© 2017 IJSRST | Volume 3 | Issue 6 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

High Performance and Area Efficient DSP Architecture using Dadda Multiplier

V.Kiran Kumar Reddy[1], K. Sravan Kumar[2]

[1] PG scholar, JNTUA College of Engineering, Ananthapuramu, Andhra Pradesh, India
[2] Lecturer, JNTUA College of Engineering, Ananthapuramu, Andhra Pradesh, India

ABSTRACT

Modern embedded structures required to implement high-end software domain in Digital Signal Processing (DSP). Now-a-days multipliers and DSP functions plays significant role to do fast computations. The efficient DSP functions are implemented by adopting the functions of flexible data path architecture with Modified Booth (MB) multiplier using Carry save arithmetic adder from previous works and now it was extended with Dadda multiplier using CS (Carry Save). In this architecture the structure comprising of flexible computational unit (FCU) in Dadda multiplier with CS to perform fast DSP functions. The flexible DSP functions used to increase the performance by reduce the area, power and delay with carry save arithmetic level abstraction entity were proposed. The FCU architecture design of Dadda multiplier with CS results considerable compressions in power, area and combinational path delay as 13.8 %, 15.6 %, and 15.5 % respectively compared with the MB multiplier with CS.
Keywords: Carry Save, Modified Booth Multiplier, Dadda Multiplier.

I. INTRODUCTION

Nowadays embedded systems aims at high-end applications and its domains require fast operating Digital Signal Processing (DSP) operations. To do fast operations particularized hardware accelerators merged which reduces the power and area [1]. Many research papers were proposed that different accelerators [2] to increase the flexibility of IC's without effecting the function of the accelerator. In design multiplier was fundamental component and it run complex operations in DSP and in general processors. Multiplication procedure demands generation of PP's (partial products) plus its accumulation. The fast of multiplication should be increased with diminishing the PP's number and/or by partial products accumulation process. In this the High-performance architecture model of DSP for synthesis by mixing the optimization technique. We introduce the high-performance DSP architecture which uses CS (Carry-Save) templates in optimized manner to perform the operations. In proposed architecture FCUs (Flexible Computation Unit) perform the executions of different set of templates constitute in DSP architectures.

The architecture templates include complicated chained operations extracted from the library [1]. Design of architecture will severely impact on the performance parameters as power, area and delay. In portable devices area and power were main considerations so that to possible level values would be reduced.

Among the all methods of implementation of high speed multipliers, there was simple approaches namely DADDA multiplier Tree compressors. In this the effective function execution of a speed multiplier using this approaches. The Dadda employs CS Adder to cumulate the partial products. This brings down the power, and the chip area. To enhance further, operation speed, CLA (carry-look-ahead adder) is exploited for the final addition [6].

II. CARRY-SAVE (CS) ARITHMETIC :MOTIVATIONAL OBSERVATIONS

Carry Save Arithmetic was widely used in design of fast arithmetic blocks to make use of eliminating large number carry propagations. By fast assigning adder and driving of its output from the input the multiplier, increases importantly both area and delay of the design. CS (Carry-Save) arithmetic optimization technique [4], [5] uses multiple input additions onto Carry-Save compressors. All filtering related applications of DSP dominated with multiplications. Every time that multiplication is required carry-save to binary conversion technique is into action by using the distributive theorem.

This is a simple circuit in which multi-input adder on compressor tree abided by carry propagate adder. The CS adder propagates the all carries to LSB of partial products (PP's) and then generates LSB in advance for the purpose of decrease in input bits in final adder. By compounding multiplication process accumulation with a composite combination of carry save (CS) adder, total functioning was improved.

In this we manage the limitation by using Carry-Save to MB (Modified Booth) recoding every time the multiplication is done within the data-path. Also Carry-Save to Dadda recoding is also done to compare the performance parameters. Therefore the calculations throughout the multiplication are done by carry-save arithmetic and operations are performed in the data-path and not including any other adder for carry-save to binary format conversion, then total performance is improved.

III. FLEXIBLE COMPUTATIONAL UNIT (FCU)

In flexible accelerator structure was shown in Fig.1. Every FCU directly operates on Carry-Save operands and gives data in same form to reprocess intermediate results.

Every FCU works on 16-bit data operands to so extreme a degree it is sufficient bit-length for almost all DSP datapaths [7] but FCU architecture concept is simply used for small or large bit-lengths. Design of FCU is obligatory by the design of designer.

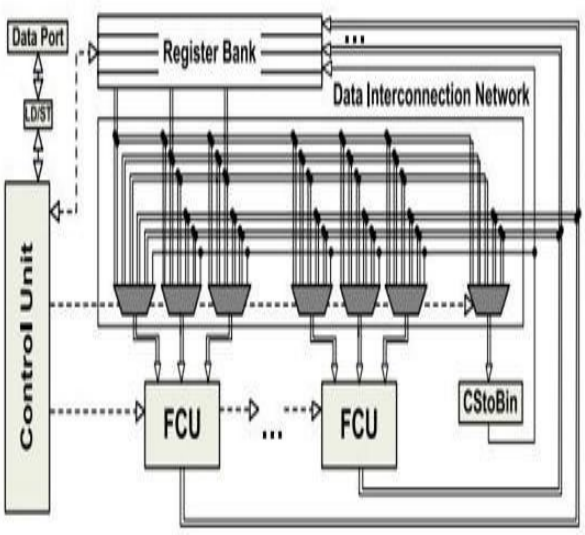

Figure 1. Abstraction form of Flexible datapath.

CS to Binary module is RCA (Ripple Carry Adder) which converts carry-save form to 2's complement form. The register bank comprise scratch registers these are helpful to keep intermediate results also share operands among FCU's. Control unit aims at total architecture in all clock cycles.

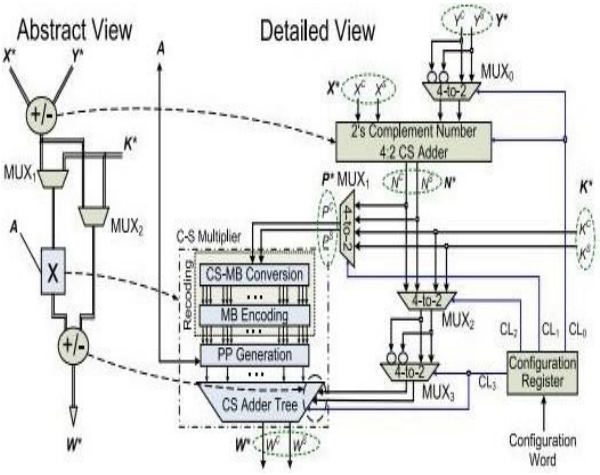

Figure 2. FCU

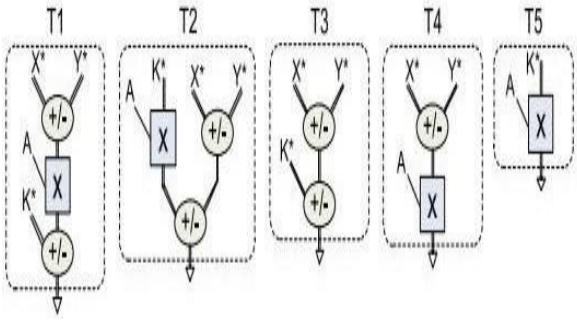

Figure 3. Template Library of FCU

A. Proposed Structure of FCU

The FCU design is able use for high performance operation chaining to do operation templates [2]. FCU is configured with any one from operation templates as T1-T5 as depicted in Fig.3. The proposed intra-template of FCU operation chaining along different additions which performed before or after multiplication and also carryout partial template operations such as following complicated operations:

$$W^* = A \times (X^* + Y^*) + K^*$$
$$W^* = A \times K^* + (X^* + Y^*)$$

The following equation holds for CS data: $X^* = \{X^C, X^S\} = X^C + X^S$. Operand A is 2's complement number. The substitute for execution path in FCU are mentioned after setting of control signal to the multiplexers MUX_2 and MUX_1 (Fig. 2). The MUX_0 output is $=$ when control-bit $CL_0 = 0$ (i.e., X^*+Y^* is executed) or Y^* when $X^* - Y^*$ is necessary and $CL_0 = 1$.

The 2's complement produced by the 4:2 Carry Save is $N^* = X^* - Y^*$ if input carry is equals 1. The MUX_1 evaluate if $K^*(2)$ or N^* (1) is multiplied by A. The MUX_2 determines N^* (2) or $K^*(1)$ is summed with multiplication product. The MUX3 considers the output of MUX2 and its one's complement and outputs the early one when an addition accompanying the multiplication product is needed (i.e., control bit CL3 is Zero) or the posterior one if subtraction is done (i.e., control bit CL3 is One). The 1-bit adopt for the subtraction is summed in CS adder tree.

B. Multiplier operation in FCU Architecture

The multiplier composed of Carry-Save to MB (Modified Booth) module, which take up the new proposed techniques [9] to recode 17-bit P^* corresponds to MB digits accompanying minimum carry propagation. But all inputs of FCU has 16-bits and there was no overflow provided will produce the 34-bit output W^* are placed in the suitable FCU whenever required.

In this multiplier is replaced with Carry-Save to Dadda module, which is helpful to reduces the area (LUT's) and power consumption when compared with the Carry-Save to MB module to analysis the performance.

IV. Multipliers in design

A. Modified Booth Multiplier:

To perform speed multiplication process algorithms exploiting parallel counters, as the MB algorithm [9] was proposed, and there are some multipliers available based on algorithm implementations for practical purpose. The Modified Booth's algorithm, presents an efficient solution in multiplication that helps the demands of speed multipliers, and also hardware design and area complexity to be effective. Modified Booth is predominant form in multiplication. It is proposed by Booth-Mac Sorley. It is an encoding technique with radix-4 where sign-bit is redundant. The MB algorithm has modified array to extend the sign for increase of bit density of operands. Because of this advantage it reduces the partial-products significantly compared with radix-2, reduce adders, it provides speed advantage.

B. Dadda Multiplier:

In this technique partial products are reduced at each every part by exploiting (3,2) and (2,2) compressors it means that grouping of 3-bits by (3,2) and 2-bit by (2,2) in every column it results partial sum and carry [10]. Dadda scheme fundamentally minimises the adder stages to perform the summation of PP's (partial products).

In Dadda multiplier of 16-bit, six recoding stages are used for minimization of PP's such as reduced to 13,9,6,4,3,2 from stage one to stage six respectively. It is achieved with use of half adders and full adders to minimize number of rows with number of bits in every stage so that it will reduce the PP's drastically and final addition will give the product result.

V. SIMULATION RESULTS

The FCU architecture implementation was written in Verilog code with Xilinx to perform the all types of operations occurred in DSP with the consideration of combinational template. The performance analysis was studied with parameters such as power, area, and delay for different FPGA families by implementing it with MB technique and Dadda technique.

So that the comparison was made by considering the data as critical delay from timing port, and area from device utilization summary and power from XPower analyzer.

The following table gives the analysis of performance parameters of both methods in FCU design with Xilinx Virtex-6 FPGA family.

S.NO	Multiplier type in FCU design	PERFORMANCE PARAMETERS		
		POWER (in W)	Area (in terms of LUT'S)	Delay (in ns)
1.	Modified Booth Multiplier (Existing method)	3.970	1231 out of 150720	8.986
2.	Dadda Multiplier (Proposed method)	3.422	1038 out of 150720	7.587

Table I. Parameter comparison of FCU design in both methods

The following graphs area, power, and delay show that clear analysis of the performance parameters.

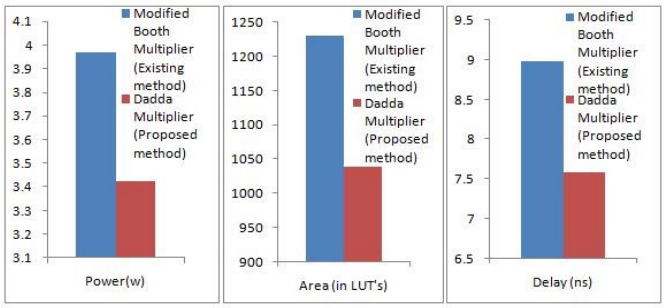

Figure 4. Power, Area and Delay comparison of FCU design using Modified Booth and Dadda multiplier

As shown in graphs the area utilization is lessened by 15.6%, power dissipation is lessened by 13.8% and delay is lessened by 15.5%.

VI.CONCLUSION

In this paper we introduced FCU architecture that utilizes the CS to do complex multiplication and addition operations, with capacity to function the both 2's complement and Carry-Save arrange data operands. Experimental analyses have shown that that the proposed solution forms an efficient design with parameters of power, area and delay. The parameters will further improved with adopting CLA in place of CS adder.

VII. REFERENCES

[1] Kostas Tsoumanis, Sotirios Xydis, GeorgiosZervakis, and KiamalPekmestzi "Flexible DSP Accelerator Architecture Exploiting Carry-Save Arithmetic" IEEE Trans. Very Large Scale Integr. (VLSI) Syst. Vol. 24, No. 1, January 2016.

[2] S. Xydis, G. Economakos, D. Soudris, and K. Pekmestzi, "High performance and area efficient flexible DSP datapath synthesis," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 3, pp. 429–442, Mar. 2011.

[3] R. Kastner, A. Kaplan, S. O. Memik, and E. Bozorgzadeh, "Instruction generation for hybrid reconfigurable systems," ACM Trans. Design Autom. Electron. Syst., vol. 7, no. 4, pp. 605–627, Oct. 2002.

[4] A. K. Verma, P. Brisk, and P. Ienne, "Data-flow transformations to maximize the use of carry-save representation in arithmetic circuits," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 27, no. 10, pp. 1761–1774, Oct. 2008.

[5] T. Kim and J. Um, "A practical approach to the synthesis of arithmetic circuits using carry-save-adders," IEEE Trans. Comput.Aided Design Integr. Circuits Syst., vol. 19, no. 5, pp. 615–624, May 2000.

[6] Naveen K Gahlan, Prabhat, JasbirKaur "Implementation of Wallace Tree Multiplier Using Compressor" International Journal of Computer & Technology, Vol 3 (3), 1194-1199.

[7] G. Constantinides, P. Y. K. Cheung, and W. Luk, "Synthesis and Optimization of DSP Algorithms" Norwell, MA, USA: Kluwer, 2004

[8] Y.H. Seo and D.W. Kim, "A new VLSI architecture of parallel multiplier–accumulator based on Radix-2 modified Booth algorithm," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 201–208, Feb. 2010.

[9] K. Tsoumanis, S. Xydis, C. Efstathiou, N. Moschopoulos, and K. Pekmestzi, "An optimized modified booth recoder for efficient design of the add-multiply operator," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 4, pp. 1133–1143, Apr. 2014.

[10] N.Savithaa, Dr.C.S.Manikandababu "An Efficient Design of Dadda Multiplier Using Compression Techniques" ,International Journal of Advanced Research in Science, Engineering and Technology,Vol. 4, Issue 3 , March 2016.

About Authors:

Mr. V. Kiran Kumar Reddy completed B.Tech in EEE from G.Pulla Reddy Engineering College, Kurnool in 2014. He is pursuing M Tech in JNTUA college of Engineering, Anantapuramu. His areas of interests are Power Systems and Electrical Machines.

Mr.K.Sravan Kumar completed B.E in ECE from Magna College of Engineering, Chennai in 2005, and M.Tech from SRM University, Chennai in 2007.He is currently working as a Lecturer, ECE Department, JNTUA College of Engineering, Ananthapuramu. His areas of interests are Wireless Communications, Signal Processing and Cognitive Radio.