THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

IEICE ICDV 2015

Database Processor (DBP)

— A New Search Engine for the Big Data Era —

Xuan-Thuan NGUYEN[†], Hong-Thu NGUYEN[†], Trong-Thuc HOANG[‡],
Katsumi INOUE[††], Osamu SHIMOJO[‡‡], Toshio MURAYAMA[‡‡], Kenji TOMINAGA[‡‡] and Cong-Kha PHAM[†]

[†] The University of Electro-Communications (UEC), Tokyo, Japan
[‡] The University of Science, VNU-HCM (HCMUS), Vietnam
[††] Advanced Original Technologies Co., Ltd (AOT), Tokyo, Japan
[‡‡] Nippon Computer Dynamics Co. Ltd (NCD), Tokyo, Japan
E-mail: [†]xuanthuan@vlsilab.ee.uec.ac.jp, pham@ee.uec.ac.jp, [‡]ugg44151@nifty.com, [††]kenji.tomi@ncd.co.jp

Abstract High-speed information detection plays an important role in many applications nowadays. As a result, new hardware architecture and computing technology using memory have become increasingly attractive to researchers so far. In this paper, we have been proposed a Search-Less Information Detection (SLID) with an excellent memory-based processor for data retrieval and analysis. Database Processor (DBP) is mainly concentrated on because it can operate with high speed, high precision, and low power. The experimental results in FPGA prove that this proposed DBP will likely become a new type of information detection processor for the big data society.

Keywords Database Processor (DBP), memory-based processor, big data, inverted index, Search-Less Information Detection (SLID)

1. Introduction

1.1. Metadata

In a traditional processor system, a large number of data are stored in memory. Therefore, the memory looks like an enormous burden of data and requires a large of time to search for the information. In order to reduce such time, the metadata is exploited. A simple definition of metadata is "data about data". It describes some information about data such as content, quality, condition, and other characteristics of data. Some significant purposes of metadata are described as below.

- Data browser: There are two levels of metadata. In the simple level, metadata describes the basic or general information of data, whereas in the compound level, metadata store detailed information of individual data set. Depend on specific requirement, the users can choose the appropriate level of metadata. Due to the information described in the metadata, it is easy to filter or search for information on metadata.
- Data transfer: Metadata can improve the searching process in memory. It contains the information of data such as value, index, characteristic and is considered as a part of the map. Therefore, metadata is easily transferred between map producers and users.
- Documentation: Metadata describes from basic to detailed information of data, which requires to understand and effectively use the data. The metadata such as the hash table, tree algorithm, binary search, inverted index are generated from the original data.

In spite of many advantages, metadata still possesses some challenges. Firstly, it represents the data, thereby being necessary to select and restore useful information of data. As a result, the number of metadata such as the index is too large. Sequentially, the system becomes heavy and requires a memory space up to several times in comparison with its original data. Secondly, whenever the original data changes, the metadata needs some modifications. If there

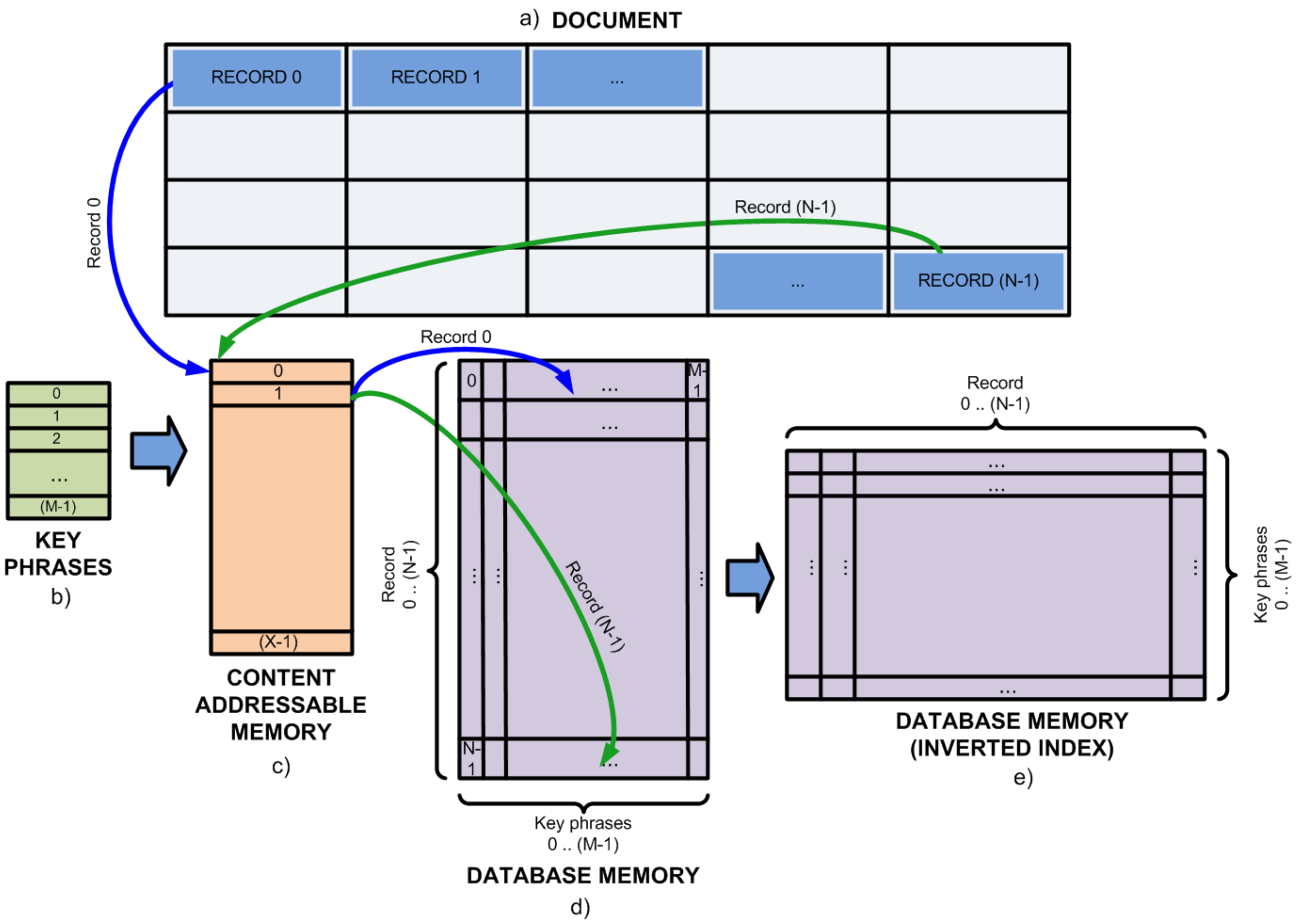


Figure 1: The illustration of DBP system.

is a heavy data, it takes the time to modify the metadata. Thirdly, the performance of metadata system is extremely degraded because if the metadata indexes have been overused, the system becomes a heavy burden of data. For instance, replacing the order of data demands the sort of entire document.

1.2. Previous work

In order to overcome the disadvantages of metadata and essentially solve the bottleneck of searching information, many approaches have been proposed so far [1]-[3]. Besides, one novel method so-called Search-Less Information Detection (SLID) was presented in previous work. The core search is a kind of memory-type processor namely Set Operating Processor (SOP) that acts like a human brain with efficient real-time processing for the recognition applications such as the image, speech, and character recognition [4]-[7]. The SOP applications in real-time processing recognition intend to speed up the critical pattern matching process. It is also possible to be applied to the analysis of full-text search and three-dimensional search.

The previous SLID was implemented on hardware system without increasing the detection time and system complexity. The architecture was largely based on Content Addressable Memory (CAM) that exploits the FPGA parallel structure [4]-[8] to achieve the fast detection, accelerate the search performance, but consume the least resource. This CAM was implemented based on the dual-port RAM structure due to their similarity. The most advantages of proposed system was CAM blocks can operate with parallel match output and process operations with simple logic circuits. Therefore, the system achieves low resource utilization and fast data detection.

Because the hardware structure for SLID is a completely new concept of information processing, we have been researching and developing this proposed memory-type processor. The combination of SLID memory-based processor, e.g. SOP, DBP, with the conventional processor, i.e. CPU, and inheriting of the current software assets can overcome the biggest weakness of the current computer architecture. In this paper, a Database Processor (DBP) is proposed. DBP is the best device for using the big data. In the DBP, after receiving the conditional input searching, DBP process a massively simple parallel operation to get the output results. Therefore, the DBP can operate in high speed, high precision and low power [8].

The remains of this paper are organized as follows. Section 2 describes in detail Database processor as well as its application example. Section 3 presents the experimental results of the proposed system. Section 4 finally gives the conclusion.

2. Database Processor (DBP)

2.1. Overview

Fig. 1a presents an example of N-record document and suppose that we want to determine which records contain the query *phrase 0* AND *phrase 1* AND NOT *phrase 2*. The simplest way to do that is to start at the beginning and to read through the document, noting for each record whether it contains *phrase 0* and *phrase 1* and excluding it from consideration if it contains *phrase 2*. It can be seen that searching through text can be a very effective process, especially given the speed of modern computers. Additionally, it often allows useful possibilities for wildcard pattern matching through the use of regular expressions. However, this traditional method may not process large document collections quickly. In fact, the amount of online data has grown at least as quickly as the speed of computers and then leads to a total of billions to trillions of words that need to be searched.

The way to avoid linearly scanning the texts for each query is to index the records in advance. A hardware system including an SOP and DBP is proposed to achieve a high-performance data retrieval system. Each record is initially loaded to SOP, a CAM-based highly efficient information detection, to generate the necessary indexes. The best advantages of CAM is to seek out the particular key phrases in a certain record and then return the position of matching results with low latency due to the parallelism in FPGA. For example, the value of one of bit j in row i of DBP memory asserts the presence of key phrase j in the record i. After generating all indexes, inverted file is generated by performing matrix transposition in DBP memory. The inverted file [9]-[10] is one of the most efficient index structure for text query evaluation. Table 1 then shows how the answer to the example above is effectively obtained owing to the inverted index. In fact, we take the vectors for *phrase 0*, *phrase 1*, and *phrase 2*, complement the last, and then do a bitwise AND, i.e. *1101...1* AND *0100...1* AND *0100...1* = *0100...1*. The final answer proves that at least record 1 and (N-1) contains the defined query.

Table 1: An example of inverted file.

	Rec. 0	**Rec. 1**	**Rec. 2**	**Rec. 3**	...	**Rec. (N-1)**
Phrase 0	1	1	0	1	...	1
Phrase 1	0	1	0	0	...	1
Phrase 2	1	0	1	1	...	0
...	...	...	...	...	...	...
Phrase (M-1)	0	1	1	1	...	0

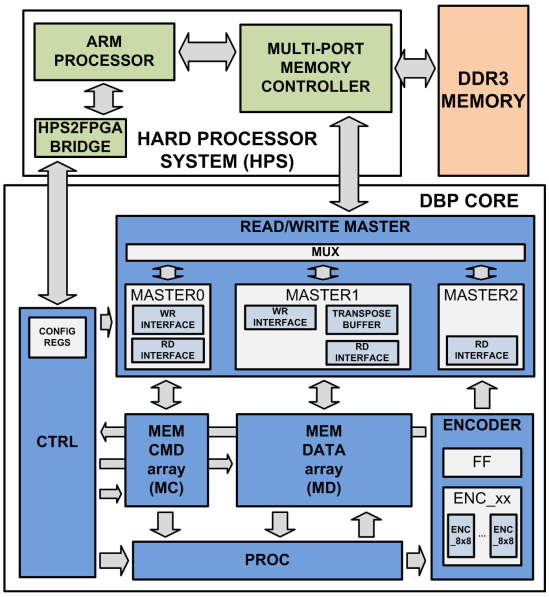
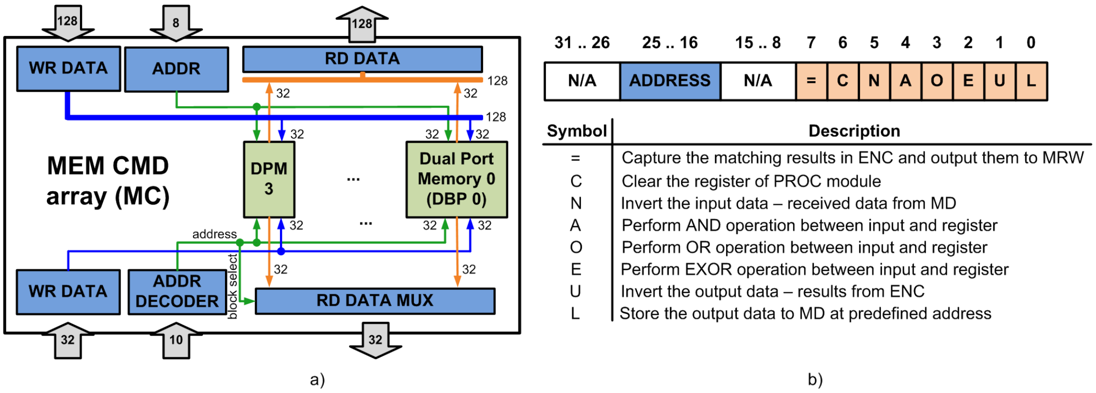


Figure 2: The hardware architecture of DBP.

2.2. Architecture

The proposed DBP system is composed of two parts: an ARM-based Hard Processor System (HPS) and a DBP hardware core, which is illustrated in Fig. 2. The Linux-based software is implemented in HPS to manage the TCP/IP communication between our system with other devices. The DDR3 memory temporarily stores data and commands from external devices. Besides, configuration parameters are transferred from HPS logic to FPGA fabric and vice versa by 128-bit-data-width HPS-to-FPGA bridge due to their low bandwidth requirement. On the other hand, Direct Access Memory (DMA) mechanism is deployed to provide the high-speed link so that both command and data are transmitted as fast as possible. Multi-port memory controller guarantees the simultaneous connections between ARM processor and DBP core. DBP core, the heart of DBP system, includes six main modules whose functions are de-



31 .. 26	25 .. 16	15 .. 8	7	6	5	4	3	2	1	0
N/A	ADDRESS	N/A	=	C	N	A	O	E	U	L

Symbol	Description
=	Capture the matching results in ENC and output them to MRW
C	Clear the register of PROC module
N	Invert the input data – received data from MD
A	Perform AND operation between input and register
O	Perform OR operation between input and register
E	Perform EXOR operation between input and register
U	Invert the output data – results from ENC
L	Store the output data to MD at predefined address

Figure 3: The hardware architecture of MEM_CMD (a) and the structure of command (b).

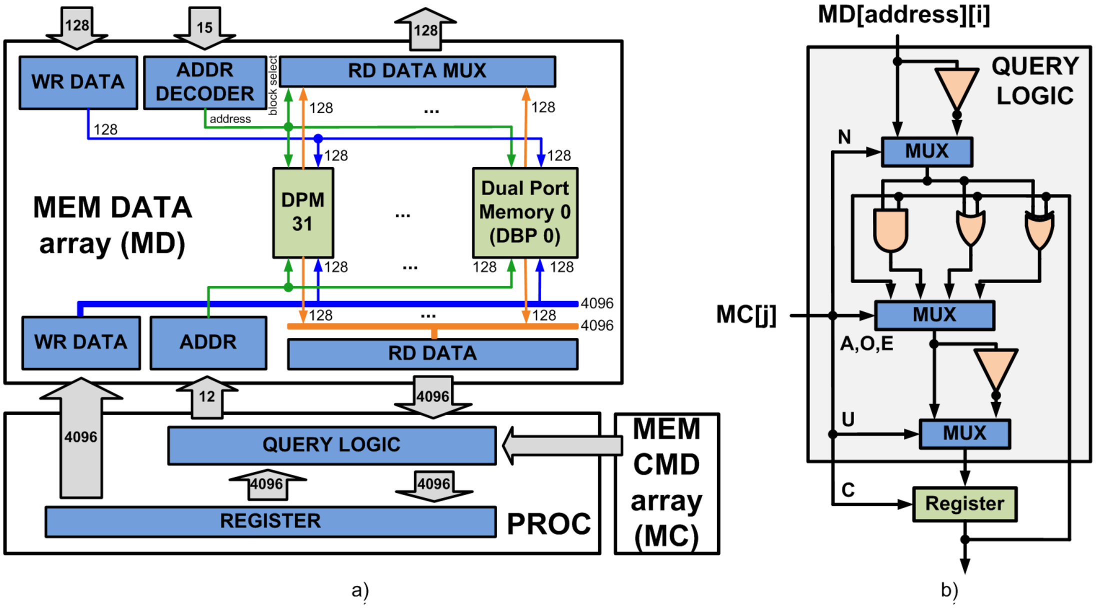


Figure 4: The hardware architecture of MEM_DATA (a) and the structure of query logic (b).

Table 2: An example of command for DBP core.

Command	Description of command function
0x0000 0040	Clear the register of PROC
0x0000 0008	Store the MD[0] to PROC by OR command
0x0001 0010	Perform AND between MD[1] and PROC
0x0002 0030	Perform invert MD[2] and then AND with PROC
0x0000 0080	Output the match record

scribed as follows.

- CONTROL module (CTRL): contains several internal registers to keep the configuration parameters receiving from HPS. It also holds the result and status signals of DBP core.

- READ/WRITE MASTER module (MRW): receives and sends command, data, and matching results from and to HPS through MASTER 0, MASTER 1, and MASTER 2, respectively. The 128-bit 50-MHz DMA channel is deployed to improve the transaction bandwidth up to 6.4 Gbps between DBP core and multi-port memory controller.

- MEM CMD array (MC): stores 1024 32-bit commands of DBP, as shown in Fig. 3a. Because the data width of MASTER 0 is 128 bits, four 32-bit dual-port memories (DBP) are exploited in parallel so that received command are put into each row of all memories in turn. The 32-bit outputs, on the contrary, are read out consecutively by the address decoder and multiplexer. According to Fig. 3b, one command includes 10-bit address, i.e. key phrase, and 8-bit operations namely *L*,

Table 3: Hardware utilization of DBP core using FPGA

	DBP 4K x 1K	**DBP 8K x 1K**	**DBP 16K x 1K**
Device	Arria V 5ASTFD5K3F40I3	Arria V 5ASTFD5K3F40I3	Arria V 5ASTFD5K3F40I3
Logic utilization (in ALMs)	25,976 (15%)	46,881 (27%)	78,917 (45%)
- Combinational ALUTs	- 39,131	- 64,754	- 110,501
- Dedicated registers	- 34,953	- 59,461	- 108,562
Block memory bits	4,227,584 (18%)	8,421,888 (36%)	16,996,864 (73%)
Frequency (MHz)	70.5	68.2	64.1

U, *E*, *O*, *A*, *C*, and =. The function of each operation is described in the table below.

- MEM DATA array (MD): holds the similar structure to MC, which includes a memory array, address decoder, and multiplexer. As can be seen in Fig. 4a, 32 128-bit DBP are arranged in parallel to form a 1,024-row MD where each row contains 4,096 bits. In other words, 128-bit data are loaded sequentially to MD from left to right and up to bottom while 4096-bit data are pushed out in one clock.

- PROCESS module (PROC): performs the query operations. To begin with, query logic analyzes the commands in MC and then requests corresponding data in MD. After completing each computation, the result is stored back to the register for next operations. Fig. 4b shows the bit-level architecture of query logic with three multiplexers and several fundamental logic gates. The first and last multiplexer decide whether the input and output must be inverted or not. On the other hand, the second multiplexer selects the operations, i.e. *A* AND, *O* OR, and *E* EXOR. Operator = and *L* are applied directly to ENC and MD module, respectively, while *C* clears the register of PROC.

- ENCODER module (ENC): looks for all the records that contain the query and then dispatched them to MASTER2 module. Besides, the number of matching result is transmitted to CTRL.

Assume the example above, in order to generate the query, six commands are employed. Firstly, the register of PROC is cleared and then data of MD[0] is stored to PROC by OR command. The next two commands calculate the results and the final command outputs the number of matched record.

3. Experimental Results

The proposed DBP system is validated by an Arria V development board containing an Altera Arria V 5ASTFD5K3F40I3 FPGA and a 1-GB external DDR3 SDRAM. The software on HPS receives TCP/IP streaming data and then transfers them to the embedded memory and configuration registers in DBP core. Table 3 shows the hardware resource utilization of DBP when the number of key phrases of 1,024 and the number of records is 4,096, 8,192, and 16,384. If each record is 4 KB, one DBP can manage from 16 MB to 256 MB. The DBP core communicates with HPS at the bandwidth up to 6.4 Gbps. Furthermore, PROC and ENC module require three and two clocks for one operation, respectively. In other words, if the number of records is 16,384, PROC module attains the approximate bandwidth of 273 Gbps.

4. Conclusion

We have proposed the architecture of DBP which can apply to a novel ultra high-speed data retrieval system. The design was verified in FPGA to prove the performance. Total performance becomes constant irrespective of the storage capacity of the DBP. The information processing capability is in proportion to the communication speed.

References

[1] Rene Mueller, Jens Teubner, Gustavo Alonso, ”Data Processing on FPGAs,” Jour. Proc. VLDB Endowment, Vol. 2, Iss. 1, Aug. 2009.

[2] Bharat Sukhwani, H. M., M. T., P. D., B. I., B. B., D. D., Sameh Asaad, ”Database Analytics Acceleration using FPGAs,” Proc. 21st Int. Conf. Parallel architectures and compilation techniques (PACT), pp. 411-420, Sep. 2012.

[3] Werner S., Groppe S., Linnemann V., Pionteck T., ”Hardware-accelerated Join Processing in Large Semantic Web Databases

with FPGAs," Int. Conf. High Performance Computing and Simulation (HPCS), pp. 131-138, Jul. 2013.

[4] Duc-Hung Le, Katsumi Inoue, Masahiro Sowa and Cong-Kha Pham, "An FPGA-based Information Detection Hardware System Employing Multi-Match Content Addressable Memory," IEICE Trans. Fundamentals of Electronic, Communications and Computer Sciences, Vol. E95-A, No. 10, pp. 1708-1717, Oct. 2012.

[5] Katsumi Inoue, Duc-Hung Le, Masahiro Sowa and Cong-Kha Pham, "Set Operating Processor (SOP): Application for Image recognition," The Institute of Electronics, Information and Communication Engineers, Technical report of IEICE. VLD 113 (235), pp. 35-40, Sep. 2013.

[6] Duc-Hung Le, Katsumi Inoue and Cong-Kha Pham, "Design a Fast CAM-based Exact Pattern Matching System on FPGA and 0.18μm CMOS process," IEICE Trans. Fundamentals of Electronic, Communications and Computer Sciences, Vol. E96-A, No. 9, pp. 1883-1888, Sep. 2013.

[7] Duc-Hung Le, Katsumi Inoue and Cong-Kha Pham, "A CAM-based Information Detection Hardware System for Fast Image Matching on FPGA," IEICE Trans. on Electronics, Vol. E97-C, No. 1, pp. 65-76, Jan. 2014.

[8] Katsumi Inoue and Cong-Kha Pham, "DataBase processor (DBP) which can search data ultra-high-speed: The Computing by Memory provides big innovation for information processing," The Institute of Electronics, Information and Communication Engineers, Technical report of IEICE. ICD 114 (13), pp. 91-96, Apr. 2014.

[9] Justin Zobel and Alistair Moffat "Inverted Files for Text Engines," ACM Computing Surveys, Vol. 38, No. 2, 2006.

[10] Christopher D. M., Prabhakar R., Hinrich S., "An Introduction to Information Retrieval," Cambridge University Press, Jul. 2008.