

Design of 16-point Radix-4 Fast Fourier Transform in 0.18µm CMOS Technology

¹Siva Kumar Palaniappan and ²Tun Zainal Azni Zulkifli

¹RFIC Design Group, Faculty of Electrical and Electronics Engineering, Universiti Sains Malaysia

²Engineering Campus, 14300 Nibong Tebal, Seberang Perai Selatan, Penang, Malaysia

Abstract: This paper introduces detail design of semi-custom CMOS Fast Fourier Transform (FFT) architecture for computing 16-point radix-4 FFT. FFT is one of the most widely used algorithms in digital signal processing. It is used in many signal processing and communication application as an important block for various multi-carrier systems such as for WLAN (Wireless local area network). This paper describes the design of an ASIC (Application Specific Integrated Circuit) CMOS FFT processor for 16-point radix-4 complex FFT computation, realized utilizing 0.18µm standard CMOS technology. Fixed point data format is preferred in comparison of floating point data format for a shorter dynamic range and reduced hardware utilization; thus, catering to the needs of portability. Furthermore, computations results at particular stage are rounded to avoid overflow issue and to be stored in register. The computation speed of the design is observed to be 50MHz after the synthesis process. Compared to traditional radix-4 algorithm the architecture proposed for 16-point FFT results in 1.73% of power saving and 5.5% of area reduction.

Key words: ASIC, Digital Signal Processing (DSP), FFT

INTRODUCTION

The Discrete Fourier Transform (DFT) plays a significantly important role in many applications of digital signal processing. Basically, it has been applied in a wide range of fields such as linear filtering, spectrum analysis, digital video broadcasting and orthogonal frequency demodulation multiplexing (OFDM). The rapidly increasing demand of OFDM-based applications, including modern wireless telecommunication such as LAN, needs real-time high speed computation in Fast Fourier Transform algorithm. This has made the design of FFT processor a critical requirement for the up coming wireless technology^[1]. With the advent of this requirement, the study of high performance VLSI FFT architecture is likewise of increasing importance. Many different hardware architectures have been proposed for the implementation of FFT algorithms. The main concern of the design approach will be power and architectural size.

Among various FFT algorithms, radix-2 FFT with Cooley-Turkey algorithm, is very popular because it makes efficient use of symmetry and periodicity properties of the twiddle factor/coefficient

$W_N^n = \exp\left(\frac{-2\pi jn}{N}\right)$ which reduce the computational complexity from $O(N^2)$ to $O(N\log_2 N)$ ^[2]. Several architectures have been proposed based on Cooley-Turkey algorithm to further reduce the computation complexity, including radix-4, radix-2, and split-radix. Basically, this Fast Fourier Transform algorithm use Divide-and-Conquer approach to divide the computation recursively and then extract as many common twiddle factors as possible.

The number of required real additions and multiplications is usually used to compare the efficiency of different FFT algorithms. In terms of the multiplicative comparison, the split-radix FFT is computationally better to all the other algorithms because it has most trivial multiplications^[3]. Eventually, this algorithm has a drawback because of irregular structure that leads this algorithm not suitable for implementation on digital signal processors. Structural regularity is also important in implementation of FFT algorithms on dedicated chips such as in ASIC (Application Specific Integrated Chip). Hence, radix-2 and radix-4 FFT algorithm are preferable in terms of speed and accuracy.

This paper presents an area and power efficient 16-point radix-4 Fast Fourier Transform. The approach in

Corresponding Author: Tun Zainal Azni Zulkifli, RFIC Design Group, Faculty of Electrical and Electronics Engineering, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia, Tel:+604-5996086, Fax:+604-5941023

re-utilizing the stored identical component enhances the physical finger print of the architecture. An improved complex multiplication is introduced in FFT butterfly computation to realize a cost efficient hardware. 16-point FFT radix-4 architecture is implemented utilizing 0.18µm technology from Artisan. The 16 bit imaginary and 16 bit real input-output is realized at 1.8V with operating frequency of 50MHz.

The chip is designed for fixed-point data format. Great care had been taken into account to overcome the overflow issue in fixed-point data format^[4]. During the FFT computation, results at a particular stage are rounded and stored in the register memory. Since the FFT computation is an iterative process, the successive rounding errors at each output of butterfly accumulate over the FFT stages. The issue is solved by maintaining the error at the successive butterfly small. Twiddle factor/coefficient value are pre-calculated and stored in the register memory as 16-bit two's complement signed fixed-point words.

This paper is organized as follows. Conventional radix-4 algorithm is described followed by the modified radix-2 description. Subsequently, the proposed radix-4 circuit implementation is presented with butterfly architecture and controller. The comparison results between conventional radix-4 and modified radix-4 architecture realized in 0.18µm CMOS technology are reported in simulation results. The paper is summarized with an elaboration of a conclusion describing contribution of this work.

RADIX-4 ALGORITHM: The N-point Discrete Fourier Transform DFT of a sequence $x(n)$ is defined as [5]:

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

where,
 $k = 0, 1, \dots, (N-1)$

The twiddle factor W_N is given by:

$$W_N = e^{-j(2\pi/N)} \quad (2)$$

where $x(n)$ is the time domain discrete input signal and $X(k)$ is the DFT. Value n represents the discrete time-domain index, while k is the normalized frequency domain index.

Divide-and-Conquer approach is adopted in DFT algorithm to make the computation more efficient. The basic idea of this approach is to decompose the N-point DFT into successively smaller DFTs. This algorithm is known as FFT. Among the entire FFT algorithm, radix-4 decimation in time approach is used in this

paper. N can be factored as a product of two integers that is^[3]:

$$N = LM \quad (3)$$

the sequence $x(n)$, $0 \leq n \leq N-1$ is stored in two-dimensional by l and m , where $0 \leq l \leq L-1$ and $0 \leq m \leq M-1$.

Thus, the sequence $x(n)$ is stored in rectangular array by mapping of index n to the indexes (l, m) as follow:

$$n = l + mL \quad (4)$$

Thus, the stored sequence $x(n)$ is shown in Fig. 1.

	m				
L	0	1	2	...	M-1
0	x(0)	x(L)	x(2L)	...	x((M-1)L)
1	x(1)	x(L+1)	x(2L+1)	...	x((M-1)L+1)
2	x(2)	x(L+2)	x(2L+2)	...	x((M-1)L+2)
:	:	:	:	...	:
:	:	:	:	...	:
L-1	x(L-1)	x(2L-1)	x(3L-1)	...	x(LM-1)

Fig. 1: Arrangement of $x(n)$ sequence for data array

A similar arrangement is used to map index k to a pair of indices (p, q) , where $0 \leq p \leq L-1$ and $0 \leq q \leq M-1$.

Thus, the sequence $X(k)$ is stored in rectangular array by mapping of index k to the indexes (p, q) as follow:

$$k = Mp + q \quad (5)$$

$X(k)$ is mapped into corresponding rectangular array $X(p, q)$ and $x(n)$ is mapped into the rectangular array $x(l, m)$. The DFT can be expressed as a double sum over the elements of the $x(n)$ and $X(k)$ multiplied by the corresponding phase factors as follow:

$$X(p, q) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l, m)W_N^{(Mp+q)(mL+l)} \quad (6)$$

where,

$$W_N^{(Mp+q)(mL+l)} = W_N^{MLmp} W_N^{mLq} W_N^{Mpl} W_N^{lq}$$

however,

$$W_N^{Nmp} = 1, W_N^{mqL} = W_{N/L}^{mq} = W_M^{mq}$$

and,

$$W_N^{Mpl} = W_{N/M}^{pl} = W_L^{pl}$$

From the simplification, equation 7 can be expressed as

$$X(p, q) = \sum_{l=0}^{L-1} \left\{ W_N^{lq} \left[\sum_{m=0}^{M-1} x(l, m) W_M^{mq} \right] \right\} W_L^{lp} \quad (7)$$

For radix-4, the flow graph of a 16-point FFT based on the above formulation is shown in Fig.2. The corresponding equations are as follows:

$$X(4p + q) = \sum_{l=0}^3 F(l, q) W_4^{lp} \quad (8)$$

where,
 $p = 0, 1, 2, 3$ and
 $F(l, q)$ is given,

$$F(l, q) = W_N^{lq} \sum_{m=0}^3 x(4m + l) W_4^{mq} \quad (9)$$

given that,

$$l = 0, 1, 2, 3 \text{ and } q = 0, 1, 2, \dots, \frac{N}{4} - 1$$

Figure 2 shows a signal flow graph of a radix-4 16-point FFT. The number inside the circle is the value of q (for stage 1) or p (for stage 2) [6]. The number outside the circle is the FFT coefficient applied.

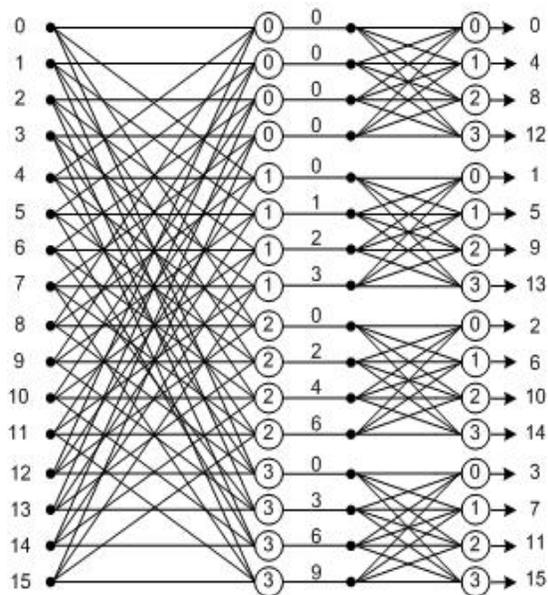


Fig. 2: Signal flow graph of a FFT radix-4 16-point

Modified Radix-4: Radix-4 for computation increases the addition/subtraction count compare to radix-2. Thus, to reduce the addition/subtraction of the radix-4 design, matrix of the linear transformation is used as follows:

$$\begin{bmatrix} X(0, q) \\ X(1, q) \\ X(2, q) \\ X(3, q) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} F(0, q) \\ F(1, q) \\ F(2, q) \\ F(3, q) \end{bmatrix} \quad (10)$$

The total number of complex addition/subtraction is reduced to $N \log_2 N$, which is identical to the radix-2 algorithm. This approach saves 33% of adders/subtract required.

A complex multiplication can be reduced to three multiplications by the following improved algorithm^[1]:

$$A + Bj = (X + Yj)(L + Mj) \quad (11)$$

Real,

$$A = (L - M)Y + L(X - Y) \quad (12)$$

Imaginary,

$$B = (L + M)X - L(X - Y) \quad (13)$$

Value of $L + M$, $L - M$ and $L(X - Y)$ are calculated manually and saved in registers. This algorithm, manage to reduce to three constant multiplication and three addition/subtraction of the computation. The complex multiplication structure is shown in Fig. 3.

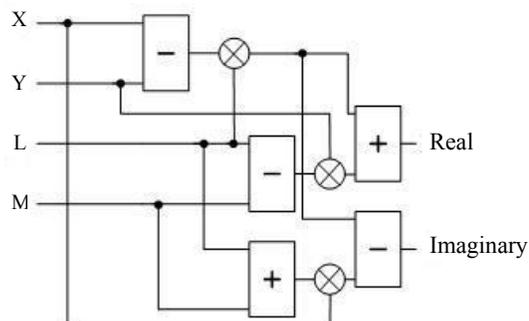


Fig. 3: Complex multiplier architecture

Circuit Implementation: The radix-4 16-point FFT was designed using verilog code and simulated in NcVerilog Cadence in order to verify its functionality. The design is synthesized utilizing 0.18μm technology provided by Artisan Library. Timing constraint is set with operating frequency 50MHz.

FFT architecture is divided into three main process blocks. The block diagram of process block is shown in Fig. 4.

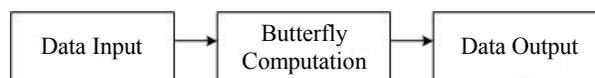


Fig. 4: Data process block

This block consist of data input, butterfly computation and data output. The data is read in every rising edge of clock and stored in the memory register. Butterfly computation block compute the stored data before going to data output process. The data is kept in the register before it is read out.

The FFT radix-4 processor architecture consist of a butterfly architecture, memory register, control circuit, serial to parallel and parallel to serial converter. Twiddle factor are stored as 16-bit two's complement signed fixed point word. The block diagram representation of FFT architecture design is shown in Fig. 5.

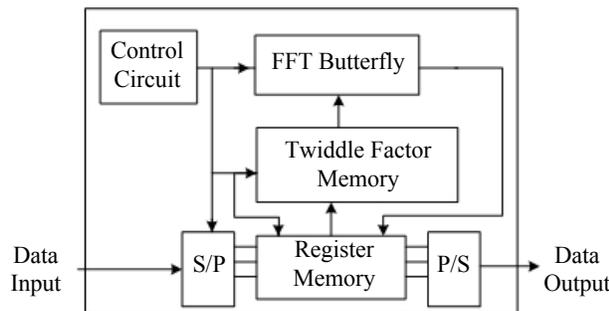


Fig. 5: FFT processor architecture

A. Butterfly Architecture: The most important element in FFT processor is a butterfly structure. It takes two signed fixed-point data from memory register and computes the FFT algorithm. The output results are written back in same memory location as the previous input stored. This method is called in-placement memory storage whereby it can reduce the hardware utilization. The butterfly architecture is shown in Fig. 6.

The adder sums the input before being multiplied by the twiddle factor. The multiplier forms the partial product of the complex multiplication and produce two times bigger then input bit. Shift register would shift the bits to avoid overflow issue. Output of this butterfly would be kept in the register for the subsequent stage.

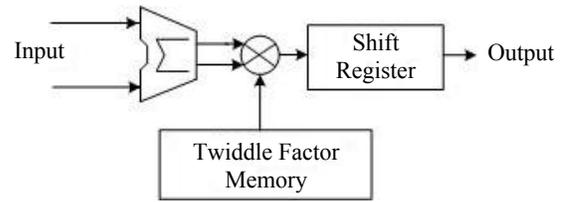


Fig. 6: Butterfly architecture

B. Controller: The FFT processor event is determined by the control circuit depending on the feedback it receives from the surrounding unit. Moore machine approach is adapted whereby the output signal dependant to the value of next state. This design functions as a synchronous design which controlled by "CLK" signal. The input signal "RST" is used to reset the FFT processor including the input buffer which holds data for next stage. Input "EN" signal is used to control the state transition of the processor. Signal "SYNC_OUT" would be enable when the output signal is generated.

SIMULATION RESULTS

After the synthesis process, gate-level-simulation was performed in order to verify its functionality with SDF (standard delay format) back annotation. The operating frequency of the design is 50MHz. Table 1 summarizes the cell area of the conventional radix-4 and proposed radix-4 as reported from Encounter (Cadence) back-end tool.

Table 1: Area comparison between conventional radix-4 and proposed radix-4

Design	Area (mm ²)
Conventional Radix-4	0.5625
Proposed Radix-4	0.5314

In comparison the proposed radix-4 design is enhanced in area consumption, than the conventional architecture. The fingerprint area for conventional radix-4 and proposed radix-4 is shown in Fig. 7.

The dynamic power consumption of the proposed radix-4 is observed to be less than the conventional radix-4. Table 2 shows power comparison between the proposed radix-4 and conventional radix-4 architecture.

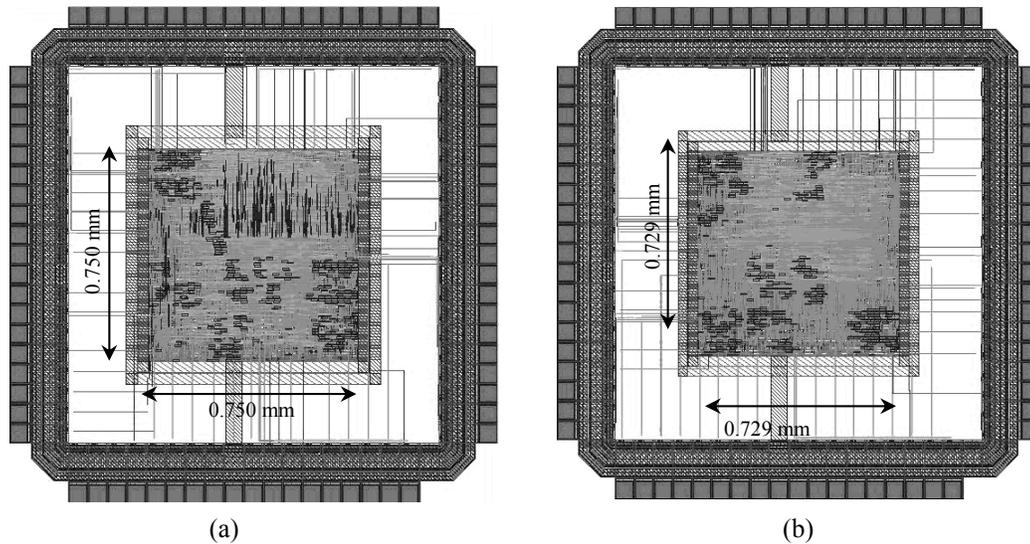


Fig. 7 (a) Conventional Radix-4, (b) Proposed Radix-4

Table 2: Power comparison between conventional radix-4 and proposed radix-4

Design	Results(mW)
Conventional Radix-4	22.8
Proposed Radix-4	22.4

CONCLUSION

Simulation results shows proposed FFT radix-4 represents a better and efficient architecture for computing FFT. This design facilitates the efficient computation of long FFT which usually require a huge architecture. In this design process, many identical components are being reused in which it reduces the gate count of the design. This is due to simplification of the mathematic algorithm in FFT structure. The comparison shows that the chip can reach low cost and low power for OFDM system applications.

ACKNOWLEDGMENT

The authors would like to acknowledge Albert Victor Kordesch from Silterra Malaysia Sdn.Bhd., for fabrication of the experimental ASIC chip in 0.18 μ m CMOS technology.

REFERENCES

1. M. Jiang, B. Yang, Y. Fu, A. Jiang, X. Wang, X. Gan, B. Zhao, and T. Zhang, 2004. Design of FFT processor with low power complex multiplier for OFDM-based high-speed wireless applications: Symp. ISCIT'04: 639-641.
2. W. Chang, and T. Nguyen, 2006. An OFDM-Specified Lossless FFT Architecture: IEEE Transaction on Circuits and Systems – I: 1235-1243.
3. J G. Proakis and D. G. Manolakis, 1996. Digital Signal Processing Principles, Algorithm, and Applications. Prentice Hall, pp: 448-475.
4. E. Cetin, R. C. S. Morling, and I. Kale, 1997. An integrated 256-point complex FFT processor for real-time spectrum analysis and measurement: IEEE Proc. of Instrumentation and Measurement Technology Conference'97: 96-101.
5. M. El-Khashab, and E. E. Swartzlander Wegmuller, 2003. An architecture for a radix-4 modular pipelined Fast Fourier Transform: Proc. of the Application Specific Systems, Architectures and Processors (ASAP'03).

6. M. Hasan, T. Arshan, and J.S. Thomson, 2003. A novel coefficient ordering based low power pipelined radix-4 FFT processor for wireless LAN applications: IEEE Trans. of Consumer Electronics: 128-134.
7. S. Xin, Z. Tiejun, and H. Chaohuan, 2004. A High-performance power-efficient structure of FFT (Fast Fourier Transform) processor: International Conference of Signal Processing: 555-558.
8. C. Meletis, P. Bougas, G. Economakos, P. Kalivas, and K. Pekmestzi, 2004. High-speed pipelined implementation of radix-2 DIF algorithm: Trans. of Engineering, Computing and Technology (ENFORMATIKA'04): 305-308.
9. V. J. Alberto, R. R. Jesus, and O. Alejandro, 2005. VHDL core for 1024-point radix-4 FFT computation: Proc. of the 2005 International Conference on Reconfigurable Computing and FPGAs (ReConFig 2005).
10. N. Weste, M. Bickerstaff, T. Arivoli, P. J. Ryan, J. W. Dalton, D. J. Skellern and T. M. Percival, 1997. A 50MHz 16-point FFT processor for WLAN application: IEEE 1997 Custom Integrated Circuits Conference: 457-460.