Universal Journal of Electrical and Electronic Engineering 7(6): 307-314, 2020 http://www.hrpub.org
DOI: 10.13189/ujeee.2020.070602

A CMOS Based Self Repair Fault Tolerant Adder for Low Power Biomedical Systems

Sureshkumar Pittala*, Sasikala Duraisamy

Department of Electronics and Communication Engineering, RVR & JC College of Engineering (Autonomous), Guntur, Andhra Pradesh, India

Received January 25, 2020; Revised August 4, 2020; Accepted August 28, 2020

Cite This Paper in the following Citation Styles

(a): *[1] Sureshkumar Pittala, Sasikala Duraisamy , "A CMOS Based Self Repair Fault Tolerant Adder for Low Power Biomedical Systems," Universal Journal of Electrical and Electronic Engineering, Vol. 7, No. 6, pp. 307 - 314, 2020. DOI: 10.13189/ujeee.2020.070602.*

(b): *Sureshkumar Pittala, Sasikala Duraisamy (2020). A CMOS Based Self Repair Fault Tolerant Adder for Low Power Biomedical Systems. Universal Journal of Electrical and Electronic Engineering, 7(6), 307 - 314. DOI: 10.13189/ujeee.2020.070602.*

Abstract In recent years, processor cores for biomedical signal processing play a vital role for portable devices. The processing unit in the device performs acquisition, feature detection and decision making. The computing algorithm comprises of adders, multipliers, registers, and buffers. Adders are the basic building blocks of multipliers, filters and feature extraction units. Discrete Wavelet transform is one of the important methods used to filter and extract information from biomedical signals which can be one dimensional or multidimensional. Fault avoidance and tolerance have been approached in past for achieving efficient and reliable system. These prevention and tolerance through redundancy can avoid failures. Different redundancies are adopted through information processing, hardware, software and time redundancy. Fault tolerant circuits can improve the efficiency of the biomedical system like DWT cores. The main objective is to design a reconfigurable full adder with self-checking and self-repairing faults for the DWT architecture. This paper presents a proposed FPGA based fault detection and repairing circuit including the fault location. The existing method uses an on-chip based adder which is complex and is less efficient and not suitable for reprogramming. In the proposed design, DWT architecture was designed using the reconfigurable multiplier, fault tolerant adder and Flip Flop. The implementation was carried out in Quartus Tool for different FPGA kits designed with 90nm and 65nm CMOS technology. Parameters like LUT, power dissipation and delay are investigated. The proposed approach is suitable for portable devices.

Keywords CMOS, Adder, Fault Tolerant, RCA, CSA, Low Power, Processing Element

1. Introduction

Biomedical systems in recent years, uses integrated circuit (IC) designed using LSI, VLSI and ULSI technology. Today a greater number of transistors are combined to form blocks to manage the space in the chip. These blocks form the basic building units of Application Specific Integrated Circuits (ASICs) for biomedical application. In past microprocessors and microcontrollers were used in the design of medical devices. They were the part of data acquisition unit to acquire biomedical signals, filter it and process the same. FPGA based implementation made the biomedical devices compact and reliable. Compared to processors FPGA are costlier but more efficient. In ASICs using gate level implementation, the area of the circuit can be reduced. The power consumption can be reduced using CMOS devices instead of BJT and FET devices. Gates forms the basic element of any circuit. Different types of arithmetic and logic operations are implemented using gates. The adder forms the basic building blocks of several operations like subtraction, division, multiplication, signal processing units and compression units. The multiplier circuit determines the critical delay occurring in the system. Since adders are

building blocks of multipliers, faster adders can reduce the delay. Several adders like carry skip adder, ripple carry adder, carry save adder etc. are used for multiplier implementation. The blocks of digital signal processing algorithms need adders has functional blocks. All real time applications handling signals with varying frequencies use blocks like Fast Fourier Transform, decimation etc. which needs adders. Results can be obtained in faster rate if the complex adder block is modified structural wise or carry propagation (Dhanasekar et al, 2019). Since in Signal Processing and microprocessor applications, full adder plays a vital role to perform various operations faults cannot be tolerated. The adder faults are categorized into permanent, transient or intermittent faults. Mendoza-Hernandez et al (2006) described the step for identification of fault and the replacement of the full adder is the faults are more. Since the fault will create another fault, the adder is to be replaced. But if it is an on-chip adder the whole IC has to be replaced which is not easy. Fault tolerance techniques can increase the cost but avoids complete system failure. The computing system is highly susceptible to transient errors and permanent faults. Gnana et al (2017) introduced self checking and self-repairing fault tolerant full adder to categories single faults and the double faults. This technique can recognize faults but occupies more area. The problem in fault tolerant may cause complexity. This failure is rectified by the self checking technique. To perform fault tolerant mechanism İlke Ercan et al (2017) uses hamming code designed using CMOS reversible logic. The reversible CMOS circuits are used to improve the performance and efficiency. Sonal gupta et al (2018) designed fault tolerant full adder with faster checking and repairing capability for single fault and double faults. Kaur et al (2012) designed a reversible logic based adder which preserves the input matches. Yasamin and Mohammad investigated adder circuit which is tolerant to all faults. Prachi Palsodkar et al (2018) described a multiple error detection and correction in adders. Several error are considered but the circuit is difficult to design due to the complexity and more number of elements. Ankur Sarker et al (2014) presented fault detection capability by combining the adder and subtractor function. The design was based on reversible gates. Sajib Kumar Mitra and Ahsan Raja Chowdhury (2015) reported a Reversible Full Adder (RFT-FA) of Carry Skip and Look-Ahead logic type. Sensor adder can provide better performance and power concumption but the tradeoff is speed (Ning-Chi Huang et al, 2019). The arithmetic computation is performed with more number of components which increases the area and power. Few methods use non redundant cells and are based on XOR and adder (Gurmohan Singh et al, 2018).

2. Background Methodology

Fault avoidance and tolerance have been approached in past for achieving efficient and reliable system. These prevention and tolerance through redundancy can avoid failures. Different redundancy is adopted through information processing, hardware, software and time redundancy. The failures normally happen through the unidentified/identifiable faults. In complex megasystems the avoidance problem is maximum and it is difficult to undergo. Hence the module reliability of the system can be improved by fault-tolerance, validation and error correction. Validation measures the system performance during construction process to reduce errors. Error correction is one of the taxonomy of the approaches to reduce failures and latent errors. These can be avoided or tolerated through redundancy. This system fails if the latent error prevents maintenance. Effective error processing tries to correct the error after it becomes effective and by masking. Hardware redundancy approach the two effective redundancies is used by Sonal gupta et al (2018). In parallel computation process triple modular redundancy and double modular redundancy are used for fault identification. The tradeoff is additional component requirements which increases the area. The software redundancy is the most important challenge facing the problem in the area of fault-tolerance. The software errors are different from the hardware errors because in hardware error it will not recur after they discovered and corrected. Hence the software redundancy will correct the programming error to create new error. This redundancy has more complex and immature art than hardware design. The time redundancy is performed for checking the results several times at the same module instead of doing in parallel. This redundancy does not require an extra hardware for performing hardware information's. Figure 1 shows the self checking adder designed by Vasudevan et al (2007). It contains elements like multiplexer, gates, checker and adders. The faults are detected online. Combining the outputs the faults are detected. The design fails to detect the location of adders under fault. The other major problem is fault propagation.

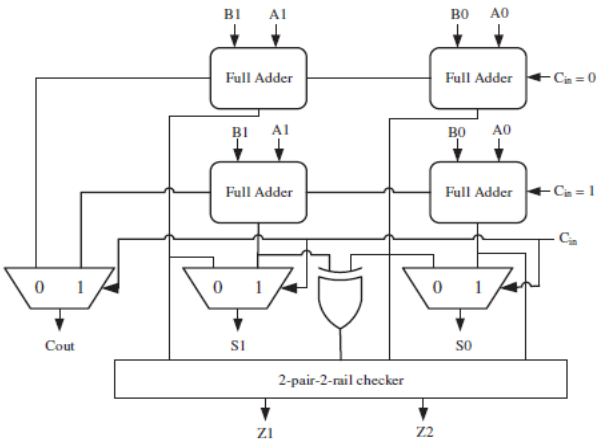

Figure 1. 2-Bit self checking carry select adder

The self repairing adder presented by Mohammad ali akbar et at (2014) detects single and multi-net faults with their location as shown in Figure 2. The online detection faces problems in self checking and testing. The testing and repairing was performed by XNOR gates and adders. The outputs of the XNOR are compared to identify the fault.

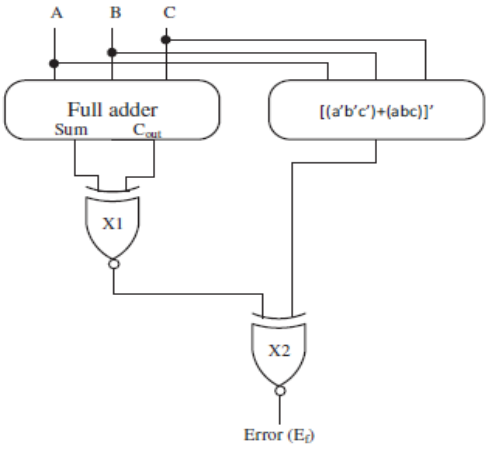

Figure 2. Self checking full adder

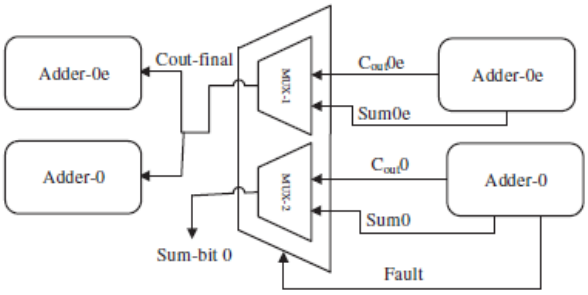

Figure 3. Self repairing adder

The previous methods are based on SOC based design but the proposed design is reconfigurable. In existing method the self-repairing adder used two identical adders which is a hardware redundancy method. The architecture is large and fails if both the adder fails (Muhammad Ali Akbar et al,). The method does not affect the performance but the area and power increases. The area overload can be avoided by using time redundancy but the penalty paid is the delay. Information redundancy needs circuits which should work error free. They are not compatible with memory circuits. The fault detection can be solved through addition program line when software redundancy is used. (Tsai, 1998)

3. On-Chip Methodology of Fault Tolerant Full Adder

The system of adders which repairs the single and double fault without disturbing the normal operation from both outputs is presented by Pankaj Kumar et al (2017). The carry select adder (CSA) compared to RCA, it has better computation time and minimum hardware cost. In this work the authors checked online and detected the faults using the sum and carry. In figure 4 shows the self checking adder structure where the XNOR does the detection operation of the faults. The fault is detected on comparison of (A'B'C + ABC'), (XNOR-1) and (XNOR-5) are made

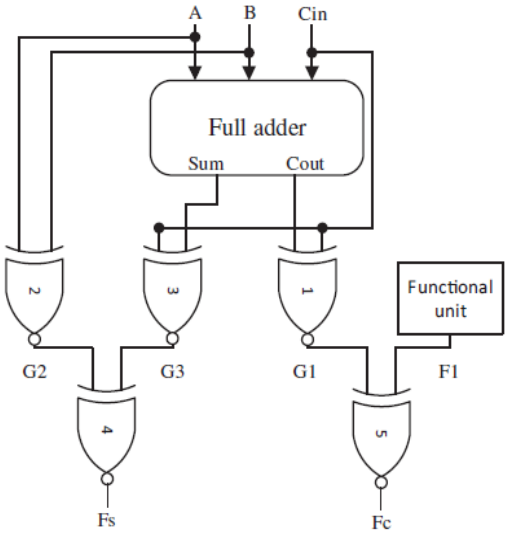

Figure 4. Self checking Full adder design

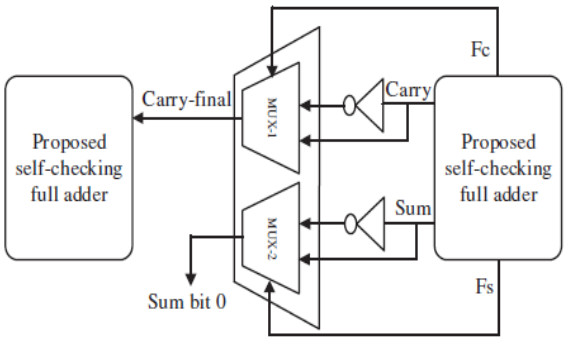

Figure 5. Self repairing Full adder design

When fault is appeared, it is checked and repaired by the carry select adder and self repairing adder respectively. This design repairs all faults.

1. The combined input is given to the full adder cell dependent on the sum and carry outputs either 1 or 0.
2. If signal Fs is controlled by inverted sum output of multiplexer then it is sum fault.
3. If signal Fc is controlled by inverted carry output of multiplexer then it is carry fault.

The operations are managed by the control signals.

The previous methods are based on SOC based design but the proposed design is reconfigurable. In existing method the self-repairing adder used two identical adders which is a hardware redundancy method. The architecture is large and fails if both the adder fails (Muhammad Ali Akbar et al,). The method does not affect the performance

but the area and power increases. The area overload can be avoided by using time redundancy but the penalty paid is the delay. Information redundancy needs circuits which should work error free. They are not compatible with memory circuits. The fault detection can be solved through addition program line when software redundancy is used. (Tsai, 1998)

4. Proposed DWT Architecture with Adder

For Biomedical systems efficient methods for noise removal and information retrieval using Very Large Scale Integration technology is needed. The hardware design can be of ASIC or FPGA type. Most of biomedical signal and image processing application uses wavelet transform and its hardware implementation is required. Wavelet transform is implemented using filter banks. The structures used are direct form structure, lattice structure, poly-phase decomposition etc. The problems in the convolution-based DWT architecture like more area and processing elements are rectified using the Lifting- based architecture. The block diagram of the lifting scheme is shown in figure 6 and the different blocks in Figure 7. The architecture contains adders, multipliers and Flip-Flops.

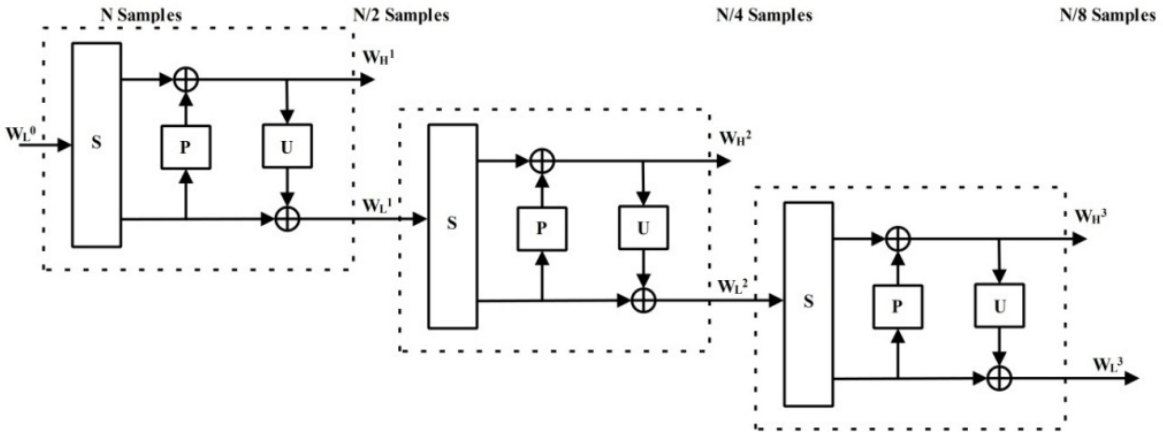

Figure 6. Block Diagram of the Lifting Based Scheme

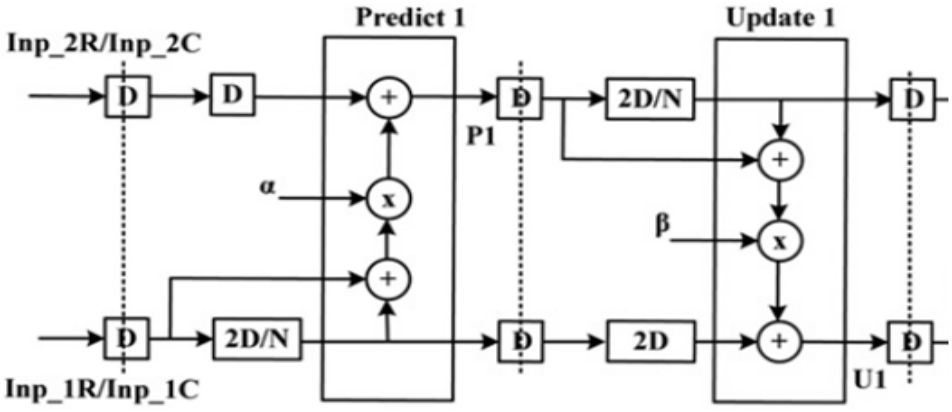

Figure 7. Lifting based DWT Architecture unit. The adder block is the Self Repair Fault Tolerant Adder

The proposed lifting-based design and convolution based design has components like self repair fault tolerant adders, multipliers and delay elements. The paper did not address the design of memory elements, but left for future work. The use of memory is to store the intermediate data of predict and update blocks. The extra overhead on area is the only issue in the proposed architecture but the design has less computation and higher speed. The speed is tested with SPICE tools. The circuits can operate till 4GHz. The error checking adder is used in the convolution based architecture also but it occupies more memory for storing the filter co-efficients. But the proposed lifting based DWT architecture utilizes self repair fault tolerant adder in the predict and update block. The different blocks of the DWT architecture is shown in table 1.

The work is extended for the convolution based design. Direct convolution methods are inefficient. Due to the inefficiency in the structure, poly-phase structure design is chosen. Here the filter coefficients are splited as even and odd samples and convolved. DWT architecture for biomedical systems should be efficient and in this work the efficiency is improved by using the adder in the computational unit which comprises the adder, multiplier, multiplexer and D-Flip Flop. Since the poly-phase structure (Figure 8.(a)) uses less number of multipliers when compared to other structures the additional fault detecting and correcting features of the adder will enhance its performance. These structures are suitable for biomedical signal processing. The 3 tap-broadcast FIR filter with lattice structure is shown in figure 8 (b).

Table 1. Different Blocks in the lifting - based DWT Architecture

DWT Blocks	Equivalent VLSI Blocks	VLSI Blocks Used in This Work
Co-Efficient Storage Block	Memory Device	Memory Device
Decomposition Block	Multiplexer, Delay Unit	Multiplexer with Gating, Delay Unit
Predict Block	Delay Element Multiplier (4*4) Full Adder Register	Gated Delay Element Proposed Multiplier (4*4) Proposed Full Adder Register
Update Block	Delay Element Multiplier (4*4) 7.Full Adder Register	Gated Delay Element Proposed Multiplier (4*4) Proposed Full Adder Register

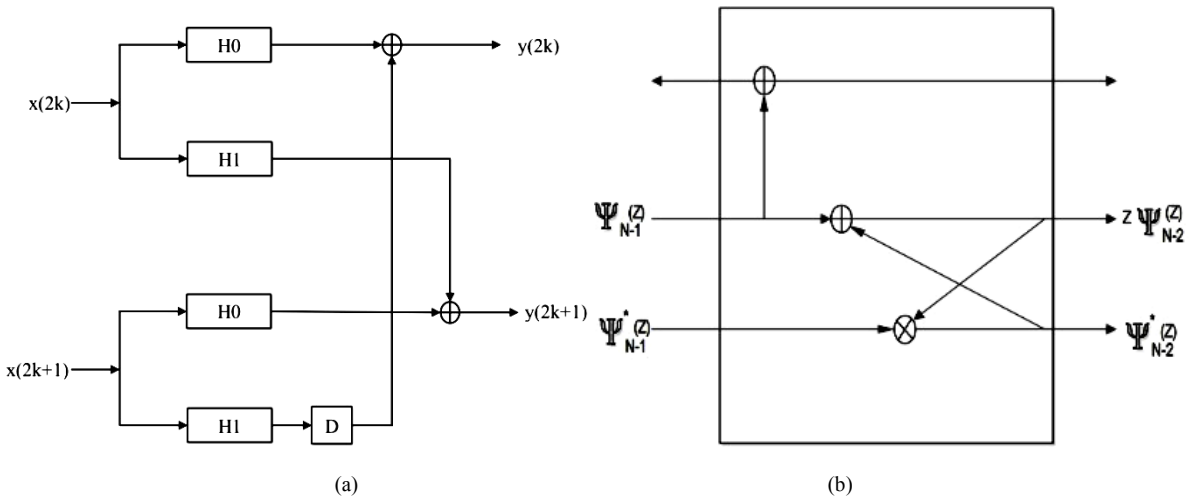

Figure 8. (a) Polyphase decomposition technique (b)N-th order lattice structure

4.1. Multipliers

The computational unit of processor core in DWT architecture contains the multiplier as the main block. The block along with adder forms the MAC unit. For convolution based DWT the MAC is the main unit while for lifting scheme shift registers and adders forms the basic units. The proposed DWT architecture uses the Braun Multiplier, the fault tolerant adder, D flip flop and MAC unit. Conventional Array multiplier is easy to design but occupies more area and power. The effective multiplication through parallel architecture can be achieved using the Braun multiplier. In this work the optimization in multiplier and flip-flops are less addressed. The optimization is taken forward for the future research, where gating principles are to be designed to reduce power further. The rest of the blocks are the coefficient units which are to be implemented using RAM cell. In the proposed FPGA based DWT structure filter coefficients are programmable while the adder, multiplier and other blocks follow the proposed design. These units also can be reconfigurable for future enhancements if required. The proposed reconfigurable architecture can be extended for IDWT processing architecture design also which comprises of multiplexer, multiplier, Flip flop, adder and sampling blocks The up samplers are also implemented using flip-flops. For reconstruction of input coefficients of both HPF and LPF are selected and using multiplier and adder.

The area overload increases as additional transistors are used for error checking and correcting.

5. Result and Discussion

The existing methods are mainly implemented on system on chip architectures using various devices. The results are given below. These results are based on device. But in this work the results are computed using reconfigurable architecture which is different from the technology used in existing methods in ASIC ICs. The implementation of an adder, multiplier, multiplexer, flip flop, blocks are done using Quartus software for different kits. For different technologies (Table.2 and Table.3), the adders are evaluated using the metrics like LUT, power and delay are measured and tabulated. Similarly the Table 4, 5 and 6 shows the performance of the Multiplier, D Flip flop and multiplexer for different technologies. The CMOS technology 90nm and 65nm were used for implementation.

Table 2. Parameter analysis of Self Check Adder in different nm technologies

FPGA Family	Device	LUT			Power Dissipation (mW)				Delay (nS)
		Available	Used	Utilization (%)	Core Dynamic Power Dissipation	Core Static Power Dissipation	I/O Thermal Power Dissipation	Total Thermal Power Dissipation	
Cyclone II	EP2C5F256C6	4608	3	<1	11.00	7.00	11.82	29.82	11.293
Cyclone III	EP3C5F256C6	5136	3	<1	30.08	16.03	9.53	55.64	7.809
Stratix II	EP2S15F484C3	12480	3	<1	228.25	74.72	20.74	323.71	10.206
Stratix III	EP3SL50F484C2	38000	3	<1	245.46	124.55	27.24	397.26	8.978

Table 3. Parameter analysis of Self Repair Adder in different nm technologies

FPGA Family	Device	LUT			Power Dissipation (mW)				Delay (nS)
		Available	Used	Utilization (%)	Core Dynamic Power Dissipation	Core Static Power Dissipation	I/O Thermal Power Dissipation	Total Thermal Power Dissipation	
Cyclone II	EP2C5F256C6	4608	2	<1	0.30	18.02	22.13	40.45	9.813
Cyclone III	EP3C5F256C6	5136	2	<1	0.13	46.12	14.44	60.68	6.987
Stratix II	EP2S15F484C3	12480	2	<1	1.23	303.08	30.32	334.63	9.697
Stratix III	EP3SL50F484C2	38000	2	<1	0.16	370.12	33.68	403.95	7.513

Table 4. Parameter analysis of Braun Multiplier in different nm technologies

FPGA Family	Device	LUT			Power Dissipation (mW)				Delay (nS)
		Available	Used	Utilization (%)	Core Dynamic Power Dissipation	Core Static Power Dissipation	I/O Thermal Power Dissipation	Total Thermal Power Dissipation	
Cyclone II	EP2C5F256C6	4608	183	4	2.50	15.50	14.74	32.75	25.414
Cyclone III	EP3C5F256C6	5136	183	4	6.05	40.08	11.87	58.01	20.805
Stratix II	EP2S15F484C3	12480	157	1	10.40	290.58	21.62	324.60	16.747
Stratix III	EP3SL50F484C2	38000	157	<1	50.03	320.00	28.00	398.03	15.793

Table 5. Parameter analysis of DFF in different nm technologies

FPGA Family	Device	LUT			Power Dissipation (mW)				Delay (nS)
		Available	Used	Utilization (%)	Core Dynamic Power Dissipation	Core Static Power Dissipation	I/O Thermal Power Dissipation	Total Thermal Power Dissipation	
Cyclone II	EP2C5F256C6	4608	14	<1	5.71	80.04	60.33	146.09	4.285
Cyclone III	EP3C5F256C6	5136	14	<1	2.64	46.13	23.68	72.45	2.527
Stratix II	EP2S15F484C3	12480	14	<1	5.41	303.32	48.66	357.39	3.337
Stratix III	EP3SL50F484C2	38000	14	<1	6.25	370.36	43.04	419.66	3.573

Table 6. Parameter analysis of Mux in different nm technologies

FPGA Family	Device	LUT			Power Dissipation (mW)				Delay (nS)
		Available	Used	Utilization (%)	Core Dynamic Power Dissipation	Core Static Power Dissipation	I/O Thermal Power Dissipation	Total Thermal Power Dissipation	
Cyclone II	EP2C5F256C6	4608	6	<1	1.29	18.09	53.61	72.99	10.169
Cyclone III	EP3C5F256C6	5136	6	<1	1.35	46.14	29.44	76.93	8.210
Stratix II	EP2S15F484C3	12480	6	<1	3.74	303.40	57.78	364.92	9.749
Stratix III	EP3SL50F484C2	38000	6	<1	3.21	370.45	51.49	425.15	10.184

The implementation in various technologies shows the difference in power consumption of different blocks of the DWT architectures. There are variations in power consumption, area and delay in different blocks. The implementation using Cyclone III kit i.e. 65nm CMOS technology has optimized performance when compared to other devices for different blocks.

6. Conclusions

The paper presents the FPGA implementation of the faster, less area self-checking and self-repairing fault tolerant adder circuit for the DWT architecture. The circuit provides low power consumption. The full adder design is suitable for Biomedical portable devices since it is fault tolerant and provide high efficiency. The adders for checking/repairing, multiplier, D-FF, and multiplexer of the DWT architecture were designed and implementation was done in Cyclone and Stratix kit using Quartus Tool. The performance is evaluated for different CMOS technology. In future the work will be extended in optimizing the DWT structure and blocks like flip- flop and samplers.

REFERENCES

[1] Dhanasekar J, Aarthy Balan, Akshara R, Anusuya P & Deepa S 2019, 'Design of Various Adders Using Self Fault Detecting Full Adder', International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 08, no. 05, pp. 581 – 584.

[2] Mendoza-Hernandez, Linares-Aranda F M & Victor Champac 2006, 'Noise-tolerance improvement in dynamic CMOS logic circuits', IEEE Proceedings-Circuits, Devices and Systems, vol. 153, no. 06, pp. 565 – 573.

[3] Gnana Pallavi Panchumarthi & Sarada Musala A Surendar 2017, 'A Review Article on Fin-FET based Self-Checking

Full Adders', Journal of Advanced Research in Dynamical & Control Systems, vol. 09, no. 04, pp. 1 – 8.

[4] İlke Ercan, Ömercan Susam, Mustafa Altun & Hüsrev Cilasun M 2017, 'Synthesis and fundamental energy analysis of fault-tolerant CMOS circuits', 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), pp. 1 – 4.

[5] Sonal Gupta, Shweta Meena & Vahiuddin Syed Khaja 2018, 'Real-Time Double Fault Tolerant Full Adder Design Using Fault Detection', Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1 – 6.

[6] Parminder Kaur & Balwinder Singh Dhaliwal 2012, 'Design of fault tolearnt full Adder/Subtarctor using reversible gates', International Conference on Computer Communication and Informatics, pp. 1 – 5.

[7] Yasamin Mahmoodi & Mohammad A Tehrani 2014, 'Novel fault tolerant QCA circuits', 22nd Iranian Conference on Electrical Engineering (ICEE), pp. 1 – 6.

[8] Prachi Palsodkar, Prasanna Palsodkar & Rupali Giri 2018, 'Multiple Error Self Checking-Repairing Fault Tolerant Adder-Multiplier', IEEE Region 10 Humanitarian Technology Conference (R10-HTC), pp. 1 – 4.

[9] Ankur Sarker, Avishek Bose & Shalini Gupta 2014, 'Design of a compact fault tolerant adder/subtractor circuits using parity preserving reversible gates', 17th International Conference on Computer and Information Technology (ICCIT), pp. 1 – 7.

[10] Sajib Kumar Mitra & Ahsan Raja Chowdhury 2015, 'Minimum Cost Fault Tolerant Adder Circuits in Reversible Logic Synthesis', 25th International Conference on VLSI Design, pp. 334 – 339.

[11] Ning-Chi Huang, Szu-Ying Chen & Kai-Chiang Wu 2019, 'Sensor-Based Approximate Adder Design for Accelerating Error-Tolerant and Deep-Learning Applications', Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 692 – 697.

[12] Gurmohan Singh, Balwinder Raj & Rakesh K Sarin 2018, 'Fault-tolerant design and analysis of QCA-based circuits', IET Circuits, Devices & Systems, vol. 12, no. 05, pp. 638 – 644.

[13] Vasudevan D P, Lala P K & Parkerson J P 2007, 'Self-checking carry-select adder design based on two-rail encoding', IEEE Transaction Circuits System-I Regular Paper, vol. 54, no. 12, pp. 2696 – 2705.

[14] Ali Akbar & Lee 2014, 'Self repairing adder using fault localization', Micro-electronics and Reliable system, vol. 54, no. 06, pp. 1443 – 1451.

[15] Pankaj Kumar & Rajender Kumar Sharma 2017, 'Double fault tolerant full adder design using fault localization', 3rd International Conference on Computational Intelligence & Communication Technology (CICT), pp. 1 – 6.

[16] Tsai, T.: Fault tolerance via N-modular software redundancy. In: Twenty-Eighth Annual International Symposium on Fault-Tolerant Computing Digital Paper, pp. 201–206. IEEE (1998).

[17] Akbar, Muhammad Ali, and Jeong-A. Lee. "Self-repairing adder using fault localization." Microelectronics Reliability 54, no. 6 (2014): 1443-1451.