GROSHEC: Is the processor for rough set methods in sight?

Maciej Kopczynski, Tomasz Grzes, and Jaroslaw Stepaniuk

Faculty of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
{m.kopczynski,t.grzes,j.stepaniuk}@pb.edu.pl
http://www.wi.pb.edu.pl

1 Introduction

The rough sets' theory developed in the eighties of the twentieth century by Prof. Z. Pawlak is an useful tool for data analysis. Therefore a lot of rough sets algorithms were implemented in scientific and commercial tools for data processing. But data processing efficiency problem is arising with increase of the amount of data. Software limitations led to searching the new possibilities.

Field Programmable Gate Arrays (FPGAs) are the digital integrated circuits which function is not determined during the manufacturing process, but can be programmed by engineer any time. One of the main features of FPGAs is the possibility of evaluating any boolean function. That's why they can be used for supporting rough sets calculations.

At the moment there are some hardware implementation of specific rough set methods. Detailed summary can be found in [1] and in [6].

None of the above are the complex solutions as they are only implementations of the specific rough sets method. Authors propose the fully operational System-on-Chip (SoC) named GROSHEC (**G**eneral **RO**ugh **S**et **H**ardware **E**nhanced **C**omputer) based on Altera NIOS II core. SoC was implemented in Altera's Stratix III FPGA using TerasIC DE3 development board. Some details on previous authors' work can be found in [5, 2, 3].

2 System architecture

Startix III FPGA contains processor control unit implemented as NIOS II embedded core with extended instruction set designed to support rough sets calculations. FPGA chip includes also hardware calculation blocks which act as a coprocessors for rough sets calculations.

Since the beginning of our research, following hardware calculations blocks have been designed, implemented and tested:

- blocks for calculating **upper and lower approximations**,
- module for creating **discernibility matrix**,
- two versions of modules that allows to obtain **reduct**,
- five versions of blocks designed to calculate the **core**.

3 Results and Conclusions

All of these blocks were synthesized into FPGA and tested in real chip. Functionally equal software implementations were prepared for each of mentioned methods to do time comparison tests.

Obtained results showed that the approximations and discernibility matrix for small datasets (about 100 objects, Diabetes dataset [4]) can be calculated by hardware over 200000 times faster than software. Calculating reduct is nearly 3000 times faster. For big dataset (10^6 objects, multiplied Diabetes) the core is calculated 10 times faster (648 times faster taking clock speed difference between PC and FPGA into consideration).

The hardware implementation is the main direction of using scalable rough sets methods in real time solutions focused on processing big data. Results obtained in experiments proved that proposed solution can be used for speeding up the calculations of rough sets methods, also for very big data sets.

Currently authors are preparing hardware implementation for big datasets attribute discretization. Results will be published soon.

Acknowledgements

The research is supported by the Polish National Science Centre under the grant 2012/07/B/ST6/01504 (Jaroslaw Stepaniuk, Tomasz Grzes) and by the scientific grant S/WI/3/2013 (Maciej Kopczynski).

References

1. M. Kopczyński, J. Stepaniuk: Hardware Implementations of Rough Set Methods in Programmable Logic Devices, Rough Sets and Intelligent Systems - Professor Zdzisław Pawlak in Memoriam, eds. Andrzej Skowron, Zbigniew Suraj, Intelligent Systems Reference Library 43, Heidelberg, Springer, 2013, pp. 309–321
2. M. Kopczyński, T. Grześ, J. Stepaniuk: FPGA in Rough-Granular Computing: Reduct Generation, WI 2014: The 2014 IEEE/WCI/ACM International Joint Conferences on Web Intelligence Vol.2, eds. Andrzej Skowron, Lipika Dey, Adam Krasuski, Yuefeng Li, Warsaw, IEEE Computer Society, 2014, pp. 364-370.
3. M. Kopczyński, T. Grześ, J. Stepaniuk: Generating core in rough set theory: Design and implementation on FPGA, Lecture Notes in Computer Science Vol.8537: Lecture Notes in Artificial Intelligence, Rough sets and intelligent systems paradigms RSEISP2014 : Held as Part of JRS 2014, eds. Marzena Kryszkiewicz, Chris Cornelis, Dave Ciucci, Jesus Medina-Moreno, Hiroshi Motoda, Zbigniew W. Raś, Berlin, Springer-Verlag, 2014, pp. 209-216.
4. J. Stepaniuk, Rough–Granular Computing in Knowledge Discovery and Data Mining, Springer, 2008.
5. J. Stepaniuk, M. Kopczyński, T. Grześ: The First Step Toward Processor for Rough Set Methods, Fundamenta Informaticae Vol. 127, 2013, pp. 429-443.
6. K. S. Tiwari, A. G. Kothari, Design and Implementation of Rough Set Algorithms on FPGA: A Survey, in: (IJARAI) International Journal of Advanced Research in Artificial Intelligence, vol. 3, no. 9, 2014, pp. 14–23.