International Journal of New Computer Architectures and their Applications (IJNCAA) 4(4): 201-210
The Society of Digital Information and Wireless Communications, 2014 (ISSN: 2220-9085)

Design of a Reversible Fused 32-Point Radix -2 Floating Point FFT Unit Using 3:2 Compressor

A.V.AnanthaLakshmi[1], G.F. Sudha[2]
[1]Assistant Professor, Department of ECE, Pondicherry Engineering College, Puducherry, India
[2]Professor, Department of ECE, Pondicherry Engineering College, Puducherry, India

ABSTRACT

This paper aims on the design of a reversible fused 32-Point Radix-2 single precision floating point FFT unit using 3:2 compressor. The work focuses on the realization of three reversible fused floating point units: reversible floating point add-sub unit, reversible floating point multiply-add unit and reversible floating point multiply-subtract unit. The proposed work requires the design of a reversible single precision floating point adder, a reversible single precision floating point subtractor and a reversible single precision floating point multiplier. A reversible single precision floating point adder and subtractor is designed with less quantum cost, less number of reversible gates and less constant inputs. A reversible single precision floating point multiplier is implemented using 3:2 compressor as the 24x24 bit multiplier based on 3:2 compressor is highly efficient when compared with the design using 4:3 compressors. A reversible fused 32-Point Radix-2 floating point FFT Unit using 3:2 compressor consumes less number of resources, operates at a slightly greater speed and dissipates less power when compared with the reversible discrete 32-Point Radix-2 floating point FFT Unit. The proposed Fused 32-Point Radix-2 floating point FFT unit using 3:2 compressor dissipates 2.074W while the same design as a discrete implementation dissipates 2.176W.

KEYWORDS

Reversible floating point adder, Reversible floating point subtractor, Reversible floating point multiplier, Reversible Fused 32-Point Radix -2 floating point FFT Unit.

1 INTRODUCTION

The Discrete Fourier Transform (DFT) is extensively used in digital signal processing (DSP) algorithms. DFTs are computed using the Fast Fourier Transform FFT). FFT finds importance in many areas such as communications, signal processing, instrumentation, biomedical engineering, acoustics, numerical methods, and applied mechanics. The performance of the system as well as the functionality per die increases as the chip size decreases. Thus, the aim is to build resource, speed and power efficient architecture for FFT processors—with an importance on a VLSI implementation. Further as the performance of the system increases, power consumption also increases. Thus, a new methodology has to be evolved for to implement today's complex system with less consumption of power. Landauer in his research has shown that an erasure of a bit causes an information loss [1]. Further, he has pointed out that the use of conventional irreversible logic gates leads to power dissipation. Every irreversible bit operation dissipates kTln2 energy, where T is the absolute temperature at which the computation is performed, and k is Boltzmann's constant. Later Bennett showed that in order to avoid energy loss it is necessary that all the computations have to be performed in a reversible way [2]. Conventional logic circuits are irreversible.

A circuit is said to be reversible when there is a one – to –one mapping between the input and output assignment. A reversible circuit maps each input vector into a unique output vector and vice versa. Some of the important considerations involved in the design of a reversible logic circuit are to:

- Use minimum number of reversible gates.
- Use minimum number of garbage outputs.
- Use minimum constant inputs [3].

The output which cannot be used further for computation process is known as garbage output. The quantum cost of a reversible or quantum circuit is defined as the number of 1×1 or 2×2 gates used to implement the circuit.

A work on fused floating point FFT based on conventional logic has been reported which operates at a greater speed when compared with the discrete implementation [4]. A work on 24-bit reversible fused

Radix-2 FFT unit using 4:3 compressor has been implemented which consumes more number of logical resources as reported in [5]. The need for fused element is to increase the speed as well as to make use of the resources efficiently for signal processing applications. Thus, three reversible fused floating point elements are introduced in the proposed work namely reversible fused floating point add-sub (RFFAS), reversible fused floating point multiply-add (RFFMA) and reversible fused floating point multiply-sub (RFFMS). To realize the three reversible fused elements, the proposed design requires a reversible single precision floating point adder, a reversible single precision floating point subtractor and a reversible single precision floating point multiplier. Three works on reversible single precision floating point adder has been reported so far. A work on reversible single precision floating point adder has been reported in [6]. This work produces more number of garbage outputs and also increases the quantum cost and the number of constant inputs required. A space efficient reversible single precision floating point adder is proposed in [7]. The quantum cost of this work is less when compared with the previous work. Another work on reversible single precision floating point adder has been reported in [8]. This work has reduced the quantum cost, number of gates used and the number of constant inputs required with a slight increase in garbage output. Only one work has been reported so far on reversible single precision floating point subtractor with less number of reversible gates, constant inputs, quantum cost and garbage outputs [9]. Several works were reported on reversible multipliers. [10], [11], [12], [13], [14], [15], [16]. Thus, the aim of the work is to design a reversible fused 32-Point Radix-2 floating point FFT unit using 3:2 compressor with less number of reversible gates, quantum cost and garbage outputs.
The present section gives the introduction to reversible logic circuits and existing works reported so far on reversible floating point arithmetic operations. Section two discusses the proposed implementation of the fused reversible 32-point Radix-2 floating point FFT unit. Results are contained in section 3 and conclusions are contained in section 4.

2 MATERIALS AND METHODS

2.1 Design of the proposed Reversible 32-Point Radix-2 Floating Point FFT Unit as a Discrete and as a Fused Element

In the FFT algorithm, the complex floating point add, subtract, and multiply operations shown in figure 1 can be realized with a discrete implementation that uses three reversible single precision floating point adders to perform the complex add and four reversible single precision floating point multipliers and three reversible single precision floating point subtractors to perform the complex subtract.

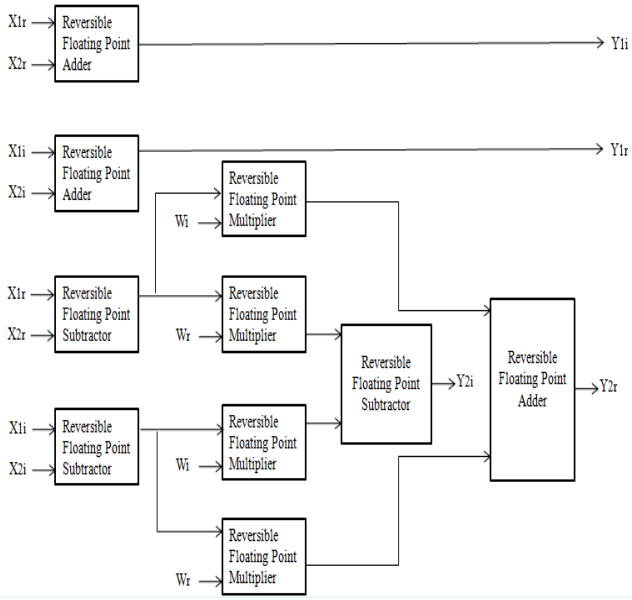


Figure 1. Discrete Implementation of the Reversible Radix-2 floating point FFT Unit

Alternatively, as shown in figure 2, the complex add and subtract can be performed with two reversible fused floating point add-subtract units (marked as RFFAS in figure 2) and the complex multiplication can be realized with two reversible fused floating point product units (marked as RFFMA and RFFMS).

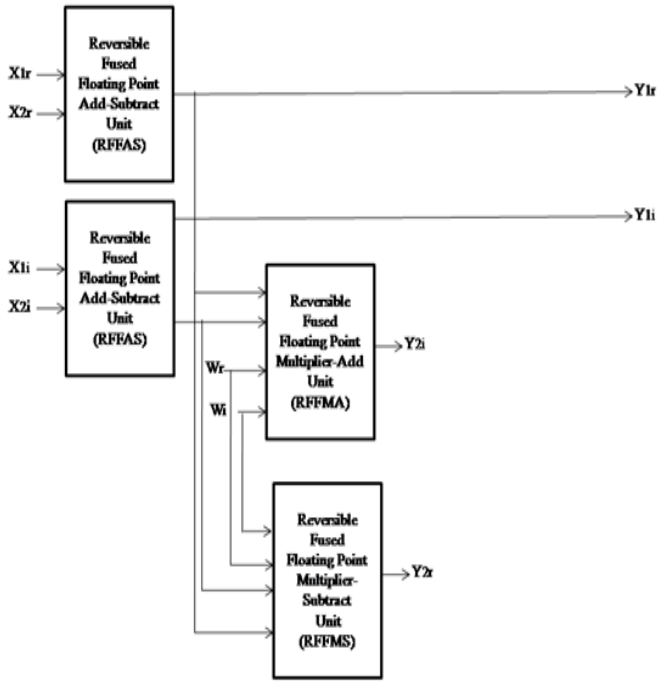


Figure 2. Fused Implementation of the Reversible Radix-2 floating point FFT Unit

In the above figure, Reversible fused add-subtract (RFFAS) represents the realization of a reversible single precision floating point adder and subtractor as a single unit. Reversible fused floating point multiply-add (RFFMA) unit consists of two reversible floating point multipliers and one reversible floating point adder. Reversible fused floating point multiply-subtract (RFFMS) unit consists of two reversible floating point multipliers and one reversible floating point subtractor.

2.2 Design of the Proposed Reversible Single Precision Floating Point Adder/ Subtractor

In IEEE 754 format, the single precision floating point number can be expressed as,

$$A = A_s A_e A_m \quad (1)$$

The floating point addition/subtraction of two numbers can be expressed as,

$$A + B = (A_m * 2^{Ae}) \pm (B_m * 2^{Be}) \quad (2)$$

The operations involved in floating point addition/subtraction are (a) aligning the binary point, (b) addition/subtraction of aligned significand and (c) normalization of the result. The detailed steps are as follows:

(1) Compare the significand of the operands.

(2) Align binary point:

Initial result exponent: the larger of A_e, B_e

Compute exponent difference: $B_e - A_e$

If $B_e >= A_e$ Right shift A_m that many positions to form $A_m 2^{Ae-Be}$

If $A_e > B_e$ Right shift B_m that many positions to form $B_m 2^{Be-Ae}$

(3) Add/Subtract the aligned significand: i.e. $A_m 2^{Ae-Be} \pm B_m$ or $A_m \pm B_m 2^{Be-Ae}$

(4) If normalization of result is needed, then a normalization step follows:

If ((A_e != B_e and cout /bout = 0) or if $A_e = B_e$), then left shift result, decrement result exponent (e.g., if result is 0.001xx…) else if A_e != B_e and cout / bout = 1,then Right shift result, increment result exponent (e.g., if result is 10.1xx…) Continue until MSB of data is 1.

(5) Check result exponent:

If larger than maximum exponent allowed return exponent overflow.

If smaller than minimum exponent allowed return exponent underflow.

From the steps involved in floating point addition/subtraction, the following observations are made:

i) To align the binary point, the initial exponent result should be larger of A_e, B_e. To realize this, an 8- bit reversible comparator is required to compare the two exponents.
ii) To determine $A_e - B_e$ or $B_e - A_e$, an eight bit reversible subtractor is required.
iii) To align the signifand, A_m or B_m needs to be shifted for which a 24-bit reversible shift register needed to be designed.
iv) To add/subtract the significand, a 24-bit reversible carry propagate generate adder and a 24-bit reversible subtractor is to be designed.
v) To determine the sign part, a 24-bit reversible comparator to compare the mantissa is required.
vi) For normalization of the result, a 32-bit reversible leading zero detector is to be designed.

Thus the floating point addition/subtraction unit requires the implementation of an 8-bit reversible comparator, an 8-bit reversible subtractor, a 24-bit reversible shift register, a 24-bit reversible carry propagate generate adder, a 24-bit reversible subtractor, a 24-bit reversible comparator and a 32- bit reversible leading zero detector. An efficient reversible single precision floating point adder is designed as proposed in [8]. Table 1 shows the number of gates required, garbage outputs produced, constant inputs required and the quantum cost of the reversible single precision floating point adder.

Table 1. Cost of the Reversible Single Precision Floating Point Adder

Module	No. of Reversible Gates Required	No. of Constant Inputs Used	No. of Garbage Outputs Produced	Quantum Cost
8-Bit Subtractor	15	8	15	46
24-Bit Comparator	118	118	163	590
8-Bit Comparator	34	34	44	170
24-Bit Shift Register	72	48	49	312
24-Bit Carry Look-Ahead Adder	48	24	48	432
32-Bit Leading Zero Detector	109	109	131	425
Exponent Adjustment Unit for Normalization & Rounding	475	240	415	1392
Total	**871**	**581**	**865**	**3367**

From the table, it is inferred that the total quantum cost of the proposed reversible single precision floating point adder is 3367. Table 2 shows the comparison of the proposed Reversible single precision floating point adder with the existing works.

Table 2. Comparison of the Proposed Reversible Single Precision Floating Point Adder with existing works

Reversible single precision floating point adder	No. of Gates Required	No. of Constant Inputs Used	No. of Garbage Outputs Produced	Quantum Cost
Nachtigal et al., [6]	Not Given	2646	2926	7458
Nguyen et.al., [7]	917	757	824	4873
Proposed Design	**871**	**581**	**865**	**3367**
Improvement wrt [7]	**5%**	**23%**	**-**	**31%**

From Table 2, it is inferred that the proposed reversible single precision floating point adder reduces the quantum cost with an improvement of 31% when compared to the existing work [7]. The number of constant inputs required in the proposed design also gets reduced leading to an improvement of 23% when compared to the existing work [7]. The number of reversible gates required in the design also gets reduced leading to an improvement of 5% when compared to the existing work [7]. Further, the proposed work has produced an addition of only 41 garbage outputs when compared with the existing work [7].

A reversible single precision floating point subtractor is designed as proposed in [9]. Table 3 shows the number of gates required, garbage outputs produced, constant inputs required and the quantum cost of the reversible single precision floating point subtractor.

Table 3. Cost of the Reversible Single Precision Floating Point Subtractor

Reversible Single Precision Floating Point Subtractor	No. of Gates Required	No. of Constant Inputs Used	No. of Garbage Outputs Produced	Quantum Cost
24-Bit Subtractor	47	24	47	142
8-Bit Subtractor	15	8	15	46
24-Bit Comparator	118	118	163	590
8-Bit Comparator	34	34	44	170
24- Bit Left Shift Register	72	48	49	312
32-Bit Leading Zero Detector	109	109	131	425
Exponent Adjustment Unit for Normalization	475	240	415	1392
Total	**870**	**581**	**864**	**3077**

2.3 Design of the Proposed Reversible Single Precision Floating Point Multiplier

To perform the multiplication of two numbers A and B, the following steps are conventionally followed.

i) Add the exponent of the two numbers and then subtract the result from the bias i.e., $E = A_e + B_e - 127$

ii) Multiply the Mantissa of the two numbers, $M = A_m * B_m$

iii) Calculate the sign of the result by XORing the sign of the two numbers using a reversible XOR, i.e., $S = A_s$ XOR B_s

iv) Normalize the result; i.e. obtaining 1 at the MSB of the Mantissa of the Product.

From the steps involved in floating point multiplication, the following observations are made:

- To add the two exponents, an 8-bit Reversible Propagate Generate Adder (RPGA) is required. To subtract the result from the bias value of 127, an 8-bit Reversible Bias Subtractor (RBS) is required.
- To multiply the two mantissas, a 24x24 bit reversible multiplier is to be designed.
- To calculate the sign bit, a reversible XOR gate is required.
- To normalize the result, a 47-bit reversible Shift Register is to be designed.

The block diagram representation of the proposed Reversible Single Precision Floating Point Multiplier derived from the observations made is shown in figure 3.

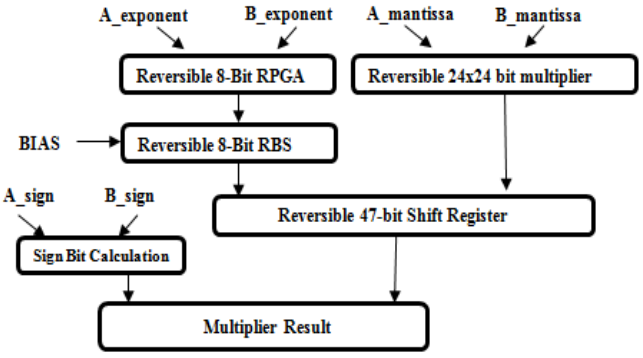


Figure 3. Block Diagram Representation of the Proposed Reversible Single Precision Floating Point Multiplier

The first two steps involved in the design of floating point multiplication are implemented as proposed in [19]. The designs of the reversible 24x24 bit unsigned multiplier and reversible normalizer are discussed in the following subsections.

2.3.1 Wallace Tree Multipliers and 4:3 Compressor Design

In high-speed designs, the Wallace tree method is a technique to add all the partial products in a column and produces two outputs (a sum and a carry). In conventional multipliers, compressors form the essential requirement of high speed multipliers. The speed, area and power consumption of the multipliers will be in direct proportion to the efficiency of the compressors. A higher order multiplication using higher order compressors is realized in [17]. The 4:2compressor is extremely used for high performance multipliers in order to lower the latency of the partial product accumulation stage. Owing to its regular interconnection, the 4:2 compressors is perfect for the construction of regularly structured Wallace tree with low complexity. A work on reversible single precision floating point multiplier using 4:2 compressors is reported in [18]. But the accuracy obtained with the 4:2 compressors was 80%. Thus, to achieve 100% accuracy, reversible 4:3 compressors are proposed in [19]. From the implementation results, it is inferred that the design proposed in [19] based on 4:3 compressor uses more number of reversible gates when compared to the existing work [18]. The number of garbage outputs produced is 159 in the proposed design and 129 in the case of the existing work [18]. Also the design based on 4:3 compressor has used more number of constant inputs and has an increased critical path delay when compared to the existing work [18]. Thus, the proposed design based on 4:3 compressor has not only increased the critical path delay when compared to the existing work but has drawbacks in terms of number of gates used, number of garbage outputs produced and number of constant inputs used. To overcome these drawbacks, a lower order compressor such as a 3:2 compressor is proposed in this work. A reversible 8x8 multiplier is realized using a reversible 3:2 compressor i.e. a Reversible Propagate Generate Adder was designed as shown in figure 4.

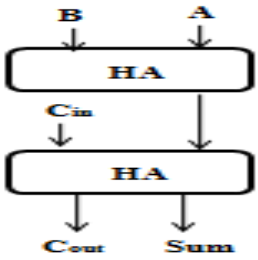


Figure 4. Realization of adder using 3:2 compressor

2.3.2 Design of a 24x24 Bit Multiplier using 3:2 Compressor

In the proposed design of the floating point multiplier, the 24-bit operands A and B are decomposed into three partitions of 8 bits each. Thus the 24x24 bit multiplier is performed through nine reversible 8x8 bit multipliers based on 4:3 as well as 3:2 compressors as shown in figure 5.

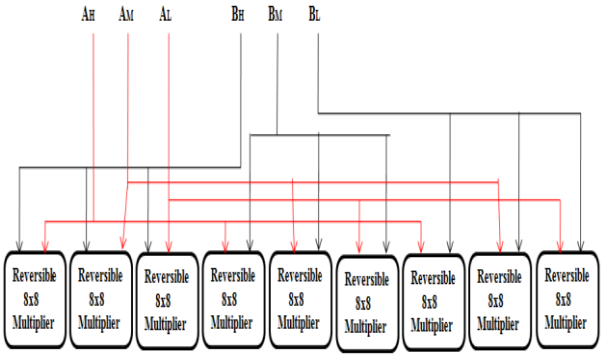


Figure 5. Realization of a 24x24 Bit multiplier using 8x8 Bit Multiplier

A comparison of the proposed reversible 24x24 bit multiplier with the existing work [18] is shown in Table 3.

Table 3. Comparison of the Proposed Reversible 24x24 bit Multiplier with the existing work

Realization of Reversible 24x24 bit multiplier	No. of Gates Required	No. of Constant Inputs Used	No. of Garbage Outputs Produced	Critical Path Delay
Nachtigal et al., [18] Using 4:2 Compressor	124	531	1387	677
Proposed work using 4:3 Compressor	127	873	321	628
Proposed work using 3:2 Compressor	**111**	**531**	**830**	**412**
Improvement wrt [18]	**11%**	**Same**	**40%**	**39%**

As can be understood from the table 3, the proposed design based on 3:2 compressor uses less number of reversible gates, produces less number of garbage outputs and has less critical path delay when compared to the existing design [18] based on 4:2 compressor. This design is also highly efficient in terms of number of reversible gates used and critical path delay when compared with the 4:3 compressor though the number of garbage outputs are more. Thus, the proposed reversible 24x24 bit multiplier based on 3:2 compressor offers an improvement of 11% in terms of gates required, 40% reduction in terms of garbage outputs produced and 39% reduction in critical path delay. However the number of constant inputs required remains the same. Hence, the proposed design based on 3:2 compressor significantly improves the speed of the circuit.

2.3.3 Realization of a 47-Bit Reversible Shift Register

Normalization is obtaining '1' at the MSB of the product. Since the hidden bit in mantissa is always '1' as per IEEE 754 standard, there is no chance of getting 48^{th} and 47^{th} bit of the 24x24 bit multiplier as '0'. Hence, there exist only three possible cases:

i) 48^{th} bit = 0 and 47^{th} bit = 1: If this is the case the product is already in the normalized form.
ii) 48^{th} bit = 1 and 47^{th} bit = 0: In this case, one time right shift is required.
iii) 48^{th} bit = 1 and 47^{th} bit = 1: For this case too, one time right shift is required.

With respect to case ii) and iii), to right shift the product once, a 47-bit reversible right shift register is required.

A 47-bit reversible serial-in serial-out shift register is designed using the proposed reversible D-Flip-Flop as in [8] with the newly proposed reversible gate AS. Table 4 shows the cost of the realization of 47-bit Reversible Serial In Serial Out Shift Register.

Table 4. Cost of the Proposed 47-bit Reversible Shift Register

Realization of 47-Bit Reversible Shift Register	No. of Gates Required	No. of Constant Inputs Used	No. of Garbage Outputs Produced	Quantum Cost
Proposed work	**141**	**141**	**95**	**611**

The realization of the 47-bit Reversible Shift Register requires 94 AS gates and 47 Feynman Gates thus, a total of 141 reversible gates. The total quantum cost of

the proposed design is 611. The proposed design uses 141 constant inputs and produces 95 garbage outputs.
Table 5 shows the total cost comparison of the proposed Reversible Single Precision Floating Point Multiplier based on 3:2 compressor with the existing design [18] based on 4:2 compressor.

Table 5. Cost comparison of the Proposed Reversible Single Precision Floating Point Multiplier

Reversible Single Precision Floating Point Multiplier	No. of Gates Required	No. of Garbage Outputs Produced	Quantum Cost
Existing work using 4:2 compressor [18]	547	1491	6957
Proposed work using 3:2 compressor	**398**	**949**	**4100**
Improvement wrt [18]	**27%**	**36%**	**41%**

On comparison, it is observed that the proposed design using 3:2 compressor is efficient in terms of utilizing the number of gates, number of garbage outputs produced and quantum cost. The proposed design uses 398 gates while the existing design uses 547 gates i.e. an improvement of 27%. The number of garbage outputs produced in the proposed design is 949 while the existing work produces 1491 and thus showing an improvement of 36%. The total quantum cost of the existing work is 6957 while the cost of the proposed design is 4100, thereby yielding an improvement of 41%.

3 DISCUSSION

A reversible 32-Point discrete and fused Radix-2 floating point FFT unit is designed as shown in figure 1 and figure 2. Table 6 shows the total cost of the proposed Fused and Discrete implementation of the Reversible 32-Point Radix-2 floating point FFT unit.

Table 6. Cost of the Proposed Reversible Fused and Discrete 32-Point Radix-2 Floating Point FFT unit using 3:2 compressor

Unit	No. of Reversible Gates Required		Quantum Cost		No. of Garbage Outputs Produced	
	Discrete	Fused	Discrete	Fused	Discrete	Fused
Add-Sub Unit	3482	2314	12888	9254	3458	2366
Multiply-Add Unit	1667	1383	11567	10345	2763	2573
Multiply - Sub Unit	1666	1382	11277	10055	2762	2573
Total	6815	**5079**	35662	**29654**	8983	**7511**

The results indicate that the cost of the three proposed fused reversible operations such as fused floating point add-sub, fused floating point multiply-add and fused floating point multiply-sub is less when compared to the discrete implementations. The total number of gates required is 5079. The number of garbage outputs produced is 7511 and the total quantum cost of the proposed design is 29654 which is less when compared to the discrete implementation thereby making the design a computationally speed, resource and power efficient.

3.1 Simulation Result of the Designed FFT unit

Figure 6 shows the simulation result of the proposed Reversible Single Precision Fused 32-Point Radix-2 Floating Point FFT unit using 3:2 compressor. In this figure, a1, a2, m1, m2 represents the 33 – bit input and cin, bin = 0. The signal "a_out1" represents the adder out, "a_out2" represents the subtract result, "m_out1", "m_out2", "m_out3" and "m_out4" represents the multiplier result. The signals "a_out3" represents the fused multiply_add result and "a_out4" represents the fused multiply_subtract result..
Thus for a1 = 010000001100000000000000000000000 (decimal = 4), a2 = 010000000100000000000000000000000 (decimal= 2), m1 = 010000001100000000000000000000000 (decimal = 4), m2 = 010000001100000000000000000000000 (decimal= 4), The adder result a_out1 = 010000001100000000000000000000000 (decimal = 6) represents the addition between the signals a1 and a2. The signal a_out2 represents the subtraction result

between the signals a1 and a2. Hence the result is 2 i.e. 01000000010000000000000000000000. The signal a_out3 indicates the result of the operation ((m1*a_out2) + (m2*a_out2)) i.e. 16 which in binary is 01000001110000000000000000000000. Finally s_out indicates the result of the operation ((m1*a_out2) – (m2*a_out2)) i.e. 0. Hence the result is 00000000000000000000000000000000.

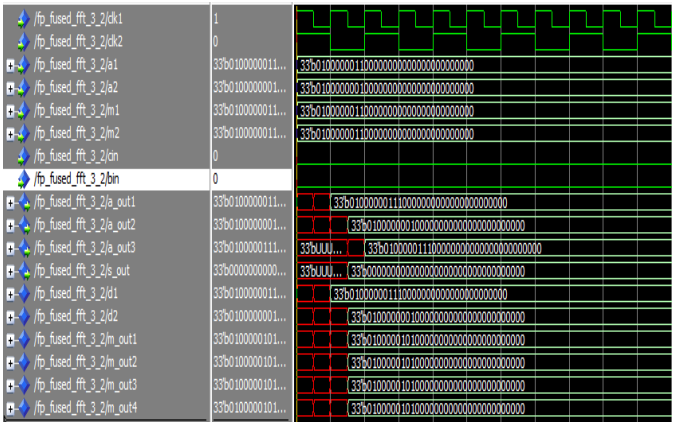


Figure 6. Simulation Result of the Proposed Reversible Fused 32-Point Radix-2 Floating Point FFT Unit using 3:2 Compressor

3.2 Synthesis Report of the Designed FFT unit

The entire design was synthesized using Xilinx Virtex5vlx30tff665-3. Table 7 shows the comparison of the discrete and fused implementation of the three operations.

Table 7. Comparison of the discrete and fused implementation of the three reversible operations RFFAS, RFFMA and RFFMS

	Reversible Floating Point Add-Subtract Unit		Reversible Floating Point Multiply-Add Unit		Reversible Floating Point Multiply-Subtract Unit	
	Discrete	Fused	Discrete	Fused	Discrete	Fused
Time taken	11.956 ns	**6.795 ns**	8.967 ns	**2.989 ns**	8.967 ns	**3.511 ns**

The results indicate that all the three reversible fused arithmetic units designed in FPGA are faster than the implementations constructed with discrete units. Table 8 shows the comparison of the discrete and fused implementation of a 32-Point Reversible Radix-2 Floating point FFT unit.

Table 8. Comparison of the discrete and fused implementation of a 32-Point Reversible Radix-2 Floating Point FFT unit

Reversible 32-point Radix-2 Floating Point FFT	Discrete FFT Unit	Fused FFT Unit
Time taken	24.166 ns	**22.037 ns**

Thus, the fused implementation of a Reversible 32-Point Radix-2 floating point FFT unit operates at a comparatively greater speed than the discrete implementation. Table 9 shows the device utilization summary of the proposed discrete and fused implementation of a 32-Point Reversible Fused Radix-2 Floating Point FFT unit.

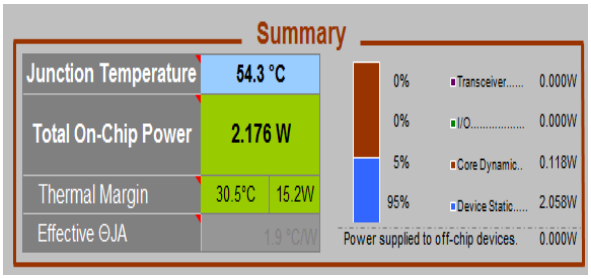
Table 9. Synthesis Report of the Proposed 32-Point Reversible Fused Radix-2 Floating Point FFT unit

Reversible 32-Point Radix-2 Floating Point FFT unit	No. of Slice LUTs	No. of Slice Registers	No. of Bonded IOBs
Discrete FFT Unit	15,784	930	333
Fused FFT Unit	**9726**	**479**	**333**
Improvement	**38%**	**48%**	-

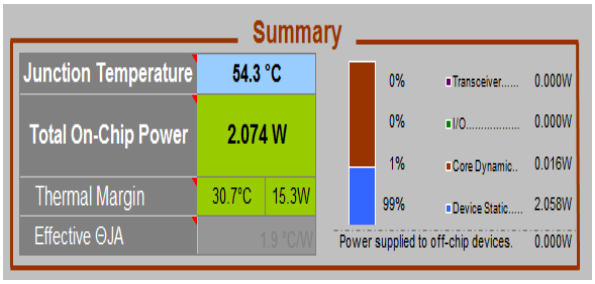
Thus, the proposed fused implementation of the Reversible 32-Point Radix-2 Floating Point FFT unit consumes less number of LUTs and slice registers than the discrete version of the 32-Point Reversible Radix-2 Floating Point FFT butterfly unit, i.e an improvement of 38% and 48% respectively is achieved.

3.3 Power consumption of the discrete and fused implementation of the reversible 32-Point Radix-2 floating point FFT unit

Figure 7 (a) and (b) shows the power consumption of the discrete and fused implementation of the Reversible 32-Point Radix-2 floating point FFT unit.



(a)



(b)

Figure 7. Power Consumption of the Reversible 32-Point Radix-2 Floating Point FFT unit (a) Discrete Unit (b) Fused Unit

The discrete implementation consumes a power of 2.176W while the proposed fused implementation of the reversible 32-Point Radix-2 floating point FFT unit consumes a power of 2.074W. Thus, the proposed fused reversible 32-Point Radix-2 Floating Point FFT unit dissipates slightly less power when compared with the discrete implementation and is also efficient in terms of number of reversible gates used, quantum cost, garbage outputs and speed.

4 CONCLUSION

The present work describes the design of three reversible fused floating point arithmetic units and their application to the implementation of 32-Point Radix-2 floating point FFT operation. Although the reversible fused floating point add-subtract unit is specific to FFT applications, the reversible fused floating point multiply-add unit and the reversible fused floating point multiply-sub unit are applicable to a wide variety of digital signal processing algorithms. The proposed reversible 24x24 bit multiplier using 4:3 compressors requires more number of gates as well as increases the critical path delay. Thus, the proposed reversible fused 32-Point Radix-2 floating point FFT unit is realized using 3:2 compressors with less number of resources, operates at a comparatively greater speed and also dissipates less power when compared with the discrete implementation. Thus, higher order compressors are not suitable for reversible implementation.

REFERENCES

1. Landauer, R.: Irreversibility and heat generation in the computational process. IBM J. Research and Development. 5, 183-191, 1961.
2. Bennett, C. H.: Logical reversibility of computation. IBM J. Research and Development. 17, 525- 532 1973.
3. Perkowski, M., Kerntopf, P.: Reversible Logic. In: Proc. 2001 Euro-Micro Workshop, Poland, 2001.
4. Earl Swartzlander. E., Hani Saleh, H. M.: FFT Implementation with Fused Floating-Point Operations. IEEE Transactions on Computers, 61, 2, 284 – 288, 2012.
5. AnanthaLakshmi, A. V., Sudha, G. F.: Area and Speed Efficient Reversible Fused Radix- 2 FFT Unit Using 4:3 Compressor. Int. J. on Recent Trends in Engineering and Technology. 10, 172 – 186, 2014.
6. Nachtigal, M., Thapliyal, H., Ranganathan, N.: Design of a reversible floating-point adder architecture. In: Proc. 2011 IEEE International Conference on Nanotechnology, Portland Marriott, Portland, Oregon, USA, August 2011.
7. Nguyen, T. D., Meter, R. V.: A Space-Efficient Design for a Reversible Floating Point Adder in Quantum Computing. Journal of Quantum Physics, arXiv:1306.3760 [quant-ph], 2013.
8. AnanthaLakshmi, A.V., Sudha, G. F.: Design of an efficient reversible single precision floating point adder. Int. J. of Computational Intelligence Studies. Article in Press.
9. AnanthaLakshmi, A.V., Sudha, G. F.: Design of a reversible single precision floating point subtractor. SpringerPlus. 3:11, doi: 10.1186/2193-1801-3-11, 2014.
10. Thapliyal, H., Srinivas, M. B.: Novel reversible TSG gate and its application for designing components of primitive/reversible quantum ALU. In: Proc. 2005 IEEE Intl. Conf. on Information, Communications and Signal Processing. 1425-1429, 2005.
11. Thapliyal, H., Srinivas, M. B.: Novel Reversible multiplier architecture using Reversible TSG gate. In: Proc. 2006 IEEE Intl. Conference on Computer Systems and Applications. 100-103, 2006.
12. Haghparast, M., Jassbi, S. J., Navi, K., Hashemipour, O.: Design of a novel reversible multiplier circuit using HNG gate in nanotechnology. World Applied Science Journal. 3, 974- 978, 2008.
13. Shams, M., Haghparast, M., Navi, K.: Novel reversible multiplier circuit in nanotechnology. World Applied Science Journal. 3, 806–810, 2008.
14. Banerjee, A., Pathak, A.: An analysis of reversible multiplier circuits. 1-10, ArXiv:0907.3357 2009.

15. Islam, M. S., Rahman, M. M., Begum, Z., Hafiz, M. Z.: Low cost quantum realization of reversible multiplier circuit. Information Technology Journal. 8, 208, 2009.
16. Bhagyalakshmi, H. R., Venkatesha, M. K.: An improved design of a multiplier using reversible logic gates. Intl. J. Engineering Science and Technology. 2, 3838-3845, 2010.
17. Menon, R., Radhakrishnan, D.: High Performance 5:2 Compressor Architectures. In: Proc. 2006 IEEE Intl Conference on Circuits, Devices and Systems. 447—452, 2006.
18 Nachtigal, M., Thapliyal, H., Ranganathan, N.: Design of a reversible single precision floating point multiplier based on operand decomposition. In: Proc. 2010 IIEEE Intl Conference on Nanotechnology Joint Symposium with Nano, Korea. 233-237, 2010.
19. AnanthaLakshmi, A. V., Sudha, G. F.: An Efficient Implementation of a Reversible Single Precision Floating Point Multiplier Using 4:3 Compressor. In: Proc. 2013 Elsevier International Conference on Advances in Information Technology and Mobile Communications(AIM), 229 – 238, 2013.