Performance Evaluation of Turbo Encoder Implementation on a Heterogeneous FPGA-CPU Platform Using SDSoC

Mohamed El Adawy¹, Ahmed Kamaleldin², Hassan Mostafa³ and Sameh Said
¹,²,³,⁴ Electronics and Communications Engineering Department, Cairo University, Giza 12613, Egypt.
Emails: {eladawy1988 & ah.kamal.ahmed}@gmail.com, hmostafa@uwaterloo.ca, msmsab@hotmail.com

Abstract - Recently, Developing Internet of Things (IoT) devices has been one of the most important products in the technology industry. Most of IoT products are designed based on System on chip (SoC) platform that allows producing faster and higher system level efficiency products. The Software Develop System-on-Chip (SDSoC) is a C/C++ development environment used to create hardware-software co designs on a heterogeneous FPGA-CPU platform. The questions of what platform and what implementation, whether hardware or software is best suited for more efficient platform. In this paper, these questions are sought to be answered through analysis scenarios of hardware and software implementation using SDSoC tool. This paper introduces a different platforms implementation for hardware-software co-design of turbo encoder synthesized by SDSoC. The turbo encoder widely used in various wireless communication standard such as the 3rd Generation Partnership Project (3GPP). The purpose of this paper is to select platform that achieves performance metrics such as area and power. In addition, new metric was defined to select platform that achieves the best performance.

Index Terms - IoT, SDSoC, Xilinx, FPGA, ZYNQ.

I. INTRODUCTION

Recently, The Internet of Things (IoT) is fueling innovation in nearly every part of our lives. The NarrowBand IoT (NB-IoT) is a narrowband radio technology standard developed for the Internet of Things (IoT) and standardized by the 3GPP [2]. The NB-IoT focuses on long battery life, low component cost and excellent indoor coverage. The NB-IoT enables large number of connected devices to the Internet. The NB-IoT is optimized for applications that need to communicate small amounts of data over long periods. The NB-IoT provides secure high quality of service communication because it operates in licensed spectrum. The fast growing of IoT devices promise to find rapid solution for developing IoT products to the markets. Xilinx announces SDSoC tool target developing IoT products. The SDSoC tool provides short design cycle to develop heterogeneous FPGA-CPU platform with simple interface logic generated by the tool to handle the data flow between hardware and software. In addition, SDSoC supports estimating performance, hardware utilization and latency calculations that makes developing design short and fast. A detailed description of the features of the tool is provided in [1]. The SDSoC tool integrated with a High-level synthesis (HLS) which used to implement hardware on FPGA synthesized from a C/C++ language description. Process of transforming C/C++ code to an RTL implementation using HLS is provided in [3]. Mobility is the most advantage of wireless communication, however, the wireless channel is more affected by noise and this leads to errors in data transmitting. In additions, Shannon calculated a theoretical maximum rate at which data could be transmitted over additive white Gaussian noise (AWGN) channel with an arbitrarily low bit error rate. For efficient transmission of data, the turbo codes were introduced whose performance in terms of Bit Error Rate (BER) is close to the Shannon limit [4].

In this paper, hardware-software co-design of turbo encoder synthesized by SDSoC for a heterogeneous FPGA-CPU platform was proposed. The rest of this paper is organized as follows. Section II gives an overview of the turbo encoder. Section III explains the implementation of turbo encoder using SDSoC tool. Section IV explains the experimental results. In the final Section V, the paper finishes with a conclusion and future work.

II. TURBO ENCODER

The design specs of the turbo encoder for LTE have been introduced in [5]. Fig. 1 shows the block of turbo encoder. The turbo encoder is the parallel concatenation of Recursive Systematic Convolutional (RSC) encoder, separated by an interleaver. The information bits flow goes into the first RSC encoder, and after interleaving, it feeds a second RSC encoder. The multiplexing and puncturing block accepts the information bits and outputs from RSC encoder to generate the cabled bits. The details of the design of the turbo encoder blocks are illustrated in [6].

Fig. 1: Turbo Encoder Block Diagram.
II. IMPLEMENTATION OF TURBO ENCODER

This section explains implementation of turbo encoder on a heterogeneous FPGA-CPU platform using SDSoC tool. The turbo encoder function is written using C programming language and integrated with other functions to verify operation of it. M. Rodriguez, E. Magdaleno, F. Perez and C. Garcia explain in details in [6] how to select independent functions to be implemented in hardware, how to write C/C++ code to be implemented as pipelined hardware and illustrates the design flow using SDSoC tool. Si-Dong Roh, Keol Cho and Ki-Seok Chung present in [7] comparing between optimized and non-optimized implementation of LDPC decoder on a Xilinx ZYNQ heterogeneous FPGA-CPU platform.

A. Turbo Encoder Implementation

The purpose of this paper is to implement multiple scenarios for turbo encoder function. Each scenario generates an Embedded FPGA platform which dependent on the implementation of turbo encoder sub-functions either software function or hardware-accelerated function synthesized by HLS. Table 1 shows all possible configuration scenarios to implement turbo encoder sub-function. For example, in turbo11 platform the Two RSC encoder and interleaver sub-functions are implemented as hardware-accelerated function and multiplexer-puncturing sub-function is implemented as software function.

Table 1: Turbo Encoder Sub-Function Configuration Scenarios:
(0) Software function and (1) Hardware-accelerated function

<table>
<thead>
<tr>
<th>platform name</th>
<th>Interleaver</th>
<th>Mux_punc</th>
<th>RSC_Enc 2</th>
<th>RSC_Enc 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>turbo0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>turbo1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>turbo2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>turbo3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>turbo4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>turbo5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>turbo6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>turbo7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>turbo8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>turbo9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>turbo10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>turbo11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>turbo12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>turbo13</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>turbo14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>turbo15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

B. Configurable Embedded FPGA Platform

Fig. 2 shows the proposed configurable Embedded FPGA platform. The configurable embedded FPGA platform consist of processing system and programming logic. The processing system side is consist of fixed implementation functions used for integration and verification the operation of the turbo encoder function. Example of fixed implementation functions, the random_test function used to generate random information bits, the noise function used to generate AWGN noise, and finally the main function which is integration all functions together. Also, the processing system is consist of the turbo_encoder function which is consist of the configurable implementation functions. The term configurable means that each sub-function of the turbo_encoder function is implemented as software function or hardware-accelerated function synthesized by HLS according to configuration in table 1.

The programming logic side consists of fixed implemented hardware logic used to handle the data flow between processing system and programming logic. The fixed logics are generated by SDSoC tool and dependent on the number of connection ports between software functions and hardware-accelerated functions. Also, the programming logic is consist of turbo_encoder function implemented using HLS. As described in the turbo encoder implementation section, each sub-function of turbo_encoder function is implemented as software function or hardware-accelerated function synthesized by HLS according to configuration in table 1.

IV. RESULTS AND COMPARATIVE STUDIES

This section shows the implementation results of all possible configurations scenarios shown in table 1. Sixteen projects are generated to cover all possible scenarios between software implementation and hardware-accelerated implementation generated using HLS. Xilinx ZYNQ ZC702 device was used for implementation. It consists of dual ARM Cortex A9 core as the processing system and XC7Z020-CLG484 based FPGA as the programming logic [7].

A. Hardware Utilization

Fig 3 shows the hardware utilization of the generated platforms of the synthesized hardware. The Hardware utilization is sum of number of Look-Up Tables (LUT), number of flip-flops and number of Muxes. The turbo0 platform is software implementation of all turbo encoder sub-functions so it has zero hardware utilization and not included in Fig 3. The turbo2 platform has the minimum hardware utilization. The turbo2 platform consist of RSC_Enc2 is implemented as hardware-accelerated function and other turbo encoder sub-functions are
implemented as software functions. The turbo13 platform has the maximum hardware utilization. The turbo13 platform consists of RSC_Enc1, interleaver and mux_punc are implemented as hardware-accelerated functions and RSA_Enc2 is implemented as software function. The turbo13 platform has the maximum hardware utilization because of the large number of functions in that scenario synthesized by HLS in addition, the sequence of the connection between software functions and hardware functions generate the maximum number of fixed logics to handle the data flow between the processing system and the programming logic.

B. Latency

Fig. 4 shows the latency calculations of the generated platforms of the synthesized hardware. The latency measure of delay in number of clock cycles. The turbo0 platform is software implementation of all turbo encoder sub-functions so the latency is not defined and not included in Fig 4. The turbo4 platform has the minimum latency. The turbo4 platform consists of mux_punc is implemented as hardware-accelerated function and other turbo encoder sub-functions are implemented as software functions. The turbo15 platform has the maximum latency. The turbo15 platform is hardware-accelerated implementation of all turbo encoder sub-functions. The turbo15 platform has the maximum latency because it is the scenario that generates the largest pipelined hardware design.

C. Dynamic Power

Power is an important metrics for any communication system. For FPGA platform, power calculation include the power consumption in the arm processor and the static power calculation. We focus on the dynamic power only, so we subtract the arm processor power and the static power from the total power consumptions. Fig. 5 shows the dynamic power for the generated platforms of the synthesized hardware. The dynamic power is measured in watts. The turbo0 platform is software implementation of all turbo encoder sub-functions so it has zero dynamic power and not included in Fig 5. The turbo2 platform has the minimum dynamic power. The turbo2 platform consist of RSA_Enc2 is implemented as hardware-accelerated function and other turbo encoder sub-functions are implemented as software functions. The turbo13 platform has the maximum dynamic power because the scenario that generates the largest hardware utilization design.

D. Energy

Energy is an important metrics as it enables faster design exploration with energy as an optimization metric. The energy is calculated by multiplying the latency by the dynamic power. Fig 6 shows the energy calculations of the generated platforms of the synthesized hardware. The energy is measured in number of clock cycle multiply by watts. The turbo4 platform has the minimum energy. The turbo4 consist of mux_punc is implemented as hardware-accelerated function and other turbo encoder sub-functions are implemented as software functions. The turbo15 platform has the maximum energy. The turbo15 is hardware-accelerated implementation of all turbo encoder sub-functions.
E. Hardware Acceleration

Hardware acceleration is metrics defined by SDSoC tool. Hardware acceleration is number of clock cycles improvement in execution of system if implementing function as hardware function in the programming logic. Fig 7 shows the hardware acceleration for the generated platforms of the synthesized hardware. The turbo0 platform is software implementation of all turbo encoder sub-functions so the hardware acceleration is not defined and not included in Fig 7. The turbo2 platform has the maximum hardware acceleration. The turbo2 consist of RSA_Enc1 and RSA_Enc2 implemented as hardware-accelerated functions and other turbo encoder sub-functions are implemented as software function. The turbo4 platform has the minimum hardware acceleration. The turbo4 platform consist of implementation of mux_punc sub-function is implemented as hardware-accelerated function and other turbo encoder sub-functions are implemented as software implementation.

![Fig. 7: Hardware Acceleration](image)

E. Figure of Merit

To know the platform that achieves the best overall performance, Figure of Merit (FoM) metrics was defined as follows:

\[
FoM = \frac{\text{(acceleration)}}{\text{(area)\,(power)\,(latency)}}
\]

Equation (1) shows that acceleration effect directly proportional with FoM performance and show that area, power and latency effect reversely proportional with FoM performance. Fig. 8 shows the FoM calculations of the generated platforms of the synthesized hardware. The turbo2 platform has the best FoM calculation. The turbo2 consist of RSA_Enc1 and RSA_Enc2 are implemented as hardware-accelerated functions and other turbo encoder sub-functions are implemented as software functions. The turbo7 platform has the worst FoM calculation. The turbo7 platform consist of RSA_Enc1, RSA_Enc2 and mux_punc are implemented as hardware-accelerated functions and interleaver is implemented as software function.

FoM shows that studying hardware acceleration, area, power and latency metrics alone is not sufficient to implement platform that achieves the best performance. FoM metric has very important value because it may help us to develop tools used to implement platform that targeting selected performance.

![Fig. 8: Figure OF Merit](image)

IV. CONCLUSION

This section shows results of all possible configuration scenarios for the turbo encoder implementation on a heterogeneous FPGA-CPU platform using SDSoC. Sixteen projects are generated to cover all possible scenarios between software implementation and hardware-accelerated implementation generated using HLS. The Xilinx ZYNC ZC702 device is used for implementation. Xilinx ZYNC ZC702 device consist of dual ARM Cortex A9 core as the processing system and XC7Z020-CLG484 based as the programming logic. This paper analyzes and presents comparative study for performance metrics such as area, power, latency, energy and hardware acceleration. In addition, FoM performance metrics was defined as metric for platform that achieves the best overall performance. Comparative studies help us to select suitable platform according to product specs. In addition, FoM metric help us to select platform that achieves the best performance. FoM metric may help us to develop tools used to generate platforms according to selected performance metric.

ACKNOWLEDGMENT

The authors would like to thank the Opto-Nano-Electronics laboratory (ONE Lab), Department of Electronics and Communications Engineering, Cairo University, Cairo, Egypt and the supporting and funding agencies. This work was supported in part by Cairo University, the Zewail City of Science and Technology, in part by AUC, in part by the STDF, in part by Intel, in part by Mentor Graphics, in part by ITIDA, in part by SRC, in part by ASRT, in part by NTRA, and in part by MCIT.

REFERENCES


[8] Si-Dong Roh, Keol Cho and Ki-Seok Chung, “Implementation of an LDPC decoder on a heterogeneous FPGA-CPU platform using SDSoC”, Region 10 Conference (TENCON), Nov. 2016, ISSN 2159-3450