HOSMII: A Virtual Hardware Integrated with DRAM

Yuichiro Shibata,[1] Hidenori Miyazaki,[1] Xiao-ping Ling,[2] and Hideharu Amano[1]

[1] Dept. of Computer Science, Keio University
[2] Dept. of Information and Computer Science, Kanagawa Institute of Technology
{shibata, miyazaki, ling, hunga}@aa.cs.keio.ac.jp

Abstract. WASMII, a virtual hardware system that executes dataflow algorithms, is based on an MPLD, an extended FPGA with multiple sets of configuration SRAM. Although we have developed an emulation system and software environment for WASMII, it has tended to be unrealistic due to the difficulty of the MPLD implementation. However, with recent technologies of semiconductors, an FPGA and DRAM can be implemented into a single LSI chip. We propose novel virtual hardware called HOSMII using such an FPGA/DRAM chip which can hold hundreds of configuration data and switch them instantaneously.

1 Introduction

FPGAs (Field Programmable Gate Arrays) have made possible new varieties of *reconfigurable systems* and *custom computing machines* [1][2]. In these systems, an algorithm is translated into an FPGA configuration and then executed directly in hardware. However, two major obstacles have limited the use of these systems: first, there is few standard, practical methods of translating an algorithm into an FPGA configuration; second, the size of available FPGAs has restricted their applications to small problems.

To address these problems, we have designed a data-driven virtual hardware system called WASMII [3] based on an MPLD (Multifunction Programmable Logic Device), which is an FPGA extended to provide multiple sets of internal configuration SRAM. An algorithm to be executed on WASMII is written in dataflow language and then translated into a collection of FPGA configurations. Some backup memory units are also attached outside the chip and configuration data is loaded to an unused module of the configuration SRAM during execution.

Although we have developed an emulation system and software environment for WASMII, it has tended to be an unrealistic system due to the difficulty of the MPLD implementation. However, with recent technologies of semiconductors, an FPGA and DRAM can be implemented into a single LSI chip. By using the column buffer of the DRAM array as configuration memory of the FPGA, replacement of the configuration data can be done almost the same speed as an MPLD. Compared with the MPLD approach, a large amount of data can be stored in the integrated DRAM.

2 WASMII

The original WASMII architecture is based on a dataflow paradigm of computation. In this paradigm, a program is written in dataflow language and then translated into a dataflow graph. Each node of the graph corresponds to a function — a simple function such as an adder, multiplier, comparator or possibly a more complex function. Input data to a node is called a token. A node executes its function when all its needed tokens arrive at its input ports. Using WASMII, nodes and links of a graph are directly represented in an FPGA configuration. This direct representation distinguishes WASMII from most other dataflow machines.

A direct representation of a dataflow graph in hardware limits the size of the graph to the size that the hardware can represent. WASMII bypasses this limitation by reconfiguring the hardware dynamically, using additional memory to store the requisite configurations.

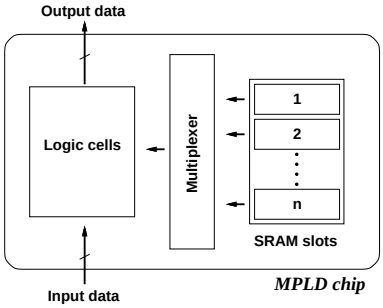

Fig. 1. Structure of an MPLD

WASMII is based on Fujitsu's MPLD [4], an extended FPGA that implements multiple sets of functions on a single chip. As shown in Fig. 1, the MPLD chip contains a set of SRAM slots. Each SRAM slot contains a *configuration page* that represents a dataflow graph. Using a multiplexer, a specified configuration page called an *activated page* is selected and configured. Selecting a new SRAM slot, in effect, is corresponding to the reprogramming of the MPLD's logic. The same idea is also addressed by Xilinx [5].

The MPLD's capacity can be extended by connecting the additional off-chip memory through a bus. We call this mechanism *virtual hardware*, in analogy to virtual memory. While one configuration page is being executed, another one can be loaded in parallel from backup memory; this action is called *preloading*.

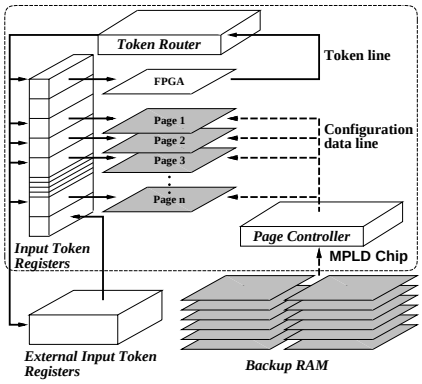

Fig. 2. Single chip WASMII

When the current configuration page of the MPLD is replaced by another one, all state information held in the MPLD's logic circuits are invalidated. Therefore we use a set of *token registers*, one per configuration page, to store the states of the computation as shown in Fig. 2.

When an activated page produces an output token, it is sent to the token register of the appropriate configuration page by a *token router*. A page is ready to be activated when all its input tokens

are present. When the current activated page has completed its computation and dispatched its all of output tokens, one of ready pages is selected to be replaced.

3 HOSMII

FPGA integrated with DRAM. Although we have developed an emulation system and software environment for WASMII [6], it has been unrealistic as no MPLD has been fabricated. The ratio of the area for logic and configuration SRAM is about 3 or 4 to 1. This means that the number of SRAM slots of an MPLD is limited to a small number. In this case, the communication between an unused SRAM slot and off-chip backup memory degrades the performance.

However, with recent technologies of semiconductors, an FPGA and DRAM can be implemented into a single LSI chip. In such a chip, the column buffer of the DRAM array can be used as a configuration memory of the FPGA directly. MIT's DPGA [7] and NEC's FPGA/DRAM chip [8] are examples of such chips.

Fig. 3 shows the concept of such an FPGA chip integrated with DRAM. In these chips, by the use of the large bandwidth provided by a wide internal bus, replacement of the configuration data can be done almost the same speed as an MPLD. Compared with the MPLD approach, a large amount of data (at least 20 times as that of an MPLD) can be stored in the integrated DRAM. Moreover, using small DRAM arrays connected with logic cells and controller units including token registers and the router, flexible partial replacing of dataflow graphs can be implemented. Hence, an FPGA with DRAM chip can be a hopeful candidate for a relief of an MPLD in WASMII.

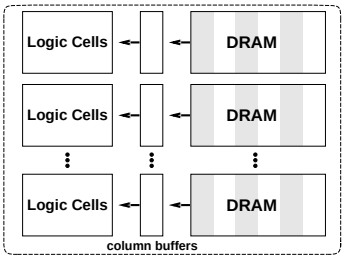

Fig. 3. FPGA/DRAM chip

HOSMII architecture. HOSMII is novel virtual hardware that exploits an advanced FPGA integrated with DRAM. Fig. 4 shows an example structure of HOSMII chips. For this chip, the backup RAM is almost unnecessary since 256 pages of configuration data each of which is corresponding to 19,000 gates can be stored in DRAM inside the chip [8]. As shown in this diagram, several HOSMII units might be implemented on one chip. Here one example of such a structure in which four HOSMII units are connected to form a ring structure is shown. The number of units and connection topology are design choices of a HOSMII chip.

Each HOSMII unit consists of two blocks; for control and for execution. The control block provides the hardware for management of pages and tokens such as token registers, the token router and the page controller. This hardware is also reconfigurable from attached DRAM arrays for adapting the dataflow graph in an execution block. In the execution block, the dataflow graph is directly configured and executed.

Unlike original WASMII, the execution unit does not need to work in a pure dataflow manner. In WASMII, data stored in each flip-flop is disappeared when the page is replaced. However, in NEC's FPGA/DRAM chip, the column buffer can be also used as flip-flops in a configured logic circuit, and can be swapped to the DRAM modules. Using this mechanism, each node of the dataflow graph can hold its own states and the number of tokens to transmit between pages can be reduced. In the current front end of WASMII, this description is not supported. We are now investigating a modified dataflow model and its description language.

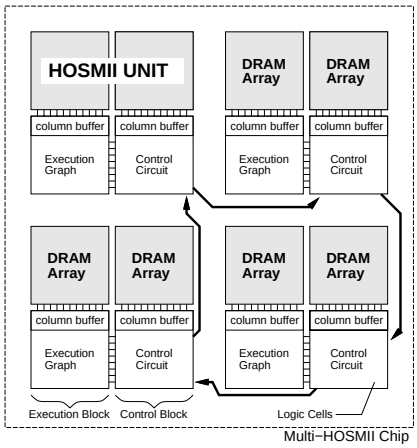

Fig. 4. HOSMII chip

4 Evaluation

Execution performance. As a feasibility study of HOSMII, some forms of virtual hardware are evaluated through simulation. The application adopted here is a Hopfield's style neuron model [9] for solving the 16-Queens problem and translated into a dataflow graph with 3,746 nodes. Parameters for the simulation are estimated based on Xilinx's MPLD and NEC's FPGA/DRAM chip as shown in Table 1. The number of logic gates for execution graphs of WASMII is settled to be equal with that of HOSMII for easy comparison. Seven systems shown in Table 2 are evaluated.

In this neural network example, each node operates data of 5 bit width which is sufficient for the convergence to the solution. As a result of actual mapping of the node onto an XC4000E FPGA, it is shown that 10,000 gates of logic cells can execute 20 nodes of a graph at 33 MHz. The seven systems are modeled in VHDL at the RTL level and simulated using the above-mentioned parameters with Mentor Graphics' QuickHDL simulator.

Table 1. Estimated parameters

Parameters	WASMII	HOSMII
Logic gates	10,000	10,000
SRAM/DRAM	1 Mb	16 Mb
Time for replace	30 ns	100 ns
Off-chip replace	100 μs	100 μs
Configuration pages	8	256

Table 2. Evaluated systems

Abbrev	Architecture	Units	Node with states
WA	WASMII	1	not supported
HO	HOSMII	1	not supported
HO1s	HOSMII	1	supported
HO2	HOSMII	2	not supported
HO2s	HOSMII	2	supported
HO4	HOSMII	4	not supported
HO4s	HOSMII	4	supported

Table 3. Graph decomposition

System	Pages	Nodes/Page	Reusability (%)
WA	576	20	89.9
HO1	578	20	89.9
HO1s	322	20	85.2
HO2	666	10	86.6
HO2s	500	10	76.0
HO4	1280	5	88.6
HO4s	1024	5	87.9

The target graph is decomposed into subgraphs as shown in Table 3. In these systems, subgraphs that have the same figure can be reused without the replacement procedure to reduce the time for swapping the configuration data. The dataflow graph of this network has high degree of regularity as Table 3 shows.

The results of simulation are shown in Fig. 5. The execution time of original WASMII is clearly dominated by the communication with off-chip memory even under the high degree of page reusability (over 80%). With HOSMII, the execution time is much improved as the whole graph can be stored inside the chip. Moreover using the multiple units inside the chip, the performance is further improved. With a special node which can keep its states inside the node, at last the execution time is improved about 43 times than that of original WASMII.

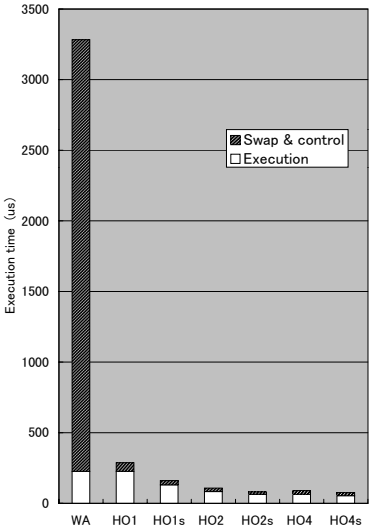

Fig. 5. Execution time of 1 iteration

Overhead of the control mechanism. Generally this kind of parallel system easily suffers a large overhead of data communication. Also in HOSMII, an unsophisticated token transferring mechanism may cause a *stall*. That is, there is no page to be activated and a unit must wait for tokens from other units without execution. This situation degrades the system performance severely. To address this problem, the time required for transferring a token from an execution block to its destination register and the breakdown of the operation time are evaluated.

As Table 4 shows, the average latency of the transmission of a token is 14 clock cycles, and this latency increases to 65 clock cycles in the worst case.

Table 4. Latency of token transmission

System	Max. (ns)	Min. (ns)	Avr. (ns)
HO2	270	30	150
HO2s	270	30	150
HO4	1950	30	148
HO4s	1950	30	148

Table 5. Breakdown of the operation time

System	Execution	Page swap	Token issue	Stall
WA	6.8 %	92.4 %	0.8 %	0.0 %
HO1	78.6 %	12.6 %	8.8 %	0.0 %
HO1s	81.4 %	12.1 %	6.5 %	0.0 %
HO2	76.9 %	13.8 %	9.3 %	0.0 %
HO2s	78.8 %	13.6 %	7.6 %	0.0 %
HO4	71.3 %	16.8 %	11.9 %	0.0 %
HO4s	71.1 %	17.2 %	11.7 %	0.0 %

On the other hand, the breakdown of the operation time in Table 5 shows that there is no occurrence of the stall situation as far as this application is concerned. This means that the page management manner based on a data-driven method can effectively hide the latency of transfers of tokens and makes the best use of parallelism. In this example, the pure execution time in which calculation is actually done by each node accounts more than 70% and the replacement of pages does not bottleneck the system any longer.

5 Conclusion

The virtual hardware called HOSMII is proposed. The initial simulation results show that the problem of original WASMII with an MPLD is almost solved. We are currently investigating a new description language, programming environment, and applications for HOSMII.

Acknowledgment. The authors would like to express their sincere gratitude to Dr. Motomura at NEC for useful discussion, and Mentor Graphics for their support of the tools we used.

References

1. T. Miyazaki: "Reconfigurable Systems: A Survey" Proc. ASP-DAC '98, 7C–1, 1998.
2. H. Amano, Y. Shibata: "Reconfigurable Systems: Activities in Asia and South Pacific" Proc. ASP-DAC '98, 7C–3, 1998.
3. X.-P. Ling, H. Amano: "WASMII: A Data Driven Computer on a Virtual Hardware" Proc. FCCM '93, pp. 33–42, 1993.
4. S. Yoshimi (Fujitsu Corp.): "Multifunction Programmable Logic Device" Patent (A) HEI–2–130023, 1990.
5. S. Trimberger, D. Carberry, A. Johnson and J. Wong: "A Time-Multiplexed FPGA" Proc. FCCM '97, pp. 22–28, 1997.
6. X.-P. Ling, Y. Shibata, H. Miyazaki, H. Amano, K. Higure: "Total System Image of the Reconfigurable Machine WASMII" Proc. PDPTA, pp. 1092–1096, 1997.
7. E. Tau, I. Eslick, D. Chen, J. Brown, A. DeHon: "A First Generation DPGA Implementation" Proc. FPD'95, pp. 138–143, 1995.
8. M. Motomura, Y. Aimoto, A. Shibayama, Y. Yabe, M. Yamashina: "An Embedded DRAM-FPGA Chip with Instantaneous Logic Reconfiguration" Proc. Symposium on VLSI Circuits, 1997.
9. Y. Takefuji: "Neural Network Parallel Computing" Kluwer Academic Publishers, 1992.

This article was processed using the LaTeX macro package with LLNCS style