Hardware Implementation Of The Fast Fourier Transform With Non-Power-Of-Two Problem Size

Samiha Aouissi
Badji Mokhtar University.Annaba
. Algeria
aouissi-samiha @hotmail.fr

Mohamed Benouaret
Badji Mokhtar University.Annaba.
Algeria
mohamed.benouaret @gmail.com

Saliha Harize
Badji Mokhtar University. Annaba.
Algeria
shrz.dj@gmail.com

***Abstract*— Transforms such as the famous Fast Fourier transform (FFT) are very commonly used, practically, in spectral analysis, communications, and scientific computing applications, FFT Processor becomes an indispensable and essential tool in the real-time implementation of algorithms dedicated to the behavioral frequency analysis of the cyclical phenomena supposed to be stationary. This paper deals with design and implementation of the FFT processor with non-power-of two problem size, using the radar algorithm which enjoys a special importance and has a specific structure whose main task is to overcome the constraints imposed by the choice of the complex multiplier which is usually accompanied by the butterfly architecture that shows itself greedy in terms of internal resources of the used FPGA target. The circuits have been described in Very high speed integrated circuit Hardware Description Language (VHDL).**

Keywords— rader,FFT; FPGA;VHDL

I. INTRODUCTION

Fast Fourier transform (FFT) is widely used in almost all fields of science and engineering ,where frequency- domain representation of a signal has to be analyzed [1–3].In communication area. Therefore, highly efficient algorithms for computing the FFT are of great importance. FFT sequence lengths include powers-of-two and also non-powers-of two [4]. In the applications of OFDM demodulation and modern microscopy, the sequence lengths may be non-powers-of-two.

They are difficult and time-consuming to implement in hardware because of the large number of algorithmic and data path options that must be considered in order to tailor the implementation to a particular application's cost and performance requirements [5-6]. The right choices are highly dependent on the context. These problems are worsened when the transform size is not powers of two, [7–10] algorithms become less regular and the relationship between algorithm and hardware implementation become more complicated. Most existing work on hardware implementations of the FFT with non-power-of- two problem size focuses on building an implementation to meet a specific application's cost and performance goals, not on creating a space of designs with different cost/performance characteristics[11]. For these reasons, This article presents the implementation methodology of one most effective methods based on FFT algorithm using Non-Power-Of-Two Problem Size in fpga circuit. Field Programmable Gate Array (FPGA) are becoming more and more popular and are used in many applications. However, it is well known that the performance is limited comparing to full ASIC implementation, but for many applications the speed requirements fit the ones provided already by existing FPGA circuits [12]. One of the important points in FPGAs is that the designer has to be done to generate anything and distribute clocks wherever they are necessaries (complex problem in the ASIC).

The rest of the paper is organized as follows. Section 2 Introduces the proposed rader algorithm in detail, Section3 Present the proposed methodology of implementation of rader algorithm for Power-Of-Two(prime) Problem Size. Section4 analyzes the performance of the proposed algorithm by comparing its computational com-plexity ,accesses to lookup table, and execution time with the deferent's points. Finally, Section5 offers concluding remarks.

II. THEORETICAL

Rader's algorithm (1968) is a fast Fourier transform (FFT) algorithm that computes the discrete Fourier transform (DFT) of prime sizes by re-expressing the DFT as a cyclic convolution. Since Rader's algorithm only depends upon the periodicity of the DFT kernel, it is directly applicable to any other transform (of prime order) with a similar property. The discrete Fourier transform of a sequence of N points [13] .

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

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

We **need** a notation for an integer modulo N. For brevity we will indicate this by the following expression:

$$X(k) = X(g^k \bmod N) \qquad (2)$$

If N is prime there is some number, a generator **g,** not necessarily unique, that generates all primitive elements such that there is a one-to-one mapping of the integers ***i***= 0,. . , N - 2 to the integers j=0,..,N-2,given by :

$$j = g^i \bmod N$$
$$g^i \in Z_{p/\{0\}}$$

For example, let N=7 and **g**= 3. The table below gives the mapping of i on to j.

i	0	1	2	3	4	5
j	1	3	2	6	4	5

Table 1: primitive root the order of data

Computation **of *X(k)*** when N **is prime**
In (1) we have an expression for ***X,*** for all ***k.***

and is to be computed directly. For the other X, we observe that x_0 is not to be multiplied, and we choose to add it last into the summation. We are left with the sequence {X(k)-x_0}, k = 1,2, . . , N- 1 to compute, given by the below cyclic convolution expression:

$$\boldsymbol{X}\,(g^k mod\ N) - x(0) = \sum_{n=0}^{N-2} x(g^k mod\ N)\, w_N^{g^{(k+n)} mod\ N} \quad (3)$$

In general way we first compute the DC component which is particularly simple:

$$x_0 = \sum_{n=0}^{N-1} x(n) \qquad (4)$$

And in second step, we proceed to evaluate the cyclic convolution of $\boldsymbol{X}(k) - x_0$ in matrix notation :

$$\begin{bmatrix} X(1) \\ X(3) \\ X(2) \\ X(6) \\ X(4) \\ X(5) \end{bmatrix} - x_0 = \begin{bmatrix} w_N^1 & w_N^3 & w_N^2 & w_N^6 & w_N^4 & w_N^5 \\ w_N^3 & w_N^2 & w_N^6 & w_N^4 & w_N^5 & w_N^1 \\ w_N^2 & w_N^6 & w_N^4 & w_N^5 & w_N^1 & w_N^3 \\ w_N^6 & w_N^4 & w_N^5 & w_N^1 & w_N^3 & w_N^2 \\ w_N^4 & w_N^5 & w_N^1 & w_N^3 & w_N^2 & w_N^6 \\ w_N^5 & w_N^1 & w_N^3 & w_N^2 & w_N^6 & w_N^4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_2 \\ x_6 \\ x_4 \\ x_5 \end{bmatrix} \quad (5)$$

We reorder the rows and columns according to table1 it yields:

$$\begin{bmatrix} X(1) \\ X(3) \\ X(2) \\ X(6) \\ X(4) \\ X(5) \end{bmatrix} - x_0 = \begin{bmatrix} x_1 & x_3 & x_2 & x_6 & x_4 & x_5 \\ x_5 & x_1 & x_3 & x_2 & x_6 & x_4 \\ x_4 & x_5 & x_1 & x_3 & x_2 & x_6 \\ x_6 & x_4 & x_5 & x_1 & x_3 & x_2 \\ x_2 & x_6 & x_4 & x_5 & x_1 & x_3 \\ x_3 & x_2 & x_6 & x_4 & x_5 & x_1 \end{bmatrix} \begin{bmatrix} w_N^1 \\ w_N^3 \\ w_N^2 \\ w_N^6 \\ w_N^4 \\ w_N^5 \end{bmatrix} \quad \textbf{(6)}$$

A matrix of this form is called a circular matrix of Toplitz structure. In fact, this matrix used to facilitate the calculation and to reduce the complexity of the mathematical problem .as in our case the problem modeled by filter as shown in section So, the graphically interpreted is depicted in e following design :

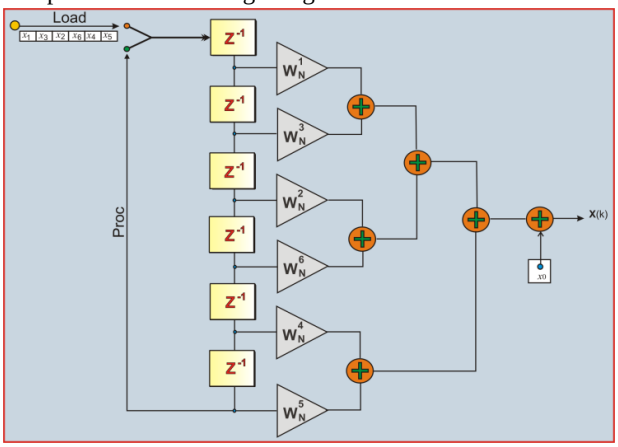


Figure 1: structure of filter

According to the formula number (6) transcript for designing a matched FIR filter of the DFT. In addition, implementing a Rader prime-factor algorithm is equivalent to implement the transposed and fast FIR filter structure using reduced adder graph (RAG) algorithm which seems better attractive rather than canonical signed digit (CSD) multiplier or a fully pipelined distributed arithmetic architecture (DA).

III. RADER ALGORITHM BASED FPGA IMPLEMENTION

Before embarking in the implementation phase it is convenient, first, to proceed to the signed 8-bit word quantization of the filter coefficients. Assuming that stream data and coefficients to be represented as a signed number the RAG implementation of all coefficients needs 10 adders for constant coefficients multipliers. So, the quantized coefficients are listed in the table 2. Because the Rader algorithm is restricted to prim lengths $L = 2^n \pm 1$ it shows less symmetry in coefficients representation.

	$256(W_7^k)$	
k	Real	Imag.
1	160	-200
3	-231	-111
2	-57	-250
6	160	200
4	-231	111
5	-57	250

Table 2: A signed 8-bit word of quantized coefficients

The implementation effort of the circular FIR filter in its transposed form is illustrated in table 3.

It can be seen from this table that the effort for CSD-adder filters can be calculated by $\left(M-\left(\frac{N+1}{2}\right)\right)/2$, where M is the coefficient bit width and N is the FFT filter length. For RAG-adder structure, the effort is only N(for M=19 bit, a factor $\approx {}^{19}/_{2} = \approx 10$ adders). For longer filters, RAG needs only one additional adder for each factor, because the pre-synthesized coefficient produces a dense grid of small coefficients representation.

FFT length	sin/cos	RAG adder	CDS adder	Auxiliary Factors
7	6	10	52	1,8,20,28
17	16	21	138	3,35,103,415,1153,1249
31	30	36	244	3,9,133,797,877,975
61	60	66	496	5,39,51,205,265,3211
127	124	126	1060	5,15,25

Table 3 implementation effort of proposed filter structure.

The implementation process has been divided into three components namely the rader_coeffs, rader_filter and the rader_state.

The rader_coeffs component is the first step of the design process and its main task is to calculate the coefficient data

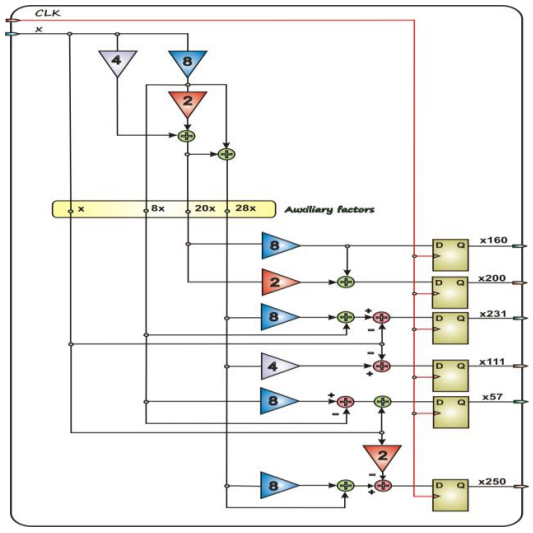
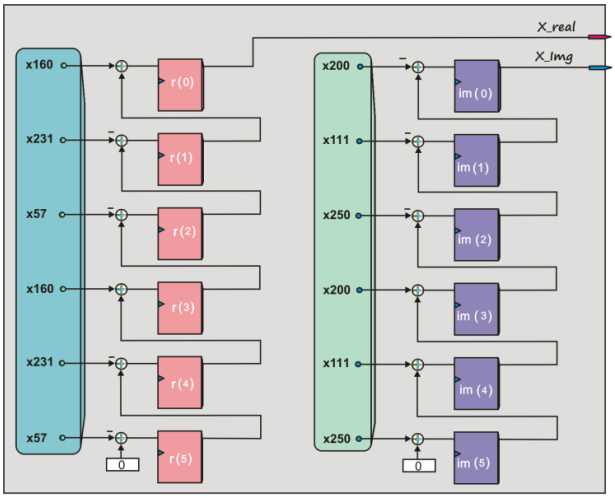


Figure 1: the RAG form for the quantized filter coefficients

The second component Composed by a register of shift and adder for filtered real and imagery data .



Real and imaginary parts of transposed structure FIR filter

Figure 2: implementation task of proposed FIR filter

The controller consists of a scaling accumulator and a time controller. This component is also responsible for the generation of the sampling signal of the Rader algorithm and for the assertion of the signal which enables the last circuit of the algorithm to output the final result. this component composite by three steps each step of specific condition.

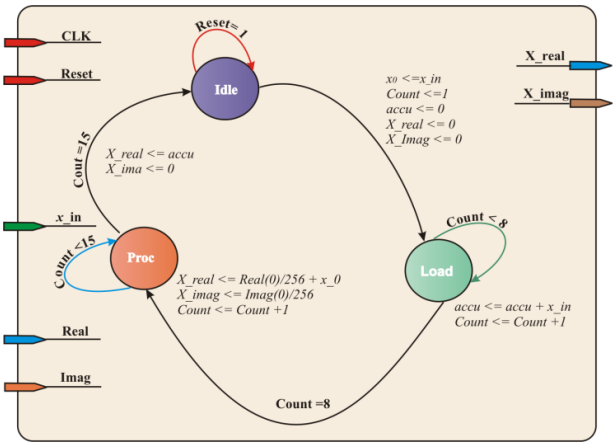


Figure 3: state machine circuit diagram.

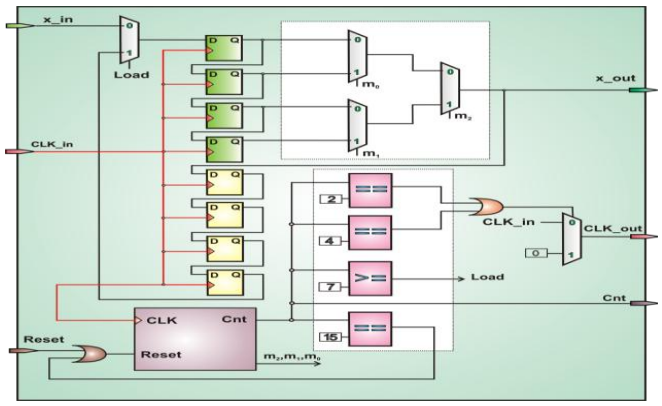


Figure 4 : incoming data splitter mechanism

Figure 4 show the graphic scheme of incoming data on radar algorithm for respected order in table 1. as well illustrated in the following lookup table (table 4).

Cnt	m_2	m_1	m_0	x_out	CLK_out
1	0	x	0	x1	CLK_in
2	0	x	1	x1	0
3	0	x	0	x3	CLK_in
4	0	x	1	x3	0
5	1	1	x	x2	CLK_in
6	0	x	0	x6	CLK_in
7	1	1	x	x4	CLK_in
8	1	1	x	x5	CLK_in
9	1	0	x	x1	CLK_in
10	0	x	1	x3	CLK_in
11	0	x	1	x2	CLK_in
12	0	x	1	x6	CLK_in
13	0	x	1	x4	CLK_in
14	0	x	1	x5	CLK_in
15	Done				

Table4: stream data feeding device

The delay to obtain the output (FFT of data) is function of the chosen resolution. QuartusII. It shows the different components of the implemented the 7 point FFT by rader method, how they are interconnected as well as the various control signals.

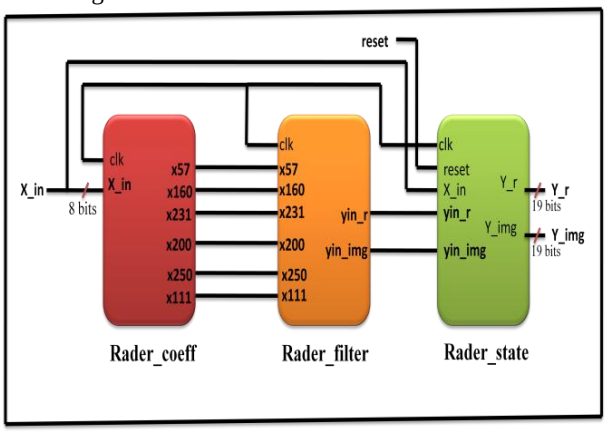


Figure 5: Main component for 7 points fft

IV Results and discussion

The high level description of the implementation.was written in VHDL language. The simulations were run on Modelsim , and the results compared with those obtained from Matlab. The input data pattern chosen in the range [-128 127]. The Modelsim simulation result is shown in figure 2. The output data pattern in the range [-262144 262143]. The implemented rader algorithm has a latency of 11 clock cycles . In the example shown in Figure 6 the results simulation gives the fast Fourier transform of serial points .At the end of the simulation, the output are reorder for obtain the correct result .

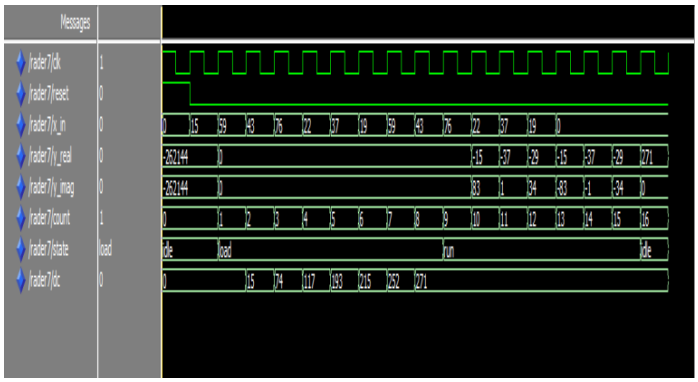


Figure 6: Modelsim simulation result for 7 points fft

IV Hardware Synthesis

This section presents the hardware evaluation results of the proposed architecture for the implementation of Rader algorithm for 7 ,11, 17 and 31 points FFT. The proposed architecture design was synthesized on a Quartus II based Altera's EP2C5F256C6 FPGA device of Cyclone II family. Table 5, shows the resource or component utilization of the proposed design implemented on abovementioned FPGA kit in terms of frequency, logic element and power consumption. Which was implemented on FPGA Programs were written in VHDL.

N points	Frequency (MHZ)	Logic element	Power consumption (mw)
7	138.73	493 (3%)	72.38
11	104.26	788 (5%)	72.68
17	96.79	1281 (9%)	72.86
31	91.80	1818 (13%)	73.12

Table 5 : resource utilization.

IV. CONCLUSION

Many features are added to the FPGAs with every generation, making it possible toper for computations at higher clock frequencies and in the most recent FPGAs, larger blocks aimed at digital signal processing (DSP) computations. this paper presents theoretical and practical aspects of implementing rader algorithm-based generators in FPGAs. The main results can be summarized as follows: A trade-off speed and area will determine the right structural approach to rader algorithm implementation for an application. Module count and operating speed depend significantly on the used synthesis tool. Simulation has shown that the redundant adder can improve the efficiency of rader FPGA implementations for bit-lengths the 20-bits. The hardware description language (HDL) code is generated to structurally simulate and verify the functionality of the design.

References

[1]P.Duhamel,M.Vetterli, Fast Fourier transforms: a tutorial review and a state of the art,Signal Process. 19(4) (1990) 259 –299.
[2] M. Frigo, S.Johnson, The design and implementation of FFTW3,Proc. IEEE 93(2) (2005) 216–231.
[3] G.S. Engel, T.R.Calhoun,E.L.Read,T. K.Ahn, T.Man cal,Y.-C. Cheng, R.E. Blankenship,G.R .Fleming, Evidence for wavelike energy trans-fer through quantum coherence in photo synthetic systems,Nature 446(7137)(2007)782–786.
[4] G.Bi,S.K. Mitra, S.Li, Sampling rate conversion based on DFT and DCT, Signal Process. 93 (2(February))(2013) 476–486.
[5] C.Rader ,Discrete Fourier transform swhen the number of data samples is prime,Proc. IEEE 56(6)(1968) 1107–1108.
[6] S.Winograd, On computing the discrete Fourier transform ,Math. Comput. 32 (141) (1978) 175–199.
[7] Y.Suzuki,T.Sone,K.Kido,A new FFT algorithm of radix 3,6, and 12, IEEE Trans. Acoust.Speech Signal Process.ASSP-34(2(April))(1986) 380–383.
[8] C. Temperton ,A new set of minimum-add small -N rotated DFT modules, J.Comput. Phys.75(1)(1988)190–198.
[9] W.Zheng,K.Li,Split-rdix algorithm for length 6 ^{m} DFT, IEEE Signal Process. Lett. 20(2013) 713–716.
[10] W. Zheng ,KenliLi, Keqin Li, A fast algorithm based on SR FFT for length N¼ q 8 2m DFTs, IEEE Trans. Circuits Syst .II:Express Briefs 61 (2013)110–114.
[11] Yicong Zhou [a,,] Weijia Cao [a], Licheng Liu [a], Sos Agaian [b], C.L. Philip Chen , Fast Fourier transform using matrix decomposition , Elsevier, Information Sciences 291 (2015) 172–183.
[12] Weihua Zheng [a], Kenli Li [a,n], Keqin Li [a,b] Data path-regular implementation and scaled technique For N=3*2^{N} DFTs, Elsevier, Signal Processing113(2015)68–79.
[13]https://en.wikipedia.org/wiki/Rader%27s_FFT_algorithm