A 2048-point Split-Radix Fast Fourier Transform Computed using Radix-4 Butterfly Units

Sonali D. Patil, Manish Sharma

Abstract*: *For the low-power consumption of fast fourier transform, Split-radix fast Fourier transforms are widely used. SRFFT uses less number of mathematical calculations amongst the different FFT algorithms. Split-radix FFT has the same signal flow graph that of conventional radix-2/4 FFT's. Therefore, the address generation method is same for SRFFT as of radix-2. A low power SRFFT architecture with modified butterfly units is presented over here. Here, it is shown that the, a 2048-point SRFFT is computed using radix-4 butterfly unist. Dynamic power is saved, on compromising the use of extra hardware. Here, the architecture size is increased from radix-2 to 4 and the dynamic power consumption is evaluated.

Keywords : ***Butterfly unit, low-power, Split-radix FFT, shared-memory.***

INTRODUCTION

In digital applications, fast fourier transform (FFT) plays very essential and fundamental role. There are various types of FFT"s. In 1984, Duhamel and Hollmann [1] invented a different type of FFT algorithm termed as split-radix FFT (SRFFT). SRFFT requires very less multiplications and additions amongst the known FFT algorithm. Mathematical operations majorly contribute to overall system power consumption therefore, Split-radix is a good option for the implementation of a low-power FFT processor.

FFT processors are of two types: pipelined processors and shared-memory [2] and [3] processors. Pipelines architectures produce high throughputs, but needs more hardware resources. Opposite to this, shared-memory architecture requires very less hardware resources, but, produces slow throughputs. FFT data in conventional FFT shared- memory architecture is arranged into two separate memory storage banks. For every clock cycle, two of the FFT data are given back by RAM memory and one butterfly unit is used to process/calculate the data. For the very next clock cycle, the output is written back to the memory banks and the previous data is replaced. This process limits only for the shared-memory architecture. A memory-shared architecture requires an effective address generation technique for FFT data and twiddle factors. An L-shaped butterfly data-path is involved in this SRFFT. Here, it is shown that SRFFT is evaluated using a radix-4 butterfly structure.

The remaining paper is grouped as following. In section I, FFT basic equations are described. Section II discusses the present system. Section III describes the results and simulation output and Section IV represents conclusion.

Revised Manuscript Received on July 22, 2019.

* Sonali D. Patil, studying at Dr. D. Y. Patil College of Engineering, Akurdi, Pune-44 (sonalidipakpatil@gmail.com). Manish Sharma, currently working as Associate Professor at D. Y. Patil College of Engineering, Akurdi, Pune-44 (manishsharma.mitm@gmail.com).

I. BASIC FFT EQUATIONS

For the N-point conventional radix-2 discrete fast fourier transform,

$$X(k) = \sum_{n-0}^{N-1} x(n) W_N^{nk} \quad (1)$$

Where k = 0, 1,……, N-1 and $W_N^{nk} = e^{-2\pi nk/N}$. If we separate the even terms and odd terms, radix-2 is obtained as

$$X(2k)=\sum_{n=0}^{\frac{N}{2}-1}[x(n) + x(n+N/2)]W_{N/2}^{nk} \quad (2)$$

$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1}[x(n) - x(n+N/2)]W_N^n W_{N/2}^{nk} \quad (3)$$

SRFFT uses radix-2 index map for the even terms and a radix-4 map to the odd terms. For the even terms it can be obtained as (2). For the odd one"s, it can be obtained as

$$X(4k+1) = \sum_{n=0}^{\frac{N}{4}-1}[x(n) - x\left(n+\frac{N}{2}\right) - j(x\left(n+\frac{N}{4}\right) - x(n+3N/4))]W_N^n W_{N/4}^{nk} \quad (4)$$

$$X(4k+3) = \sum_{n=0}^{\frac{N}{4}-1}[x(n) - x\left(n+\frac{N}{2}\right) + j(x\left(n+\frac{N}{4}\right) - x(n+3N/4))]W_N^n W_{N/4}^{nk} \quad (5)$$

Where k = 0, 1, ……., N/4. Equations 4 and 5 produce L-shaped split-radix butterfly structure. If $N = 2^S$ point FFT, SRFFT and radix-2 requires S number of passes to finish the computation, as seen in fig 1 and fig 2. The total number of L-shaped butterflies for SRFFT N_{SR} is given by [2]

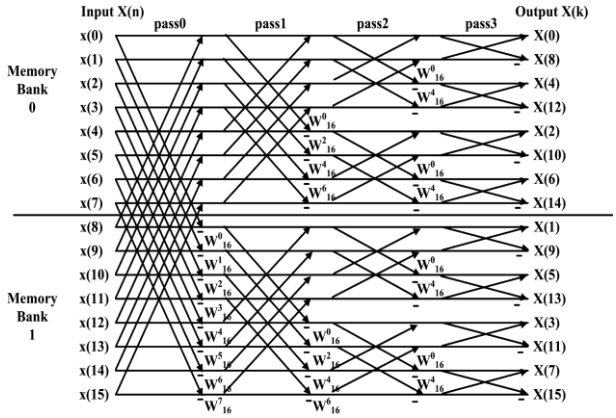

Fig. 1. Radix-2 signal Flow Graph

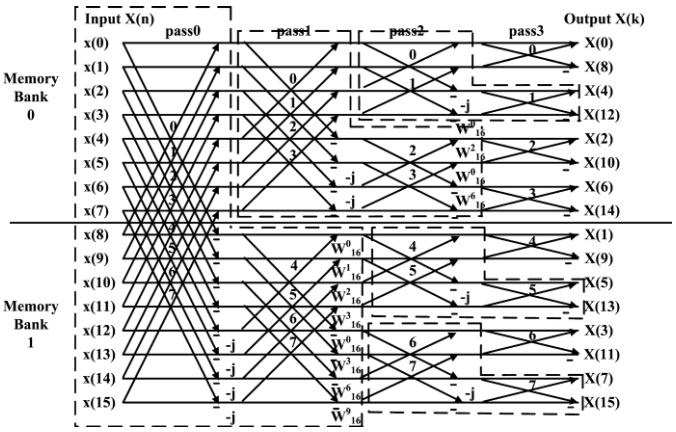

Fig. 2. SRFFT signal flow graph

$$N_{SR} = [(3S-2)2^{S-1} + (-1)^S]/9 \quad (6)$$

L shaped butterfly has two nontrivial complex multiplications, and so, the total number of nontrivial complex multiplications M_{SR} in SRFFT is

$$M_{SR} = [(3S-2)2^{S-1} + (-1)^S]2/9. \quad (7)$$

Number of SR butterfly N_{S-1} in the $(S-1)^{th}$ pass is,

$$N_{S-1} = [2 + (-1/2)^{S-2}]N/12. \quad (8)$$

Every L shape butterfly does not have any nontrivial twiddle factors, and so, the number of nontrivial multiplications M'_{SR} in SRFFT is,

$$M'_{SR} = M_{SR} - 2N_{S-1} \quad (9)$$

The complex multiplications M_{R2} in the conventional FFT algorithm is,

$$M_{R2} = 2^{S-1}(S-1). \quad (10)$$

A. Memory-Shared Architecture

Fig 3 shows a shared-memory processor. The RAM and ROM banks, stores the FFT data and twiddle factors respectively. The flow graph of SRFFT is similar to that of conventional FFT therefore; the data addressing methods could also be applied to SRFFT.

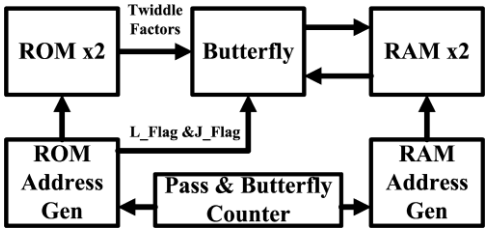

Fig. 3. Shared-memory architecture

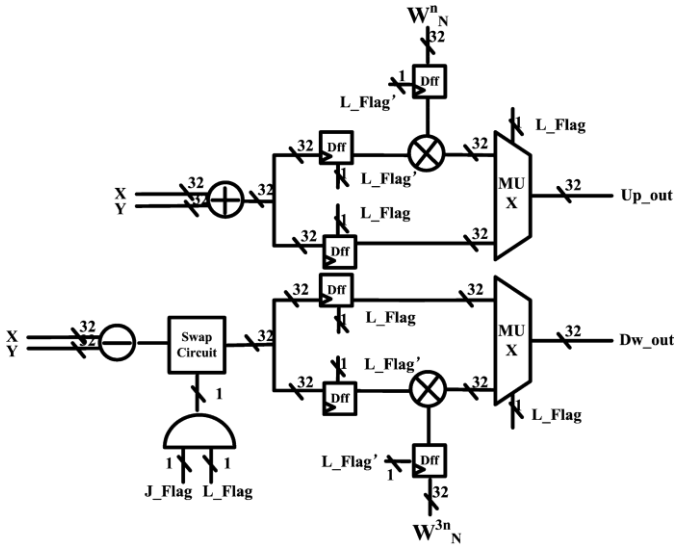

Fig. 4. Modified butterfly structure.

B. Modified Radix-4 butterfly Unit

Fig 4 shows a butterfly unit. For avoiding the switching activity, registers are applied in the multiplier path and some registers are placed at the address port of memory banks for the synchronization of the full design. The idea behind the modification is to know about which butterflies has no multiplications, trivial multiplications those are swapped one's, and non-trivial multiplications.

C. Generation of Address for coefficients

Fig 2 shows the flow graph (SFG) for a 16-point SRFTT. There are two coefficients j and W_n. Operations involving j are called trivial multiplications and those involving W_n are said to nontrivial. Area surrounded by dashed line represents one L block [8], and there are five L-blocks for a 16-point SRFFT.

II. PRESENT SYSTEM

Here, radix-2 is increased to radix-4, and the point size 1024 is increased to 2048 and the performance of area, power and delay is carried out. Radix-4 burst input/output engine to process butterfly unit [6]. Data is separately loaded and unloaded from the computation. Data in/out and processing are not one after the other. The data-frame gets loaded after FFT is started. After the loading of a complete frame, the processor computes the transform. When the calculations are over, unloading of the data is carried out,

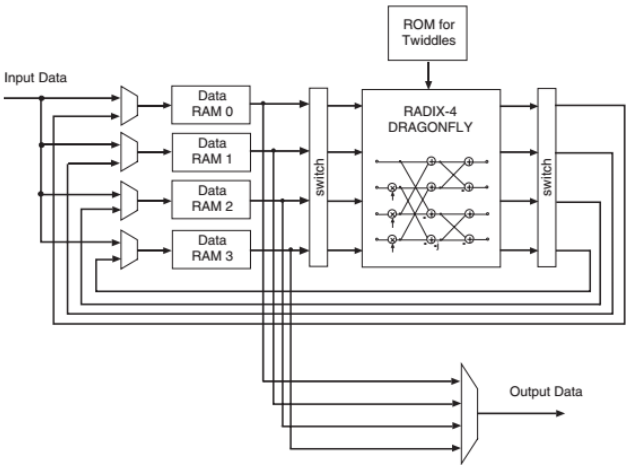

Fig. 5. Proposed radix-4 SRFFT system

but data loading or unloading during the calculation process is not allowed. Overlapping of data appears if the data is unloaded in digit reversed order. Data and phase factors can be stored in RAM.

A. SRFFT Block Diagram

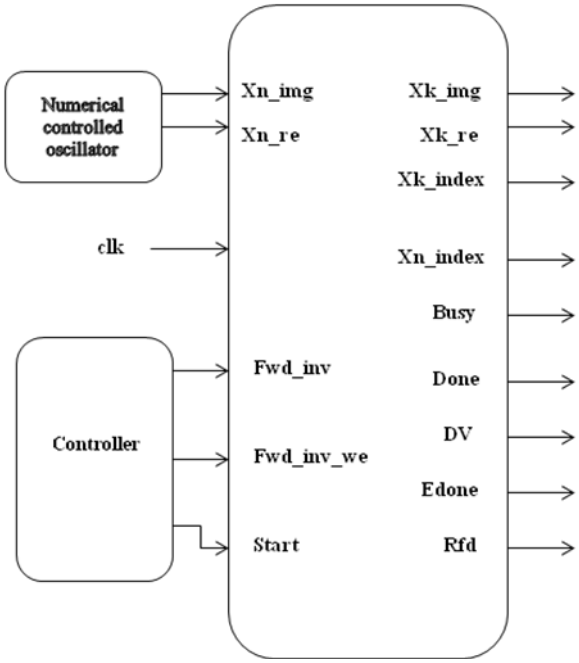

Fig. 6. SRFFT Block Diagram

From fig 6 it can be seen that the input is given from the NCO to FFT Block of X_n. NCO generates the sine signal.

B. Numerical Control Oscillator

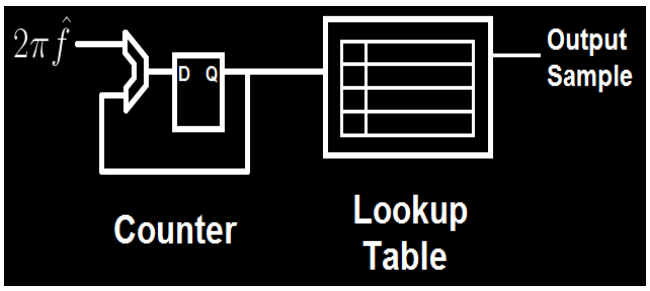

Fig. 7. Basic Numerical Control Oscillator

An NCO has a lookup table with waveform data usually a sinusoidal and a counter. Change in the counter describes the frequency of the output wave, in normalized units. The counter is coined as a 'phase accumulator,' because it stores the current value of the sine's phase. In this, an NCO tracks the argument to sin(2πfn) in a counter and uses a look up table to calculate the value of sin(2πfn).

C. Radix-4 Butterfly Calculations

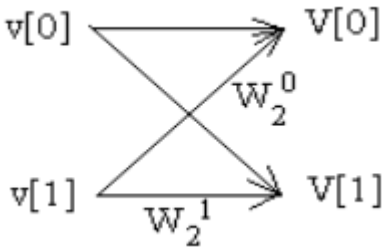

Figure 8: Butterfly Calculation

To save computations the FFT divides every fourth output sample into smaller length DFTs. The radix-2 and radix-4 FFTs needs only 75% complex multiplies. From Figure 8 it is seen that 16-point FFT requires two operations, while 256-points FFT requires four operations.

III. RESULT AND DISCUSSION

The design is compared with the previous work [9]. Synthesis of FFT"s is done using Xilinx. The behavioral simulation is done using ModelSim software. Power is measured by X-power analyzer of Xilinx. For a 2048-point FFT the proposed algorithm achieves almost same power that of existing system. The dynamic power consumption reduction is because the ROM banks and multipliers are active only when needed. The SRFFT has the irregular signal flow graph. The proposed architecture provides more efficiency, less delay and burst mode data transfer.

Table-I: Comparison of radix-2 and radix-4 SRFFT's

SRFFT	FF	LUT	BRAM	POWER (MW)
RADIX-2 1024 POINT	166	468	3	20.01
RADIX-4 2048 POINT	3091	2322	3	20

IV. CONCLUSION

Here, a shared-memory based SRFFT processor is presented using a modified butterfly units. The used methodology achieves almost same dynamic power consumption. SRFFT has minimal number of multiplications compared to other types. Here, burst-mode data transfer is used for faster computations. The synthesis is done using Xilinx and behavioral simulation is held using ModelSim Altra tool. Computations are also faster. It is seen that radix-4 dynamic power consumption is almost equal to radix-2.

REFERENCES

1. Pierre Duhamel and Henk Hollmann, "Split-radix fast fourier transform algorithm", Electron. Lett., vol. 20, no. 1, pp. 14–16, Jan. 1984.
2. Mark Richards, "On hardware implementation of the split-radix fast fourier transform," IEEE Transaction. Signal Processing., vol. 36, no. 10, pp. 1575–1581, Oct. 1988
3. Steven Johnson and Matteo frigo, "A Modified Split-Radix fast fourier transform with Fewer Arithmetic Operations," IEEE Transaction. Circuits Systems II, Analog Digital Signal Processing, vol. 39, no. 5, pp. 312–316, May 1992.
4. Amarnath Reddy and Venkata Suman "Design and Simulation of FFT Processor Using Radix-4 Algorithm Using FPGA,"International Journal of Advanced Science and Technology, vol. 61, pp.53-62, 2013.
5. Zhuo Qian, Nasibeh Nasiri, Oren Segal, and Martin Margala, "FPGA implementation of low-power split-radix fast fourier transform processors," in Proc. 24th International Conference. Field Program. Logic Applications., Munich, Germany, Sep. 2014, pp. 1–2.
6. Dr. Skodras and Prof Constantinides, "Efficient computation of the split-radix fast fourier transform," IEEE Pro. Radar Signal Process., vol. 139, no. 1, pp. 56–60, Feb. 1992.
7. Henrik V. Sorensen, Michael T. Heideman, and Sidney Burrus, "On computing the split-radix FFT," IEEE Transaction. Acoust., Signal Process., vol. 34, no. 1, pp. 152–156, Feb. 1986.
8. Wen-Chang Yeh and Chein-Wei Jen, "High-speed and low-power split-radix FFT," IEEE Transaction. Signal Processing, vol. 51, no. 3, pp. 864–874, March. 2003.
9. Zhuo Qian and Martin Margala, "Low-Power Split-Radix FFT Processors Using Radix-2 Butterfly Units," IEEE transaction. Very large scale integration., vol. 24, no. 9, March 2016

Sonali D. Patil received the B. E. degree from Dr. Babasaheb Ambedkar Marathwada University Aurangabad, Maharashtra in 2010.

Currently pursuing M.E. from D. Y. Patil College of Engineerng Akurdi, Pune-44

Dr. Manish Sharma received the B.E. degree from R. G. V. P. Bhopal in 2004, the M.E. degree from D. A. V. V. Indore in 2006, and the Ph.D. degree from D. A. V. V. Indore in 2014.

He is currently working as Associate professor in D. Y. Patil College of Engineering Akurdi, Pune-44. He posses research interests in Adaptive control, Wavelet neural network, signal processing, Cognitive radios, Software defined radios, and FPGA implementation.