K. Umapathy, D. Rajaveerappa / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.293-296

Area Efficient 128-point FFT Processor using Mixed Radix 4-2 for OFDM Applications

K. UMAPATHY[1] D. RAJAVEERAPPA[2]
[1]Research Scholar, JNT University, Anantapur, India
[2]Professor, Department of ECE, Loyola Institute of Technology, Chennai, India

Abstract

This paper presents a single-chip 128-point FFT processor based on the cached-memory architecture (CMA) with the resource Mixed Radix 4-2 computation element. The 4-2-stage CMA, including a pair of single-port SRAMs, is introduced to increase the execution time of the 2-dimensional FFT's. Using the above techniques, we have designed an FFT processor core which integrates 5,52,000 transistors within an area of 2.8 x 2.8 mm^2 with CMOS 0.35µm triple-layer-metal process technology. This processor can execute a 128-point, 36-bit-complex fixed-point data format, 1-dimensonal FFT in 23.2µsec and 2 dimensional FFT in only 23.8 msec at 133 MHz clock operation.

Keywords: FFT, CMA (cached memory architecture), OFDM, Mixed Radix4-2

1. Introduction

The Fast Fourier transform (FFT) transmutes a set of data from the time domain to the frequency domain or vice versa. Since it saves a considerable amount of computation time over the conventional discrete Fourier transform (DFT) method the FFT is widely used in various signal processing applications. The earlier implementations of the FFT algorithms have been mainly used for running software on general-purpose computers. But with the increasing demands for high-speed signal processing applications, the efficient hardware implementation of FFT processors has become an important key to develop many future generations of advanced multi-dimensional digital signal and image processing systems.

This paper describes a small-area, high-performance 128-point FFT processor using both the cached-memory architecture (CMA) and the double buffer structure. The CMA significantly reduces the hardware resource of the computation element when compared to single-memory or dual-memory architecture, even if the radix number is high. A high-performance, multi-dimensional FFT is achieved using the 2-stage CMA based on the double buffer structure.

2. Cached-Memory Architecture

Since FFT makes frequent accesses to data in memory, it can be calculated in O ($\log_r$ N) stages, where N is the number of point of the transform and r is the radix number of the FFT decomposition. In FFT, each stage requires the reading and writing of all N data words. So higher-radix decomposition is required to increase the FFT execution time. The proposed CMA performs a high-radix computation with low-power and high-performance.

Figure 1 shows the block diagram of system using the CMA. It is similar to single-memory architecture except for a small cache memory comprising a pair of registers between the computation element and the main memory. This pair of registers enables hiding the memory access cycle behind the computation cycle shown in Figure 2. This tightly-coupled processor-cache pair increases the effective bandwidth to the main memory during the mixed radix-4-2 execution thereby avoiding the processor to access the main memory often.

The advances in semiconductor technology have not only enabled the performance and integration of FFT processors to increase steadily, but also led to the requirement of the higher-radix computation. Assuming that the computation element has one mixed radix 4-2 execution unit, then we have the following equations-

Memory Access Cycle = r2 (1)

Execution Cycle = r [$\log_r$ N] / 2 (2)

Figure 1. Block diagram of the System using CMA.

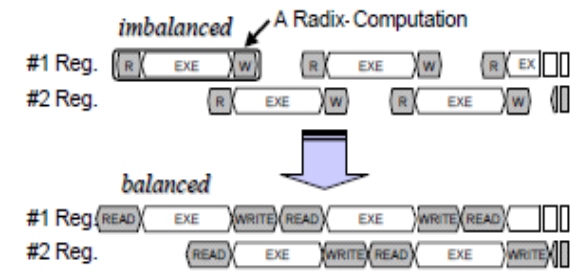

Figure 2. Operation sequence for each register. It is required to keep a balance between the memory access cycle and the execution cycle to achieve 100% efficiency.

Equations (1) and (2) show that the higher the radix number, the smaller will be the ratio of the execution cycle to the memory access cycle. This shows that with increasing the number of radix number, it takes more time for execution than accessing the memory and this causes the imbalance as shown in Figure 2. To solve this problem, multiple computation units with multi-datapath are required. Hence we have developed the resource-saving multidatapath computation element without any interconnection bottleneck caused by large crossbar, bus, or network structure connected to partitioned memories.

3. Mixed Radix 4-2 Computation Element

In the Decimation-in-time (DIT) Mixed Radix-4-2 FFT algorithm, an 8-word transformation can be done by a 4-row x 3-column Mixed Radix-4-2 butterfly operations shown in Figure 3(a). Since the butterfly execution unit, which is composed of a complex multiplier, adder and subtractor, is huge, it is very important to reuse the computation resources in order to shrink the size of the chip. To achieve this, we have developed the resource-saving Mixed Radix 4-2 computation element (RM-R4-2CE).

Since each column is composed of 4 independent butterflies and data flows in a consistent pattern in the 8-word groups throughout the entire 128-word transform, this symmetry makes it possible to divide the normal 8-word computation into 3 passes as shown in Figure 3(b). Each pass is processed by a single RM-R4-2CE consisting of Mixed Radix 4-2 butterfly execution units. Using the RM-R4-2CE, all the communication can be easily sorted and stored in the register closer to the following butterfly unit by changing the address of the data stored in the cache one after the other. In this manner, RM-R4-2CE can reduce the processor area down to 1/3 and half of the interconnection area without any delay penalties.

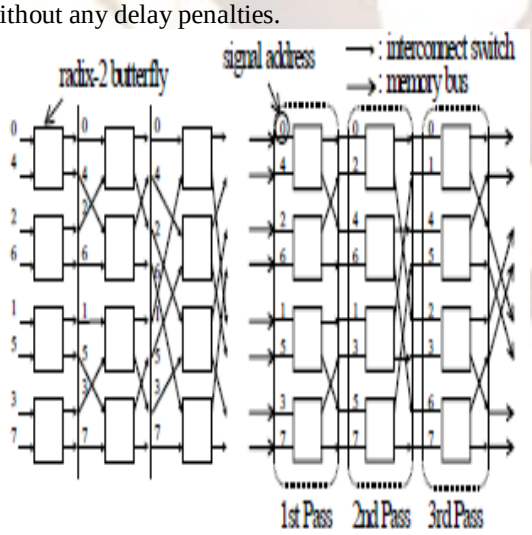

Figure 3. Dataflow diagram of the 8-word group
(a) Radix 2 decomposition
(b) Computation element.

Figure 4 shows the block diagram of the RM-R4-2CE with a pair of registers. In conventional FFT processors the deep pipeline architecture is adopted. But in this RM-R4-2CE, every pass is processed cyclically among the processor and the cache. Hence none of the additional pipeline registers and complex controllers is required. Thus this simple architecture avoids any processor stalls or memory access bottleneck caused by the data dependence.
The RM-R4-2CE has resource-saving routing switches for transmitting the output signals to the appropriate registers for the following pass processing. When the "select" signal shown in Figure 3 is "0", the 1st and 3rd pass configuration will be realized on RM-R4-2CE, and if "1", it is configured for the 2nd pass.

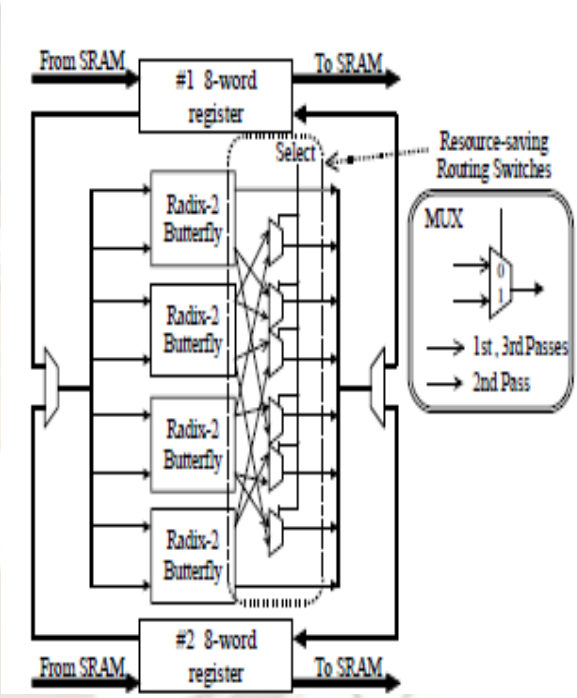

Figure 4. Block diagram of RM-R4-2CE with a pair of registers.

According to the effect of multidatapath, the execution cycle becomes non- critical. Hence we have implemented the complex multiplier in the butterfly execution unit as small as possible. Table 1 and 2 shows the evaluation of the gate count of both the multiplier and the adder. From these results, we have decided to use the array multiplier and the ripple carry adder, which can reduce the area of the complex multiplier, comprising 4 multipliers, 1 adder and 1 subtractor by about 39% when compared to the one which uses the modified Booth-2 encoded Wallace tree multiplier and the carry look ahead adder as shown in Table 3.

Table 1. Evaluation of the multiplier

Type of multiplier (18-bit width, 2's complement)	Number of gates	Gate delay [nsec]
Array	1,962	24.8
Wallace Tree	2,145	9.7
Modified Booth-2 encoded Wallace Tree	3,019	6.3

Table 2. Evaluation of the adder

Type of adder (18-bit width, 2's complement)	Number of gates	Gate delay [nsec]
Ripple Carry	95	7.5
Carry Look Ahead	128	1.6

Table 3. Evaluation of the complex multiplier

Combination of the multiplier and the adder	Number of gates	Gate delay [nsec]
Array & Ripple Carry	8,038	32.3
Modified Booth-2 encoded Wallace Tree & Carry Look Ahead	13,000	7.9

From the simulation waveforms of the cache register, data are serially read from the main memory to the register, processed in parallel, and written back to the main memory serially. During execution, data is circulated 3 times around the RM-R4-2CE and the register of one side. Then the 3-stage pass processing is finished within the same time as that of 8-word data read/write time. This clearly indicates that the RM-R4-2CE converts the data properly without any delay penalties in spite of the decrease in the number of butterfly elements down to 1/3 or the interconnect resources down to one half. It suggests that the proposed RM-R4-2CE is well suited for the FFT calculations based on the CMA.

On the other way, the processor must be stalled and forced to wait for data in the original CMA without the double buffer structure. This is important for the high-speed and small-area 2-dimensional FFT processor where a series of 1-dimensional FFTs must be executed. The dual-port SRAM is sometimes used to increase the data path width by pipeline FFT processors. Since dual-port SRAM occupies about double the area in comparison to the single-port SRAM, it is difficult to adopt the double buffer structure. However the CMA with RM-R4-2CE can achieve 100% efficiency even with the single-port SRAM configuration. This shows that the combination of the CMA, RM-R4-2CE and double buffer structure is extremely well suited for the proposed FFT processor.

4. Chip Design & Results

Figure 5 shows the die plot of the specially designed 2-stage CMA FFT processor using RM-R4-2CE. This processor has been designed with 0.35µm, 3-layer-metal CMOS technology and occupies an area of only 7.84 mm2 which contains 1,92,000 transistors for logic and 3,60,000 transistors for SRAM in 3 blocks. While 2 of these blocks are assigned for data, the other one stores the coefficients of the twiddle factor.

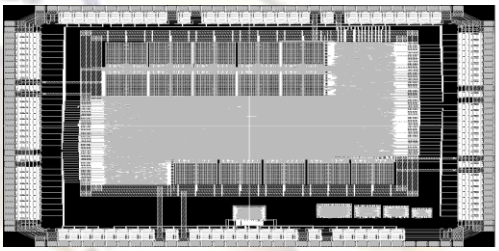

Figure 5. Die plot of the proposed FFT processor.

Table 4 shows the key features of the 2-stage CMA FFT processor in comparison with the previous works. This table shows that our proposed FFT processor is superior in terms of the data path width, the 2-dimensional FFT calculation speed, and especially the area of the silicon even though the difference in technology is taken into account. In this table, we assume constant throughput in the performance estimation. Moreover, the twiddle factor is stored not in the ROM but in the coefficients of SRAM. This makes it possible to perform the adaptive number of point FFT, which means more than 128-point FFT's.

Table 4. Key features of the proposed 128-point FFT processor

Parameters	Proposed FFT	Spiffee	NTT LSI Lab
Technology	0.35µm	0.7µm (L_{poly}=0.6µm)	0.8µm
Number of Transistors	Logic : 192k SRAM : 360k Total : 552k	460k	380k
Data path Width	36x8-bit fixed point	20-bit fixed point	24-bit floating
Area	7.84 mm^2	49 mm^2	134 mm^2
Clock Freq.	133MHz	173MHz	40MHz
128-point FFT	23.2µsec	15µsec	54µsec

5. Conclusion

A single-chip 128-point FFT processor is presented based on the CMA with the resource-saving Mixed Radix-4-2 computation element. The 2-stage CMA, including a pair of single-port SRAMs, is also introduced to speed up the execution time of the 2-dimensional FFT's. Using above techniques, we have designed an FFT processor core which integrates 5,52,000 transistors in area of 2.8 x 2.8 mm^2 with CMOS 0.35µm triple layer metal process technology. This processor can execute a 128-point, fixed-point 36-bit complex data format, 1-dimensonal FFT in 23.2 µsec and 2-dimensional FFT in only 23.8 msec at 133MHz operation. With this FFT core, we can develop many future generations of advanced multi-dimensional Digital signal and Image processing systems onto the system chip.

References

[1] M. K. A. Aziz, P. N. Fletcher, and A. R. Nix, "Performance analysis of IEEE 802.11n solutions combining MIMO architectures with iterative decoding and sub-optimal ML detection via MMSE and zero forcing GIS solutions," IEEE Wireless Communications and Networking Conference,2004, vol. 3, pp. 1451–1456, Mar. 2004.

[2] A. Ghosh and D. R. Wolter, "Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential," IEEE Communications Magazine, vol. 43, no. 2, pp. 129–136, Feb. 2005.

[3] S. Yoshizawa, K. Nishi, and Y. Miyanaga, "Reconfigurable two- dimensional pipeline FFT processor in OFDM cognitive radio systems,"in IEEE International Symposium on Circuits and Systems, (ISCAS '08),May 2008, pp. 1248–1251.

[4] L. G. Johnson, "Conflict free memory addressing for dedicated FFT hardware," IEEE Trans. Circuits Syst. II, Analog Digit. Signal rocess.,vol. 39, no. 5, pp. 312–316, May 1992.

[5] J. A. Hidalgo, J. Lopez, F. Arguello, and E. L.Zapata,"Area-efficient architecture for fast Fourier transform," IEEE Trans. Circuits Syst. II,Analog Digit. Signal Process., vol. 46, no. 2, pp. 187–193, Feb. 1999.

[6] B. G. Jo and M. H. Sunwoo, "New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy," IEEE Trans. Circuits yst. I,Reg. Papers, vol. 52, no. 5, pp. 911–919, May 2005.

[7] Z.-X. Yang, Y.-P. Hu, C.-Y. Pan, and L. Yang, "Design of a 3780-point IFFT processor for TDS-OFDM," IEEE Trans. Broadcast., vol. 48, no. 1,pp. 57–61, Mar. 2002.

[8] 3GPP TS 36.201 V8.3.0 LTE Physical Layer—General Description,E-UTRA, Mar. 2009.

[9] E. Oran Brigham, The fast Fourier transform and its applications, Prentice-Hall, 1988.

[10] Se Ho Park et.ul. "A 2048 complex point FFT architecture for digital audio broadcasting system", In Proc. ISCAS 2000.

[11] Se Ho Park e t d . "Sequential design of a 8192 complex point FFT in OFDM receiver", In Pvoc. AP-ASIC, 1999.