FPGA Implementation of DA Algritm for Fir Filter

[1,]Solmanraju Putta, [2,]J Kishore, [3,]P. Suresh
[1,]*M.Tech student,Assoc. Prof.,Professor Dept of ECE,Sunflower college og Engg.&Technology*
Challapalli, Krishna (D), A.P - 521131

ABSTRACT

MAC is composed of an adder, multiplier and an accumulator. Usually adders implemented are Carry- Select or Carry-Save adders, as speed is of utmost importance in DSP. One implementation of the multiplier could be as a parallel array multiplier. The inputs for the MAC are to be fetched from memory location and fed to the multiplier block of the MAC, which will perform multiplication and give the result to adder which will accumulate the result and then will store the result into a memory location. This entire process is to be achieved in a single clock cycle. The architecture of the MAC unit which had been designed in this work consists of one 16 bit register, one 16-bit Modified Booth Multiplier, 32-bit accumulator. To multiply the values of A and B, Modified Booth multiplier is used instead of conventional multiplier because Modified Booth multiplier can increase the MAC unit design speed and reduce multiplication complexity. SPST Adder is used for the addition of partial products and a register is used for accumulation. The product of A_i X B_i is always fed back into the 32-bit accumulator and then added again with the next product A_i x B_i. This MAC unit is capable of multiplying and adding with previous product consecutively up to as many as times.

I. INTRODUCTION

Due to the intensive use of FIR filters in video and communication systems, high performance in speed, area and power consumption is demanded. Basically, digital filters are used to modify the characteristic of signals in time and frequency domain and have been recognized as primary digital signal processing. In DSP, the design methods were mainly focused in multiplier-based architectures to implement the multiply-and-Accumulate (MAC) blocks that constitute the central piece in FIR filters and several functions. The FIR digital filter

is presented as:

$$y[n] = \sum_{k=0}^{N-1} c[k]x[n-k]$$

Where y[n] is the FIR filter output, x[n − k] is input data and c[k] represents the filter coefficients Equation(1) shows that multiplier-based filter implementations may become highly expensive in terms of area and speed. This issue has been partially solved with the new generation of low-cost FPGAs that have embedded DSP blocks. The advantages of the FPGA approach to digital filter implementation include higher sampling rates than are available from traditional DSP chips, lower costs than an ASIC for moderate volume applications, and more flexibility than the alternate approaches.

II. DISTRIBUTED ARITHMETIC (DA)

Distributed arithmetic (DA) is an important FPGA technology. It is extensively used in computing the sum of products

$$y[n] = < c, x > = \sum_{n-0}^{N-1} c[n]x[n]$$

DA system, assumes that the variable x[n] is represented by-

$$x[n] = \sum_{b=0}^{B-1} x_b[n] \times 2^b, x_b[n] \in [0,1]$$

If c[n] are the known coefficients of the FIR filter, then output of FIR filter in bit level form is:

$$y = \sum_{n=0}^{N-1} c[n] \times \sum_{b=0}^{B-1} x_b[n] \times 2^b$$

In distributed arithmetic form

$$y = \sum_{b=0}^{B-1} 2^b \times \sum_{n=0}^{N-1} f(c[n], x_b[n])$$

In Equation (5.5) second summation term realizing as one LUT. The use of this LUT or ROM eliminates the multipliers [9]. For signed 2's complement number output of FIR filter can be computed as-

$$y = -2^B \times f(c[n], x_B[n]) + \sum_{b=0}^{B-1} 2^b \times \sum_{n=0}^{N-1} f(c[n], x_b[n])$$

Where B represents the total number of bits used. Fig 4.1 shows the Distributed architecture for FIR filter and different with the MAC architecture. When x[n] <0, Binary representation of the input is [10],

$$x[n] = -x_b[0] + \sum_{b=1}^{B-1} x_b[n] 2^{-b}$$

The output in distributed arithmetic form-

$$y = -(\sum_{n=0}^{N-1} c[n] x_b[0]) + \sum_{b=1}^{B-1} (\sum_{n=0}^{N-1} c[n] x_b[n]) 2^{-b}$$

If the number of coefficients N is too large to implement the full word with a single LUT (Input LUT bit width = number of coefficients), then partial tables can be and add the results as shown in Fig 4.2. If pipeline registers are also added, then this modification will not reduce the speed, but can dramatically reduce the size of the design

III. IMPLEMENTATION

High Level Specifications are nothing but the requirements to understand and begin the design. In this stage the designer main aim is to capture the behavior of the design using mostly behavioral constructs of the HDL's. The next step after capturing the designs functionality is to segregate the design in all possible ways and try to write a synthesizable code which infers available primitives from the library. Here mostly the mixed style of modeling is used and only synthesizable constructs of the HDL is used.Then comes the synthesis step which is actually target driven. Here we have an FPGA as the target device. Then Implementation is nothing but the process of placing and routing the design on an FPGA. Mostly it is a tool driven and no manual intervention of the designer is required. Designer only needs to specify constrain file in design if any.Below Flow chart shows this in brief.

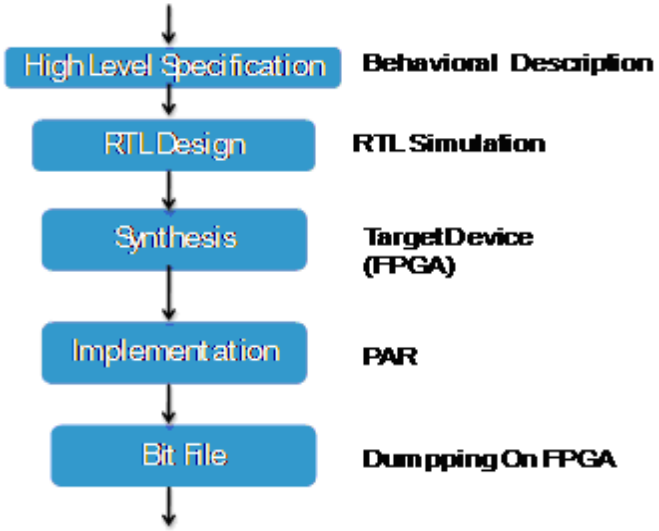


Fig: Design Flow Chart

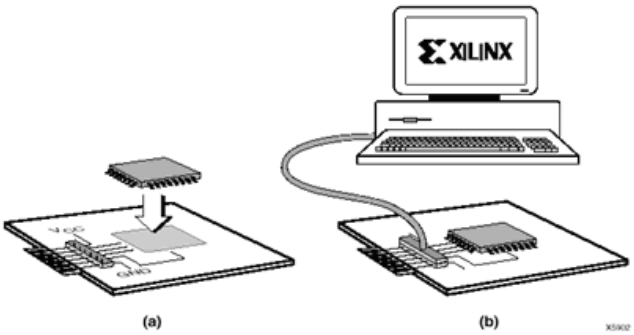


Fig: Bit File Burning

The Final stage after doing the place and route successfully and fully satisfied with mapping and other reports is the bit file generation phase. This bit file has to be downloaded on to FPGA via a JTAG cable. Above Figure shows a simple setup for this.

IV. TARGET DEVICE

Xilinx Spartan3E XC3s100E Device

The below Figure shows the Board with all the peripherals connected to it.The FPGA used belongs to Spartan 3E family and the device is XC3s100E.

Fig: Spartan kit

V. PROGRAMMING

After successfully compiling an FPGA design using the Xilinx development software, the design can be downloaded using the impact programming software and the USB cable. To begin programming, connect the USB cable to the starter kit board and apply power to the board. Then, double-click Configure Device (impact) from within Project. As shown in the figure

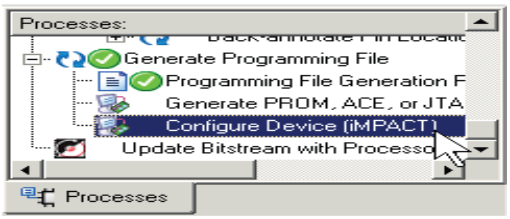


Fig: Configure Device.

If the board is connected properly, the impact programming software automatically recognizes the three devices in the JTAG programming file, as shown in Figure 5.7. If not already prompted, click the first device in the chain, the Spartan-3E FPGA, to highlight it. Right-click the FPGA and select assign New Configuration File. Select the desired FPGA configuration file and click OK. As shown in the fig 5.7 If the original FPGA configuration file used the default Startup clock source, CCLK,IMPACT issues the warning message shown in Figure 5.8. This message can be safely ignored. When downloading via JTAG, the IMPACT software must change the Startup clock source to use the TCK JTAG clock source.

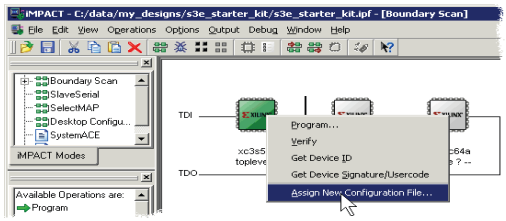


Fig: Assigning configuration File

In above figure we need to assign the correct bit stream file generated to the device and then locate the path of the bit file in the system and then check the box.

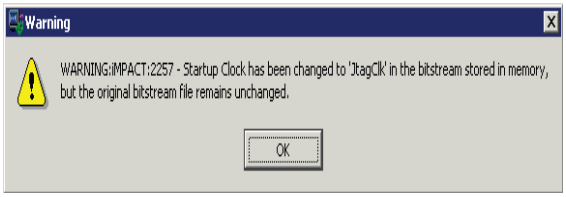


Fig: IMPACT Issues a Warning if the Startup Clock Was Not CCLK
This warning is ignored and checked to continue the burning of bit file to the FPGA.

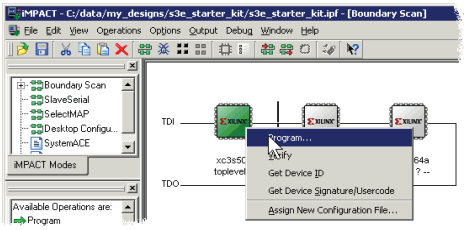


Fig: Program FPGA

When the FPGA successfully programs, the iMPACT software indicates success, as shown in below Figure. The FPGA application is now executing on the board and the DONE pin LED lights up.

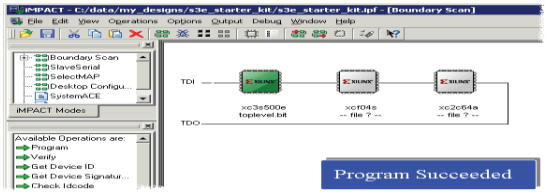
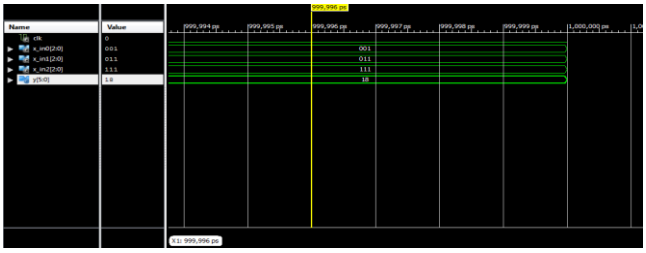


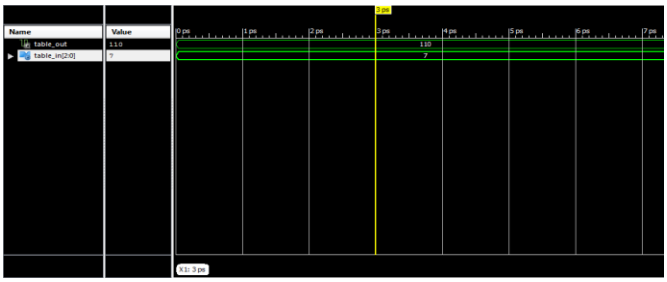
Fig : Successfully Programmed

VI. SIMULATION

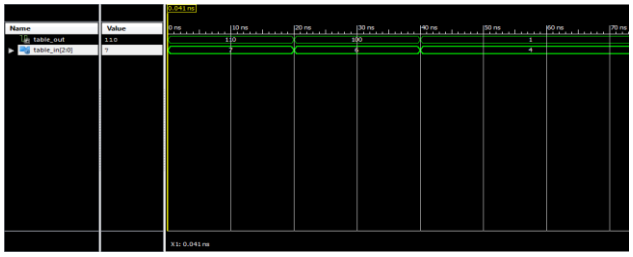
Iteration 1



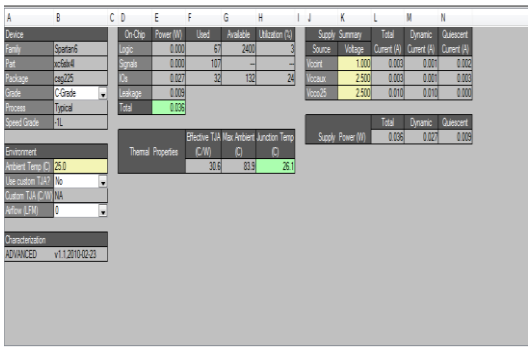
Iteration 2



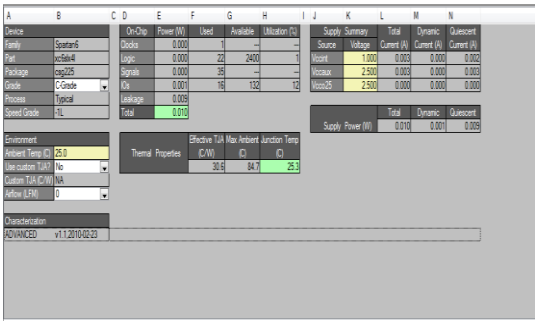
Iteration 3



WALLACE TREE



DA ALGORITHM



VII. CONCLUSION

DA algorithm which is implemented consumes low power of 0.10 watts when compared with recent implementation like Wallace tree which consumes 0.30 watts power .An achievement of 0.20 watts power has been implanted using distributed algorithm. The proposed method has been implemented for 3 bit multiplier and results obtained without any computation error.In general, a multiplier uses Booth's algorithm and array of full adders (FAs), or Wallace tree instead of the array of FA's., i.e., this multiplier mainly consists of the three parts: Booth encoder, a tree to compress the partial products such as Wallace tree, and final adder. Because Wallace tree is to add the partial products from encoder as parallel as possible, its operation time is proportional to, where is the number of inputs. It uses the fact that counting the number of 1's among the inputs reduces the number of outputs into. In real implementation, many counters are used to reduce the number of outputs in each pipeline step.

REFERANCE

[1] Fast Multiplication Algorithms and Implementation

[2] Implementation of High Speed FIR Filter using Serial and Parallel Distributed Arithmetic Algorithm

[3] FPGA Implementation of Digital FIR Filter

[4] L. Zhao, W. H. Bi, F. Liu, "Design of digital FIR band pass filter using distributed algorithm based on FPGA," Electronic Measurement Technology,2007,vol.30

[5] P. Girard, O. Héron, S.Pravossoudovitch, and M.Renovell, "Delay Fault Testing of Look-Up Tables in SRAM-Based FPGAs," Journal of ElectronicTesting, 2005, vol. 21

[6] H. Chen, C. H. Xiong, S. N. Zhong,"FPGA-based efficient programmable polyphase FIR filter," Journal of Beijing lnsititute of Technology,2005, vol. 14.

[7] Y. T. Xu, C. G. Wang, J. L. Wang,"Hardware Implementation of FIR Filter Based on DA Algorithm,"Journal of PLA University of Science and Technology, 2003, vol. 4

[8] D. Wu, Y. H. Wang, H. Z. Lu,"Distributed Arithmetic and its Implementation in FPGA,"Journal of National University of Defense Technology, 2000, vol. 22

[9] L. Wei, R. J. Yang, X. T. Cui,"Design of FIR filter based on distributed arithmetic and its FPGA implementation, "Chinese Journal of Scientific Instrument, 2008, vol. 29

[10] W. Zhu, G. M. Zhang, Z. M. Zhang,"Design of FIR Filter Based on Distributed Algorithm with Parallel Structure," Journal of Electronic Measurement andInstrument, 2007, vol. 21

[11] W. Wang, M. N. S. Swamy, M. O.Ahmad, "Novel Design andFPGA Implemention of DAFIR Filters," Journal ofSystems and Computers, 2004,vol. 13

[12] M. Nagamatsu et al., "A 15ns 32 x 32-bit CMOS Multiplier an Improved Parallel Structure,"Proc. CICC, pp.10.3.11989.