A FIXED-POINT MULTIMEDIA DSP CHIP FOR PORTABLE MULTIMEDIA SERVICES*

Soohwan Ong, Myung H. Sunwoo and Manpyo Hong*
School of Electrical and Electronics Eng., Ajou Univ.
*School of Information and Computer Eng., Ajou Univ.
email : sunwoo@madang.ajou.ac.kr

Abstract - **Existing multimedia processors having millions of transistors are not suitable for portable multimedia services and existing fixed-point DSP chips having fixed data formats are not appropriate for multimedia applications. This paper proposes a multimedia fixed-point DSP (MDSP) chip for portable multimedia services and its chip implementation. MDSP employs parallel processing techniques, such as SIMD, vector processing, and DSP techniques. MDSP can handle 8-, 16-, 32- or 40-bit data and can perform two MAC operations in parallel. In addition, MDSP can complete two vector operations with two data movements in a cycle. With these features, MDSP can handle both 2-D video processing and 1-D signal processing. The prototype MDSP chip has 68,831 gates, has been fabricated, and is running at 30 MHz.**

1. INTRODUCTION

Recently, demands on multimedia services increase and many programmable processors for multimedia processing have been proposed. Programmable processors for multimedia can be categorized into two classes: multimedia DSP chips; and multimedia extensions for general-purpose microprocessors. Multimedia DSP chips include the MicroUnity Mediaprocessor [1], Chromatic Mpact and Mpact2 [2,3], Philips TriMedia [4], TI (Texas Instrument) TMS320C62xx [5], and Analog Device ADSP-2106x [6]. Multimedia extensions for microprocessors include the MAX-1 (Multimedia Acceleration eXtensions) and MAX-2 [7] for HP PA-RISC, VIS (Visual Instruction Set) [8] for Sun UltraSparc, MMX (Multi-Media Extensions) [9] for Intel ix86, MDMX (MIPS Digital Media Extensions) [10] for SGI MIPS, and MVI (Motion Video Instructions) [11] for DEC Alpha.

The multimedia DSP chips commonly employ a combination of SIMD (Single Instruction-stream Multiple Data-stream), superscalar, VLIW (Very Long Instruction Word), multithread, vector processing, etc. to improve performance. These parallel processing techniques are suitable for computations of multimedia data having various data widths. However, the multimedia DSP chips usually contain millions of transistors that causes high cost and high power consumption. Thus, the multimedia DSP chips are not suitable for portable applications.

On the other hand, the fixed-point DSP chips [12-15] are widely used for a speech Codec (Coder/Decoder) in digital mobile communications, such as GSM (Group Special Mobile), IS-95 and IS-54, due to the small die size, low cost, and low power consumption. The upcoming IMT-2000 (International Mobile Telecommunication) standard requires moving picture services as well as speech

0-7803-4997-0/98/$10.00

and data services, and thus, mobile phones should support multimedia services. Since existing fixed-point DSP chips [12-15] supporting fixed data formats are mainly developed for one-dimensional signal processing, they are not suitable for multimedia processing. Thus, multimedia fixed-point DSP chips for portable multimedia services have to be developed.

The proposed MDSP architecture for portable multimedia services employs SIMD, vector processing and DSP architecture features. The MDSP instruction set is developed through the analysis of various multimedia algorithms including DCT (Discrete Cosine Transform) and motion estimation and instruction sets of existing multimedia processors [1-11] and fixed-point DSP chips [12-15]. The instruction set is classified into three types - SIMD instructions, DSP instructions, and vector instructions. SIMD instructions can exploit data parallelism by using partitioned arithmetic, DSP instructions can process DSP algorithms, such as FIR, IIR and adaptive filtering, and vector instructions can increase performance for DCT, FFT, convolution, etc.

We have implemented Verilog HDL (Hardware Description Language) models and performed logic synthesis using the SYNOPSYS™ tool. The prototype MDSP has actually been implemented using the 0.6μm Samsung SOG (Sea-of-Gate) cell library (KG75000), the total number of gates is 68,831 and the clock frequency is 30 MHz.

The paper is organized as follows. Section II describes the new MDSP architecture and section III describes the proposed instruction set. Section IV presents the prototype MDSP chip implementation and performance comparisons. Finally, section V contains concluding remarks.

2. THE MDSP ARCHITECTURE

For low cost and low power consumption, MDSP does not use millions of transistors used in existing multimedia DSP chips and employs a combination of SIMD, vector processing, and DSP architecture features to execute several multimedia function units in parallel. The MDSP architecture shown in Fig. 1 consists of DPU (Data Processing Unit), AGU (Address Generation Unit), PCU (Program Control Unit), X and Y data memories, program memory, three address buses, and four data buses. The instruction set supports parallel operations of three execution units (DPU, PCU, and AGU). The shaded blocks, such as DPU, AGU and PCU are integrated in the actual chip and all memories are excluded due to the gate count limitation.

Four bi-directional data buses consist of a 24-bit program data bus (PDB), a 32-bit X data bus (XDB), a 32-bit Y data bus (YDB), and a 32-bit internal data bus (IDB). These schemes support register-to-register, register-to-memory, memory-to-register, and memory-to-memory data moves and three data moves, such as one instruction fetch and two parallel data moves can be completed in a cycle. Data move among internal data buses is possible via an internal data bus switch without any latency.

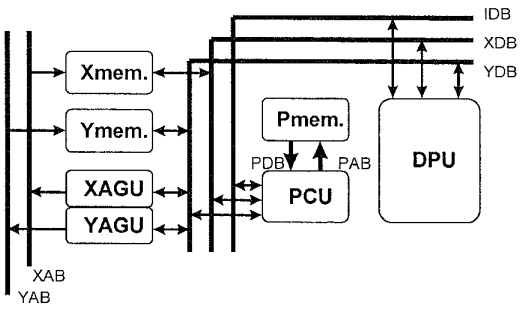

Fig. 1. The MDSP Architecture

The DPU architecture shown in Fig. 2 employs a two-stage pipeline structure. The first stage is the input register file, the switching network, two vector multipliers (VMULs), the vector ALU (VALU), and the Barrel shifter. The second stage is two vector adders (Vadders), the packing network, and the output register file. The input register file consisting of four 32-bit registers generally stores input operands. The output register file consisting of four 40-bit registers stores results, which can also be used as accumulators.

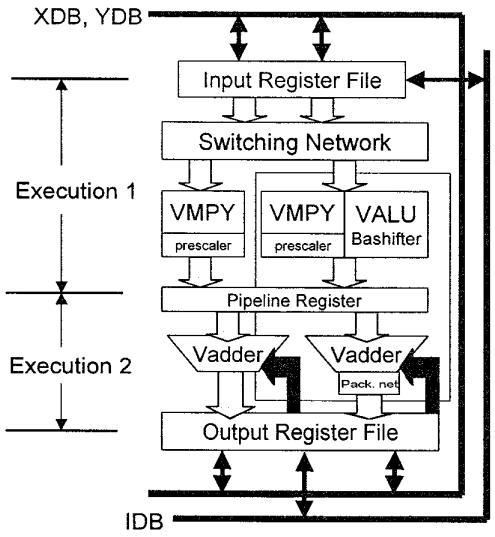

Fig. 2. The DPU Architecture

The switching network controlled by OMR (Operating Mode Register) or instructions rearranges input operands from the input register file to VMULs.

Vector instructions can easily perform various vector operations by setting OMR for the switching network. The switching network handles a global copy byte (PGCOPYB) operation, a global copy word (GCOPYW) operation and double MAC operations using DMR (Double MAC Register). The group copy byte and group copy word operations convert vector data to scalar data without the cycle overhead and any redundant usage of memory space. Instead, MMX [9] and MAX-2 [7] must copy a 16-bit coefficient by four times to store the coefficient into a 64-bit register.

Double MAC operations can be performed by using DMR. For example, as shown in Fig. 3 two convolution results are obtained by two VMULs using a coefficient twice through DMR in the switching network. Therefore, we can obtain two convolution results at the same time with two source operands.

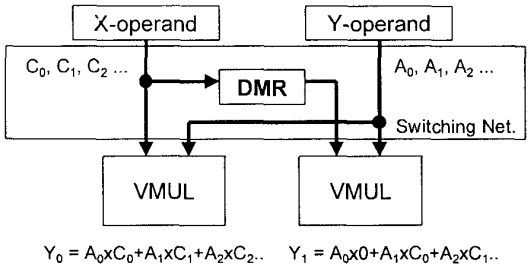

Fig. 3. Convolution using Double MAC Operations

Two VMULs and two Vadders form a vector structure to execute MAC operations. Similarly, VALU and two Vadders form a vector structure to execute multiple ALU functions in a cycle. Each VMUL performs one 16 x 16-bit multiplication or two 8 x 8-bit multiplications and Vadders sum the results of VMULs. Two pairs of VMUL and Vadder can perform two 16 x 16 multiplications and two 40-bit accumulations, or four 8 x 8 multiplications and four 16-bit accumulations simultaneously. Thus, MDSP can accelerate convolutions, correlation, and matrix multiplications which require a large number of MAC operations.

VALU performs 8-, 16-, 32-, or 40-bit arithmetic or logical operations and the Barrel shifter can shift operands by any number of bits. Since MDSP uses VALU and two Vadders to form the vector structure, MDSP can improve performance for algorithms using multiple ALU functions. For example, VALU subtracts four packed bytes and detects absolute values to get absolute differences. Two Vadders perform add operations for absolute differences, and then, 8-bit SAD (Sum of Absolute Difference) values are obtained.

The packing network increases the efficiency of the register file and memory usage. The previous 16-bit result stored in the output register file is shifted left by 16-bit and the current result is stored in the same register only updating the 16-bit LSP (Least Significant Portion). Since packing operations can be executed with data rearrangement and arithmetic operations in parallel, the number of instruction

cycles for data packing can be reduced.

With these multiple function units, MDSP can be effective for algorithms requiring matrix additions/ subtractions, matrix multiplications, MAC operations, convolutions, SAD, etc.

3. THE MDSP INSTRUCTION SET

This section describes the instruction set shown in Table 1, which uses parallel processing techniques. We have designed the instruction set through the analysis of multimedia algorithms and the analysis of instruction sets of existing DSP chips. We define three data types - packed byte, packed word and packed double word. Four bytes i.e. the 8-bit color pixel values - red, green, blue and alpha values can be packed into one 32-bit word quantity. Since MDSP using SIMD features performs the same operation for packed data simultaneously, it can easily exploit data parallelism for multimedia processing.

For the packed byte data, the saturation or wrap-around arithmetic is performed to reduce the exception processing overhead. MDSP supports both the signed saturation and unsigned saturation. In the saturation arithmetic, the result of an addition for packed bytes becomes the maximum value if there is an overflow. If there is an underflow for a subtraction, the result becomes the minimum value. In the wrap-around arithmetic, a carry and a borrow are ignored.

MDSP executes every instruction with data rearrangement operations through the switching network to perform vector operations. The instruction set consists of basic arithmetic (ADD, SUB, MUL), logical (AND, OR, XOR, NEG), compare (MAX, MIN), shift (ASL, ASR, LSL, LSR), and data conversion instructions to pack and unpack data elements (PACK, UNPACK). In addition, the instruction set includes bit manipulation and vector instructions. 8- and 16-bit data are used for pixel operations and intermediate calculations and 32- and 40-bit data are used for high precision and double precision operations.

Min/Max instructions compare packed values and fill the minimum/maximum values. Convert instructions rearrange data for pre- and post-operations required in many multimedia algorithms. Pack instructions contract data to smaller packed data having fewer number of bits. Unpack instructions expand data to larger packed data having more number of bits. In addition, two data rearrangement instructions - PGCOPY (Packed Group Copy) and PSHF (Packed Shuffle) are added. PGCOPY takes one byte from two 32-bit source operands and copies it to the destination register. PSHF performs a shuffle operation with the packed data.

Multiplication instructions perform four 8 x 8 or two 16 x 16 multiplications in parallel. To multiply four bytes packed in a source operand 1 with a single byte within a source operand 2, the group copy byte operation can be done with MUL instructions. In the same manner, the group copy word operation can be done to multiply two words with a single word. This capability is useful for vector-scalar operations without instruction overhead to rearrange data. MAC instructions perform two 16 x 16 multiplications and two 40-bit accumulations, or four 8 x 8 multiplications and four 16-bit accumulations in a cycle.

Operations	Instructions	Data path
Add / Subtract	PADD (wraparound/signed/unsigned)	8, 16, 32, 40
	PSUB (wraparound/signed/unsigned)	8, 16, 32, 40
	PADC, PSUC (add/sub with carry)	32, 40
	PINC, PDEC (increment, decrement)	32, 40
Logical	PAND, POR, PXOR, PNEG	40
Min/Max	PMAX, PMIN, SMAX*, SMIN* (parallel/serial min/max)	8
Convert	Pack, Unpack	8, 16
	PGCOPY (group copy)	8
	PSHF (shuffle)	8
Shift	ASL, ASR, LSL, LSR	8, 16, 40
	ROL, ROR	40
MUL / MAC	PMUL (multiply)	8, 16 (four 8x8, two 16x16)
	PMAC* (multiply and accumulate)	8, 16 (four 8x8+16=16, two 16x16+40=40)
	PDMAC* (double MAC for convolution)	8, 16 (four 8x8+16=16, two 16x16+40=40)
	PMADB*, PMADB16* (multiply and add, packing)	8
SAD	PSAD*, PSAD16* (sum of absolute, packing)	8
Control	FOR, FOREVER, REP, EXITcc (H/W loop and repeat instructions)	
	JMP, JMPcc, CALL, CALLcc RTI, RTS	
	WAIT, IDLE (low power instructions)	
	TRAP, NOP (S/W interrupt)	

* vector instructions

Table 1. The MDSP Instruction Set

SAD instruction is used for motion estimation. SAD instruction combines 12 operations which include four subtractions, four absolute value detections, and four summations of each absolute value for four pairs of the 8-bit packed data. In addition, most operations can overlap with two parallel moves between the register file and the data memory. Therefore, it can improve performance and increase the I/O bandwidth.

MDSP supports vector operations that can successively perform many operations in a cycle and can be effective for DCT, FFT, convolution, etc. For example, Fig. 4 shows the PMADB16 (Packed Multiply-and-Add for Byte with packing) instruction. PMADB16 with the packed group copy and two parallel moves can perform two data moves, four data copies, four 8 x 8 multiplications, three 16-bit additions, and two data packing operations in a cycle.

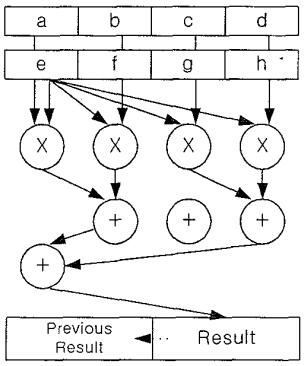

Fig. 4. PMADB16 Instruction

The second version of MDSP has more wireless communication instructions, such as PACS (Add-and-Compare-and-Select) and PSLBIT (Shift-Left-using-Selected-Bit) instructions for the Viterbi algorithm, and PCMULW (Complex-Multiplication) instruction for FFT and Modem algorithms.

4. CHIP IMPLEMENTATION AND PERFORMANCE COMPARISONS

We have implemented behavior and structure models of the proposed architecture using the Verilog HDL and simulated the models using the Cadence™ Verilog-XL. First, the behavior models realize every unit component, and then these components are mapped using the structure models. We simulated the combination of instructions and algorithms.

For gate-level logic synthesis, we have optimized the Verilog HDL models using the SYNOPSYS™ Design Analyzer. We have used the 0.6 μm SOG cell library (KG75000) and performed function and timing simulations using the Cadence™ Verilog-XL. We verified the results between the Verilog HDL models and the synthesized models. The total gate count is only 68,831 and the clock frequency is 30 MHz which is limited by the SOG technology. Fig. 5 show the photograph of the prototype MDSP chip.

Fig. 5. The Photograph of MDSP

For performance comparisons, we have programmed various algorithms including FFT, IIR filter, FIR filter, adaptive filter, DCT, BMA (Block Matching Algorithm), etc. We have benchmarked the DCT algorithm [16] and the FBMA (Full searching BMA) algorithm [17]. Table 2 shows performance comparisons for the number of instruction cycles with DSP56100 [12] and TMS320C6201 [5]. DSP56100 is a traditional 16-bit fixed-point DSP chip, and thus, it is not suitable for multimedia processing. TMS320C6201 having millions of transistors is a multimedia DSP chip, has eight functional units and can issue maximum 8 instructions per cycle. On the other hand, MDSP having quarter million of transistors has only two VMAC units and issues only one instruction per cycle. Considering the numbers of transistors, MDSP is quite comparable with the other DSP chips and is suitable for portable multimedia applications.

Algorithms	DSP56100	TMS320C6201	MDSP
FFT	10N	7N/2	7N
IIR	6N	4N	5N
FIR	N	N/2	N/2
Adaptive Filter	3N	.	2N
DCT (8 x 8)	.	226 cycles	360 cycles
BMA (352 x 240)	.	.	3 frames/sec

* N : Number of filter taps

Table 2. Performance Comparisons

5. CONCLUSIONS

This paper presented the design and implementation of the fixed-point DSP chip for portable multimedia services. MDSP targets for portable multimedia applications that can handle various multimedia data types and multiple vector operations for real-time processing. The MDSP architecture employs SIMD, vector processing, and DSP features. With these features, MDSP can improve performance for multimedia algorithms. We implemented the prototype chip using the 0.6 ㎛ SOG cell library (KG75000). MDSP has 68,831 gates and is running at 30 MHz and the clock frequency is limited due to the SOG technology. Currently we are designing the second version of MDSP having enhanced architecture

features and more refined instruction set to increase performance and to reduce the power consumption and cost.

ACKNOWLEGEMENTS

This work was performed as a part of ASIC Development Project supported by Ministry of Commerce, Industry & Energy, Ministry of Information and Communication, and Ministry of Science and Technology and was financially supported by IC Design Education Center.

References

[1] C. Hansen, "Architecture of a Broadband Mediaprocessor," *COMPCON 1996, IEEE Computer Society Press*, 1996.

[2] Microprocessor Report, "Chromatic's Mpact2 Boosts 3D," pp. 5-10., Nov. 1996.

[3] Robert E. Owen, S. Purcell, "An Enhanced DSP Architecture for the Seven Multimedia Functions: The Mpact2 Media Processor," *IEEE Workshop on SiGNAL Processing Systems*, pp. 76-85, Nov. 1997.

[4] Philips Semiconductors, "VLIW Computer Architecture," July 1996.

[5] Texas Instruments Inc., *TMS320C62xx User's Manual*, 1997.

[6] Analog Devices Inc., *ADSP 2106x SHARC User's Manual*, 1996.

[7] R. Lee, "Subword Parallelism with MAX2," *IEEE Micro*, vol. 16 no. 4, pp. 51-59, Aug. 1996.

[8] M. Trembley, M. O'Connor, V. Narayannan, L. He, " VIS Speeds New Media Processing," *IEEE Micro*, vol. 16 no. 4, pp. 10-20, Aug. 1996.

[9] A. Peleg and U. Weiser, "MMX Technology Extension to the Intel Architecture," *IEEE Micro*, vol. 16 no. 4, pp. 42-50, Aug. 1996.

[10] Silicon Graphics Inc., "MIPS Extension for Digital Media with 3D," Dec. 1996.

[11] Microprocessor Forum, Oct. 1996.

[12] Motorola Inc., *DSP56100 Digital Signal Processor User's Manual*, 1994.

[13] AT&T Microelectronics Inc., *DSP1610 Digital Signal Processor*, 1992.

[14] Texas Instruments Inc., *TMS320C5x User's Guide*, 1990.

[15] Analog Devices Inc., *ADSP2100 Family User's Manual*, 1993.

[16] W.H. Chen, S. C. Harrison, and S. C. Fralic, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. on Comm.*, vol. COM-25, no. 9, Sep. 1977.

[17] J. S. Baek, S. H. Nam, and M. K. Lee, "A Fast Array Architecture for Block Matching Algorithm," in *Proc. IEEE Int. Symp. Circuits Syst.*, London, U.K., May 1994, pp. 211-214.