Hardware Implementation Low Power High Speed FFT Core

Muniandi Kannan and Srinivasa Srivatsa
Department of Electronics Engineering, Chennai, India

Abstract: *In recent times, DSP algorithms have received increased attention due to rapid advancements in multimedia computing and high-speed wired and wireless communications. In response to these advances, the search for novel implementations of arithmetic-intensive circuitry has intensified. For the portability requirement in telecommunication systems, there is a need for low power hardware implementation of fast fourier transforms algorithm. This paper proposes the hardware implementation of low power multiplier-less radix-4 single–path delay commutator pipelined fast fourier transform processor architecture of sizes 16, 64 and 256 points. The multiplier-less architecture uses common sub-expression sharing to replace complex multiplications with simpler shift and add operations. By combining a new commutator architecture and low power butterfly architecture with this approach power reduction is achieved. When compared with a conventional fast fourier transform architecture based on non-booth coded wallace tree multiplier the power reduction in this implementation is 44% and 60% for 64-point and 16-point radix-4 fast fourier transforms respectively. The power dissipation is estimated using cadence RTL compiler. The operating frequencies are 166 MHz and 200 MHz, for 64 point and 16 point fast fourier transforms, respectively. Our implementation of the 256 point FFT architecture consumes 153 mw for an operating speed of 125 MHz.*

Keywords: *Pipelined architecture, shift register, finite state machine, common sub-expression, multiplier-less architecture.*

Received March 15, 2007; accepted June 15, 2007

1. Introduction

Fast Fourier Transforms (FFT) is the fast implementation of the Discrete Fourier Transform (DFT) which relies on mathematical simplification and classification of the input sequence to achieve their performance gain. The FFT typically requires $O(N \log_2 N)$ operations to complete in comparison to the straight DFT requiring $O(N^2)$ operations [2].

The FFT processor is used in a wide range of DSP and communication applications, such as radar signal processing and wireless LAN. Recent research work has demonstrated the pipelined FFT as a leading architecture for real time applications. In this paper, a low power and efficient multiplier-less approach is employed to substitute complex multipliers in pipelined FFTs the commutator is needed to reorder the input data. It is well known that the switching power is mainly responsible for power consumption in CMOS circuits. This power, P_{sw}, is given by

$$p_{sw} = \frac{1}{2} k c_{load} v_{dd}^2 f \tag{1}$$

where k is the average number of times the gate makes an active transition during one clock cycle, f is the clock frequency, V_{dd} is the supply voltage and C_{load} is the load capacitance of the gate. Hence, for achieving low power, one or more of the parameters C_{load}, V_{dd} and k need to be minimized. However, since C_{load} and V_{dd} are relative to the target technology, k becomes the main point of improvement.

1.1. Radix-4 Pipelined FFT

The DFT of N complex data points x(n) is defined by

$$X(k) = \sum_{n=o}^{N-1} x(n) W_N^{nk}, k = 0,\ 1,\ 2...\ N\text{-}1 \tag{2}$$

where $W_N = e^{-j\frac{2\pi}{N}}$ is twiddle factor [10]. Since 16 and 64 is a power of four, radix-4 decimation-in-frequency algorithm is used to break the DFT formula into four smaller DFTs. The FFT is the speed-up algorithm of DFT [7]. The final sets of transforms look like

$$X(4k)=\sum_{n=0}^{N/4-1}[x(n)+x(n+\frac{N}{4})+x(n+\frac{N}{2})+x(n+\frac{3N}{4})]W_N^0 W_{N/4}^{kn} \tag{3}$$

$$X(4k+1)=\sum_{n=0}^{N/4-1}[x(n)-jx(n+\frac{N}{4})-x(n+\frac{N}{2})+jx(n+\frac{3N}{4})]W_N^n W_{N/4}^{kn} \tag{4}$$

$$X(4k+2)=\sum_{n=0}^{N/4-1}[x(n)-x(n+\frac{N}{4})+x(n+\frac{N}{2})-x(n+\frac{3N}{4})]W_N^{2n} W_{N/4}^{kn} \tag{5}$$

$$X(4k+3)=\sum_{n=0}^{N/4-1}[x(n)+jx(n+\frac{N}{4})-x(n+\frac{N}{2})-jx(n+\frac{3N}{4})]W_N^{3n} W_{N/4}^{kn} \tag{6}$$

For 16-point FFT for k=0, 1, 2, 3 we get 16 equations and 64-point FFT k varies from 0 to 15. The flow graph of 16-point FFT is seen in Figure 1. In this

Figure the numbers inside the open circle represent equations which are used for computing the output in the butterfly stage.

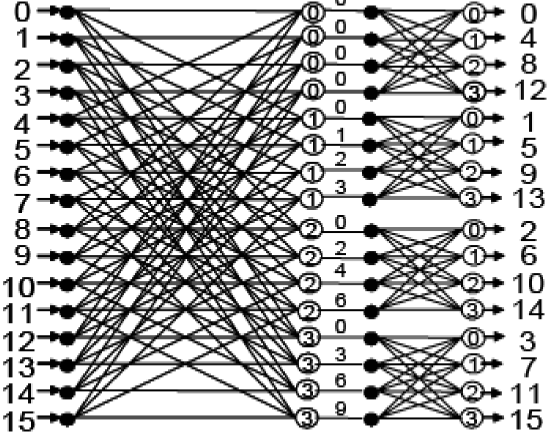


Figure 1. Signal flow graph of a radix-4 16-point FFT (DIF algorithm).

The number outside the open circle is the twiddle factor used. The 4 outputs from two commutators are fed into each simplified butterfly unit. The butterfly unit computes the four equations in a clock cycle. Coefficients are fed in to complex multipliers, respectively.

A pipelined *N*-point radix-4 FFT processor based on this architecture [6], shown in Figure 3, has $\log_4 N$ stages. Each stage produces one output within each word cycle. Each stage contains a commutator, a butterfly element and a complex multiplier. The sequential outputs at each stage must be ordered in accordance with the value of *mt*. For instance, from Figure 2 at stage 1, the outputs associated with *mt* = 0 are produced in the first four word cycles, then those associated with *mt* = 1 in the next four cycles and so on. It is clear from FFT equation that input data for each summation at *stage t* are separated in time by N_t *words*.

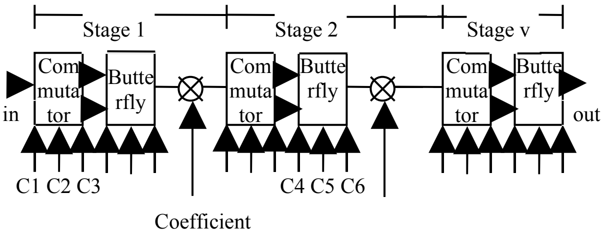


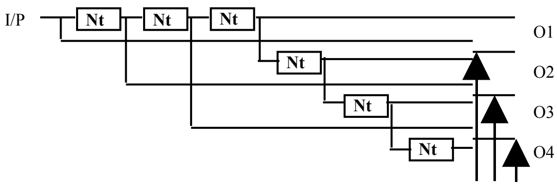
Figure 2. *N*-point radix-4 pipelined FFT processor architecture.

2. Implementation

2.1. Commutator

In realtime applications, input data is a sequential stream. Therefore, it does not match the FFT algorithm since the FFT requires temporal re-ordering of data. For this reason, the commutator is needed to reorder the input data. Among several pipelined FFT architectures, Radix-4 Single-path Delay Commutator (R4SDC) [4] is widely used, owing to its high utilization of multipliers, butterfly elements and memory blocks. The commutators will take up more proportion of the overall power consumption and act as a leading actor with the increase of FFT size. Therefore, reducing the power consumption of the communator units is crucial for the low power implementation of pipelined FFT processor.

The requisite commutator is shown in Figure 2 (this is required for both real and imaginary parts). It consists of six shift registers each providing N_t word delays. Control signals (denoted c1, *c2,* and c3) select the appropriate data via *2:*1 multiplexers. In accordance with the value of *mt,* the four complex outputs from the commutator are connected to its associated butterfly. The commuator supplies the same set of data for N_t *word* cycles. Each FIFO is implemented through a set of shift registers. The FIFO size N_t equals $4^{(5-t)}$, where *t* is the stage number.



m1	c1	c2	c3
0	1	1	1
1	0	1	1
2	0	0	1
3	0	0	0

Figure 3. Commutator for stage *t*.

2.2. Low Power Butterfly

The butterfly operation is the heart of the FFT algorithm. It takes data words from memory and computes the FFT. Low power Butterfly (LB) architecture is employed to replace the conventional butterfly based on adder/subtracters. For radix-4, the complex multiplications within the sum can be replaced by the combination of addition, subtractions, and swapping between the real and imaginary parts, as shown in Figure 4 Three complex adder/ subtractors (each comprising a real and imaginary element) are used instead of eight complex adders [1]. Control signals again select data and functions in accordance with the value of $m_{t.}$ The butterfly element produces *N* outputs consecutively over *N* word cycles in contrast to conventional configurations leaving *3 N/4* word cycles unused. Thus, only one complex multiplier is needed for the twiddle rotation at each stage instead of three in other designs of butterfly realizations.

2.3. Multiplier-Less Unit

Minimization of silicon area is achieved by reducing the number of functional units (such as adders and multipliers), multiplexers, and interconnection wires. Conventional complex multiplier consists of four real multipliers, one adder and one subtracter. Shift and addition operations with common sub expression sharing are used to pre-compute twiddle coefficients which reduces area as well as power [5].

The number of coefficients for the 16-point FFT is shown in Table 1 The multiplier-less unit as shown in Figure 7 consist of shift and addition operations with common sub expression sharing to replace complex multiplications [3]. A close observation reveals that the seven coefficients (7fff, 0000) and (0000, 8000) are the trivial coefficients 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, *Wr*) and the second one corresponds to the sine function (the imaginary part, *Wi*). 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 (0000, 8000). The rest of the coefficients can be represented by three constants (7641, 5a82 and 30fb). For example, a multiplication with the constant a57d could be realized by first multiplying the data with 5a82, and then two's complementing the result. The other two constants (89be and cf04) can be realized in a similar manner, using constants 7641 and 30fb, respectively.

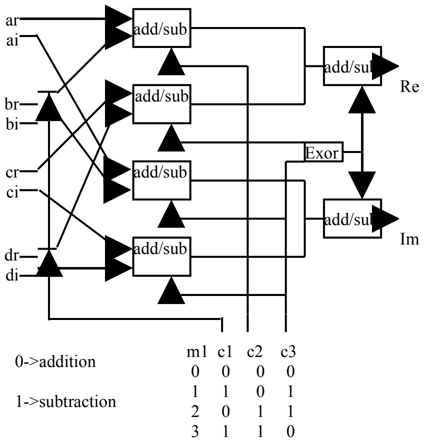


Figure 4. Butterfly element for stage for stage t.

Table 1. The coefficients for 16-point.

Coefficient Sequence m1 = 0,1	Original quantized coefficient	Coefficient sequence m1 = 2,3	Original quantized coefficient
W_o	7fff, 0000	W_o	7fff, 0000
W_o	7fff, 0000	W_2	5a82, a57d
W_o	7fff, 0000	W_4	0000, 000
W_o	7fff, 0000	W_6	a57d, a57d
W_o	7fff, 0000	W_o	7fff, 0000
W_1	7641, cf04	W_3	30fb, 89be
W_2	5a82, a57d	W_6	a57d, a57d
W_3	30fb, 89be	W_9	89be, 30fb

5a82 is represented in two's complement format, 7641 and 30fb are represented in Canonical Signed-Digit (CSD) format: 5a82 (0101101010000010), 7641 (1000-10-001000001) and 30fb (010-1000100000-10-1). We can use shifters and adders based on the three constants to carry out those nontrivial complex multiplications as shown below:

$$5a82X = 5X \ll 12 + 5X \ll 9 + 65X \ll 1$$

$$7641X = X \ll 15 + 65X - 5X \ll 9$$

$$30fbX = 65X \ll 8 - X \ll 12 - 5X$$

The common sub expressions for the two constants are 101 (5) and 1000001 (65). Figure 5 shows the shift-and-addition module for the three constants in the multiplier-less unit. ROM unit storing coefficients is replaced by a FSM unit generating control signals (s1- s8) in multiplier-less approach. The same multiplier architecture is applied to 64-point FFT is shown in Figure 6 All coefficients for 64-point FFT is represented interms of 7f62, 7d8a, a7a7d, 7641, 70c2, 6a6d, 62f2, 5a82, 5133, 471c, 3c56, 30fb, 2528, 18f8, 0c8b [8]. The following coefficients are pre-computed using common sub expression based shift and addition.

$$7f62X = X \ll 15 - 5X \ll 5 + X \ll 1$$
$$7d8aX = X \ll 15 - 5X \ll 7 + 5X \ll 1$$
$$7a7dX = 65X \ll 9 - X \ll 11 + X \ll 7 - X \ll 2 + X$$

$$7641X = X \ll 15 + 65X - 5X \ll 9$$

$$70c2X = X \ll 1 - X \ll 5 + X \ll 8 - X \ll 12 + X \ll 15$$
$$6a6dX = X - 5X \ll 2 + X \ll 7 + 65X \ll 9 - X \ll 13$$
$$62f2X = X \ll 1 + X \ll 10 + X \ll 15 - X \ll 4 - X \ll 8 - X \ll 13$$

$$5a82X = 5X \ll 12 + 5X \ll 9 + 65X \ll 1$$

$$5133X = 5X \ll 12 + 65X \ll 2 - X \ll 4 + X \ll 6$$
$$471cX = X \ll 5 - 65X \ll 2 + X \ll 11 + X \ll 14$$
$$3c56X = X \ll 7 - 5X \ll 1 - X \ll 7 - X \ll 10 + X \ll 14$$

$$30fbX = 65X \ll 8 - X \ll 12 - 5X$$

$$2528X = 5X \ll 3 + 5X \ll 8 + X \ll 13$$
$$18f8X = X \ll 8 - X \ll 3 - X \ll 11 + X \ll 13$$
$$0c8bX = X \ll 11 - 5X - 65X \ll 4 + X \ll 7$$

Similarly, coefficients for *256-point FFT is* represented interms *of 7ff6, 7fd8, 7fa7, 7f62, 7f09, 7e9d, 7e1d, 7d8a, 7ce2, 7c29, 7b5d, 7a7d, 798a, 7884, 776c, 7641, 7504, 73b5, 7255, 70e2, 6f5f, 6dca, 6c24, 6a6d, 68a6, 66cf, 64e8, 62f2, 60ec, 5ed7, 5cb4, 5a82, 5842, 55f5, 539b, 5133, 4ebf, 4c3f, 49b4, 471c, 447a, 41ce, 3f17, 3c56, 398c, 36ba, 33de, 30fb, 2e11, 2b1f, 282b, 2528, 2223, 1f19, 1c0b, 18f8, 15e2, 12c8, 0fab, 0c8b, 096a, 0647, 0324.* The coefficients are pre-computed using common sub expression based shift and addition.

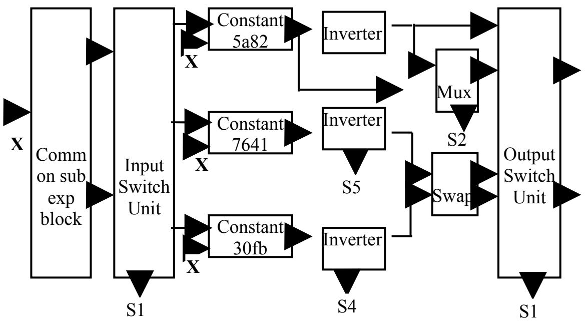


Figure 5. Block diagram of shift-and-addition module.

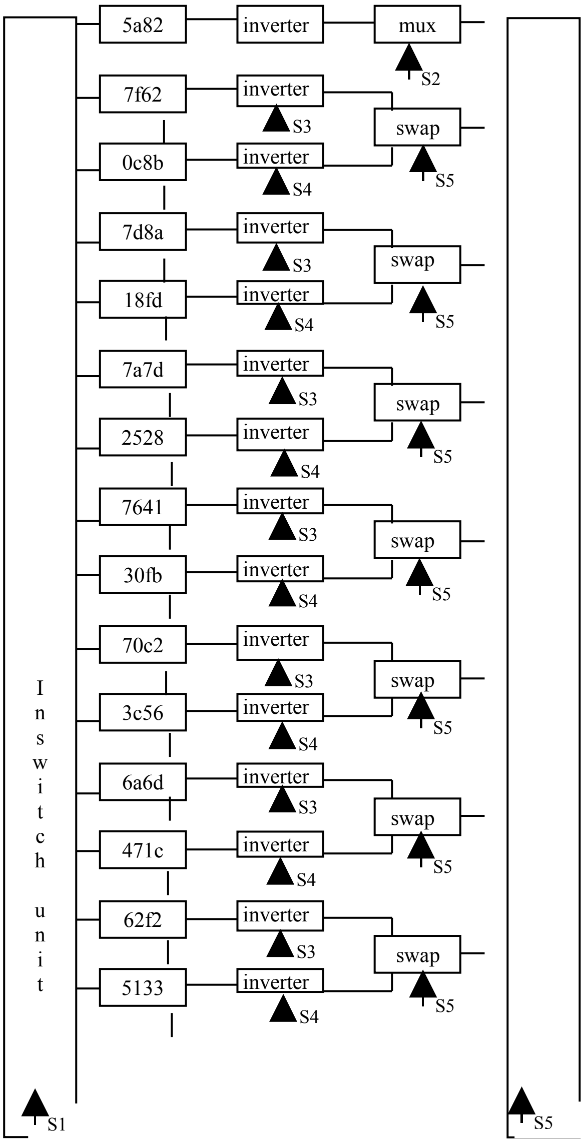


Figure 6. Block diagram of shift-and-addition module (64-point).

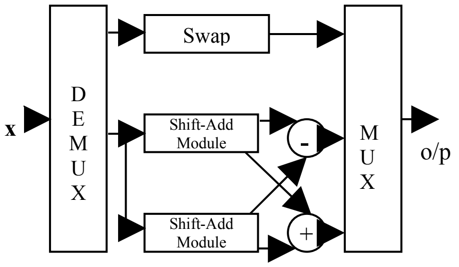


Figure 7. Block diagram of the multiplier-less unit.

3. Results

3.1. Simulation Results Using Modelsim Tool

The FFT blocks are simulated and the results are shown below using Modelsim Tool in Verilog HDL. The resulting Verilog HDL simulation models can then be used as building blocks in larger circuits (using schematics, block diagrams or system-level Verilog HDL descriptions) for the purpose of simulation. The top module is simulated for 32 bits (complex data) fixed point using 16 point, 64-point and 256-point radix-4 DIF FFT algorithm. The given inputs and corresponding outputs are as follows:

Input: *1, 1 ,1 ,1, 2 ,2 ,2 ,2 ,1 ,1 ,1 ,1 , 0 , 0 , 0 , 0,*
1, 1 ,1 ,1, 2 ,2 ,2 ,2 ,1 ,1 ,1 ,1 , 0 , 0 , 0 , 0,
1, 1 ,1 ,1, 2 ,2 ,2 ,2 ,1 ,1 ,1 ,1 , 0 , 0 , 0 , 0,
1, 1 ,1 ,1, 2 ,2 ,2 ,2 ,1 ,1 ,1 ,1 , 0 , 0 , 0 , 0

Output: *64, 0, 0, 0, -16.109-24.109i, 0, 0, 0, 0, 0, 0,*
0, 9.9864-1.9864i,0, 0, 0, 0, 0, 0, 0, 0, 0,
1.3273-6.6727i , 0, 0, 0, 0, 0, 0, 0,
4.7956+3.2044i ,0, 0, 0, 0, 0, 0, 0,
4.7956-3.2044i, 0, 0, 0, 0, 0, 0, 0,
1.3273+6.6727i, 0, 0, 0, 0, 0, 0, 0,
9.9864+1.9864i, 0, 0, 0, 0, 0, 0, 0,
16.109+24.109i, 0, 0, 0.

3.2. Synthesis Results

The proposed FFT architecture has been synthesized for 16-point, 64-point and 256-point using cadence RTL Compiler targeting the TSMC 0.18μ CMOS technology library.

3.3. Reports

RTL Compiler was used to evaluate power, area and timing report for FFTs. The Timing and power report for 16-point, 64-point and 256-point FFT core is shown in Table 2 and 3 The power and area report of different modules present in top FFT core for 16-point and 64-point FFT is shown in Tables 4 and 5 For 256-point FFT above reports are given in Tables 6 and 7.

Table 2. Timing report for FFT core (different points).

FFT Size	Frequency(MHz)
16-point	200
64-point	166.66
256-point	125

Table 3. Power report for FFT core.

Attributes	16-point	64-point	256-point
Leakage(mw)	0.0012	0.0015	0.0038
Internal (mw)	11.106	21.372	69.153
Net (mw)	1.779	2.555	7.590
Switching (mw)	12.885	23.927	76.744
Total (m w)	25.772	47.856	153.49
Area(mm^2)	0.2112	0.3216	0.8417

Table 4. Power report for 16-point FFT core (different modules).

Attribute	Comm (I)	Comm (II)	Multi-plier	Butter fly
Leakage (µw)	0.31	0.09	0.3	0.25
Internal (mw)	3.978	0.993	3.5793	3.4008
Net (mw)	0.430	0.128	0.9848	0.9624
Switching (mw)	4.408	1.122	4.5641	4.3632
Total (m w)	8.817	2.444	9.128	8.726
Cell Area (mm^2)	0.069	0.018	0.0442	0.0420

Table 5. Power report for 64-point FFT core (different , modules).

Attribute	Comm (I)	Comm (II)	Comm (III)	Multi-plier	Butter fly
Leakage (µw)	1.14	0.31	0.09	0.97	0.25
Internal (mw)	15.303	3.9781	0.9932	6.676	3.40
Net (mw)	1.572	0.4303	0.1287	1.889	0.96
Switching (mw)	16.875	4.4083	1.122	8.566	4.36
Total (mw)	33.751	8.817	2.444	17.13	8.72
Cell Area (mm^2)	0.2606	0.0690	0.0185	0.159	0.04

Table 6. Power report for 256-point FFT core (different modules).

Attribute	Comm (I)	Comm (II)	Comm (III)	Comm (IV)
Leakage (µw)	4.4	1.14	0.31	0.09
Internal (mw)	60.601	15.303	3.9781	0.9932
Net (mw)	6.140	1.572	0.4303	0.1287
Switching (mw)	66.742	16.875	4.4083	1.122
Total (mw)	133.487	33.751	8.817	2.444
Cell Area (mm^2)	1.0270	0.2606	0.0690	0.0185

Table 7. Power report for 256-point FFT core (different modules).

Attribute	Multiplier	Butterfly
Leakage (µw)	1.5	0.25
Internal (mw)	21.859	3.4008
Net (mw)	5.276	0.9624
Switching (mw)	27.135	4.3632
Total (m w)	54.271	8.726
Cell Area (mm^2)	0.044224	0.0420

4. Conclusion

In this paper a pipelined architecture for 16 point, 64-point and 256-point radix-4 DIF FFT in fixed point representation is implemented. Low power FFT processor is implemented by using multiplier less (shift add) approach for multiplying twiddle coefficient.

This paper presents a multiplier-less pipelined FFT processor architecture suitable for shorter FFTs. This design approach can also be applied to the longer FFTs. The multiplier-less architecture employs the minimum number of shift and addition operations to realize the complex multiplications. By combining a commutator architecture and low power butterfly architecture with this approach, the resulting power savings are around 43% and 59% for 64-point and 16-point radix-4 FFTs, respectively, as compared to a conventional FFT architecture based on non-booth coded wallace tree multiplier. The parameterization impact on power /speed performance has been compared.

References

[1] Bi G. and Jones E., "A Pipelined FFT Processor for Word Sequential Data," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 1982-1985, 1989.

[2] Cooley J. and Tukey J., "An Algorithm for the Machine Computation of the Complex Fourier Series," *Mathematics of Computation*, vol. 19, pp. 297-301, 1965.

[3] Han W., Arslan T., Erdogan A., and Hasan M., "A Novel Low Power Pipelined FFT Based on Sub Expression Sharing for Wireless LAN Applications," *IEEE Workshop on Signal Processing Systems,* pp. 83-88, 2004.

[4] Han W., Arslan T., Erdogan A., and Hasan M., "Low Power Commutator for Pipelined FFT Processors," *IEEE International Symposium on Circuits and Systems,* vol. 5, pp. 5274-5277, 2005.

[5] Han W., Arslan T., Erdogan A., and Hasan M., "Multiplier-Less Based Parallel-Pipelined FFT Architecture for Wireless Communication Applications," *IEEE Proceedings on Acoustics, Speech and Signal Processing*, vol. 5, pp. 45-48, 2005.

[6] Han W., Arslan T., Erdogan A., and Hasan M., "The Development of High Performance FFT IP Cores Through Hybrid Low Power Algorithmic Methodology," *in Proceedings of the Asia South Pacific Design Automation*, pp. 549-552, China, 2005.

[7] John G. and Manolakis D., *Digital Signal Processing*, MacMillian, London, 1988.

[8] Maharatna K., Grass.E., and Jagdhold U., "A 64-Point Fourier Transform Chip for High-Speed Wireless LAN Application Using OFDM," *IEEE Journal of Solid-State Circuit*s, vol. 39, no. 3, pp. 484-493, 2004

[9] Mohd H. and Tughrul A., "A Triple Port RAM Based Low Power Commutator Architecture for a Pipelined FFT Processor," *in Proceedings of the 2003 International Symposium of Circuits and Systems(ISCAS'03),* vol. 5, pp. 353-356, 2003.

[10] Rabiner L. and Gold B., *Theory and Application of Digital Signal Processing,* Prentice Hall, 1975.

Muniandi Kannan received his BE in electronics and communication engineering from MK University, Madurai, and his ME from Anna University, Chennai. Since 1993 he has been working in Anna University, Chennai, India. His area of interests includes computer architecture, VLSI design, and VLSI for signal processing.

Srinivasa Srivatsa received his BE in electronics and telecommunication engineering from Jadavpur University, his ME in electrical communication engineering, and his PhD from Indian Institute of Science, Bangalore, India. He had been a professor of electronics engineering in Anna University, Chennai, India for nearly 20 years. He is the author of 191 publications in reputed journals/conference proceedings. His area of interests includes computer networks, digital logic design, and design of algorithms and robotics.