Adder Designing Process LUT based Using FPGAs

Deepak Sankhala, Yogita Taluja, Deepak Verma

Abstract— This paper presents a imagination of reconfigurable hardware appliance of the most basic operation of mathematics function addition on FPGA. Now a day field programmable gate arrays (FPGAs) have very huge and manifold logic resources resulting in the migration of their coating domain from image low and medium volume formation designing. In this work, we use an access to straightly map the design described in a high level package i.e. on FPGA platforms. Therefore, the concept which is using for addition it take major portion in all the digital designs and also many digital concepts are available to perform addition operation. Technology dependent optimizations are carried out to utilize this FPGA primitive efficiently and the result is compared against various adder designs. The fast carry chain propagation is reached by optimizing the use of 6-input LUTs together with the dedicated MUXCY resources available in the Virtex-5 FPGA chip**.** The state of processing adequate skill of any digital contemplation is determined on the basis of various parameters such as hold, power, space and time. . This paper proposes a fast adder structure for Xilinx Virtex-5 FPGAs In this paper we consider the mapping of arithmetic adders on look-up table (LUT) based FPGAs. Representing fact as they are to assign the given Boolean function into an look at the bright side of things So net list that can implement the desired working with minimum cost. We analysis and focus on 6-input LUTs that are inherent in all the modern day FPGAs.. This paper proposes a fast adder structure for Xilinx Virtex-5 FPGAs. The fast carry chain propagation is reached by optimizing the use of 6-input LUTs together with the dedicated MUXCY resources available in the Virtex-5 FPGA chip. Technology dependent optimizations are carried out to utilize this FPGA primitive efficiently and the result is compared against various arithmetic's operation designs

Index Terms TECSA (Time Efficient Carry Select Adder), Virtex, FPGA, MAC , Boolean logic functions, VHDL, multiplexers

1 INTRODUCTION

So for fast implementation processing and less delay time we explore the possibility of using a field programming logic array. Field Programmable Gate Arrays (FPGAs) combine limited cost and reconfigurability with very high make in to one unit facility and performances. Such characteristics, along with reduced price and make them a valid alternative to the more multifold and time to market demanding Application Specific Integrated Circuits (ASICs)

Sum of digit is the main operation of each arithmetic circuit, thus improving speed performances and reducing the area occupancy of adder circuits is still an initiative research topic . In digital Very Large Scale Integration (VLSI) Circuits, full adder forms are the basic building blocks for all arithmetic operations. Therefore, adder has the great impact in performance of the circuits, which are based on the arithmetic operations.

The various existing adder structures such as Ripple Carry Adder (RCA), Carry Look Ahead Adder (CLA), Carry Save Adder (CSA), Carry Select Adder (CSEL), Carry Bypass adder (CBY) and Area Efficient Carry Select Adder (AECSA) are analyzed based on the performance. Among all structures, some structures reduce the area occupied by the circuit with the increased delay and some structures reduce the delay with the increased consumption of area. The proposed adder structure results in optimized performance, that is, the delay is reduced with the equal consumption of area which was observed in normal adder design.

The characteristics of the digital circuit are analyzed mainly based on the time and area consumption. programmable gate arrays provide an alternative approach to application specific integrated circuits (ASIC) implementation with features like large-scale integration, design verification post production, lower non-recurring costs, reconfigurable design approach etc.

Field Programmable Gate Arrays (FPGAs) combine limited cost and reconfigurability with very high integration capability and performances. Such characteristics, along with reduced low volume costs make them a valid alternative to the more complex and time to market demand-

- *Deepak Sankhala, Yogita Taluja, and Deepak Verma are currently working as assistant professor in Electronics and Communication Engineering department at JECRC, Jaipur, India. E-mail: deepak.ece@jecr.ac.in* ing Application Specific Integrated Circuits (ASICs). Addition is the main operation of each arithmetic circuit, thus improving speed performances and reducing the area occupancy of adder circuits is still an enterprising research topic. Unfortunately, as it is well known, designing efficient adders using an FPGA platform is not trivial.

These blocks are highly optimized in terms of speed or area thereby facilitating efficient realization of complex functions .One of the major changes in the FPGA architecture has been the introduction of 6-input LUT as a logic element [11, 16]. With this FPGA primitive, the logic implementation would lead to higher logic densities resulting in a minimal-depth circuit and hence higher speed - a trend towards which the current FPGAs are oriented .Perhaps the biggest issue with 6-input LUTs is their underutilization while implementing a particular logic function, since many logic functions do not require six inputs .

2 DEFINITIONS AND TERMINOLOGY

ADDER ARCHITECTURES FOR VIRTEX-5 FPGAS Ripple-carry adders are implemented within a Virtex-5 FPGA device exploiting the generic configurable resources and the dedicated carry chain structures.
The latter speed-up the precise propagation course rout and allow high-speed to be got. Xilinx Virtex-5 devices reap fast carry chains by using multiplexers (MUXCY) and track resources that can be used to implement an n-bit efficient addition circuit as delineate , where A and B are the two digit and C0 is the carry-in. The generic sum bits Si (0≤i<n) is calculated, as reported by XORing the bit propagate pi=Ai⊕Bi with the carry bit Ci computed as given .The utility of this adder structure is easily to understood when the it will take time of the set apart carry chain is compared to the generic LUT . For example, in the Virtex-5 family, the single bit sum generation stage (horizontal signal propagation through the LUT and the XORCY) is about 17 times slower than the single bit carry propagation stage (vertical propagation along the generic MUXCY).

It clear that any strive to reform performances of adder architectures using digital logic and routing resources an ineffective. We want to improve process of most arithmetic operations with respect to their corresponding designing through generic configurable logic, embedded DSP48E slices with high speed features are also made available within the Virtex-5 devices. DSP slices include dedicated arithmetic circuits, optimized to fasten addition, multiplication, accumulation, MAC (Multiply And ACcumulation) and Boolean logic functions.

A combinational circuit that performs the addition of three bits are called full adder. Two of the input variables, denoted by a and b, represents the two significant bits to be added. The third input, c, represents the Carry from the previous least significant position. The two output variables are designated as Sum and Carry. The output variables are determined from the arithmetic Sum of the input bits. When all the input bits are zero, the output is „0‟. The output Sum is equal to „1‟ when only one input is equal to „1‟ or all the three inputs are equal to „1‟.

Based on the truth table, the output Sum and Carry are described

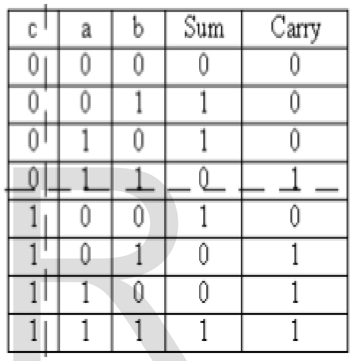
as

Sum = a + b + c

Carry = a.b + b.c + c.a (2)

In equations (1) and (2), ⊚ represents Ex-Or operation and represents AND operation. Design and Implementation of Time Efficient Carry Select Adder

Table 1: Truth Table of Full Adder



c	a	b	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The Conventional 1- bit adder circuit is shown in Figure1. It requires 6 gates (4 logic gates and 2 Ex-Or gates) to implement 1- bit adder circuit.

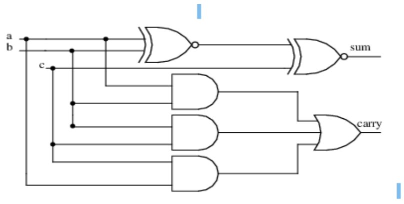


Figure 1: Conventional 1 – bit adder

Ripple Carry Adder The addition of n- bit number can be done by connecting the n – bit full adders in cascade in which the Carry output from each full adder connected to the Carry input of the next full adder and it is shown in the Figure 2. The main drawback of this structure is that the delay increases with the number of bits.

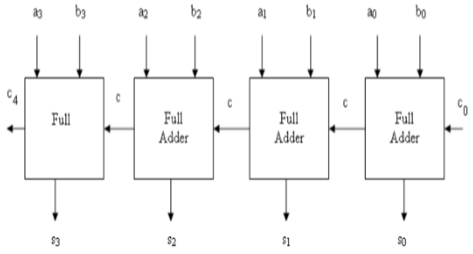


Figure 2: Ripple Carry Adder

It can be seen that each DSP48E is configured to perform the addition between the 48-bit input C (that receives the external operand B) and the concatenation of the 30-bit A and 18-bit B inputs (that receive the most significant and the least significant bits of the external operand A, respectively). In order to do this, the ALUMODE and the OPMODE attributes were set to “0000” and “0001111”

Since design entry is the only manual phase in the FPGA design flow, we try to control the mapping of the Boolean networks at the design entry step only. This involves modifying the coding style and writing VHDL codes for optimized Boolean networks based on direct instantiations of the targeted circuit elements. This is in contrast to the conventional coding styles that are typically behavioral and rely completely on the synthesizer to map the Boolean network by inferring the logic.

3 Methodology

The implementation in this work targets the XC5VLX303FF324 device from Xilinx Virtex-5 FPGA family. The implementation is carried out for an input word-length varying from 8 to 64 bits. The parameters considered are area, timing and dynamic power dissipation. Area is considered in terms of the number of occupied slices. Timing analysis may be static or dynamic. Static timing analysis gives information about the delay associated with the critical path and the maximum frequency at which the design may be operated. Dynamic timing analysis verifies the functionality of the design by applying test vectors and checking for correct output vectors.

3.1 Adder architectures for virtex-5 fpgas

Ripple-carry adders are implemented within a Virtex-5 FPGA device exploiting the generic configurable resources and the dedicated carry chain structures. The latter speed-up the critical propagation path and allow high-speed to be reached. Xilinx Virtex-5 devices realize fast carry chains by using dedicated multiplexers (MUXCY) and routing resources that can be used to implement an n-bit efficient addition circuit as depicted in Fig.1, where A and B are the addends and C0 is the carry-in. The generic sum bits Si (0≤i<n) is calculated, as reported in (1), by XOR-ing the bit propagate pi=Ai⊕Bi with the carry bit Ci computed as given

The advantage of this adder structure is evident when the delay of the dedicated carry chain is compared to the generic LUT fabric. For example, in the Virtex-5 family, the single bit sum generation stage (horizontal signal propagation through the LUT and the XORCY) is about 17 times slower than the single bit carry propagation stage (vertical propagation along the generic MUXCY). It is then clear that any attempt to improve performances of adder architectures using generic logic and routing resources would be ineffective.

3.2 FPGA Implementation of Adder

In this proposed work of Radix -4, 8-Point, DCT with XOR MUX based Adder, Multiplier and Subtractor will be implemented in Xilinx FPGA S6LX9-2TQG144 using VHDL language and compares all the parameters in terms of delay, area and power consumptions. This novelty based XOR-MUX DCT operations will take less logic size when compared to parallel adder of Ripple carry Adder with Binary Excess one converter. The simulation results of DCT Operation are implemented in Modelsim and proved for the compression ratio. Here input signal is data_re and data_im, these two are differential inputs ,it receives the audio signal from a .hex file with the help of MATLAB Software. Fig.8 shows the MATLAB GUI - Audio to Hex Conversion and Hex to Audio Conversion.

3.2 1 Mapping Your Deign Circuit to FPGA

- The first step is to make or draw plans for something a simple “1-Bit Adder” circuit.
- In the next step, you will instantiate 4 part of the 1-bit adder to implement a “4bit Adder”.
- Then, you will learn how to use Xilinx Vivado for writing Verilog and how to connect to the board.
- Finally, you will program the FPGA and get the circuit running on the FPGA board.
- Follow the instructions. Paragraphs that have a gray background like the current paragraph denote descriptions that require you to do something.
- To complete the lab you have to show your work to an assistant before the deadline, there is nothing to hand in. The required tasks are clearly marked with gray background throughout this document. All other tasks are optional but highly recommended. You can ask the assistants for feedback on the optional tasks.

3.2.2 Starting software and Creating a Project

- Start Vivado: Click on the Start menu and go to
- Programs → Xilinx Design Tools → 2016.4. 2.
- Create a ‘New Project’: Inside, choose File → New Project. This will bring up a new project wizard.

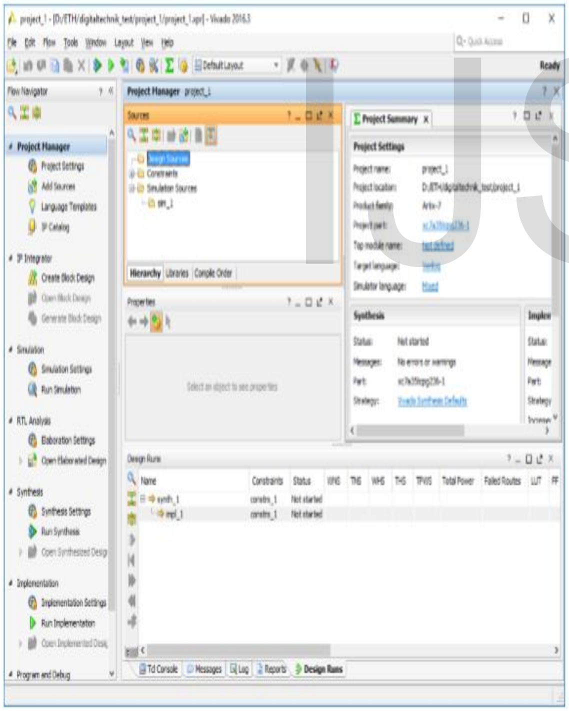
  Write down the “Project name”. You can simply call it Lab2. You can also specify where the project files will be stored using the “Project location.” Click next. –
- Select “RTL Project” as the project type and click next. –

- In the “Add Source” dialog, we do not have source files yet so just click next.
    - In the “Add Existing IP” discourse, click next. - In the “Add Constraints” dialog, click next.

In “Default Part”, we need to select the FPGA board that we are using. Type “xc7a35tcpg236-1” in the Search field and you should be left with only one option. Select and click next. –

- In “New Project Summary”, you can see the project configuration. Make sure that the default part and product family information reads:
- Default Part: xc7a35tcpg236-1
- Product: Artix-7
- Family: Artix-7
- Package: cpg236 Speed

Once everything the act check put procedure, click finish to create the project! After that .you should see the following view:



project_1 - [D:/ETH/digitaltechnik_test/project_1/project_1.xpr] - Vivado 2016.3

Project Settings

Project name: project_1
Project location: D:/ETH/digitaltechnik_test/project_1
Product family: Artix-7
Project part: xc7a35tcpg236-1
Top module name: not defined
Target language: Verilog
Simulator language: Mixed

Synthesis

Status: Not started
Messages: No errors or warnings
Part: xc7a35tcpg236-1
Strategy: Vivado Synthesis Defaults

Design Runs

Name	Constraints	Status
synth_1	constrs_1	Not started
impl_1	constrs_1	Not started

4 Processor and implement a Full Adder in Verilog

First, we will need to identify clearly and definitely the input and output of the full adder. Complete the input/output list of the Full Adder module. Your logic should look something like:

Module Full Adder(input p, input q, input c, output s, output o);

<Your logic here> end module

Implement the full adder logic inside the Full Adder module.

Running Your 4-Bit Adder on the Basys 3 FPGA Board

Now use FPGA board and program it of designing the 4 bit adder

. An FPGA is a programmable chip that concede the person to map a given arthematic (essentially a net list of interconnected gates) in itself. Depending on the type an FPGA can map anywhere from a few hundred to more than a million gates.

REFERENCES

[1] "Hardware Implementation of Discrete / Inverse Discrete Cosine Transform using Redundant Number System CORDIC Processor", Jayashri M.Rudagi, Dr. (Smt.) Shaila Subbaraman. International Journal of Applied Engineering Research, Volume 13, Number 4, 2018

[2] "Multiplierless 16-point DCT approximate for low complexity image and video coding", Thiago L.T. da Silveira, Raiza S. Oliveira, Fabio M.Bayer, Renato J.Cintra, Arjuna Madanayake, Springer-Verlag London 2016

[3] "FPGA Implementation of XOR-MUX Full adder based DWT for signal processing applications", P.Radhakrishnan, G.Themozhi, Chennai, India. Elsevier, Science Direct, Volume 73, March 2020,

[4] S. Himanshu, Markandeya and Kaushik Roy, Fellow, IEEE, “Low-Power System for Detection of Symptomatic Patterns in Audio Biological System,” IEEE Trans. on Very large scale integration (VLSI) Systems, vol. 24, No.8, AUGUST 2016. pp. 1063-8210.

[5] "Design of Reversible Adders Using A Novel Reversible BKG Gate", Bhuvana B, KanchanaBhaaskaran V S, 2016 Online International Conference on Green Engineering and Technologies (IC-GET)

[6] A. Cosoroaba “Achieve Higher Performance with Virtex-5 FPGAs”, XCell Journal, Fourth Quarter 2006, pp. 16-18.

[7] V. Jindal, A. Agarwal, “Carry circuitry for LUT-based FPGA”, Proceedings of 17th International Conference on VLSI Design, pp. 731 – 734, 2004

[8] E. Ahmed, J. Rose, “The effect of LUT and cluster size on deepsubmicron FPGA performance and density”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.12, Issue 3, pp. 288 – 298, March 2004.

[9] Virtex-5 User Guide, UG190 (v3.1) September 11, 2007, available at www.xilinx.com.