ELectronics EXpress

IEICE Electronics Express, Vol.5, No.24, 1049–1053

Area and power efficient signal reordering unit for OFDM systems

Jong-Yeol Lee[a)] **and Seung-Won Yang**

Chonbuk National University,

664–14, Duckjin-Dong, Jeonju 561–756, South Korea

a) *jong@chonbuk.ac.kr*

Abstract: An area and power efficient signal reordering unit (SRU) for OFDM systems that reorders the bit-reversed output of an FFT unit into linear order is proposed. By using a FIFO that is smaller than an N-word memory used in conventional SRUs, the proposed SRU achieves about 25% reductions in both the area and the power consumption in case of a 1024-point FFT.

Keywords: FFT, SRU, OFDM, FIFO

Classification: Science and engineering for electronics

References

[1] X. Li, Z. Lai, and J. Cui, "A Low Power and Small Area FFT Processor for OFDM Demodulator," *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 274–277, May 2007.

[2] J. Wu, K. Liu, B. Shen, and H. Min, "A Hardware Efficient VLSI Architecture For FFT Processor in OFDM Systems," *Proc. Int. Conf. ASIC*, pp. 232–235, Oct. 2005.

[3] N. Weste and D.J. Skellern, "VLSI for OFDM," *IEEE Commun. Mag.*, pp. 127–131, Oct. 1998.

1 Introduction

A signal reordering unit (SRU) in orthogonal frequency division multiplexing (OFDM) systems reorders the bit-reversed output signal of a fast Fourier transform (FFT) or an inverse FFT (IFFT) unit into linear order. Due to the increasing demand of mobile applications, the area and power efficient implementation of communication systems utilizing OFDM, such as IEEE 802.11 and IEEE 802.16e, is a critical issue. For the low power implementation of OFDM systems, some previous works have focused on the low power implementation of IFFT/FFT by reducing the computation memory which stores the immediate results because an FFT processor dominates the area and the power consumption in OFDM implementations [1, 2]. The authors of [1] developed a distributed memory architecture to meet the requirement of non-stopping and high-speed data throughput. In [2], the authors reduced

DOI: 10.1587/elex.5.1049
Received October 08, 2008
Accepted November 25, 2008
Published December 25, 2008

the memory by using a data access scheme based on alternation between linear order addressing and bit-reversed order addressing. However, no previous works dealt with the SRU whose area and power consumption are as large as the half of those of an FFT unit [3]. This letter proposes an area and power efficient SRU that can be used in OFDM systems by reducing memory requirement.

2 Signal reordering unit

Since an IFFT/FFT unit in an OFDM system produces output in bit-reversed order with linear order input, an SRU reorders the bit-reversed output into linear order so that the transmitted and the received data sequences are in correct order. In order to ensure continuous data flow, the conventional SRU for N-point FFT has a ping-pong structure which consists of two N-word memories as shown in Fig. 1 (a), where while one data frame is written to one of the memories, the previous frame is read from the other memory. The SRU in Fig. 1 (a) operates in two cycles. In the first cycle, the write enable signal of memory 0 is asserted and the IFFT/FFT result is written in bit-reversed order while output data are read from memory 1 in linear order. After reading one data frame in the first cycle, the second cycle begins with asserting the write enable signal of memory 1 and output data are read from memory 0 in linear order.

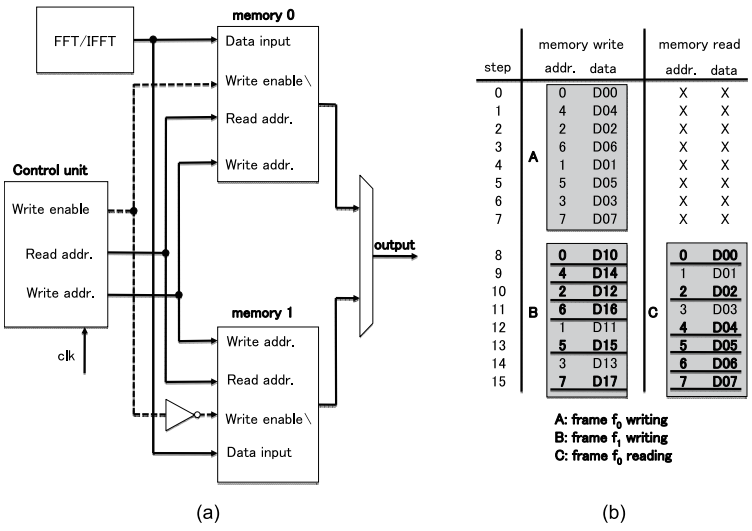

Fig. 1. (a) Block diagram and (b) operation of a conventional SRU with a ping-pong structure

3 Proposed SRU

One N-word memory in the ping-pong structure shown in Fig. 1 (a) can be eliminated by using the relationship between write addresses and read addresses. To eliminate one N-word memory, a dual-port N-word memory

DOI: 10.1587/elex.5.1049
Received October 08, 2008
Accepted November 25, 2008
Published December 25, 2008

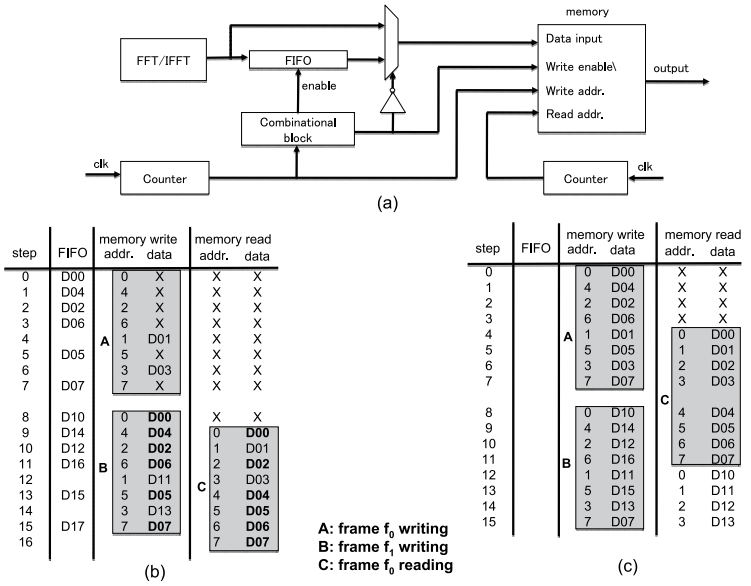

Fig. 2. (a) Block diagram of the proposed SRU, (b) operation before FIFO size reduction and (c) operation after FIFO size reduction

that can support one read operation and one write operation at the same time must be used because data writing and reading must be performed on one memory concurrently. As can be seen in Fig. 1 (b), if only one N-word memory is used, some data in frame f_0 are overwritten by data in frame f_1 before they are read because frame f_1 is stored in bit-reversed order and frame f_0 is read in linear order. In Fig. 1 (b) the data in locations with addresses 0, 2, 4, 5, 6 and 7 are overwritten before they are read.

The overwrite problem can be solved by exploiting a first-in-first-out (FIFO) buffer which is placed in front of the memory and temporarily stores the data in frame f_{j+1} which overwrite the data in the previous frame f_j as shown in Fig. 2 (a), which shows the block diagram of the proposed SRU. The data that overwrite the previous frame data are written into the FIFO instead of the memory when frame f_{j+1} is written. When frame f_{j+1} is read from the memory, the data in FIFO are moved into the memory. In the example of Fig. 2 (b), the data whose addresses are 0, 2, 4, 5, 6 and 7 are written in the FIFO when frame f_0 is written. After writing frame f_0, the writing of frame f_1 begins. The data in the FIFO are written into the memory when the data with those addresses in frame f_1 are written into the FIFO. If the reading of frame f_0 begins at step 9 as in Fig. 2 (b), all the correct data can be read from the memory. Since the write address and the read address in each step are not in the relation of bit reversion as shown in Fig. 2 (b), two counters that generate write addresses and read addresses, respectively, are exploited as shown in Fig. 2 (a).

The data to be stored in the FIFO can be found by comparing write addresses with read addresses in Fig. 1 (b). When a write address, AW, is greater than or equal to a read address, AR, in a step, the datum of the address AW must be written in the FIFO. This is because read operations

DOI: 10.1587/elex.5.1049
Received October 08, 2008
Accepted November 25, 2008
Published December 25, 2008

are performed in linear order and hence, the datum of the address AW is overwritten before it is read. For example, in Fig. 1 (b) since the write address and the read address are 4 and 1 in step 9, respectively, the datum of address 4 in the previous frame is read in step 12 after the datum of the same address in the current frame is written in step 9. To avoid overwriting the data of address 4, they should be written in the FIFO.

By using a FIFO, the area and the power consumption can be reduced. The FIFO consumes less power than the N-word memory because the size of the FIFO, S_F, is smaller than N. The number of accesses to the memory also decreases from 2N, N for writing and N for reading, to $N + S_F$ since in the proposed SRU read and write operations are performed at the same time.

4 FIFO size reduction

The size of the FIFO used to store the overwrite data can be reduced by moving the start point of data reading forward because by moving the start point forward, the number of the data which are overwritten before they are read can be reduced. Since the reading of a frame starts before the writing finishes, the start point of data reading should be determined by considering the read-after-write (RAW) data dependency. For example, in Fig. 2 (c) data reading should start after step 3 because the earlier starts cause dependency violations. In this example if reading starts in step 1, a dependency violation occurs because D03 is read in step 4 before it is written in step 6. In Fig. 2 (c), the size of the FIFO is reduced to zero by beginning frame reading at step 4.

5 Synthesis results

To validate the proposed SRU design, we modeled two SRUs for 64-point and 1024-point FFTs used in IEEE 802.11 and IEEE 802.16e, respectively. The modeled SRUs are synthesized using a 0.35 μm standard CMOS process to estimate the area and the power consumption, which are summarized in Table I. In the designed SRUs, FIFOs are 17- and 449-word long for 64-point FFT and 1024-point FFT, respectively. The dual-port N-word memory is generated by using a memory compiler. Although the area of a dual-port memory is larger than that of a single-port memory, the total gate count of the proposed SRU is smaller than that of the conventional SRU as shown in

Table I. Area and power consumption of the proposed SRU

	64-point FFT		1024-point FFT	
	Conventional	Proposed (reduction, %)	Conventional	Proposed (reduction, %)
Memory size (words)	128	81 (36.7)	2048	1473 (28.1)
Area (gate count)	25263.5	15544.4 (38.5)	397774.5	297535.3 (25.2)
Power consumption (mW)	13.00	11.68 (10.2)	239.37	179.53 (25.0)

DOI: 10.1587/elex.5.1049
Received October 08, 2008
Accepted November 25, 2008
Published December 25, 2008

Table I because only one N-word memory is used. The power consumption also decreases because 27% and 44% of the accesses to the N-word memory is changed to the accesses to the FIFO in 64-point FFT and 1024-point FFT, respectively, and the access to the FIFO demands less power than the access to the N-word memory. The amount of the power consumption reduction by eliminating one N-word memory is greater than that of the increment of the power consumption caused by using a dual-port memory and a FIFO. The proposed SRU is more effective for the 1024-point FFT in which case the area and the power consumption are reduced by about 25%.

6 Conclusion

In this letter, an area and power efficient SRU unit for OFDM systems has been proposed. In the proposed SRU, a small FIFO instead of an N-word memory, which is used in the conventional SRU, is exploited to temporarily store the data in a successive frame that overwrite current frame data based on the relationship between write and read addresses. The synthesis results show that the proposed SRU using a FIFO can reduce the area and the power consumption by about 25% in case of a 1024-point FFT.

Acknowledgments

This work was supported in part by the second stage of Brain Korea 21 Project and IDEC.

DOI: 10.1587/elex.5.1049
Received October 08, 2008
Accepted November 25, 2008
Published December 25, 2008