

EFFICIENT IMPLEMENTATION OF BIT SLICE BUTTERFLY FOR FFT PROCESSORS USING PARALLEL PREFIX ADDER

CHITYALA MAHESH¹, DR. AVALA MALLIKARJUNA PRASAD²

¹Student, M.Tech, JNTUK, Andhra Pradesh

²Professor, JNTUK, Andhra Pradesh

Abstract - FFT is most generally utilized in DSP. The quantity of applications for FFTs continues to grow and is incorporated in numerous regions such as: communications, signal process, instrumentation, medical specialty engineering, sonic, acoustics, numerical ways, and applied mechanics. High base algorithms area unit developed for economical computation of FFT however they increase the quantity of operations and quality of butterfly. The key advantage of victimization SRFFT approach is that it helps to attain low variety of arithmetic operations. The no of complicated multiplications concerned is twelve and sophisticated addition is sixteen. Unlike the antecedently developed bit-serial RSFQ BPUs, the planned bit-slice BPUs ceaselessly method 64-point information divided into eight slices of eight points every so as to scale back the hardware price. The BPUs use synchronous concurrent-flow duration and that they will be extended to any n2-point process. During this project SRFFT is meant victimization FFT design techniques for 16-point input sequence. The SRFFT architectures will be designed victimization each radiotelegraphic signal and DIF decompositions. During this project DIF decomposition technique is employed. The 8-point split-radix DIF FFT formula is enforced by divide and conquers approach in DIF domain victimization Xilinx software system. Synthesis and logic simulation is displayed in Xilinx style suit14.7. The results show the development within the speed.

Keywords: RSQF, Split, Radix, Fast Fourier Transform, Butter Fly, Verilog.

1. INTRODUCTION

The fast Fourier transform (FFT) algorithm is one of the most important algorithms in digital signal processing. Cooley and Tukey [1] first proposed the FFT algorithm. There have been lots of classes of FFT algorithms such as the radix-2^m algorithms, the Winograd algorithm [2], the prime factor algorithm [3], and so on. Among these FFT algorithms, the radix-2, radix-4, and split-radix [4] algorithms have been used most for practical applications due to their simple architecture.

In current practical applications, the FFT is calculated at very high throughout rates, even in the range of gig samples per second, so it is difficult to implement an energy-efficient FFT processor in CMOS circuits. Therefore, rapid single-flux-quantum (RSFQ) circuits have

the potential to overcome the problem for its high speed and low-power consumption [5].

Some RSFQ circuits for processors have been demonstrated [6]–[8] and designed [9]–[11].

A butterfly processing unit (BPU) is the main component in FFT processors. Several radix-2 BPUs using SFQ circuits have been demonstrated, including a 5-bit RSFQ BPU [12] 4-bit integer-type SFQ BPU [13], and a 4-bit fixed-point-type SFQ BPU [14]. In the past, researchers focused on bit-serial processing for RSFQ BPU. It takes a longer calculation time to process 64-point data in bit-serial processing. In bit-parallel processing, the hardware cost is large in order to calculate 64-point data by synchronous method.

In this article, we proposed bit-slice BPUs based on radix-2, mixed-radix, and split-radix algorithms. Bit-slice processing begins to attract researchers because it has faster speed than bit-serial processing and lower hardware cost than bit-parallel processing. To the best of our knowledge, this is the first proposal of RSFQ bit-slice BPUs.

The rest of this article is organized as follows. In Section II, the algorithm, architecture, and RSFQ logic design details of the proposed bit-slice BPUs are described. Section III compares the BPUs. Finally, Section concludes this article.

2. 8-BIT BIT-SLICE BPUS

2.1. Algorithmic program and design

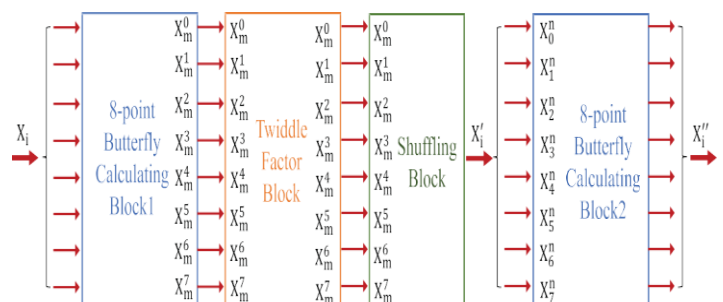


Fig-1: Block diagram of the BPUs.

There are 2 styles of FFT algorithms: decimation in time (DIT)

And decimation in frequency (DIF). DIT is multiplying twiddle factors 1st and bit-reversed for its input. DIF is doing butterfly operation 1st and bit-reversed for its output. In this article, the dot algorithmic program is adopted to implement the projected BPUs. Unlike the antecedently developed bit-serial RSFQ BPUs, the proposed bit-slice BPUs ceaselessly method bit-sliced 64-point knowledge divided into eight slices of eight purposes every so as to increase the outturn of processors and scale back the hardware cost. The computer file length is four bits, that is split into 2 slices of two bits every to be processed. Fig. one shows the diagram of the BPUs. It's composed of 2 butterfly shrewd blocks, a twiddle issue block, and a shuffling block. The block diagrams of BPUs supported radix-2, mixed-radix, and split-radix algorithms are an equivalent. In order to confirm the total capability of the BPUs, two same butterfly shrewd blocks are used. Every butterfly shrewd block contains totally different numbers of butterfly cells.

The number of butterfly calculating cells depends on which algorithm is adopted. The details of the three types of butterfly calculating cells will be shown later.

The 64-point FFT is carried out through 12 steps. At steps 0-7, the 64-point data are divided into eight slices of eight points each.

The i th slice, i.e., X_i ($0 \leq i \leq 7$), is input one by one into the eight-point butterfly calculating block1 at step i . The eight points of i th slice include $X(i), X(i + 8), X(i + 16), X(i + 24), X(i + 32), X(i + 40), X(i + 48), X(i + 56)$, where $0 \leq i \leq 7$. The input order of the eight points is determined by the algorithm used in butterfly calculating cells. According to radix-2 algorithm.

2.2. BPU Based on Radix-2

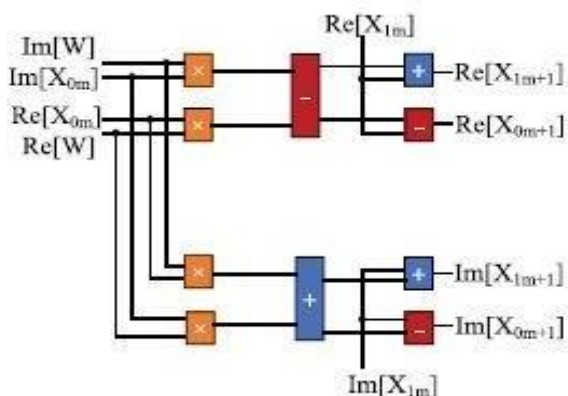


Fig -2: Block diagram of Radix-2.

Fig. 2 shows the block diagram of radix-2 butterfly

calculating cell, in which two points and a twiddle factor are input. The butterfly calculation based on radix-2 algorithm contains one complex multiplication, which consists of four real multiplications. The butterfly calculating cell is implemented by four multipliers, three adders, and three subtractors. After the multiplication operation, we take the most four significant bit (the same in the following). An eight-point butterfly calculating block based on radix-2 algorithm includes 12 radix-2 butterfly

calculating cells.



Fig-3: Block diagram of radix-4 butterfly calculating

2.3. BPU Based on Split-Radix

Split-radix algorithm combines the radix-2 and radix-4 algorithms. A radix-4 algorithm is for the odd terms of the FFT, and radix-2 algorithm is for the even terms of the FFT.

Fig. 3 shows the block diagram of radix-4 butterfly calculating cell, in which four points and three twiddle factors are input. The butterfly calculation based on radix-4 algorithm contain.

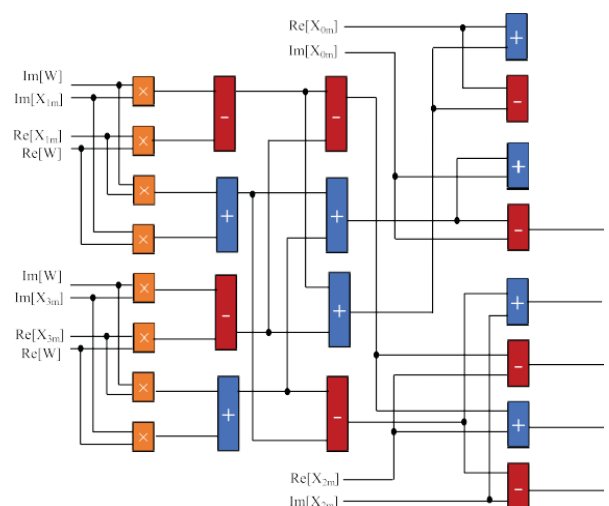


Fig -4: Block diagram of split-radix butterfly calculating cell.

Three complex multiplications. The butterfly calculating cell is implemented by 12 multipliers, 11 adders, and 11 subtractors.

Fig. 4 shows the block diagram of split-radix butterfly calculating cell, in which four points and two twiddle factors are input. The butterfly calculation based on split-radix algorithm contains two complex multiplications. The butterfly calculating cell is implemented by eight multipliers, six adders, and six subtractors.

By exploiting the radix-2, radix-4, and split-radix butterfly calculating cells, we have designed the BPU based on split-radix

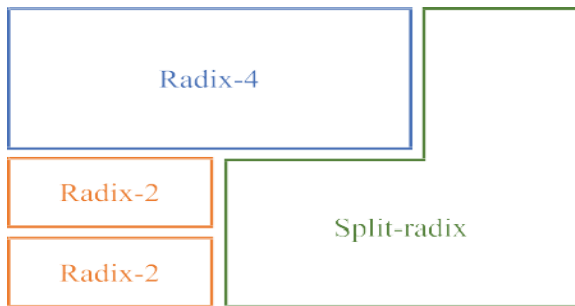


Fig -5: Block diagram of the eight-point butterfly calculation based on split

2.4. BPU Based on Mixed-Radix

As shown in Fig. 5, split-radix algorithm uses different radices in the same stages. Mixed-radix algorithm uses different radices in different stages. Fig. 6 shows the block diagram of the eight-point butterfly calculation based on mixed-radix algorithm. An eight-point butterfly calculating block based on mixed-radix algorithm includes two radix-4 butterfly calculating cells and four radix-2 butterfly calculating cells.

2.5. Logic Design

Concurrent-flow clocking is adopted to design fully pipelined synchronous RSFQ logic circuits of the proposed BPUs. DFFs are used to ensure the synchronization of the data in the BPUs. The basic RSFQ logic gates, wirings and flip-flops: AND, XOR, NOT, DFF, D2FF, NDRO, SPL (splitter), and confluence buffer (CB) in the cell library [15] for the AIST ADP2 with a critical current density of 10 kA/cm² [16] are used. The delay and junction count of each logic gate.

The butterfly calculating blocks contain different numbers of butterfly calculating cells. The butterfly calculating cells are implemented by adders, subtractors, and

multipliers. Fig. 6 the logic gate level circuit of the adder used in butterfly.

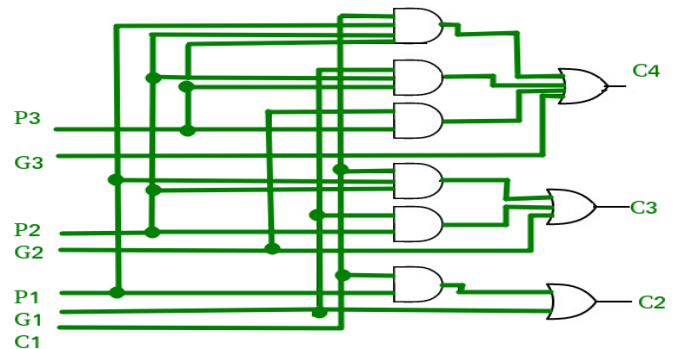


Fig -6: Logic design of the adder used in butterfly calculating cell.

A	B	C	C + 1	Condition
0	0	0	0	No Carry Generate
0	0	1	0	
0	1	0	0	
0	1	1	1	No Carry Propagate
1	0	0	0	
1	0	1	1	
1	1	0	1	Carry Generate
1	1	1	1	

Table -1: Truth Table for Carry Look Ahead Adder

Consider the complete adder circuit shown higher than with corresponding truth table. we tend to outline 2 variables as 'carry generate' and 'carry propagate' then,

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

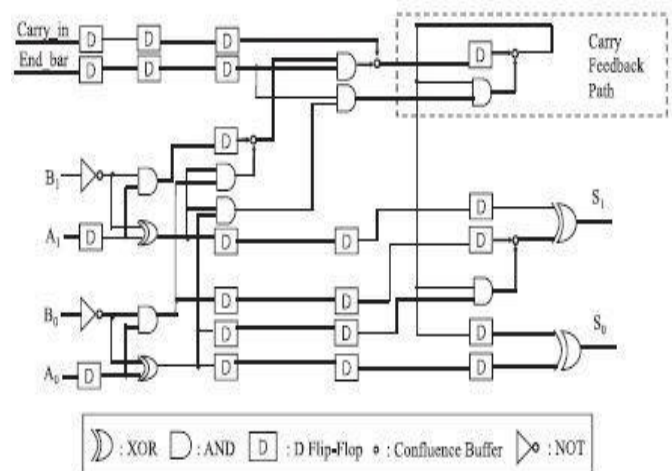


Fig -7: Logic design of the subtractor used in butterfly calculating cell

The BPUs based on mixed-radix and split-radix algorithms consist of 173 stages in total. The slices of input data are fed at the 1st to the 16th clock cycles, and the slices of the result are output at the 174th to the 189th clock cycles. The calculation for 64-point BPUs based on mixed-radix and split-radix algorithms needs 189 clock cycles.

3. DISCUSSION

We have designed RSFQ circuits of the bit-slice BPUs based

On radix-2, mixed-radix, and split-radix algorithms. The number of multipliers, adders, and subtractors used in the BPUs.

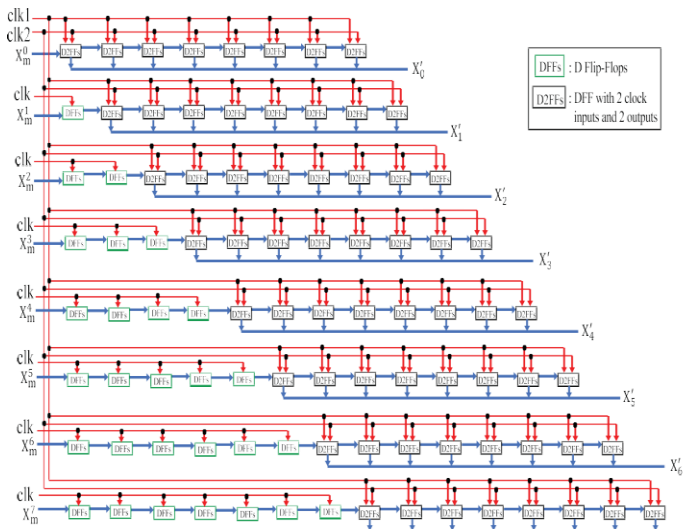


Fig -8: Logic design of shuffling block.

10 GHz which is assumed to calculate the latency of the proposed FFT BPUs. The calculation of the semiconductor FFT processor to complete 64-point 4-bit FFT needs 264 cycles at 600 MHz [18], whereas the calculation of the RSFQ FFT processor based on the proposed FFT BPU using split-radix algorithm needs 189 cycles at 10 GHz. It is obvious that the performance of the RSFQ FFT processor based on the proposed FFT BPU is better than the CMOS FFT processor.

4. RESULTS

4.1. AREA

Logic Utilization	Used	Available	Utilization
Number of Slices	4	5472	0%
Number of 4 input LUTs	8	10944	0%
Number of bonded IOBs	88	320	27%

Table -2

4.2. DELAY

Timing Summary:

Speed Grade: -10

Minimum period: No path found
 Minimum input arrival time before clock: No path found
 Maximum output required time after clock: No path found
 Maximum combinational path delay: 6.165ns



Fig -9: Output Waveform

5. CONCLUSION

We have designed RSFQ logic gate level circuits of the bit-slice BPUs based on radix-2, mixed-radix, and split-radix algorithms. The results show that the split-radix BPU has the lowest hardware cost and latency at 10 GHz. It consists of 86 multipliers, 66 adders, and 66 subtractors. It contains 173 pipeline stages, a total of 147 778 JJs, and a latency of 20 292.4 ps. According to the physical schematic designs we have designed, the hardware cost of physical schematic designs is three times that of logical designs. Based on the estimation of the interconnection, the maximum clock frequency is 75 GHz.

REFERENCES

- [1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, pp. 297–301, Apr. 1965.
- [2] S. Winograd, "On computing the DFT," *Math. Comput.*, vol. 32, no. 1, pp. 75–199, Jan. 1978.
- [3] C. S. Burrus and P. W. Eschenbacher, "An in-place, in order prime factor FFT algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP- 29, no. 4, pp. 806–817, Aug. 1981.
- [4] P. Duhamel and H. Hollmann, "Split-radix FFT algorithm," *Electron. Lett.*, vol. 20, no. 1, pp. 14–16, Jan. 1984.
- [5] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for subterahertz-clock-frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991.
- [6] Y. Yamanashi et al., "Design and implementation of a pipelined bit-serial SFQ microprocessor, CORE1 β ," *IEEE Trans. Appl. Supercond.*, vol. 17, no. 2, pp. 474–477, Jun. 2007.
- [7] Y. Ando, R. Sato, M. Tanaka, K. Takagi, N. Takagi, and A. Fujimaki, "Design and demonstration of an 8-bit bit-serial RSFQ microprocessor: CORE e4," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 5, Aug. 2016, Art. no. 1301205.
- [8] G. Tang, K. Takata, M. Tanaka, A. Fujimaki, K. Takagi, and N. Takagi, "4-bit bit-slice arithmetic logic unit for 32-bit RSFQ microprocessors," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 1, Jan. 2016, Art. no. 1300106.
- [9] G. Tang, K. Takagi, and N. Takagi, "32 32-bit 4-bit bit-slice integer multiplier for RSFQ microprocessors," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 3, Apr. 2016, Art. no. 1301005.
- [10] G. Tang, P. Qu, X. Ye, and D. Fan, "Logic design of a 16-bit bit-slice arithmetic logic unit for 32-/64-bit RSFQ microprocessors," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 4, Jun. 2018, Art. no. 1300305.
- [11] Tang, P. Qu, X. Ye, D. Fan, and N. Sun, "32-bit 4 4 bit-slice RSFQ matrix multiplier," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 7, Oct. 2018, Art. no. 1300905.
- [12] A. Mukhanov and A. F. Kirichenko, "Implementation of a FFT radix 2 butterfly using serial RSFQ multiplier-adders," *IEEE Trans. Appl. Supercond.*, vol. 5, no. 2, pp. 2461–2464, Jun. 1995.