MATEC Web of Conferences **79**, 01014 (2016) DOI: 10.1051/matecconf/20167901014
IME&T 2016

BOOLEAN DIFFERENTIATION EQUATIONS APPLICABLE IN RECONFIGURABLE COMPUTATIONAL MEDIUM

Stanislav Shidlovskiy[1,2]

[1]National Research Tomsk State University, 634050, Tomsk, Russia
[2]National Research Tomsk Polytechnic University, 634050, Tomsk, Russia

Abstract. High performance computing environment synthesis with parallel architecture reconstructing throughout the process itself is described. Synthesized computational medium involving Boolean differential equation calculations so as to function in real-time image processing. Automaton imaging was illustrated involving the rearrangement of every processing medium element to calculate the partial differentials of n-th order in respect to Boolean function variables. The method of obtaining setting codes for each element was also described. An example in calculating 2nd -order Boolean derivative to two differentials in respect to Boolean functions, depending on three arguments within the reconstructible computational medium of 8x8 processing elements was given.

1 Introduction

Architecting updated robotic systems functioning autonomously in a-priori unknown medium involves directly positioned in- robot computer systems in order to process a considerable body of sematic information. Real-time processing of this information attributable to classical computing architecture results in increasing robot weight and structure design.

One possible solution of above-mentioned problem could be applying the reconfigurable computing medium (RCM) [1–6]. Hardware-in-the-loop RCM design and its functioning is based on grouped evaluator model:

$$S = < C, G, A(P(D)) > ,$$

where, $C = \{c_i\}$ – set of processing elements c_i, $i = 0, 1, \ldots, N - 1$; G – intercomputing linkage matrix structure (grid); A – processing element algorithm set C interrelated with G, in hardware parallel support program P of data processing D.

Grouped evaluator construct $H = < C, G >$ reflects the following architectural concept:

1) parallelism in processing (parallel processing element algorithm set C intercomputing with G);

2) programmability structure;

3) homogeneity of structure H (homogeneity of processing elements $c_i \in C$ and macrostructure G).

Let's consider the case when functioning processing element algorithm c_i is the basis {AND, OR, NOT}, thus acquiring the hardware invariance property.

Algorithm design in terms of solving one specific problem-oriented task via RCM, bearing on this limitation is associated with significant peformance. This paper describes the synthesis of such RCM for Boolean differentiation implementation and integration.

2 Problem statement

Usually, binary data is represented as {0, 1} and would look like arbitrary finite dimension matrix. By zero extension it will be representable as $2^n \times 2^n$ matrix $\mathbf{R}_{2^n}$, where n – positive integer number. This matrix $\mathbf{R}_{2^n}$ could be interpreted as microsequencing or code of Boolean function system, binary image (animated object frame), dynamic binary system state at the present sampling time, operating technical system component interaction graph. Further, this matrix will be considered as a system of 2^n Boolean functions (BF) $f_j(X)$, $j = \overline{0, 2^n - 1}$, n – variables x_1, x_2, …, x_n, Each BF has its own vector value $\mathbf{X}_j = \left[x_j^{(0)} \; x_j^{(1)} \; \mathrm{K} \; x_j^{(2^n-1)} \right]^{\mathrm{T}}$, where T – transposition symbol; element $x_j^{(t)} \in (0,1)$ – j-value of BF for set of variables $x_1^{t_1}$, $x_2^{t_2}$, …, $x_n^{t_n}$; t_1, t_2, …, t_n – binary representation parameter $t = \overline{0, 2^n - 1}$; $x^0 = \overline{x}$, $x^1 = x$. The system of 2^n BF $f_j(X)$, having its own vector value $\mathbf{X}_j$, can be written in the form of the matrix

$$\mathbf{R}_{2^n} = \left[\mathbf{X}_{2^n-1} \vdots \cdots \vdots \mathbf{X}_1 \vdots \mathbf{X}_0 \right].$$

In [7] the algorithm of Boolean derivative calculations by private variable is:

$$\frac{\partial^{(k)} \mathbf{X}_j}{\partial x_1 \partial x_2 \ldots \partial x_k} = \mathbf{D}_{2^n}^{(k)} \ldots \mathbf{D}_{2^n}^{(2)} \mathbf{D}_{2^n}^{(1)} \mathbf{X}_j \pmod 2, \tag{1}$$

$$\mathbf{D}_{2^n}^{(i)} = \mathbf{I}_{2^{i-1}} \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \mathbf{I}_{2^{n-i}}. \tag{2}$$

where **I** – identity matrix.

The analysis of the algorithm shows that RCM processing element should implement one of the following automaton-based display image (Figure 1).

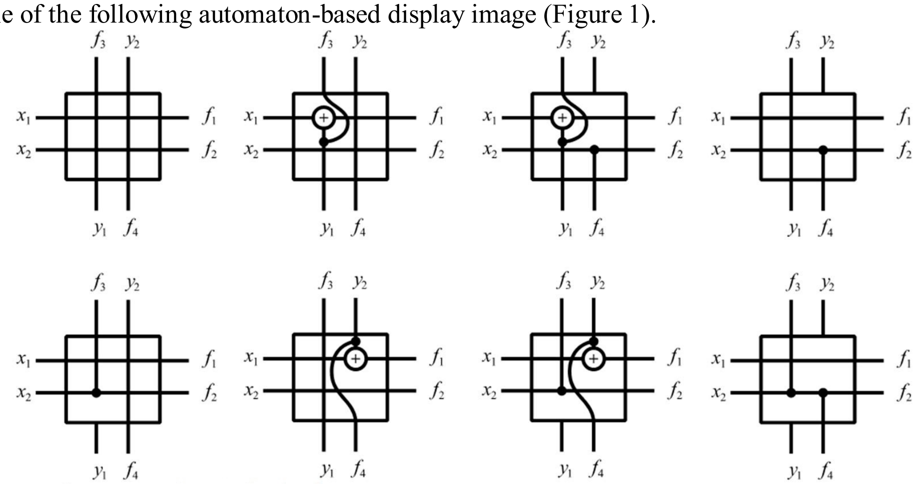

Figure 1. Automaton-based display images.

Applying automaton-structured method [5] RCM functioning processing element algorithm could be synthesized:

$$\begin{cases} f_1 = \left(x_1\bar{y}_1\bar{z}_2\right) + \left(x_1\bar{y}_2 z_2\right) + \left(x_1 z_3\right) + \left(\bar{x}_1 y_1 \bar{z}_2\bar{z}_3\right) + \left(\bar{x}_1 y_2 z_2 \bar{z}_3\right), \\ f_2 = x_2, \\ f_3 = \left(x_2 z_1 z_2 \bar{z}_3\right) + \left(x_2 z_1 \bar{z}_2 z_3\right) + \left(x_2 \bar{z}_1 z_2 z_3\right) + \left(y_1 z_1 z_2 z_3\right) + \left(y_1 \bar{z}_1 \bar{z}_2\right) + \left(y_1 \bar{z}_1 \bar{z}_3\right) + \left(y_2 z_1 \bar{z}_2 \bar{z}_3\right), \\ f_4 = \left(x_2 z_1 \bar{z}_2 \bar{z}_3\right) + \left(x_2 \bar{z}_1 z_3\right) + \left(y_2 z_1 z_2\right) + \left(y_2 z_1 z_3\right) + \left(y_2 \bar{z}_1 \bar{z}_3\right). \end{cases}$$

where, x_1, x_2, y_1, y_2 – data inputs; $f_1,\ldots, f_4$ – outputs; $z_1,\ldots, z_3$ – setup inputs of RCM processing element [3, 5].

Realization of algorithm (1) into RCM via evaluators, implementing automaton images (fig.1) are plotted as the matrix structure of computing-based system. Determining the matrix Z configuration is based on the setting up each processing element within one possible automaton image:

$$\mathbf{Z} = \sum_i \mathbf{M}_i \tag{3}$$

$$\mathbf{M}_1 = (\mathbf{A} \circ \mathbf{I}_n)k_1,$$

$$\mathbf{M}_2 = (\mathbf{A} \circ \overline{\mathbf{I}}_n \circ \mathbf{H}_n)k_2,$$

$$\mathbf{M}_3 = (\mathbf{A} \circ (\overline{\mathbf{I}}_n \circ \mathbf{H}_n)^{\mathrm{T}})k_3,$$

$$\mathbf{M}_4 = \overline{\mathbf{A}}\, k_4,$$

$$\mathbf{A} = \mathbf{D}_{2^n}^{(k)} \ldots \mathbf{D}_{2^n}^{(2)} \mathbf{D}_{2^n}^{(1)} \pmod 2 \tag{4}$$

where, o – Hadamard product; $\mathbf{I}_n$ – n x n identity matrix; $\mathbf{H}_n$ –n x n upper triangle matrix of one; vertical overline (¯) means x-element inversion in binary matrix; $^{\mathrm{T}}$ – transposition matrix symbol; k_i – automaton image code.

3 Example

It is required to calculate Boolean second-order derivative with respect to the BF variables x_1 and x_2, as in the following expression

$$f_i(\mathbf{X}) = x_1 x_2 \vee x_3. \tag{5}$$

To implement Boolean derivative calculation by private variable four basic evaluator elements are necessary (fig. 1). Each element is denoted by k_i (i = 1, 2,…, 4). Value k_i corresponds to each automaton code of its configuration, introduced as decimal numeration (as follows k_1 = 2; k_2 = 3, k_3 = 4, k_4 = 5). We apply in (5) all possible argument values:

$$\mathbf{X}_j = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}^{\mathrm{T}}.$$

According to (3) and (4) the following can be obtained:

$$\mathbf{A} = \begin{bmatrix} 1 & & 1 & & & & & \\ & 1 & & 1 & & & & \\ 1 & & 1 & & & & & \\ & 1 & & 1 & & & & \\ & & & & 1 & & 1 & \\ & & & & & 1 & & 1 \\ & & & & 1 & & 1 & \\ & & & & & 1 & & 1 \end{bmatrix} \oplus \begin{bmatrix} 1 & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & 1 & & & & 1 & \\ & & & 1 & & & & 1 \\ 1 & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & 1 & & & & 1 & \\ & & & 1 & & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & 1 & & 1 & & 1 & \\ & 1 & & 1 & & 1 & & 1 \\ 1 & & 1 & & 1 & & 1 & \\ & 1 & & 1 & & 1 & & 1 \\ 1 & & 1 & & 1 & & 1 & \\ & 1 & & 1 & & 1 & & 1 \\ 1 & & 1 & & 1 & & 1 & \\ & 1 & & 1 & & 1 & & 1 \end{bmatrix}.$$

$$\mathbf{Z}=\begin{bmatrix} 2 & 5 & 3 & 5 & 3 & 5 & 3 & 5 \\ 5 & 2 & 5 & 3 & 5 & 3 & 5 & 3 \\ 4 & 5 & 2 & 5 & 3 & 5 & 3 & 5 \\ 5 & 4 & 5 & 2 & 5 & 3 & 5 & 3 \\ 4 & 5 & 4 & 5 & 2 & 5 & 3 & 5 \\ 5 & 4 & 5 & 4 & 5 & 2 & 5 & 3 \\ 4 & 5 & 4 & 5 & 4 & 5 & 2 & 5 \\ 5 & 4 & 5 & 4 & 5 & 4 & 5 & 2 \end{bmatrix}.$$

Obtained matrix of Z configuration corresponds to RCM, depicted in Figure 2.

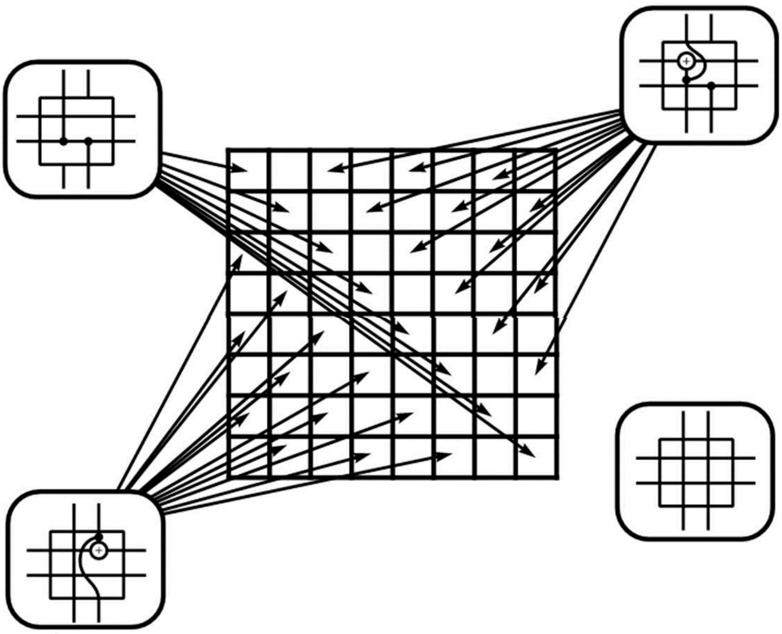

Figure 2. RCM configuration.

Having generated the signals corresponding to (6) the result providing the BF calculation by private variables (5) x_1 and x_2 is recorded on data inputs of obtained RCM and **on medium return:**

$$\frac{\partial^2 \mathbf{X}_j}{\partial x_1 \partial x_2} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}.$$

4 Conclusion

RCM algorithms were synthesized to provide the calculation of BF private variables. These algorithms meet the specified requirements applicable for VLSI process. In this case, such algorithms involve regular and localized computation communication, oriented on pipeline computations. Proposed algorithm configuration (3), involving specific setting code inputs for each RCM processing element, is responsible of calculating Boolean derivatives to different argumants within one and the same RCM.

According to the reference description, the synthesis method of specific Boolean function evaluators is based on the operating diagram of the previously selected Boolean base (AND, OR, NOT) and is housed in synthesized-based medium. It should be noted that the prescribed functional diagram or logical network within selected medium is 'fitted'. Obtained algorithms of operating RCM (reconstructible computational medium) could be the basis of developing special-purpose evaluators producing basic operations of morphologically processing digital images.

Acknowledgment

The following research has been financially supported by Russian Foundation for Basic Research (RFBR) 16-07-01138 A “Intelligent Reconfigurable Control Systems, Navigation and Image Processing for Autonomous Mobile Robots.”

References

[1] I. Kulagin, A. Paznikov, M. Kurnosov, Lecture Notes in Computer Science **9251**, 405 (2015) doi: 10.1007/978-3-319-21909-7_39
[2] N. Maa, Sh. Wang, Ali S. Mohsin, X. Cui, Yu. Peng, MATEC Web of Conferences **45**, 05001 (2016) doi: 10.1051/matecconf/20164505001
[3] S.V . Shidlovskii, J. Comp. Sys. Sci. Inter. **45**, 282 (2006) doi: 10.1134/S1064230706020122.
[4] A.I. Kalyaev, I.A. Kalyaev, J. Comp. Sys. Sci. Inter. **55**, 211 (2016) doi: 10.1134/S1064230716010081
[5] I.A. Kalyaev, I.I. Levin, A.I. Dordopulo, L.M. Slasten, IFAC Proceedings Volumes **46**, 210 (2013) doi: 10.3182/20130925-3-CZ-3023.00009
[6] D.V. Shashev, S.V. Shidlovskiy, Optoelectronics, Instrumentation and Data Processing **51**, 227 (2015) doi: 10.3103/S8756699015030036
[7] S. Janushkevich, D. Bokhmann, R. Stankovich, G. Tosic, V. Shmerko, Avtomatika I Telemeknanika (Journal of Automation & Remote Control) **6**, 155 (2004).