International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 7009-7031

http://www.ripublication.com

Design Exploration For Systolic Architecture Using Hybridization Of GA And Local Chaotic Optimization

Selva Kumar R[1], DharmishtanK.Varughese[2], Manoj Kumar Singh[3]

[1,2] Department of ECE, Karpagam College Of Engineering, Coimbatore, India,ykmselva@gmail.com
[3]Manuro Tech Research Pvt. Ltd. ,Bangalore,India, mksingh@manuroresearch.com

Abstract

Advanced processing applications have generated requirement of an efficient, scalable and localized computing architecture. This paper has proposed an automated method to estimate and explore the systolic array design vectors using help of hybridization ofgenetic algorithm and local chaotic optimization (HGACH).This hybridization can give the benefit in terms of optimal solution and faster global convergence. Number of different systolic architecture which have the characteristics of partial pipeline or fully pipelined can be easily developed with proposed solution. Characteristics of chaotic map non periodicity and initial condition sensitivity deliver the possibility of more robust search especially in local region. Hence after applying the cross-over and mutation operation in genetic algorithm local region is further searched by chaotic optimization.Lozi chaotic map has applied to define the chaotic pattern under chaotic optimization. Benefit of proposed method has also verified with numerical benchmark problem and it is observed that proposed method has delivered the better performances.Systolic array for matrix multiplication is designed because of its usefulness in different field of science and engineering.

Keywords: Systolic array, Dependence graph, Genetic algorithm, Chaos, Chaotic optimization

1. Introduction

The vital goal of evolving new computer architectures and efficient use of existing modern systems is to run greater and more complex applications faster over time. The continued demand for increased computing power led to the development of high parallel scalable multiprocessing systems. Parallelism is an instinctive and appealing

concept. There are basically two ways to improve the computer performance in terms of computational speed. One way is to use faster devices (VLSI chips). Although faster and faster VLSI components have contributed a great deal on the improvement of computation speed, the breakthroughs in increasing switching speed and circuit densities of VLSI devices will be difficult and costly in future. The other way is to use parallel processing architectures, which employ multiple processors to perform a computation task. When multiple processors working together, an appropriate architecture is very important to achieve the maximum performance in a cost-effective manner. Systolic arrays are ideally qualified for computationally intensive applications with inherent massive parallelism because they capitalize on regular, modular, rhythmic, synchronous, concurrent processes that require intensive, repetitive computation. Systolic architectures are considered to be a very suitable means of implementing parallel algorithms in several areas of science and engineering. Systolic arrays combine pipelining and multiprocessing techniques, and are composed of a number of processing elements (PE), connected together by a regular and local interconnection. Each PE is performing – usually the same and very simple - computations on its input data and local memory variables, it stores data and is communicating with the neighboring PE-s. The functioning of the PE-s is synchronized by a global clock that results in a regular data flow through the system. The systolic model suits very well for construction of high performance special purpose computer devices. Until recently, computation-intensive tasks were handled by high performance supercomputers, including pipelined computers, array processor and multiprocessor systems. The development of these computer systems has involved a through exploration of parallel computing, efficient programming, and resource optimization. However, the general-purpose nature of these machines has led to complicated system organization and severe system overheads. These machines are not very efficient from electronics point of view for real time signal processing where a very high throughput rate is absolutely essential. A solution to the real-time requirement of signal processing is to use special-purpose array processor and minimize the processing concurrency by either pipeline processing or parallel processing or both. As long as communication in VLSI remains restrictive, locally-interconnected array will be of great importance. an increase of efficiency can be expected if work load while observing the requirement of locality, i.e. short communication paths these properties of the designer of VLSI algorithm, and eventually lead to new designs of architecture and language .

2. Related work

Kyrkou&Theocharides[1] present systolic chain of processing elements towards the realization of a generic systolic array for support vector machine (SVM) object classification in embedded image and video applications. Focus has given to provides efficient memory management, reduced complexity, and efficient data transfer mechanisms. The proposed architecture is generic and scalable, as the size of the chain, and the kernel module can be changed in a plug and play approach without affecting the overall system architecture. These advantages provide versatility,

scalability and reduced complexity that make it ideal for embedded applications. Quantum-dot Cellular Automata (QCA) technology is a promising alternative to CMOS technology. It is attractive due to its fast speed, small area and low power consumption. To explore the characteristics of QCA technology, digital circuit design approaches have been investigated. Due to the inherent wire delay in this technology, QCA appears to be suitable for pipelined architectures particularly. Systolic arrays take advantage of pipelining and parallelism. Therefore, an investigation into systolic array design in QCA technology is provided by Liang Lu and others in [2]. A systolic-array processor is proposed byMadanayake [3] for implementing 2D IIR frequency-planar-beam wave digital filters (WDFs) to achieve directional enhancement of propagating broadband plane-waves based on their directions of arrival. The architecture is based on a passive LC ladder prototype network and has a high-throughput of one-frame-per-clock-cycle.Multi valued logic synthesis is a very promising and affluent research area at present because of allowing designers to build much more efficient computers than the existing classical ones. Ternary logic synthesis research has got impetus in the recent years. Based on the design of a reversible systolic array, which is one of the best examples of parallel processing, using micro level ternary Toffoligate proposed by Nower, N [4]. Castillo Atoche in [5], propose a hardware/software (HW/SW) co-design approach of high-resolution reconstruction of remote sensing (RS) imagery using systolic arrays as coprocessors. The software design is aimed at the algorithmic-level decrease of the computational load of the large-scale SAR image enhancement tasks. The algorithmic idea is based on the concept of descriptive regularization (DR) approach. Cellular Automata is one of the ways of performing computations which necessitates extremely the processing of data at high speeds. Implementing cellular automata on serial bases does not provide the required speed. Yarahmadi, A [6] proposed implementation of cellular automata on systolic base has been studied and the speed of performance in this method has been compared with that of the serial method.Ong, K.S.H [7] presented a scalable design for accelerating the problem of solving a dense linear system of equations using LU Decomposition.N. UshaBhanu [8] has proposed, an approach to design a 2-D systolic array for high-speed implementation of block-based lifting lossy 9/7 wavelet filter. A systolic architecture is proposed byHajarAsgari [9] for implementing of digital Hopfield neural networks for solving shortest path problem.

3. Systolic Architecture: Design Methodology and challenges

Systolic Architectureconsists of number of regular and local interconnection based strategy to provide the computational efficiency with help of pipelining. Number of important features which is very attractive for VLSI design and processing point of view inherently available with systolic architecture as shown in Fig.1 and are being able to use each input data item a number of times to deliver high throughput with modest requirement of bandwidth. Other advantages include modular expansibility, simple and regular data and control flow use of simple and uniform cells [13].

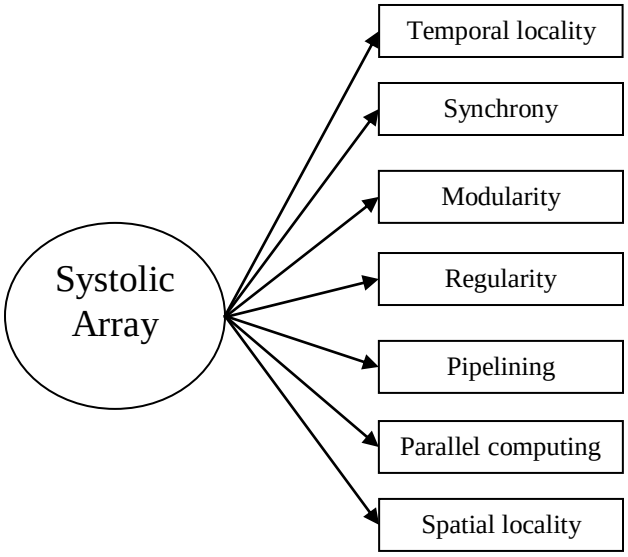

Fig.1 Systolic Array features

In general, systolic designs apply to any compute bound problem that is regular that is, one where repetitive computations are performed on a large set of data like In the field of Signal and Image processing: FIR, IIR filtering, and 1-D convolution,2-D convolution and correlation, Discrete Fourier Transform,Interpolation,1-D and 2-D median Filtering,Artificial Neural networks,Robotics.In field of linear algebra Matrix-Vector multiplication,Matrix-Matrix multiplication,Matrix triangularization (solution of linear systems, matrix inversion),QR decomposition(eigenvalue, least-square computations) and in the field of Non-numeric applications like Data structures: Stack and Queue, searching, priority queue, and sorting,Graph algorithms-Transitive closure, minimum spanning trees, and connected components,Language recognition-string matching and regular expression,Dynamic programming,Encoders (polynomial division) and Relational data base operations. Challenges associated with the problem are:

- Synthesize systolic architecture characteristics and qualities are completely depends upon mapping vectors and manual exploration to achieve the optimal vectors are very difficult.
- With minimal information from reduced dependence graph how to define the mathematical model for automated environment.
- Processing units should have the maximum processor utilization efficiency which depends on the pipelining period, the block pipeline period and data input scheme.
- Solution environment should have capability to generate multiplearchitecture for same algorithm so that domain environment of architecture could adopt in best manner otherwise a compromising situation will appear.
- Design of algorithmmethod to satisfy the objectives and analyze their performance characteristics.

A systolic system consists of a set of interconnected cells each capable of performing some simple operation. As simple, regular communication and controlled structures have substantial advantages over complicated ones in design and implementation; cells in a systolic system are typically interconnected to form a systolic array or systolic tree. Information in systolic systems flows between cells in a pipelined fashion and communication with the outside world occurs only at the "boundary cells". Systolic array architectures are designed by using linear mapping techniques on regular dependence graphs (DG). The DG corresponds to space representation, where no time instance is assigned to any computation (t=0). A DG is said to be regular, if a presence of an edge in a certain direction at any node in the DG represents presence of an edge in the same direction at all nodes in the DG. Systolic architectures have space-time representation where each node is mapped to a certain PE and is scheduled at a particular time instance. Systolic design methodology maps an N-dimensional DG to a lower dimensional systolic architecture. Mapping of N-dimensional DG to (N-1) dimensional systolic array is considered. Fundamental vectors involved in the systolic array design are :(i).Projection vector (D) (also called as iteration vector): Two nodes that are displaced by'D 'or multiples of 'D 'are executed by same processor. (ii)Processor space vector (P): Any node with index I would be executed by processor in space time representation by: P^TI. (iii)Scheduling Vector (S): Any node with index I would be executed at: S^TI. Performance quality measure of architecture design can be defined as hardware utilization efficiency (HUE), Define as $1/|S^TD|$, This is because two tasks executed by the same processor are spaced $|S^TD|$time units apart. Much systolic architecture can be designed for a given problem by selecting different projection, processor space and scheduling vectors. These vectors must satisfy the feasibility constraints. The feasibility constraints are:(i)If points A and B differ by the projection vector,i.e. (I_A-I_B) is same as D, then they must be executed by the same processor.(ii)If A and B are mapped to the same processor, then they cannot be executed at the same time. In other words: $P^TD=0$, $S^TD\neq 0$; Once the above two constraints are satisfied, edge mapping is done. If an edge E exists in a space representation or DG then an edge P^Te is introduced in a systolic array with $S^T e$ delays.

4. Problem formulation

Feasibility, Quality and characteristics of obtained systolic architecture completely depend upon the three (SDP) fundamental design vectors. It is possible to formulate the process of obtaining the good SDP vectors as problem of constraint optimization where objectives are to maximize HUE and minimize the total delay associated with each edge of processing element. Both objectives can be combine as a single objective in form of minimization as given below along with various constraints (by taking 1/HUE instead of HUE).

$$f = \min\left\{\left|S^T D\right| + \sum_{i=1}^{n} d_i\right\}$$

Constraints are

$$\begin{cases} S^T e_i \geq T_i, \forall i = 1,2,..n; \\ P^T D = 0; \\ S^T D \neq 0; \end{cases}$$

Where S, D and P represent the scheduling, Projection and Projection vector, whereas e_i and T_i represents the edge vector and required time to start the execution of destination nodes of corresponding edge. Delay associated with each edge in SA represented as *d* whereas *n* represents the total number of edges available in reduced dependence graphs (RDG). Constraint (i) is a set of scheduling inequalities, which represent the dependency of nodes in DG in terms of execution time of nodes and constraint (ii) appeared to represents the orthogonal relation between P&D, and constraint (iii) defines Non-orthogonal relation between S&D.There is no unique solution hence it gives a chance to find multiple SDP vectors as solutions and in result different types of systolic architecture.

4.1 Proposed solution

A new optimization method has developed by hybridizing genetic algorithm and local chaotic optimization (HGACH) as shown in Fig.2 to obtain the optimal solution in faster and global convergence. Characteristics of chaotic map non periodicity and initial condition sensitivity deliver the possibility of more robust search especially in local region.Hence after applying the cross-over and mutation operation in genetic algorithm local region is further searched by chaotic optimization.Lozi chaotic map has applied to define the chaotic pattern under chaotic optimization. Benefit of proposed method has also verified with numerical benchmark problem and it is observed that proposed method has delivered the better performances.

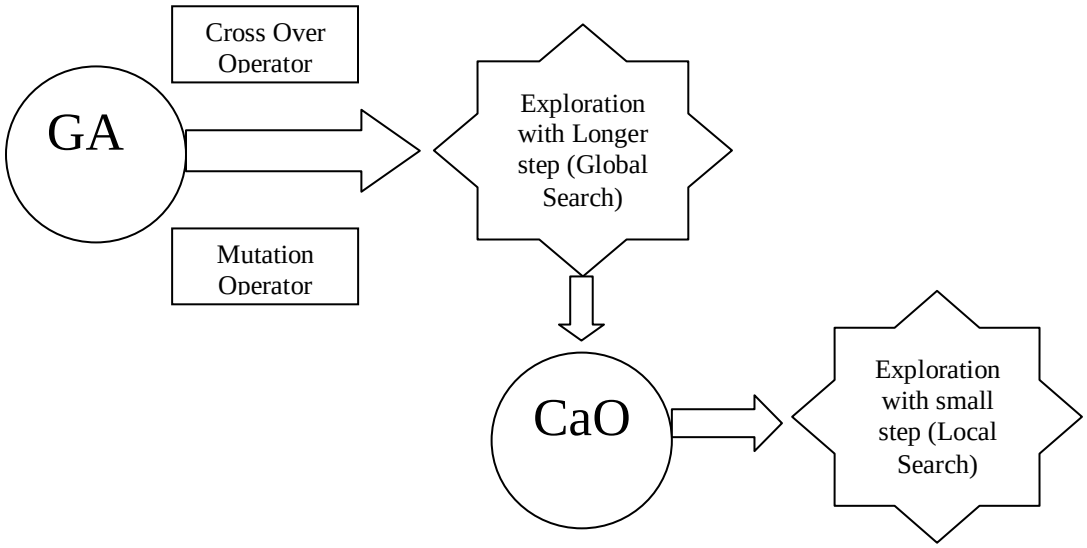

Fig.2Hybridization of GA and Chaotic optimization

4.2 Computational Intelligence

Natural computing is the computational version of the process of extracting ideas from nature to develop computational systems, or using natural materials (e.g., molecules) to perform computation [14]. Broadly It can be divided into three main branches :(1) Computational inspiration taken from nature: here nature is taking taken as processing system and observable process becomes inspiration to develop the similar mathematical model to solve the real life problem. In result there are number of algorithms which mimic the natural computation process.(2) The simulation and emulation of nature by means of computing: this support to synthesize the process in our practical life to create patterns, forms, behaviors, and organisms (3) Natural material as computing plateform: instead of silicon based artificial computing device natural materail itself taken as place where defined operation performed in controlled manner.

Therefore, natural computing can be defined as the field of interdeciplinary research that involved different branch of science like, physicists, chemists, engineers, biologists, computer scientists, among others, all have to act together in order to make natural computing to make use in the practical environment.

Evolutionary computing, also called evolutionary computation, is the field of research that draws ideas from evolutionary biology in order to develop search and optimization techniques for solving complex problems. Most evolutionary algorithms are rooted on evolutionary biology, which basically states that a population of individuals capable of reproducing and subjected to genetic variation followed by selection results in new populations of individuals increasingly fitter to their environment. The computational abstraction of these procedures resulted in the so-called evolutionary algorithms.The basic idea of EC has been to make use of the powerful process of natural evolution as a problem-solving paradigm, usually by simulating it on a computer.A standard evolutionary algorithm can thus be proposed as follows:

- A population of individuals that reproduce with inheritance. Each individual represents or encodes a point in a search space of potential solutions to a problem. These individuals are allowed to reproduce, generating offspring that inherit some traits from their parents. These inherited traits cause the offspring to present resemblance with their progenitors.
- Genetic variation. Offspring are prone to genetic variation through mutation, which alters their genetic makeup.Mutation allows the appearance of new traits in the offspring and, thus, the exploration of new regions of the search space.
- Natural selection. The evaluation of individuals in their environment results in a measure of adaptability, or fitness value to be assigned to them. A comparison of individual finesses will lead to a competition for survival and reproduction in the environment, and there will be a selective advantage for those individuals with higher fitness.

  Design of Genetic algorithm with local search operator is shown in Fig. 3.

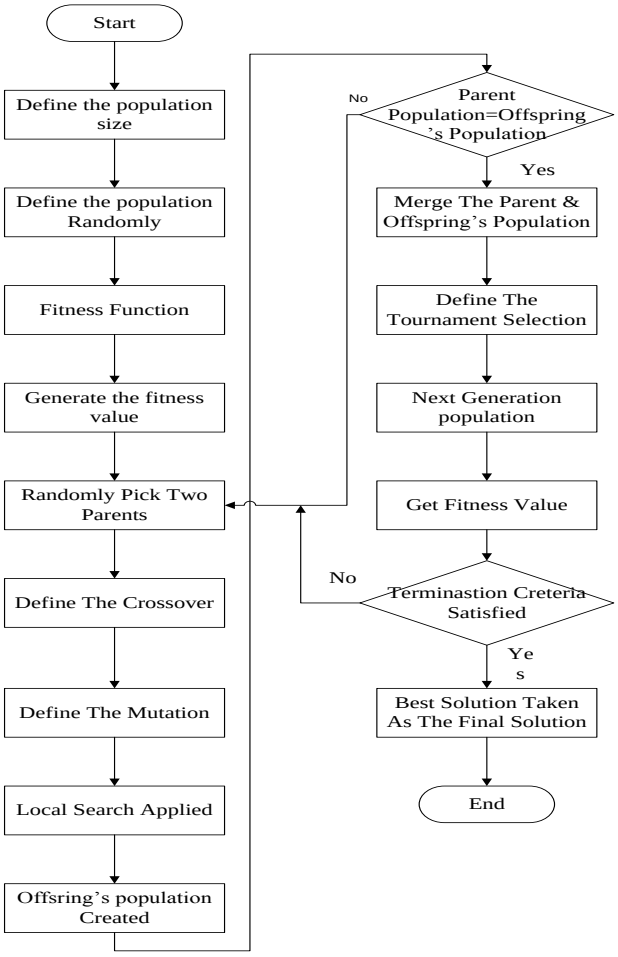

Figure 3: Flow Representation of loacel operator based GA

5. Local search using Chaotic optimization

Chaos theory containsnonlinear mathematical modeling and analysis to describe the behavior of unstable and aperiodic dynamic system or process .There are number of important characteristics available in chaos like (i) chaotic system is dynamic in nature,it means it has time varying characteristics.(ii)system behavior is not periodic and unstable. (iii) Chaotic behavior is complex but also it can have simple causes through deterministic process. (iv)it shows sensitive to initial conditions; it means that for two different initial condition same process deliver atistically uncorrelated outputs. (v) Because the system is described through deterministic process, chaotic behavior can not counted as random phenomena. On the other hand, because of the instability, aperiodicity and sensitivity to initial conditions, the behavior of chaotic systems is not predictable even though it is deterministic.(vi)there is a feedback mechanism. Systems may display both chaotic and non-chaotic behavior depending on the control parameters used [15].

5.1 Lozi Map

In many science and engineering applications chaos can be applied. An essential feature of chaotic systems is towards its initial condition sensitivity.This sensitive dependence on initial conditions is generally exhibited by systems containing multiple elements with nonlinear interactions, particularly when the system is forced and dissipative. Sensitive dependence on initial conditions can also be obtain by simple methematical equation and generated chaotic pattern can be utilise to search the solution space in faster manner because of the non-repetition in compare to probabilistic stochastic ergodic based searches .The design of approaches to improve the convergence of chaotic optimization is a challenging issue. Lozi map is a very efficient method to generate the powerful chaotic pattern.The Lozi's piecewise liner model [16] is a simplification of the He´non map and it admits strange attractors. This chaotic map involves also non-differentiable functions which difficult the modeling of the associate time series. The Lozi map can be define by (1) and (2).

$$S(+1) = 1 - P.|S(t)| + y(t) \quad (1)$$

$$y(t) = Q.S(t) \quad (2)$$

Where *'t'* is the iteration number. In this work, the values of y are normalized in the range [0,1] to each decision variable in n-dimensional space of optimization problem. This transformation is given by (3)

$$Z(t) = \frac{y(t) - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \quad (3)$$

Wherey ϵ[-0.6418, 0.6716] and [λ_{max}, λ_{min}] equals to [0.6418, -0.6716].The parameters used in this work are P=1.7and Q=0.5, these values show the sensitivity with respect to initial condition as suggested in [17]. We have done experiments for two different setting of initial condition and results of generated chaotic sequences are shown in Fig.4.

Chaotic series are generated for two different setting of parameters which are having very slight change in initial condition to show the variations in generated output. In the Fig.4 Data1 is generated with initial value of parameters S=0.1999, y=0.2Whereas data2 is generated with initial value of parameter S=0.2, y=0.2. Even there is very little difference in initial condition the generated, variation between two chaotic sequences are shown in Fig.4 for 50 samples. With the observation of Fig.4 it is very clear there is a very significant difference between these two chaotic sequences. This will make really a very difficult environment for attacker to guess the parameters of initialization, in result more robust solution.

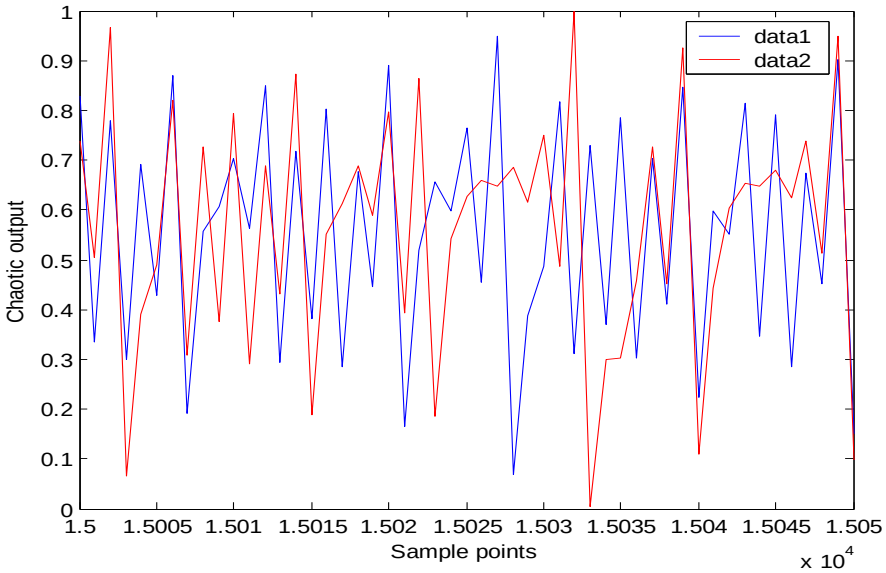

Fig.4 Sensitivity of chaotic sequence with two different initial conditions in Lozi map

5.2 Chaotic Optimization

During the chaotic local search; the step size λ is an important parameter in convergence behavior of optimization method, which adjusts small erotic ranges around current solution.

Local search algorithm

Initialize λ and β

For (m2=1,……..M2) {

If r <0.5 {

(Where r ϵ *uniformly generated random number in range [0 ,1])*

Let $\boldsymbol{X}$c = $\boldsymbol{X}$n+ $\lambda\beta$| U- $\boldsymbol{X}$n|

else

Let $\boldsymbol{X}$c = $\boldsymbol{X}$n- $\lambda\beta$| $\boldsymbol{X}$n-L| }

Let Δf=($\boldsymbol{X}$c) – f($\boldsymbol{X}$n)

If (Δf<0) *then* ($\boldsymbol{X}$n=$\boldsymbol{X}$c)

β = Z(β) }

The step size λ is employed to control the impact of the current best solution on the generating of a new trial solution. A small λ tends to perform exploitation to refine results by local search, while a large one tends to facilitate a global exploration.

6. Experimental analysis

A Numeric optimization

Proposed methods for all the three cases have applied with GA and HGACH in various test bench numerical optimizations to evaluate the performances in search of global solution with faster convergence.Two different benchmark test functions as shown in Eq.4 and in Eq.5 have taken as a problem of minimization to define the experiments with dimension size equal to 2. F1 and F2 are both multimodal problems and contain number of local minima.Maximum numbers of allowed iterations in both algorithms are equal to50. In case of HGACH, associated local chaotic optimization has given 50 iterations for each member of the offspring population. Size of population in all cases has taken as 20.In all test cases 10 independent trials have given to analyze the performance in terms of best achieved, worst possible seen, mean and standard deviation. Probability of mutation has taken as 0.1.

With respect to Goldstein price function, obtained performances have shown in Table 1. It can observe from Table 1 that HGACH has delivered the most appreciable result and very close to global solution whereasGA in all trails could not converge properly.Convergence characteristics of both the methods have shown in Fig. 5 and in Fig.6 for all the 10 independent trails.Each different color represents the convergence characteristics of independent trail. It can observe very clearly that HGACH has the fastest rate of convergence in comparison to GA. In second test Six-hump camel back function in which within the bounded region there are six local minima located performances have shown in Table2. A comparative performance analysis has given in Table 3.Convergence characteristics for both algorithms for all ten independent trails have shown in Fig.7 and Fig.8.HGACH has obtained the global solution in all trails whereas GA fail to obtain in any one.it can be concluded that fastest convergence in all trails have observed with HGACH.

(i) The Goldstein-Price function

```
F1(x1,x2) =[1+(x1+x2+1)^2·(19-
14·x1+3·x1^214·x2+6·x1·x2+3·x2^2)]
[30+(2·x1-·x2)^2·(18-32·x1+12·x1^2+48·x2-
36·x1·x2+27·x2^2)]; (4)
          -2 <= x1 <= 2, -2 <= x2 <= 2.
```

Global minimum $f(x_1,x_2)=3$; $(x_1,x_2)=(0,-1)$.

(ii) Six-hump camel back function

$$F2(x,y) = (4 - 2.1 x^2 + x^4/3) x^2 + xy + (- 4 + 4y^2) y^2; \quad (5)$$

$$-3 <= x <= 3, -2 <= y <= 2.$$

Global minimum $f(x, y) = - 1.0316$; $(x, y) = (-0.0898,0.7126), (0.0898,-0.7126)$.

TABLE1: PERFORMANCES FOR GOLDSTEIN PRICE TEST FUNCTION .

	GA			HGACH		
Trail No.	P1	P2	Function value	P1	P2	Function value
1	0	0.9404	4.5402	0.0016	0.9994	3.0006
2	0	0.9492	4.1078	0	1.0000	3.0000
3	0	0.9384	4.6489	0.0003	1.0004	3.0001
4	-0.2165	-1.2240	52.3808	0.0006	1.0010	3.0004
5	0	-1.0395	3.7445	-0.0003	-1.0011	3.0005
6	-0.0822	-0.8351	16.6464	-0.0024	-1.0001	3.0015
7	-0.5898	-0.4223	30.2877	0.0009	-1.0003	3.0003
8	-0.3636	-0.6807	36.1564	-0.0001	-0.9991	3.0004
9	-0.1004	-0.9656	6.7336	0.0003	-1.0000	3.0000
10	-0.0553	-0.7878	28.4166	0	-0.9996	3.0001

TABLE2: PERFORMANCES FOR SIX HUMP CAMEL BACK TEST FUNCTION .

	GA			HGACH		
Trail No.	P1	P2	Function value	P1	P2	Function value
1	0.0654	-0.6804	-1.0219	0.0882	-0.7127	-1.0316
2	-0.1233	0.6734	-1.0140	-0.0889	0.7134	-1.0316
3	0	0.7223	-0.9981	0.0905	-0.7139	-1.0316
4	0	0.4562	-0.6593	0.0900	-0.7129	-1.0316
5	0	0.6596	-0.9831	- 0.0906	0.7127	-1.0316
6	0	-0.5952	-0.9150	0.0901	-0.7136	-1.0316
7	0	-0.8475	-0.8093	0.0897	-0.7118	-1.0316
8	0	0.6274	-0.9547	- 0.0889	0.7135	-1.0316
9	0	0.6978	-0.9993	- 0.0899	0.7133	-1.0316
10	-0.0843	0.7898	0.9769	0.0882	-0.7127	-1.0316

TABLE3:COMPARATIVE PERFORMANCES FOR F1 AND F2 TEST FUNCTION

	$f1$		$f2$	
Perf	GA	HGACH	GA	HGACH
Best	3.7445	3.0000	-1.0219	-1.0316
Worst	52.3808	3.0015	-0.6593	-1.0316
Mean	18.7663	3.0004	-0.9332	-1.0316
Std. Dev	17.1584	0.0004	0.1148	0.0000

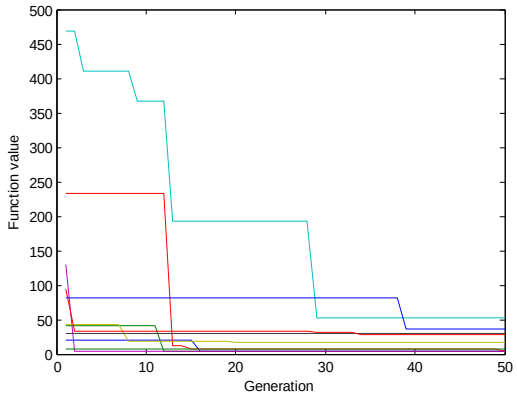

Fig.5Convergence characteristics of GA with respect to Goldenstein price function in 10 independent trails.(Each color represent the convergence behavior of one trail)

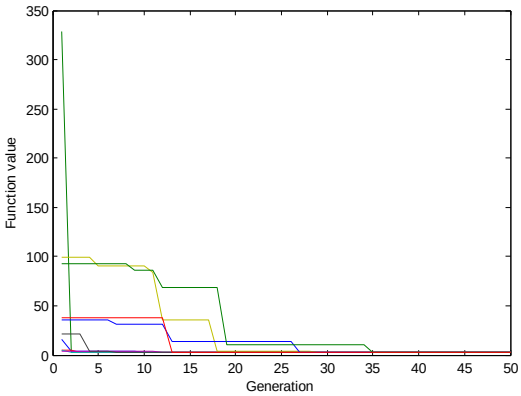

Fig.6Convergence characteristics of HGACH with respect to Goldenstein price function

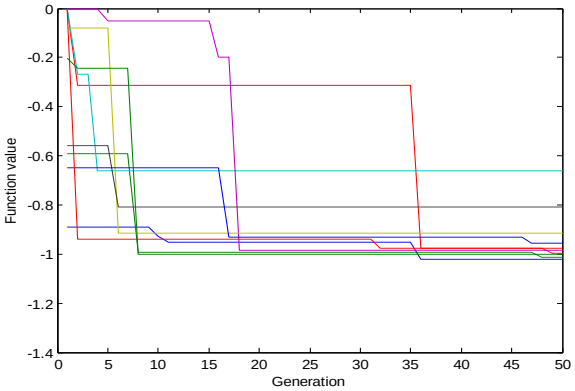

Fig.7 Convergence characteristics of GA with respect to Six hump camel back function

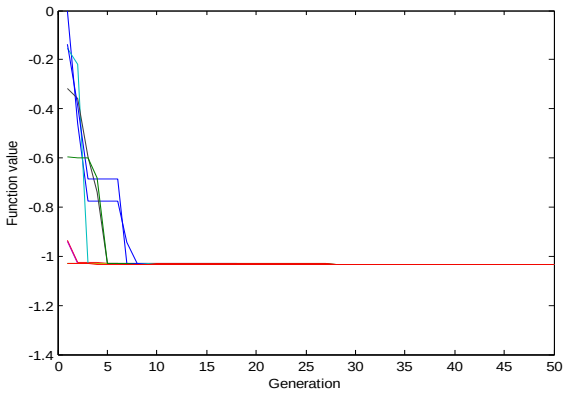

Fig.8Convergence characteristics of HGACH with respect to Six hump camel back function

A Systolic architecture design

In the sphere of information processing matrix computations are amongst the most frequently required and computationally intensive tasks. Their computation forms the basis of many important applications, such as digital signal, image and video processing, numerical analysis, computer graphics and vision, etc. The nature of matrix multiplication algorithms is such that they are perfectly suited to parallel exploitation. From the space diagram as given in Fig.9, we can write the iteration in standard output RIA form as follows.

$$\left.\begin{aligned} a(i, j, k) &= a\ (i, j-1, k) \\ b(i, j, k) &= b\ (i-1, j, k) \\ c(i, j, k) &= c\ (i, j, k-1) + a(i, j, k)\ b(i, j, k). \end{aligned}\right\} \qquad (6)$$

And the corresponding RDG is shown in Fig.9.Required input parameters for vector generation are RIA edges matrix and time taken in dependent node execution T_x.Edge matrix and the depended node execution time in RDG of Fig.9 as shown below, by assuming that consumed time for multiplication –addition equal to 1 u.t and communication time equal to 0 u.t.

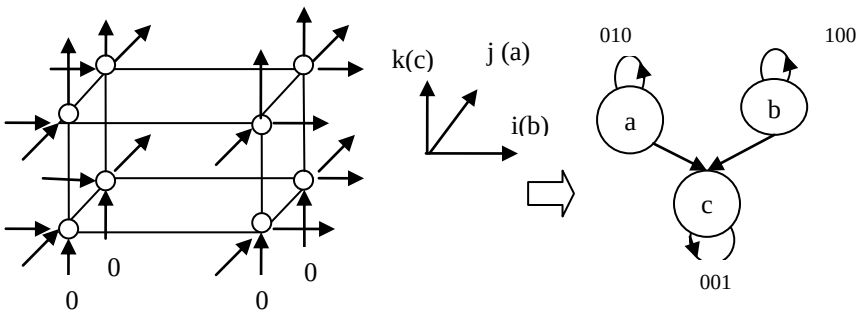

Fig.9 Reduced Dependent Graph for matrix multiplication algorithm

Design 1:

TABLE4: DESIGN VECTORS AND CORRESPONDING HUE GENERATED BY HGACH FOR DESIGN 1 .

1Projection Vector	Processor Vector	Schedule Vector	HUE(%)
[0 1 1]	[0 -1 1 ; 1 0 0]	[1 0 1]	1

TABLE5: EDGE MAPPING IN SYSTOLIC ARRAY FOR DESIGN 1

1RIA Edege	Mapped Edged in SA	Delay with Mapped Edge in SA
a(0, 1, 0)	(-1,0)	0
b(1, 0, 0)	(0,1)	1
c(0, 0, 1)	(1,0)	1

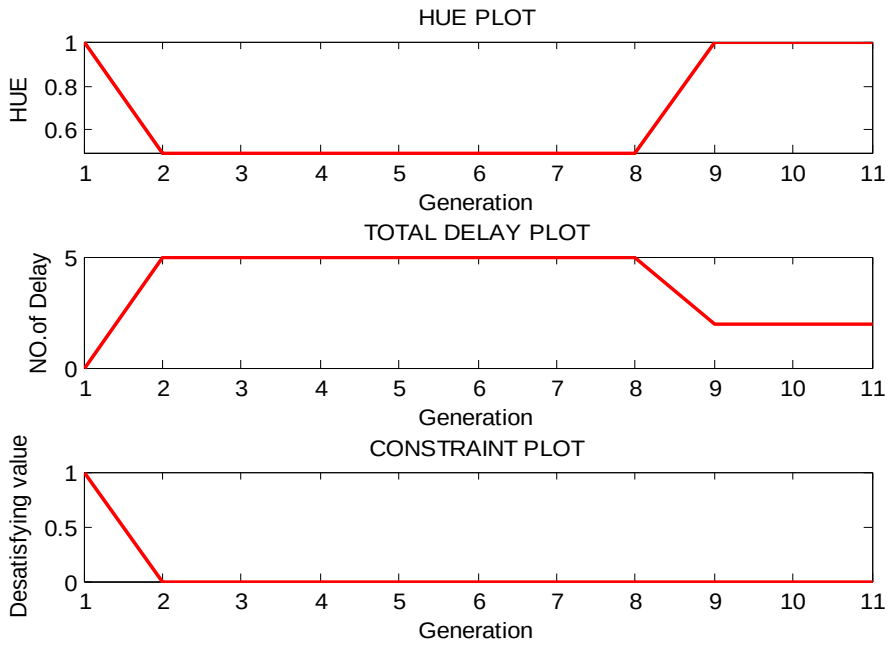

Fig.10 HGACH Performance plot with generation for design 1

Various design vectors have obtained by HGACH as shown in Table 4 and corresponding edge mapping has shown in Table 5.Various performance plot for design 1 are also shown in Fig.10. The corresponding 2D systolic architecture of this design is shown in Fig .11.

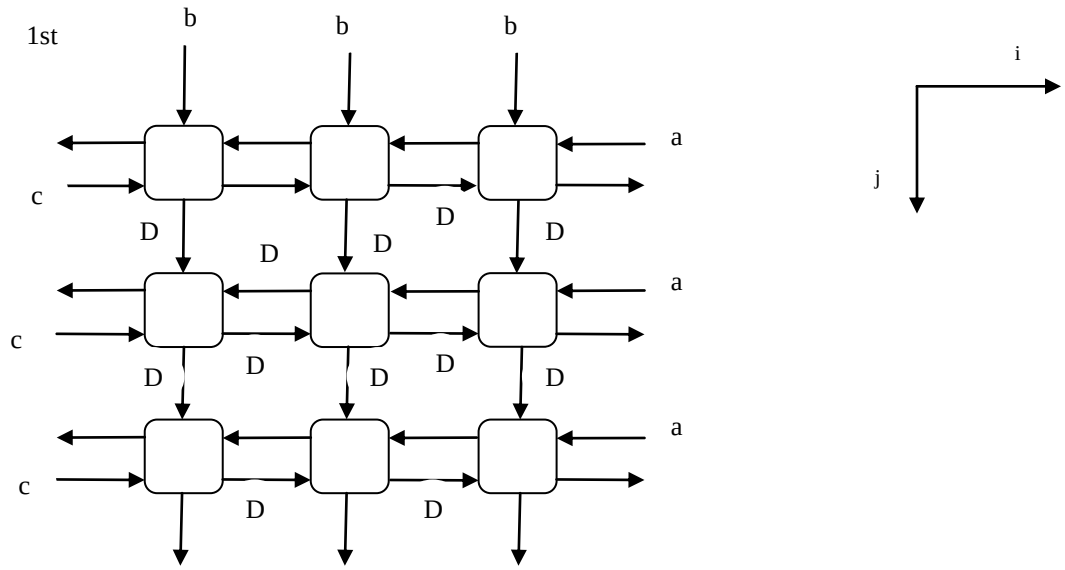

Fig.11.Systolic architecture for design 1

Design2:

TABLE6: DESIGN VECTORS AND CORRESPONDING HUE GENERATED BY HGACH FOR DESIGN 2

2Projection Vector	Processor Vector	Schedule Vector	HUE(%)
[0 1 0]	[-1 0 0 ; 0 0 -1]	[0 1 1]	1

TABLE7: EDGE MAPPING IN SYSTOLIC ARRAY FOR DESIGN 2

Edege	Mapped Edged in SA	Delay with Mapped Edge in SA
a(0, 1, 0)	(0,0)	1
b(1, 0, 0)	(-1,0)	0
c(0, 0, 1)	(0,-1)	1

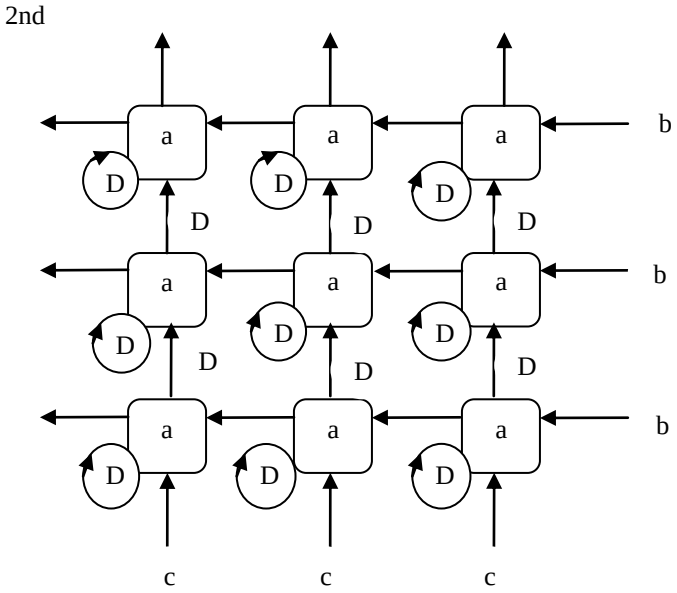

Fig.12.Systolic architecture for design 2

Design 3

TABLE8: DESIGN VECTORS AND CORRESPONDING HUE GENERATED BY HGACH FOR DESIGN 3

3Projection Vector	Processor Vector	Schedule Vector	HUE(%)
[0 1 0]	[0 0 1 ;-1 0 1]	[0 1 1]	1

TABLE9: EDGE MAPPING IN SYSTOLIC ARRAY FOR DESIGN 3

3RIA Edege	Mapped Edged in SA	Delay with Mapped Edge in SA
a(0, 1, 0)	(0,0)	1
b(1, 0, 0)	(0,-1)	0
c(0, 0, 1)	(1,1)	1

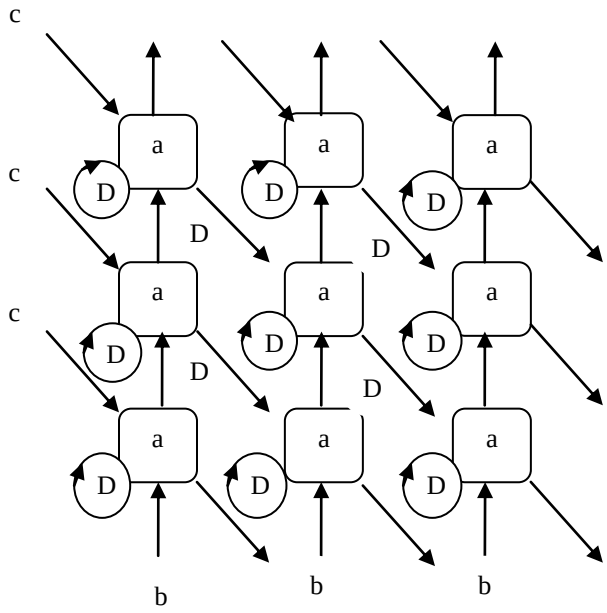

Fig.13.Systolic architecture for design 3

Design 4

TABLE10: DESIGN VECTORS AND CORRESPONDING HUE GENERATED BY HGACH FOR DESIGN 4

4Projection Vector	Processor Vector	Schedule Vector	HUE(%)
[1 0 0]	[0 1 - 1 ; 0 1 1]	[1 0 1]	1

TABLE11: EDGE MAPPING IN SYSTOLIC ARRAY FOR DESIGN 4

4RIA Edege	Mapped Edged in SA	Delay with Mapped Edge in SA
a(0, 1, 0)	(1,1)	0
b(1, 0, 0)	(0,0)	1
c(0, 0, 1)	(-1,1)	1

Design5

TABLE12: DESIGN VECTORS AND CORRESPONDING HUE GENERATED BY HGACH FOR DESIGN 5

5Projection Vector	Processor Vector	Schedule Vector	HUE(%)
[1 -1 0]	[-1 -1 0 ; 0 0 -1]	[1 0 1]	1

TABLE13: EDGE MAPPING IN SYSTOLIC ARRAY FOR DESIGN 5

5RIA Edege	Mapped Edged in SA	Delay with Mapped Edge in SA
a(0, 1, 0)	(-1,0)	0
b(1, 0, 0)	(-1,0)	1
c(0, 0, 1)	(0,-1)	1

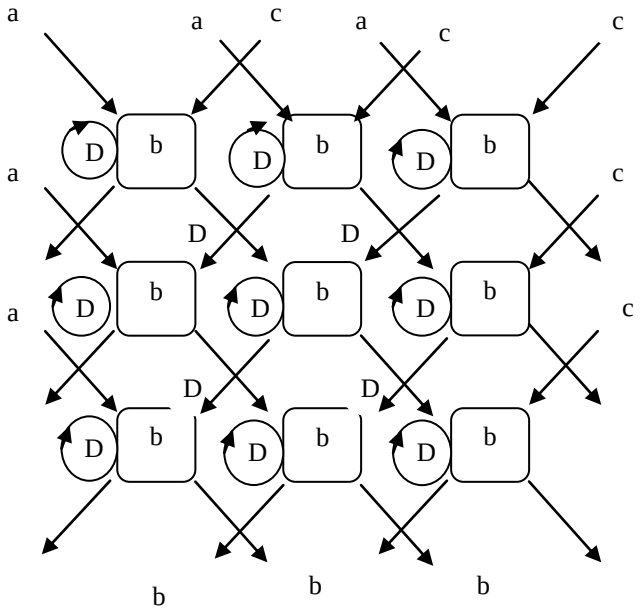

Fig.13.Systolic architecture for design 4

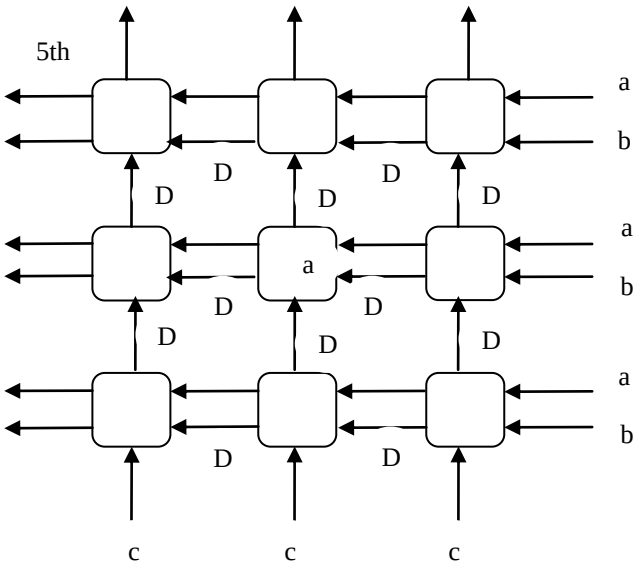

Fig.14.Systolic architecture for design 5

There is also possible to design fully pipelined systolic architecture in which all edges have at least one delay unit, with placing this as a constraint in solution development. We have developed number of systolic architecture which contains fully pipelined facility and two of them havedefined below in Design 6 and in Design 7.

Design 6

TABLE13: DESIGN VECTORS AND CORRESPONDING HUE GENERATED BY HGACH FOR DESIGN 6

Fp1Projection Vector	Processor Vector	Schedule Vector	HUE(%)
[0 0 1]	[0 -1 0 ; 1 0 0]	[1 1 1]	1

TABLE14: EDGE MAPPING IN SYSTOLIC ARRAY FOR DESIGN 6

Fp1RIA Edege	Mapped Edged in SA	Delay with Mapped Edge in SA
a(0, 1, 0)	(-1,0)	1
b(1, 0, 0)	(0,1)	1
c(0, 0, 1)	(0,0)	1

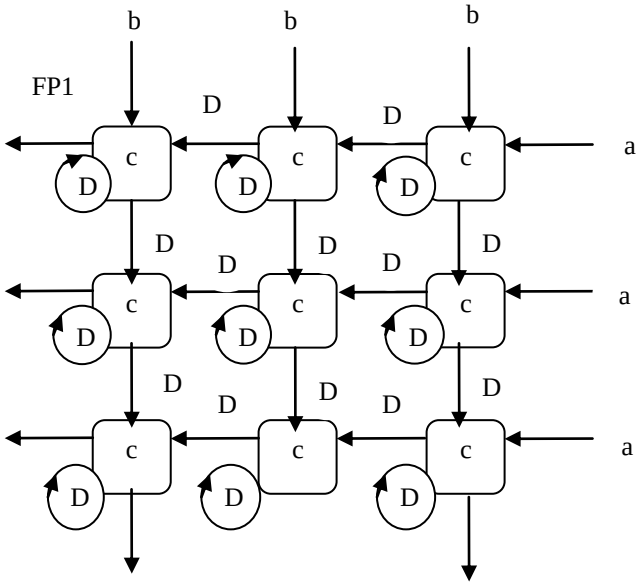

Fig.15.Systolic architecture for design 6

Design 7

TABLE14: DESIGN VECTORS AND CORRESPONDING HUE GENERATED BY HGACH FOR DESIGN 7

Fp2Projection Vector	Processor Vector	Schedule Vector	HUE(%)
[-1 0 0]	[0 1 0;0 0 1]	[1 1 1]	1

TABLE15: EDGE MAPPING IN SYSTOLIC ARRAY FOR DESIGN 8

Fp2RIA Edege	Mapped Edged in SA	Delay with Mapped Edge in SA
a(0, 1, 0)	(1,0)	1
b(1, 0, 0)	(0,0)	1
c(0, 0, 1)	(0,1)	1

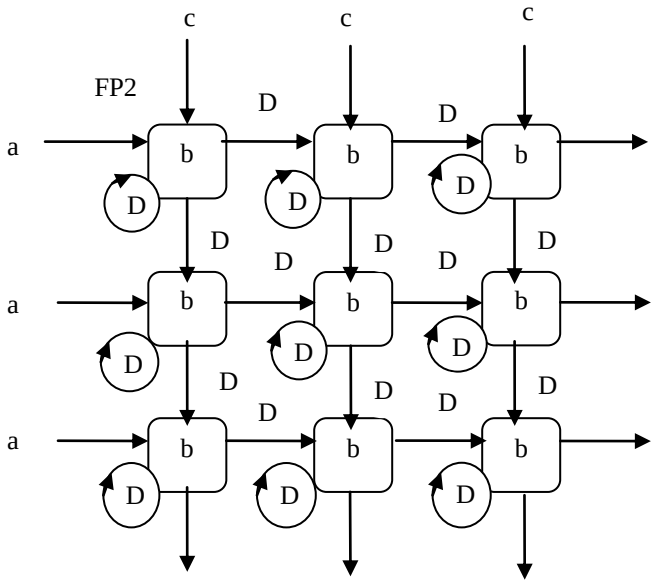

Fig.15.Systolic architecture for design 7

Conclusion

Natural computing systems have enormous capabilities but mathematical model to represents natural phenomenon are equally difficult. A major challenge associated with this is, designing a algorithm to deliver the optimal solution so that architecture efficiency could takes places in proper way. In this regard a hybrid approach between genetic algorithm and chaotic optimization has developed in this paper. As in case of natural system integrity of modules is much more robust concept to develop the optimal solution. Rather than having one solution, a set of solutions has developed. In comparison to single solution which may impose the constraint with surrounding, the

proposed solution gives better performance to satisfy the surrounding as well desired objectives. Parametric quality evaluation of systolic architecture is a challenging task and mathematical model for hardware utilizing efficiency has been applied and associated number of delays also minimize to reduce cost without compromising the quality.Various architecture for matrix multiplication under two different possibilities of systolic architecture called partial pipelined and fully pipelined have developed. It is possible to apply the same approach for other algorithm to design number of different and efficient systolic array.

References

[1] Kyrkou, C. Theocharides, T. ," SCoPE: Towards a Systolic Array for SVM Object detection",EmbeddedSystemsLetters, IEEE , Aug. 2009 ,Volume: 1 Issue: 2 ,page(s): 46 - 49 .

[2] Liang Lu, Weiqiang Liu, Máire O'Neill, Earl E. Swartzlander Jr., "QCA Systolic Matrix Multiplier," isvlsi, pp.149-154, 2010 IEEE Annual Symposium on VLSI, 2010.

[3] Madanayake, H.L.P.A. Bruton, L.T. ,"Systolic array processors for 2D IIR spatio-temporal beamforming wave-digital filters (WDFs) ",Communications, Computers and Signal Processing, 2009. PacRim 2009. IEEE Pacific Rim Conference on , page(s): 47 - 52 .

[4] Nower, N.; Chowdhury, A.R ,"Realization of systolic array using ternary reversible gates", Computers and Information Technology, 2009. ICCIT '09. 12th International Conference,page(s):192 – 196.

[5] Castillo Atoche, A.; Aguilar, J.O.; Castillo, J.V.; "Systolic array implementations for real time enhancement of remote sensing imaging", Programmable Logic, 2009.SPL. 5th Southern Conference on ,Issue Date: 1-3 April 2009 On page(s): 59 - 64 .

[6] Yarahmadi, A.; Setayeshi, S.; "An Implementation of Cellular Automata on Systolic Array",Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on ,On page(s): 330 – 333.

[7] Ong, K.S.H.,"A scalable and compact systolic architecture for linear solvers",Application-specific Systems, Architectures and Processors (ASAP), 2014 IEEE 25th International Conference on , 2014 ,Page(s):186 - 187 .

[8] N. UshaBhanu, A. Chilambuchelvan ," High-Speed Systolic VLSI Architecture for 2-D Forward Lifting-Based DWT", Arabian Journal for Science and Engineering ",August 2014, Volume 39, Issue 8, pp 6125-6135.

[9] HajarAsgari, Yousef S. Kavian, Ali Mahani," A systolic architecture for Hopfield neural networks", Conference on Electronics, Telecommunications and Computers – CETC 2013.

[10] Evans D.J. and Wan C.R., :Massive Parallel Processing for Matrix Multiplications: A Systolic Approach, Highly Parallel Computations:

Algorithms and Applications (Bekakos M.P. ed.), WITpress, Southampton, Boston, 2001.

[11] Milovanović Igor Ž. MilovanovićEmina I., RanđelovićBranislav M., Jovanović Ivan Č.:Matrix multiplication on bidirectional linear systolic arrays ,Filomat 2003, br. 17, str. 135-141

[12] Yanushkevich, S.N.: Spatial Systolic Arrays Design for Predictable Nanotechnologies, Journal of Computational and Theoretical Nanoscience, Volume 4, Number 3, May 2007 , pp. 467-481(15).

[13] H. T. Kung,"why systolic architecture",

[14] Leandro Nunes de Castro," Fundamentals of natural computing: an overview", Elsevier,Physics of Life Reviews 4 (2007) 1–36,

[15] Vicente Valle, Jr.," Chaos, Complexity and Deterrence",2000.

[16] Lozi R. Un attracteur etrange? du type attracteur de He´non. J Phys 1978;39(C5):9–10.

[17] Caponetto R, Fortuna L, Fazzino S, Xibilia MG." Chaotic sequences to improve the performance of evolutionary algorithms".IEEE Trans Evolution Comput 2003;7(3):289–304.