

FPGA Accelerator for Subgraph Isomorphism Problem

Graduate Advisor: Shuichi Ichikawa

967751

Udorn Lerdtanaseangtham

1 Introduction

Subgraph isomorphism is applicable to various applications, including scene analysis and chemical structural formula database[1]. However, subgraph isomorphism problem is generally NP complete and hard to compute practically. Custom computing machinery is a promising approach to mitigate such problem[2].

Ullmann's algorithm[1], which is one of the most popular algorithm for subgraph isomorphism, can be implemented by combinatorial logic circuit, but it requires too much circuitry for practical use. Konishi[3] proposed a simple algorithm which is suited for hardware implementation. Though Konishi's algorithm is worse on search space reduction than Ullmann's algorithm, it is simple enough for the implementation on a state-of-the-art FPGA to get performance acceleration. Konishi[3] predicted that the custom circuitry operating at 33 MHz would show 10–50 times better performance than the software of Ullmann's algorithm on 333MHz Pentium-II.

This paper presents the implementation of Konishi's algorithm on OPERL board with Lucent OR2C 2C15A FPGA. The measured performance of the hardware is also shown.

2 Implementation

We implemented 2 units of Konishi's algorithm on an OR2C15A FPGA (Fig.1). Unit0 and Unit1 operate in parallel for twice throughput. Interface circuit decodes address to select the corresponding unit, to control and initialize the units, and to poll units for the status.

Each unit consists of tree-traversing circuit and edge-check circuit. The tree-traversing circuit enumerates possible combinations of vertices between graph G_α and G_β . The edge-check circuit checks the correspondence of edges from G_α to G_β . This hardware can handle G_α, G_β of maximally 15 vertices. The hardware resource and operating frequency are summarized in Table 1. PFU is a basic logic element of OR2C FPGA. An OR2C15A chip consists of 400 PFUs.

3 Evaluation

The measured performance of hardware (2 unit) is shown in Figure 2 for various number of vertices p_α, p_β when the edge density ed_α, ed_β are both 0.4. The specification of host computer is as follows: AMD K6-III 400MHz, 64MB main memory, gcc-2.7.2.1, with FreeBSD-2.2.1R. In Figure 2, the performance is shown as a ratio to the performance of software of Ullmann's algorithm executed on Intel Pentium-II 400MHz, 256MB main memory, gcc-2.8.1 with FreeBSD-3.1R. The software of Ullmann's algorithm was written in C language.

In Figure 2, the performance ratio is about 10–40 in the best case. On the other hand, performance is less than 1 in some cases, because the pruning ability of Konishi's algorithm is less than Ullmann's algorithm. In such a case, we had better use the software implementation of Ullmann's algorithm on host computer for better performance.

The cooperative approach is a method that can overcome such problem. In cooperative approach, the hardware will be used only when the performance ratio is expected to be greater than 1. Otherwise, the host processor performs Ullmann's algorithm to solve subgraph isomorphism problem. To predict which one would be better, the statistical data for various $p_\alpha, p_\beta, ed_\alpha$ and ed_β are used. Figure 3 shows the performance of cooperative approach, which shows better performance than Figure 2.

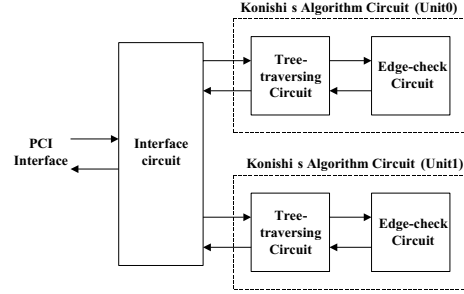


Figure 1: Block Diagram of Accelerator

Table 1: Hardware Resource and Operating Frequency

Circuit	Number of PFUs	Frequency(MHz)
Interface	23	33
Unit0, Unit1	160	16.5

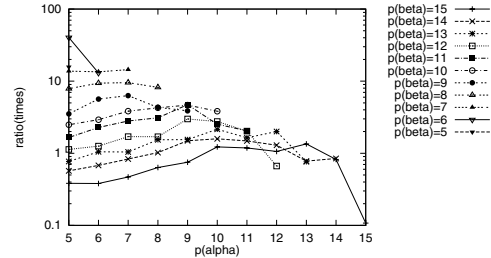


Figure 2: Performance of Accelerator (ed_α, ed_β) = (0.4, 0.4)

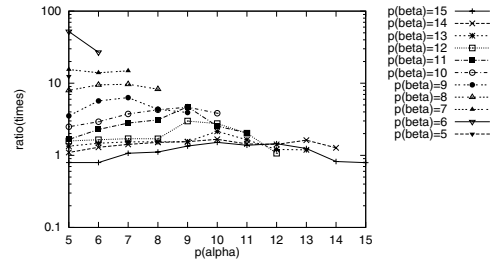


Figure 3: Performance of Cooperative Approach (ed_α, ed_β) = (0.4, 0.4)

4 Conclusion

Hardware accelerator for subgraph isomorphism shows 10–40 times better performance in the best case, compared to software implementation. More performance can be derived by pipelining hardware to operate at 33MHz. It is also easy to implement three or more units on a large-scale FPGA to get more performance.

References

- [1] J. R. Ullmann, "An Algorithm for Subgraph Isomorphism," *J. ACM*, Vol. 23, No. 1, pp. 31-42, 1976.
- [2] Duncan A. Buell et al., "Splash 2: FPGAs in a Custom Computing Machine," IEEE Computer Society Press, Los Alamitos, 1996.
- [3] Kouji Konishi, Shuichi Ichikawa, "An Implementation method for the Subgraph Isomorphism Judgment Algorithm using FPGA," 1999 IEICE (A-3-2), March 1999.
- [4] Shuichi Ichikawa, Toshio Shimada, "The trial and the evaluation for the reconfigurable board added to PCI bus," IEICE tech. reports CPSY96-97, pp. 159-166, 1996.