Design and Implementation of a 1024-point Pipeline FFT Processor

Shousheng He and Mats Torkelson
Department of Applied Electronics, Lund University
S-22100 Lund, Sweden
email: he@tde.lth.se; torkel@tde.lth.se

Abstract— **Design and implementation of a 1024-point pipeline FFT processor is presented. The architecture is based on a new form of FFT, the radix-2^2 algorithm. By exploiting the spatial regularity of the new algorithm, minimal requirement for both dominant components in VLSI implementation has been achieved: only 4 complex multipliers and 1024 complex-word data memory for the pipelined 1K FFT processor. The chip has been implement in 0.5μ CMOS technology and takes an area of 40 mm^2. With 3.3v power supply, it can compute $2^n, n = 0, 1, \ldots, 10$ complex point forward and inverse FFT in real time with up to 30MHz sampling frequency. The SQNR is above 50dB for white noise input.**

I. Introduction

Fast Fourier Transform (FFT) has been used in a wide range of applications, such as wide-band mobile digital communication system based on Orthogonal Frequency Division Multiplexing (OFDM) principle [1, 2], where the system implementation is only feasible when the equipment complexity and power consumption are greatly reduced by utilizing a real-time FFT transformer to replace the bank of (de)modulators for each individual sub-carriers.

FFT operation has been proven to be both computational intensive, in terms of arithmetic operations, and communicational intensive, in terms of data swapping/exchanging in the storage. For real-time processing of FFT transform, $O(\log N)$ arithmetic operations are required per sample cycle, where N is the length of the transform. High speed real-time processing can be accomplished in two different ways. In a conventional, general purpose processor approach, a single processor driven to a very high clock frequency, which is $O(\log N)$ times the sampling frequency, is used to carry out the operation. While in an application specific approach, parallel or concurrent/pipelined processors, operating on a clock frequency close or equivalent to the sampling frequency, are used to attain the performance. Analysis has shown that the second approach is more preferable when power consumption is limited by the application environment, such as in mobile communication [3].

Pipeline FFT processor is a class of architectures for application specific real-time DFT computation utilizing fast algorithms. It is characterized by non-stopping processing on a clock frequency of the input data sampling. A lower clock frequency is a clear advantage for pipeline architectures, when either a high speed processing or a low power solution is sought. In addition, pipeline structure is highly regular, which can be easily scaled and parameterized when Hardware Description Language (HDL) is used in the design. It is also more flexible when transforms of different lengths are to be computed with the same chip.

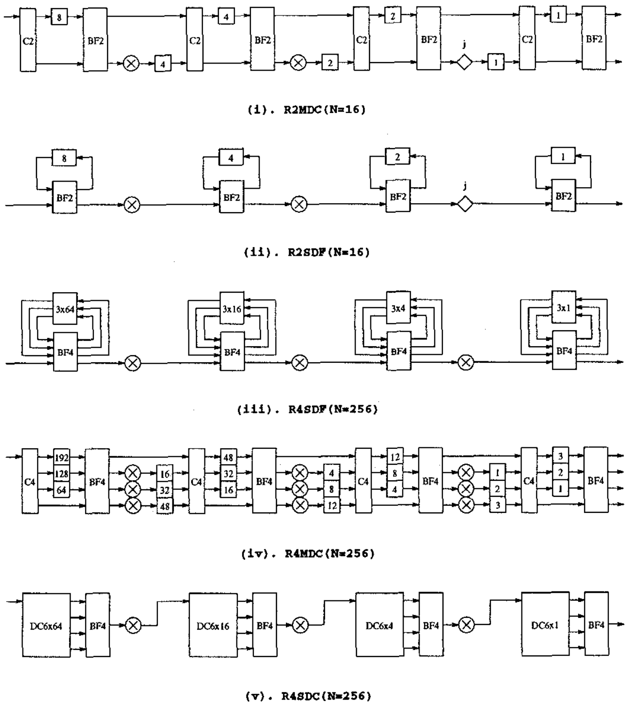
In this paper we present the implementation of a 1024-point pipeline FFT processor based on a new FFT algorithm. In the following section, pipeline FFT processors are briefly reviewed. Then the new architecture is presented as compared with previous ones. Functional and physical implementation issues are discussed in the subsequent section. Finally we conclude with a brief performance estimation of the newly designed 1K FFT chip.

II. Pipeline FFT processor architectures

The architecture design for pipeline FFT processor had been the subject of intensive research as early as in 70's when real-time processing was demanded in such applications as radar signal processing [4]. Several architectures have been proposed over the last 2 decades. Here different approaches are put into functional blocks with unified terminology. The additive butterfly has been separated from multiplier to show the hardware requirement distinctively, as in Fig. 1. The control and twiddle factor reading mechanism have been also omitted for clarity. All data and arithmetic operations are complex, and a constraint that N is a power of 4 applies.

R2MDC: Radix-2 Multi-path Delay Commutator [4] was probably the most classical approach for pipeline implementation of radix-2 FFT algorithm. The input sequence has been broken into two parallel data stream flowing forward, with correct "distance" between the data elements entering the butterfly scheduled by proper delays. Both butterflies and multipliers are in 50% utilization.

0-7803-4292-5/97/$10.00 © 1998 IEEE



(i). R2MDC(N=16)

(ii). R2SDF(N=16)

(iii). R4SDF(N=256)

(iv). R4MDC(N=256)

(v). R4SDC(N=256)

Figure 1: Various schemes for pipeline FFT processor

$\log_2 N - 2$ multipliers, $\log_2 N$ radix-2 butterflies and $3/2N - 2$ registers (delay elements) are required.

R2SDF: Radix-2 Single-path Delay Feedback [5] uses the registers more efficiently by storing the one butterfly output in feedback shift registers. A single data stream goes through the multiplier at every stage. It has same number of butterfly units and multipliers as in R2MDC approach, but with much reduced memory requirement: $N - 1$ registers. Its memory requirement is minimal.

R4SDF: Radix-4 Single-path Delay Feedback [6] was proposed as a radix-4 version of R2SDF, employing CORDIC[1] iterations. The utilization of multipliers has been increased to 75% by storing 3 out of 4 radix-4 butterfly outputs. However, the utilization of the radix-4 butterfly, which is fairly complicated and contains at least 8 complex adders, is dropped to only 25%. It requires $\log_4 N - 1$ multipliers, $\log_4 N$ full radix-4 butterflies and storage of size $N - 1$.

R4MDC: Radix-4 Multi-path Delay Commutator [4] is a radix-4 version of R2MDC. It has been used as the architecture for the initial VLSI implementation of pipeline FFT processor [7] and massive wafer scale integration [8]. However, it suffers from low, 25%, utilization of all components, which can be compensated only in some special applications where four FFTs are being processed simultaneously. It requires $3 \log_4 N$ multipliers, $\log_4 N$ full radix-4 butterflies and $5/2N - 4$ registers.

R4SDC: Radix-4 Single-path Delay Commutator [9] uses a modified radix-4 algorithm with programable $1/4$ radix-4 butterflies to achieve higher, 75% utilization of multipliers. A multiplexed Delay-Commutator also reduces the memory requirement to $2N - 2$ from $5/2N - 1$, that of R4MDC. The butterfly and delay-commutator become relatively complicated due to programmability requirement. R4SDC has been used recently in building the largest ever single chip pipeline FFT processor for HDTV application [10].

A swift skimming through of the architectures listed above reveals the distinctive merits of the different approaches: First, the delay-feedback approaches are always more efficient than corresponding delay-commutator approaches in terms of memory utilization since the butterfly output share the same storage with its input. Second, radix-4 algorithm based single-path architectures have higher multiplier utilization, however, radix-2 algorithm based architectures have simpler butterflies which are better utilized. The new approach developed in the following sections is highly motivated by these observations.

III. Radix-2^2 FFT Algorithm based Architecture

As it has been well known, VLSI complexity is related not only to the number of operations required for the computation, but also to the structure of the algorithm. Efficient resources utilization requires spatial regularity of the operations in the Signal Flow Graph (SFG), so that pipelines can be fully scheduled. The bandwidth constraint of on-chip communication also demands that the fan-in and fan-out of each processing node in the SFG to be small. For example, radix-2 butterfly is better in this respect than that of radix-4. With these considerations in mind, a *hardware oriented* radix-2^2 FFT algorithm has been derived recently to restructure the SFG for efficient VLSI implementation [11]. The SFG of a 16-point Decimation-In-Frequency (DIF) radix-2^2 FFT is shown in Fig. 2, where small diamonds represent trivial multiplication by $W_N^{N/4} = -j$, which involves only real-imaginary swapping and sign inversion.

A unique feature of the Radix-2^2 FFT algorithm is that it has the same multiplicative complexity as radix-4 algorithms, but still retains the radix-2 SFG structure. The multiplicative operations are in such an arrangement that only every other column has non-trivial multiplications. This is a great structural advantage over other algorithms when pipeline architecture is under consideration.

Mapping radix-2^2 DIF FFT algorithm to R2SDF architecture discussed in above section, a new architecture of Radix-2^2

[1] The Coordinate Rotational Digital Computer

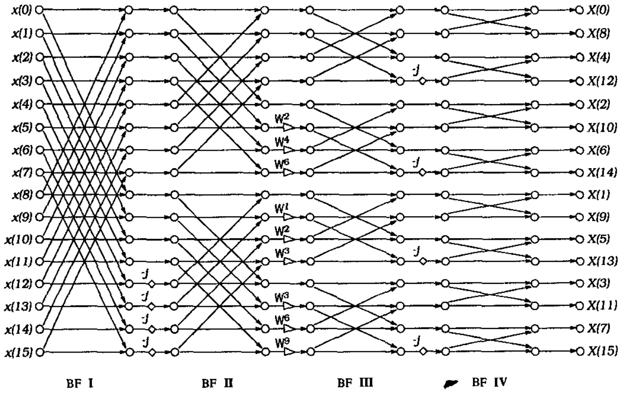


Figure 2: Radix-2^2 DIF FFT flow graph for $N = 16$

Single-path Delay Feedback (R2^2SDF) approach is obtained [12]. Fig. 3 outlines an implementation of the R2^2SDF architecture for $N = 256$. Note the similarity of the data-path to R2SDF and the reduced number of multipliers. The implementation uses two types of butterflies, one identical to that in R2SDF, the other contains also the logic to implement the trivial twiddle factor multiplication.

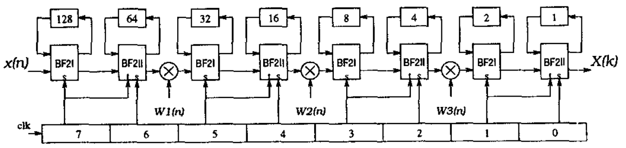


Figure 3: R2^2SDF pipeline FFT architecture for $N = 256$

Due to the spatial regularity of Radix-2^2 algorithm, the synchronization control of the processor is very simple. A ($\log_2 N$)-bit binary counter serves two purposes: synchronization controller and address counter for twiddle factor reading in each stage.

Table 1: Hardware requirement comparison

	multiplier #	adder #	memory size	control
R2MDC	$2(\log_4 N - 1)$	$4\log_4 N$	$3N/2 - 2$	simple
R2SDF	$2(\log_4 N - 1)$	$4\log_4 N$	$N - 1$	simple
R4SDF	$\log_4 N - 1$	$8\log_4 N$	$N - 1$	medium
R4MDC	$3(\log_4 N - 1)$	$8\log_4 N$	$5N/2 - 4$	simple
R4SDC	$\log_4 N - 1$	$3\log_4 N$	$2N - 2$	complex
R2^2SDF	$\log_4 N - 1$	$4\log_4 N$	$N - 1$	simple

The hardware requirement of proposed architecture as compared with various approaches is shown in Table 1, where not only the number of complex multipliers, adders and memory size but also the control complexity are listed for comparison. For easy reading, base-4 logarithm is used whenever applicable. It shows R2^2SDF has reached the minimum requirement for both multiplier and the storage, and only second to R4SDC for adder. This makes it an ideal architecture for VLSI implementation of pipeline FFT processors.

IV. Functionality Implementation

The architecture proposed in the above section has been modeled in hardware description language VHDL with generic parameters for transform length and word-length, using fixed point arithmetic. Implementation details, such as the control logic for forward/inverse transform, wordlength variation and scaling are taken into consideration. Data by-pathing has been provided so that shorter transforms can be also carried out on a design for longer transform, as long as the transform lengths are of integer power of 2. Special scheduling has been also arranged for "loose" pipeline to allow an arbitrary gap between 2 successive data frames. This happens, for example, when a prefix insertion is required for OFDM system.

The area/power consumption in the pipeline architectures is dominated by the First-In First-Out (FIFO) register files at each stage and the complex multipliers at every other stage. To reduce the switching activities due to the unnecessary data moving in the FIFO, reconstruct the storage is mandatory. A known approach is to construct FIFO with 2-port RAM with read and write addresses displaced by a constant [13], as shown in Fig. 4(i). The main drawback of using 2-port RAM is that each RAM cell takes an area 33% larger than the corresponding single port RAM cell [14], and it consumes more power due to constant read/write operation.

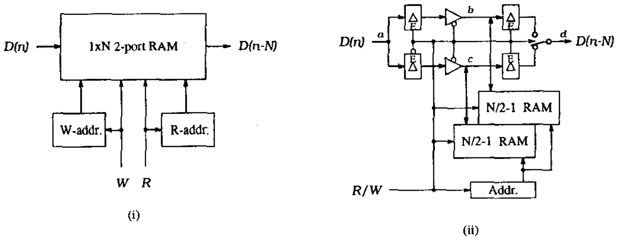


Figure 4: Implementing shift register with (i). 2-port RAM. (ii). single port RAM

In our implementation, single port RAM modules are used to construct FIFOs. To avoid read/write conflict, a 2-words at a time scheme is delineated. For length-N FIFO an $(N/2 - 1)$ double-word RAM is used, as shown in Fig. 4(ii). The read and write operations are interleaved and each of them is active every other clock cycle.

For complex multiplier, the other dominant component in the architecture, an efficient structure based on distributed arithmetic has been used to reduce the area/power consumption for the operation [15].

Targeting on applications in OFDM systems, a Signal-To-Quantization-Noise-Ratio (SQNR) higher than 40dB has been specified with input/output interface wordlength set to 12x2bit complex word. Applying gradual wordlength growth up-to 16x2bit and programmable scaling control, a SQNR of 51dB has been obtained for white noise input.

V. Physical Layout and testing considerations

A design for 1024 complex point FFT has been synthesized using the VHDL model described above with a standard cell library and a RAM/ROM generator in 0.5μ CMOS technology. The synthesized chip consists 6 RAM modules of different sizes, 2 ROM modules and over 21K standard cells, in which 2K are flip-flops. A clock trunk driven by parallel drivers in the library is employed to reduce the clock skew. Placed and routed with a channelless router, the design takes an area of $5 \times 8\text{mm}^2$. Fig. 5 shows the final layout of the chip. Back annoted timing analysis and simulation shows that under worst-case condition, clock rate above 20MHz have been achieved.

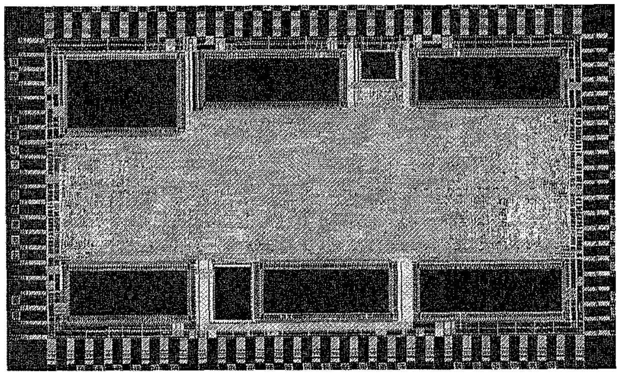


Figure 5: Layout of the 1K FFT processor

A unique testing strategy has been used on the chip which exploits the design's capability to compute different length FFTs. With just a few additional test pins, testing vector can be applied in parallel at primary input pins, which not only reduces the degradation of performance due to testing overhead, but also greatly accelerates the testing procedure. Detailed discussion of the scheme is beyond the scope of this paper and is deferred elsewhere.

VI. Conclusion

A chip for pipelined processing 1024-point FFT has been implemented in 0.5μ CMOS technology and sent for fabrication. The architecture is based on a new form of FFT, which minimizes both complex multiplier and data memory, the dominant components in the implementation, resulting in smaller area and lower power consumption. VHDL is used to model and synthesis the design. The chip can be used to compute forward and inverse FFT of $2^n, n = 0, 1, \ldots, 10$ complex points, with real time sampling frequency well above 20MHz under 3.3v power supply. This translates to above 30MHz sampling frequency under normal condition. The power consumption is expected to be low when voltage scaling is applied, owing to the low clock frequency. The chip will be available for testing by the time of the publication this paper.

References

[1] L. J. Cimini. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE Trans. Commun.*, COM-33(7):665–675, Jul. 1985.

[2] M. Alard and R. Lassalle. Principles of modulation and channel coding for digital broadcasting for mobile receivers. *EBU Review*, (224):47–69, Aug. 1987.

[3] A.P. Chandrakasan and R.W. Brodersen. *Low Power Digital CMOS Design.* Kluwer Academic Publishers, 1995.

[4] L.R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing.* Prentice-Hall, Inc., 1975.

[5] E.H. Wold and A.M. Despain. Pipeline and parallel-pipeline FFT processors for VLSI implementation. *IEEE Trans. Comput.*, C-33(5):414–426, May 1984.

[6] A.M. Despain. Fourier transform computer using CORDIC iterations. *IEEE Trans. Comput.*, C-23(10):993–1001, Oct. 1974.

[7] E. E. Swartzlander, W. K. W. Young, and S. J. Joseph. A radix 4 delay commutator for fast Fourier transform processor implementation. *IEEE J. Solid-State Circuits*, SC-19(5):702–709, Oct. 1984.

[8] E. E. Swartzlander, V. K. Jain, and H. Hikawa. A radix 8 wafer scale FFT processor. *J. VLSI Signal Processing*, 4(2,3):165–176, May 1992.

[9] G. Bi and E. V. Jones. A pipelined FFT processor for word-sequential data. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(12):1982–1985, Dec. 1989.

[10] E. Bidet, D. Castelain, C. Joanblanq, and P. Stenn. A fast single-chip implementation of 8192 complex point FFT. *IEEE J. Solid-State Circuits*, 30(3):300–305, Mar. 1995.

[11] S. He and M. Torkelson. Radix-2^2 FFT algorithm. to appear.

[12] Swedish patent application No. 95/01371, Nov. 1995.

[13] E.E. Swartzlander Jr. *VLSI Signal Processing Systems.* Kluwer Academic Publishers, 1986.

[14] N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design.* Addison-Wesley, 1985.

[15] S. He and M. Torkelson. A complex array multiplier using distributed arithmetic. In *Proc. IEEE CICC'96*, pages 71–74, San Diego, CA, May 1996.