MorphoSys: A Coarse Grain Reconfigurable Architecture for Multimedia Applications

Hooman Parizi, Afshin Niktash, Nader Bagherzadeh, and Fadi Kurdahi

Department of Electrical and Computer Engineering
University of California, Irvine 92697
{hparizi, aniktash, nader, kurdahi}@ece.uci.edu
http://www.eng.uci.edu/morphosys

Abstract. MorphoSys is a reconfigurable architecture for computation intensive applications. It combines both coarse grain and fine grain reconfiguration techniques to optimize hardware, based on the application domain. M2, the current implementation, is developed as an IP core. It is synthesized based on the TSMC 0.13 micron technology. Experimental results show that for multimedia applications MorphoSys has a performance comparable to ASICs with the added benefit of being able to be reconfigured for different applications in one clock cycle.

1 Introduction

Reconfigurable systems are an intermediate approach between the Application Specific Integrated Circuits (ASICs) and general purpose processors. They have wider applicability than ASICs while their performance is comparable to them. On the other hand, multimedia applications comprise of several subtasks with different characteristics. This feature in addition to a large set of input and output data, lead to an uneconomical solution in ASIC, and low performance solution on general purpose architectures. Reconfigurable systems are considered as an alternative approach for developing architectures for multimedia and DSP applications, Raw [1], PipeRench [2], Garp [3] and MorphoSys[4] are ongoing research projects in this area.

In this paper M2, a new implementation of MorphoSys is introduced. Comparison to M1, the previous implementation, M2 consists of high performance functional units, and optimized memory architecture. M2 implementation goes through a fully automated methodology starting from an IP core.

In Sections 2 basic structure of MorphoSys are described. Section 3 describes M2 reconfigurable cell with emphasis on the new features. Section 4 discuses the parallel structure of MorphoSys. Section 5 evaluates some multimedia applications on the M2.

B. Monien and R. Feldmann (Eds.): Euro-Par 2002, LNCS 2400, pp. 844–848.

2 MorphoSys Architecture

Figure 1 shows the basic building blocks and their connections in MorphoSys. RC-Array is the reconfigurable part of the system. It consists of an 8 by 8 array of reconfigurable cells (RCs). The configuration data is stored in the context memory. During the execution, the context word is loaded from the context memory to the context registers of the reconfigurable cell. Frame buffer is an embedded data memory in the MorphoSys. It gets the data from the external memory and feed the RC-Array with the appropriate data. All the data movements between the MorphoSys memory elements and the external memory are handled by the DMA controller. TinyRisc [5] is a general purpose 32-bit RISC processor. It controls the sequence of operations in MorphoSys as well as executing non-data parallel operations.

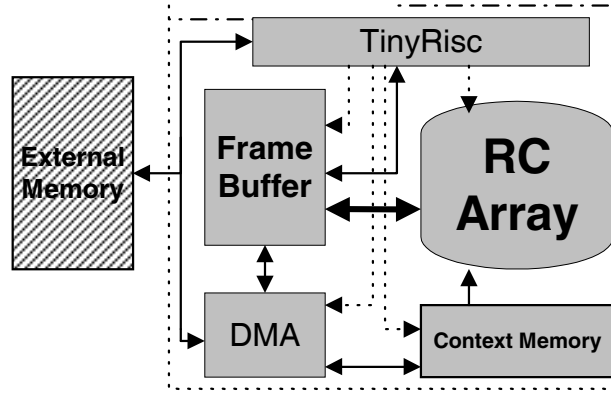

Figure 1- MorphoSys Architecture

3 Reconfigurable Cell Architecture

Reconfigurable Cells (RCs) are the main processing units in MorphoSys. Each RC consists of four types of basic elements. Functional units for arithmetic and logic operations, Memory element to feed the functional units and store their results, input and output modules to connect cells together to form the RC-Array architecture and a fine grain reconfigurable logic block. Figure 2 shows the block diagram of each RC.

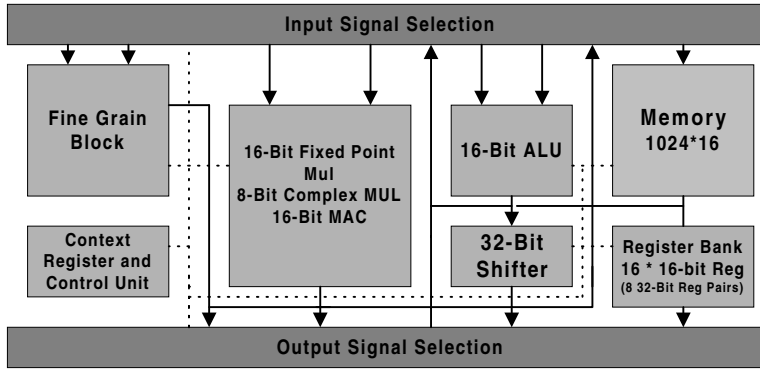

Figure 2 - Reconfigurable Cell Block Diagram

Table 1- Reconfigurable Cell features in M2 versus M1 implementation

Reconfigurable Cell	M2	M1
MAC	16-bit multiplication and addition 8-bit Complex multiplication and conjugate 32-bit internal register to increase precision	16x12-bit multiplication and addition NOT supported NOT supported
ALU	flags to support conditional operations variable length shifter	NOT supported Shifter with fixed number of shifts
FGB	supports custom functional units based on the application	NOT supported
Memory	1KB accessible as 1Kx8 or 512x16 Indexed address register, useful for Lookup Tables	NOT supported NOT supported
Register File	Fourteen 16-bit Registers Auto-increment and auto decrement registers	Four 16-bit Registers NOT supported
Context Reg	32-bit configuration register	32-bit configuration register

Table 1 summaries the RC features in M2 and our previous implementation M1. More details on RC functional units are in [6].

4 MorphoSys Parallel Architecture

MorphoSys parallel architecture is based on the connection of reconfigurable cells and the way that they are connected to the memory. In this section these structures will be discussed.

4.1 Reconfigurable Cell Array

RC-Array forms the parallel architecture of MorphoSys. In M2 all RCs in a single row or column of RC-Array are connected together, while in M1 the connection is only pyramid based. High connectivity in M2 simplifies data movement between RCs. Another new feature in RC-Array is register sharing between RCs in a row or column. By this feature all RC registers in a row or column can be accessed by other RCs in a single cycle. These features simplifies the programming of the system.

4.2 Frame Buffer

Frame buffer is a dual port memory architecture. It gets the data from the external memory and provides them for RC-Array and TinyRisc. Two ports can access the frame buffer simultaneously, so reading or writing can be done by the RC-Array or TinyRisc and at the same time DMA is transferring data between the frame buffer and

external memory. Figure 3 shows the frame buffer interface in the system. Frame buffer interface to RC-Array is through a reconfigurable bus. To exchange data between the frame buffer and the RC Array first the bus configuration should be established by loading the appropriate data to the frame buffer configuration tables and then the read or write operation can be done. The flexibility of the frame buffer makes it simple to access data, reorder them and feed the processing elements as fast as possible.

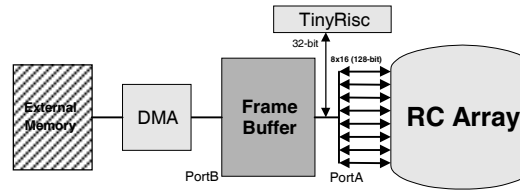

Figure 3. Frame Buffer Interface in M2

5 Algorithm Mapping and Performance Analysis

A number of kernels have been studied for the purpose of benchmarking and performance analysis of M2.

DCT. The forward and inverse DCT is used in MPEG encoders and decoders for transformation of image pixels to the frequency domain and back to the spatial domain. Chen's algorithm for an eight-point 1-D DCT is considered for mapping. M2 requires 21 cycles to complete 2-D DCT (or IDCT) on 8x8 block of pixel data. 2 extra cycles are required for loading/storing data from/into Frame Buffer. The cycle count for this benchmark on TMS320C64 is 154 and on C62 is 208.

FFT. We used the similar approach to DCT mapping for implementation of 64 point complex 2-D FFT. The computational cycles of 2-D FFT in this case is 42 cycles. This is much less than 276 cycles on C64 and 835 cycles on C62.

Correlation. For 64 point correlation of 16-bit real inputs, one can use one RC/point for first input stream. At each iteration the first

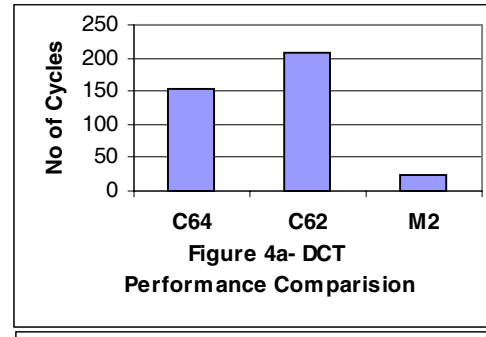

Figure 4a- DCT Performance Comparision

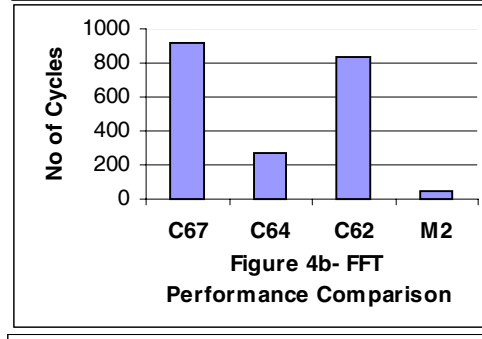

Figure 4b- FFT Performance Comparison

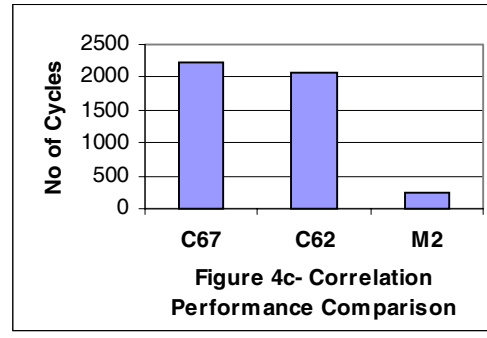

Figure 4c- Correlation Performance Comparison

input stream is loaded to all RCs. After increasing the address pointer of the first input, then the second input is loaded one by on to the whole RC and multiply add (MAC) operation is performed. The total number of cycles in this approach is 256 cycles and the result is 32 bit. Figure 4 compares the result of these benchmarks with TMS320 [7].

Conclusion

In this paper M2, a new implementation of MorphoSys has been introduced. M2 follows the basic concepts of MorphoSys, but it is optimized for computation intensive applications. It is a coarse grain reconfigurable architecture, with a set of fine grain blocks. Its memory architecture is highly optimized to overcome the high demands for data movement and shuffling in multimedia applications. Experimental results show that Morphoys architecture has a performance comparable to multimedia processors and ASICs.

Acknowledgments

This research was partially supported and funded by National Science Foundation (CCR-0083080), DARPA (F-33615-97-C-1126), and University of California CoRe Project in collaboration with Broadcom Corp (99-10061).

References

[1] M. Taylor, J. Kim, J. Miller, D. Wentzlaff, et al, “The Raw Microprocessor: A Computational Fabric for Software Circuits and General Purpose Programs,” IEEE Micro, Mar/Apr 2002.

[2] S. Goldstein, H. Schmit, M. Mao, M. Budiu, S. Cadambi, R. Taylor and R. Laufer, “PipeRench: A Coprocessor for Streaming Multimedia acceleration,” Proceeding of the 26th Annual Symposium on Computer Architecture, pages 28-39, May 1999.

[3] T. Callahan, J. Hauser, and J. Wawrzynek, “The Garp Architecture and C Compiler," *IEEE Computer*, April 2000.

[4] H. Singh, et al, “MorphoSys: An Integrated Reconfigurable System for Data-Parallel and Computing-Intensive Applications”, IEEE Transaction on Computers, Vol. 49, No. 5, May 2000.

[5] A. Abnous, C. Christensen, et al, “Design and Implementation of the TinyRISC Microprocessor,” Microprocessors and Microsystems, Vol. 16, No. 4, pp. 187-194, 1992.

[6] H. Parizi, “MorphoCell: A Reconfigurable Cell Targets DSP applications,” Technical Report, University of California, Irvine, April 2001.

[7] Texas Instrument online documents, “http://www.ti.com/”