Design of a Novel SIMD Architecture by Fusing Operations and Registers

Jih-Ching Chiu
Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, 804, Taiwan
chiujhc@ee.nsysu.edu.tw

Kai-Ming Yang
Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, 804, Taiwan
d953010024@gmail.com

Yu-Liang Chou
Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, 804, Taiwan
d943010010@student.nsysu.edu.tw

ABSTRACT
This paper presents an architecture called “multi-streaming SIMD” that enables one SIMD instruction to simultaneously manipulate multiple data streams. To efficiently and flexibly realize the proposed architecture, an operation cell is designed by fusing the logic gates and the storage cells together. The operation cells then are used to compose a register file with the ability of performing SIMD operations called “Multimedia Operation Storage Unit (MOSU)”. Further, many MOSUs are used to compose a multi-streaming SIMD computing engine that can simultaneously manipulate multiple data streams and exploit the subword parallelisms of the elements in each data stream. This paper also designs three instruction modes (global, coupling, and isolated modes) for programmers to dynamically configure the multi-streaming SIMD computing engine at the instruction level to manipulate different amounts of data streams. Simulation results show that when the multi-streaming SIMD architecture has four-4-register MOSUs, it provides a factor of 3.3x to 5.5x performance improvement compared with Intel’s MMX extensions on eleven multimedia kernels.

Categories and Subject Descriptors
C.1.2 [Processor Architectures]: Multiple Data Stream Architectures

General Terms
Performance, Design, Measurement, Experimentation

Keywords: SIMD, PIM, Processor-in-memory, Multimedia extensions, MMX

1. INTRODUCTION
Current computer applications are multimedia-rich. How to meet the computational requirements of these applications is a challenge of modern processor design. The most popular solution in the past decade has been the enhancement of the general-purpose processor with MMX-like extensions [1]. The key idea in these multimedia extensions is the exploitation of subword parallelism in a single instruction multiple data (SIMD) fashion. It has been shown that MMX-like extensions can improve the performance of many multimedia kernels and applications. However, these MMX-like extensions are single data stream SIMD operations and can only exploit subword parallelism. Therefore, these extensions lack the ability to exploit the full data-level parallelism (DLP) of the multimedia applications. In this paper, a multi-streaming SIMD architecture is proposed to further extend current MMX-like extensions to manipulate multiple data sets in parallel. Three features of proposed architecture are described as below:

A) The Operation Cell Concept
We proposed a Processor-In-Memory-like [2] register-file architecture including SIMD logic for general-purposed processors to further extend current MMX-like extensions to obtain high performance. Under the controls of one MMX-like instruction, many SIMD units perform the same operations in parallel on different data sets and each SIMD unit also exploit the subword parallelisms of the elements in each data set. Therefore, the major restriction (lack of the ability to exploit more DLP) of current MMX-like extensions can be resolved.

Figure 1 shows a simplified block diagram of the 3-stage pipeline processor with the multi-streaming SIMD architecture. The multi-streaming SIMD architecture can act two different roles, first simple as the register file while general-purposed instructions are being executed and second as the SIMD units support multiple data streams while SIMD instructions are being performed. To realize the proposed architecture, the operation cell, which fuses logic gates and D flip-flops together, is proposed to hierarchically implement the proposed architecture. The implementation steps are described as below:

1. To fuse computing logics and D flip-flops together as an operation cell. Hence, the operation cell has the dual capabilities of operation and storage for bit slicing (see the upper right block of Figure 1).

Figure 1: A simplified block diagram of the three-stage pipeline architecture with the multi-streaming SIMD architecture.
Figure 2: Relationship between three-mode instruction and resource allocation.

2. To implement a special SIMD unit named MOSU (Multimedia Operation Storage Unit), which has the dual capabilities of subword SIMD operations and storage, by means of combining many operation cells (see the right block of Figure 1).

3. Four MOSUs are bounded to construct a basic multi-streaming SIMD computing cell named BMMOC (Basic Multi-streaming Multimedia Operation Cell) to handle different data streams (one, two, or four data streams) according to the three-mode instructions. (see the right lower block of Figure 1)

4. Every Four BMMOCs are grouped with a switch box (SB) and four grouped BMMOCs are also grouped with a SB. The SB is responsible for handling the data communications among the BMMOCs. Through the connections of SBs, many BMMOCs are bounded to construct a multi-streaming SIMD computing engine. (see the center block of Figure 1)

By this hierarchical implementation method, the hardware designers can provide different levels of SIMD computational ability for the general-purposed processors according to their needs.

B) Three Instruction Modes for Dynamically Configuring the SIMD resources

Three instruction modes (global, coupling, and isolated modes) are defined for the MMX-like extensions to modulate the amount of parallel data streams and to efficiently utilize the computation resources. Current MMX-like instruction set can be upgraded by prefixing “G”, “C” and “I” to the MMX-like instructions to become the three-mode instruction set. Figure 2 shows an example of all the programmer-visible SIMD operation resources and data streams supported by the three instruction modes. Given \( N \) SIMD units and \( M \) registers, where \( M \) is divisible by \( N \), the isolated mode can operate on \( N \) different data streams. However, programmers can only use \( M/N \) logical registers to program. Comparing to the isolated mode, the coupling mode operates on fewer data streams (only \( N/2 \) data streams), whereas programmers can use more logical registers (increased to \( 2M/N \) logical registers) to program. This mechanism provides programmers with greater flexibility. It is possible to run a smaller number of data streams, each of which requires more registers, or a larger number of data streams, each of which requires fewer registers. In contrast to the isolated and coupling modes, the global mode only operates on one data stream. In this mode, however, programmers can use all the \( M \) registers. In the proposed architecture, the main purposes of the global mode instruction are (1) to rapidly transfer and manipulate data in different data streams and (2) to maintain backward compatibility with original MMX-like instruction set.

C) The Two-dimensional Data Manipulation Concept

By fusing the operations and the register file together, the two-dimensional data manipulation concept can be realized efficiently for multimedia computations. The two-dimensional data manipulation concept makes multiple SIMD processing elements stored in the proposed architecture can be re-permuted, transposed and accumulated under the controls of one instruction. We have proposed a new instruction, “MTRANS”, for our architecture basing on the two-dimensional data manipulation concept. The “MTRANS” instruction can transpose a 4x4 matrix whose four rows are stored in 4 MOSUs respectively. The “MTRANS” instruction makes rows into columns and enables MMX-like extensions to execute the same operation on all elements of the column in parallel. This instruction can benefit the applications that are not fully data parallel like the 2D idct, rgb-to-ycc conversion, etc. In the future, we will propose more excellent instructions for multimedia application basing on the two-dimensional data manipulation concept.

2. SIMULATION RESULTS

Figure 3 shows the simulation results of proposed architecture. The Intel’s MMX technology is chosen as the baseline single-streaming SIMD extensions to be further extended into the multi-streaming SIMD extensions. The eleven representative kernels of current multimedia applications are chosen to evaluate the performances. The area spent on implementing the multi-streaming SIMD architecture with a BMMOC (16 registers) that has four 4-register MOSUs and supports “MTRANS” instruction is 2.8 times the area spent on implementing the MMX-enhanced processor, and the speedups of proposed architecture can over MMX technology ranging from 3.3x to 5.5x.

3. REFERENCES
