

A Novel Coefficient Ordering based Low Power Pipelined Radix-4 FFT Processor for Wireless LAN Applications

M. Hasan, T. Arslan and J.S. Thompson

Abstract — *The FFT processor is a critical block in all multi-carrier systems used primarily in the mobile environment. The portability requirement of these systems is mainly responsible for the need of low power FFT architectures. This paper proposes a technique to reduce the power consumption of a popular low power radix-4 pipelined FFT processor by modifying its operation sequence. The complex multiplier is one of the most power consuming blocks in the FFT processor. The switching activity at its fixed coefficient input can be drastically reduced by coefficient ordering and hence its power consumption. Coefficient ordering requires a novel commutator architecture which can handle the corresponding data sequencing as per new coefficient ordering. The resulting power saving is around 23% and 9% for the 16-point and 64-point radix-4 pipelined FFT processor respectively. This approach is very attractive for orthogonal frequency division multiplexing (OFDM) based wireless LAN (IEEE 802.11) requiring short FFTs but it can also be applied to the penultimate stage of longer FFTs used in Digital audio and video broadcasting¹.*

Index Terms — Coefficient ordering, Low power, Pipelined FFT, Switching activity.

I. INTRODUCTION

One of the fastest growing areas in the computing industry is the provision of high throughput DSP and Telecommunication systems in portable forms. With the advent of SoC (System on chip) technology, DSP algorithms such as Fast Fourier Transform (FFT) are being prototyped as parameterisable cores which could be embedded within the SoC platform. For high performance low power applications like OFDM Transmitter/Receiver shown in Fig. 1, there is a continuous demand for FFT/IFFT cores [1]-[3], which provide high throughput while minimising power consumption.

It can be shown that the main source of power consumption in a typical CMOS logic gate, is due to the switching power,

P_{sw} , given by [4]:

$$P_{sw} = (1/2)kC_{load}V_{dd}^2f \quad (1)$$

¹ This work was supported by the British Council.

M. Hasan is with the School of Engineering and Electronics, University of Edinburgh, Edinburgh, UK (e-mail: mh@ee.ed.ac.uk).

T. Arslan, is with both School of Engineering and Electronics, University of Edinburgh, Edinburgh and System level Integration Institute, Livingston, UK. (e-mail: Tughrul.Arslan@ee.ed.ac.uk).

J. S. Thompson is with the School of Engineering and Electronics, University of Edinburgh, Edinburgh, UK (e-mail: John.Thompson@ee.ed.ac.uk).

Where V_{dd} is the supply voltage, f is the clock frequency, C_{load} is the load capacitance of the gate, and k is the switching activity factor which is defined as the average number of times the gate makes an active transition in a single clock cycle. Therefore, for achieving low power in CMOS circuits one must target minimising one or more of the parameters C_{load} , V_{dd} and k . This paper primarily deals with low power architectures obtained by reducing the switching activity.

A number of researchers has investigated the area of low power implementation of FFT processors. In [5], the author has implemented a low power cache-memory architecture by using an algorithm that offers good data locality over large portions of the computation. This architecture is more suitable for longer length transforms. In [6][7], the authors have proposed a low power architecture based on an algorithm that effectively minimises the number of complex multiplications. In [8][9], the authors have investigated the realisation of low power FFT processors by using asynchronous processing elements. Work on the application of order based processing to FFT coefficients is restricted. The only reported work involves order based processing of coefficients and data at the inputs of the FFT computational units so as to minimise the overall switching activity of successive coefficients and data samples [10][11]. This results in coefficient switching activity reduction of just 19% whereas the data activity increases by 1% for a 9-point FFT [11]. The authors in [10][11] have not shown the actual power reduction obtained by their scheme in the presence of hardware overheads in the full FFT architecture.

This paper reduces the power consumption of a popular low power radix-4 pipelined FFT processor architecture proposed in [12] by modifying its operation sequence. The complex multiplier within the butterfly processing unit is one of the most power consuming block of the pipelined FFT processor. The switching activity between successive coefficients fed to the complex multiplier can be drastically reduced by coefficient ordering and hence its power consumption. The coefficient ordering requires corresponding data sequencing as per new coefficient ordering. Data sequencing is performed by a commutator in the pipelined FFT processors. Hence, a novel commutator architecture is proposed to handle the new data sequencing for stage 1 of a 16-point FFT processor. The data sequencing for stage 2 is restored by using a dual port RAM (DM) along with a ROM for its address generation. This ordering technique is suitable only for stage 1 of a 16-point radix-4 FFT processor due to the need of restoring data ordering for the following stage. This in turn requires only small hardware overhead in the form of a six word additional

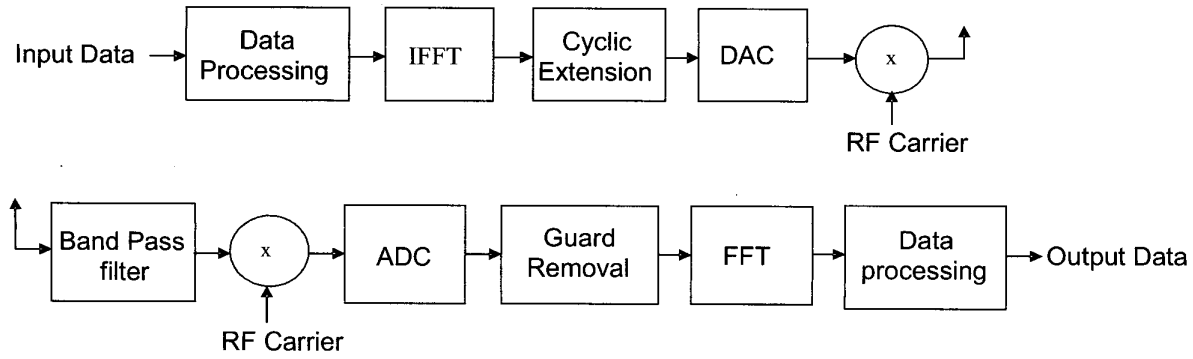


Fig.1. Simplified block diagram of OFDM Transmitter/Receiver

DM (*ADM*) following stage 1 of a 16-point FFT contrary to a much larger *ADM* following stage 1 of a 64-point FFT. The large size *ADM* is required because stage 1 of a 64-point FFT handles 64 data samples compared to only 16 for stage 1 of a 16-point FFT. Hence, this ordering technique is limited to stage 1 of a 16-point FFT processor or stage 2 of a 64-point FFT processor and so on. Moreover, the commutator design for incorporating ordering becomes very difficult for the bigger stage 1 of a 64-point FFT processor.

II. ALGORITHM

The N -point DFT of a finite duration sequence $x(n)$ is defined by (2) as follows.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad k = 0, 1, \dots, N-1; \quad (2)$$

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

Let N be a composite number of v integers so that $N = r_1 r_2 \dots r_v$ and define

$$N_t = N / r_1 r_2 \dots r_t \quad 1 \leq t \leq v-1$$

Where t is the stage number of the decomposed DFT and r_t its radix. The pipelined FFT processor is obtained by decomposing an N -point DFT into v stages. The final stage is defined in [12] as follows.

$$X(r_1 r_2 \dots r_{v-1} m_v + \dots + m) = \sum_{q_{v-1}=0}^{r_v-1} x_{v-1}(q_{v-1}, m_{v-1}) W_{r_v}^{q_{v-1} m_v} \quad (3)$$

Whereas intermediate stages (t) are given by the following recursive equation.

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

Where $2 \leq t \leq v-1$, $0 \leq m_t \leq r_t-1$, $0 \leq q_t \leq N_{t-1}-1$ and $2 \leq i \leq v$

For $r_1 = 4$, the flowgraph of a 16-point FFT based on the above formulation is shown in Fig. 2. The corresponding equations are as follows.

$$X(4m_2 + m_1) = \sum_{q_1=0}^3 x_1(q_1, m_1) W_4^{q_1 m_2} \quad (5)$$

$$x_1(q_1, m_1) = W_{16}^{q_1 m_1} \sum_{p=0}^3 x_1(4p + q_1) W_4^{p m_1} \quad (6)$$

Where $0 \leq m_1, m_2 \leq 3$

In Fig. 2, each open circle represents the summation while the dots define the stage boundaries. The number inside the open circle is the value of m_1 (for stage 1) or m_2 (for stage 2). The number outside the open circle is the FFT coefficient applied.

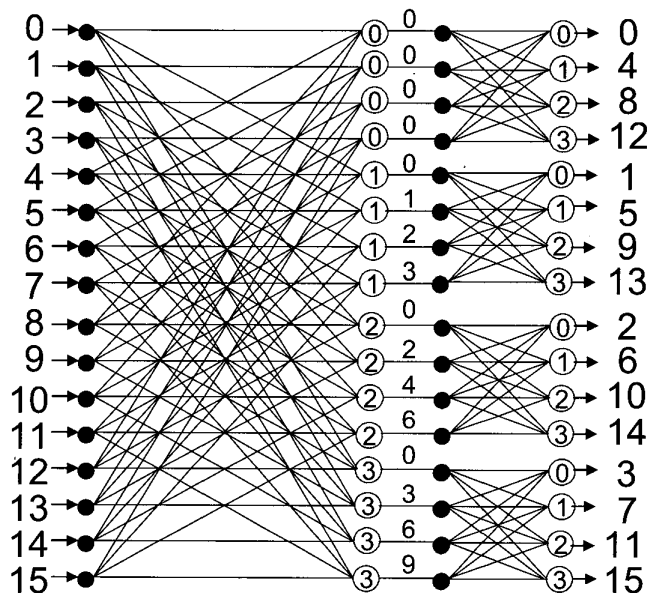


Fig. 2. Signal flowgraph of a radix-4 16-point FFT.

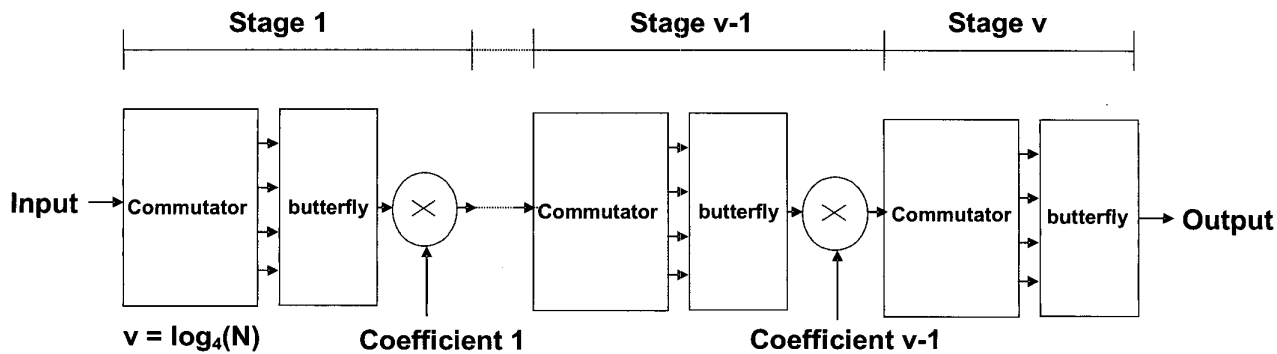


Fig. 3. N-point radix-4 pipelined FFT Architecture

III. ORDERED PIPELINED FFT ARCHITECTURE

A pipelined N-point radix-4 FFT processor based on the previously described algorithm, shown in Fig. 3, will have $\log_4 N$ stages. Each stage produces one output within each word cycle. Each stage contains a commutator, a butterfly element (for summation) and a complex multiplier. The sequential outputs at each stage must be ordered in accordance with the value of m_i . For instance, from Fig. 2 at stage 1, the outputs associated with $m_1=0$ are produced in the first four word cycles, then those associated with $m_1=1$ in the next four cycles and so on. As seen in equation (3), the input data for each summation at stage t are separated in time by N_t words. The requisite commutator comprises of six shift registers along with three multiplexers and is given in [12].

This paper proposes altered operation sequencing for stage 1 of a 16-point radix-4 pipelined FFT processor based on its signal flowgraph shown in Fig. 4. Normally, the fixed coefficients are fed to the complex multiplier in an order given in Fig. 2 starting from $m_1 = 0$ and ending with $m_1 = 3$ for stage 1 of a 16-point FFT processor. Our approach involves ordering the coefficient sequence so as to minimise switching activity between successive coefficients fed to the

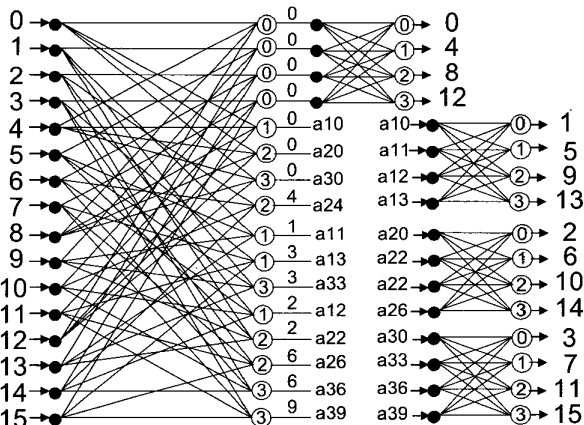


Fig. 4. Signal flowgraph of a radix-4 ordered 16-point FFT.

TABLE I
ORDERED AND CONVENTIONAL COEFFICIENT SEQUENCE FOR A 16-POINT RADIX-4 FFT

Coefficient sequence after stage 1 of a 16-point FFT	16-bit quantised coefficient sequence (real, Imag)	Ordered Coefficient sequence	Ordered quantised Coefficient sequence (flag, real, Imag)
W_0	7ff, 0000	W_0	1, 8001, 0000
W_0	7ff, 0000	W_0	1, 8001, 0000
W_0	7ff, 0000	W_0	1, 8001, 0000
W_0	7ff, 0000	W_0	1, 8001, 0000
W_0	7ff, 0000	W_0	1, 8001, 0000
W_1	7641, cf04	W_0	1, 8001, 0000
W_2	5a82, a57d	W_0	1, 8001, 0000
W_3	30fb, 89be	W_4	0, 0000, 8000
W_0	7ff, 0000	W_1	0, 7641, cf04
W_2	5a82, a57d	W_3	1, cf05, 89be
W_4	0000, 8000	W_3	1, cf05, 89be
W_6	a57d, a57d	W_2	0, 5a82, a57d
W_0	7ff, 0000	W_2	0, 5a82, a57d
W_3	30fb, 89be	W_6	1, 5a83, a57d
W_6	a57d, a57d	W_6	1, 5a83, a57d
W_9	89be, 30fb	W_9	1, 7642, 30fb

Normal Switching activity= 192

Switching activity with ordering = 78

multiplier for stage 1 of a 16-point FFT or stage 2 of a 64-point FFT as listed in Table I and also shown in Fig. 4. The coefficients are ordered so as to minimise switching activity between successive coefficients by minimising the Hamming distance between them. The ordered coefficient set is obtained by first arranging only the imaginary part of the coefficient set on the basis of Hamming distance. It is followed by picking up the corresponding real part of the coefficient or its two's complement depending upon the Hamming distance with respect to the previously arranged real part. A flag bit is asserted to indicate the presence of real part in two's complement form. This flag bit is also used to selectively complement the multiplier output [13]. The switching activity decreases from 192 to just 78, a reduction of 59% by following this ordering approach. The coefficient ordering

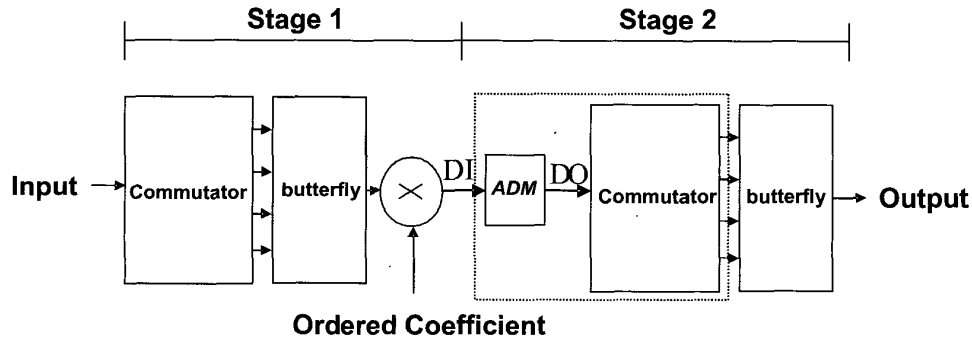


Fig. 5. Ordered 16-point radix-4 FFT processor architecture

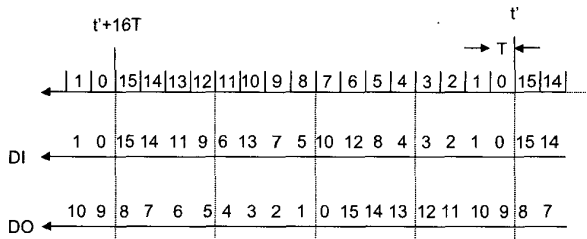


Fig. 6. Input and output sequence of ADM

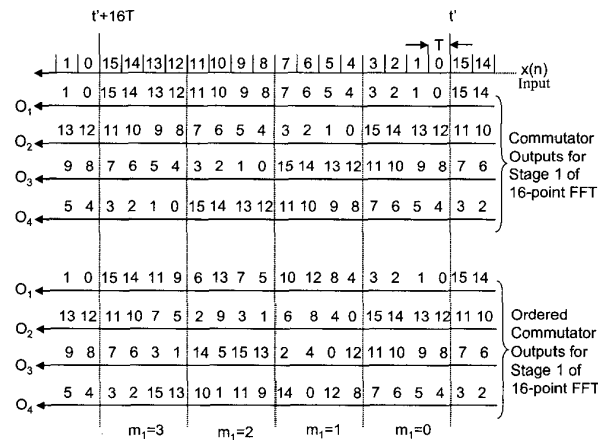


Fig. 7. Timing diagram of normal and ordered commutator outputs for the first stage of a 16-point radix-4 FFT processor

requires corresponding data ordering. The data ordering is performed by a novel design of the commutator for stage 1 of a 16-point radix-4 FFT processor. The ordered data sequence at the output of the complex multiplier for stage 1 of the 16-point FFT processor has to be converted back into a normal data sequence for its stage 2. This data sequence conversion is accomplished by the combination of ADM along with a ROM (ROM0) for its addressing as given in Fig. 5. The new architecture of the 16-point ordered pipelined FFT processor is shown in Fig. 5. The input and output sequences of ADM namely DI and DO respectively are shown in Fig. 6. It is clear that DO is in normal order to be directly fed to the stage 2 commutator. The stage 2 commutator will be the same as

given in [12]. The stage 1 commutator design, to support the ordering scheme, is as follows.

A. Stage 1 Commutator design for a 16-point FFT Processor

As seen in equation (3) and Fig. 2, the input data for each summation at stage 1 of a 16-point FFT are separated in time by four words. The timing of the ordered data sequence corresponding to the ordered coefficient sequence and the normal sequence as a function of time is shown in Fig. 7, t' is the instant when the first input word arrives. Each input word occupies a word slot of duration T and is numbered according to its appearance in time. This ordered data sequence can be generated with the help of a commutator. It is difficult to generate the ordered data sequence with the help of a conventional FIFO based on shift registers (SRs) or DMs as given in [12]. In order to achieve flexibility, the commutator is constructed by using double size (eight words) three triple port RAM (TM) based FIFOs rather than six four word DM based FIFOs. The additional read port in TM greatly helps in generating the ordered sequence. The commutator comprises

TABLE II
CONTENTS OF CONTROL ROM (ROM1) FOR GENERATING ADDRESSING AND CONTROL SIGNALS FOR THE COMMUTATOR

Address	Contents
	{cs,c[2:0], aw, adrf, adre, adrd, adrc, adra, m[7:0]}
0000	2e8804, 29
0001	1b4184, e4
0010	2fac86, 29
0011	16a34d, 89
0100	17a7df, 89
0101	38534d, e5
0110	346595, 9a
0111	2c1a6d, a9
1000	2c3efd, a9
1001	386595, e5
1010	3877dd, e5
1011	000000, 41
1100	008041, 41
1101	010082, 41
1110	0180c3, 41
1111	160104, 49

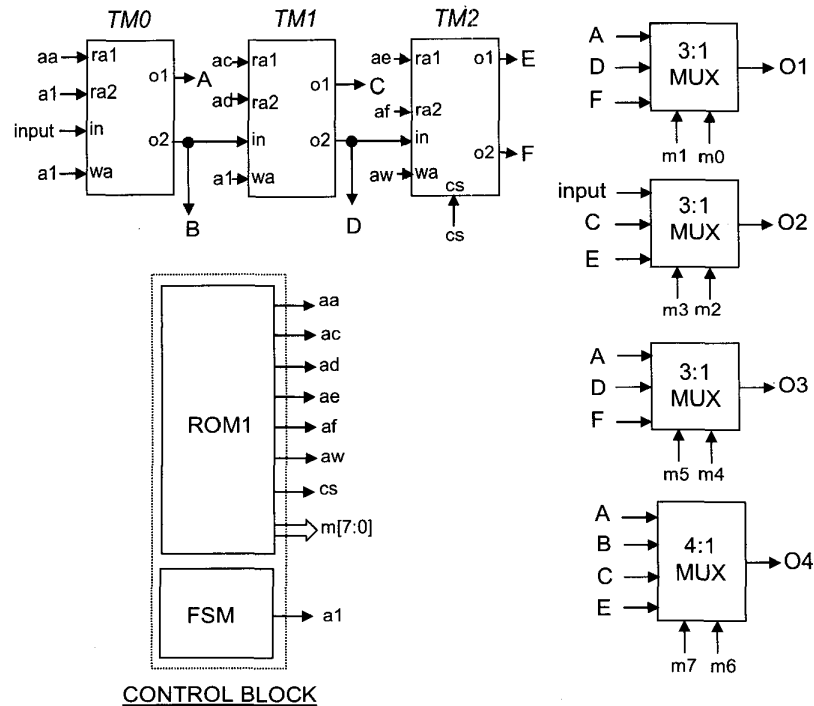


Fig. 8: First stage commutator architecture for a radix-4 ordered 16-point FFT processor

of three *TMs* (Two read ports and one write port), a finite state machine (FSM), a ROM (ROM1) and four multiplexers of variable size as shown in Fig. 8. A *TM* acts as a FIFO with two possible outputs for flexibility. The three *TMs* have six possible outputs but only four outputs are chosen at any time by the four multiplexers depending upon the desired ordered sequence. Each *TM* has a depth of eight words for stage 1 of a 16-point FFT processor. The three bit sequential address *a1* is generated by the FSM of the CONTROL BLOCK for the write ports of *TM0* and *TM1* and the B port of *TM0* respectively. The ROM1 of the CONTROL BLOCK provides the addresses of all the other read and write ports along with multiplexer controls to generate the ordered data sequence. It also helps to keep the unused outputs of *TMs* to their previous values. The ROM1 contents are listed in Table II. The lower byte of the 31-bit wide ROM1 controls the four multiplexers ($m[7:0]$). The next more significant 18-bits generate the write address (*aw*) and the read addresses (*addrf*, ..., *addra*) of *TMs*. The still higher 3-bits control the stage 2 butterfly ($c[2:0]$) and the most significant bit controls the chip select of *TM2* (*cs*). *TM2* is selectively disabled for writing by the logic high on its chip select input. This is done to avoid unnecessary writing of *TM2* thereby reducing power consumption. It is clear from the highlighted nibbles of Table II that the lower three bits (*addra*) of these nibbles remain fixed in two blocks. It means that '*addra*' remains fixed for all these locations and hence there is no switching activity on port A for almost half of the time duration. This addressing approach reduces the switching activity on the

unused ports and hence the power consumption. This sort of addressing is also employed for ports C, E and F.

IV. RESULTS

The conventional and ordered pipelined FFT processor architectures have been implemented in register transfer level hardware description language and then synthesized using 0.18 μ CMOS technology library. Power evaluation was then carried out on the circuit netlist using a supply voltage of 1.8V and a clock frequency of 100 MHz. The switching activity decreases from 192 to 78, a reduction of 59% as per Table I. The comparative results in terms of power for different FFT lengths, FIFO implementation styles and two common low power multiplier types [14] are given in Table III. The FIFO was implemented in three different ways namely based on *SR*, *DM* and our *TM*. *DM-SR* approach uses *DM* based FIFO for stage 1 commutator and *SR* based FIFO for stage 2 commutator in case of a 64-point processor. The design of a 16-point FFT processor has been carried out for two different multiplier types namely carry save array type (*csa*) and Non-Booth coded Wallace tree type (*nbt*). It is clear from Table IV that our ordered architecture gives power savings for the two multiplier types and different FIFO architectures for the 16-point and 64-point FFT processors. The percentage power saving of our ordered approach is less for *nbt* multiplier type in most cases but the *nbt* multiplier based architecture consumes less power than the one based on *csa*. Moreover, *DM* based approach is better for stage 1 of a

TABLE III
COMPARATIVE POWER CONSUMPTION FOR THE ORDERED AND CONVENTIONAL FFT PROCESSORS

FFT size	Multiplier type	<i>SR</i> based (in mW)	<i>DM</i> based (in mW)	<i>DM-SR</i> based (in mW)	Ordered (in mW)	% Saving / <i>SR</i>	% Saving / <i>DM</i>	% Saving / <i>DM-SR</i>
16-point	<i>csa</i>	125.32	135.42	-	96.48	23	29	-
	<i>nbw</i>	93.77	114.28	-	80.19	14	30	-
64-point	<i>csa</i>	351.60	289.14	276.73	252.73	28	13	09
	<i>nbw</i>	296.18	238.06	225.18	217.18	27	09	04

64-point FFT whereas *SR* based approach outperforms *DM* for the smaller FIFO required in stage 1 of a 16-point FFT processor. Hence, *DM-SR* architecture for the 64-point FFT is more effective than the *SR-SR* and *DM-DM* architectures. Our ordered approach gives power savings of the order of 23% and 29% with respect to *SR* and *DM* respectively for the 16-point FFT processor using *csa* multiplier. The power saving of our ordered approach is 28%, 13% and 9% with respect to *SR*, *DM* and *DM-SR* respectively for the 64-point FFT processor using *csa* multiplier. The percentage power saving with respect to the overall power consumption will go down further for longer FFTs because the ordering approach is restricted only to stage 1 a 16-point FFT or stage 2 of a 64-point FFT processor. This restriction is imposed in view of the large *ADM* requirement and commutator design complexities for initial stages of longer FFTs. This large *ADM* will more than offset any power saving due to ordering in the complex multipliers.

Table IV lists the power consumed by the major cells of the pipelined FFT processor for different FIFO implementations. It is clear from Table IV that the *SR* based FIFO architecture is inferior to the other architectures for the 64-point FFT due to the high power consumption in the large FIFO blocks of stage 1 commutator. This high power consumption is attributed to the shifting (switching) of all data samples on every clock cycle in the traditional

FIFO based on *SR*. It is also evident from Table IV that the power saving in the ordered approach is taking place not only in the multiplier but also in the novel stage 2 commutator. The novel stage 2 commutator architecture comprises of three double size FIFOs based on *TMs* rather than the traditional six *DM* or *SR* based FIFOs. The new commutator architecture consumes much less power due to less data movement and hence switching activity among the three FIFOs as compared to the traditional six FIFOs. The power consumption in the additional read ports of *TM* is reduced by keeping the outputs of the unused read ports to their previous values by addressing these ports through ROM1. The stage 3 commutator consumes more power in the ordered approach as compared to the other approaches because its power consumption also includes the power consumed by *ADM*. The stage 2 butterfly in the ordered approach consumes more power than the other approaches mainly due to the different stage 2 commutator architecture. The stage 3 butterfly in our approach consumes much less power than the other approaches because it has been designed using XOR gates (control inverters) and a summer rather than the traditional programmable adders/subtractors [12]. The stage 1 and stage 2 butterflies in our ordered approach consume more power than the other approaches due to the different input/output conditions in the form of a new stage 2 commutator architecture based on larger size *TMs*. The stage 1 multiplier consumes less power in our

TABLE IV
COMPARATIVE MAJOR CELLS POWER CONSUMPTION FOR THE ORDERED AND CONVENTIONAL FFT PROCESSORS FOR THE 64-POINT FFT PROCESSOR WITH *csa* MULTIPLIER

S.N.	FFT Cells	<i>SR</i> based (in mW)	<i>DM</i> based (in mW)	<i>DM-SR</i> based (in mW)	Ordered (in mW)
1.	Stage 1 Commutator	91.01	53.87	53.84	53.77
2.	Stage 2 Commutator	23.27	33.35	22.77	15.95
3.	Stage 3 Commutator	9.64	9.29	9.37	15.17
4.	Stage 1 Butterfly	1.82	1.63	1.63	1.78
5.	Stage 2 Butterfly	2.01	1.81	1.59	3.04
6.	Stage 3 Butterfly	6.15	6.10	6.10	4.40
7.	Stage 1 multiplier	26.32	26.50	26.35	23.89
8.	Stage 2 multiplier	25.27	25.81	25.81	18.31

ordered approach due to different input/output conditions in the form of a different stage 1 butterfly and stage 2 commutator architectures. The stage 2 multiplier consumes substantially less power in our ordered approach because of the drastic reduction in the switching activity at its coefficient input by ordering.

V. CONCLUSION

This paper has presented a novel order based pipelined FFT processor architecture suitable for shorter FFTs. However, this design approach can also be applied to the last stages of longer FFTs. The switching activity reduction obtained by ordering is around 59%. The corresponding power saving varies from 23% to 14% for a 16-point FFT and 9% to 4% for a 64-point FFT using commonly used low power multipliers. This low power design has a lot of promise in wireless LAN applications requiring short FFTs. It can also be used for longer FFTs in other OFDM applications like Digital audio and video broadcasting.

REFERENCES

- [1] J. A.C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," IEEE Communication Magazine, vol. 36, pp. 112-117, Feb. 1998.
- [2] R. Van Nee and R. Prasad, "OFDM wireless multimedia communications", Artech House, Boston, 2000.
- [3] J. Terry and J. Heiskala, "OFDM wireless LANs: A Theoretical and Practical Guide", Sams Publisher, 2002.
- [4] I. S. Abu-Khater, A. Bellaouar and M. I. Elmasry., "Circuit Techniques for CMOS Low-power High-performance Multipliers", IEEE Journal of Solid-State Circuits, vol. 31, no. 10, pp. 1535-1546, Oct. 1996.
- [5] M. B. Bevan, "A Low-Power, High-Performance, 1024-point FFT Processor", IEEE Journal of Solid-State Circuits, vol. 34, no. 3, pp. 380-387, March 1999.
- [6] 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.
- [7] L. Jia, Y. Gao, J. Isoaho and H. Tenhunen, "Implementation of a low power 128-point FFT processor", in Proceedings of Fifth International conference on Solid-state and Integrated Circuit Technology, pp. 369-372, 1998.
- [8] K.S. Stevens and B. W. Suter, "A mathematical approach to a low power FFT Architecture", Proc. IEEE International symposium on Circuits and Systems, pp. II-21- II-24, 1998.
- [9] B.W. Hunt, K.S. Stevens, B.W. Suter and D.S. Gelosh., "A single chip low power asynchronous implementation of an FFT algorithm for space applications", Proceedings Fourth Internal symposium on advanced research in asynchronous circuits and systems, pp. 216-223, 1998.
- [10] K. Masselos, P. K. Merakos, T. Stouraitis and C.E. Goutis, "Novel techniques for bus power consumption reduction in realizations of Sum-of-Product computations", IEEE Transactions on Very large Scale Integration Systems, vol.7, no. 4, December, 1999.
- [11] K.Masselos, S. Theoharis, P.K Merakos., T. Stouraitis and C.E.Goutis, "A novel methodology for power consumption reduction in a class of DSP algorithms", Proc. IEEE International symposium on Circuits and Systems, pp.VI-199- VI-202, 1998.
- [12] Bi Guoan and E.V.Jones, "A Pipelined FFT Processor for Word-Sequential Data", IEEE transaction on acoustics, speech, and signal processing, vol. 37, no. 12, December 1989.
- [13] M. Hasan and T.Arslan, "Implementation of low power FFT processor cores using a novel order based processing scheme", accepted for publication in IEE proceedings on Circuits, Devices and systems.
- [14] A.T. Erdogan, M. Hasan and T. Arslan, "Algorithmic low power FIR cores", accepted for publication in IEE proceedings on Circuits, Devices and systems.



Mohd. Hasan received the B.Sc. Engg. degree in Electronics Engineering in 1990 from A.M.U., Aligarh. Subsequently, he completed his M.Tech. in Integrated Electronics and Circuits from the Indian Institute of Technology, Delhi. He joined Electronics Engg. Dept., A.M.U., Aligarh as Lecturer in 1992. He became Reader in 1997. Currently, he is pursuing Ph.D. in low power architectures for signal processing and Telecommunications in the School of Engineering and Electronics., University of Edinburgh, U.K. His research interests include Low power IC design for wireless communications, Reconfigurable systems on FPGA's etc.



Tughrul Arslan is a Reader in the School of Engineering and Electronics in the University of Edinburgh. He is a member of the Integrated Systems Group and leads the System Level Integration activity. His research interests include: Low Power Design, DSP Hardware Design, System-On-Chip (SoC) Architectures, Evolvable Hardware, and the use of Genetic algorithms in Hardware design issues. He is the principal investigator in a number of projects funded by EPSRC, DTI, Scottish Enterprise together with a number of industrial partners. Dr Arslan serves on the technical committee of a number of international conferences including the International Symposium on Circuits and Systems (ISCAS), and the Annual ASIC/SoC, and NASA/DoD on Evolvable Hardware.



John S Thompson received his BEng and PhD degrees from the University of Edinburgh in 1992 and 1996, respectively. Since September 1999, he has been working as a lecturer in what is now the School of Engineering and Electronics at the University of Edinburgh. His research interests include estimation theory, signal processing algorithms and antenna array techniques for wireless communications. He has 60 publications to date and is currently an honorary editor of IEE Proceedings on Vision, Image and Signal Processing