A Study of Highly Functional Memory Controller with MAC Engine

Toshiharu Imai (210202)

School of Infomation Science,
Japan Advanced Institute of Science and Technology

2003.8/13

Keywords: Memory Controller, MAC Engine, Multiply and Accumulate, Filter.

1 Introduction

Recently, multimedia applications targeting pictures and sounds are getting spread as performance of a computer is improved. Higher performance for signal processing is necessary since data size required by high bit or high sampling rates such as in DVD is increasing.

In this thesis, we propose a method of fast response and high throughput based on load balancing, by equipping a memory controller (MC) with ability to reduce memory access time and alleviate a gap between CPU and memory speed, and MAC which is a feature of DSP.

2 Data Transfer Method

Stride Data Transfer (SDT) [1] is a method of reading stride data fast and transfer them to a processor, by specifying a row address only once, automatically generating column addresses, and give them to DRAM. Software execution can set the number of data transferred and the stride width. Therefore, this is effective for non-locality data accesses.

A FIFO buffer provided by a reconfigurable cache [2] can receive data transferred by SDT, by utilizing a part of the cache memory. The reconfiguration can be done by software. This function is more effective than usual caches for non-locality data sequences.

3 Digital Filter

A digital signal processor (DSP) has a functional unit for processing data and a control unit for controlling itself, which are similar to those of a general-purpose processor. One of the features of this study is embedding a MAC unit in MC. The problem of using a DSP is that there are large overheads of data transfer or start- and end-notification of processing between a CPU and DSP. In addition, there is another problem that it forces

large cost and hardware size. The method proposed in this study provides only a MAC unit in MC, and achieves lightweight processing and reduction of hardware.

4 Highly Functional Memory Controller with MAC Engine

In our MC, MAC (Multiply and ACcumulate) is embedded and SDT cooperates with it, which achieves fast filtering processing.

MC includes registers for storing finite impulse responses, reads input data sequence through SDT from DRAM, and performs a series of multiply-accumulate calculations between them. The output sequence is transferred to a processor and inserted into the FIFO buffer that the reconfigurable cache provides. The processing is supported by Memory Management Unit (MMU) in the processor.

5 Basic Performance Evaluation

We evaluated the MC designed in VHDL by RTL simulation. When two data sequences whose size was 32 were multiplied and accumulated, the execution with MAC engine was 3,108 cycles faster than that by CPU alone. Moreover, we synthesized the RTL of MC and obtained the hardware size of the MAC engine, 14,547 gates approximately.

Evaluation of the amount of hardware.

6 Related Work

Stream buffer [3] is a buffer in a processor and receives data. This is separated from a data cache, and therefore, the memory resource cannot be utilized when a running program does not process stream data.

Impulse [4] and SMC [5] is a memory controller and have methods of transferring data sequence. Impulse requires aliases of data in a program, which burdens programmers. In addition, this introduces extra address translation, which leads to large overheads. In SMC, read data are stored in the MC buffers. Therefore, it is necessary for a processor to obtain the data by non-cacheable accesses, which brings large overheads.

7 Conclusion

In this thesis, we proposed a method of achieving fast filtering by using SDT for fast data transfer and FIFO buffer, and equipping MC with a MAC function which performs filtering calculation for stream data. Moreover, we implemented the method targeting a FPGA. In RTL simulation, we showed that the proposed scheme could achieve much faster execution than a conventional CPU alone.

In future work, we will use a processor with FIFO function and evaluate the effectiveness on the actual hardware.

References

[1] Fukawa, Tanaka, Miyazaki: "Highly Functional Memory Controller for Main Memory Database", IPSJ SIG Report ARC, Vol.2002, No.112, pp.77–82.

[2] Khairuddin bin Khalid, K.Tanaka: "Implementation of FIFO Buffer Using Cache Memory", IPSJ SIG Report ARC, Vol.2002, No.112, pp.83–88.

[3] Farkas, N.P.Jouppi and P.Chow, "How Useful Are Non-blocking Loads, Stream Buffers, and Speculative Execution in Multiple Issue Processors?", Proc. of the 1st HPCA, pp. 78–89, 1994.

[4] J.B.Carter, W.C.Hsieh, L.B.Stoller, M.R.Swanson, L.Zhang, E.L.Brunvand, A.Davis, C.-C.Kuo, R.Kuramkote, M.A.Parker, L.Schaelicke, and T.Tateyama: "Impulse: Building a Smarter Memory Controller", Proc. of 5th HPCA, pp. 70–79, 1999.

[5] S.A.McKee, A.Aluwihare, B.H.Clark, R.H.Klenke, T.C.Landon, C.W.Oliver, M.H.Salinas, A.E.Szymkowiak, K.L.Wright, W.A.Wulf, J.H.Aylor: "Design and Evaluation of Dynamic Access Ordering Hardware", Proc. of 10th ICS, pp. 125–132, 1996.