

A Low-Power High Throughput Configurable FFT/IFFT Processor for WLAN and WiMax Protocols

Renan Netto, Pedro Michel, José Luís A. Güntzel

renan77@inf.ufsc.br, pedromichel@grad.ufsc.br, guntzel@inf.ufsc.br

**Embedded Computing Lab. (ECL) – Department of Informatics and Statistics
Federal University of Santa Catarina – Florianópolis, Brazil**

Abstract

This paper presents a configurable Fast Fourier Transform (FFT) processor targeting the IEEE 802.11n (WLAN) and the IEEE 802.16 (WiMax) wireless protocols. Such processor is based upon the Radix-2 Single-Path Delay Feedback (R2SDF) architecture and can be configured to operate on 64/128/512/1024/2048-point sequences. It was synthesized for a 90nm commercial standard-cells library by using Synopsys Design Compiler tool. Synthesis results shown that the designed processor consumes up to 76% less power than two similar processors found in the literature, but at a cost of up to 2.6 times more area. The SQNR values were evaluated by comparing the results provided by the designed FFT processor to that obtained from a software implementation of the FFT algorithm.

1. Introduction

Wireless communication is becoming a must in electronic products. Currently, it is found not only in personal portable devices but also in printers, keyboards, mice, among many others. Since such products have distinct operation features, new wireless communication protocols, more suited for specific applications, are being devised. Currently, many contemporary electronic products adopt the IEEE 802.11n WLAN or the IEEE 802.16 WiMax wireless communication standards (or even both). Wireless modulation/demodulation requires the application of the Discrete Fourier Transform (DFT) and the Inverse Discrete Fourier Transform (IDFT).

The original DFT algorithm has an $O(n^2)$ complexity and, therefore, is not used in direct hardware realizations [1]. A lower complexity DFT form, referred to as Fast Fourier Transform (FFT) algorithm, is used instead. The FFT algorithm has come into focus through a paper by Cooley and Tukey [2] in 1965. Also, the Inverse Fast Fourier Transform (IFFT) is used to calculate the IDFT.

The WLAN protocol requires the FFT/IFFT calculation of sequences with 64 or 128 points, whereas the WiMax protocol requires the FFT/IFFT for sequences with 128, 512, 1024 or 2048 points [3]. To meet such requirements in a single solution while aiming current electronic devices, we present and evaluate a low-power high throughput configurable FFT/IFFT processor to calculate 64/128/512/1024/2048-point FFT/IFFT based on the Radix-2 Single-Path Delay Feedback (R2SDF) architecture [4].

The proposed configurable FFT/IFFT processor was modeled in Verilog HDL and synthesized to TSMC 90nm standard-cells library using Synopsys Design Compiler tool [5]. It was validated through the simulation of testbenches in ModelSim environment. The SQNR in FFT/IFFT calculation was also evaluated. Synthesis results were used to compare the designed processor to two similar ones found in the literature.

The remaining of this paper is organized as follows. Section 2 presents the chosen decomposition of the FFT/IFFT algorithm. Section 3 highlights the main characteristics of the basic Radix-2 Single-Path Delay Feedback (R2SDF) architecture and presents the changes made to allow for the configurability. Section 4 presents the synthesis results and establishes comparisons between the designed processor and two similar processors. Finally, section 5 draws some conclusions and comments on future works.

2. Basic FFT/IFFT algorithms

The Discrete Fourier Transform (DFT) for an N -point sequence $x(n)$ is given by (1), where $X(k)$ and $x(n)$ are complex numbers, n is the time index and k is the frequency index [1]. By using this equation the DFT computation requires N^2 complex multiplications and $N(N-1)$ complex additions, leading to a complexity of $O(N^2)$. In [2] Cooley and Tukey proposed (actually they "rediscovered") an algorithm to compute the DFT with complexity $O(N \log_2 N)$. Such algorithm, known as Fast Fourier Transform (FFT), is widely used in hardware accelerators.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi kn/N} \quad (1)$$

Direct hardware implementations of the FFT algorithm either use the Decimation in Time (DIT) decomposition or the Decimation in Frequency (DIF) decomposition [1], the latter used in this work. In the DIF approach an N -point DFT is decomposed into two $N/2$ -point DFTs, one for the even-indexed frequency outputs

and another for the odd-indexed frequency outputs. The inputs to the even-indexed outputs $N/2$ -point DFT are sums between first half inputs and second half inputs. The inputs to the odd-indexed outputs $N/2$ -point DFT are differences between first half inputs and second half inputs, multiplied by constants that are referred to as "twiddle factors".

Each $N/2$ -point DFT may be further decomposed into two $N/4$ -point DFTs. Such decomposition may be applied recursively until reaching 2-point DFTs, which can be assumed as basic computation elements. Fig. 1 shows in detail the signal flow chart (SFG) for an 8-point DIF FFT.

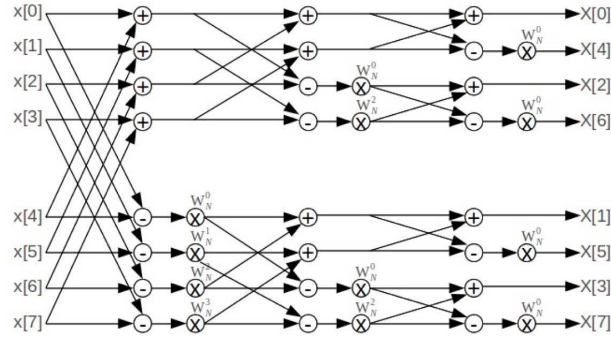


Fig. 1 – Signal flow graph (SFG) for 8-point DIF FFT.

A 2-point FFT processing element, normally referred to as "butterfly", is composed by one complex multiplication, one complex addition and one complex subtraction. Therefore, an N -point FFT has $\log_2 N$ stages where each stage has $N/2$ complex multiplications, $N/2$ complex additions and $N/2$ complex subtractions, resulting in $O(N \log_2 N)$ complexity, which is significantly lower than that of a direct implementation of (1).

The Inverse Discrete Fourier Transform (IDFT) is given by (2). As it can be seen, the same algorithm used to calculate the DFT can be used to calculate the IDFT, with minor changes (the sign of the imaginary part of the coefficients in the multiplication and a division by N of the final results).

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cdot e^{\frac{j2\pi kn}{N}} \quad (2)$$

3. R2SDF-based FFT/IFFT configurable processor

We have designed a dedicated processor that can be configured to compute the FFT/IFFT for sequences with 64, 128, 512, 1024 or 2048 points, as required by the IEEE 802.11n (WLAN) and the IEEE 802.16 (WiMax) protocols. The processor architecture is based upon the well-known Radix-2 Single-Path Delay Feedback (R2SDF) architecture [4], which was carefully modified to allow the desired configurability.

The original R2SDF architecture datapath is built up from the basic stage, shown in Fig. 2, which is composed by one radix-2 butterfly (either DIF or DIT), a ROM memory (to hold the stage twiddle factors), two multiplexers and a delay element ("D"). Both ROM memory and delay element were implemented using registers. The delay element is organized as a FIFO (first-in first-out) structure, being able to store $N/2^i$ inputs, where $i=1$ in the first stage and $i=\log_2 N$ in the last stage (N is the total number of points in a sequence i.e., the inputs applied to the first stage). At the beginning, the first $N/2^i$ inputs to the stage are fed into the FIFO, one after another. As a second step, each of the second $N/2^i$ inputs is directly applied to the butterfly along with each of the inputs previously stored in the FIFO. Then, the resulting lower butterfly output is fed back to the FIFO while the resulting upper butterfly output is fed to stage $i+1$.

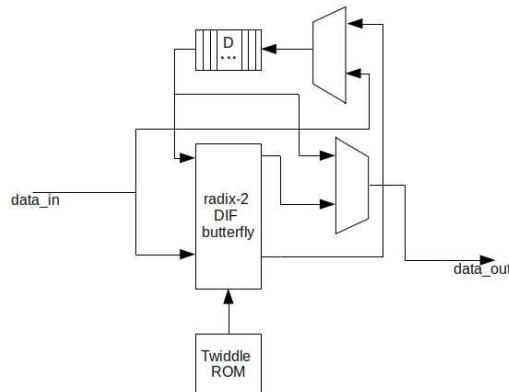


Fig. 2 – The i -th stage in the R2SDF architecture datapath.

Fig. 3 shows the simplified datapath block diagram of the designed R2SDF-based FFT/IFFT configurable processor, hereafter referred to as "CR2SDF". It is composed by 11 stages, being the first 10 stages similar to the one shown in Fig. 2. The last stage ($i=11$) was simplified, since it needs only one twiddle factor. Multiplexers were added to some stages to provide the desired configurability according to the desired number of points: 64, 128, 512, 1024 or 2048. To allow the configurability between FFT and IFFT the complex multiplier inside each butterfly was appropriately modified. As long as all of the possible numbers of points are powers of 2, the division by N in the IFFT can easily be done by shifting the result to the right. Besides the datapath of Fig. 3, the CR2SDF processor has a control block that generates the addresses used to access the twiddle factors in the corresponding ROMs. In order to implement the control block we have chosen the DIF decomposition, since it allows for a less complex circuitry.

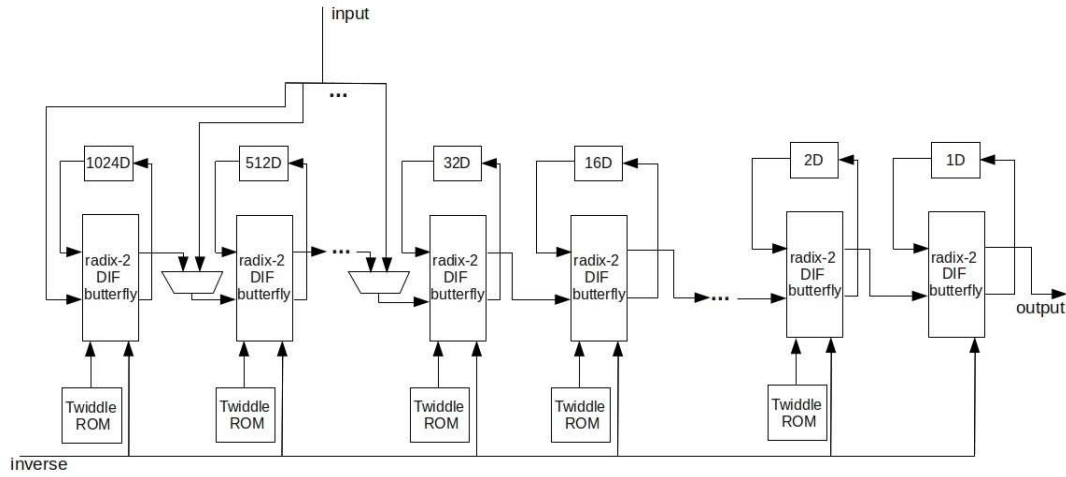


Fig. 3 – Datapath block diagram for the CR2SDF processor.

It is important to notice that, in the CR2SDF processor datapath (Fig. 3), the butterfly upper output of a given stage i is directly fed to the butterfly input in stage $i+1$. This way, there is a long combinational path that begins at the lower input of the first stage butterfly, traverses all 11 stages and ends at the upper output of the last stage butterfly. However, at every clock edge, a new point is applied to the first stage input, while the contents of the first stage FIFO is shifted. This assures high throughput because the data is forced to move forward along this combinational path at every clock edge, making the datapath to behave as a “wave pipeline”.

4. Synthesis results

The CR2SDF processor was described in Verilog HDL and synthesized to TSMC 90nm using Synopsys Design Compiler tool [5]. The synthesis results are summarized in Tab 1.

Tab.1 - Synthesis results for CR2SDF processor.

Technology	90 nm
Vdd	1.0 V
Length support	64 to 2048 bits
Area	2.6 mm ²
Power	66.14 mW
Max frequency	142 Mhz
Latency	8190 clock cycles @ 2048-point
1/Throughput	4096 cycles/sequence @ 2048-point

The synthesized design was compared with [6] and [7]. To establish a fair comparison, the normalized area and power were calculated using equations from [7]. The results of this comparison are shown in Tab. 2. As it can be seen, the normalized energy of CR2SDF processor is 76% lower than that of [6] and 44% lower than that of [7]. Besides that, the proposed processor is also able to compute 64-point FFT, which is not the case of the processor in [6]. On the other hand, the CR2SDF processor presents a normalized area that is 1.5 times larger than that of [6] and 2.6 times larger than that of [7]. As it can be seen, even that the CR2SDF processor uses more area and has a higher operation frequency, it achieves a lower normalized energy. This happens due to the high throughput of the designed processor, that results in a lower energy consumption per FFT.

To verify the designed processor, a script in Python language was written to generate appropriate random inputs, which were used only to verify the functionality by calculating an FFT/IFFT using the CR2SDF

processor and a software solution based on the FFTW library [8], written in C language. After that, the signal-to-quantization-noise ratio (SQNR) values of the designed processor were calculated. They are shown in Tab. 3. As one may observe, SQNR values are good, even for 2048-point sequences.

Tab.2 - Synthesis results comparison.

	CR2SDF (Proposed)	Chhatbar [6]	Jen-Chi Kuo [7]
Technology	90 nm	180 nm	350 nm
Vdd	1.0 V	1.8 V	3.3 V
Length support	64 to 2048 bits	128 to 2048 bits	64 to 2048 bits
Normalized area	1269 $\mu\text{m}^2/\text{point}$	825 $\mu\text{m}^2/\text{point}$	491 $\mu\text{m}^2/\text{point}$
Normalized energy	232 $\mu\text{W}/\text{point.Hz}$	1005 $\mu\text{W}/\text{point.Hz}$	428 $\mu\text{W}/\text{point.Hz}$
Maximum frequency	142 MHz	40 MHz	80 MHz

Tab.3 - SQNR of CR2SDF processor

Points	FFT SQNR (db)	IFFT SQNR (db)
64	77.82	73.05
128	66.98	68.14
512	64.72	64.43
1024	64.20	61.18
2048	64.30	64.63

5. Conclusions and future work

This paper presented an R2SDF-based FFT/IFFT processor (CR2SDF) that can be configured to process 64/128/512/1024/2048-point sequences. Such FFT/IFFT processor was modeled with Verilog HDL and synthesized for TSMC 90nm standard-cells library by using Synopsys Design Compiler tool.

To compare with similar processors, the power and area obtained from the synthesis were normalized. The CR2SDF processor showed a normalized energy that is up to 76% lower and a frequency that is up to 3.5 times higher than those of other similar processors. However, it presents a normalized area that is up to 2.6 times larger than other processors.

Compared to an FFT software implementation the CR2SDF processor presented a good precision: the lowest SQNR value was 61.18 db.

As future works we intend to synthesize the designed processor with alternative fast adders, low power multipliers and low-power techniques as Low-Vdd/High-Vt, in order to compare it to a wider set of similar processors. We also devise several architectural modifications in the CR2SDF datapath to reduce hardware cost and to increase energy efficiency as well, like the use of butterflies with higher radix (allowing more energy efficient addition and multiplication schemes).

6. References

- [1] A. V. Oppenheim and R. W. Schafe, Discrete-time signal processing. Englewood Cliffs: Prentice Hall. 1989.
- [2] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput, vol. 19, pp. 297-301, April 1965.
- [3] G. Yang and Y. Jung, "Scalable FFT processor for MIMO-OFDM based SDR systems," In: 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC), 2010. pp. 517-521.
- [4] H. L. Groginsky and G.A. Works. "A Pipeline Fast Fourier Transform", IEEE Transactions on Computers, Vol.C-19(11), pp. 1015-1019, November 1970.
- [5] Synopsys, Inc, <<http://www.synopsys.com/home.aspx>> Accessed in February 2012.
- [6] T. D. Chhatbar and A. D. Darji, "High Speed High Throughput FFT/IFFT Processor ASIC for Mobile Wi-Max", Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09, pp. 402-405
- [7] J. Kuo, et al, "VLSI Design of a Variable-Length FFT/IFFT Processor for OFDM-Based Communication Systems", EURASIP Journal on Applied Signal Processing 2003:13, pp. 1306-1316.
- [8] FFTW, <<http://www.fftw.org/>>. Accessed in January 2012.