ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

An Efficient Flexible Architecture for Error Tolerant Applications

Sheema Mol K.N [1], Rahul M Nair [2]

M.Tech Student (VLSI DESIGN), Department of Electronics and Communication Engineering, Nehru College of Engineering and Research Centre, Thrissur, Kerala, India[1]

Assistant Professor, Department of Electronics and Communication Engineering, Nehru College of Engineering and Research Centre, Thrissur, Kerala, India[2]

ABSTRACT: This paper introduces an efficient flexible architecture for error tolerant applications to implement DSP kernels. The proposed methodology is more compact than traditional arithmetic units which enable the exploitation of error tolerant adder. The flexible architecture comprises of flexible computational units which execute large number of operation templates, exploits carry save format for its operations and it is based on modified booth algorithm. It is able to handle 8 bit, 16 bit and 32 bit operations and also provides high computational density, fast operations and reusability of data. Comparing this with the existing architectures, the proposed method yields better performance in terms of power, speed, and area.

KEYWORDS: Flexible architecture, Flexible computational unit, DSP kernels, Carry save form, Modified booth algorithm, Error tolerant adder.

I. INTRODUCTION

Digital Signal Processing (DSP) is intended especially to perform mathematical calculations. DSP [13] applications usually requires more time for execution of these operations, the reconfigurable architecture is a way out for this. The main classifications of reconfigurable architectures are coarse grained [1] and fine grained architectures. In respect to their interconnections, coarse grained can again classified as row [2] and array based [3]. The row based configuration incorporated the optimization [4] concepts like horizontal parallelism [3], [6], vertical parallelism [3], or operation chaining [6], [5]. The related works chose the row based coarse grained reconfigurable architectures for this reason. In [7], a single flexible architecture differentiates from other approaches because it encompass the whole architectural optimizations which delivers high performance data path.

The flexible architecture consists of flexible computational units (FCU) realized the high performance data path. Thus the operation speed of DSP kernels can enhance. The computational units can be Arithmetic & Logic Unit (ALU), Multiplier and Shifter. The flexibility is achieved by using limited number of multiplexers and it allows the execution a large set of operation templates. The main objective of every design is to reduce the delay, area and power consumption. Carry propagation design introduces more delay as well as power consumption in arithmetic circuits. Therefore binary numbers are represented using carry save (CS) [8] form. This paper mainly proposes an accuracy adjustment flexible computational architecture which uses carry save format for its operations by exploiting error tolerant adder (ETA).

The remaining section of the paper is organized as follows: Section II, described about flexible architecture template and the depiction of existing flexible computational unit. After that about the proposed FCU using error tolerant adder was presented in Section III. Section IV and Section V respectively presents the comparison and simulation results. Potential applications were described in Section VI. Finally conclusion was made in Section VII.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

II. FLEXIBLE COMPUTATIONAL UNIT

The template of a flexible architecture [9] and Flexible Computational Unit is shown in Fig.1. FCU realizes the data paths which boost up the execution of more operation templates. The Flexible Computational Unit contains multiplexers, multiplier, and adders. Carry save adders (CSA) [10] which allow the addition of more than two operands is used over here and modified booth algorithm was used to generate partial products. All operations are performed in carry save form. Thus it increases the performance of DSP kernels.

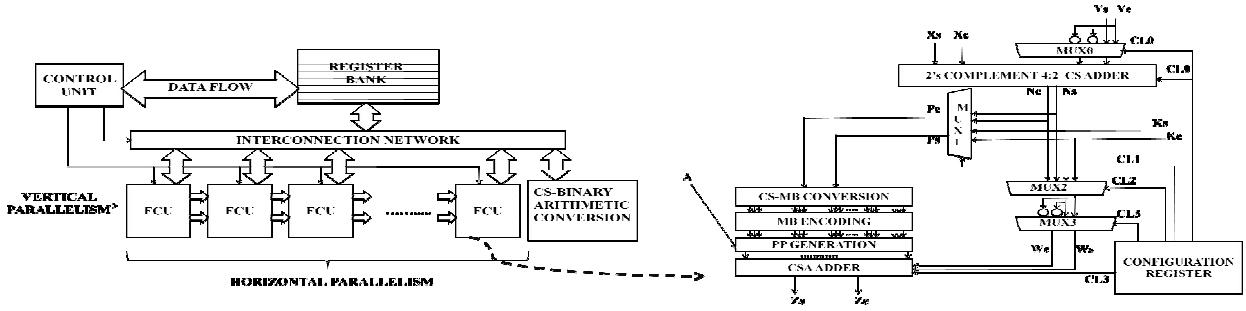

Fig.1. Flexible Architecture Template and Detailed Structure of Flexible Computational Unit

The input operands X*, Y*, K* and the output Z* are expressed in carry save form. The multiplicand P (Pc,Ps) and output of carry save adder are in carry save form is converted using S-MB (Sum to Modified Booth) recoding technique [14] and ripple carry adder (RCA) respectively. The number of FCU is mainly depends upon the application. The width of operands can be 8, 16, or 32 bit. The operation which we need to perform is determined by choosing the suitable configuration (CL3, CL2, CL1, CL0). The configuration bits acts as control signals for multiplexers MUX0, MUX1, MUX2, MUX3 respectively. Depending upon the configuration shown in Table I the multiplexers will select the inputs need to passed.

TABLE I. FCU OPERATIONS

Configuration	Operation
0000	(X+Y)*A+(X+Y)
0001	(X-Y)*A+(X-Y)
0010	K*A+(X+Y)
0011	K*A+(X-Y)
0100	(X+Y)*A+K
0101	(X-Y)*A+K
0110	K*A+K
0111	K*A+K
1000	(X+Y)*A-(X+Y)
1001	(X-Y)*A-(X-Y)
1010	K*A-(X+Y)
1011	K*A-(X-Y)
1100	(X+Y)*A-K
1101	(X-Y)*A-K
1110	K*A-K
1111	K*A-K

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

III. PROPOSED FLEXIBLE COMPUTATIONAL UNIT

The proposed model was developed using error tolerant adder for applications where we can give up some amount of accuracy. The proposed FCU replaced the CS-B (Carry Save to Binary) conversion with RCA and carry save adder in existing FCU with ETA array as shown in Fig.2. Even though there are many low power adder designs have been introduced most of the adders are not suitable for low power applications, whenever the size of operands increases. Still there are always trade-offs between speed and power. This problem can be sacrificed by using error tolerant adder [11], [12]. Thus by sacrificing some accuracy ETA array can attain improvement in power, area, and speed in FCUs.

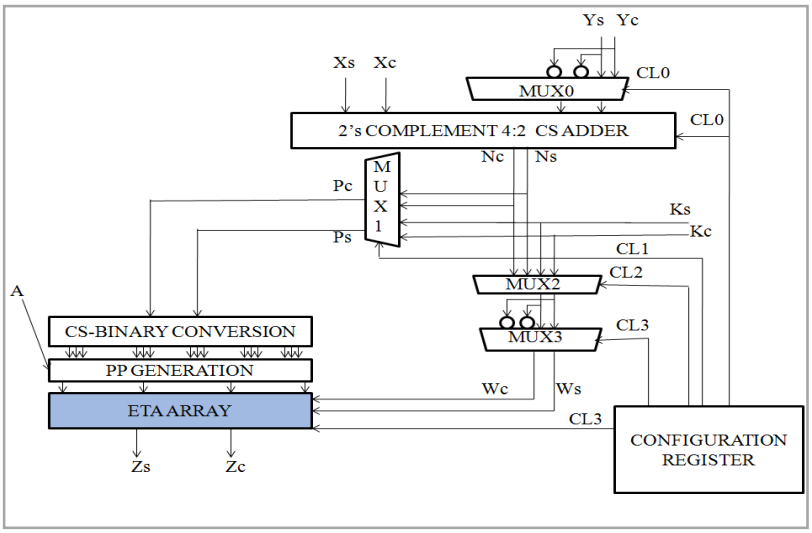

Fig.2. Proposed Flexible Computational Unit

A. Error Tolerant Adder

The main objective of every design is to enhance the speed, reduce the power consumption and area utilization. The delay of a circuit is mainly due to carry propagation and glitches in carry propagation causes dynamic power dissipation. Thus by eliminating carry propagations, speed can enhance and low power can achieve. The ETA adder eliminates the carry propagation by adopting a specific strategy for addition as shown in Fig.3. It divides the inputs into two: accurate and inaccurate part. The higher order bits and lower order bits will be in accurate and inaccurate part respectively. The main step used to facilitate the implementation of the ETA was its dividing strategy which determines the performance of the system. This is determined by verification and checks the parameters: accuracy, power, area, and speed. After verification, if the requirement does not as specified by the designer we have to change the current dividing strategy and the checking process repeated until all specifications are met. The addition proceeds from the point of separation towards the opposite direction simultaneously. Normal addition will be carried out in accurate part using ripple carry adder from right to left. A special mechanism of addition is carried out in inaccurate part to avoid the carry propagation from left to right using control block and carry free propagation block. The design of control block and carry free propagation block is depicted in Fig.4. Area utilization is effectively reduced by the design of inaccurate part.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

The addition mechanism in inaccurate part is described as follows:

- Check every bit from left to right.
- If both bits are '0' or 'different' normal addition will be carried out and checking process will jump to next bit position.
- If both bits are '1', then all sum bits from that bit position to LSB will set to '1'.

.

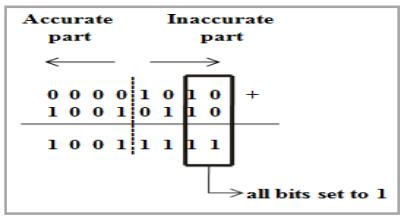

Fig.3. Addition Mechanism of ETA.

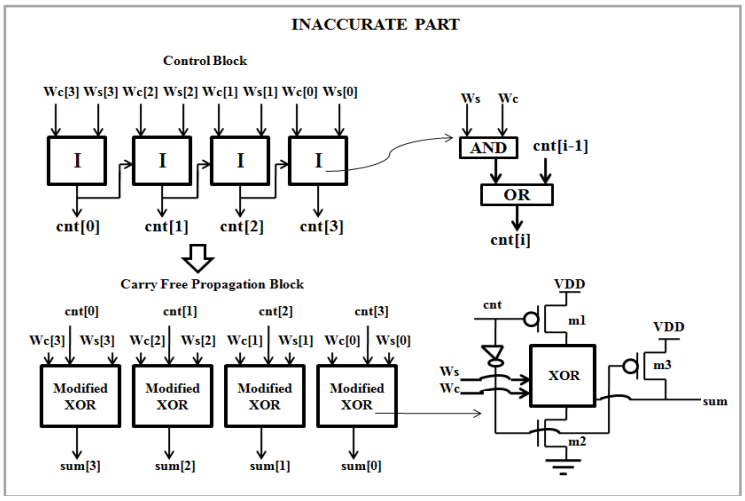

Fig.4. Block diagram of Inaccurate Part.

IV. COMPARISONS AND DISCUSSIONS

This section presents a comparison of accuracy, power consumption, area utilization, and speed performance of proposed design with existing one.

A. Accuracy

The proposed flexible computational unit achieves the minimum acceptable accuracy (MAA) [12] required for applications. The MAA is acceptable when the accuracy is greater than the minimum threshold value. Most of the designs have a minimum threshold value lies in the range 95% to 100%. The accuracy can be calculated as follows:
Accuracy = (1-(overall error) / correct result) *100…….. (1)

Overall Error = difference of correct result and obtained result ……... (2)

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Table II shown below presents the accuracy and overall error of the proposed flexible architecture for all configurations. Let us consider the inputs as follows: Xc (50), Xs (50), Yc (-106), Ys (-28), Kc (100), Ks (-122), A (255).

TABLE II.

OVERALL ERROR, ACCURACY OF PROPOSED FCU.

CONFIGURATION	CORRECT OUTPUT	OBTAINED OUTPUT	OVERALL ERROR	ACCURACY(%)
0000	-8704	-8704	0	100
0001	59904	59904	0	100
0010	-5644	-5644	0	100
0011	-5376	-5632	256	95.23
0100	-8692	-8692	0	100
0101	59648	59904	256	99.57
0110	-5632	-5632	0	100
0111	-5632	-5632	0	100
1000	-8636	-8636	0	100
1001	59436	59948	512	99.13
1010	-5576	-5576	0	100
1011	-5844	-5588	256	95.61
1100	-8648	-8648	0	100
1101	59692	59948	256	99.57
1110	-5588	-5588	0	100
1111	-5588	-5588	0	100

It is clear from TABLE II, the proposed technique suitable for all applications whose required MAA was lesser than or equal to 95.23%.

B. Power Consumption

The proposed design reduces dynamic power consumption by

- ETA which eliminates carry propagation, thus rippling does not occur.

C. Area Utilization

The area utilization of introduced design will greatly decreases by

- Usage of ETA which reduces the requirement of AND, OR, and NOT gates to generate the sum bits.

D. Speed

The proposed FCU has a significant reduction in delay by

- Eliminating the carry propagation in ETA.
- Operands are represented in carry save form.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

TABLE III.

AREA, DELAY, POWER, AREA DELAY PRODUCT, POWER DELAY PRODUCT OF EXISTING AND PROPOSED FCU.

Types	Area (no of slices)	Delay (ns)	Power (mW)	Area Delay Product (ADP)	Power Delay Product (PDP)
Existing FCU[9]	207	46.898	287	9707.886	13459.726
Proposed FCU	162	41.999	274	6803.838	11507.726

The efficiency of proposed design can notice from the comparison of power, delay, and area shown in Table III. Table III also shows the design metrics: area-delay product and the power delay product providing an obvious outlook of the advantages of the proposed approach and can conclude that the proposed FCU is efficient.

V. SIMULATION RESULTS

The proposed methodology is designed using XILINX 9.1i using VERILOG HDL code and simulated using MODELSIM 6.3f. The input frequency was set to 100 MHz. Here we got an accuracy of about 95% to 100%. The simulation results of existing and proposed designs are shown in Fig.6 and Fig.7 respectively.

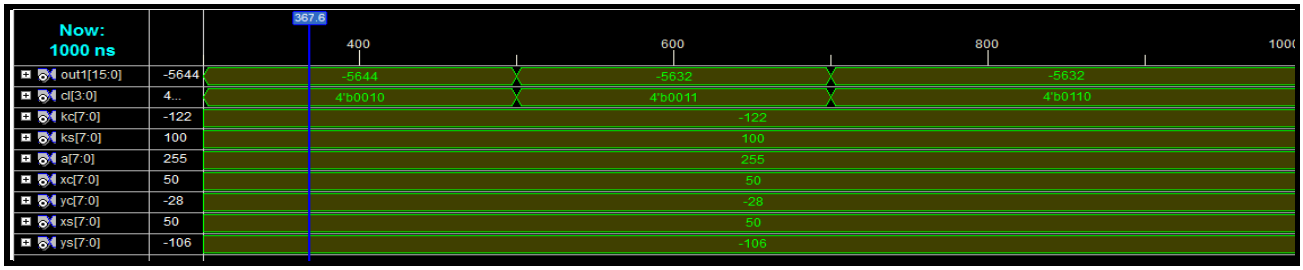

Fig.6. Output waveform of Existing FCU

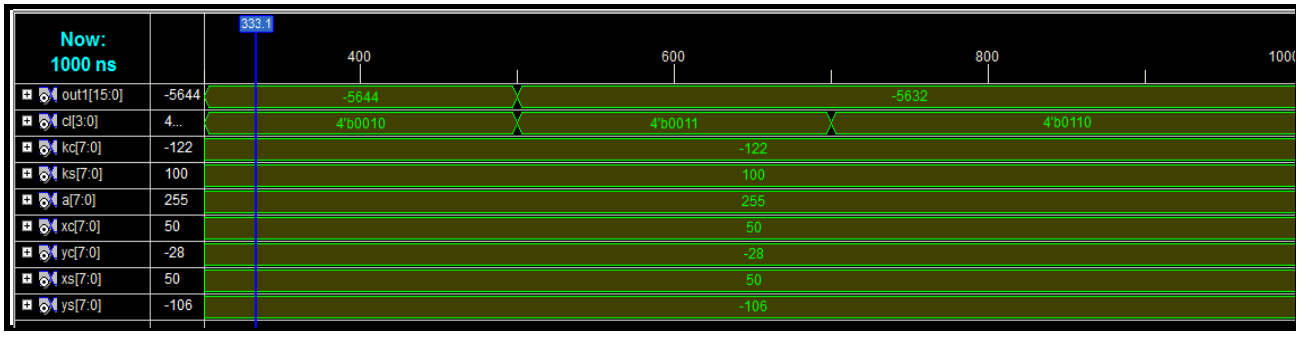

Fig.7. Output waveform of Proposed FCU

VI. APPLICATIONS

The proposed design can used in many DSP applications such as image processing and speech processing which can tolerate some amount of error. These systems contains DSP kernels such as finite impulse response (FIR) filter, Matrix

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Multiplication, Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT) which involve a large number of additions and multiplications. So this is an excellent platform for embedding our planned design.

TABLE IV.

AREA DELAY PRODUCT, POWER DELAY PRODUCT FOR DSP KERNELS MAPPED ONTO THE EXISTING FCU AND PROPOSED FCU.

DSP Kernels	Area Delay Product (ADP)		Power Delay Product (PDP)	
	Existing FCU[9]	Proposed FCU	Existing FCU[9]	Proposed FCU
FIR 8	35468.292	13568.04	36155.662	18429.921
Sym FIR 8	8495.97	5523.816	7444.088	6255.888
Sym FIR 16	18640.344	10387.816	13277.34	9021.944
Matrix Multiplication	60080.248	25668.426	71588.425	45049.998

Table IV considered the two design metrics: Power Delay Product (PDP) and Area Delay Product (ADP) of DSP kernels. Here a set of DSP kernels was considered to demonstrate the efficiency of the proposed FCU. The DSP kernels are as follows1) 8 tap FIR filter, 2) symmetric 8 tap FIR filter (Sym FIR 8), 2) symmetric 16 tap FIR filter (Sym FIR 16), and 3) matrix multiplication of [4x4]*[4x1].The proposed FCU outperforms the existing one for all these DSP kernels.

VII. CONCLUSION

In this paper we have introduced flexible architecture for error tolerant applications which includes flexible computational units operating on carry save operands. The proposed method tradeoffs some amount of accuracy with significant power saving, less area utilization and high speed. Studied and simulated about the existing FCU and proposed error tolerant FCU. The effectiveness of proposed methodology has also been verified using DSP kernels. On analysing the results we can conclude that the proposed method performs more efficiently compared to existing one.

REFERENCES

[1] G. Theodoridis, D. Soudris, and S. Vassiliadis, "A survey of coarse-grain reconfigurable architectures and cad tools," in Fineand Coarse Grain Reconfigurable Computing, S. Vassiliadis and D. Soudris, Eds. New York: Springer, 2007, pp. 3–149.
[2] H. Singh, M. Lee, G. Lu, F. Kurdahi, N. Bagherzadeh, and E. Filho, "Morphosys: An integrated reconfigurable system for data-parallel and computation-intensive applications," IEEE Trans. Comput., vol. 49, no.5, p. 465, May 2000.
[3] C. Ebeling, D. Green, and P. Franklin, "RaPiD: Reconfigurable pipelined datapath," in Proc. FPL, 1996, pp. 126–135.
[4] G. De Micheli, Synthesis and Optimization of Digital Circuits. New York: McGraw-Hill, 1994.
[5] York: McGraw-Hill, 1994.P. Marwedel, B. Landwehr, and R. Domer, "Built-in chaining: Introducing complex components into architectural synthesis," in Proc.ASP-DAC, 1997, pp. 599–605.
[6] M. Galanis, G. Theodoridis, S. Tragoudas, and C. Goutis, "A high performance data-path for synthesizing DSP kernels," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 25, no. 6, pp.1154–1163, Jun. 2006.
[7] Sotirios Xydis, George Economakos, Dimitrios Soudris and Kiamal Pekmestzi, "High Performance and Area Efficient Flexible DSP Datapath Synthesis" IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 19, No. 3, March 2011
[8] K. Hwang, "Computer Arithmetic," in John Wiley and Sons, 1979.
[9] Kostas Tsoumanis, Sotirios Xydis, Georgios Zervakis, and Kiamal Pekmestzi, "Flexible DSP Accelerator Architecture Exploiting Carry-Save Arithmetic," IEEE Transactions On Very Large Scale Integration (VLSI) Systems, 2015
[10] J. Um, T. Kim. An Optimal Allocation of Carry-Save-Adders in Arithmetic Circuits. IEEE Trans. on Computers,50(3):215–233, 2001.
[11] Tadgiri Aruna, R.Bhadraiah, "An Implementation on 32-Bit High Speed Truncation- Error -Tolerant Adder with Low power Consumption" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-6, August 2013
[12] H. Chung and A. Ortega, "Analysis and testing for error tolerant motion estimation," in Proc. Defect and Fault Tolerance in VLSI Syst. Symp., 2005, pp. 514–522.
[13] S. W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing," California Technical Publishing, 1997, pp. 503-534.
[14] 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.