VOL. 17, NO. 14, JULY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

IMPLEMENTATION OF AREA EFFICIENT AND HIGH SPEED HNG GATE BASED MULTIPLIER FOR DSP APPLICATIONS

K. Manikanta and N. Siddaiah
Department of Electronics and Communication Engineering, Klef, Andhra Pradesh, India
E-Mail: 2001080007@kluniversity.in

ABSTRACT

In the scientific literature, there is a lot of discussion on approximate multipliers and circuits based on approximate 4-2 compressors. The designer who wants to employ an approximation 4-2 compressor is confronted with the challenge of choosing the proper topology due to the enormous number of options. A complete study and comparison of the roughly 4-2 compressors that have been suggested in the literature are presented here. One more compressor is shown, so that there are now twelve distinct approximate 4-2 compressors evaluated. The goal is to create logic gates that can be reversed. Quantum computing relies heavily on the reversibility of logical operations. This technology's gadgets run at very fast speeds and utilize very little power. Hardware description language (HDL) is used to create simple reversible logic gates. The Verilog implementation of the Wallace tree multiplier uses a simple half adder and a full adder. The reversible logic gates have been created, as well as a 4bit reversible adder, irreversible adder along multiplier has been developed. Layouts are designed using a variety of foundry technologies, and these methods are compared. We've reached the end of the road in terms of low-power dissipation. An 8x8 bit reversible multiplier circuit has been suggested and developed in this article. In terms of speed and complexity, the suggested reversible multiplier is superior to the current multipliers. In terms of the number of gates, garbage outputs, and constant inputs, it is superior to the current alternatives. The "HNG" 8x8 reversible gate was recently suggested by Haghparast and Navi. It is possible to use a reversible HNG gate as a reversible full adder when it is used alone. The reversible multiplier circuit in this study is built using HNG gates. Two 8-bit binary integers may be multiplied using the HNG gate in the proposed reversible multiplier circuit. A generalized version of the suggested reversible 8x8 multiplier circuit may be used for NxN bit multiplication as well.

Keywords: reversible gates, power dissipation, verilog, adder, HNG multiplier.

INTRODUCTION

As one of the most crucial computing blocks, multipliers are commonly used in digital signal processor (DSP) systems. Examples include computer graphics, scientific computation, and image processing, where multipliers are increasing. Processor speed is influenced by the multiplier speed, and high-speed is prioritised above low-energy usage by designers. A partial production, reducing, & addition stage are all included in the multiplier design. The partial product reduction step receives the lot of the season, power, and range of multiplication. As a result, this step is often implemented by compressors, which assist minimise the amount of partial products as well as shorten the circuit's critical path.

This is done using the 3-2, 4-2, 5-2 framework. Compressor circuit with a full adder cell: The overall performance of the system will increase if these compressors are upgraded in design. XOR-XNOR and multiplexers are part of the compressor's internal construction. Mathematical systems, multipliers, compressors, parity-control systems and other circuits employ XOR-XNOR logic. XOR/XNOR Improved XOR-XNOR gates increase the performance of the multiplier circuit. New XOR-XNOR modules and a 4-2 compressor are proposed in this study. Product accumulation may be achieved by reducing transistor and power consumption.
Full adder computation has been widely explored as an approximation for full adder cells. These adders were compared by Liang et al., who came up with a slew of new metrics for evaluating approximation and probabilistic figures' supplements. It is the arithmetic distance between both the incorrect and the right outputs in a circuit that is referred to as the error distance (ED). Approximation multipliers, on the other hand, have not gotten the attention they deserve. Using approximation additional items in the architecture of a multiplier does not work since it is difficult, complex and inefficient in terms of performance measures. " An estimated multiplier was given to a number of applicants. Truncated multiplication is used in these designs, which estimate the constant-sized columns of partial products. When partials are multiplied using an incorrect array multiplier, this neural network mistake occurs (and thus removing some adders in the array). adjustment constant added to a shortened multiplier.

Low-power VLSI design has been more popular in recent years, and one of the most important needs is reducing the amount of power dissipated. These reversible logic gates allow the system to be operated in both directions, as well as produce inputs from outputs that may be paused and returned to any point in the computation history. As long as the output vector can be retrieved from of the input vector in exactly the same way, and vice versa, a circuit is said to be reversible. If computing is made information-lossless, energy dissipation may be decreased or even eliminated. Moore's law still holds true for higher-order integration and increasing scaling mechanisms, however with older technologies, the heat generated by each IC doubles.

When information is lost, the temperature drops by kTLn2 joules, where k is the Boltzmann constant & T

is the operating temperature, in conventional logic circuits. Low-power CMOS and QCA technologies make use of reversible logic gates because of their many benefits. It performs a wide range of tasks in a short period of time. Quantum computers can make better use of these gates since they are more malleable. Bit-loss recovery and heat generation are both possible uses for this technology. Inputs and outputs are objectively linked. A unique output is generated for each input. As a result, since reversible computing results in no data loss, it is the ideal option for QCA nanotechnologies. There are five primary ways in which reversible logic differs to irreversible Boolean logic synthesis: One definition of constant input is an input that is always set to 1 or 0. For such an efficient circuit, the number of stay true must be minimised. A logic gate's fan-out is the maximum number of gate inputs it can drive. Only one of gate's inputs may be linked to the gate's sole output. The output of a gate that is not utilised by another gate is known as a garbage signal. The lower the trash value, the more efficient the circuit will be. For example, with reversible circuits, the number of input and output pins is the same, unlike in irreversible circuits. Because there is no feedback route in this logic circuit, it is acyclic. The number of basic gates required to synthesise a particular logic function is a measure of hardware complexity. There is no loss of information owing to the one-to-one mapping idea in reversible computing. The number of inputs and outputs are the same in these gates. As a result, they're each able to be retrieved in a different way. Reconstructing inputs from outputs does not use any electricity. The energy that is dissipated by irreversible gates may be significant. Reversible gates may be used to overcome this sort of issue. The negligible power dissipation of reversible logic gates is a benefit. Reversible gates also have a high rate of operation. Online and offline fault testing are two examples of applications that make use of reversible logic. Constant inputs & garbage outputs are used to balance a reversible gate. In order to make a gate reversible, the quantity of gates must be maintained constant. Garbage outputs are those that have been thrown away. Unused outputs are not linked to any other inputs. Reversible logic gates are able to create both inputs and outputs from the same logic gate. Different variables including such garbage outputs, gate count, constant input, latency, and hardware complexity are used in the evaluation of reversible circuits.

EXISTING SYSTEM

Multiplier Using 4-2 Compressor

Full adder design for XOR and Multiplexer circuits was presented instead of traditional full adder design for three distinct 4:2 compressors. This may operate in either an accurate or an approximation computing mode, depending on the situation. Accuracy is sacrificed in favor of fast speed and low battery consumption in this approximation mode. When using an approximation mode, each compressor has a different degree of precision and power dissipation than when using an exact mode. Accuracy and dynamic power and speed changes may be achieved using these compressors inside the approximate multiplier. Many logic gates are saved by using an approximate 4:2 compressor-based partial multiplier, and this suggested multiplier was tested in image processing applications, such as multiplication and sharpening. Xilinx S6LX9 FPGA was used to generate this design in Verilog HDL and compare all parameters ln terms of area, delay, and power consumption.

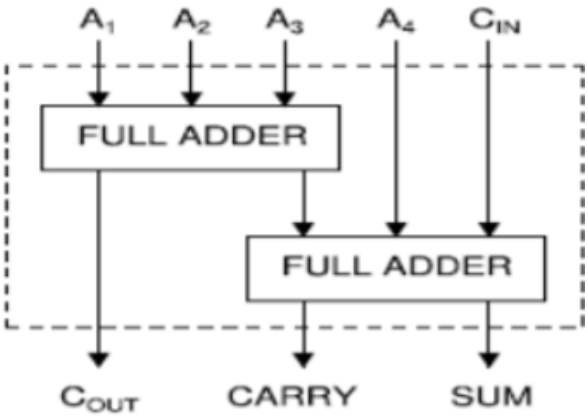

Figure-1. Conventional 4:2 compressor.

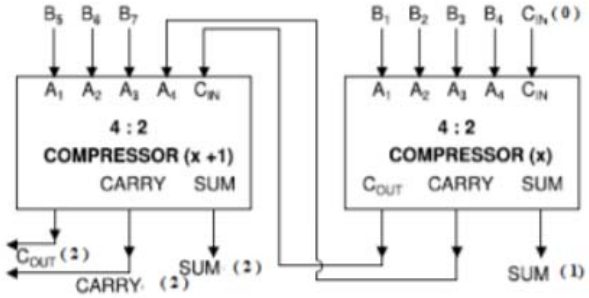

Figure-2. 4:2 Compressor chain.

Multiplication is an essential mathematical operation in AI and DSP applications. Because of the high-speed parallel nature of these applications, high-accuracy multiplier designs are required. Faster calculations may be achieved with less hardware complexity, latency, and power consumption by using approximation in multipliers. In the process of multiplication the propagating delay in an adder structure is caused by a speed limiting procedure called partial product summation. Compressors are used to reduce the propagation delay. Compressors concurrently compute the total and transport it through each step of the process. Next, the carryover is multiplied by a larger significant sum bit to get the final result. This cycle is repeated until the finished product is obtained. The A1, A2, A3, A4, and Cin inputs and outputs of a 4-2 compressor are as follows: (Sum; Carry; Cout). The binary weights of all the input and output bits are the same. A preceding block of ordering one binary bit lower in importance is fed into the compressor, which generates outputs Cout & Carry of

order 1 binary bit higher in importance. Figure-1 depicts the overall block diagram of the precise 4-2 compressor.

SUM = A1 XOR A2 XOR A3 XOR A4 XOR Cin ---------- (1)

CARRY= Cin (A1 XOR A2 XOR A3 XOR A4) +A4 (~ (A1 XOR A2 XOR A3 XOR A4)) ---- (2)

Cout = A3 (A1 XOR A2) +A1 (~ (A1 XOR A2)) ----------(3)

Two output bits may be used to create the approximate 4-2 compressor. Using this approximation compressor as a multiplier implementation reduces the amount of output bits, which in turn decreases the amount of input bits for subsequent compressors. Output bit Cout is always 0 except for the input combination "X4X3X2X1 = 1111". Cout is not taken into account while developing approximation compressors.

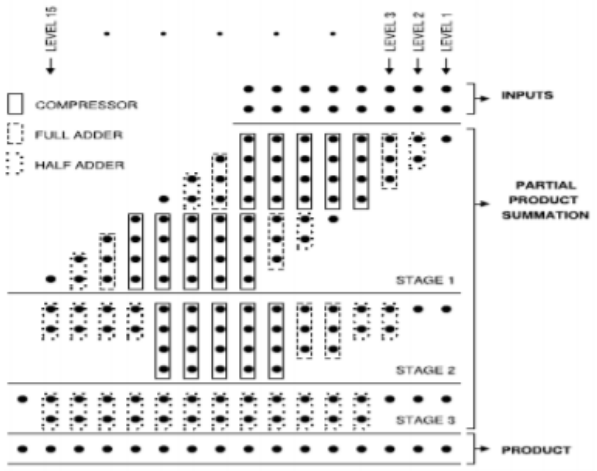

Figure-3. 8×8 Approximate multiplier.

PROPOSED SYSTEM

Low-power design, optical processing information, DNA computing, informatics, quantum computing, and nanotechnology all benefit from reversible logic circuits because of their interest in minimizing power consumption. A new 8x8 bit reversible multiplier circuit is proposed in this study. In terms of speed and complexity, the suggested reversible multiplier is superior to the current multipliers. In relation to the number of gates, garbage outputs, and constant inputs, it is superior to the current alternatives. The "HNG" gate, suggested by Haghparast and Navi, is an 8x8 reversible gate. It is possible to use a reversible HNG gate as a reversible adder circuit when it is used alone. The reversible multiplier circuit in this study is built using HNG gates. Two 8-bit binary integers may be multiplied using the HNG gate in the allows us to change multiplier circuit. NxN bit multiplication may be used to the allows us to change 8x8 multiplier circuit.

You can undo the reasoning. Function with n-inputs and n-outputs if the inputs and outputs of F are identical, then F is said to have been reversible. As a result, the output vector may be used to uniquely identify the input vector. logic gates using reversible logic: As an example, the following equation may be used to describe a nxn reversible gate: IV = (I1,I2,I3,...,In) OV = (O1,O2,O3,...,On). In recent years, a number of other reversible gates have indeed been put out for consideration. Eighteen of the most important gates are located between them: the Feynman gate, FFG [8], the Toffoli gate (TG [9], the Fredkin gate (FRG [10], and the Peres and New Gate (NG [12]). We'll go over reversible logic gates in this part. Many of them are offered so that they may be compared to previous research. 2-CNOT, also known as the Feynman gate, is a 2x2 reversible gate that may be defined by the following two equations: P = B and Q = A*B, where A is the control bit and B denotes the data bit. In Figure-1, it is displayed. A 3x3 reversible logic gate, the Toffoli gate (TG) is also known as a controlled controlled-not (CCNOT). An example of the Toffoli gate's input and output vectors: IV = A, B, C; OV = (P = A, Q = B, R = ABC) Figure-2 depicts the Toffoli gate. Three-by-three reversible logic gates are called Fredkin gates (FRGs). In mathematical terms, IV = (A, B, C) Inlet and outlet vectors are IV and OV.it is shown. This gate is conservative in the sense that the Hamming value of its input and output vectors are the same. A 3x3 reversible logic gate, the Peres gate (PG) or New Toffoli Gate (NTG), combines the Toffoli Gate with the Feynman Gate. It's written as: Iv =. (A, B, C) P = A, Q=A+B+C, R=AB+C where Iv and Ov are indeed the input and output vectors respectively. Figure-4 depicts the Peres gate. Toffoli Gate coupled by Feynman Gate produces the same transformation as Peres gate. Reversible 3x3 new gate (NG). It's written as: Iv =. (A, B, C) A is the input and B is the output vector of Ov = (P=A, Q=ABC, R=A′C′B′). Figure-5 depicts the new gate. Four by four reversible logic gate called the TSG gate. Figure 6 depicts the TSG gate with its outputs labelled with logic expressions. A 4x4 reversible gates gate, the MKG gate is one kind. Iv = (A, B, C, D) Ov = (P = A, Q = C, R = (A′D′B′) C, S = (A′D′B′). The MKG gate may be expressed as: For the input and output vectors, C(ABD)). MKG gate is seen in Figure-7 with each output labelled with a logic statement. 4x4 reversible logic gate, the HNG gate. To describe the HNG gate mathematically, we may write it as follows: Iv = ((A, B, C, D)) Ov = C⊕ AB⊕ D) The inlet and outlet vectors are Iv and Ov, respectively.

NOVEL REVERSIBLE MULTIPLIER CIRCUIT USING HNG GATES

This one-to-one mapping makes it possible to reverse the logic gate's inputs and outputs. This logic gate helps decide the outputs from the inputs, and it may also be used to recover the inputs from the outputs. Using a limited set of reversible logic gates, the principles of reverse computing are founded on the link between entropy & heat transport inside the system, the chance that a quantum particle will ever be in a specific state. To determine the intricacy and performance of a reversible circuit, there are a number of parameters: Reversible gates (N): A overall number of reversible logic employed in the circuit. To synthesise the given logical function, the

number of inputs must be kept constant at either 1 or 0. This is known as the b) constant inputs (CI) definition. There are four "garbage outputs" in a reversible logic circuit, which means that there are four "unused" outputs. It is impossible to go without them in order to accomplish reversibility. To put it another way, "quantum cost" (QC) is how much a single basic gate costs. Gate levels (GL) represent the number of gates inside the circuit that are necessary to implement a certain logic function.

Reversible logic gates like the HNG gate are shown in Figure-4. Because it has k inputs & k outputs, the HNG gate is referred to as a k*k gate. The quantum cost of an HNG gate is six. Reversible logic gates have been used in several combinational logic circuits during the last few decades. In a single gate, it may create both sum and carry, thereby reducing the amount of waste and the number of gates needed. The suggested multiplier uses just one HNG gate to structure the whole multiplier in order to prevent this.

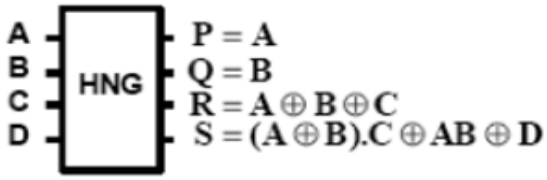

Fig 4: HNG gate

Reversible logic is gaining a lot of traction among IC designers right now since it consumes less power. Applications of reversible logic include DSP (Digital Signal Processors), quantum computers, and high-speed VLSI structures. Many reversible logic gates are used in the implementation of reversible logic. A low power VLSI design approach known as reversible will be utilised to minimize the energy consumption and space usage of a VLSI design without sacrificing the speed. Reversible logic gates are used in this project to create a high-speed multiplier. Verilog coding has been used to suggest an efficient high-speed multiplier. High-speed multiplication is provided using an 8-bit reversible multiplier. The reversible 8x8 multiplier circuit we're proposing comprises two components. To begin, Peres gates are used to create the partial products in parallel. A full adder (FA) that accepts three bits is the underlying cell for this multiplier. The reversible complete adder is represented using HNG gates. Reversible HNG full adders are used in the circuit design of the reversible multiplier suggested here. It also requires four half-adders that may be reversed. As an alternative to the HNG gate, which is more complicated in hardware and has a higher quantum cost, we employ the Peres gate as a reversible half adder. A comparable or slightly larger set of gates are used in reversible computation and other developing technologies including quantum computation, optical computing, and nanotechnology. It's also been possible to create the first reversible logic chips using CMOS technology. In opposed to static CMOS logic, which sinks all signal energy for each gate, reversible logic is ideal for reusing signal energy (as opposed to sinking the transmitted signal with each gate). It has been shown that reversible circuits may lower energy usage by a factor. These implementations suffer from another transistor rule, which states that power consumption is exactly proportional to the frequency at which instructions are executed. The energy usage per calculation climbs as the number of calculations per second increases. Per calculation, less energy is used if fewer computations are performed at the same time. Reversible circuits may be used for a wide range of applications; however, this means that not all of them are appropriate for use. While many embedded devices don't need to conduct billions of calculations per second, there are those that do. The remainder of this section explains how reversible gates are implemented in CMOS. CMOS transistor implementation fundamentals are reviewed first, and then an explanation of how this is employed in the development of reversible gates is provided. There are many different kinds of reversible gates, as listed below. The HNG gate is important because it may be used to construct a reversible complete adder circuit. (A, B, Cin), the output vector of HNG will be (P=A;Q=Cin;R=Sum) when the input vectors are (A;B). As a result, it's possible to receive both sums and carryouts. Figure-5 shows the implementation of the HNG gate-based reversible full adder. Two of the four outputs are sum and carry, with the other two being waste. As the multiplier's most important unit, the adder has a significant impact on the overall performance of the system, including power consumption, delay, and space required. To create an 8-bit binary multiplier, you'll need six 4-bit ripples carry adders. Figure-4 shows the schematic design of an 8-bit ripple carry adder employing 9 HNG gates.

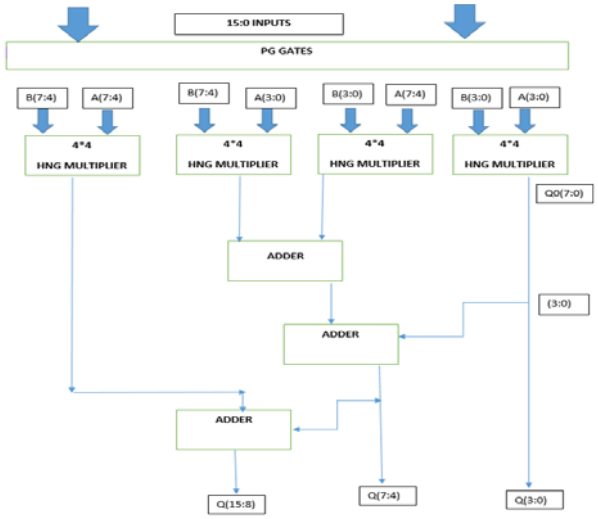

Fig. 5: Proposed 8*8 reversible multiplier circuit using HNG gates and Peres gates

SIMULATION RESULTS

Using Verilog HDL, we've generated all of our synthesis and simulation findings. Xilinx ISE 14.7 is used for the synthesis and simulation. Figures following illustrate the simulation results. Below are the matching Approximate 4:2 Multiplier simulation results.

EXISTING METHOD

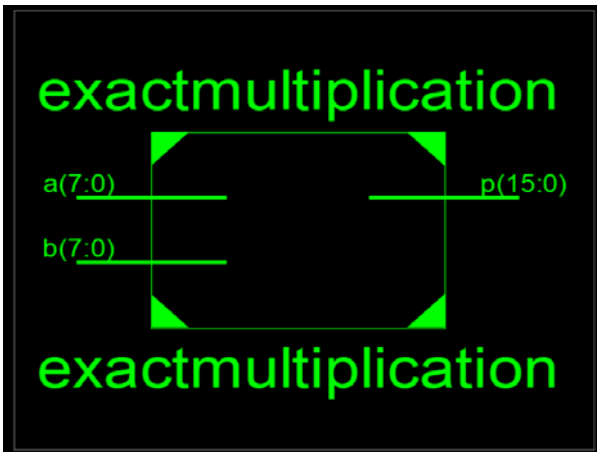

Figure-6. RTL schematic.

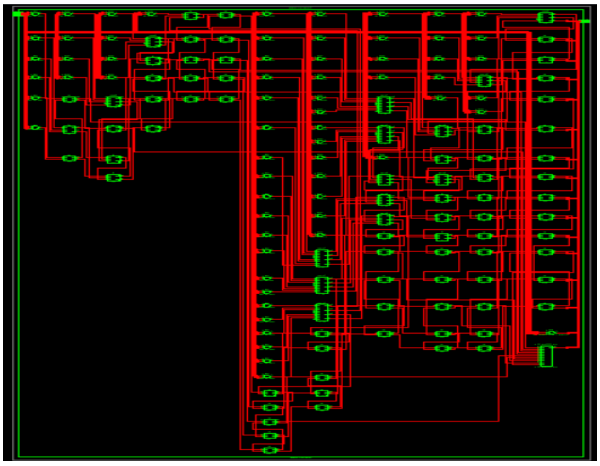

Figure-7. Technology schematic of internal block multiplier.

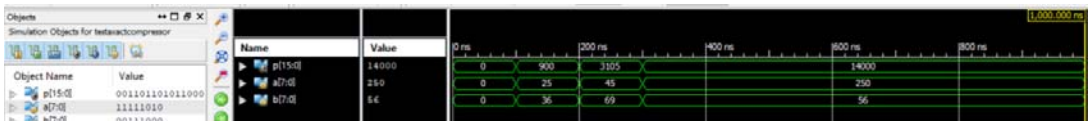

Figure-8. Simulation results.

PROPOSED RESULTS

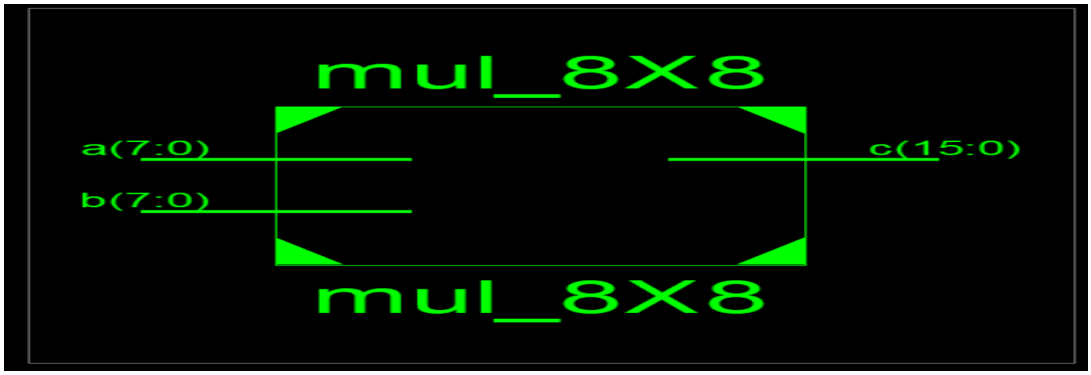

Figure-9. RTL schematic.

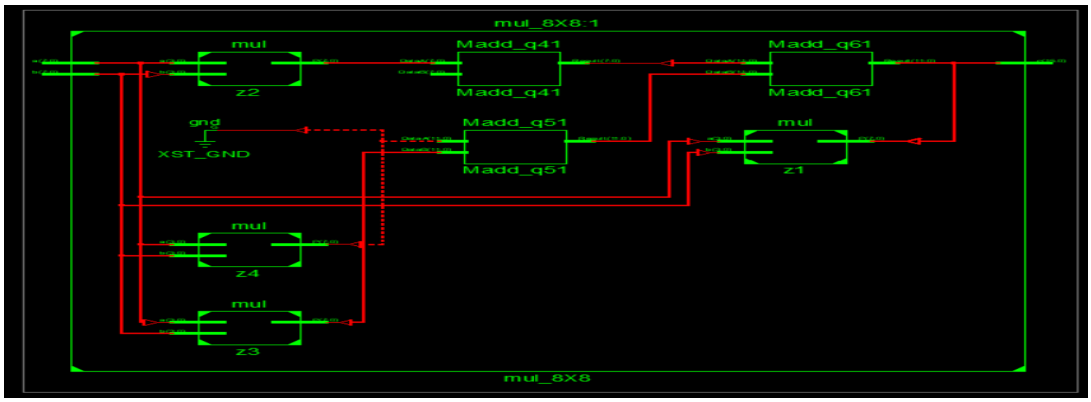

Figure-10. Technology schematic of internal block multiplier.

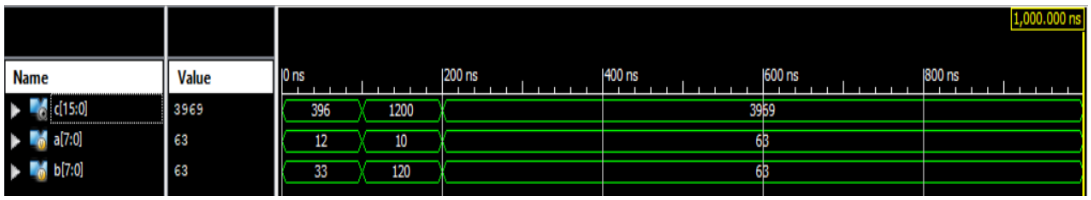

Figure-11. Simulation results.

AREA

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	121	9,312	1%	
Number of occupied Slices	61	4,656	1%	
Number of Slices containing only related logic	61	61	100%	
Number of Slices containing unrelated logic	0	61	0%	
Total Number of 4 input LUTs	121	9,312	1%	
Number of bonded IOBs	32	232	13%	
Average Fanout of Non-Clock Nets	3.21			

DELAY

```
    ----------------------------------------
    Total                      19.483ns (13.522ns logic, 5.961ns route)
                                        (69.4% logic, 30.6% route)

=========================================================================


Total REAL time to Xst completion: 5.00 secs
Total CPU time to Xst completion: 4.39 secs
```

POWER

Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Temp Grade	Commercial
Process	Typical
Speed Grade	-5

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	121	9312	1
Signals	0.000	145	--	--
IOs	0.000	32	232	14
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

Supply	Power (W)	Total	Dynamic	Quiescent
		0.081	0.000	0.081

Table-1. Comparison table.

Parameter	Existing Method	Proposed system
LUTS	155 LUTS	121 LUTS
Delay	21.9.12 NS	19.483 NS
Power	0.081 mW	0.081 mW

CONCLUSIONS

This project developed an effective reversible multiplier based only on HNG gates. The reversible gate built on ripple carry adder was used to design the schemed multiplier. Because the HNG gate is a reversible logic gate, it may be utilized to lessen the delay when the number of bits increases. Verilog coding is used to model the multiplier circuit, which is then run in Xilinx ISE 14.7. This Table-1 shows that now the delay is shorter than it was with the previous multiplier. DSP and high-speed VLSI circuits will benefit from the suggested multiplier.

Although delay minimization has typically been the primary emphasis of multiplier design, power consumption concerns have lately been added to the list of priorities. First and foremost, we wanted to figure out how multipliers use power and, second, how we might cut down on it. We have investigated the power dissipation of a multiplier, as well as several strategies for reducing this circuit's power usage. In light of the relevance of multipliers, subsequent research efforts will likely focus on improving this blocks for delay and power consumption. Utilizing HNG gates & Peres gates, we devised a new 8x8 bit reversible multiplication circuit. A comparison of the hardware cost, amount of gates, garbage outputs, and constant inputs and outputs demonstrates that the new reversible multiplier circuit performs better than the current designs (see Table-1). The limitations of reversible circuits too were avoided to the greatest extent possible. Reversible multiplier circuits, such as the one we've presented, may be used to create complicated nanotech systems. There are no quantum logic or optical logic implementations available, hence all of the suggested circuits are technology-independent.

REFERENCES

[1] M. Alioto. 2012. Ultra-low power VLSI circuit design demystified and explained: A tutorial. IEEE Trans. Circuits Syst. I, Reg. Papers. 59(1): 3-29.

[2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy. 2013. Low-power digital signal processing using approximate adders. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 32(1): 124-137.

[3] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie and C. Lucas. 2010. Bio-inspired imprecise computational blocks for efficient VLSI implementation of softcomputing applications. IEEE Trans. Circuits Syst. I, Reg. Papers. 57(4): 850-862.

[4] R. Venkatesan, A. Agarwal, K. Roy and A. Raghunathan. 2011. MACACO: Modeling and analysis of circuits for approximate computing. in Proc. Int. Conf. Comput.-Aided Design. pp. 667-673.

[5] F. Farshchi, M. S. Abrishami and S. M. Fakhraie. 2013. New approximate multiplier for low power digital signal processing. in Proc. 17th Int. Symp. Comput. Archit. Digit. Syst. (CADS). pp. 25-30.

[6] P. Kulkarni, P. Gupta and M. Ercegovac. 2011. Trading accuracy for power with an underdesigned multiplier architecture. in Proc. 24th Int. Conf. VLSI Design. pp. 346-351.

[7] D. R. Kelly, B. J. Phillips, and S. AlSarawi. 2009. Approximate signed binary integer multipliers for arithmetic data value speculation. in Proc. Conf. Design Archit. Signal Image Process. pp. 97-104.

[8] K. Y. Kyaw, W. L. Goh and K. S. Yeo. 2010. Low-power high-speed multiplier for error-tolerant application. in Proc. IEEE Int. Conf. Electron Devices Solid State Circuits (EDSSC). pp. 1-4.

[9] A. Momeni, J. Han, P. Montuschi and F. Lombardi. 2015. Design and analysis of approximate compressors for multiplication. IEEE Trans. Comput. 64(4): 984-994.

[10] K. Bhardwaj and P. S. Mane. 2013. ACMA: Accuracy-configurable multiplier architecture for error-resilient system-on-chip. in Proc. 8th Int. Workshop Reconfigurable Commun.- Centric Syst.-Chip. pp. 1-6.

[11] K. Bhardwaj, P. S. Mane and J. Henkel. 2014. Power- and area-efficient approximate wallace tree multiplier for error-resilient systems. in Proc. 15th Int. Symp. Quality Electron. Design (ISQED). pp. 263-269.

[12] J. N. Mitchell. 1962. Computer multiplication and division using binary logarithms. IRE Trans. Electron. Comput. EC-11(4): 512-517.

[13] V. Mahalingam and N. Ranganathan. 2006. Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition. IEEE Trans. Comput. 55(12): 1523-1535.

[14] Nangate 45nm Open Cell Library, accessed on 2010. [Online]. Available: http://www.nangate.com/

[15] H. R. Myler and A. R. Weeks. 2009. The Pocket Handbook of Image Processing Algorithms in C. Englewood Cliffs, NJ, USA: Prentice-Hall.

[16] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park and N. S. Kim. 2015. Energy-efficient approximate multiplication for digital signal processing and classification applications. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 23(6): 1180-1184.

[17] S. Hashemi, R. I. Bahar and S. Reda. 2015. DRUM: A dynamic range unbiased multiplier for approximate applications. in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Austin, TX, USA. pp. 418-425.

[18] C.-H. Lin and I.-C. Lin. 2013. High accuracy approximate multiplier with error correction. in Proc. 31st Int. Conf. Comput. Design (ICCD). pp. 33-38.

[19] A. B. Kahng and S. Kang. 2012. Accuracy-configurable adder for approximate arithmetic designs. in Proc. 49th Design Autom. Conf. (DAC). pp. 820-825.

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli 2004. Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. 13(4): 600-612.