A Model of Turbo Encoder Based on Field Programmable Gate Array (FPGA) for Nano Satellite Application

Laila Prakasita, Heroe Wijanto, and Budi Syihabuddin

***Abstract*—Satellite communication is an important factor in nano satellite remote sensing system. Error correction in satellite communication must be designed to guarantee the image data is received correctly by the ground station. In satellite communication, convolutional code and turbo code are commonly used for error correction coding schemes. In this paper, nano satellite communication schemes are observed by comparing the simulation of image transmission using nano satellite communication design without channel code, with rate 1/3 turbo code, and with rate ½ turbo code. The result shows that rate 1/3 turbo code yields the best performance. At BER of 10^{-6}, Eb/No is 5,55 dB for rate 1/3 turbo code, 9,67 dB for rate ½ turbo code, and 16,6458 dB for system without channel code. Based on this result, rate 1/3 turbo encoder has been designed for nano satellite application using FPGA. The turbo encoder is implemented on FPGA ATLYS Spartan-6 XC6SLX45 CSG324C. Based on the simulation, the delay process of turbo encoder using 8 x 228 bits of interleaver is 11.043 clocks or 11,043 us. The turbo encoder built in this research can be implemented on FPGA because the FPGA resource which has been used is below FPGA resource constraint.**

***Index Terms*—Turbo Code, FPGA, RSPL, Nano Satellite.**

I. Introduction

Satellite communications has been implemented in various areas such as remote sensing, mobile communications and digital satellite TV [1]. Nanosatellite, a kind of satellite which weighs less than 10 kg, is being developed for research by several universities. One of them is Telkom University. In 2015, Telkom University is developing nanosatellite with remote sensing mission. Remote sensing payload needs reliable communication scheme to transmit the image data to ground station.

Characteristics of data transmission via satellite are power limited, high delay, bursting noise, and low BER [1]. Therefore, communication system that requires minimum power and reliable are needed for satellite communication, particularly nano satellite. Power required for data transmission in a satellite communication system can be shown from Eb/No or energy bits of information per noise density. The reliability of digital communication systems is shown from the calculation of Bit Error Rate (BER) at the receiver [2]. One way to minimize Eb/No is by using channel code in the system [2].

Basically, channel code is divided into two types, convolutional code and block code. Convolutional code is widely used for communication systems with noisy channel, such as satellite communication and LTE [3][4]. The convolutional code is then developed into turbo code. Turbo code is a Parallel Concatenated Convolutional Code (PCCC), which consists of Recursive Systematic Convolutional (RSC) and the internal interleaver. Turbo code is one of the channel coding recommended by the Consultative Committee for Space Data Systems (CCSDS). The CCSDS recommended turbo code rate for satellite communication is 1/2, 1/3, ¼, and 1/6 [3].

In this paper, the performance of nano satellite communication systems without channel coding and with channel coding has been simulated. The turbo code rates used in this research are 1/2 and 1/3 with QPSK modulation. This rate is chosen because it has lower level of circuit complexity than rate 1/4 and 1/6. The best scenario encoder is then implemented on Spartan-6 FPGA ATLYS XC6SLX45 CSG324C.

The research in [5] has implemented turbo encoder based on Virtex-E FPGA for 3G systems using random interleaver and four constraint lengths. In this paper, the turbo encoder has been implemented on Spartan-6 FPGA ATLYS XC6SLX45 CSG324C for nano satellite communication systems by using block interleaver and five constraint lengths. FPGA is used in this paper because it serves as the main data processing on nano satellite communications, allows faster data processing, and can be configured by the user.

II. System design

A. Turbo Code Performance

To determine the turbo encoder that will be implemented on the FPGA, the satellite communication system has to be simulated first. The simulation system consists of a transmitter, channel, and receiver. Transmitter block consists of random bit generator, turbo encoder, and QPSK modulator.

Manuscript received July 1, 2015.

Laila Prakasita is student in School of Electrical Engineering, Telkom University, Jalan Telekomunikasi Terusan Buah Batu, Bandung, Indonesia; e-mail: lailaprakasita@students.telkomuniversity.ac.id

Heroe Wijanto, and Budi Syihabuddin are with the School of Electrical Engineering, Telkom University, Jalan Telekomunikasi Terusan Buah Batu, Bandung, Indonesia; e-mail: heroe@telkomuniversity.ac.id, budisyihab@telkomuniversity.ac.id

Because the ground station is fixed, the channel used in the simulation is AWGN channel. Receiver block consists of QPSK demodulator and decoder with viterbi hard decision algorithm.

Turbo encoder rates simulated in this research are rate 1/2 and 1/3 with internal block interleaver. Based on the information of block length from CCSDS standard recommendation, this research uses interleaver with the size of matrix 223 (rows) x 8 (columns). To achieve the performance at BER of 10^{-6}, 40 times of iterations are performed.

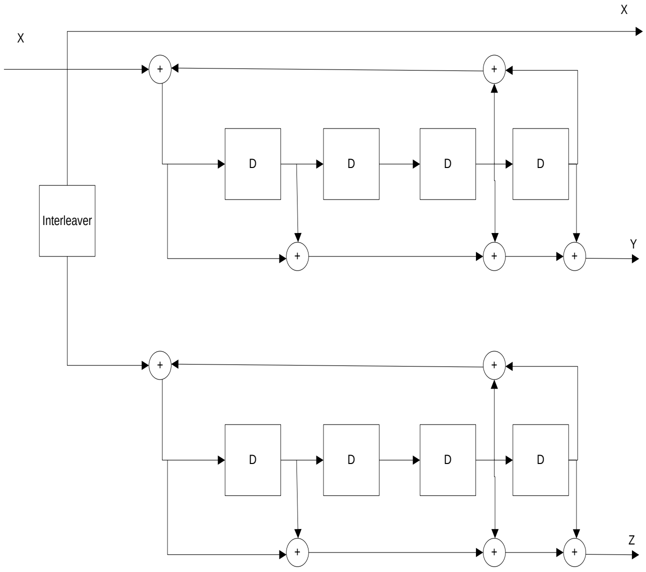


Fig. 1. A model of rate 1/3 turbo encoder.

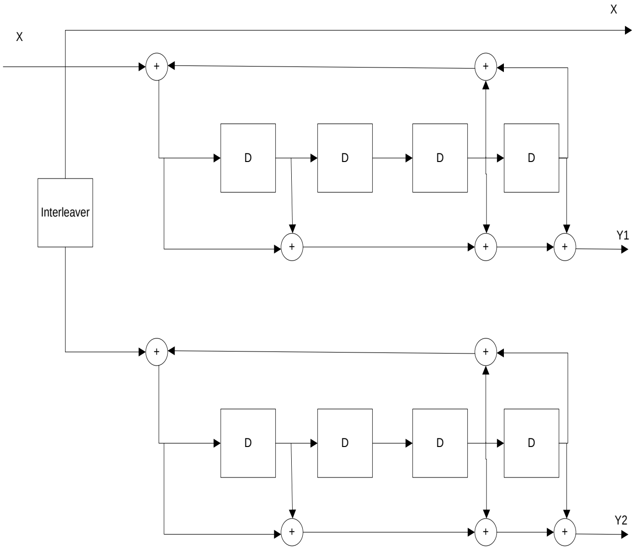


Fig. 2. A model of rate 1/2 turbo encoder.

In Fig. 1 and Fig. 2, simulation of turbo encoder model are shown. Each simulated turbo encoder has five constraint lengths. According to CCSDS standard recommendation, the polynomial generator used for backward connection vector for both component codes and all code rates shall be G0 = 10011 and the polynomial generator used for forward connection vector for both component codes and rates 1/2 and 1/3 shall be G1 = 11011 [3]. So, the polynomial generator of the turbo encoder uses octal number (23,33). The equation for the polynomial generator (23,33) is shown as follows:

$$G(D) = \left[1 \;\; \frac{1+D^1+D^3+D^4}{1+D^3+D^4}\right] \tag{1}$$

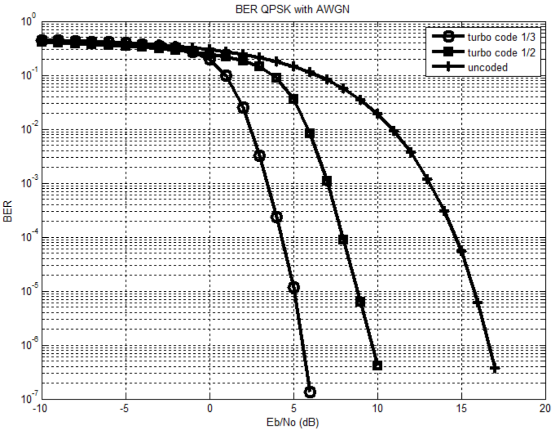


Fig. 3. Comparison of Eb/No required for system with rate 1/2 turbo code, rate 1/3 turbo code, and without channel coding.

As shown in Fig. 1, if the input X = {X_1, X_2, ..., X_n}, then the output becomes X' = {X_1, Y_1, Z_1, X_2, Y_2, Z_2, ..., X_n,Y_n,Z_n}. For rate 1/3, the number of code words is three times as many as the number of input bits. Y is the output from the input bits that have been encoded according to polynomial generator, while Z is the output from input bits that have been randomized by the block interleaver and encoded by polynomial generator.

For rate 1/2, the number of the code words is twice as many as the number of bits of input, so if the input X = {X_1, X_2, ..., X_n}, then the output becomes X' = {X_1, Y_1, X_2, Y_2, ..., X_n,Y_n}. Y_1 is the output from the input bits that have been encoded according to polynomial generator, while Y_2 is the output from input bits that have been randomized by the block interleaver and encoded by polynomial generator. The circuit model can be seen in Fig. 2. From the parameters that have been used for simulation, the simulation result is shown in Fig. 3.

The simulation result shows that to achieve the performance at the BER of 10^{-6}, the systems with rate 1/3 turbo code requires smaller Eb/No than rate1/2 turbo code and without channel coding. Eb/No required is 9,67 dB for rate ½ turbo code, 5,55 dB for rate 1/3 turbo code, and 16.6458 dB for communication without channel coding. The system simulation shows that the smaller rate of the turbo code gives better performance. This is proved by the results of simulation.

The coding gain acquired is 11.0958 dB for rate 1/3 turbo code and 6.9758 dB for rate 1/2 turbo code.

B. VHDL Turbo Encoder Design

Based on system simulation result, from the Eb/No against BER graph, the performance of rate 1/3 turbo code is better than the rate 1/2 turbo code. Therefore, rate 1/3 turbo encoder is chosen to be implemented on FPGA. This turbo encoder is designed by using VHDL language based on turbo encoder model shown in Fig. 1. This turbo encoder consists of bit generator, block interleaver, RSC, and parallel to serial block. This design has two additional blocks, control unit block and clock divider. The VHDL system model is shown in Fig. 4.

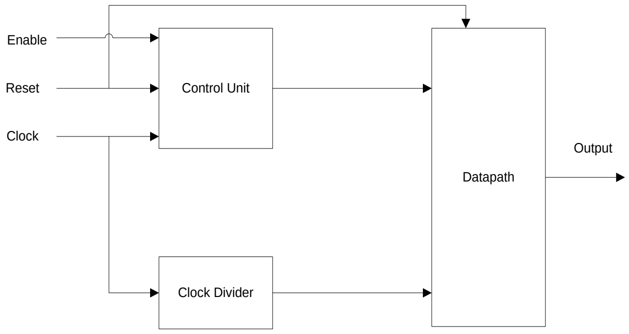


Fig. 4. Schematic of Turbo Encoder VHDL design

1) Control Unit

In this research, sub-module of control unit consists of three input ports and seven output ports. Input port consists of **clock**, **reset**, and **clock_enable**. The output port consists of a control signal for each datapath components, **en1**, **en2**, **en4**, **en5**, **en6**, **en8** and **en9**. **En1** port controls the signal input bits, **en2** port controls the signal for RSC1, interleaver, and block buffer1 when storing bits. **En4** port controls signal for RSC2, **en5** port controls signal when buffer1 block starts to pull out the bit, **en8** controls signal when buffer2 block stores bits. **En6** port controls signal when buffer2 block pulls out bits, and **en9** port controls signal when the parallel to serial block begins to pull out the bits.

2) Clock Divider

In this research, sub-module of clock divider consists of an input port and an output port. Input port consists of system clock. The output port clock is a clock which is passed down three times lower than the bit period of the system clock.

3) Datapath

Sub-module of datapath consists of bit generator, interleaver block, Recursive Systematic Code, and Parallel to serial block.

a) Bit Generator

This block is used to generate 1784 bits of number. This block consists of two signals, three input ports and an output port. Signals in this block are **sreg** and **count**. The data type of **sreg** signal is std_logic_vector, it’s used to store 1784 1784 which will be processed. Bits generated is “11111110” which is repeated 223 times. This data is generated to make testing process easy.

b) Block Interleaver

This block is used to randomize the input bits. Interleaver block will read the row bits and write the column bits. Interleaver block consists of eleven signals, four input ports and an output port. Signals of this block are **sreg2**, **sreg4**, **ulang**, **tipu**, **count2**, **sudah**, **keluar**, **penuh**, **acakx**, **acaky** and **count**. The data type of **sreg2** signal and **sreg4** is std_logic_vector, it is used to store 1.784 bits that will be processed according to the size of the matrix interleaver. Signal **count**, **reset**, **tipu**, **count2**, **acakx**, and **acaky** are used as a counter. The other signal is used as a control signal in the block interleaver.

c) Recursive Systematic Code

This block function is to randomize bits. RSC block consists of a signal, four input ports and an output port. **Sreg** is signal in this block. The data type of this signal is std_logic_vector, it is used to store four bits. The initial value of this block is '0000'. Input ports of the RSC block are **clock**, **reset**, **data_in**, and **enable**.

d) Paralel To Serial Block

This block is used to pull out bits serially from three inputs at once. Parallel to serial block consists of two signals, six input ports and an output port. Input ports of the Parallel to serial block are **clock**, **reset**, **en1**, **m**, **n**, and **o**. Clock port is a port that is used as a system start sign that is connected directly to the FPGA clock port "L15". Parallel to serial block will start to read the input when the clock rises. Reset port is used to return the signal to the initial state or predetermined condition and it’s connected directly to the port "P3" on the FPGA.

III. Result And Analysis

A. Clock Divider

Clock divider is used to slow down the clock period of the system three times slower to avoid the data accumulation at the parallel to serial block. Fig. 5 shows that the clock divider has clock period three times slower than the clock system.

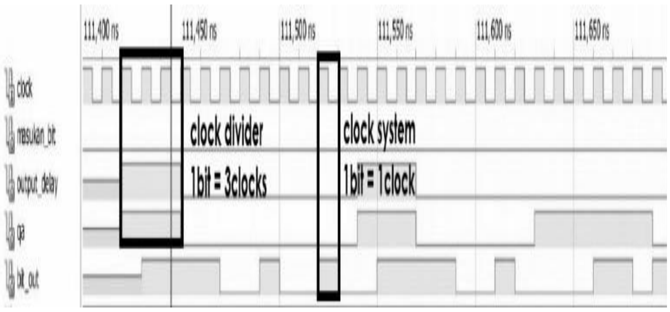


Fig. 5. Simulation result system with clock divider

B. Control Unit

This sub-module is used as a control unit to enable the data path. In this research, there are seven enable signals, **en1**, **en2**, **en4**, **en5**, **en6**, **en8** and **en9**. **En1** is activated from the 1st to 1784th clock cycle. **En2** is activated from the 2nd to 5464th clock cycle. Enable signal **en4**, **en5** and **en6** are activated from 3680th to 5461th clock cycle. **En8** is activated from the 3rd to 1786th clock cycle. **En9** is activated from 3681th to 10818th clock cycle. The simulation result of control unit sub-module is shown in Fig. 6

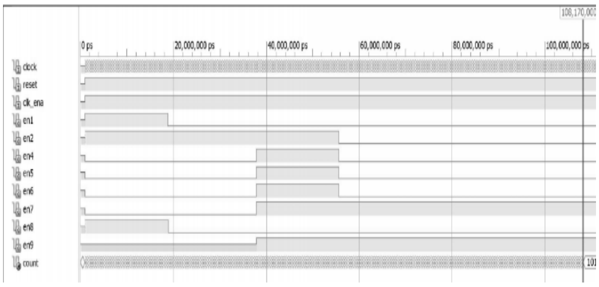


Fig. 6. Simulation result control unit

C. Datapath

1. Bit Generator

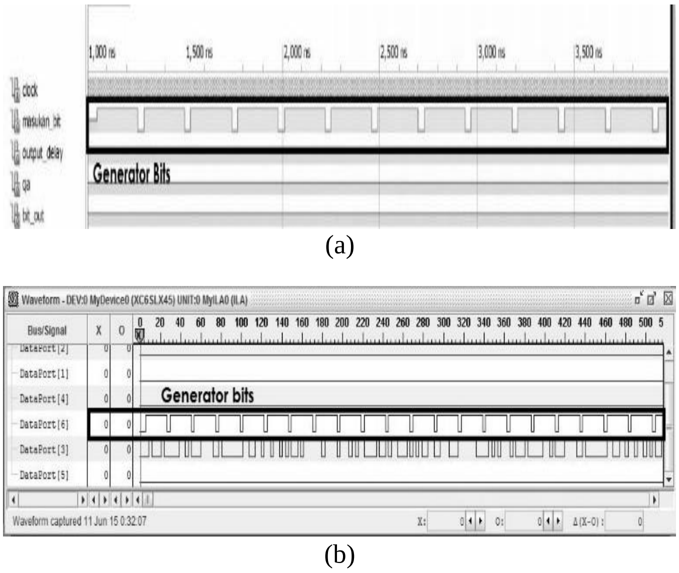
In this research, the generated bit is '1111110' which is repeated 223 times to generate 1784. Fig. 7 shows the output signal from the bit generator block simulation results and the implementation on FPGA gives the same result. This shows that the bit generator block is appropriate for the system requirements because the output signals from the ISim and Chipscope show the same output value, '11111110' which is repeated 223 times.

2. Recursive Systematic Convolutional 1

Fig. 8 shows that the output signal of the simulation RSC1 block and the implementation on FPGA gives the same result. This shows that the block RSC1 is appropriate with the system requirement because the output signal of simulation results gives the same results as the output signal of implementation and output signal value RSC1 is appropriate with the polynomial generator in equation 1.

3. Interleaver

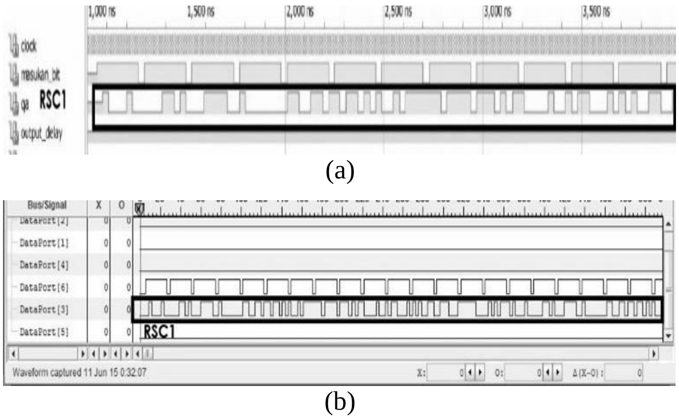
Block interleaver is used to randomize the bits by reading the bits by rows and writing it by columns. Interleaver Block works as the system requirements. This is shown by the output signals from the simulation and the implementation which give the same result. The generated bits are '1111110' which are repeated 223 times. Then, after entering the interleaver block, 1st bit to 1561th bit value will be ‘1’, while the 1562th bit to 1784th bit value will be '0'. The output signal is shown in Fig. 9.



(a)

(b)

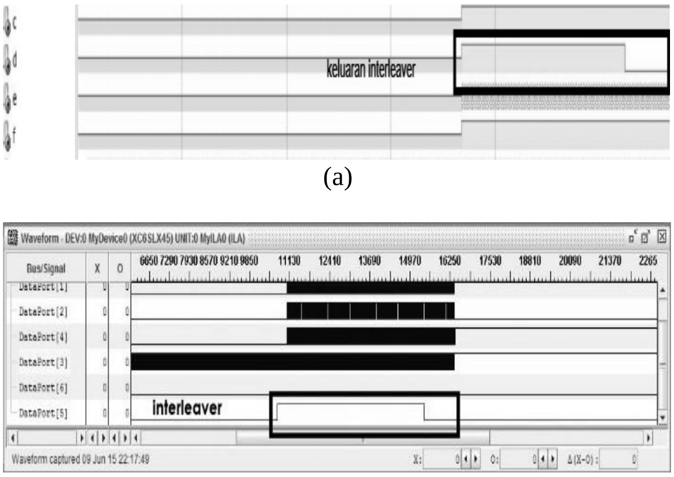
Fig. 7. Simulation result (a) and Implementation result (b) of bit generator.



(a)

(b)

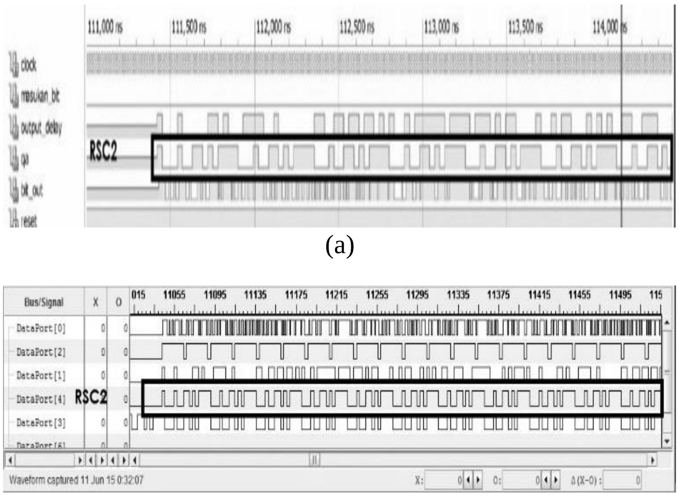
Fig. 8. Simulation result (a) and Implementation result (b) of RSC1



(a)

(b)

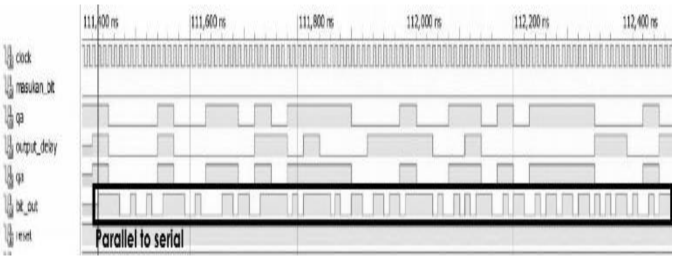
Fig. 9. Simulation result (a) and Implementation result (b) of Interleaver.



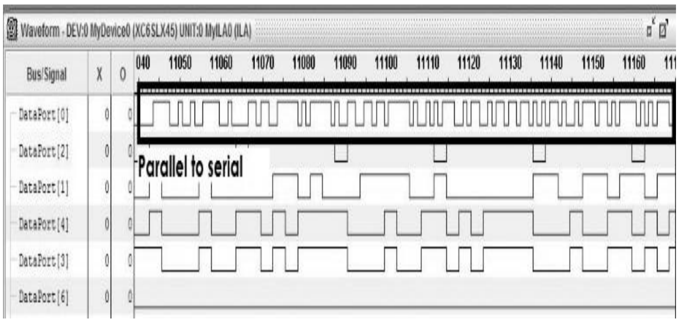
(a)

(b)

Fig. 10. Simulation result (a) and Implementation result (b) of RSC2.



(a)



(b)

Fig. 11 Simulation result (a) and Implementation result (b) of parallel to serial.

4. Recursive Systematic Convolutional 2

Polynomial generator RSC2 used to compile has the same polynomial generator as the RSC1. The difference between RSC1 and RSC2 is the input bit. RSC1 has the input from the bit generator and RSC2 has the input from the interleaver block. In this system, the RSC2 output signal from ISim simulation and Chipscope gives the same result. This shows that the RSC2 block has been successfully implemented on the FPGA.

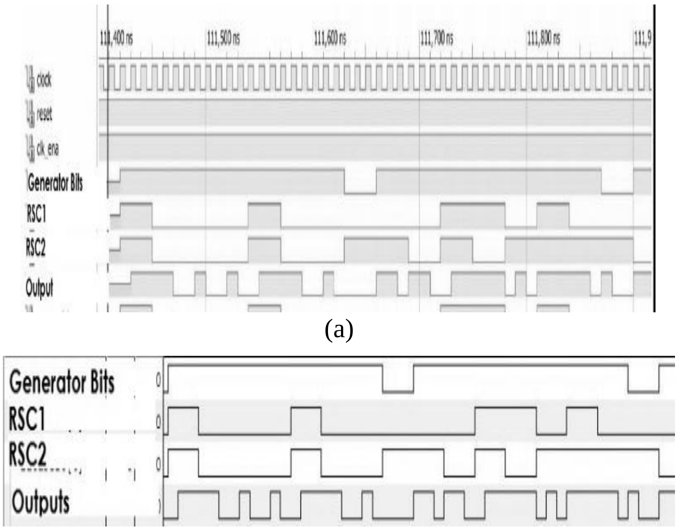
5. Parallel to serial

Parallel to serial block has clock period three times faster than the other blocks in the datapath sub-module. The input bits are from the bit generator, RSC1, and RSC2. Parallel to serial block has been implemented in FPGA according to system requirements. The output signal from ISim simulation and Chipscope gives the same output signal and it has period three times faster than the bit generator block, RSC1, and RSC2. This is shown in Fig. 11.

D. Top Module Design

Rate 1/3 Turbo Encoder 1784 bits design prototype has been successfully implemented on ATLYS Spartan-6 XC6SLX45 CSG324C board. It can be concluded rom the number of generated output bits by the system which is 5352 bits. From the simulation results, the system needs 11.043 clock delay. Fig. 12 shows that the 16-bits input sample is '1111111011111110' and generated codeword is '111 100 100 100 111 100 100 001 101 100 111 110 101 111 101 001'.

From Table I, logic utilization which is used by the system is no more than 100% so this system can be implemented on FPGA. The synthesization of Turbo encoder Top Module is shown in Fig. 13. It is shown that the RTL schematic result is the same as circuit model schematic in Fig. 4. From Fig. 13, this design consists of three sub-modules, control as control unit, clockdive as Clock Divider, and datapath1 as datapath.



(a)

(b)

Fig. 12. Simulation result (a) and Implementation result (b) of Turbo Encoder.

TABLE I
RESOURCE UTILIZATION ON FPGA

Logic Utilization	Used	Available	Utilization
Number of Slice Register	240	54576	0.0044 %
Number of Slice LUTs	865	27288	3%
Number of fully used LUT-FF pairs	225	892	25%
Number of bonded IOBs	4	218	1%
Number of BUFG/ BUFGCTRL/BUFHCEs	2	16	12%

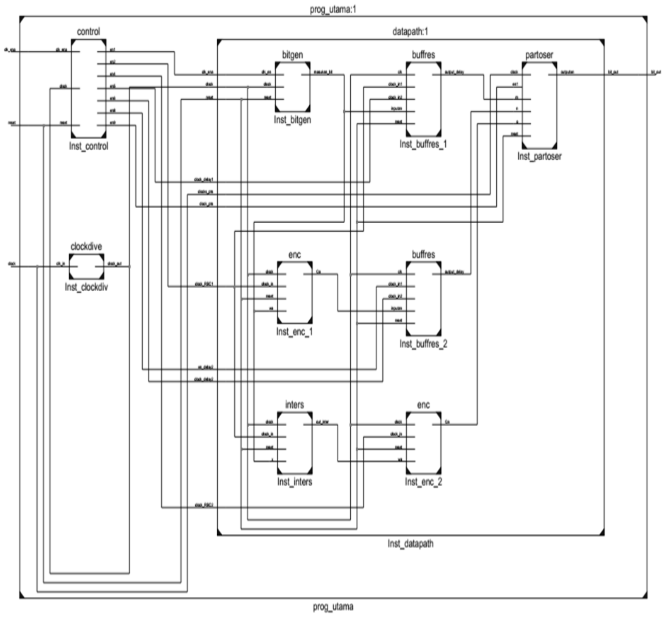


Fig. 13. RTL Schematic of Turbo Encoder Top Module

IV. Conclusion

The simulation results showed that to achieve performance at 10^{-6} of BER, turbo encoder with smaller rate needed to have greater coding gain than the turbo code with larger rate. Based on the simulation, rate 1/3 turbo encoder was implemented on the Spartan-6 FPGA ATLYS XC6SLX45 CSG324C. The FPGA resource utilization was 0.0044% Number of Slice Registers, 3% Number of Slices LUTs, 25% Number of fully used LUT-FF pairs, 1% Number of bonded IOBs, 12% Number of bufg / BUFGCTRL / BUFHCEs. This turbo encoder needed 11.043 clocks to process 1784 bits of data. The maximum frequency of the system was 169.122 MHz.

References

[1] L. Tianjun dan D. Wenrui , “Typical Channel Coding and Modulation ne for Satellite Communication System,” dalam *8th International osium on Antennas, Propagation and EM Theory, 2008. ISAPE 2008.* , ing, 2008.

[2] S. Haykin, Communication Systems 4th Edition, John Wiley & Sons, Inc, 2001.

[3] CCSDS, "CCSDS Recommended Standard For TM Synchronization And Channel Coding (bluebook)," *CCSDS Secretariat, Washington DC,* 2011.

[4] P. S. Bhanubhai, M. G. Shajan and U. D. Dalal, "Performance of Turbo Encoder an Turbo Decoder for LTE," *International Journal of Engineering and Innovative Technology (IJEIT),* vol. 2, no. 6, pp. 125-128, 2012.

[5] R. Z and A.-M. , "Implementation of Parallel and Pipelined Turbo Encoder for Real-Time Applications," *Journal of Engineering and Development*, vol. 13, pp. 127-142, 2009.

Laila Prakasita was born in Sleman, Yogyakarta, Indonesia, 15 March 1993. She received bachelor degree with cum-laude in telecommunication engineering from School of Electrical Engineering, Telkom University, Bandung, Indonesia in 2015.

Since 2013 until 2015, she joined with Nanosatellite Laboratory in School of Electrical Engineering Telkom University to develop Tracking Telemetry and Command subsystem, with main mission for remote sensing payload.

Heroe Wijanto (S’94–M’13) was born in Malang, East Java, Indonesia, 31 January 1968. He received bachelor degree (Ir.) in 1992, master degree (MT) in 2001, and then doctor degree (Dr.) in 2011, all of degrees are in telecommunication engineering from School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia.

He is a lecturer of telecommunication engineering and a researcher in antenna laboratory, Telkom University, Bandung, Indonesia since 1992. His interests and publications are in wireless communication systems, electromagnetics, antenna and propagation, pattern recognition, geoscience and remote sensing. Recently, he develops a nano satellite system for educational purposes.

Dr. Wijanto is vice rector in academics and information system in Telkom University, Bandung, Indonesia and is often asked by the national government for consultations on higher education and regulation in telecommunication.

Budi Syihabuddin (M’13) was born in Jakarta, Indonesia, 10 November 1985. He received his bachelor degree in 2008 and master degree in 2012, in telecommunication engineering from School of Electrical Engineering, Telkom University, Bandung, Indonesia.

In 2010, he joined Telkom University as a lecturer in telecommunication engineering and also as a researcher in microwave laboratory. His interests and publications are in RF microwave devices, wireless communication system and embedded system. He is co-chief engineer in the project for development a nanosatellite system for educational purposes.