International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:2, 2009

Design of Multi-disease Diagnosis Processor using Hypernetworks Technique

Jae-Yeon Song, Seung-Yerl Lee, Kyu-Yeul Wang, Byung-Soo Kim, Sang-Seol Lee, Seong-Seob Shin, Jae-Young Choi and Chong Ho Lee, Jeahyun Park and Duck-Jin Chung

***Abstract*—**In this paper, we propose disease diagnosis hardware architecture by using Hypernetworks technique. It can be used to diagnose 3 different diseases (SPECT Heart, Leukemia, Prostate cancer). Generally, the disparate diseases require specified diagnosis hardware model for each disease. Using similarities of three diseases diagnosis processor, we design diagnosis processor that can diagnose three different diseases. Our proposed architecture that is combining three processors to one processor can reduce hardware size without decrease of the accuracy.

***Keywords*—**Diagnosis processor, Hypernetworks, Leukemia, Mask, Prostate cancer, SPECT Heart data

I. Introduction

THE Hypernetworks can be used diagnosis processor which can diagnose disease without specific information of disease [1][2]. But disease diagnosis process using Hypernetworks requires respectively designed HW because it spends huge amount operation for the classification. The disparate diseases require specified diagnose hardware. For the classification of the various diseases, lots of the HWs should be designed. This causes increase of the HW size. So we design the merged model by using similarities of the each architecture. It can diagnose 3 difference diseases. This paper is consisted by follows. In chapter 2, we introduce Hypernetworks model and we explain some constraints to high accuracy in chapter 3. In chapter 4, we describe how to design our hardware model. And we explain how to merge 3 difference diagnose processor. In finally, the advantage of proposed model was described in chapter 6.

II. Hypernetworks

The general method of disease diagnosis use characteristic information of disease. But this information requires huge expenses. But Hypernetworks model does not require this information. It uses only reference data set which is generated resulting from previous patient's diagnosis results. Basic mechanism is a using of already classified reference data. On new test pattern, it is compared with classified reference data and then the processor makes a decision by using comparison result. This procedure grants correctness with simple operation.

Jae-Yeon Song, Seung-Yerl Lee, Kyu-Yeul Wang, Byung-Soo Kim, Sang-Seol Lee, Seong-Seob Shin, Jae-Young Choi and Chong Ho Lee, Jeahyun Park and Duck-Jin Chung are with School of information and communication engineering, INHA University, Incheon, Korea. (phone: +82–32–874–1663; fax: +82–32–864-1664; e-mail: ojzjv@inhaian.net, e-mail: djchung@inha.ac.kr)

But it could not classify test data if there are no same data in reference data. Hypernetworks model can compensate that fault. It requires not identity but similarity at comparing test data with reference data. Namely, it compares not whole reference data but particle of reference data with test data. So it can have a wrong decision. To enhance the accuracy, we compare not just one time but many times. We define a comparison count as the Population Size. And The Order was defined as a particle size [1][2].

The Hypernetworks model uses concept of Hyperedge. Hyperedge is a particle of reference data. It includes indices of reference data sequence and the values of those positions. Generally, the count of indices and the selection of indices are randomly generated. But, using Hyperedge concept, it require huge memory size to store Hyperedge indices and values. And it requires the module to select indices. As a solution of that problem, we can use concept of Mask [3] [4].

With Mask concept, it doesn't store and find selected indices. It makes an n-bits sequence and set the selected indices. In other words, it can remove memory to store Hyperedge. And comparison can be replaced by simple bit operation. As a result, Comparison of one Hyperedge is simplified as an equation 1. The parameter 'n' depicts a data size of disease.

$$\begin{aligned} match_{i,j} &= \prod_{k=1}^{n} \{\overline{r_i(k) \otimes t_i(k)} + \overline{m_j(k)}\} \\ R_i &= \{r_i(1), r_i(2), ...r_i(n-1), r_i(n)\} \\ T_i &= \{t_i(1), t_i(2), ...t_i(n-1), t_i(n)\} \\ M_i &= \{m_i(1), m_i(2), ...m_i(n-1), m_i(n)\} \end{aligned} \quad (1)$$

The equation 1 can be translated as equation 2 by De Morgan' law.

$$match_{i,j} = \prod_{k=1}^{n} \{\overline{(r_i(k) \otimes t_i(k)) \bullet m_j(k)}\} \quad (2)$$

The 'R' is an already classified reference data and the 'T' is a test data which is required classification. And M is a Mask that is set the selected indices. Reference data not organized just one data.It organized with many data for each class. Hypernetworks model makes comparison as many as the number of population size per each reference data. So matching count of each class per one test data can be expressed equation 3.

$$MC_c = \sum_{i=1}^{d} \sum_{j=1}^{p} match_{i,j} \quad (3)$$

Where, the 'd' is a reference data count per each class and the 'p' is a population size. It compares matched counts of each class, and finally selects a class based on majority decision.

III. Constraints for High Accuracy

Hypernetworks model has some characteristics. It is formed of Hyperedge generating method. The number of indices and the composition of the index are selected randomly. Because of this, the accuracy of model is not constancy. In other words, the accuracy of the Hypernetworks model is fluctuated according to organized Hyperedge set. So, the first method to achieve better accuracy is contrived by selecting fixed indices size. In other words, to get the high accuracy, we can control the Order. In general, matching count is increased in case of small order comparison. But small order can not reflect characteristic of reference data. In other side, big Order mask can reflect characteristic of reference data. But matching count can be decreased. So, we can enhance accuracy of Hypernetworks model by finding optimal Order. To find optimal Order, we use empirical method. We make a Hypernetworks model with voluntary Order and then find optimal order by checking accuracy with changing order. The following Fig 1 shows the accuracy per order of each disease. We can find optimal order at highest accuracy point.

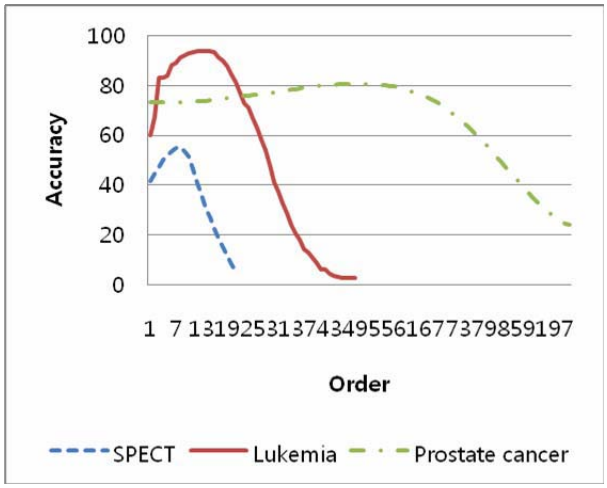

Fig. 1 Accuracy per Order

The second constraint to get enhanced accuracy is population size. In general, we can achieve acceptable accuracy with high population size. The hardware size and operation time are increased dramatically when population size are increased. So the contraction of the population size is required in the range of the reasonable accuracy reduction. We also use empirical method to find optimal population size. We increase population size in fixed order, and check accuracy per population size. In the Fig 2, the results depict accuracy per population size. Therefore, we define optimal population size as 100. The accuracy hardly increased at that point.

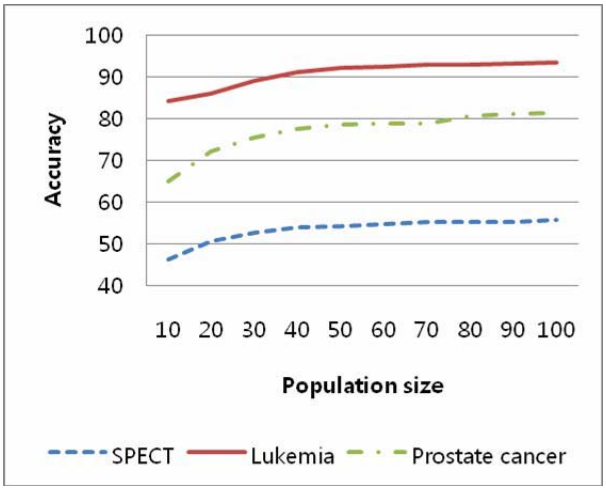

Fig. 2 Accuracy per population size

IV. Hardware Architecture

Hypernetworks based disease classifier spends lots of the execution time and computation for the diagnosis of the patient's disease pattern. So, there are some needs that it should be designed to application specific IC or HW because it can reduce the complexity of the Hypernetworks [3] [4].

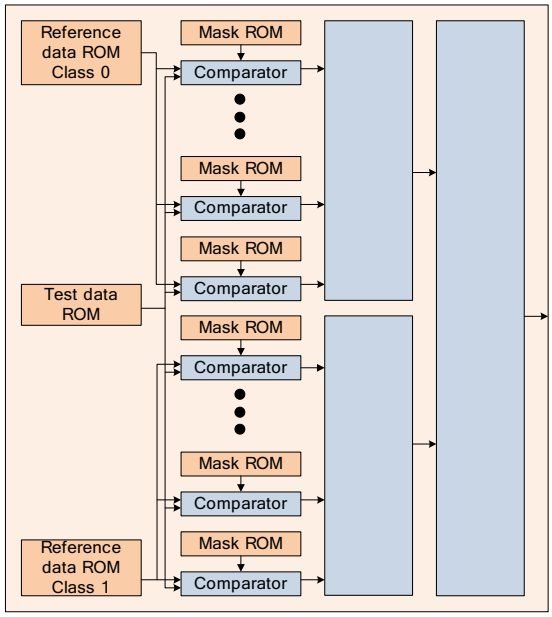

Fig. 3 DNAC processor architecture

The disease diagnosis processor consists of comparator for data comparison, accumulator for accumulation of the comparison results, decision block to decide the class of the patient's data by using the accumulated results, and memory

block to store the reference data and mask. The figure 3 depicts block diagram of the HW architecture to classify the class of the disease.

In the Fig. 4, the comparator consists of XOR and NAND gates responding to the equation 2. If test data and reference data are united or the mask doesn't enforced by '1', the output signal is enforced to true. In the other cases, the comparator generates false as an output. The n-input AND gate generates true if all outputs of each selective comparison are true.

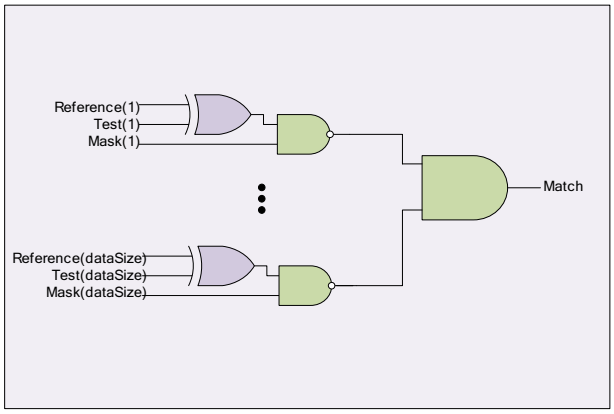

Fig. 4 Comparator Architecture

In the following, the Fig.5, the accumulator makes summation of the matching results which is generated by comparator. The accumulation is enabled while all reference data are treated.

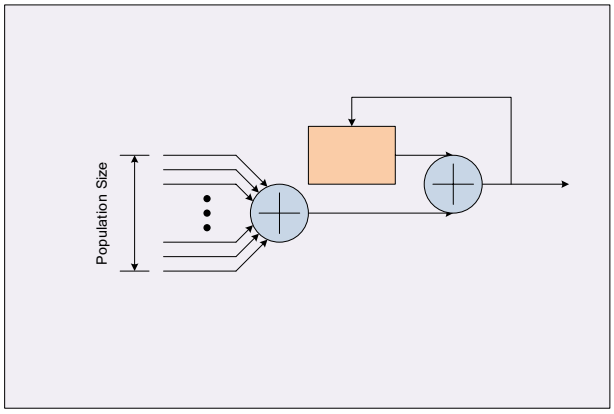

Fig. 5 Accumulator Architecture

The decision block compares accumulation results of the each class for the class decision of the test data. The decision is done resulting from majority of each accumulation results.

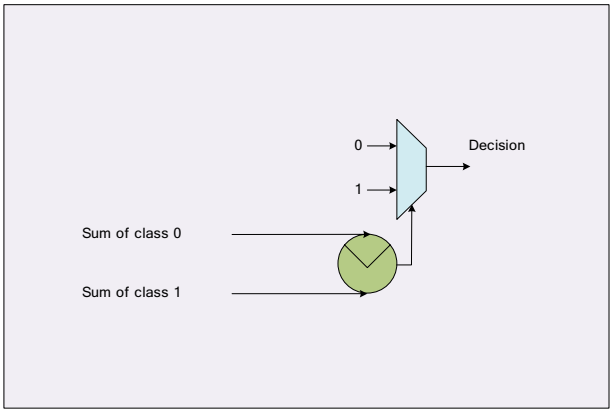

Fig. 6 Decision Block Architecture

V. Multi-Disease Diagnosis Hardware Architecture

In generally, each disease data has a different sequence length and the optimum Order. So, we have to design several processors to diagnose diseases. Almost modules are identical except the comparator size. So, they are shared. But a Mask, reference data, and test data memory are excluded because they contain specific diseases information.

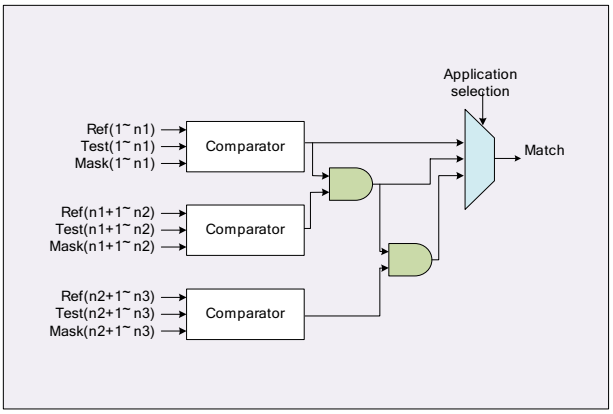

Fig. 7 New Comparator Architecture

In the fig. 7, we propose novel comparator that can be applied to three disease applications. It is composed of a couple of AND gates, Multiplexes, and conventional comparators. Each application A, B, and C's data sequence size are n1, n2, and n3 respectively (n1<n2<n3). In this case, match1, match2, and match3 are expressed in the equation 4.

$$match1_{i,j} = \prod_{k=1}^{n1} \{\overline{(r_i(k) \otimes t_i(k)) \bullet m_j(k)}\}$$

$$match2_{i,j} = \prod_{k=n1+1}^{n2} \{\overline{(r_i(k) \otimes t_i(k)) \bullet m_j(k)}\} \quad (4)$$

$$match3_{i,j} = \prod_{k=n2+1}^{n3} \{\overline{(r_i(k) \otimes t_i(k)) \bullet m_j(k)}\}$$

TABLE I APPLICATION PARAMETER AND DESIGN RESULT

	Each application model			
	SPECT Heart[5]	Leukemia[6]	Prostate cancer[7]	Merge model
Data sequence length	22	50	178	178
Reference data count	38(19, 19)	38(27, 11)	102(52, 50)	178(98, 80)
Test data count	180(7, 173)	35(21, 14)	34(25, 9)	249(53, 196)
Optimal order	8	12	48	Each order
Optimal population size	100	100	100	100
Occupied slices	2,722	5,522	18,526	23,348
FIFO16/RAMB16s	103	103	309	309
Max frequency[MHz]	84.382	84.382	84.382	84.382
Accuracy[%]	56	94.3	82.4	Each accuracy

The matching results of the each application are represented by following equation 5.

$$\begin{aligned} matchA_{i,j} &= match1_{i,j} \\ matchB_{i,j} &= match1_{i,j} \bullet match2_{i,j} \\ matchC_{i,j} &= match1_{i,j} \bullet match2_{i,j} \bullet match3_{i,j} \end{aligned} \quad (5)$$

In this architecture, we divided reference, test, and mask memory to fragments according to each application and the addresses are used to address the pattern. We designed the processor to be used in three kinds of applications by using architecture of the fig. 3.

VI. CONCLUSION

The processor designed by Verilog HDL and it was synthesized by Xilinx ISE 10.2i for virtex-4 xc4vlx200 FPGA device [8]. The table 1 shows each disease parameter and implementation results. The occupied slices mean used hardware resource. If processor was disparately designed, it requires 26,770 slices. But the proposed processor is designed with 23,348 slices. In other words, the proposed processor is designed with 13% decrease of the hardware resource except reduction of the accuracy. In the next study, we will study mask selection method to get high accuracy.

REFERENCES

[1] Zhang, B.-T. and Jang, H.-Y., “A Bayesian algorithm for in vitro molecula revolution of pattern classifiers”, Lecture Notes in Computer Science (DNA11),3384:458-467,2005.
[2] Byung-Tak Zhang and Joo-Kyung Kim, “DNA Hypernetworks for Information Storage and retrieval, ” Lecture Notes in Computer Science(DNA12), LNCS4287, pp.298 -307, 2006.
[3] Byung-Soo Kim, "Hardware Implementation of the Pattern Recognize Processor base on DNA Computing technique", ITC_CSCC vol. 1, pp.211-213, 2007.
[4] Kyu-Yeul Wang, "Implementation of Cardiovascular Disease (CVD) Level Prediction Processor using Aptamer Biochip" ITC_CSCC, vol. 2, pp.679-681, 2007.
[5] http://archive.ics.uci.edu/ml/datasets/SPECT+Heart
[6] http://www.broad.mit.edu/publications/broad939s
[7] http://www.cancercell.org/cgi/content/full/1/2/203
[8] Xilinx Corp., Virtex-4 User Guide, www.xilinx.com, Jan. 2007.