

An Effective Method for Implementing 64-bit MAC using Wallace Tree Multiplier on FPGA through Chip Scope Pro

Prathap Soma¹, Rajeshwari Soma²

¹CVR College of Engineering, ECE Department, Hyderabad, India

Email: prathap.soma@gmail.com

²CVR College of Engineering, ECE Department, Hyderabad, India

Email: rajeshwari.473@gmail.com

Abstract—Most of the Digital Circuit Designers are failing at programming the FPGA when the net list of design exceeds specified NETs of the desired FPGA. This paper presents the simplest way to programming the FPGA is not only programming, but also testing Circuit Under Test (CUT) by applying all possible test vectors manually through the keyboard. Presently running On-Board values can also be viewed on the monitor. If designers are new to FPGA design, this paper will help them to learn some of the debugging options.

Index Terms- Chip scope pro, ILA core, Vio core, ICON.

I. INTRODUCTION

The MAC (Multiplier-Accumulator) unit is made up of multiplier and an accumulator. This contains the sum of previous successive products. The multiplier used in this is a Wallace Tree Multiplier and the adder is Carry Save Adder (CSA). MAC inputs are obtained from the memory location and given to the multiplier block. Since it is a 64 bit MAC [1] unit, multiplier inputs are 64 bits. Therefore the MAC unit requires 128 bit adder and 129 bit accumulator.

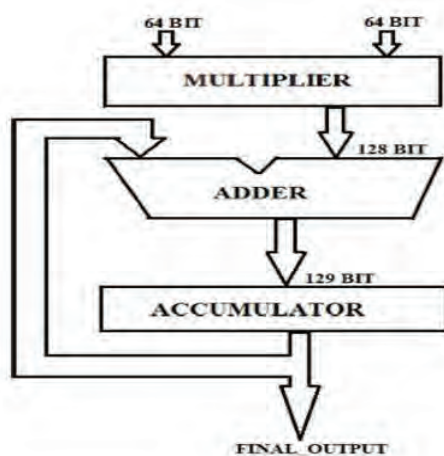


Figure 1.1 64-bit MAC Block Diagram

With the recent expeditious advances in communication systems and multimedia, real time signal processing like audio, video and image signal processing, large-capacity data processing are increasingly being required. In computing, generally the multiply-accumulate operation is a rarely used step. It computes the product of two numbers and adds that product to an accumulator. The hardware unit which performs the operation of both multiply and accumulates known as a multiplier-accumulator (MAC, or MAC unit) the operation itself is called a MAC or MAC operation.

The principle of multiplication of the MAC unit is based on twofold method, i.e. it evaluates partial products (pp) and the summing process takes place with shift partial products.

The least significant bit (the right most bit) of the multiplier is then multiplied with the multiplicand bit, corresponding partial products are stored in a register. The same process is repeated until the most significant bit (left most bit) is reached. The process of multiplication is shown in fig. 1.2.

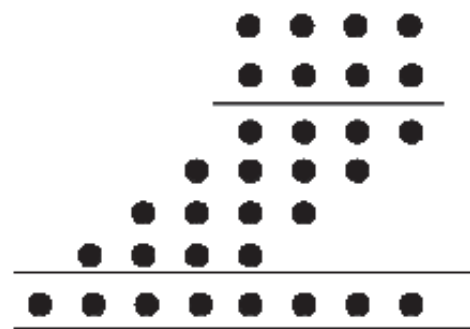


Figure 1.2 Basic Multiplication

A MAC unit consists of a multiplier and an accumulator containing the sum of the previous successive

products. The multiplier used is Wallace Tree Multiplier [2] and the adder used is Carry Save Adder [3]. The MAC inputs are obtained from the memory location and given to the multiplier block.

The multiplier used in this is a Wallace Tree Multiplier. It is an efficient multiplier with reduced complexity as compared with the conventional multiplier. The power consumption of Wallace Tree Multiplier is very less. The multiplication process using Wallace Tree Multiplier is shown in fig 1.3.

It uses a tree structure and reduces the number of additions in the critical path to $O(\log n)$ rather than $O(n)$. The number of partial products in Wallace Tree Multiplier are reduced by the use of half and full adders in the design. It also uses compressors to reduce the complexity.

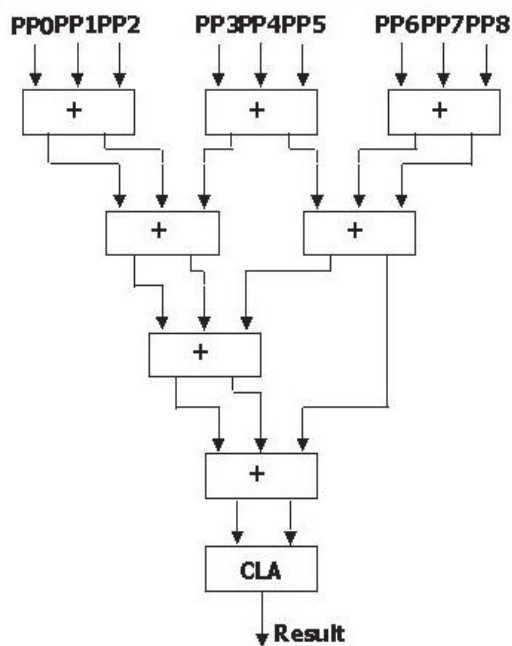


Figure1.3. Wallace tree multiplier

II. NEED OF CHIP SCOPE PRO

‘Chip Scope’ is a set of tools designed by Xilinx that allows probing the signals, i.e., testing the signals of design inside an FPGA, much as would do with a logic analyzer. For example, while design is running on the FPGA, the designers can trigger when required events are taking place and can view any internal signal of the design. Since, the ‘Chip Scope’ analyzer’s logic is implemented in the FPGA, it has some important limitations.

The sample memory of the analyzer is confined by the memory resources of the FPGA. In a design that uses much of the FPGA’s memory, there may not be much of memory left for the ‘Chip Scope’ cores [4]. Also, ‘Chip Scope’ cannot sample as quickly as external logic analyzer. Generally, the Chip Scope sampling rate will be the same as the design’s clock frequency. It is therefore not possible to detect glitches. In order to use the Chip Scope internal logic analyzer in an existing design project, first generate

the Chip Scope core modules, which perform the trigger and wave form capturing functionality on the FPGA. Afterwards, initiate these cores in Verilog or VHDL code, and connect those modules to the signals that are to be monitored. The complete design is then recompiled, instead of loading the resulting “.bit” file onto the FPGA using iMAPCT. The ‘chip scope’ analyzer also provides the interface for setting the trigger criteria for the Chip Scope cores, and displays the waveforms recorded by those cores.

III. CORE GENERATOR FLOW

There are certain rules and regulations to add the cores for a design. These rules and regulations are to be properly maintained, to configure the design to FPGA board.

They are

1. Configuring the design
2. Creating cores
3. Adding the cores
4. Configuring.

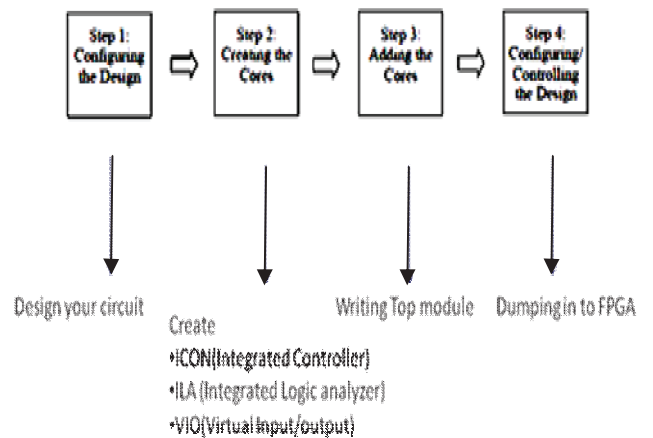


Figure3.1 Core generation flow

Step1: Configuring the design: Designer should complete the circuit with all aspects like power optimization, area optimization.

Step2: Creating cores: Like ILA, VIO& ICON cores need to be created. The creation of core of a design particularly depends on the number of inputs and outputs.

Step3: Adding cores: Top level program in structural manner either Verilog or VHDL language.

Step4: Configuring: i.e. programming FPGA and verifying.

IV. IDEA BEHIND THE CHIPSCOPE PRO

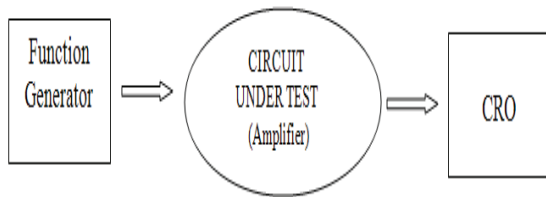


Figure 4.1 Idea behind the chip scope pro

In case of hardware, the designer tries to check the circuitry, whether it meets design constraints or not with the help of FG (Function Generator) and CRO (Cathode Ray Oscilloscope).

In case of software(VHDL or Verilog), same digital circuits can be tested using ‘Chip Scope Pro’ irrespective of the number of inputs and outputs of the design. This can be done by adding cores to the design and that cores are ILA (Integrated Logic Analyzer), VIO (Virtual Logic Input/output) & ICON (Integrated Controller).

V. DESIGN AND IMPLEMENTATION OF CORE GENERIC FLOW

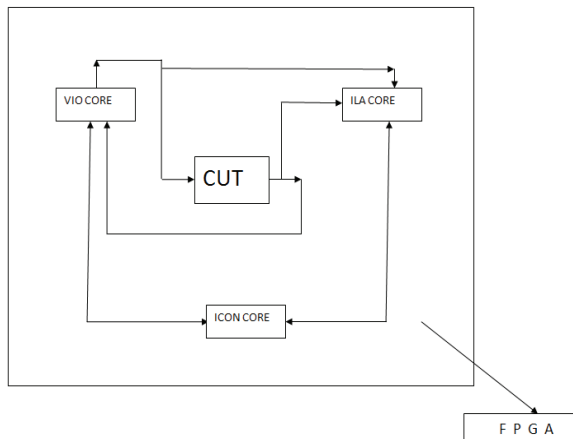


Figure 5.1 Arrangement of added cores to FPGA

From the steps mentioned above, first there is a need to add the core to design circuitry or CUT (Circuit Under Test).The cores are ILA (Integrated Logic Analyzer), VIO (Virtual Logic Input/output) and ICON (Integrated Controller). After that, writing the top level module either in Verilog or VHDL [5][6].Fig.5.1shows the basic idea of writing a top level module to the circuit.

VI. ANALYSIS PROCEDURE

ICON core insertion:

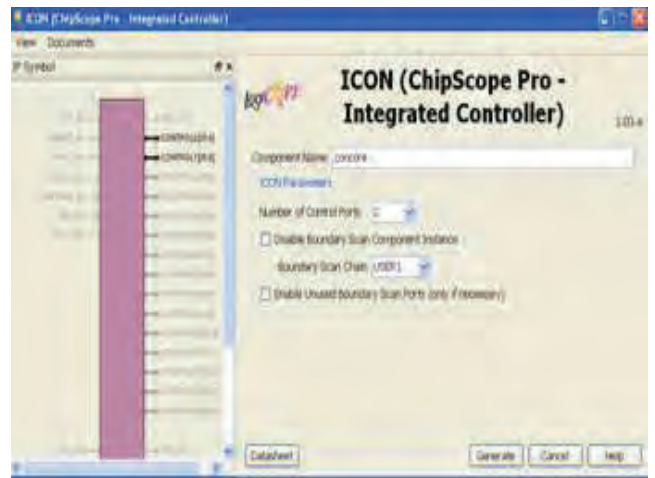


Figure 6.1 ICON Core insertion

Usually controller is useful for controlling the design. Here the ICON is also working same as Controller in general design. Here it controls 15 numbers of cores, whereas this MAC unit requires two control signals with each 36 bits wide.

VIO core insertion:

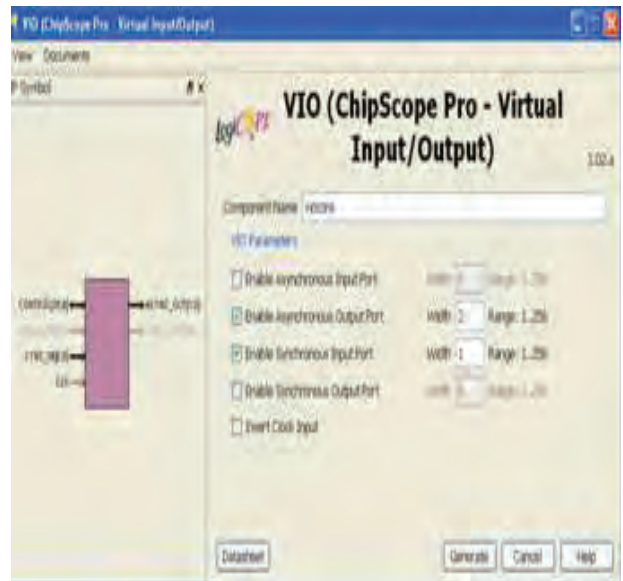


Figure 6.2 VIO Core Insertion

It is basically used for generating Inputs to MAC unit. Sometimes called it as test vector generator for the circuit under test. The total number of inputs required for this core is 129 and possible generation of test vectors is 128.These 128 TV's are fed to the input of 64-bit MAC unit.

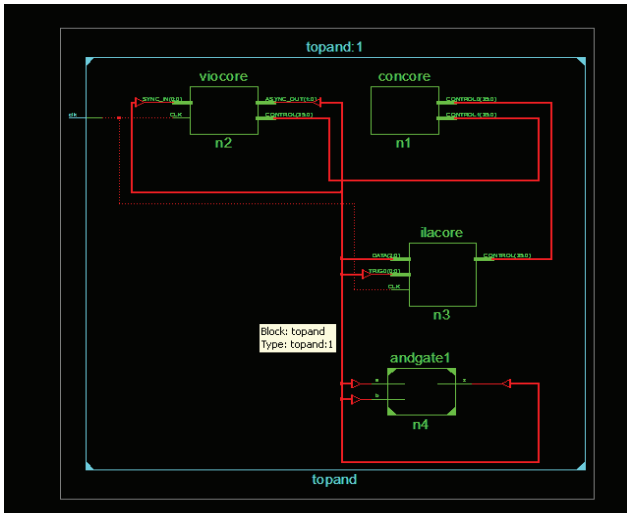


Figure 7.3 RTL Schematic of MAC unit after adding cores.

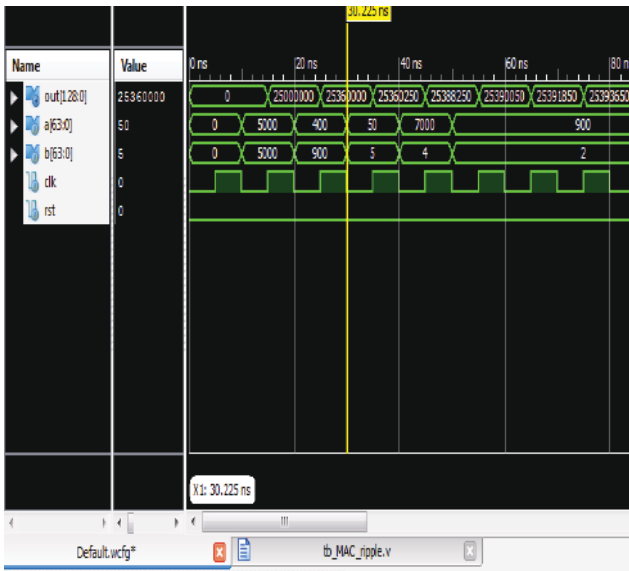


Figure 7.4 Simulation results of MAC unit before programming.

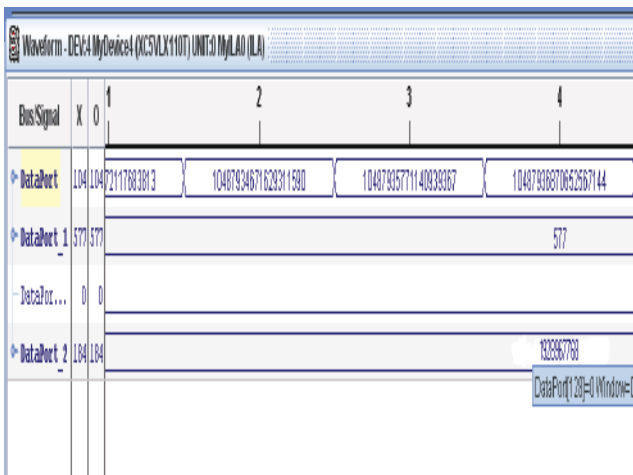


Figure 7.5 Simulation results of MAC unit after programming using chip scope pro.

VIII CONCLUSIONS

It is concluded that, using ‘chip scope pro’ one can analyze any circuit with any number of inputs. With Spartan 3E FPGA only four signals can be analyzed, but by interfacing with ‘Chip scope pro’ it can be increased up to 256. Similarly, Virtex 5 ranges to 1024 from its 8 pins.

REFERENCES

- [1]. Fast Multipliers-Pipeline Wallace by Anagha Patwardhan in Southern Illinois University Carbondale, 2007.
- [2]. http://access.ee.ntu.edu.tw/course/VLSI_design_89second/course_outline/8.2.8%20Wallace%20Tree%20Multiplier%2005-31-2001.pdf
- [3]. Arithmetic and Logic in Computer Systems By Mi Lu, 2010.
- [4]. http://www.xilinx.com/itp/xilinx10/isehelp/ise_c_process_analyze_design_using_chipscope.htm
- [5]. Digital Systems Design With Vhdl And Synthesis: An Integrated Approach by John Wiley & Sons, 2007.
- [6]. http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_6/ug750.pdf