A Design of Rough Set Processor for Knowledge Discovery

Matsumoto Mitsuhiro
Dept. of Electronic Engineering
Tokyo Denki Univ.
2-2, Kanda-Nishiki-cho, Chiyoda-ku,
Tokyo, Japan 101-8457
E-mail: 05gmd17@ed.cck.dendai.ac.jp

Kanasugi Akinori
Dept. of Electronic Engineering
Tokyo Denki Univ.
2-2, Kanda-Nishiki-cho, Chiyoda-ku,
Tokyo, Japan 101-8457
E-mail: kanasugi@d.dendai.ac.jp

Abstract

This paper propose a basic design for rough set processor for knowledge discovery. In this paper, the architecture and the algorithm, the simulation, the experiment of the dedicated processor are shown.

1 Introduction

Rough set theory is widely paid to attention in recent years. This theory is very effective to the database including rough expressions and contradictions [1]. The processing of rough set is simple, but it is difficult to obtain the quick response. The design of the hardware based on the rough set theory have been reported, but the dedicated processor for the large-scale data mining has not been reported yet. The proposed design can be applied in the field of robotics, medical science, industry, and so on.

2 Architecture and Design

In the case of the huge database, it is necessary to treat the logical function that consists of almost two thousand variables. In order to process huge logical functions, three units named “Core-Selector”, “Covering-Unit”, and “Reconstruction-Unit” were designed [2]. “Core-Selector” and “Covering-Unit” reduce the data in the pre-processing, and “Reconstruction-Unit” extracts the rules in the post-processing. “Core-Selector” selects some core data and transfers the selected core data to reduct the logical function data. On the other hand, “Covering-Unit” deletes the data that can be deleted by using the selected core data. Figure 1 shows the core-data selection and the function-data reduction. The post-processing time of “Reconstruction-Unit” is reduced by the pre-processing by “Core-Selector” and “Covering-Unit”. “Reconstruction-Unit” discovers the important rules from the reduced logical function. In the post-processing, ”Reconstruction-

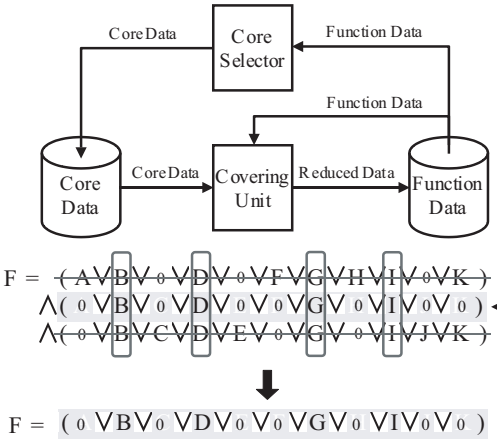

Figure 1: Core-Data Selection and Data Reduction

Unit” repeats the processing of the matrix data that is the reduced function data. Figure 2 shows the discovery of the impormant rules in the post-processing. “Reconstruction-

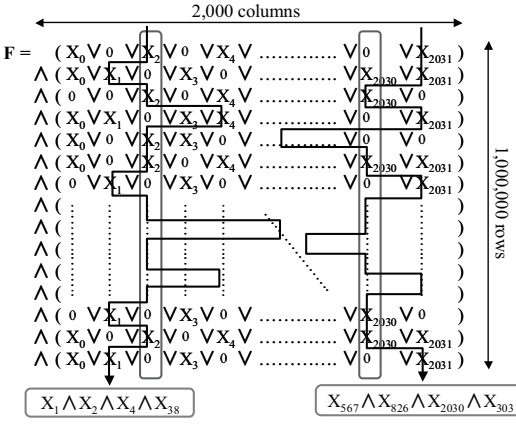

Figure 2: Discovery of the Important Rules

Unit” consists of “Reconstruct-Controller”, “Parallel-

Counter", and "Sorter". Figure 3 shows the block diagram of "Reconstruction-Unit". It is necessary for

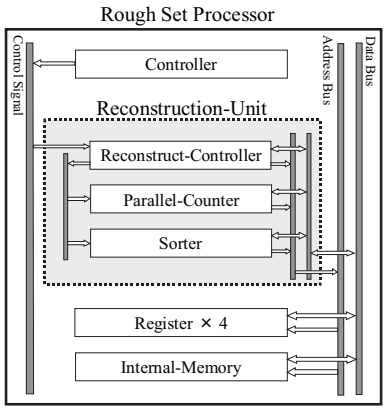

Figure 3: Block Diagram of "Reconstruction-Unit"

"Reconstruction-Unit" to repeat three steps to discover the important rules. The step 1 is the parallel count process to find the number of each variable in the logical function. The step 2 is the sort process to select the variable that has the largest counts. The step 3 is the delete process to delete the row that has the selected variable. The discovery of the most important rules is realized by repeating these three steps until all rows are deleted. Fugure 4 shows the schematic of "Reconstruction-Unit".

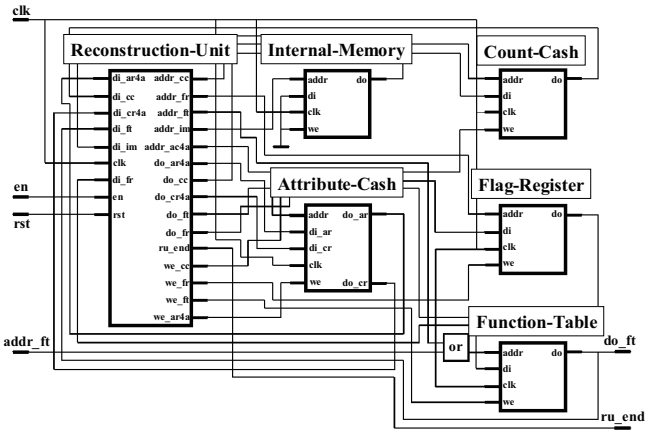

Figure 4: Schematic of "Reconstruction-Unit"

3 Experiments

The authors performed the design and the simulation using the logic synthesis tool (Xilinx ISE WebPACK 8.2i) and HDL simulator (Mentor Graphics ModelSim XE 6.1e). The experiment of the designed processor is performed on

Table 1: Gate Count for Design and Processing Time

unit name	gate count (gates)	time (μs)
Core-Selector (16 core-data)	9,400	82.5
Covering-Unit (16 core-data)	2,100	263.1
Reconstruction-Unit (4 rules)	36,200	4,152.7

the FPGA evaluation board (Xilinx Spartan-3E Starter Kit, 500 thousand gates). Table 1 shows the number of gates and the processing time of the proposed processor. Figure 5 shows the experiment of "Reconstruction-Unit" in the post-processing.

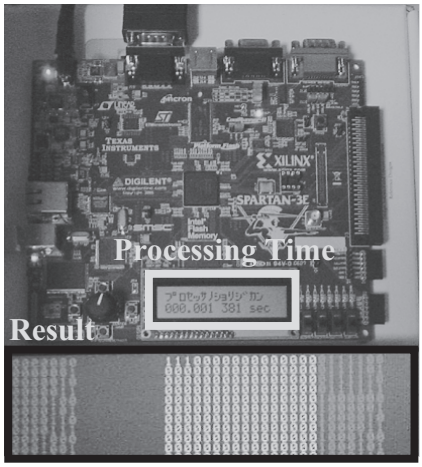

Figure 5: Experiment of "Reconstruction-Unit"

4 Conclusion

In this paper, the design and the experiment of rough set processor is described. The future works are the experiment with the actual database and the improvement of the processing speed.

This research is patically supported by Research Institute for Science and Technology of Tokyo Denki University under Grant Q03-04.

References

[1] Pal S K and Skowron A (1999), Rough Fuzzy Hybridization - A New Trend in Decision - Making, Springer Verlag.

[2] Kanasugi A (ed. by Inuguchi M et al.) (2002), Rough Set Theory and Granular Computing, Springer-Verlag, pp. 273-280.