Data-dependent Hardware for Subgraph Isomorphism Problem

Graduate Advisor: Shuichi Ichikawa 003317 Shoji Yamamoto

1 Introduction

Subgraph isomorphism problems have various important applications, while generally being NP complete. Though Ullmann and Konishi proposed the custom circuit designs to accelerate subgraph isomorphism problem, they require many hardware resources for large problems. The previous study [3] described that the logic scale can be reduced when the circuit is implemented as data-dependent circuit. However, the data-dependent circuits were not implemented in the previous study.

In this study, the data-dependent circuits for subgraph isomorphism problem are designed and evaluated. The purposes of this study are to evaluate the logic scale, execution time and generation time of data-dependent circuits of two algorithms by Ullmann and Konishi, on YDK MIRE-MULTI3000 board having Xilinx XC2V3000.

2 Subgraph isomorphism problem

Assume that a graph G is defined by (V, E), in which V is the set of vertices and E is the set of edges. A graph $G_\alpha = (V_\alpha, E_\alpha)$ is subgraph of another graph $G_\beta = (V_\beta, E_\beta)$, if both $V_\alpha \subseteq V_\beta$ and $E_\alpha \subseteq E_\beta$ hold. G_α is isomorphic to G_β, if and only if there is a 1:1 correspondence between V_α and V_β that preserves adjacency. The subgraph isomorphism problem is a decision problem to determine whether G_α is isomorphic to a subgraph of G_β. The subgraph isomorphism problem is a kind of search problem to search a mapping function from V_α to V_β that satisfies the adjacency. The custom circuit for this problem is composed by an enumeration unit and a pruning unit.

3 Data-dependent hardware

Generally, the logic circuit can be reduced, if any input of the logic circuit turns out to be constant (Fig.1). This reduction can be applied recursively, consequently reducing the logic scale of the circuit. The derived circuit would operate at a higher frequency than the original one, because the logic depth and wiring delay can thereby be reduced. As the consequent circuit becomes dependent on the input data instance, it is called a *data-dependent circuit* in this study.

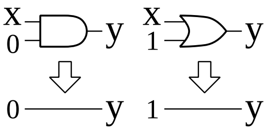

Figure 1: Logic Reduction

The obvious drawback of the data-dependent approach is that the derived circuit is not reusable, because it is specialized to an input instance. This naturally means that we have to generate the circuit for each input and that reconfigurable devices such as FPGA must be used. The total execution time T of a data-dependent circuit is given by the sum of the circuit generation time T_{gen} and the execution time T_{exec}.

4 Design

Ullmann's circuit (Uo) is based on *INDEP* in the paper [3]. *INDEP* was targeted for Lucent OR2C FPGA. Uo is re-targeted for Xilinx XC2V FPGA in this study.

Ullmann's data-dependent circuit (Ud) is the data-dependent version of Ullmann's circuit (Uo). Ud is based on *BOTH2* in the paper [3]. Ud is re-targeted for Xilinx XC2V FPGA in this study.

Konishi's circuit (Ko) was originally implemented in the preceding study [2]. As described in Sec. 3, the logic reduction can't be applied beyond memory elements. The implementation [2] has the sequential version of the pruning circuit, and doesn't suit for data-dependent circuit. In this study, a new pruning circuit is implemented by combinational logic circuit.

Konishi's data-dependent circuit (Kd) is the data-dependent version of Konisi's circuit (Ko).

5 Evaluation

In following discussion, $p = p_\alpha = p_\beta$ is assumed. For each p, 100 pairs of random graph were generated. Random graphs have a parameter called edge probability, which are set to 0.3 and 0.6 for G_α and G_β, respectively.

T_{gen} is the sum of processing time of data-dependent HDL source code generation, logic synthesis CAD "Synopsys FPGA Compiler II 3.7.0" and FPGA implementation CAD "Xilinx ISE 5.2i" on Athlon XP 2600+ PC (1GB memroy, Windows 2000 professional SP4). A parallel I/O board "ADTEK aPCI-P31A" was used to sense termination signal to measure T_{exec}. Soft designates the precessing time of software implementation of Ullmann's algorithm on Athlon XP 2600+ PC (512MB, Redhat Linux 9).

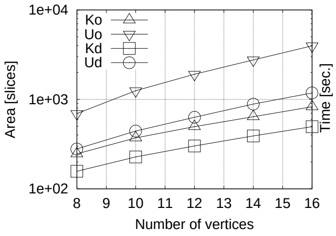

Figure 2: Area

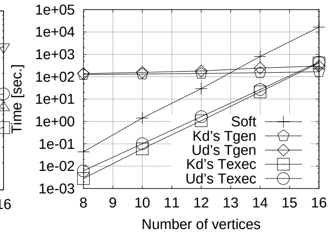

Figure 3: T_{gen} and T_{exec}

For $8 \leq p \leq 16$, the logic scale of Ud is 40.3–29.7% of Uo, while Kd is 63.8–59.9% of Ko (Fig. 2). T_{gen} of Ud is 1.1–1.7 times larger than Kd (Fig. 3). Generally, T_{gen} depends on the logic scale. Since the logic scale of Kd is smaller than Ud, T_{gen} of Kd is shorter than Ud. The performance of Kd is 15.7–40.1 times faster than Soft (Fig. 3). The efficiency of algorithm depends on the input data [2], and Kd was more effective than Ud in this input data set.

6 Conclusion

Data-dependent circuit is small in logic scale, and therefore we can implement the data-dependent circuit for large p in the same logic scale. However, data-dependent circuit incurs T_{gen}. In this study, data-dependent hardware for subgraph isomorphism problem was quantitatively evaluated for random input graphs. Even if the circuit generation time was included, Kd was 28.6 times faster than Soft ($p = 4$ in Fig.4). In this evaluation, Soft was operated at 2.1GHz clock, Kd and Ud were operated at 25MHz clock.

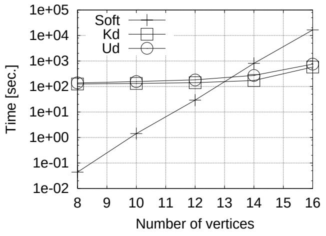

Figure 4: T

References

[1] J. R. Ullmann. An Algorithm for Subgraph Isomorphism. *J. ACM*, 23(1):31-42, 1976.

[2] S. Ichikawa and L. Udorn and K. Konishi. An FPGA-Based Implementation of Subgraph Isomorphism Algorithm. *IPSJ Transactions on High Performance Computing Systems*, 41(SIG5(HPS1)):39–49, 2000.

[3] S. Ichikawa and S. Yamamoto. Data dependent circuit for subgraph isomorphism problem. *IEICE Transactions on Information and Systems*, E86-D(5):796–802, 2003.