ICSICT98-Beijing

Implementation of A Low Power 128-Point FFT

Lihong Jia, Bingxin Li, Yonghong Gao and Hannu Tenhunen

Electronic System Design Laboratory
Royal Institute of Technology, Stockholm, Sweden
Email: jialh@ele.kth.se, lbingxin@ele.kth.se, gaoyh@ele.kth.se and hannu@ele.kth.se

***Abstract* - In this paper a low power 128 point fast Fourier transform (FFT) processor is implemented based on our new VLSI-oriented FFT algorithm - radix-2/4/8, which can effectively minimize the number of complex multiplications. A new management of the on-chip memory further reduce its power consumption. This FFT processor has been designed in 0.6 μm 3.3 V triple-metal CMOS process with an area of 10 mm^2. The chip is capable of computing a 128 point FFT every 3 μs and the power dissipation is 400 mW at 50 MHz input frequency.**

I. INTRODUCTION

The low power high speed FFT processor will be essential in many advanced signal processing applications. The new VLSI-oriented radix-2/4/8 algorithm presented in our recent paper [1] is an improved version of split radix FFT algorithm [2]. The advantage of this algorithm is its ability of decreasing the number of complex twiddle factor multiplications. Reducing the number of multiplications implies reduced number of computations required to compute an overall FFT algorithm. Therefore there is a corresponding saving in power consumption from performing fewer computations as well as decreased hardware circuit complexity. On the other hand, in designing an intensive memory system, the on-chip memory is one main source of system power dissipation. FFT processor is a typical intensive memory system. In general FFT design, on-chip memory (RAM) will occupy about 20% to 30% of whole power dissipation. The conventional way to manage RAM is one address storing one data, i.e., one data will be read or written at one time. In this paper we will utilize a new management of on-chip memory according to the analysis of the power consumption characteristics for a typical 0.6 μm 3.3 V CMOS asynchronous RAM library. The power dissipation of RAM cell will not increase proportionally with its word width. So, the power dissipation can be reduced by storing several data in one address. If one address can store two or more data, we can read or write several data at one time and the memory can operate at a low frequency for the same throughput rate. Meanwhile it implies that the supply voltage can be dropped for a fixed throughput rate and then power dissipation can be further reduced when RAM operates at lower power supply.

Based on radix-2/4/8 algorithm and the new management of on-chip memory, a 128 point FFT has been designed in 0.6 μm 3.3 V triple-metal CMOS process. Since all the stages are on chip, no external memories are required. In section II, we recall the radix-2/4/8 algorithm which is derived from radix-2/8 and present a pipelined VLSI architecture based on the new algorithm. In section III, we will show some analysis results about RAM power consumption and describe the new on-chip memory arrangement. Section IV is the implementation of the 128 point FFT processor. The last section is the conclusions.

II. THE RADIX-2/4/8 FFT ALGORITHM

The N-point discrete Fourier transform (DFT) is presented by

$$C_k = \sum_{n=0}^{N-1} x_n \cdot W^{nk} \quad k=0, 1,..., N-1 \qquad \textit{there}$$

$$W^{nk} = e^{-j \cdot \frac{2\pi}{N} \cdot nk}$$

W^{nk} is called “twiddle factor”. Fig. 1 shows a part of twiddle factors of N-point FFT.

The basic concept underlying the radix-2 algorithm is the use of symmetry between the twiddle factors W^{nk} and $W^{nk+N/2}$ ($W^{nk} = - W^{nk+N/2}$). Another symmetry of twiddle factors ($W^{nk+N/4} = -W^{nk+3N/4} = - jW^{nk}$) has been utilized to minimize the number of complex multiplications by $\pm j$ in radix-4 and split radix algorithms [3][4]. Nevertheless, the symmetry of the twiddle factors $W^{N/8}$, $W^{3N/8}$, $W^{5N/8}$ and $W^{7N/8}$ (in (1) and (2)) has not been effectively utilized. In this section, the new algorithm radix-2/4/8 is discussed, which replaces the complex multiplications ($\frac{\sqrt{2}}{2} \cdot (1 \pm j)$) by 12 additions.

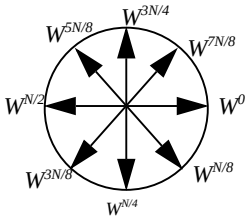


Fig. 1 A part of twiddle factors of N-point FFT

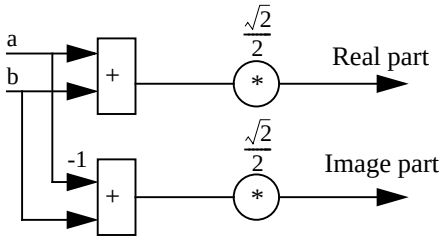


Fig. 2 A simplified complex multiplication with $W^{N/8}$

$$W^{N/8} = -W^{5N/8} = \frac{\sqrt{2}}{2} \cdot (1 - j) \qquad (1)$$

$$W^{3N/8} = -W^{7N/8} = -\frac{\sqrt{2}}{2} \cdot (1 + j) \qquad (2)$$

A.Complex Multiplications by $\frac{\sqrt{2}}{2} \cdot (1 \pm j)$

In the process of computing FFT, there are many multiplications by the twiddle factors $W^{N/8}$, $W^{3N/8}$,$W^{5N/8}$ and $W^{7N/8}$. The complex multiplications with these twiddle factors are expressed in (3) and (4):

$$(a + jb) \times W^{N/8} = -(a + jb) \times W^{(5N)/8}$$

$$= (a + jb) \times \left(\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}\right) = \frac{\sqrt{2}}{2}((a + b) + j(b - a)) \qquad (3)$$

$$(a + jb) \times W^{(3N)/8} = -(a + jb) \times W^{(7N)/8}$$

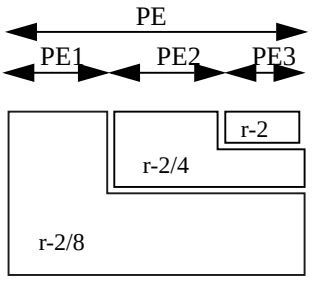
$$= (a + jb) \times \left(-\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}\right) = \frac{\sqrt{2}}{2}((b - a) - j(b + a)) \qquad (4)$$

So, this kind of complex multiplication (include four real multiplications and two additions using traditional complex multiplication) can be realized by two real multiplications and two additions. Fig. 2 is a simplified complex multiplication with $W^{N/8}$.

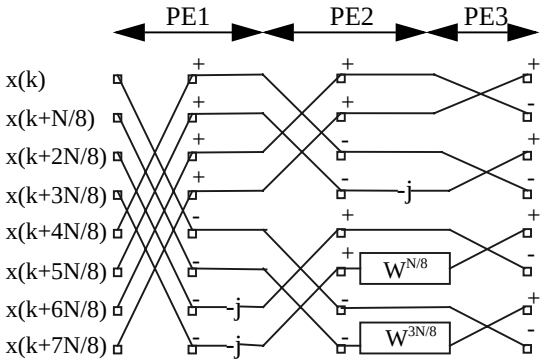
In addition, the multiplication by $\frac{\sqrt{2}}{2}$ can be further reduced to additional shift operations to create a multiplication free operation (see (5)).

$$\frac{\sqrt{2}}{2} = 0.70710678 = 2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-9} \qquad (5)$$

It is clear that the multiplication by $\frac{\sqrt{2}}{2}$ can be expressed



(a) Simplified signal-flow for a process element (PE)



(b) Butterflies for a process element (PE)

Fig. 3 The basic process element (PE) for radix-2/4/8 algorithm

by 5 additions. As a result, the complex multiplications by $W^{N/8}$,$W^{3N/8}$, $W^{5N/8}$ *and* $W^{7N/8}$ (see (3) and (4)), each of which has four real multipliers and two additions, can be replaced with 12 additions.

B. The Radix-2/4/8 Algorithm

A new regular algorithm is developed from radix-2/8, which takes the advantage of radix-2, radix-2/4 and radix-2/8, so we call it "radix-2/4/8" algorithm. This algorithm minimizes the number of complex multiplications with $\pm j$ and $\frac{\sqrt{2}}{2} \cdot (1 \pm j)$, thus minimizing the number of non-trivial multiplications required for the computation of a length 8^n FFT. The new algorithm requires less complex multiplications than traditional radix-2, radix-4 and split radix (radix-2/4) algorithms, at the cost of additions. This new algorithm improves the computational efficiency. On the other hand, it is more regular compared to split radix algorithm or radix-2/8 algorithm. It is easy to be implemented in VLSI, especially in a pipelined architecture.

In the radix-2 or radix-4 algorithm, the radix-2 or radix-4 butterfly is employed as the basic arithmetic unit. But in this new radix-2/4/8 algorithm, the basic arithmetic unit, called "process element" (PE), consists of three parts: PE1, PE2 and PE3. Fig. 3 (a) shows the simplified signal flow

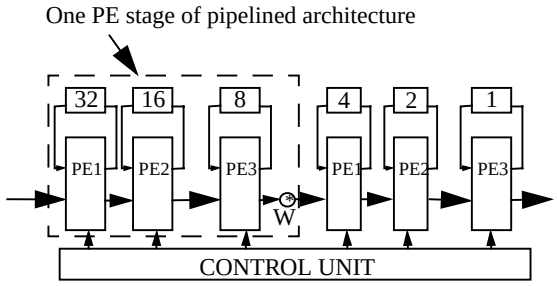


Fig. 4 A 64-point radix-2/4/8 pipelined architecture

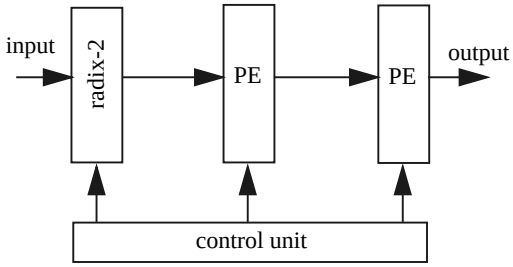


Fig. 5 Radix-2/4/8 pipelined architecture for 128 point FFT

for a process element (PE) used in radix-2/4/8 algorithm. In each of PE the radix-2/8, radix-2/4 and radix-2 butterflies are employed. Fig. 3 (b) is corresponding butterflies inside one PE.

C. Pipelined Architecture

Pipelined architecture is the best choice for high throughput applications. Due to its regular structure and relatively simple control, it is the best choice to implement high speed long size FFT. Several pipelined architectures have been developed, such as Multi-path Delay Commutator (MDC), Single-path Delay Feedback (SDF) and Single-path Delay Commutator (SDC) [5][6][7]. SDF architecture was taken as a basis for our implementation.

Based on the radix-2/4/8 algorithm, an improved SDF architecture is designed. This architecture is similar with R2SDF pipelined architecture [2], but it uses radix-2/4/8 algorithm instead of radix-2 as the basic arithmetic unit. Fig. 4 shows a 64-point radix-2/4/8 SDF architecture.

The architecture allows designers to speed up the circuits by some pipelining of the arithmetic units, without complex modification and it can be easily modified for length = 8^n *(n=1,2,3,...)*, bypassing or cascading some of PE units. Because 128 is not the power of 8, the 128-point FFT chip is made of a radix-2 stage and two PE stages (each PE including PE1, PE2 and PE3), see Fig. 5.

In next section we'll describe a new memory management for lower power consumption.

Word Depth	Word Width (bits)						
	8	12	16	20	24	28	32
64	0.6	0.7	0.7	0.8	0.9	1.0	1.1
128	0.7	0.8	0.9	1.0	1.0	1.1	1.2
256	0.8	1.0	1.1	1.3	1.4	1.5	1.6
512	1.1	1.3	1.6	1.8	1.9	2.1	2.3
1024	1.5	1.9	1.6	1.8	1.9	2.1	2.3
2048	1.5	1.9	1.6	1.8	1.9	2.1	2.3

Table 1: Power dissipation (mW/MHz) of RAM cells with different word width and word depth (Power supply: 3.6 V)

III. ANALYSIS OF RAM POWER CONSUMPTION

We have analysed a typical 0.6 μm 3.3 V CMOS asynchronous RAM library and summarized the relationship between power dissipation and word width/depth in table 1 [8]. Table 1 clearly shows that the power dissipation of RAM cell will increase with its word width but will not increase proportionally with its word width. For example, to a RAM cell with word depth of 1024, if its word width increase 300% (from 8 bits to 32 bits) its power dissipation only increase 54% (from 1.5 mW/MHz to 2.3 mW/MHz).

Based on above analysis of RAM power consumption if we increase word width of RAM and decrease word depth of RAM (i.e., keep the size of RAM as a constant) more data can be read/written from RAM at one time, which means RAM can work at a lower frequency but can maintain the same throughput rate. Because the power dissipation of RAM is proportional to its working frequency, its power dissipation can be reduced by increasing its word width as above. For example, assuming the data width is 8 bits in a digital system. To a RAM cell with word depth of 1024 and word width of 8 bits, if it operates at 50 MHz its power dissipation will be 1.5 mW/MHz * 50 MHz = 75 mW; if its word width increases to 16 bits (one address stores two data) and word depth decreases to 512 and it is clocked at 25 MHz for the same throughput, its power dissipation will be 1.6 mW/MHz * 25 MHz = 40 mW; if its word width increases to 32 bits (one address stores four data) and word depth decreases to 256 and it is clocked at 12.5 MHz for the same throughput, its power dissipation will be 1.6 mW/MHz * 12.5 MHz = 20 MW, see Fig. 6. So if one address of RAM stores four data the power dissipation of RAM can decrease about 70%. With the same throughput RAM working frequency can decreased 75% as well. In the FFT design we use such kind of RAM man-

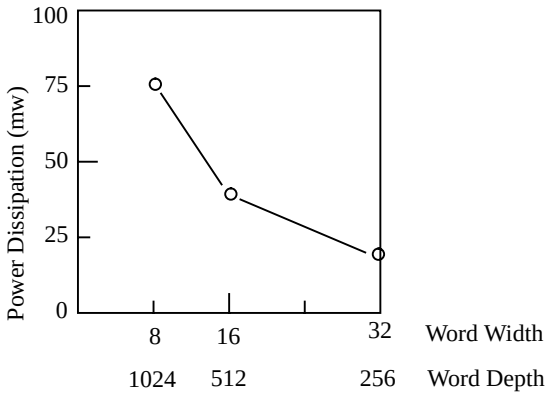


Fig. 6 Power dissipation of RAM cells with same size but with different word width/depth at same throughput rate

agement to save the power consumption of on-chip RAMs.

Meanwhile operating at a lower frequency implies that the power supply of RAM can be decreased for a fixed throughput and then power dissipation of RAM can be further reduced. The VDD source voltage has significant impact on the power dissipation in CMOS process because the power consumption is proportional to the VDD^2. In our library, the RAM can work from 2.5 V to 3.6 V. If the power supply of RAM decreases from 3.6 V to 2.5V it means the power consumption of RAM can be reduced 50%.

IV. IMPLEMENTATION

The 128 point FFT chip was designed in a 0.6 μm 3.3 V triple-metal CMOS process. The PE blocks are generated by using Compass datapath compiler; the control logic has been designed by synthesizing a VHDL description and then mapping it to logic level with standard cells; the on-chip RAM and ROM cells are generated from Compass RAM/ROM compiler. Input data and internal coefficients are all 2*10 bits (include real part and image part) and internal results and output data are 2*12 bits. The chip features have been obtained after careful simulation and layout design. The chip size is 10 mm^2. The chip can compute a 128 point FFT every 3 μs. Its power consumption is 400 mW when input frequency is 50 MHz.

V. CONCLUSIONS

In this paper, we recalled our VLSI-oriented FFT algorithm - radix-2/4/8, which removes the complex multiplications with $\frac{\sqrt{2}}{2}\cdot(1\pm j)$, at a cost of additions. The advantage of this algorithm is its ability of decreasing the number of complex multiplications and its regularity. Reduced number of multiplications implies reduced power consumption as well as decreased hardware circuit complexity. The regularity means efficient implementation in VLSI. On the other hand, the new algorithm can be easily implemented for different length FFT without complex modification. A new management of on-chip memory further reduce the chip power consumption. This new management of on-chip RAM can also be utilized in other memory-intensive system design.

REFERENCES

[1] Lihong Jia, Yonghong Gao, Jouni Isoaho and Hannu Tenhunen, “A New VLSI-Oriented FFT Algorithm and Implementation”, accepted by 11th Annual 1998 IEEE International ASIC Conference, September 1998, Rochester, New York, USA.

[2] P. Duhamel and H.Hollmann, “Split-radix FFT algorithm”, *Electronic Letters*, vol.20, No.1, pp.14-16, Jan., 1984

[3] M.Vetterli and P.Duhamel, “Split-radix algorithm for length - p^m DFT’s”, *IEEE Trans. on Acoustic, speech, signal processing*”, vol.37, No.1, pp.57-64, Jan., 1989.

[4] M.A.richard, “On hardware implementation of the split-radix FFT”, *IEEE Trans. on Acoustic, speech, signal processing*”, vol.36, No.10, pp.1575-1581, Oct.,1989.

[5] E.H. Wold and A.M.Despain, “Pipeline and parallel-pipeline FFT processors for VLSI implementations”, *IEEE Trans. on Computers*, vol.C-33, No.5, pp.414-426, May, 1984.

[6] J.A. Johnston, “Parallel pipeline fast Fourier transformer”, *Proc. IEE pt.* F. Vol. 130, pp564-572, 1983

[7] J.Melander, “Design of SIC FFT architectures”, LiU-Tel-Lic-1997:19, Linkoping university, May, 1997.

[8] “Manual of COMPASS 0.6 μm 3.3 V CMOS Libraries”, COMPASS On-line Documents.