DESIGN OF A PIPELINED R4SDF PROCESSOR

Nima Aghaee, Mohammad Eshghi

Department of Electrical Engineering, Shahid Beheshti University
Tehran, Iran
phone: + (98) 21 2990 2284, email: nima_aghaee@ieee.org, m-eshghi@sbu.ac.ir

ABSTRACT

In this paper, a pipelined architecture for a 16-point word-serial R4SDF Fast Fourier Transform processor is presented. In this design, a pipelined complex multiplier which consists of pipelined real multipliers is used in order to increase the speed of the processor. The architecture of R4SDF is itself pipelined, but it is further pipelined in this paper and is called a pipelined R4SDF.

The evaluation of the b-bit word length design shows that a speed up of (b/3)+1 with respect to a nonpipelined design is achieved. The average Signal to Error Ratio of the designed R4SDF processor for a word length of 9 bits, b=9, is 30.70 dB. The Speed Up for this word length is 4.

1. INTRODUCTION

The Discrete Fourier Transform (DFT) is a transform from the discrete time domain to the discrete frequency domain. The direct computation of the DFT includes multiplications of the order of N^2, where N is the length of the sequence. Cooley and Tukey in [1] present a design to reduce the computational complexity to the order of $O(N \log_2 N)$. Such DFT algorithms which are modified to have a higher speed are called the Fast Fourier Transform (FFT). Cooley-Tukey FFT (CTFFT) algorithm is a radix-2 algorithm. Structures proposed in [2] are also based on radix-2 algorithm and include an array processor, a column processor, and a Multi-path Delay Commutator (MDC).

A feedback based architecture which is similar to R2MDC, called Radix-2 Single-path Delay Feedback (R2SDF) is introduced in [3]. The feedback mechanism reduces delay elements with respect to the R2MDC.

A radix-4 SDF (R4SDF) architecture which uses a COordinate Rotation DIgital Computer (CORDIC) circuit, instead of a multiplier for twiddle factors, is introduced in [4]. Another R4SDF architecture is used and reported in [5].

FFT algorithms do not naturally interface with the sequential input data. To deal with this problem, two solutions are presented in [6]. These two solutions are adding the input delay stages before the first butterfly element, and using a fast real time processor. A new variant of Radix-4 Single-path Delay Commutator (R4SDC) processors is also introduced in [6]. The commutator structure is modified in order to reduce the memory size.

A pipelined structure of a long (8192 points) complex FFT is implemented on the VLSI circuit [7]. The authors in [7] proposed some pipelined arithmetic parts in order to increase the speed. A new internal scaling method is also used in that paper.

A radix-4 modular and pipelined DFT algorithm is introduced in [8]. Two $\sqrt{N}$ -point conventional pipelined FFT modules are combined to compute the N -point DFT.

A brief review of the various architectures for the pipelined FFT processors such as R2MDC and R4SDF are given in [9].

The MDC, SDC, and SDF architectures are inherently pipeline structures, based on Radix-*x* algorithms, *x=2, 4,* etc. Neither the pipeline architectures, nor the algorithms are limited to the above instances. For example a pipelined FFT processor based on the Prime-factor FFT Algorithm is introduced in [10].

In this paper, a new architecture for R4SDF processor using the pipelined complex and real multipliers is presented. The architecture of R4SDF is itself pipelined, but it is further pipelined in this paper and therefore is called pipelined R4SDF. The algorithm for a 16-point R4SDF processor is described in section 2. The new design of architecture for implementation of this algorithm is presented in section 3. The evaluation of the design is given in the section 4. The conclusion is presented in section 5.

2. ALGORITHM FOR A 16-POINT R4SDF

The DFT is defined in [1], [11], [12] as

$$X[K] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{Kn} \qquad K = 0,1,\ldots,N-1. \tag{1}$$

Where $x[n]$ is the time sequence, and $X[K]$ is the frequency sequence. N is the length of the sequence. The variables n and K are the time and the frequency indices, respectively. The W_N is defined as $W_N \underline{\underline{\Delta}} \exp(-j \cdot 2\pi / N)$.

The radix-4 algorithm, which is used in the R4SDF architecture, is applicable on a sequence that its length, N , is a power of 4. In this paper N is chosen to be 16, $N = 16 = 4^2$.

With the index mapping schemes given in (2) and (3), n is mapped to n_1 and n_2; and K is mapped to K_1 and K_2. The index mappings on the time index, n , in (2), and on the frequency index, K , in (3), are the same.

$$n = 4n_2 + n_1,\ n = 0,1,\ldots,N-1, \text{ for } n_1, n_2 = 0,1,2,3 \quad (2)$$

$$K = 4K_2 + K_1,\ K = 0,1,\ldots,N-1, \text{ for } K_1, K_2 = 0,1,2,3 \quad (3)$$

By using the index mappings of (2) and (3), equation (1) reformulates to (4).

$$X[4K_2 + K_1] = \sum_{n=0}^{N-1} x[4n_2 + n_1] \cdot W_N^{(4K_2+K_1)(4n_2+n_1)} \quad (4)$$

When $N = 16$, for $K_1 = 0,1,2,3$; equation (4) is rewritten as four sets of equations, which lead to a matrix form as (5).

$$\begin{bmatrix} X[n+0] \\ X[n+4] \\ X[n+8] \\ X[n+12] \end{bmatrix} = \mathbb{F}_4 \times \begin{bmatrix} b[4n+0] \\ b[4n+1] \\ b[4n+2] \\ b[4n+3] \end{bmatrix},\ n = 0,1,2,3 \quad (5)$$

where $\mathbb{F}_4$, the operator matrix which represents a 4-point DFT, is

$$\mathbb{F}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}, \quad (6)$$

and

$$\begin{cases} b[n+0] = a[n+0]W_N^0 \\ b[n+4] = a[n+4]W_N^n \\ b[n+8] = a[n+8]W_N^{2n} \\ b[n+12] = a[n+12]W_N^{3n} \end{cases},\ n = 0,1,2,3. \quad (7)$$

The " a " array is

$$\begin{bmatrix} a[n] \\ a[n+4] \\ a[n+8] \\ a[n+12 \end{bmatrix} = \mathbb{F}_4 \times \begin{bmatrix} x[n] \\ x[n+4] \\ x[n+8] \\ x[n+12] \end{bmatrix},\ n = 0,1,2,3\ . \quad (8)$$

Equations (8), (7), and (5) show that it is possible to perform a DFT in three steps [12]. These steps are in order and serial in time; i.e. the first step is (8), the second step is (7), and the last step is (5). The output values of this algorithm are in the bit-reversed order.

The above procedure to extract R4SDF algorithm for a 16-point transform is applicable for any DFT with a length of 4 to the power of *p*. In this case, for *p* greater than 2, there will be more 4-point DFTs and Twiddle factor multiplications in the final algorithm.

3. DESIGN OF A PIPELINED R4SDF

A 16-point R4SDF processor consists of two *Butterfly units*, a *Multiplier unit* and some register banks, organized in three steps [9], as shown in Figure 1. In this figure, there are six register banks. Each register bank in step-1 contains four registers, and in step-3 contains one register. The *Butterfly units* and the *Multiplier unit* are the processing units of the architecture.

The bottleneck of this pipelined processor is the *Multiplier unit*. In this paper, a pipelined multiplier, instead of a nonpipelined multiplier, is proposed In order to solve the bottleneck problem and to increase the total speed of the FFT processor.

3.1 Design of the step-1

As shown in Figure 1, the step-1 includes a *Butterfly unit* and three register banks. The *Butterfly unit* includes a 4-point butterfly and four multiplexers as shown in Figure 2. The three inputs, to the *Butterfly unit* are the outputs of three register banks. The remaining input is the processor's input. The three outputs, of the *Butterfly unit* are fed back into the three register banks. The forth output is the input of the *Multiplier unit*. The 'Sel' is a one bit input of the *Butterfly unit* to select its mode of operation.

The *Butterfly unit* has two operational modes, selected by the 'Sel' signal, see Figure 2. In the first mode, the *Butterfly unit* passes the inputs to the outputs. In the second mode, the *Butterfly unit* passes the DFTs of the inputs.

The register banks, always, save the three outputs of the *Butterfly unit*. There are four registers in each register bank, which compose a shift register.

3.2 Design of a Pipelined Complex Multiplier using Pipelined Real Multipliers

The complex multiplier is designed based on the following reformulated expansion [13]

$$(xr + jxi)\cdot(Wr + jWi) = \{[(xi + xr)\cdot Wr] - [Ws \cdot xi]\} + j\{[(xi + xr)\cdot Wr] + [Wd \cdot xr]\}. \quad (9)$$

Where Wr and the Wi are the real and the imaginary parts of the W_N, respectively. The Wd and Ws are the difference and the sum of the imaginary and the real parts of W_N.

$$Wd = Wi - Wr \quad (10)$$

$$Ws = Wi + Wr \quad (11)$$

For each W_N instead of saving two real values, Wr and Wi, three real values, Wr, Wd, and Ws are saved in the memory. The block diagram of this modified complex multiplier based on (9) is shown in Figure 3.

The direct design of a complex multiplier includes four real multipliers and two real adders. By using this modified design, instead of the direct design, the number of the real multipliers reduces to three, and the number of the real adders increases to three. Since the real multiplier is larger than the adder, the modified design is smaller than the direct design. Therefore, the modified complex multiplier [13] is used in this paper.

The *Multiplier unit* has the greatest delay time among the parts of the processor. The *Multiplier unit* is pipelined in order to increase the speed of the R4SDF processor.

The complex multiplier is pipelined as shown in Figure 4. This architecture has three steps. In this complex multiplier,

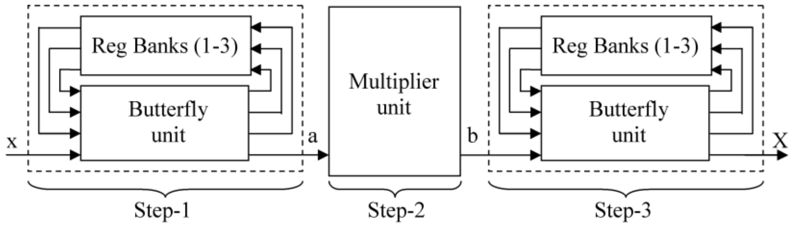

Figure 1 – Computation steps of radix-4 DFT

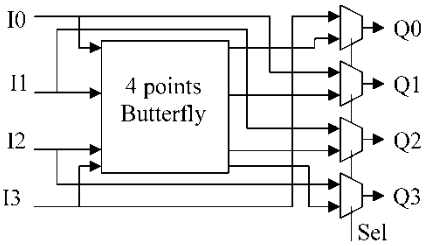

Figure 2 – Internal structure of a Butterfly unit.

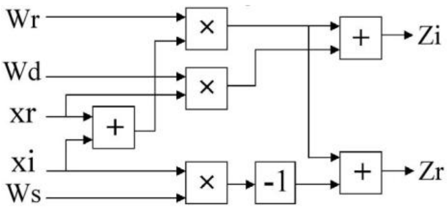

Figure 3 – Internal block diagram of a complex multiplier (3 real adders and 3 real multipliers).

the Sign Magnitude (SM) values of the x_r, x_i, and $x_i + x_r$ are used by the real multipliers. The Two's Complement (TC) to SM converter unit, called *conv*, is used to convert TC input numbers to SM numbers. The initial adder and TC to SM converters are in step-1. In step-2, three real multiplications are performed on the magnitudes only. Each real multiplier, *mult*, also passes its inputs to the outputs.

In step-3, the inputs passed by the *mult* are used to form the sign of the product by using an XOR gate. This sign is used to convert back the results into TC value, by using the *conv* unit. Multiplication of "m" by -1, as shown in Figure 3, is done by enabling/disabling the *conv*, just before the final additions. The trivial multiplication by –j is performed by changing the sign of the real part of input and by exchanging the real and the imaginary parts. For the trivial multiplication by 1, input is passed to output.

The design of a radix-4 FFT processor, Figure 1, using a pipelined complex multiplier, Figure 4, reduces the clock pulse period time. However, the bottleneck still is on the real multiplier of this complex multiplier. To overcome this problem, the real multiplier, *mult*, is also designed in a pipelined manner using the successive additions technique.

In the successive additions, the multiplier adds the partial products to form the final product. The block diagram of a pipelined real multiplier is shown in Figure 5. This structure contains *Partial Product adders*, arrays of AND gates which are called *AAND* units, and registers. A *Partial Product Adder* includes an *AAND* unit and a real adder, Figure 5(b). The *AAND* unit consists of *b* AND gates, Figure 5(c). The parameter *b* is the word length of the processor.

In the proposed multipliers, following each addition, the least significant bit is omitted to keep the word length constant throughout the multiplier.

3.3 Design of the step-3

The step-3 is similar to the step-1 except that each register bank includes only one register. The control signals of the step-1 and the step-3 are also different.

4. THE EVALUATION OF THE DESIGN

4.1 The speed up evaluation

The R4SDF structure presented in [9] has three steps with a nonpipelined complex multiplier, Figure 1. The delay of the *butterfly* unit is *t(B)=t(c)+2×t(a)*, where *t(c)* is the delay of the *conv* unit, and *t(a)* is the delay of the real adder. Since *t(c)≈ t(a)*, the unit delay is defined as *t(U)=t(c)=t(a)*, therefore, *t(B)=3t(U)*. As shown in Figure 3, the delay of the complex multiplier using successive adders in its real multipliers leads to a delay time of *t(CM)=2×t(c)+(b+1)×t(a)=(b+3)×t(U)*. The parameter *b* is the processor's word length. Figure 4 and Figure 5, excluding registers, show the details of this calculation.

In the conventional R4SDF, the complex multiplier in step-2 has the maximum delay time with respect to the other steps. Therefore, the total delay time of the conventional R4SDF processor for *m* inputs, is *t0=(m+15)×t(CM)* or

$$t0=(m+15)(b+3)\times t(U). \qquad (12)$$

The pipelined R4SDF processor proposed in this paper has a pipelined complex multiplier, Figure 4, in which the real multipliers are also pipelined, Figure 5. The maximum delay time of the pipelined complex multiplier is *t(PCM)=2t(U)*, and the delay time of the *butterfly* unit, *3t(U)*, is dominant. Total number of clock cycles for *m* DFT operations is the number of cycles needed in the non-piplined design (*m*+15) in addition to the clock cycles required by the pipelined complex multiplier which is *(b+1)*. The required clock cycles by the complex multiplier consist of the required clock cycles by the real multiplier shown in Figure 5(a), which is equal to *(b-1)*, plus two cycles for the step-1 and step-3 of complex multiplier, as shown in Figure 4. Therefore, the total delay time of the proposed pipelined R4SDF is

$$\begin{aligned} t1 &= (m + 15 + b + 1) \times \text{maximum\_delay\_time} \\ &= (m + b + 16) \times t(B) \qquad (13) \\ &= (m + b + 16) \times 3t(U). \end{aligned}$$

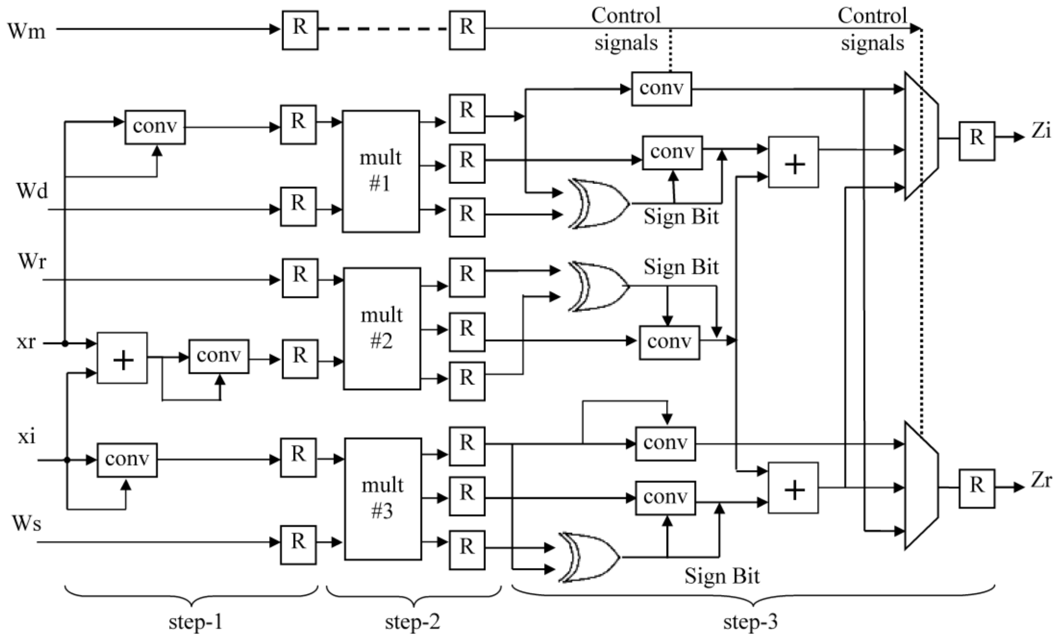

Figure 4 – Block diagram of a pipelined complex multiplier using real multipliers, *mult*s.

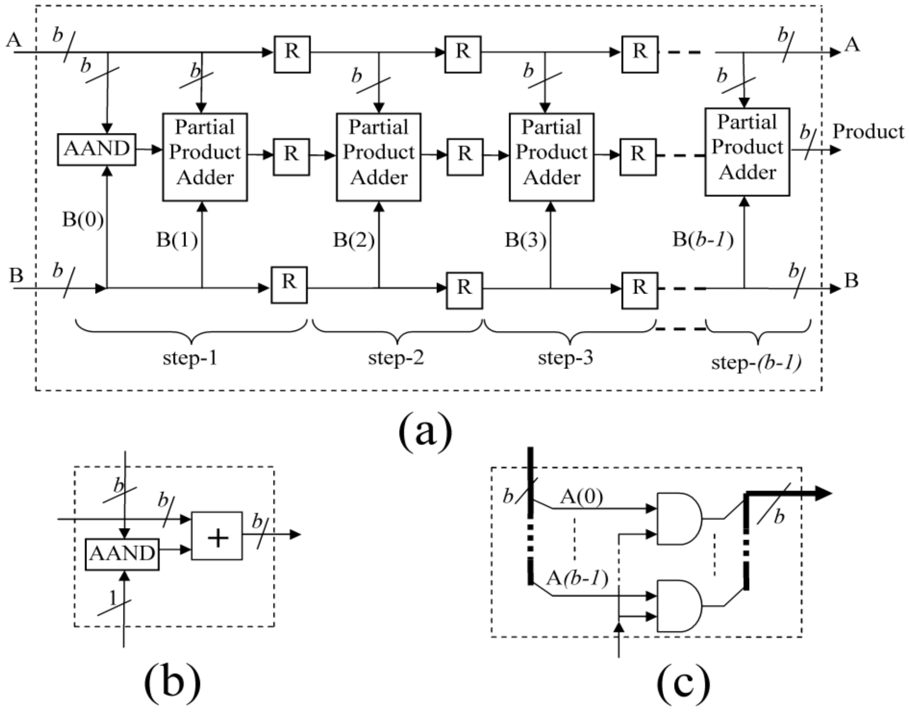

Figure 5 – Block diagram of a pipelined real multiplier, *mult*, using successive additions. (a) Pipelined real multiplier using *Partial Product Adder*s and *AAND* units. (b) Internal structure of *Partial Product Adder*s. (c) Internal structure of an *AAND* unit.

The Speed Up (SU) [14] is defined as

$$SU = \frac{\text{Total delay before pipelining}}{\text{Total delay after pipelining}}. \tag{14}$$

The SU of the proposed design with respect to the conventional design [9] is the ratio of (12) to (13).

$$SU = t0/t1 = \\ ((b+3)\times m + 15b + 45)/(3m + 3b + 48) \tag{15}$$

The speed up with respect to *m* is shown in Figure 6. As indicated by Equation 15, For the large number of inputs, $m >> b$, the SU of the proposed design converges to

$$SU = (b/3) + 1. \qquad (16)$$

SU is proportional to the value of b. However with the increase of b, due to increased delays of Partial Product Adders in the multiplier, shown in Figure 5, the overall working frequency is decreased.

4.2 The precision evaluation

Signal to Error Ratio (SER) is considered as a measure of precision [7]. SER is defined in dB as

$$SER = 10\log\left(\sum_k X_k^2 \Big/ \sum_k e_k^2\right). \qquad (17)$$

Where X_k s are elements of the original Fourier transform and e_k s are elements of the error vector. A 9-bit word length, $b=9$, design is simulated for four types of inputs, including real random test sequences, real sine test sequences with different frequencies, complex random test sequences, and complex sine test sequences with different frequencies. Average SERs resulted from four above situations are 27.29, 31.38, 30.57, and 33.45 dB, respectively.

Althghou the proposed architecture has an speed of (b/3)+1 compared to the conventional design, it has no effect on the SER criterion. The SER of the both designs depends on the word length of intermediate circuitry. Since the word length is unchaged, the SER of the pipelined and the conventional designs are equal.

5. CONCLUSION

In this paper a new pipelined architecture for a 16-point R4SDF processor is presented. This structure sets up a pipelined complex multiplier between the two main steps of the design. The modified design of the complex multiplier [13], which includes three real multipliers and three real adders, is used to reduce the number of the real multipliers. In this paper, these real multipliers are also pipelined in order to achieve better performance out of the processor. The architecture of R4SDF is itself pipelined, but it is further pipelined in this paper and is called a pipelined R4SDF processor.

The Speed Up of the pipelined R4SDF processor with respect to a processor employing a nonpipelined multiplier is *(b/3)+1*. The variable *b* is the word length of the processor. For example, for a 9-bit word length, *b*=9, the proposed pipelined processor is four times faster than the conventional design. The overall Signal to Error Ratio (SER) of the processor for *b*=9 is 30.70 dB.

REFERENCES

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comp.*, vol. 19, pp. 297-301, 1965.

[2] A. Antola, R. Negrini, and N. Scarabottolo, "Arrays for discrete Fourier transform," In Proc. of the European Signal Processing Conference (EURASIP), Amsterdam, Netherlands, 1988.

[3] H. L. Groginsky and G. A. Works, "A pipeline fast Fourier transform," *IEEE Trans. Computers*, vol. C-19, pp. 1015-1019, 1970.

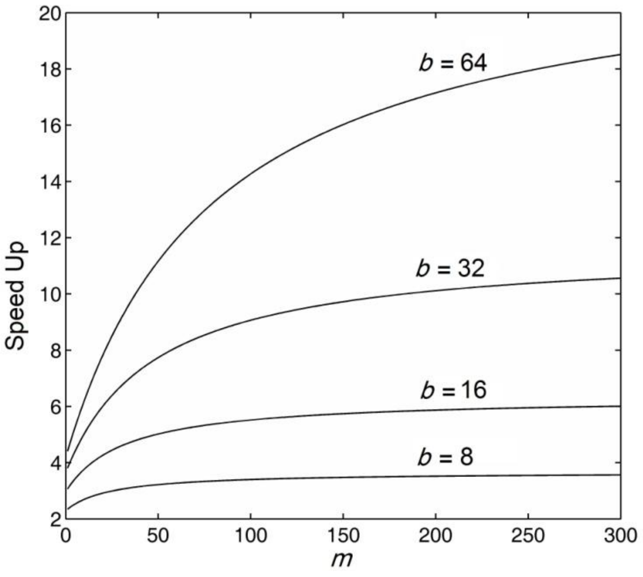

Figure 6 – The Speed Up of the proposed architecture versus number of successive DFT computations, *m*, for different word lengths, *b*.

[4] A. M. Despain, "Fourier transform computer using CORDIC iterations," *IEEE Trans. Computers*, vol. C-23, pp. 993-1001, 1974.

[5] E. H. Wold and A. M. Despain, "Pipeline and parallel-pipeline FFT processors for VLSI implementation," *IEEE Trans. Computers*, vol. C-33, pp. 414-426, 1984.

[6] G. Bi and E. V. Jones, "A pipelined FFT processor for word-sequential data," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1982-1985, 1989.

[7] E. Bidet, D. Castelain, C. Joanblanq, and P. Senn, "A fast single-chip implementation of 8192 complex point FFT," *IEEE J. Solid-State Circuits*, vol. 30, pp. 300-305, 1995.

[8] A. M. El-Khashab and J. Earl E Swartzlander, "An architecture for a radix-4 modular pipeline fast Fourier transform," In *Proc. of the Application-Specific Systems, Architectures, and Processors (ASAP'03)*, 2003.

[9] S. He and M. Torkelson, "A new approach to pipeline FFT processor," *In Proc. of the 10th International Parallel Processing Symposium (IPPS)*, Honolulu, Hawaii, USA, 1996.

[10] Aghaee, N.; Eshghi, M., "A Pipelined Architecture for a 20-point PFA," *TENCON 2006. 2006 IEEE Region 10 Conference*, vol., no., pp.1-4, 14-17 Nov. 2006.

[11] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete time signal processing*. Upper Saddle River, NJ: Prentice-Hall Inc., 1999.

[12] H. K. Garg, *Digital signal processing algorithms*. Boca Raton, Florida: CRC Press LLC, 1998.

[13] W. Li and L. Wanhammar, "A pipeline FFT processor," Signal Processing Systems, 1999. SiPS 99. 1999 IEEE Workshop on, pp.654-662, 1999.

[14] M. M. Mano, *Computer system architecture*, 3rd ed. Englewoods Cliffs, New Jersey: Prentice-Hall Inc., 1993.