSI implementation process can be extended to a large extent by using radix-k based design.

2.1 EXISTING WORK

Turbo codes are error correction codes that are widely used in communication systems. Turbo codes exhibits high error correction capability as compared with other error correction codes. This paper proposes a Very Large Scale Integration (VLSI) architecture for the implementation of Turbo decoder. Soft-in-soft out decoders, interleavers and deinterleavers is used in the decoder side which employs Maximum-a-Posteriori (MAP) algorithm. The number of iterations required to decode the information bits being transmitted is reduced by the use of MAP algorithm. For the encoder part, this paper uses a system which contains two Recursive convolutional encoders along with pseudorandom interleaver in encoder side. The Turbo encoding and decoding is done using Octave, Xilinx Vivado, Cadence tools. The system is implemented and synthesized in Application Specific Integrated Circuit (ASIC).Timing analysis has been done and GDSII file has been generated.

Turbo encoding and decoding is done using Verilog HDL. The decoder is developed based on MAP algorithm. Layout of Turbo decoder algorithm identifies the most probable bit of information that was sent.

Original MAP algorithm is used because the number of iterations for process decoding is reduced. Turbo codes and LDPC codes that played key enablers in 3G and 4G LTE systems.

CHAPTER-3

PROPOSED WORK

In a communication system, when data is transferred from the source system to a destination system, errors can be present in the received signal at the source end. So error correction is required to retrieve the original message. Polar codes, which were first introduced in 1993, represent a quantum leap in channel coding techniques and a turning point for modern digital telecommunication. Polar codes is one of existing powerful error correcting codes. Polar codes has inspired the coding community with the possibility of using an iterative decoding technique that relies solely on simple constituent code to achieve close channel capacity. Polar coder architecture comprises of polar encoder and polar decoder. Encoder consists of two Recursive Convolutional Encoders(RSC) and interleaver. In this paper, pseudo-random interleaver is used due to which the interleaved version of the code tends to be long and scrambled, that gives good performance of random codes. In polar code implementation, RSC encoders are employed rather than conventional convolutional encoders since it generates low weight parity codes. MAP algorithm is used for the decoding of polar encoded data in which errors are intentionally added and verified an error free decoded data after decoding.

3.1 INTRODUCTION

Polar codes were introduced by Erdal Arıkan in 2008. They are the first family of error-correcting codes that attain the capacity of binary-input memoryless and symmetric channels with efficient encoding, decoding, and construction algorithms. Since their introduction, polar codes have been generalized and shown to be capacity achieving in numerous other communications settings.

The original construction of polar codes relies on the recursive application of an invertible linear transformation, which, when combined with a successive-cancellation decoder, effectively splits the original binary-input memoryless and symmetric communication channel into a number of bit subchannels. Increasing the recursion depth causes these bit subchannels to converge either to noiseless or purely noisy channels.

Virtually error-free transmission can be achieved by sending the data over noiseless subchannels. While related code constructions had been suggested before (e.g., N. Stolte, I. Dumer and K. Shabunov), Arıkan's work was the first to prove the polarization phenomenon and thus prove that polar codes are capacity achieving.

Unfortunately, the subchannels converge to these limiting cases relatively slowly, meaning that the error-correcting performance of Arıkan's polar codes improves more slowly with the blocklength than other widely-used codes, such as Turbo and LDPC codes. However, polar codes have been shown to provide excellent error-correcting performance with low decoding complexity for practical blocklengths when combined with more advanced decoding algorithms. These favorable traits have led to polar codes being used in the 5G wireless standard, which is a testament to their outstanding performance.

In this Best Readings, we summarize several papers on the theoretical foundations of polarization theory, the construction and decoding of practical polar codes, as well as some generalized polar codes, which can help to overcome limitations of classical Arıkan polar codes. We also focus on practical implementation issues because, despite the simple structure of the encoding and decoding algorithms of polar codes, their practical implementation poses numerous challenges.

Polar codes [1] are constructed from the generator matrix $G^{\otimes M}_2$ with $G_2 = [\,1\ 1\ 0\ 1\,]$, where $\otimes M$ denotes the Mth Kronecker power. It has been shown in [1], that the synthesized channels seen by individual bits approach two extremes, either a noiseless channel or a pure-noise channel, as the block length $N = 2^M$ grows large. The fraction of noiseless channels is close to the channel capacity. Therefore, the noiseless channels, termed unfrozen bit channels, are selected for transmitting message bits while the other channels, termed frozen bit channels, are set to fixed values known by both encoder and decoder. Therefore, polar codes are the first family of codes that achieve the capacity of symmetric binary-input discrete memoryless channels under a lowcomplexity successive cancellation (SC) decoding algorithm as the block length N approaches infinity. However, the performance of polar codes at short to moderate block lengths is disappointing under the SC decoding algorithm. Later, a successive cancellation list (SCL) decoding algorithm for polar codes was proposed [2], which approaches the performance of the maximum-likelihood (ML) decoder as the list size L is large. However, the performance levels of polar codes are still inferior to those of low-density parity-check (LDPC) codes even under

the ML decoder. To strengthen polar codes, a serial concatenation of a cyclic redundancy check (CRC) code and a polar code, termed the CRC-aided polar code, was found to be effective to improve the performance under the SCL decoding algorithm [2]. The performance levels of CRC-aided polar codes under the SCL decoding algorithm are better than those of LDPC and turbo codes [2], [3]. As the SCL decoder is capable to achieve the ML performance, it is important to study the block error rate (BLER) of polar codes under the ML decoder. However, in the literature, there are no analytical results regarding the ML performance of polar codes. The BLERs of polar codes rely on simulations that are time-consuming. A possible way to analyze the BLER performance of a coding scheme is to use the BLER upper bound which is a function of the weight enumerating function (WEF) as that used to analyze turbo codes [4]. However, if the code size is large, obtaining the exact WEF of a polar code with the heuristic method is prohibitively complex. Approximations of WEFs of polar codes are proposed in [5], [6] based on the probabilistic weight distribution (PWD) [7]. In this paper, we propose to randomize the polar code using interleavers between the inter-3 mediate stages of the polar code encoder. Codes constructed on the basis of this idea are called interleaved polar (i-polar) codes. The ensemble of i-polar codes is formed by considering all possible interleavers. The regular polar code is just one realization of the ensemble of i-polar codes. Based on the concept of uniform interleaver, i.e., all interleavers are selected uniformly at random from all possible permutations, the average WEF of a code selected at random from the ensemble of i-polar codes can be evaluated. The concept of uniform interleaver has also been used in the analysis of turbo codes [4]. Note that the WEF analysis in this paper is not an approximation to the WEF of a polar code, but is an exact WEF averaged over the ensemble of i-polar codes. Based on the average WEF, a BLER upper bound, termed simple bound [8], can be used to evaluate the BLER performance averaged over the ensemble of codes. Simulation results show that the BLER upper bounds can well predict the ML performance levels of i-polar codes at high SNRs. Also, we will show by simulations that a specific realization of i-polar codes outperforms a regular polar code under the SCL decoder of the same list size. We also propose a concatenated coding scheme that employs P identical high rate codes as the outer code and Q identical i-polar codes as the inner code with an interleaver in between. CRC codes are the most popular outer codes employed in the concatenation of polar codes.

We propose as an alternative to use systematic regular repeat-accumulate (RRA) codes or irregular repeataccumulate (IRA) codes [9] as the outer component code. The average WEF of the concatenated code is derived based on the uniform interleaver assumption. Simulation results show that the BLER upper bounds can well predict the BLER performance levels of the concatenated codes. One advantage of the proposed concatenated code is that, for Q > 1, the code can be decoded using Q SCL decoders working in parallel which can significantly reduce the decoding latency when Q is large. Analytical and simulation results both show that the performance of the proposed concatenated code with P = Q = 2 is better than that of the CRC-aided i-polar code with P = Q = 1 of the same length and code rate at high SNRs. Therefore, the proposed coding scheme is suitable for ultra-reliable low-latency communications (URLLC) [10]. The rest of the paper is organized as follows. We begin with a brief introduction of polar codes in Section II. The construction of i-polar codes is presented in Section III. Section IV presents the WEF and IOWEF analysis of i-polar codes. In Section V, a concatenated coding scheme with the i-polar code as the inner component code is proposed and the WEF of the concatenated code is presented. Analytical and simulation results are given in Section VI. Finally, conclusions are given in Section VII.

Polar codes are error-correcting codes, which are able to achieve the capacity of binary-input memoryless symmetric (BMS) channels. This means that one can transmit at the highest possible rate over that class of channels. In addition, the encoding and decoding operations can be performed with low complexity, thanks to recursive techniques.

Formally, a specific polar code is fully defined by a 4-tuple (N, R, A, uAuA) where:

- N is the block length, i.e. the total number of bits transmitted over the channel.
- R is the rate, R∈[0,1]∈[0,1], i.e. the amount of information contained in one bit.
- A is the information set, A⊂{1,...,N}A⊂{1,...,N}, i.e. the set of positions which contains the information bits.
- uAcuAc are the frozen bits, uAc∈{0,1}N(1−R)uAc∈{0,1}N(1−R) , i.e. bits which have fixed values, shared between the encoder and the decoder.

High-level transmission scheme

Fig. 1 - High-level transmission scheme, using polar encoding and successive cancellation (SC) decoding.

The transmission process works as follow:

- We must first choose the information set A. Note that the choice of A depends on the particular channel over which transmission takes place, i.e. different channels yield different information sets.
- We then want to transmit NRNR information bits contained in the vector uAuA.
- N(1−R)N(1−R) frozen bits contained in the vector uAcuAc are fixed.
- uAcuAc and uAuA are combined to obtain uN1u1N.
- uN1u1N is encoded into xN1x1N using the polar recursive encoding. This is a fast algorithm which allows to perform in O(N log N) the linear encoding xN1=uN1GNx1N=u1NGN, with GN=[1100]⊗log2NGN=[1010]⊗log2N where ⊗⊗ is the Kronecker product.
- xN1x1N is transmitted over the channel WNWN (which corresponds to N uses of the channel W), and yN1y1N is received.
- From yN1y1N, the successive cancellation (SC) decoder produces u^N1u^1N, which is an estimate of uN1u1N (making use of the frozen bits values!). The complexity of this operation is O(N log N).
- Finally, only the components of u^N1u^1N corresponding to information bits are kept, yielding u^Au^A.

We provide a simple implementation in MATLAB, in order to complement the material referenced below.

The implementation contains a polar encoder-decoder pair for a binary erasure channel (BEC). Please start with the readme file as entry point.

3.2 ARCHITECTURE OF POLAR CODER

Polar encoder and decoder together comprises the Polar coder architecture

(shown in figure 1).Two identical Recursive convolutional encoders(RSC) and a pseudorandom interleaver constitutes the polar encoder (figure 2.1a).LTE employs a 1/3 rate parallel concatenated polar code. Each RSC works on two different data. Original data is provided to the first encoder, while the second encoder receives the interleaved version of the input data. A specified algorithm is used to scramble the data bits and the method is called Interleaving. An appreciable impact on the performance of a decoder is seen with the interleaving algorithm when used. The RSC1 and RSC2 encoder outputs along with systematic input comprises the output of polar encoder, that is, a 24-bit output is generated which is illustrated in figure 6. This will be transmitted through the channel to the Polar decoder. A standard polar decoder block diagram is shown in Figure 3 that contains two modules of SISO decoders together with two pseudorandom interleavers and a pseudorandom deinterleaver.

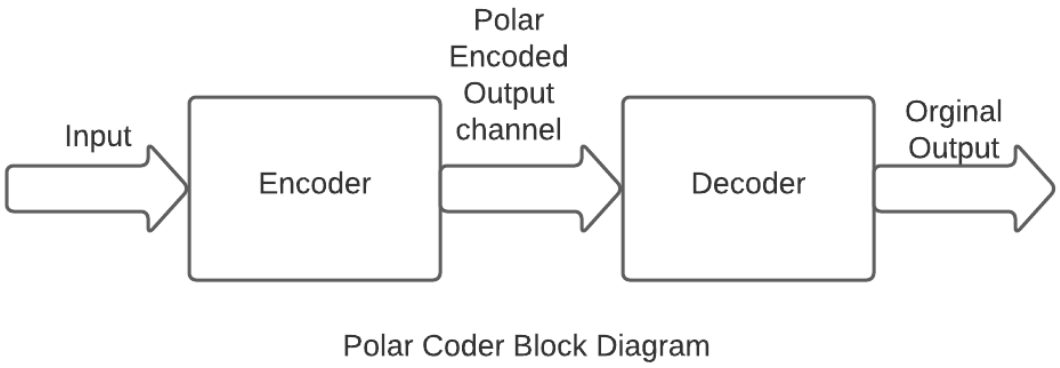


Fig 3.2a-POLAR CODER BLOCK DIAGRAM

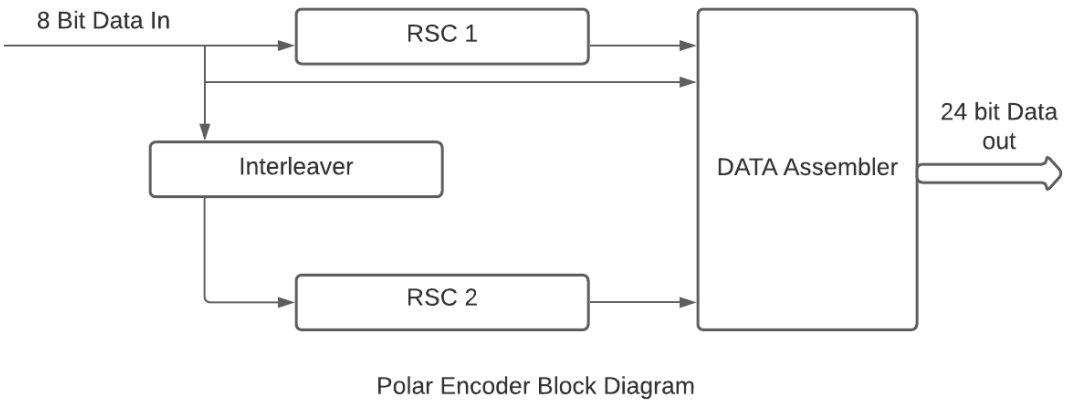


Fig 3.2b POLAR ENCODER BLOCK DIAGRAM

The usually used method of polar code decoding is carried out using the BCJR algorithm. The fundamental and basic idea behind the polar decoding algorithm is the iteration between the two SISO part decoders which is illustrated in figure 3. It comprises a pair of decoders, those which work simultaneously in order to refine and upgrade the estimate of the original information bits. The first and second SISO decoder, respectively, decodes the convolutional code generated by the first or second CE. A polar-iteration corresponds to one pass of the first component decoder which
is followed by a pass of the second component decoder.

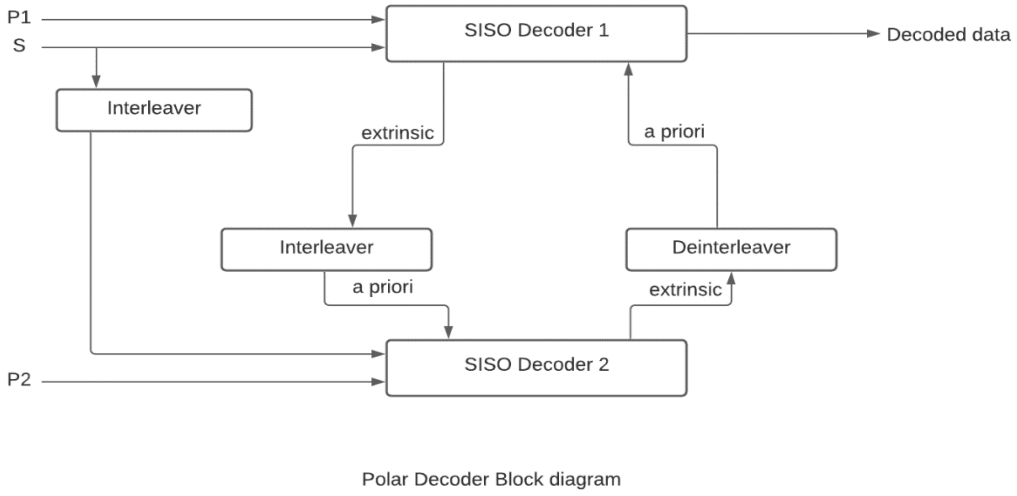


Fig 3.3c. Polar Decoder Block diagram

3.2.1 SISO Decoder

The signal which is received at the input of a soft-in-soft-out (SISO) decoder is the real (soft)value of that signal. An estimate of each input bit The decoder then generates an approximation for each data bit expressing the probability that the transmitted data bit is equal to one. The maximum a-posteriori (MAP) algorithm is used in the polar-decoder under consideration in this paper for the SISO component decoder. The MAP algorithm never restricts the set of bit estimates to correspond strictly to a valid path through the trellis. Therefore, the results produced by a Viterbi decoder that recognizes the most likely true path through the trellis should differ from those generated by that.

AES can be implemented either in software or hardware. Software implementation requires less resources and cost and its implication also limited, having low seed. Nowadays we require large volume data and high speed requirements made it to implement in hardware. Hardware nothing but we have application specific IC and FPGA. FPGA is reconfigurable device which supports wide range of functionality than ASIC. So we prefer FPGAS to implement AES.

3.2.2 Interleaver

Interleaving is a tool that is used to enhance existing error correcting codes so that they can be used to perform burst error corrections as well.

Most error correcting codes (ECCs) are designed to correct random errors, i.e. error caused by additive noise that is independent of each other. Burst error are the errors that occur in a sequence or as groups. They are caused due to defects in storage media or disruption in communication signals due to external factors like lightning etc. Interleaving modifies the ECC or does some processing on the data after they are encoded by ECCs.

Interleaving Procedure

During interleaving, the message symbols are arranged over multiple code blocks by the interleaver before sending over network channels. Due to this, long burst noise sequences are spread out among multiple blocks. When the decoder rearranges the blocks, the errors appear

as independent random errors or burst errors with short lengths. The decoder is able to correct the errors using the error correcting algorithm.

The depth of interleaving required depends upon the length of the noise bursts that the ECC can recover from.

Types of Interleaving

There are two types of interleaving

Periodic Interleaving, In this case, the message is ordered in a repeating sequence of bytes. The interleaver accepts data symbols in blocks and performs identical permutations on the blocks before transmitting them. For example, the sequential blocks of code may be written to a matrix in a row-wise manner and then read out from the matrix in a column-wise manner. Block interleaving is a type of periodic interleaving.

Pseudo-random Interleaving Pseudo-random interleaving involves rearrangement of the message blocks in a pseudo-random sequence generated by specific algorithms.

Choosing the interleaver is a significant part of the polar code design. Interleavers scramble data in a pseudorandom order to lessen the resemblance between adjacent bits at the input of the convolutional encoder. The interleaver is used on both the encoder part and the decoder part. It produces a long block of data on the encoder side, while it compares two SISO decoders' output in the decoder portion and helps to fix the error. Pseudo-random deinterleaver functions in a complimentary manner of pseudo-random interleaver.

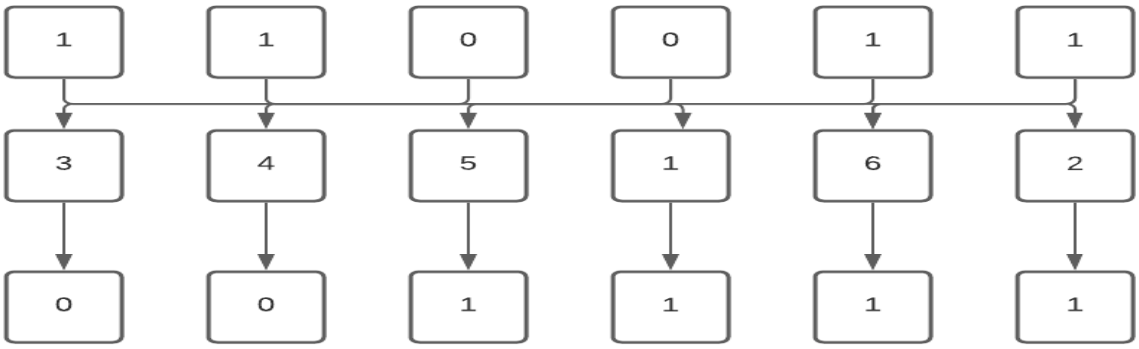


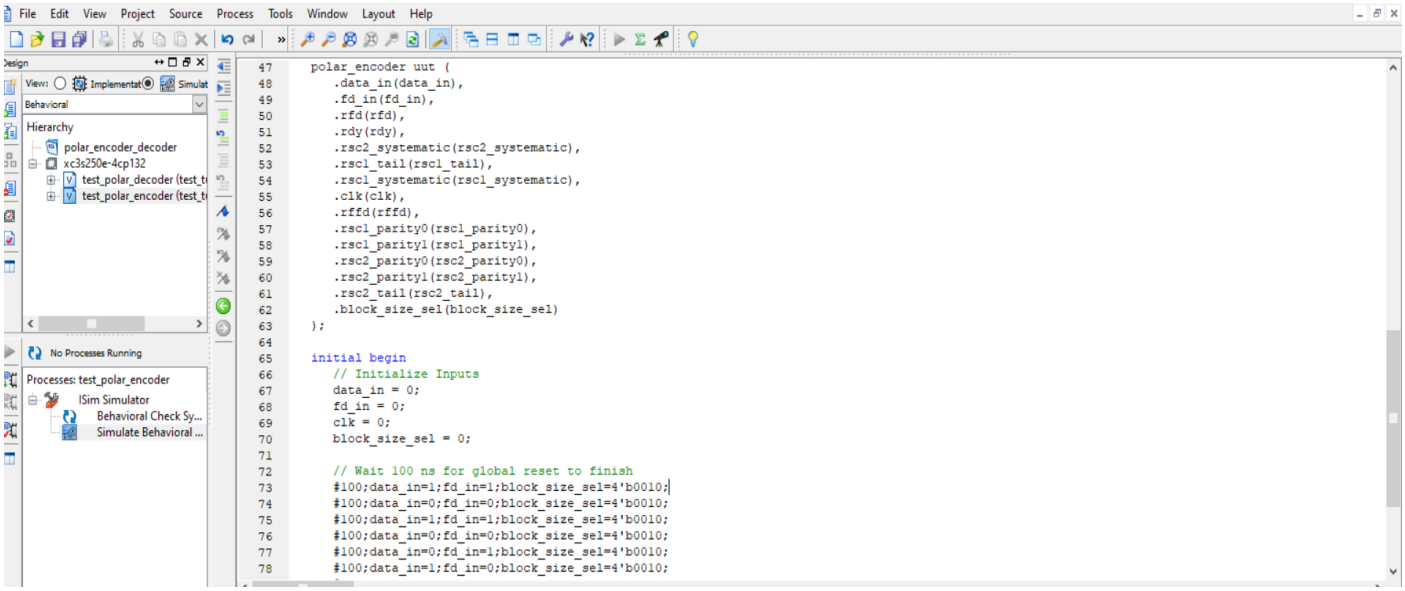
Fig 2.3a Pseudo-random interleaver working principle

Fig. 3.2.2a Pseudo-random interleaver working principle

CHAPTER 4

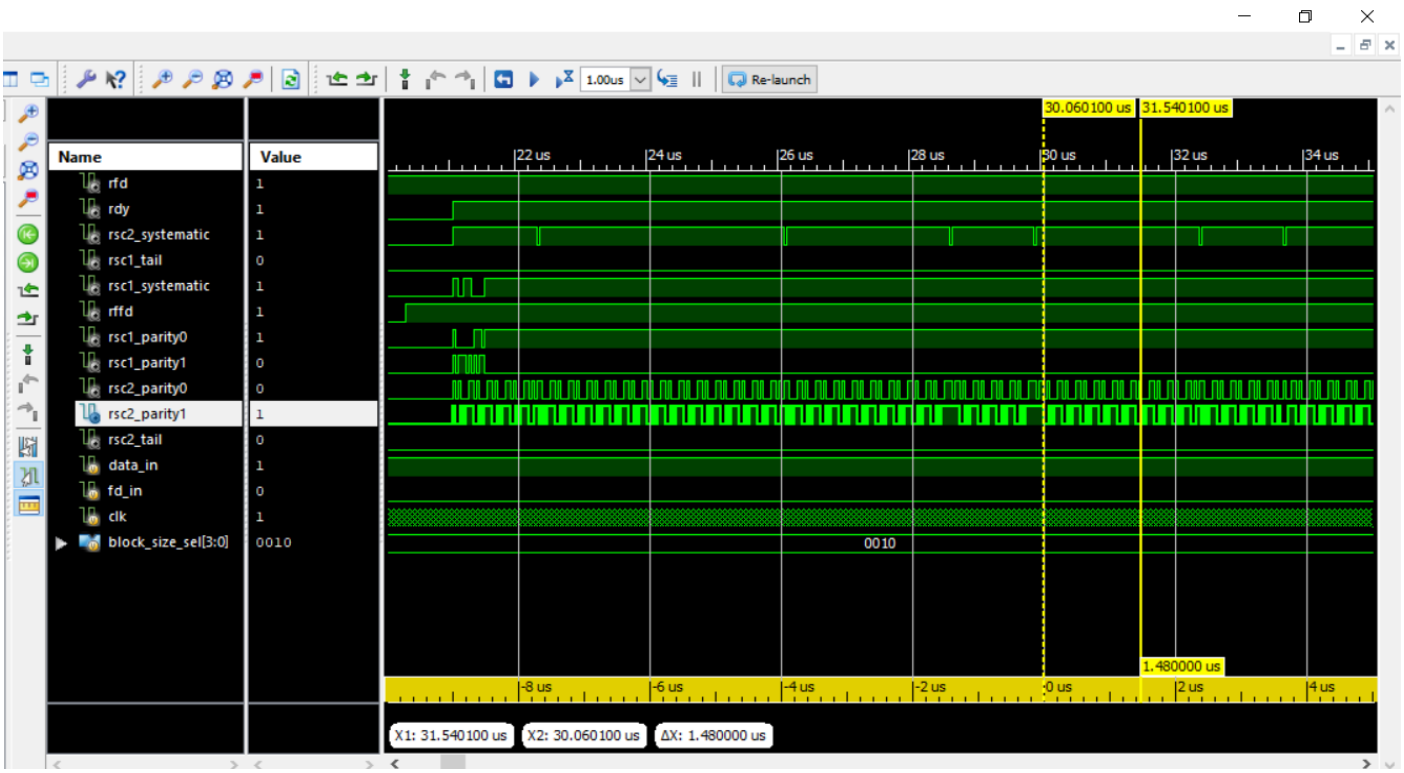
RESULTS AND DISCUSSIONS

For Encoder

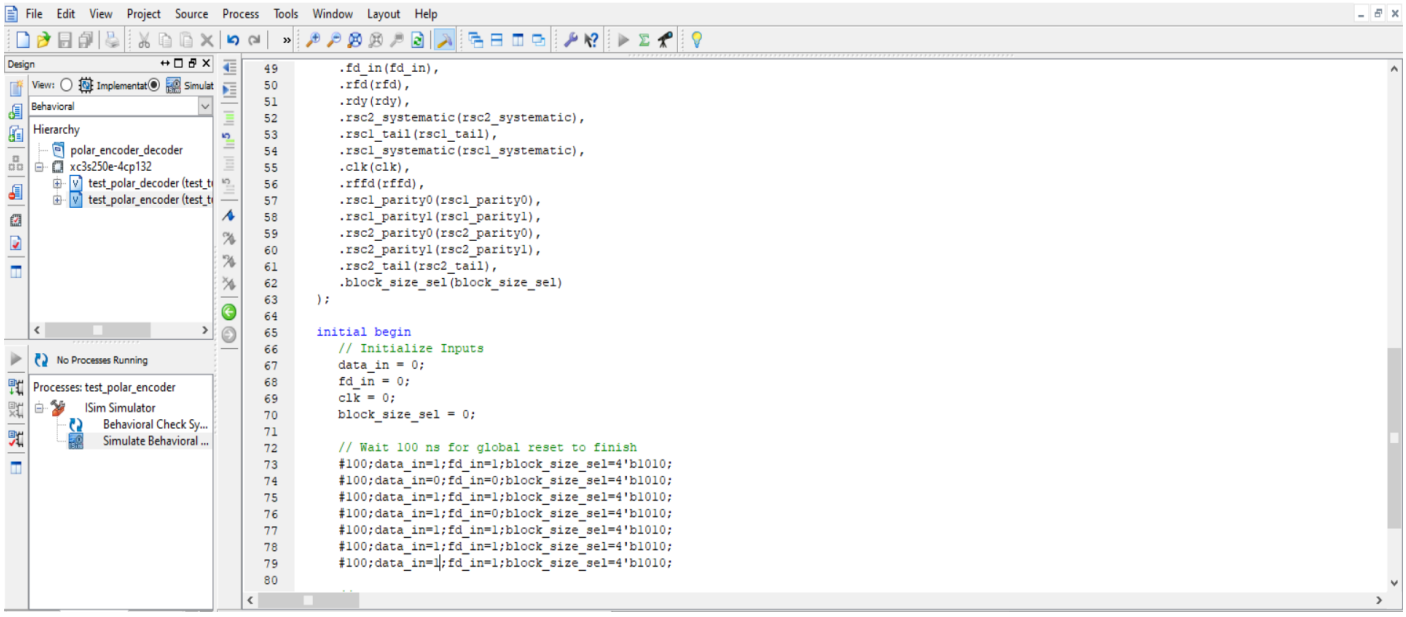


```
polar_encoder uut (
    .data_in(data_in),
    .fd_in(fd_in),
    .rfd(rfd),
    .rdy(rdy),
    .rsc2_systematic(rsc2_systematic),
    .rsc1_tail(rsc1_tail),
    .rsc1_systematic(rsc1_systematic),
    .clk(clk),
    .rffd(rffd),
    .rsc1_parity0(rsc1_parity0),
    .rsc1_parity1(rsc1_parity1),
    .rsc2_parity0(rsc2_parity0),
    .rsc2_parity1(rsc2_parity1),
    .rsc2_tail(rsc2_tail),
    .block_size_sel(block_size_sel)
);

initial begin
    // Initialize Inputs
    data_in = 0;
    fd_in = 0;
    clk = 0;
    block_size_sel = 0;

    // Wait 100 ns for global reset to finish
    #100;data_in=1;fd_in=1;block_size_sel=4'b0010;
    #100;data_in=0;fd_in=0;block_size_sel=4'b0010;
    #100;data_in=1;fd_in=1;block_size_sel=4'b0010;
    #100;data_in=0;fd_in=0;block_size_sel=4'b0010;
    #100;data_in=0;fd_in=1;block_size_sel=4'b0010;
    #100;data_in=1;fd_in=0;block_size_sel=4'b0010;
```

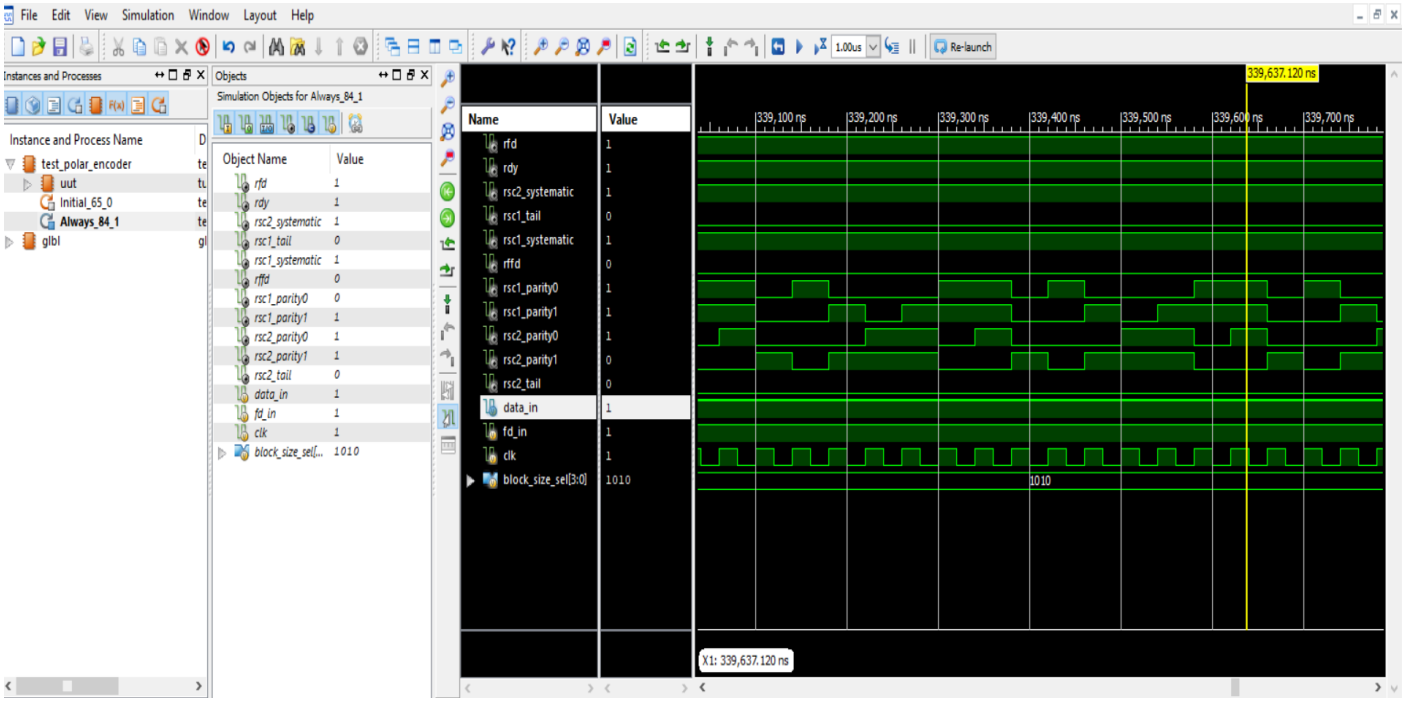
Here the data_in 1010011 and fd_in 1010100 is transmitted for every 100µ seconds and the block size is 0010



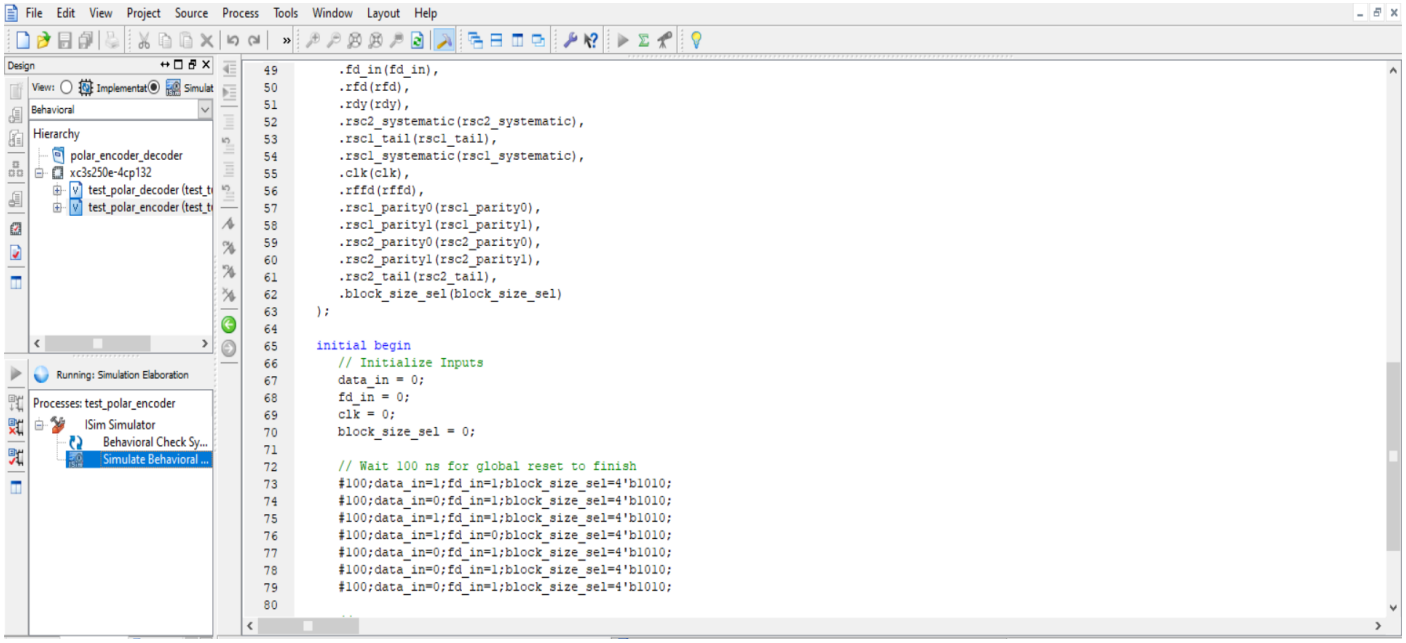
We can observe the block size 0010 and data_in 1 and d_in 0 is transmitted and the transition occurred at posedge of clock.



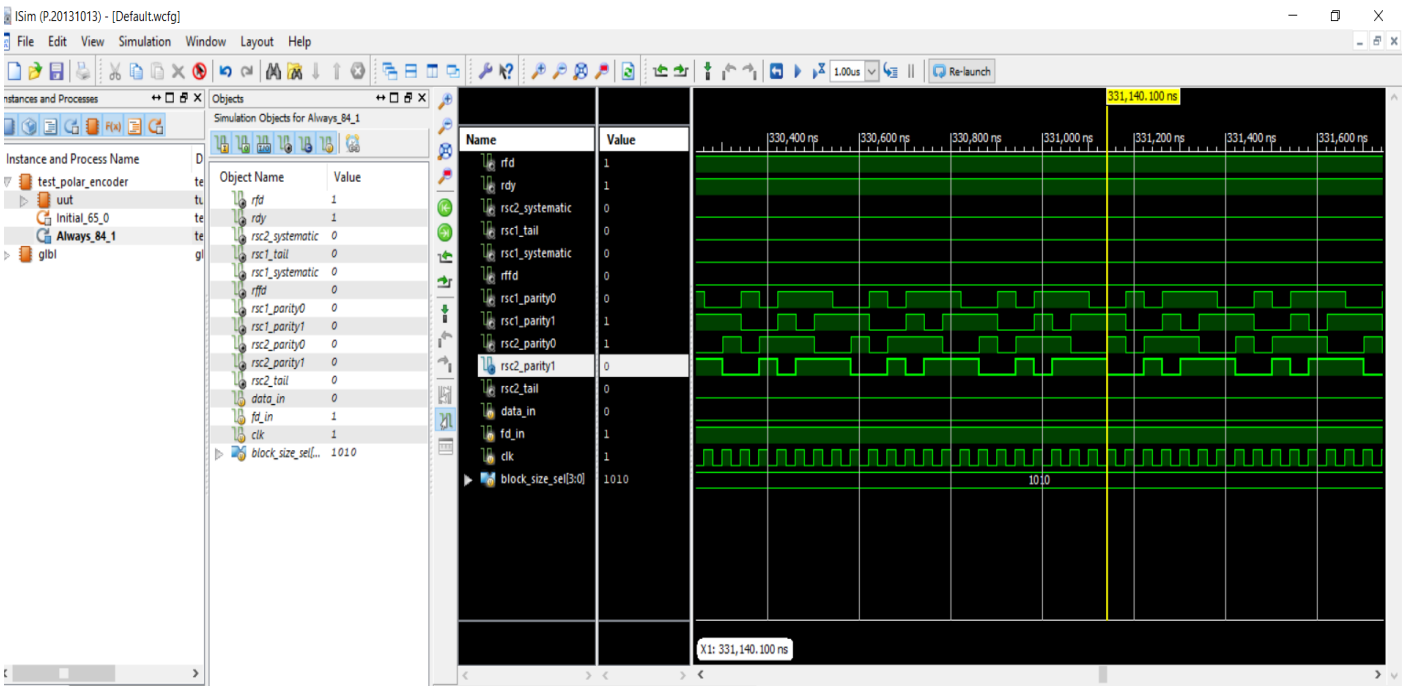
Here the data_in 1011111 and fd_in 1010111 is transmitted for every 100µ seconds and the block size is 0010



We can observe the block size 1010 and data_in 1 and fd_in 1 is transmitted and the transition occurred at posedge of clock.

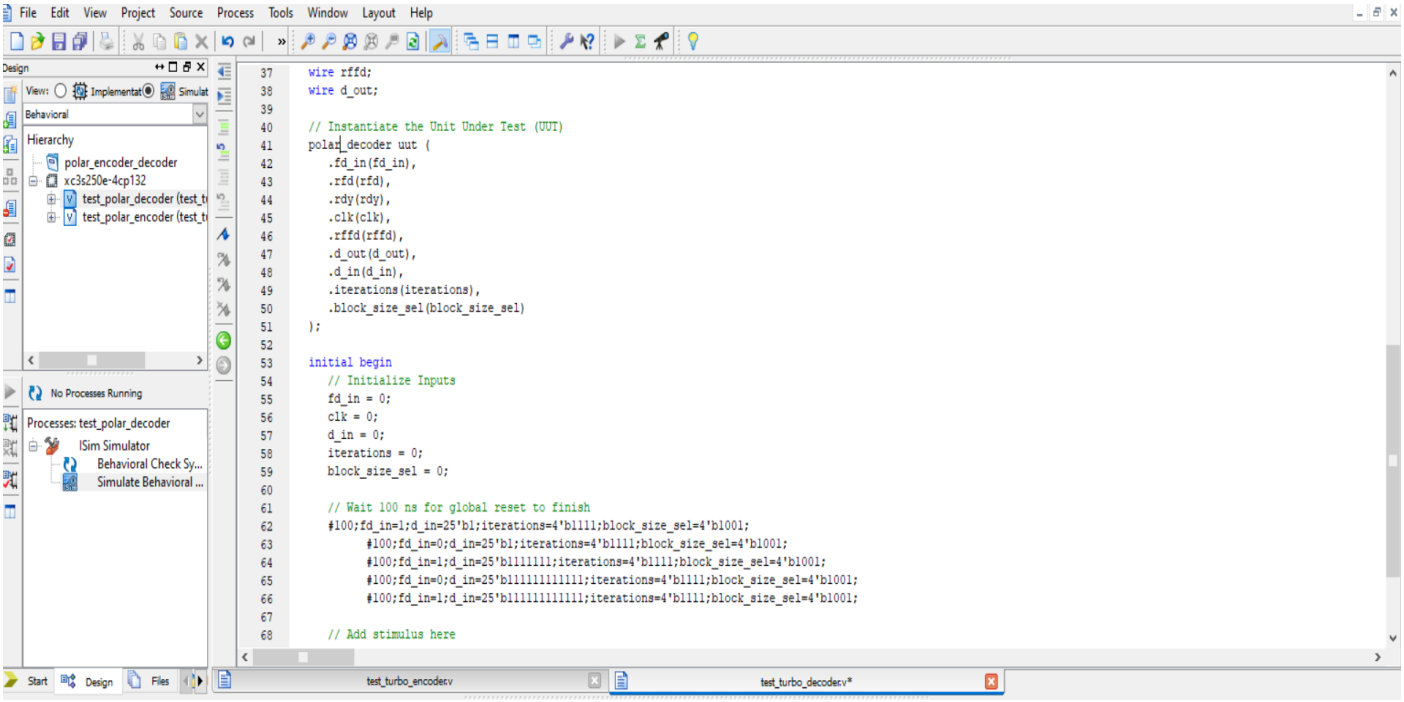


Here the data_in 1011000 and fd_in 1110111 is transmitted for every 100µ seconds and the block size is 1010



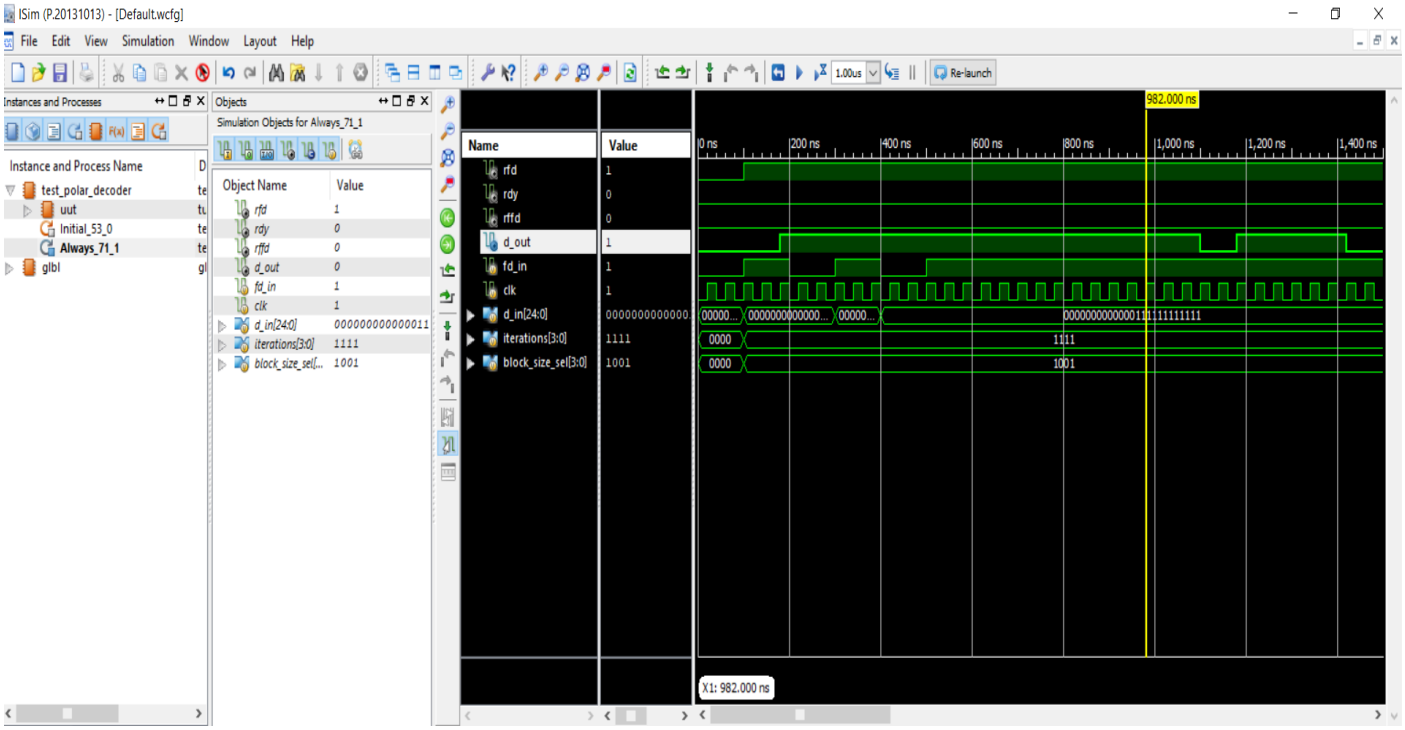
We can observe the block size 1010 and data_in 0 and fd_in 1 is transmitted and the transition occurred at posedge of clock.

For Decoder

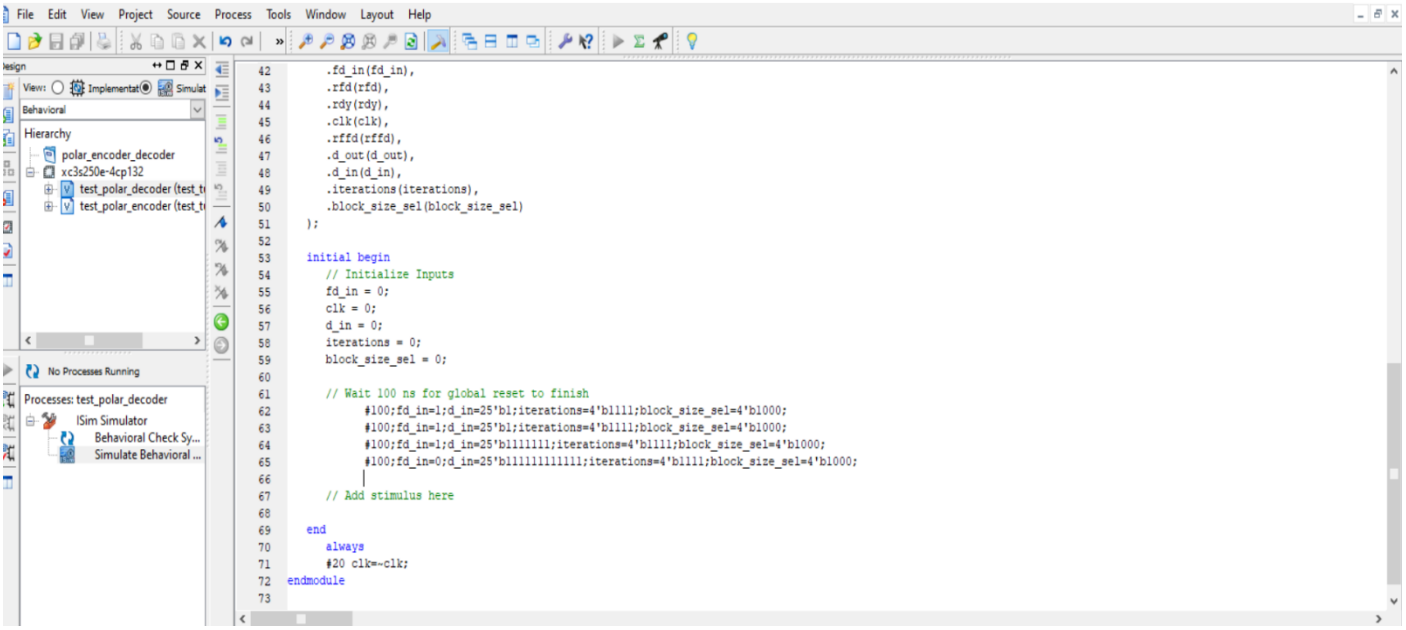


```
37    wire rffd;
38    wire d_out;
39
40    // Instantiate the Unit Under Test (UUT)
41    polar_decoder uut (
42       .fd_in(fd_in),
43       .rfd(rfd),
44       .rdy(rdy),
45       .clk(clk),
46       .rffd(rffd),
47       .d_out(d_out),
48       .d_in(d_in),
49       .iterations(iterations),
50       .block_size_sel(block_size_sel)
51    );
52
53    initial begin
54       // Initialize Inputs
55       fd_in = 0;
56       clk = 0;
57       d_in = 0;
58       iterations = 0;
59       block_size_sel = 0;
60
61       // Wait 100 ns for global reset to finish
62       #100;fd_in=1;d_in=25'b1;iterations=4'b1111;block_size_sel=4'b1001;
63            #100;fd_in=0;d_in=25'b1;iterations=4'b1111;block_size_sel=4'b1001;
64            #100;fd_in=1;d_in=25'b1111111;iterations=4'b1111;block_size_sel=4'b1001;
65            #100;fd_in=0;d_in=25'b1111111111111;iterations=4'b1111;block_size_sel=4'b1001;
66            #100;fd_in=1;d_in=25'b1111111111111;iterations=4'b1111;block_size_sel=4'b1001;
67
68       // Add stimulus here
```

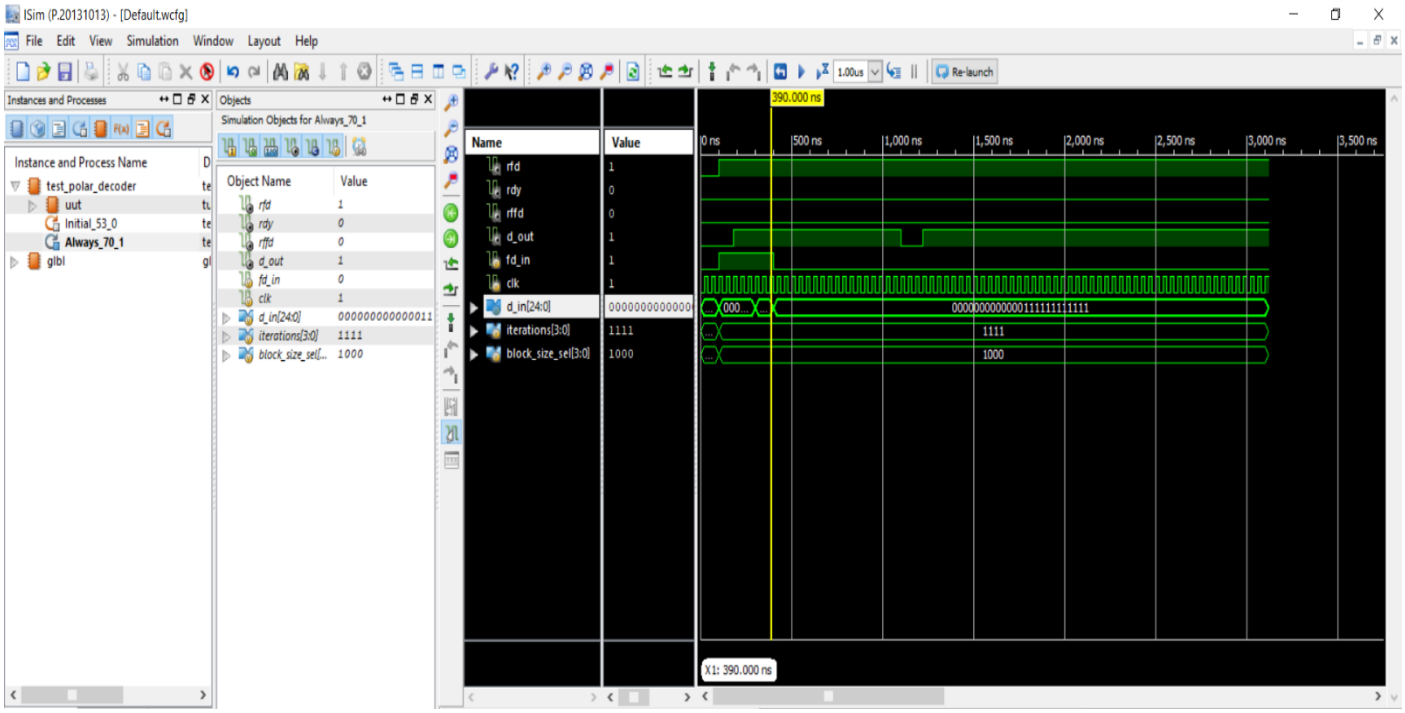
Here the block size is 1001 and fd_in 10101 and d_in 11111 is transmitted for the given iterations for every 100µ seconds



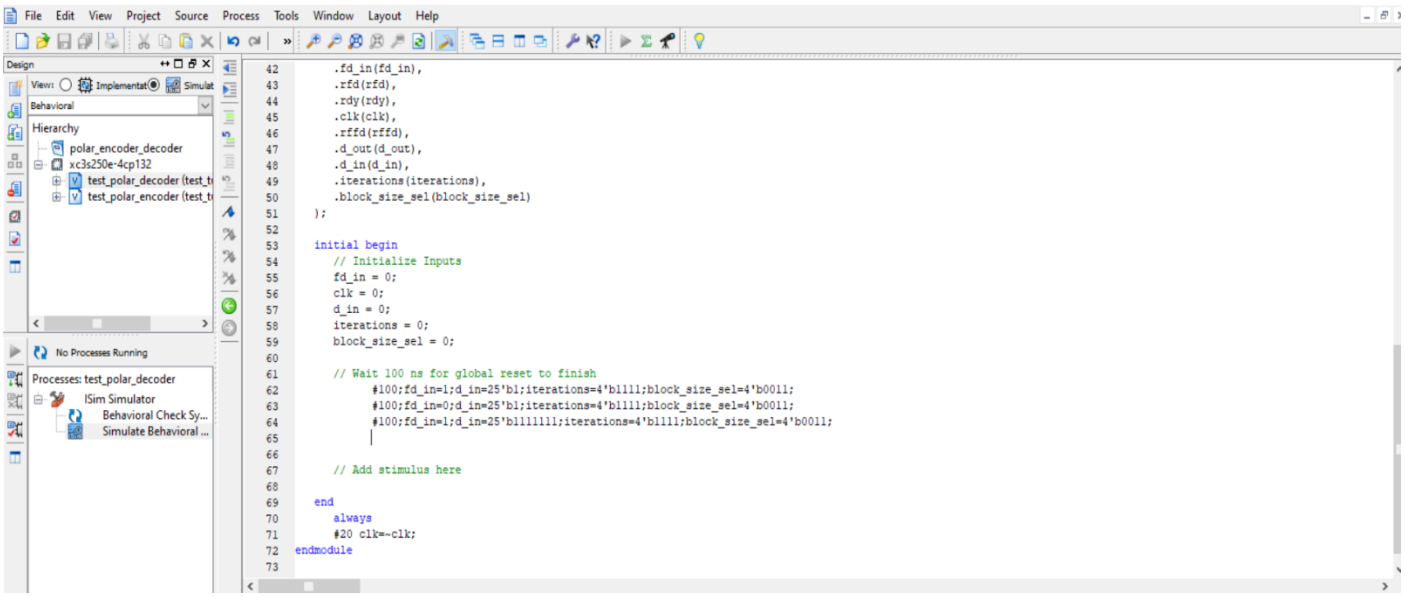
We can observe the block size 1001, iterations 1111 and d_out 1 and fd_in 1 is transmitted and the transition occurred at posedge of clock.



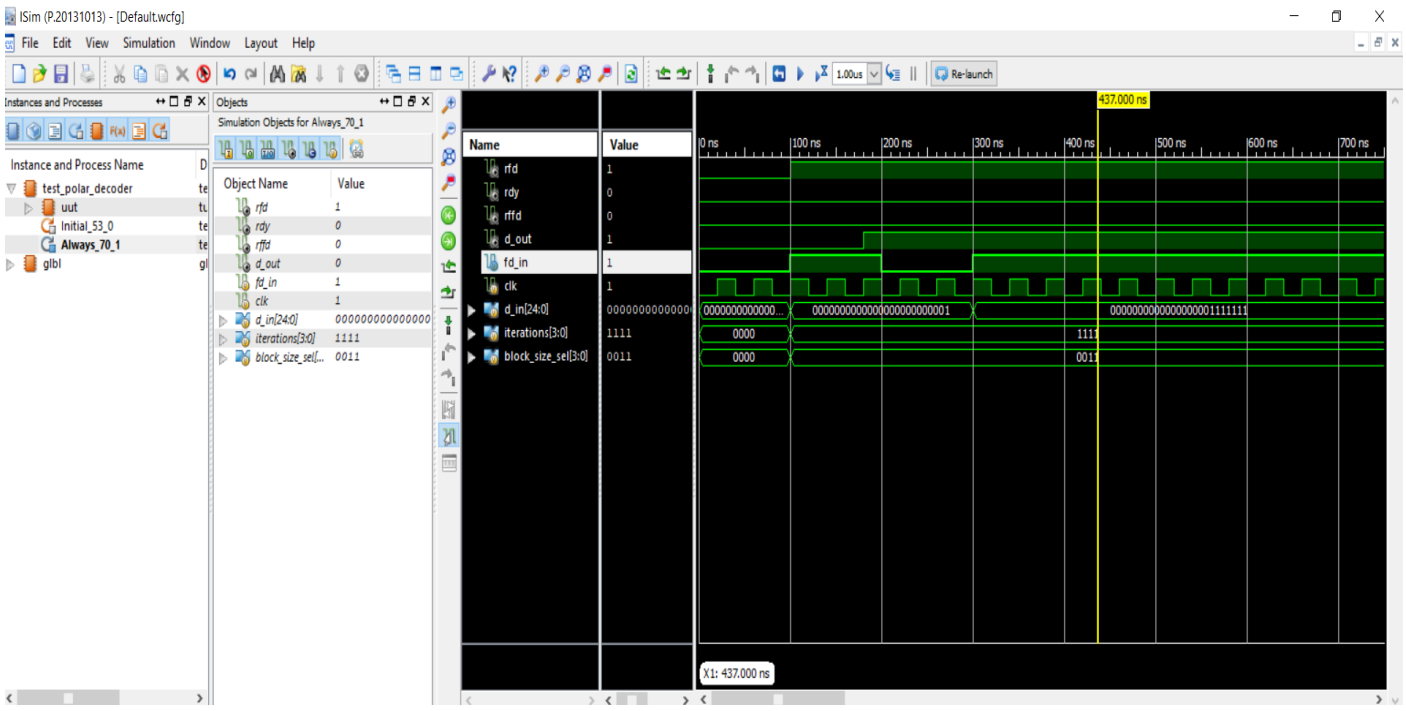
Here the block size is 1000 and fd_in 1110 and d_in 11111 is transmitted for the given iterations for every 100µ seconds



We can observe the block size 1000, iterations 1111 and d_out 1 and fd_in 1 is transmitted and the transition occurred at posedge of clock



Here the block size is 0011 and fd_in 1101 and d_in 11111 is transmitted for the given iterations for every 100µ seconds



We can observe the block size 0011, iterations 1111 and d_out 1 and fd_in 1 is transmitted and the transition occurred at posedge of clock.

CHAPTE

R 5

CONCLU

SION

The polar encoder and decoder designs are done using Verilog HDL and simulation has done in Xilinx ISE13.2. The RTL and Technology schematics have observed in XILINX. Synthesis repost has shown the details of our proposed design. This example highlights one of the polar coding schemes (CRC-Aided Polar) specified by 3GPP for New Radio control channel information (DCI, UCI) and broadcast channel (BCH). It shows the use of components for all stages of the processing (encoding, rate-matching, rate-recovery and decoding) and uses them in a link with QPSK over an AWGN channel. Highlighted performance results for different code rates and message lengths show agreement to published trends, within parametric and simulation assumption variations.

REFERENCES

[1] Arikan, E., "Channel Polarization: A Method for constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," IEEE Transactions on Information Theory, vol. 55, No. 7, pp. 3051-3073, July 2009.

[2] Tal, I, and Vardy, A., "List decoding of Polar Codes", IEEE Transactions on Information Theory, vol. 61, No. 5, pp. 2213-2226, May 2015.

[3] Niu, K., and Chen, K., "CRC-Aided Decoding of Polar Codes," IEEE Communications Letters, vol. 16, No. 10, pp. 1668-1671, Oct. 2012.

[4] Niu, K., Chen, K., and Lin, J.R., "Beyond turbo codes: rate compatible punctured polar codes", IEEE International Conference on Communications, pp. 3423-3427, 2013.

[5] Stimming, A. B., Parizi, M. B., and Burg, A., "LLR-Based Successive Cancellation List Decoding of Polar Codes", IEEE Transaction on Signal Processing, vol. 63, No. 19, pp.5165-5179, 2015.

[6] 3GPP TS 38.212. "NR; Multiplexing and channel coding (Release 15)." 3rd Generation Partnership Project; Technical Specification Group Radio Access Network.

[7] R1-1711729. "WF on circular buffer of Polar Code", 3GPP TSG RAN WG1 meeting NR Ad-Hoc#2, Ericsson, Qualcomm, MediaTek, LGE. June 2017.

[8] E. Boutillon C. Douillard and G. Montorsi "Iterative decoding of concatenated convolutional codes: Implementation issues" Proc. IEEE vol. 95 no. 6 pp. 1201-1227 June 2007.

[9] Stimming, A. B., Parizi, M. B., and Burg, A., "LLR-Based Successive Cancellation List Decoding of Polar Codes", IEEE Transaction on Signal Processing, vol. 63, No. 19, pp.5165-5179, 2015.

[10] N. Priyadarshini and K. Pargunarajan "A Constrained Search Algorithm for

Selection of Optimal Generators in Convolutional Encoder" International Journal of Computer Science Engineering and Technology (IJCSET) vol. 1 no. 6 pp. 324-326 July 2011.

[11] V. Kavinilavu S. Salivahanan V. S. Kanchana Bhaaskaran Samiappa Sakthikumaran B. Brindha and C. Vinoth "Implementation of Convolutional Encoder and Viterbi Decoder using Verilog HDL" IEEE 3rd International Conference on Electronics Computer Technology