Asynchronous full adders implementation: case study

V Gopeyenko[1]*, I Lemberski[2]

[1]Smart Technology Research Centre Ventspils University College, Ventspis, ISMA, Riga, Latvia
[2]International Radio Astronomy Centre, Ventspis University College, Ventspis, Baltic International Academy, Riga, Latvia

**Corresponding author's e-mail: viktors.gopejenko@isma.lv*

Abstract

Case study implementation a full 1- and 4-bit asynchronous adders in XILINX environment is discussed. The proposed adders can be realised based on WebPack application which has initially been developed for synchronous design. Our main goal is to examine XILINX environment as a tool for asynchronous design. An approach to asynchronous design has been proposed for this purpose.

Keywords: Full Asynchronous Adder, Xilinx, BLIF and EDIF Formats, Self-Timed Schematic.

1 Introduction

Asynchronous schemas work faster in comparison with the synchronous ones, however, there are special constraints for the realization. Furthermore, there are no special environments for asynchronous design. To overcome these problems a model for asynchronous schemes implementation using XILINX ISE WebPack Design software (WebPack) [1] has been suggested.

2 General

The main difference between synchronous and asynchronous schemas is as following: any port in synchronous schemas, e.g. Port A in Table 1 should be split in two Ports, e.g. A1 and A2. Value 1 for port A corresponds to combination 01 for asynchronous schemas and value 0 corresponds to combination 1. Combination 11 is prohibited and combination 00 is used to reset schemas.

TABLE 1 Synchronous and asynchronous logic relations

Synchronous Port	A		**1**	**0**	
Asynchronous Port	A1	0	**0**	**1**	1
Asynchronous Port	A2	0	**1**	**0**	1

The pairs of ports (A1 and A2) must be mapped on the same slice to provide asynchronous logic - this is a technological constraint.

To describe asynchronous schemas BLIF (Berkeley Logic Interchange Format) [1] is used, however, XILINX environment does not support this format. So, a temporary procedure has been suggested to solve this problem. The main idea of this procedure is the decomposition of functions, representation in BLIF format and mapping into LUT [3] (Look-Up-Table). Based on it schematic design is created. This is one of the conventional source types of the input data for WebPack application. The implementation (mapping, routing, slices distribution) is done using WebPack procedure under an user constraint file.

3 Implementation

A full 1- and 4-bit asynchronous adders were realised and simulated. The results of the time analysis show that asynchronous full 4-bit adder has value of maximum delay approximately the same as a full 1-bit synchronous adder [4].

4 Conclusion

XILINX WebPack environment can be used for asynchronous design. BLIF to EDIF (Electronic Design Interchange Format, which is one of Webpack input formats) converter should be created.

Acknowledgments

This work is supported by ERAF Project „Methods and Tools for the Design in Reconfigurable Environment" Nr. 1.1.1.1/16/A/234.

References

[1] *Berkeley Logic Interchange Format (BLIF)* University of California Berkeley July 28, 1992
[2] www.xilinx.com
[3] *Spartan6 FPGA Configurable Logic Block*, User Guide, UG384
[4] Bobbinpreet Kaur, Girdhari Agarwal, Amandeep Kaur 2015 Analysis and Implementation of a Full Adder Circuit using Xilinx Software *International Journal of Computer Applications (0975 – 8887) International Conference on Advancements in Engineering and Technology (ICAET 2015)* p.5-8