

EMPIRICAL REVIEW OF LOW POWER COLUMN BY PASS MULTIPLIER

E.r Neha Gupta

M.tech (ECE), SRMIET

Dr.B.K Sharma

Director Principal, SRMIET

ABSTRACT: Multiplication is an essential arithmetic operation for common DSP applications, such as filtering and fast Fourier transform (FFT). To achieve high execution speed, parallel array multipliers are widely used. These multipliers tend to consume most of the power in DSP computations, and thus delay efficient multipliers are very important for the design of DSP systems.

To achieve high execution speed, parallel array multipliers are widely used. The field Programmable Gate Arrays (FPGAs) is currently the dominant technology that could be reconfigured at the same time. A new methodology for low power Column bypass multiplier is proposed that inserts more number of zeros in the multiplicand thereby switching activities as well as delay consumption will be reduced. Simplified adder structure will be an added feature for this design

Keywords: Column bypassing multiplier, Modified booth algorithm, Spartan-3AN

INTRODUCTION

In scaling of C-MOS circuit the most important parameter is power dissipation. Thus power reduction is the key design goal for communication design and computing system power consist of static power and dynamic power. Static power dissipation is due to leakage current and dynamic power dissipation due to charging and discharging. Many low power designs have been found. Power reduction can be improved using structure modification.

In this paper a modified structure with reduced switching activity is presented through optimization of design.

In this session , Array multiplier based on Braun multiplier is defined. Proposed modified multiplier design is presented in next section

Experimental results with various multiplications are discussed in section III.

I PRELIMINARIES

Parallel multiplier: Consider the multiplication of two unsigned n-bit numbers, where $A = a_{n-1}, a_{n-2}, \dots, a_0$ is the multiplicand and $B = b_{n-1}, b_{n-2}, \dots$

..., b_0 is the multiplier. The product $C = c_{2n-1}, c_{2n-2}, \dots, c_0$, can be written as follows:

$$C = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (a_i * b_j) 2^{i+j}$$

The additions are shown in the corresponding row in Figure 1. On the other hand, the Baugh-Wooley multiplier uses the same array structure to handle 2's complement multiplication, with some of the partial products replaced by their complements. The multiplier array consists of (n-1) rows of CSA (carry save adder), in which each row contains (n-1) FA (full adder) cells. Each FA in the CSA array has two outputs: the sum bit goes down while the carry bit goes to the lower-left FA. For an FA in the first row, there are only two valid inputs, and the third input bit is set to 0. Therefore, it can be replaced by a two-input half-adder. The last row is a ripple adder for carry propagation. In this research, we propose a low-power design for this multiplier.

Low-power multipliers with row-bypassing: A low-power multiplier design may disable the operations in some rows to save power. If bit b_j is 0, all partial products $a_i b_j$ are zero. An array implementation, known as the Braun multiplier, is shown in Figure. 2. Therefore; the additions in the corresponding row in Fig. 1 can be bypassed. The row by passing multiplier is shown in figure. 3. Each cell in the CSA (carry save

adder) array is augmented with three tri-state gates and two multiplexers.

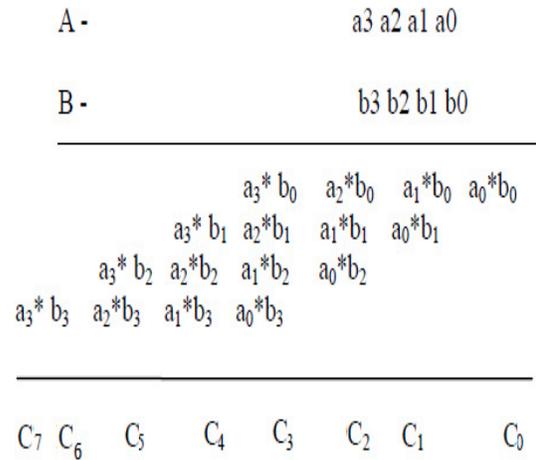


Figure 1: 4x4 multiplication

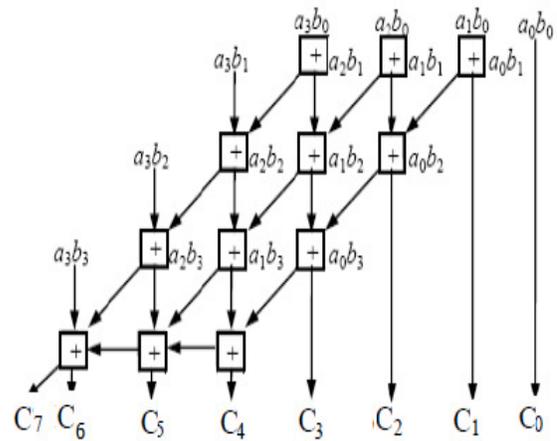


Figure 2: 4x4 Braun multiplier ref [4]

For example, let b_2 be 0 in Figure 3. In this case, the carry save adder in the second row (enclosed in the circle) can be bypassed, and the outputs from the first row are fed directly to the third

row carry save adder . However, since the rightmost FA in the second row is disabled, it does not execute the addition and thus the output is not correct. To remedy this problem, an extra circuit must be added, and these elements locate in the triangle.

LITERATURE SURVEY

[1] Floating point numbers are one possible way of representing real numbers; the IEEE standard presents two different floating point formats, Binary interchange format and Decimal interchange format. Multiplying floating point numbers is a critical requirement for DSP applications involving large dynamic range. It focuses only on single precision normalized binary interchange format. it consists of a one bit sign (S), an eight bit exponent (E), and a twenty three bit fraction (M or Mantissa). An extra bit is added to the fraction to form what is called the significand. If the exponent is greater than 0 and smaller than 255, and there is 1 in the MSB of the significant then the number is said to be a normalized number; in this case the real number is represented.

[4] The switching activity of the component used in the design depends on the input bit coefficient. Instead of bypassing rows of full adders, a multiplier design in which columns of adders are bypassed. In this approach, the

operations in a column can be disabled if the corresponding bit in the multiplicand is 0. There are two advantages of this approach. First, it eliminates the extra correcting circuit. Second, the modified FA is simpler than that used in the row-bypassing multiplier.

[3] Spartan-3 FPGAs: Xilinx family includes Spartan-3 FPGA (Xilinx, 2009) as their fifth generation. Spartan3 is purposely designed to meet the requirements of high volume, low unit cost electronic systems. The family comprises of eight member offering densities ranging from 50,000 to five million system gates. The Spartan-3 family includes L, E, A and -3A DSP, Spartan-3AN and the extended Spartan-3A FPGAs. Spartan-3AN combines all the features of Spartan-3A FPGA family plus leading technology in-system flash memory for configuration and nonvolatile data storage.

[6] Most common structure of the FFT radix-2 DIT with four multipliers, three adders and three subtractions for the calculation of the real and imaginary parts based on the structure registers are used to create the pipelines. One and two stages of pipelining. For one pipeline stage, the register barrier was placed only after the multipliers. This was done because a large amount of glitching activity produced by the multipliers can be significantly reduced. Besides, as the pipeline breaks the critical path

of the circuits, it allows that the synthesis tool can allocate slower cells and therefore with lower power consumption along the butterfly structures. there are sixteen forms to compute a complex multiplication using three real multipliers.

CONCLUSION

In this paper we have concluded reduced delay using modified booth algorithm with implemented on Spartan 3-AN family. Optimization has been achieved using VERILOG instead of VHDL. This technique achieves higher delay reduction with lower hardware overhead.

REFERENCES

- [1] Mohamed Al-Ashrafy, Ashraf Salem, Wagdy Anis "An Efficient Implementation of Floating Point Multiplier", IEEE, 978-1-4577-0069-9/11/ 2011
- [2] Stefan Langemeyer, Peter Pirsch, Holger Blume, "FPGA Architecture for real time processing of variable length FFTs", IEEE, 978-1-4577-0539, 7/11/2011
- [3] Muhammad H. Rais and Mohammed H. Al Mijalli, "Braun's Multipliers: Spartan-3AN based Design and Implementation", Journal of Computer Science, ISSN 1549-3636© 2011 Science Publications. 1629-1632, 7 /11/2011
- [4] Tushar V. More and Dr. R.V.Kshirsagar, "Design of Low Power Column Bypass Multiplier using FPGA", IEEE, 978-1-4244-8679, 3/11/2011
- [5] Dr. Ravi Shankar Mishra, Prof. Puran Gour, Braj Bihari Soni, Design and Implements of Booth and Robertson's multipliers algorithm on FPGA, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 1, Issue 3, pp.905-910
- [6] C.-Y. Pai, W.E. Lynch and A.J. Al-Khalili "Low-power data-dependent 8 _ 8 DCT/ IDCT for video compression", IEEE, IEEE Proceedings online no. 20030564 doi: 10.1049/ip-vis:20030564 IEE Proc.-Vis. Image Signal Process., Vol. 150, No. 4, August 2003
- [7] Li Shang, Alireza S Kaviani, Kusuma Bathala, "Dynamic Power Consumption in Virtex™-II FPGA Family" ACM 1-58113-452-5, February 24-26, 2002