

A DSP Architecture for High-Speed FFT in OFDM Systems

Jaesung Lee, Jeonghoo Lee, Myung H. Sunwoo,
Sangman Moh, and Seongkeun Oh

This paper presents digital signal processor (DSP) instructions and their data processing unit (DPU) architecture for high-speed fast Fourier transforms (FFTs) in orthogonal frequency division multiplexing (OFDM) systems. The proposed instructions jointly perform new operation flows that are more efficient than the operation flow of the multiply and accumulate (MAC) instruction on which existing DSP chips heavily depend. We further propose a DPU architecture that fully supports the instructions and show that the architecture is two times faster than existing DSP chips for FFTs. We simulated the proposed model with a Verilog HDL, performed a logic synthesis using the 0.35 μm standard cell library, and then verified the functions thoroughly.

I. INTRODUCTION

Today, various communication standards have been rapidly developed: WLAN, DTV, Cable modem, WCDMA, CDMA2000, etc. With these systems, after their algorithms have been thoroughly fixed and verified, custom application specific integrated circuit (ASIC) chips have been implemented to reduce their cost, size, and power consumption. However, ASIC-based solutions may be inadequate for adopting various standards since they must be redesigned for each application. With the rapid increase in transistor density, it has become feasible to keep the functionality entirely in a programmable digital signal processor (DSP), allowing much faster changes and upgrades [1].

However, recent DSP technologies have not yet satisfied the requirements of high-speed communication standards. In particular, orthogonal frequency division multiplexing (OFDM) and discrete multitone (DMT) modem systems [2], which are necessary to achieve high-speed data transmission in narrow bands, need to perform several hundred or thousand points of fast Fourier transform (FFT) within a few tens of microseconds. Commercial DSP chips have not yet reached these requirements [3], [4].

High-speed FFT computations may be one of the main research topics for the next generation wire/wireless communications. To meet high-speed FFT computations on DSP chips, this paper proposes instructions and their data processing unit (DPU) architecture which can be embedded as the core in DSP chips. The proposed instructions support new FFT operation flows that are different from the multiply and accumulate (MAC) flow in typical DSP chips. The proposed architecture uses few additional data-path circuits, without

Manuscript received July 12, 2001; revised Aug. 7, 2002.

This work was supported in part by the NRL (National Research Laboratory) program and in part by IDEC (IC Design Education Center).

Jaesung Lee (phone: +82 42 860 5728, e-mail: ljshide@etri.re.kr) and Sangman Moh (e-mail: smmoh@etri.re.kr) are with Computer Architecture Research Team, ETRI, Daejeon, Korea.

Jeonghoo Lee (e-mail: junghoo@madang.ajou.ac.kr), Myung H. Sunwoo (e-mail: sunwoo@madang.ajou.ac.kr), Seongkeun Oh (e-mail: oskn@madang.ajou.ac.kr) are with the School of Electronics Engineering, Ajou University, Suwon, Korea.

modification or addition of the arithmetic units used in the existing DSPs [5]-[12].

We modeled the proposed architecture with Verilog HDL, synthesized it using the HYUNDAI™ 0.35 μm standard cell library with a SYNOPSIS™ tool, and did a timing simulation. The proposed architecture performed the FFT operation flow about 2 times faster than the existing DSP chips [5]-[7] in terms of execution cycles.

The rest of this paper is organized as follows. Section II describes the FFT algorithm and existing DSP-based FFT implementations. Section III presents the proposed instructions and their hardware architecture for high-speed FFT, and section IV discusses implementation and the performance comparisons with the existing DSP chips [5]-[9]. Finally, section V contains concluding remarks.

II. EXISTING DSP-BASED FFT IMPLEMENTATIONS

We first describe an FFT algorithm and existing DSP-based FFT implementations. The radix-2 FFT is represented by (1),

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk} \quad (1)$$

$$= \sum_{n=0}^{N/2-1} x[2n] \cdot W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x[2n+1] \cdot W_{N/2}^{nk},$$

where $W_N^k = e^{j2\pi k/N}$ is the complex twiddle factor. This equation is computed by repeating the radix-2 butterfly operation [13]. Figure 1 shows how to compute the above butterfly with its DPU units on general DSP chips [14].

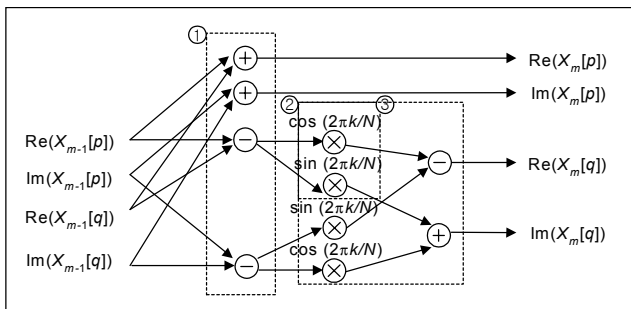


Fig. 1. The flow graph of the radix-2 butterfly on general DSP chips.

In Fig. 1, ①, ②, and ③ represent the first, second, and third clock cycles, respectively, when one dual MAC instruction, which is generally used on existing DSP chips, is performed in one clock cycle. Because four multiplications are required, if we have two multipliers, then two clock cycles (i.e., cycles ② and ③) are needed. A deeper pipeline of the DPU

can make a higher operating clock frequency. However, the computation of one butterfly requires the same clock cycles even if the operating clock frequency varies.

The latest DSP chips have one dual-MAC and four or more ALU units in their DPUs [5]-[7], [9]-[12]. Accordingly, the DSP chips mainly depend on dual-MAC units with some adders/subtractors for computing the FFT butterfly. Because multipliers are larger than other arithmetic units, the number of multipliers is limited to two or four in the DPU [8]. Consequently, the flow graph in Fig. 1 is the appropriate solution for computing the FFT on dual MAC-based DSP chips.

This section presents new FFT instructions based on the enhanced complex multiplication [15], the proposed operation flows, and their new DPU architecture. If (2) and the flow graph in Fig. 2 for complex multiplications [15] are used, then the flow graph of the general complex-multiplication in Fig. 1 can be replaced. Note here that, $Xr = \text{Re}(X_{m-1}[p]) - \text{Re}(X_{m-1}[q])$ and $Xi = \text{Im}(X_{m-1}[p]) - \text{Im}(X_{m-1}[q])$ in (2).

$$\text{Re}(X_m[q]) = Xr \left[\cos\left(\frac{2\pi k}{N}\right) + \sin\left(\frac{2\pi k}{N}\right) \right] - \left[(Xr + Xi) \sin\left(\frac{2\pi k}{N}\right) \right]$$

$$\text{Im}(X_m[q]) = Xr \left[\cos\left(\frac{2\pi k}{N}\right) - \sin\left(\frac{2\pi k}{N}\right) \right] + \left[(Xr + Xi) \sin\left(\frac{2\pi k}{N}\right) \right] \quad (2)$$

In Fig. 2, one addition is performed first and then three multiplications are performed. Finally, one addition and one subtraction complete the complex multiplication. This scheme requires only three multiplications instead of the four in Fig. 1.

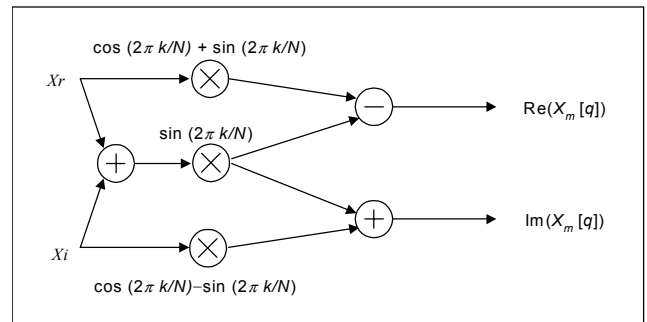


Fig. 2. The flow graph of the enhanced complex-multiplication.

III. THE PROPOSED DSP INSTRUCTIONS AND THEIR ARCHITECTURE

The flow graph shown in Fig. 3 can be obtained for two radix-2 butterflies using the complex multiplication of Fig. 2. In Fig. 3, ①, ② and ③ represent the first, second, and third clock cycles, respectively, and they represent two radix-2

butterflies. As shown in Fig. 3, the number of arithmetic operations in the flow graph can be optimized because the number of multiplications is six. In contrast, the two radix-2 butterflies using the scheme in Fig. 1 require eight multiplications.

The flow graph in Fig. 3 needs new instructions different from the MAC instruction to perform an addition first and then a multiplication next as in cycle ① or ②. To fulfill this requirement, we propose the new add and multiply (AMPY) instruction, which is a one cycle instruction. Since two AMPYs can be executed in parallel, we used dual AMPY instructions. The consecutive-ADD instruction that performs one addition after two subtractions by the dual AMPY instruction in one cycle is needed to perform cycle ① or ②. The multiply and double-accumulate (MDAC) instruction performs one addition and one subtraction after one multiplication. Since two MDAC instructions are executed concurrently in cycle ③, we used dual MDAC instructions.

If another scheme (Fig. 4) is used, the add and double-multiply (ADMPY) instruction is needed to perform two multiplications after one subtraction as in cycle ①. In cycle ②, we need the add and dual MAC (ADMAC) instruction that performs an addition first and then a dual MAC operation next.

The scheme using the dual-AMPY instruction and the dual-MDAC instruction performs two butterflies in 3 cycles (Fig. 3), and thus, it takes 1.5 cycles per one radix-2 butterfly. The other scheme using the ADMPY instruction and the ADMAC instruction takes 2 cycles per one radix-2 butterfly (Fig. 4).

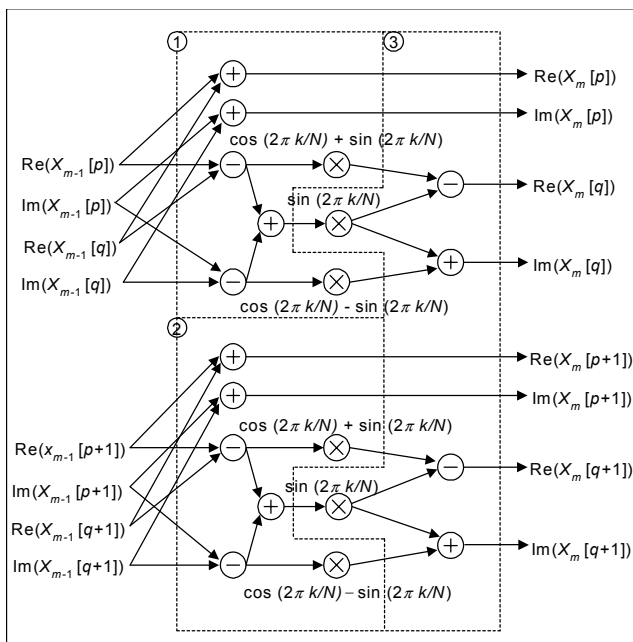


Fig. 3. The proposed flow graph of two radix-2 butterflies using the enhanced complex multiplication.

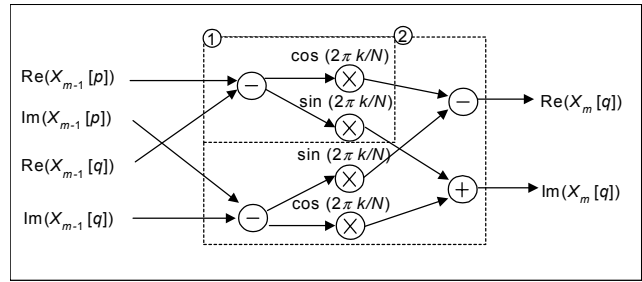


Fig. 4. The proposed flow graph using the AMAC instruction.

These instructions can be used along with other instructions using parallel ‘||’ notations as in very long instruction word DSP chips [7].

To perform the new instructions and other operations in Figs. 3 and 4 efficiently, the existing DPU architectures must be modified. However, we need neither to append more arithmetic units in existing DPUs, nor to modify the internal architecture of the typical arithmetic units (adders or multipliers) themselves.

Figure 5 depicts the proposed DPU architecture which can support dual AMPY, consecutive-ADD, dual MDAC, ADMPY, and ADMAC instructions, as well as arithmetic instructions as in general DSP chips.

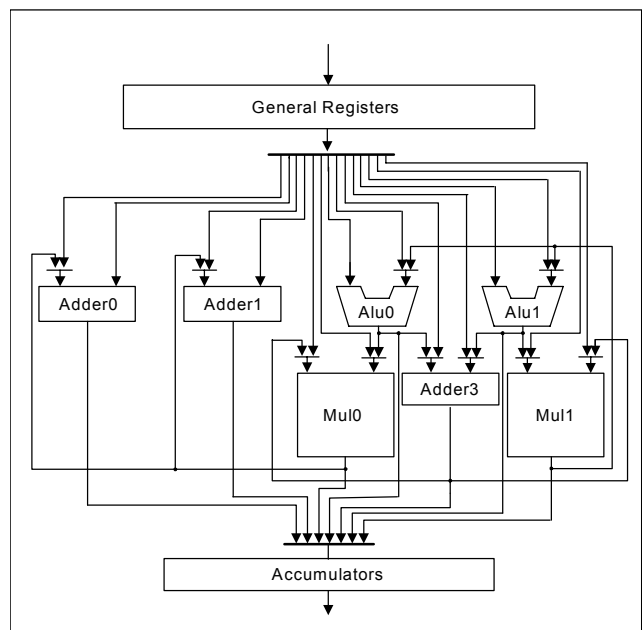


Fig. 5. The proposed DPU architecture.

The dual AMPY instructions in Fig. 5, which perform cycle ① or ② in Fig. 3, are executed using Adder1, Adder2, Adder3, Alu0, and Alu1, while the consecutive-ADD instruction that performs cycle ③ in Fig. 3 is executed using Alu0, Alu1, and Adder3. Cycle ① in Fig. 4 is performed

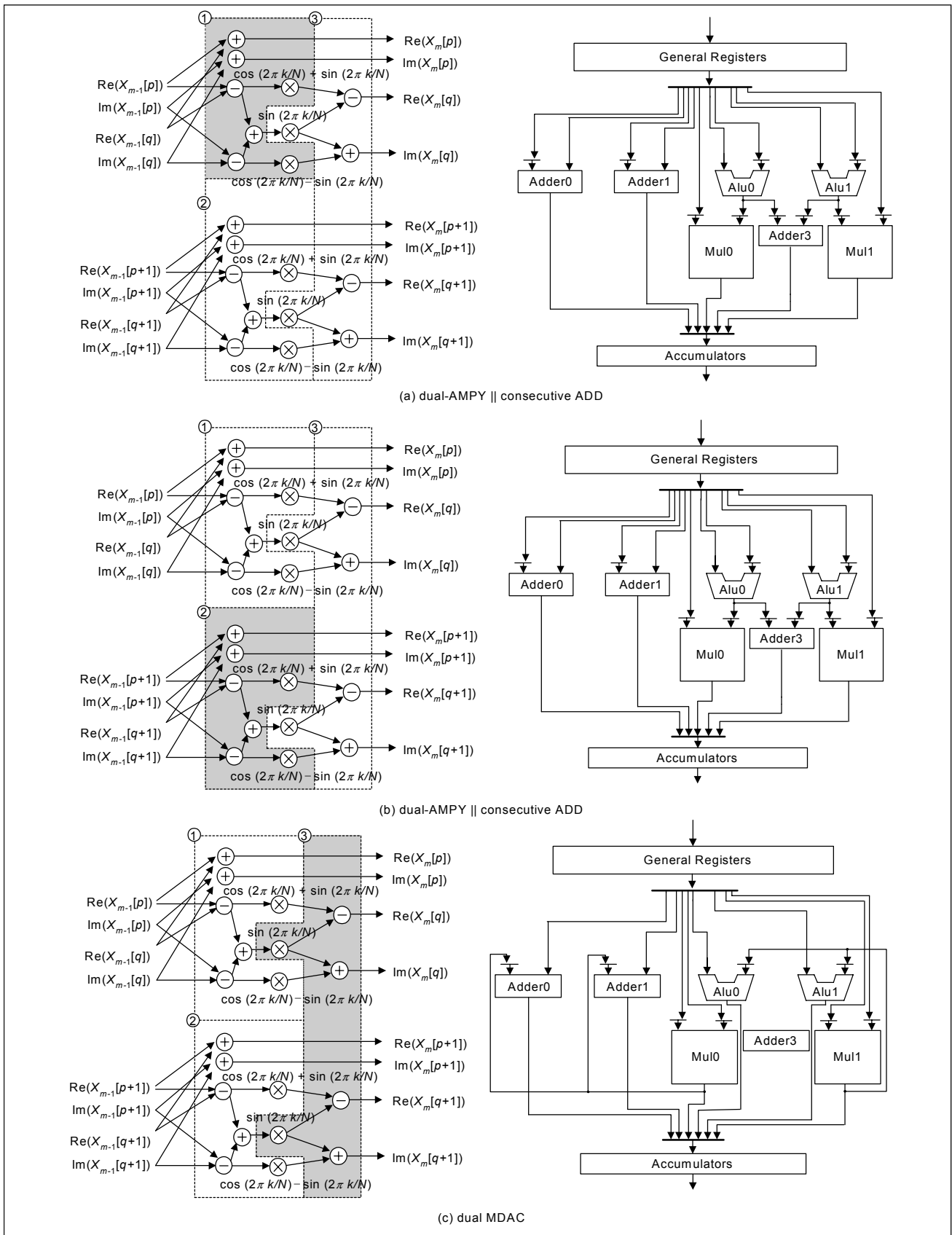


Fig. 6. The switched data-paths corresponding to the operations in Fig. 5.

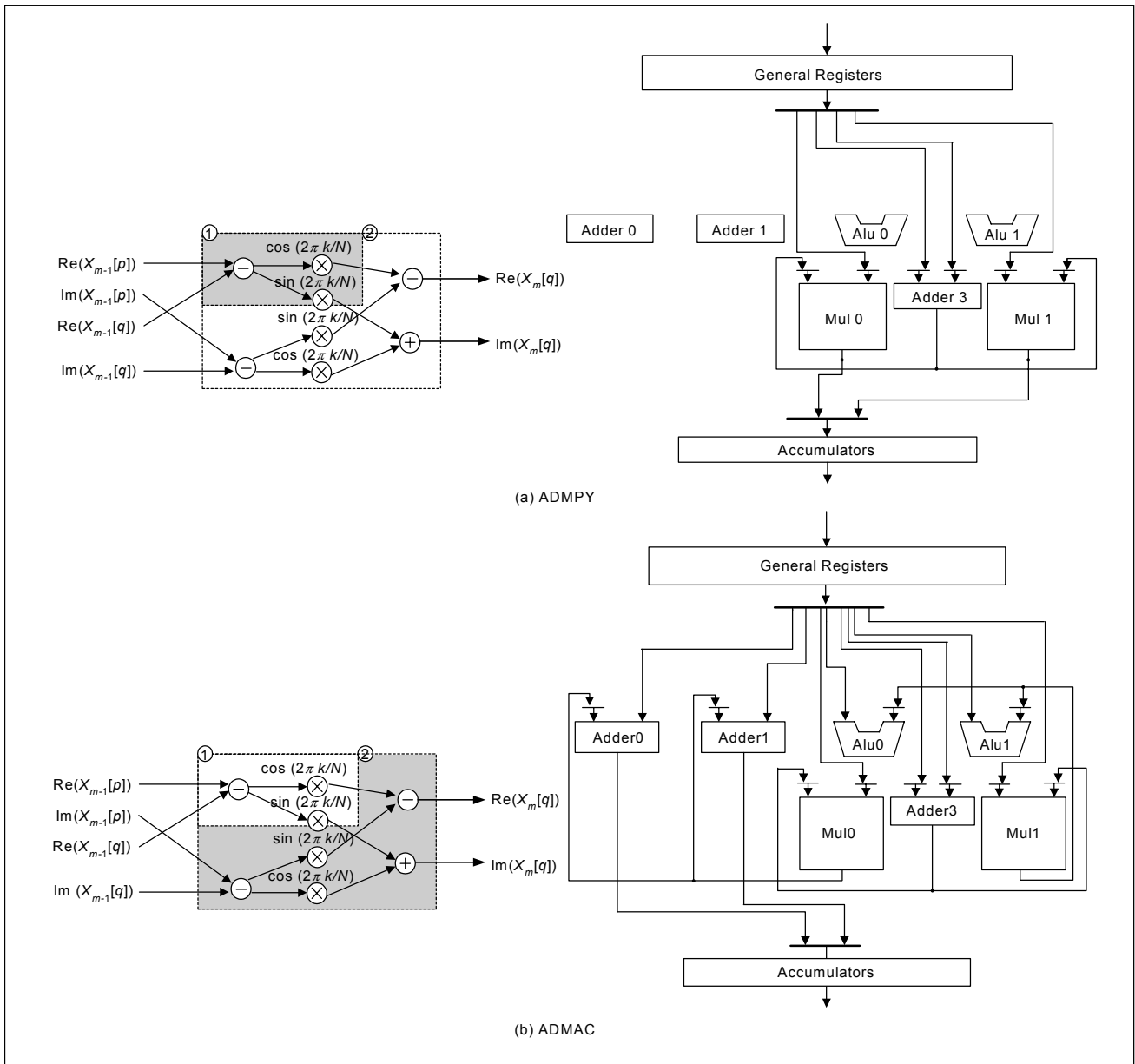


Fig. 7. The switched data-paths corresponding to the operations in Fig. 6.

using the ADMPY instruction which uses Adder3, Mul0, and Mul1 while cycle ② is performed using the ADMAC instruction that uses Adder0, Adder1, Alu0, Alu1, Mul0, and Mul1. Finally, the MDAC instruction is executed using Mul0, Mul1, Adder0, Adder1, Alu0, and Alu1.

The next figures illustrate the flows mentioned above. Figure 6 explains how to compute two radix-2 butterflies using dual-ADMPY, consecutive ADD, and dual MDAC successively and describes the corresponding switched data-paths in the DPU. Here, “switched” means that the data-paths are changed by 2-by-1 Multiplexers. In Fig. 6, (a) is performed at the first clock cycle, (b) is performed at the second clock cycle, and then (c) is

performed at the third clock cycle.

Figure 7 explains how to compute two radix-2 butterflies using ADMPY and ADMAC and describes the corresponding switched data-paths in the DPU. In Fig. 7, (a) is performed at the first clock cycle and then (b) is performed at the second cycle.

IV. IMPLEMENTATION

The timing simulation using the CADENCE™ Verilog-XL shows the maximum delay path is about 6.92 ns, and thus, the maximum operating clock frequency is about 144.5 MHz. If a

deeper pipeline is used, a higher operating clock frequency can be obtained. Hence, the required FFT computation time can be reduced.

Table 1 presents performance comparisons among the DSP architectures for FFT computation [5]-[8]. Note that the performance figures of commercial DSP chips are given by their data sheets or references.

Table 1. Performance comparisons of the FFT computation in various DSP chips (The units are clock cycles and a blank means that the estimated data was not available).

DSP chips \ FFT points (N)	512	1024
TMS320C62x	9,416	20,780
CARMEL DSP Core	5,342	11,628
STARCORE™ (SC140)		10,239
The proposed DPU	3,456	7,680

For $N=512$, $(512/2) \times \log_2 512$ butterflies must be computed. Since the proposed architecture takes 1.5 cycles for one butterfly, $(512/2) \times \log_2 512 \times 1.5 = 3456$ clock cycles are needed for completing a 512-point FFT. Similarly, for $N=1024$, $(1024/2) \times \log_2 1024 \times 1.5 = 7680$ clock cycles are taken. Table 1 shows that the proposed architecture performs better than the other architectures. The required time for 256-point FFT processing is $1536 \times 6.92 \text{ ns} = 10.63 \mu\text{s}$. The required time for 1024-point FFT is $7680 \times 6.92 \text{ ns} = 53.15 \mu\text{s}$. Hence, the proposed architecture can satisfy the requirement of high-speed communication standards that use OFDM or DMT modulation.

STARCORE™ (SC140) has four MAC units. Hence, it can complete the complex multiplication in one cycle and may have a performance similar to the proposed DPU. However, it requires larger hardware than the proposed DPU. Using only half the number of operation units, the proposed DPU can show a better performance than the SC140.

We used a one stage pipelined Carry Look-ahead Adder as an adder and a three stage pipelined Wallace-tree multiplier as a multiplier. The proposed architecture was modeled by a Verilog HDL. We performed a logic synthesis using the HYUNDAI™ 0.35 μm standard cell library with a SYNOPSIS™ Design Compiler and did a timing simulation was performed.

V. CONCLUSIONS

This paper proposed DSP instructions and their DPU architecture for high-speed FFTs in OFDM systems. First, we

proposed the novel instructions that are necessary to perform FFT computation and then a DPU architecture that can support the proposed instructions as well as general DSP instructions. The proposed architecture, having little hardware overhead, can perform FFTs about two times faster than the existing DSP chips in terms of execution cycles. In addition, it is clear that the power consumption of the proposed architecture is lower than existing architectures because it uses fewer function units.

REFERENCES

- [1] J. Glossner, J. Moreno, M. Moudgill, J. Derby, E. Hokenek, D. Meltzer, U. Shvadron, and M. Ware, "Trends in Compilable DSP Architecture," *Proc. IEEE Workshop Signal Processing Systems*, 2000, pp. 181-199.
- [2] VDSL Alliance, *VDSL Alliance Draft Standard Proposal*, Apr. 1999.
- [3] B.R. Wiese and J.S. Chow, "Programmable Implementations of xDSL Transceiver System," *IEEE Comm. Mag.*, vol. 39, May 2000, pp. 114-119.
- [4] J.G. Cousin, M. Denoual, D. Saille, and O. Sentieys, "Fast ASIP Synthesis and Power Estimation for DSP Application," *Proc. IEEE Workshop Signal Processing Systems*, 2000, pp. 591-600.
- [5] *CARMEL DSP Core Data Sheet*, Infineon Technologies Inc., 1999.
- [6] Philips Semiconductors Inc. "Philips Semiconductors' R.E.A.L. DSP Core for Low-Cost Low-Power Telecommunication and Consumer Applications," *Technical Backgrounder From Philips Semiconductors*, Sept. 1998, [Online] Available: <http://www.us-3.semiconductors.com>.
- [7] *TMS320C62xx User's Manual*, Texas Instruments Inc., Dallas, TX, 1997.
- [8] *SC140 DSP Core Reference Manual*, Motorola Semiconductors Inc., Denver, CO, 2000.
- [9] *DSP16210 Digital Signal Processor Data Sheet*, Lucent Technologies Inc., Allentown, PA, 2000.
- [10] O.B. Sheva, W. Gideon, and B. Eran, "Multiple and Parallel Execution Units in Digital Signal Processors," *Smart Cores Articles*, 1999, [Online] Available: <http://www.dspg.com>.
- [11] Soohwan Ong, Myung H. Sunwoo, and Manpyo Hong, "A Fixed-Point Multimedia DSP Chip for Portable Multimedia Services," *Proc. IEEE Workshop on Signal Processing Systems Design and Implementation*, Oct. 1998, pp. 94-102.
- [12] Soohwan Ong and M.H. Sunwoo, "A Fixed-Point DSP(MDSP) Chip for Portable Multimedia," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E82-A, June 1999, pp. 939-944.
- [13] A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ, Prentice-Hall, 1989.
- [14] P. Pirsch, *Architectures for Digital Signal Processing*, New York, Wiley, 1998.
- [15] A. Wenzler and E. Luder, "New Structures for Complex Multipliers and Their Noise Analysis," *Proc. IEEE Int'l Symp. Circuits and Syst.*, Apr. 1995, pp. 1432-1435.

- [16] Sangman Moh and Sukhan Yoon, "High-Speed Array Multipliers Based on On-the-Fly Conversion," *ETRI J.*, vol. 19, no. 4, Dec. 1997, pp. 317-325.
- [17] Kyoung Park, Sunghoon Choi, Yongwha Chung, Woojong Hahn, and Sukhan Yoon, "On-Chip Multiprocessor with Simultaneous Multithreading," *ETRI J.*, vol. 22, no. 4, Dec. 2000, pp. 13-24.



Jaesung Lee received the BS and MS degrees in electronic engineering from Ajou University, Suwon, Korea in 1999 and 2001. He is currently working in Electronics and Telecommunications Research Institute (ETRI) in Daejeon, Korea. His research interests include VLSI architectures, design of parallel processors, DSP chips, and

SoC design.



Jeonghoo Lee received the BS degree in electronic engineering from Ajou University, Suwon, Korea in 2002. He is currently working toward the MS degree in Ajou University. His research interests include include SOC design and application-specific DSP chip design.



Myung H. Sunwoo received the BS degree in electronic engineering from Sogang University in 1980, the MS degree in electrical and electronics from Korea Advanced Institute of Science and Technology in 1982, and the PhD in electrical and computer engineering from The University of Texas at Austin in 1990. He

worked for Electronics and Telecommunications Research Institute (ETRI) in Daejeon, Korea from 1982 to 1985 and Digital Signal Processor Operations Division, Motorola USA, from 1990 to 1992. Since 1992, he has been a Professor with Electrical and Computer Engineering Department, Ajou University in Suwon, Korea. His research interests include VLSI architectures, SOC design for multimedia communications, and application-specific DSP chip design. He has served on the Technical Committee of the IEEE Circuit and Systems VSATC since 1996, and on the Program Committee of the IEEE Workshop on SIGNAL PROCESSING SYSTEMS and the IEEE International ASIC/SOC Conference. He is a senior member of IEEE.



Sangman Moh received the PhD degree in information and computer engineering from Information and Communications University (ICU), Daejeon, Korea, in 2002. He has been with the Computer System Department of Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, where he is currently a Project Leader, since he received the MS degree in computer science from Yonsei University, Seoul, Korea, in 1991. He received the national qualification for Professional Engineer (PE) in information technology from the Korean Government in 1993. His research interests include computer architecture, parallel processing systems, mobile computing, computer arithmetic, and SoC design. He has published more than 30 papers as the first author in international and domestic journals and proceedings and has held more than 40 overseas and domestic patents. Dr. Moh is a member of the IEEE Computer Society, the ACM, the KISS, and the IEEK.



Seongkeun Oh received the BS degree in electronics engineering from Kyungpook National University, Daegu, Korea, in 1983, and the MS and PhD degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1985 and 1990. From 1988 to 1993, he was with Transmission Systems Lab, Samsung Electronics Inc, Seoul, Korea, as a Senior Researcher. Since 1993, he has been with Ajou University, Suwon, Korea, where he is currently an Associate Professor at the School of Electronics Engineering, leading the Communication Systems Research Group. During 1996-1997, he was a Visiting Professor at Simon Fraser University, Burnaby, BC, Canada. His research interests include smart antennas, space-time coding, MIMO systems, CDMA systems, and digital transmission technologies.