Design for 32-Bit Parallel Polar Encoder Architecture

Praveen Kumar N1, Abhinav Ranjan2, Hemanth Kumar K S3

1PG Scholar, Digital Electronics, VTU, VTU Extension Centre, UTL technologies LTD, Bangalore
2 PG Scholar, Digital Electronics, VTU, VTU Extension Centre, UTL technologies LTD, Bangalore
3 Assistant Professor, Digital Electronics, VTU Extension Centre, UTL technologies LTD, Bangalore

Abstract - This paper work is entirely focused on implementation of 32-bit polar encoder by using the algorithm based on FFT (Fast Fourier Transform) method. Associated theory of Polar code and related architecture and application is briefly described in this paper. In this paper FFT algorithm for constructing Polar Encoder is highlighted in one of subsection. Polar encoder design is mainly targeted for Xilinx spartan6 FPGA (field programmer gate array) and designed using Verilog HDL (Hardware Description Language). Simulation result of the designed 32-bit polar encoder (by using Verilog HDL) is taken using Xilinx ISE (Integrated Synthesis Environment) 14.2 tool.

Key Words: Polar code, Polar Encoder, Folding Technique, Very-Large-Scale Integration (VLSI)

1. INTRODUCTION

The polar code was developed by Erdal Arikan. It belongs to the new class of error correction codes. The polar code approaches the channel capacity property which is asymptotical, and it is having good error correcting performance capabilities which is obtained for long polar code. The fully parallel architecture is intuitive and easy to implement, but it is not suitable for long polar codes due to much more hardware complexity. Hence the paper present the encoding process in the viewpoint of VLSI implementation and partially parallel architecture is proposed. The proposed encoder is highly efficient in implementing a long polar encoder, because it can achieve a high throughput with a smaller hardware complexity.

2. POLAR ENCODER

The encoding process is characterized by the generator matrix. The generator matrix $G_N$ for code word length $N$ or $2^n$ is generated by applying the $x^t$th Kronecker power to the kernel matrix

$$F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

For a generator matrix, the code word length is determined by $x = u \cdot G_N$, where $u$ and $x$ indicates information data and code word vectors, respectively.

Throughout this paper, it is assumed that the information data vector $u$ is arranged in a natural order, whereas code word vector $x$ is in the bit-reversed order to simplify the encoding process. A straightforward fully parallel encoding architecture is presented in [1], which has encoding complexity of $\mathcal{O}(n \log n)$ for a polar code of length $N$ and takes $n$ stages when $N = 2^n$. A polar code with a length of 32-bit is implemented with 80 XOR (exclusive-or) gates and processed with five stages, as shown in Fig. 1. In the fully parallel encoder, the whole encoding process is completed in a one clock cycle.

3. PROPOSED POLAR ENCODER

Partially parallel structure to encode long polar codes efficiently is described and shown in figure 1. To clearly show the proposed approach and how to transform the architecture, a 4-parallel encoding architecture for the 32-bit polar code and the fully parallel encoding architecture after transformation to a folded architecture is illustrated in depth as shown in figure 2.

3.1 FOLDED TECHNIQUE

The DFG (Data Flow Graph) of the 32-bit polar code is same as that of Fast Fourier Transform (FFT), and it shown in figure 3 and it uses the kernel matrix in the place of butterfly operation. The 4-parallel folded architecture is realized by placing two (2) functional units in each stage, since each of the functional units calculate two bits at a time. Considering the four parallel input sequences in normal order. The initial folding sets is given as: For stage 1: {P0, P2, P4, P6, P8, P10, P12, P14}. Here, the two functional units of stage 1 namely P0 and P1 perform concurrently at the beginning and P2 and P3 at the following clock cycle. The stage whose index $s$ is less than or equal to $\log_2 P$, where $P$ is the level of parallelism and has the same folding set as that of the earlier one. The stage 2 has the same order as those of
stage 1, because it performs the operation within the same four inputs. At future stages, the folding sets are calculated by the property that the functional unit that process a pair of inputs whose indices vary by \(2^{(s-1)}\) is exploited. Thus the folding set of stage 2 is given as \(\{Q_0, Q_2, Q_4, Q_6, Q_8, Q_{10}, Q_{12}, Q_{14}\}\), \(\{Q_1, Q_3, Q_5, Q_7, Q_9, Q_{11}, Q_{13}, Q_{15}\}\). In the stage 3, the indices of the two data vary by a factor of four, thus cyclic shifting of four bits right by one can be done by introducing a delay of one time unit. Thus the folding sets of stage 3 are given by \(\{R_{14}, R_0, R_2, R_4, R_6, R_8, R_{10}, R_{12}\}\), \(\{R_{15}, R_1, R_3, R_5, R_7, R_9, R_{11}, R_{13}\}\). The folding set of stage 4 and stage 5 is generated by cyclic shifting of stage 3 by two in order to enable full utilization of functional units with neighbouring iterations. The folding sets of stage 4 and stage 5 is given as \(\{S_{10}, S_{12}, S_{14}, S_0, S_2, S_4, S_6, S_8\}\), \(\{S_{11}, S_{13}, S_{15}, S_1, S_3, S_5, S_7, S_9\}\), \(\{T_2, T_4, T_6, T_8, T_{10}, T_{12}, T_{14}, T_{10}\}\), \(\{T_3, T_5, T_7, T_9, T_{11}, T_{13}, T_{15}, T_1\}\) respectively.

4. RESULT

The simulation results for 32-bit polar encoder which is obtained by simulating the Verilog encoder for the design and RTL schematic as shown in figure 4 and 5 respectively.
3. CONCLUSION

This paper minimizes the hardware resources for the 32 bit polar encoder. Many optimization techniques are implemented in steps to arrive at the proposed architecture for various folding levels. The simulation results show that the folded structure abides the polar encoder functionality and RTL schematic is obtained.

REFERENCES


