A Novel Low Power Approach for Radix-4 commutator FFT Based on CSD Algorithm

[1]BANOTHU DHARMA, [2] O.RAVINDER, [3] B.HANMANTHU
[1,2]*Dept. of ECE, Sree Chaitanya College of Engineering, Karimnagar, T.S. India*
[3]*Dept. of CSE, KITS, Warangal, T.S, India*

[1]dhanakvs2@gmail.com,[2]ravinderoranganti@gmail.com,[3]bhcsekits@gmail.com

Abstract -- This paper proposes a low power pipelined FFT for wireless LAN applications based on canonic signed digit (CSD) algorithm. New techniques and approaches are required at all levels of design abstractions as future technologies are expected to provide unprecedented levels of computations performance in small hands-held units. Since the evolution in battery technology has not yet caught up with the demands in computational requirements, this provides us with a motivation to consider new approaches to reduce computation without compromising the constraints on system performance. This paper proposes several low power approaches for radix-4 single path delay commutator FFT processors which utilize the minimum number of shifters and adders to replace the complex multiplier and low power butterfly architecture. Both power consumption and area is reduced, due to operation substitution compared to a conventional FFT architecture**.**

Keywords*:* CSD, FFT, Hand-held units.

I.INTRODUCTION

The Fast Fourier Transform (FFT) is an important tool used in the Digital Signal Processing (DSP) applications. In recent years, because of the popularity of the signal processing there been a lot of development to increase its performance both at the algorithmic level and the hardware implementation level. On the other hand, developers of the VLSI systems are including features in their design that improves the system performance for applications requiring FFTs. For the portability requirement, the need of low power FFT architecture for telecommunication systems in portable form is attached more and more importance.

Due to the nature of non-stopping processing on the same clock frequency of sampling data, pipelined FFT is preferred especially for a high throughput demand or low power solution [6]. In the pipelined architectures, the commutator and the complex multiplier at each stage contribute a dominating part of the whole power consumption. A number of researchers have explored the scope of low power implementation for FFT processors. In [2], the authors combined voltage over scaling and algorithmic noise-tolerance techniques to reduce power consumption in butterfly blocks. The author of [4] Presented low power cache-memory architecture by using an algorithm offering good data locality to increase speed and energy efficiency. In [5], the authors proposed a new radix algorithms, which can effectively minimize the number of complex multiplications in pipelined FFTs. A novel ordering based low power pipelined radix-4 FFT was presented in [1]. Coefficient ordering reduces the switching activity between successive coefficients fed to the complex multiplier and hence leads to low power consumption [1]. However, as far as we know, until now, there are no explorations on FFTs. In the past, some researchers used shifters and adders to replace the complex multiplication by some special constant coefficients. In [5], the authors used 12 additions to realize the complex multiplication by $\sqrt{2}/2(1\pm i)$. The authors in [7] employed seven shift-and-add units to carry out seven multipliers in parallel, each by a constant coefficient.

This paper explores the application of common subexpression sharing across coefficients to the first stage of 16-point FFTs based on a popular pipelined FFT, R4SDC [8]. Complex multiplications are replaced by the minimum number of shift and addition operations. Hence, both area and power consumption for the multiplier unit are reduced. This new FFT processor architecture also employs a new commutator architecture based on dual port RAMs [9] and improved low power butterfly elements [10].

II. CANONIC SIGNED DIGIT (CSD) ALGORITHM

The number of add operations required in a constant coefficient multiplication equal to one less than the number of nonzero bits in the constant coefficient. In order to further reduce the area and power consumption, the constant coefficient can be encoded such that it contains the fewest number of nonzero bits, which can be accomplished using canonic signed digit (CSD) algorithm. It is common to use the redundancy of signed digit code to replace the conventional multiplier digits, such that addition operations in a multiplication can be reduced with the increase of average shift length across the zeros in the multiplier [12]. Canonical Signed-Digit (CSD) is one widely used signed digit approach. In CSD code of a number, each bit is set to 0, 1 or -1 and no two consecutive bits are nonzero.

The advantage of CSD form is that no value has more than (N+1)/2 nonzero bits, often fewer, and so the multiplication by a constant requires not more than that number of additions for its implementation.

An algorithm for computing the CSD format of a W-bit number is,

```
$\hat{a}_{-1} = 0$
$\gamma_{-1} = 0$
$\hat{a}_W = \hat{a}_{W-1}$
for (i = 0 to W-1)
{
    $\theta_i = \hat{a}_i \text{ (xor) } \hat{a}_{i-1}$
    $\gamma_i = \bar{\gamma}_{i-1} \theta_i$
    $a_i = (1-2\hat{a}_{i+1}) \gamma_i$
}
```

For example, the input is the number 0111011001000001 (constant coefficient, 7641) in two's complement form .Its CSD representation is computed as follows

TABLE1 CSD number representation

i	w	P															0	-1
$\hat{a}_i$	0	0	1	1	1	0	1	1	0	0	1	0	0	0	0	0	1	0
θ_i		1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	1	
γ_i		1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0
Q		1	0	0	0	-1	0	-1	0	0	1	0	0	0	0	0	1	

In the above table1 P=W-1and Q = $(1-2\hat{a}_{i+1}) \gamma_i$.

III. RADIX-4 SINGLE PATH DELAY FFT ALGORITHM

When the real-time signal processing is required pipelined FFT is the suitable option because of its high throughput and low power demands. Radix-4 single delay commutator (R4SDC) architecture is researched in this dissertation. R4SDC is the most popular pipeline FFT architectures, because of its efficient use of butterflies and multipliers. In this a low power technique for the pipeline FFT architecture is discussed.

The Discrete Fourier Transform (DFT) of N complex data points X (n) is defined by

$$x(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad K=0,1,\ldots\ldots N-1; \qquad (1)$$

Where $W_N = e^{-j(2\pi/N)}$,W_N is twiddle factor.

In [8], the authors presented the R4SDC pipelined FFT algorithm for word-sequential data. For radix r_1, equation 1 can be written as follows:

$$x(k) = \sum_{q1=0}^{N1-1} W_N^{q1k} \sum_{p=0}^{r1-1} x(N1p + q1) W_{r1}^{pk} \text{............(2)}$$

The N-point DFT in (2) can be decomposed into v stages where $N = r_1 r_2 \ldots r_v$. The final stage is defined by

$$X(r_1 r_2 \ldots r_{v-1} m_v + r_1 r_2 \ldots r_{v-2} m_{v-1} + \ldots + r_1 m_2 + m1)$$
$$= \sum_{q_v-1=0}^{r_v-1} x_{v-1}(q_{v-1}, m_{v-1}) W_{r_v}^{q_{v-1} m_v} \text{.......................(3)}$$

While intermediate stages are given by the recursive equation 4 below [8]:

$$x_t(q_t, m_t) =$$
$$W_{N_{t-1}}^{q_t m_t} \sum_{p=0}^{r_t-1} x_{t-1}(N_t p + q_t, m_{t-1}) W_{r_t}^{pm_t} \text{............(4)}$$

Where, for both (3) and (4)

$0 \le q_i \le N_i - 1$, $2 \le i \le v$, $N_t = N/(r_1 r_2 \ldots r_v)$, $2 \le t \le v-1$ and $0 \le m_i \le r_i - 1$

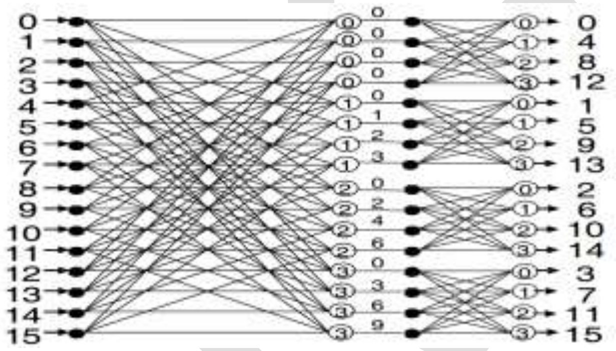

Fig.1. Signal flow graph of a radix-4 16-point FFT

When r1=4, we use16-point FFT whose flow graph based on the above equations can be seen in Figure1. As can be seen, each open cycle denotes a summation while the dots define the stage borders.

The number inside the open circle is the value of m1 (for the first stage) or m2 (for the second stage). The number outside the open circle is the twiddle factor used. N-point pipelined FFT processor based on this architecture shown in Figure 2. It achieves 75% utilization of the complex multiplier and 100% utilization of the butterfly element respectively.

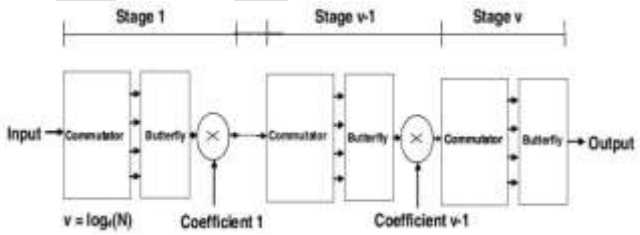

Fig.2.N-point R4SDC pipelined FFT processor architecture [10]

IV. COMMON SUB EXPRESSION SHARING

Common sub-sharing or common sub-expression elimination pre-computes the subexpression among several multiplication-accumulation operations in order to reduce the total number of shift and addition operations [11] which are utilized to carry out multiplications. The approach is very effective for reducing the hardware cost of multiple constants multiplications, especially for the filter-like operation. For example, for a 3-tap FIR filter, the output Y (2) is given as follow.

$$Y(2) = \sum_{i=0}^{2} A_i + x_{n-i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

The weights A_i are the filter coefficients. Suppose the coefficients are given as A0=00111011, A1=00101011, and A2=10110011. The coefficients are represented in two's complement format. According to (5),

$Y(2) = A_0X_2+A_1X_1+A_2X_0$.

Using shifts and additions to replace the multiplications,gives:
$Y(2)=X_2+X_2{<<}1+X_2{<<}3+X_2{<<}4+X_2{<<}5+X_1+X_1{<<}1+X_1{<<}3+X_3{<<}5+X_0+X_0{<<}1+X_0{<<}4+X_0{<<}5-X_0{<<}7$.

The computation requires 12 additions, 1subtraction and 11 shifts. However, if pre-computing $X_{02} = X_0 + X_2$; $X_{12} = X_1+X_2$; $X_{012} = X_{12}+X_0$, the output can be shown as:

$Y(2) = X_{012}+X_{02}{<<}4+X_{012}{<<}1+X_{012}{<<}5+X_{12}{<<}3-X_0{<<}7$. This computation only needs 7 additions, 1 subtraction and 5 shifts. X_{02}, X_{12}, X_{012} are the available common sub expressions for this case. From the above examples, it can be shown that common subexpression sharing can reduce the number of additions and subtractions from 13 to 8 (38% reduction).

V. DESIGN AND IMPLEMENTATION

A. Multiplier-less Units of 16-point R4SDC FFT

In multiplier-less 16-point R4SDC pipelined architecture FFT, the conventional complex multiplier consists of one subtracter, one adder and four real multipliers. The coefficient for the stages can be previously calculated which means from the previous stages.

However, since the complex coefficients for all stages can be pre-computed, we can apply shift and addition operations with common subexpression sharing to those stages where the number of coefficients islimited. For example, the number of coefficients for the first stage of 16-point FFTs is 16. These coefficients are shown in Table2. A close observation of these coefficients reveals that seven of these are (7fff, 0000), one is (0000, 8000) which are the quantized representation for (1, 0) and (0,-1) in 16-bit two's complement format respectively. In each set, the first entry corresponds to the cosine function (the real part, W_r) and the second one corresponds to the sine function (the imaginary part, W_i). For the trivial coefficients (7fff, 0000) and (0000, 8000), the complex multiplication is not necessary. Data can directly pass through the multiplier unit without any multiplication, when data is multiplied with (7fff, 0000). Only an additional unit, which swaps the real and imaginary parts of input data, and inverts the imaginary part, is needed for those data by (0000, 8000).

TABLE II
The coefficients for 16-point R4SDC FFT

Coefficient sequence m1 =0,1	**Original Quantized coefficient**	**Coefficient sequence m1 =2,3**	**Original Quantized coefficient**
W0	7fff,0000	W0	7fff,0000
W0	7fff,0000	W2	5a82,a57d
W0	7fff,0000	W4	0000,8000
W0	7fff,0000	W6	a57d,a57d
W0	7fff,0000	W0	7fff,0000
W1	7641,cf04	W3	30fb,89be
W2	5a82,a57d	W6	a57d,a57d
W3	30fb,89be	W9	89be,30fb

The rest of the coefficients are composed of only 6 constants (7641, 5a82, 30fb, a57d, 89be, cf04). However, one can see that only 3 of these constants (7641, 5a82 and 30fb) would be enough to implement all of the coefficients. For example, a multiplication with the constant a57d could be realized by first multiplying the data with 5a83, and then two's complementing the result. Note that a multiplication by the constant 5a82 already existents. Therefore, the multiplication with the constant 5a83 can simply be obtained by adding the data to the already existing multiplication with 5a82. The other two constants (89be and cf04) can be realized in a similar manner, using constants 7641 and 30fb respectively. 5a82 is represented by two's complement format, 7641 and 30fb are represented by CSD format as follow,

5a82	0101101010000010
7641	1000-10-1001000001
30fb	010-1000100000-10-1

The mixed use of CSD and two's complement is for minimizing the number of addition/shift operations. We can use shifters and adders based on the three constants to carry out those nontrivial complex multiplications as shown below:

$$5a82X = 5X << 12 + 5X << 9 + 65X << 1$$
$$7641X = X << 15 + 65X - 5X << 9$$
$$30fbX = 65X << 8 - X << 12 - 5X$$

In the above 'X' is input data. The common subexpression for the three constants are 101 (5) and 1000001 (65).Figure 3 shows the shift-and-addition module for the three constants in 16-point FFT. The module carries out the multiplications in which the real part (X_r) or imaginary part (X_i) of input data will be multiplied with W_r and W_i respectively.

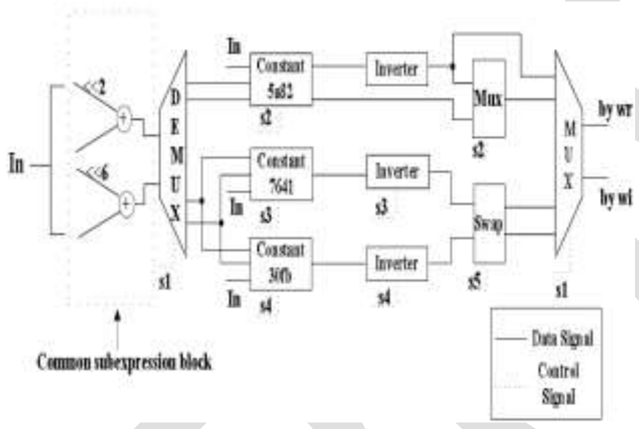

Fig.3. Block diagram of the shift-and-addition module in multiplier-less unit of 16-point R4SDC FFT

The shift-and-addition module is equipped with 5 single-bit control signals s1 - s5. Firstly the input data are fed into the common subexpression block. The signal s1 indicates which constant channels will be chosen for processing the input data. Each channel carries out shift, negation and addition for the constant. The control signal s3 indicates that the constant 7641 block outputs the product either by 7641 or 7642.Similarly, the signals s2 and s4 control the outputs of constant 5a82 and 30fb blocks respectively. The invert units following the constant units either invert the outputs of the constant units or pass them without any change. The swap unit provides the appropriate swapping for input data, depending on whether the coefficient is (30fb, 7641) 0r (7641, 30fb). The demultiplexer unit judges which couple of products are final outputs. Totally, 11 adders are used to compose the shift-and-addition module.

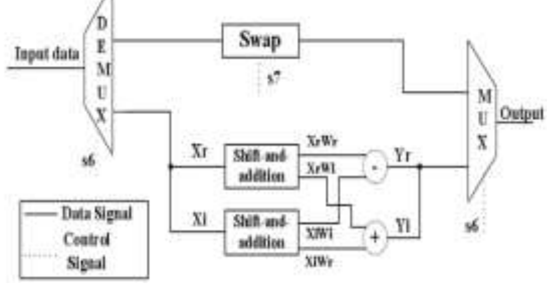

Fig.4. Block diagram of the multiplier-less unit in 16-point R4SDC FFT

Based on the above discussion, the complex multiplication unit in 16-point radix-4 pipelined FFT can be substituted by a multiplier-less unit. The block diagram of the unit is depicted in Figure 4. Only those data, which multiply nontrivial complex coefficients, are fed into the shift-and-addition units. Two shift-and-addition units are needed for the real part (X_r) and imaginary part (X_i) respectively. There are two single-bit control signals s6 and s7 in the multiplier-less unit. The signal s6 indicates that whether the input data is corresponding to a nontrivial complex coefficient or not. When the signal s7 is asserted to logic 1 state, the real and imaginary parts of the input data are swapped, and the imaginary part is inverted. Here, in the multiplier-less unit, 22 adders are used to substitute the four real multipliers in the complex multiplier unit. Due to the use of multiplier-less unit, the Rom unit storing the coefficient will be replaced by a FSM unit generating control signals (s1 - s7).

B. Commutator Based On Dual Port RAMs For R4SDC

The commutator unit is one of the main power-consuming components in R4SDC FFT. Previous approaches to implement commutators include shift register architecture (SR) [8], conventional dual port RAM architecture (DR) [13], and triple port RAM architecture (TR) [13]. These architectures are based on the same interconnection topology among different FIFO elements [13]. In [9], a new architecture based on dual port RAMs, termed as IDR. IDR exploits a new interconnection topology among dual port RAM blocks [3]. IDR efficiently reduces the switching activity through maintaining the unused outputs of RAMs at their previous values [9]. IDR also reduces the number of write operations to memory blocks [3]. With IDR architecture each RAM block is enabled 5/3 times on average, during the four periods. Whereas, for DR and TR, each RAM block is enabled 4 and 10/3 times respectively [13]. Hence, IDR is significantly more power efficient than both DR and TR.

C. Improved Butterfly Architecture

In [10], low power butterfly architecture was presented. Two 4-input summation blocks were employed to replace six adder/subtracts. However, since inversions were implemented based on one’s complementing (and not two’s complementing), this architecture introduced a small error in the butterfly operations. In R4SDC FFT, the butterfly element performs the summations of Equations 4 and 5. The conventional butterfly architecture consists of 6 adders/ subtracts. We improve this architecture by eliminating this error. Figure 5 shows the improved low power butterfly architecture. Six inverters (CI1 to CI6) are used to generate the normal or the one’s complement form under the control of C5, C6 and C7. The signal C4 controls the four multiplexers (M1 to M4) for directing appropriate data to the inputs of the summation blocks. Two 5-input summation blocks (SUM0 and SUM1) are employed to generate the real and imaginary parts of the output respectively. An additional decoder unit is used to generate compensation for eliminating the error due to the one’ complement based inversion.

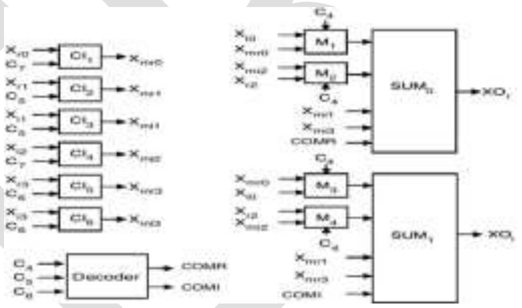

Fig.5. Block diagram of the improved butterfly architecture

VI. RESULTS AND DISCUSSIONS

The proposed architectures have been implemented in Verilog HDL for 16-point multiplier-less R4SDC pipelined FFT and simulated in Xilinx 9.2i (ISE Simulator). Input data used are 32 bits complex data. The 16-point R4SDC FFT was synthesized at 10ns clock cycle, using Synopsys Design Compiler targeting the UMC 0.18µ CMOS library. Power evaluations were carried out, using Synopsys Design Power, at 8ns clock cycles for 16-point FFT respectively. Less multiplier FFT architecture employs CSD multiplier unit to carry out the complex multiplication operations shown in fig.5. The input to FFT module is sequential input data will be separated into 4 parallel output data streams by the commutator, these parallel data summed into the butterfly element and then CSD multiplier performs the multiplications of complex input data with all the synchronized constant coefficients and generates real and imaginary outputs. The clock and reset are global signals.

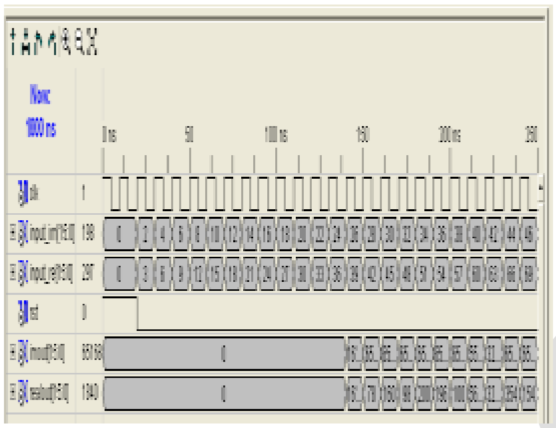

Fig.6.Simulation waveform for multiplier less FFT

The comparative power and area results are shown in following table.

TABLE III
Comparative results of Power and Area

Instance	Cells	Dynamic Power (mW)	Total Power (mW)	Cell Area
Conventional FFT	4692	29.083	29.084	181209
CSD FFT	3528	20.588	20.589	141069

Clearly, for 16-point FFT the best power saving of 50.2% is achieved. The multiplier-fewer units in 16-point FFT power reduction, as compared to the complex multiplier based on non-Booth coded Wallace tree. All designs have the commensurate area with each other. The application of the new techniques brings a slight reduction in the cost of area and power consumption.

VII. CONCLUSION

This paper presents a low power pipelined R4SDC FFT processor architecture based on canonic signed digit multiplier suitable for shorter FFTs. This design approach can also be applied for the last stages of longer FFTs. The multiplier-less architecture employs the minimum number of shift and addition operations to realize the complex multiplications. This reduces the power consumption of the multiplier by using the architectures such as low power butterfly and commutator scheme as compared to conventional non-Booth coded Wallace tree multiplier based R4SDC FFT architecture.

REFERENCES:

[1] M. Hasan, T. Arslan, and J.S. Thompson,"A novel coefficient ordering based low power pipelined radix-4 FFT processor for wireless LAN application", IEEE Transaction on Consumer Electronics, Vol. 49, no.1, pp. 128-134, FEB. 2003.

[2] S.R. Sridhara and N.R. Shanbhag "Low-power FFT via reduced precision redundancy" Signal Processing Systems, 2001 IEEE Workshop pp. 117-124 Sep. 2001.

[3] Wei Han, T. Arslan, A. T. Erdogan and M. Hasan "A novel low power pipelined FFT based on sub expression sharing for wireless LAN applications. School of Engineering and Electronics University of Edinburgh, Edinburgh UK, pp. 83-88, 2004.

[4] M.B. Bevan, "A Low-Power, High-Performance, 1024-point FFT Processor", IEEE Journal of Solid-State Circuit, Vol. 34, no. 3, pp. 308-387, March 1999.

[5] L. Jia, Y. Gao, J. Isoaho and H. Tenhunen, "A new VLSI oriented FFT algorithm and implementation", in Proceedings of Eleventh annual IEEE International ASIC conference, pp. 337-341, 1998.

[6] S. He and M. Torkelson, "Design and implementation of 1024-point pipeline FFT processor" Custom Integrated Circuits Conference, 1998. Processing of the IEEE 1998, 11-14 May 1998. pp. 131 – 134.

[7] K. Maharatna, E. Grass and U. Jagdhold "A 64-Point Fourier Transform Chip for High-Speed Wireless LAN Application Using OFDM" IEEE Journal of Solid-State Circuit. Vol. 39, No. 3, March 2004.

[8] G. Bi and E. V. Jones, "A pipelined FFT processor for word-sequential data", IEEE Transactions on acoustic, speech, and signal processing, Vol. 37 , no: 12 , Dec. 1989, pp.1982- 1985.

[9] W Han, T. Arslan, A.T. Erdoga, and M. Hasan "A Low power Dual Port RAM Based Commutator for A Pipelined FFT Processor" submitted to 18th International Conference on VLSI Design.

[10] M. Hasan Low Power Techniques and Architectures for Multicarrier Wireless Receivers PhD thesis, University of Edinburgh, 2003

[11] T.S. Chang, J.I. Guo and C.W. Jen, "Hardware-Efficient DFT Designs with Cyclic Convolution and Subexpression Sharing" IEEE Transaction on Circuit and Systems Vol.47, No. 9, Sep. 2000.

[12] Kai Hwang Computer Arithmetic: principles, architecture, and design John Wiley & Sons, Inc. 1979 pp. 149-151

[13] Mohd. Hasan and Tughrul Arslan, "A triple port RAM based low power commutator architecture for a pipelined FFT processor" Circuits and Systems, 2003. ISCA '03. Proceedings of the 2003 International Symposium, Vol. 5, on 25-28 May 2003, pp. V-353 - V-356

[14] S. Hong, S. Kim, M. C. Papefthymiou, and W. E. Stark, "Power-complexity analysis of pipelined VLSI FFT architectures for low energy wireless communication applications," in Circuits and Systems, 1999. 42nd Midwest Symposium, Aug. 1999, vol. 1, pp. 8–11