Simulation Result Analysis of Efficient Design of Systolic Architecture

Naveen M P
IT dept., M.tech
DSCE
Bangalore, India
Naveenmp47@gmail.com

Ebenezer V
IT dept., Associate professor
DSCE
Bangalore, India
Vebenezer@gmail.com

***Abstract*---In This Project an attempt has been made to develop a new concept for finding the optimal values for the entire three fundamental design vectors namely: scheduling, projection and processor so that not only architecture design could be feasible along with that maximum hardware utilizing efficiency could be achieved. Project also having the focus to minimize the total delay involved with systolic architecture design. Evolutionary programming has applied to find the optimal solution.**

Presented work and result will provide facility to designer without any involvement to find out best suited architecture for a particular application. Proposed method having capability to find the large number of optimal vectors for any algorithm which can be implemented in systolic architecture.

Keywords- Systolic arrays, Evolutionary Computation, linear mapping technique, Scheduling vector and systolic array design using RDG.

I.INTRODUCTION

The essential goal of developing new computer architectures and efficient use of existing modern systems is to run larger and more complicated applications faster over time. The continued demand for increased computing power led in the late 1980's to the development of high parallel scalable multiprocessing systems. Parallelism is an intuitive and appealing concept.

Parallel computing is a form of computation which many calculations are carried out simultaneously, operating on the principle that large problems can often

be divided into smaller ones, which are then solved concurrently ("in parallel").

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 Breakthrough 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.

VLSI (Very Large Scale Integration) is low-cost, high-density, high-speed technology. This technology is especially suitable for designs which are regular, repeatable, and with high localized communications. So we need a high-performance, special-purpose computer system to meet all these specific application. “A systolic array is a design style for VLSI.”

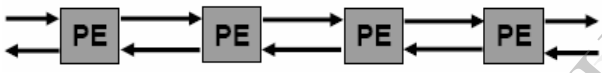
Very large scale integrated (VLSI) circuit technology makes it clear that simple and regular interconnections lead to inexpensive implementations and high densities, and high densities enable both high performance and low overhead. For these reasons, we are interested in designing parallel algorithms that have simple and regular data flows. We are also interested in using pipelining as a general method for implementing these algorithms in hardware. By pipelining, processing may proceed concurrently with input and output, and, consequently, overall execution time is minimized. Pipelining plus multiprocessing at each stage of a pipeline should lead to the best possible performance. architectures, an architectural concept originally proposed for VLSI implementation of some matrix operations.

A systolic array is an arrangement of processors in an array where data flows synchronously across the array between neighbors, usually with different data flowing

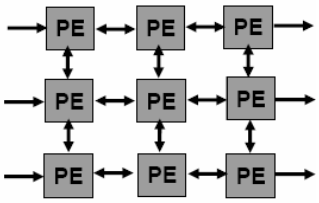
in different directions. Each processor at each steptakendata from one or more neighbors processes it and, in the next step, outputs results in the opposite direction.

H. T. Kung and Charles Leiserson were the first to publish a paper on systolic arrays in 1978, and coined the name. Systolic array is a specialized form of parallel computing with multiple processors connected by short wires. Each unit is an independent processor. Cells (processors) compute data and store it independently of each other. Every processor has some registers and an ALU. The cells share information with their neighbors, after performing the needed operations on the data. Systolic system is easy to implement because of its regularity and easy to reconfigure. This architecture can result in cost-effective, high-performance special-purpose systems for a wide range of problems.

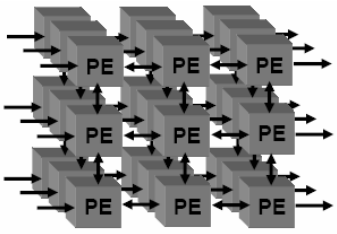
Some simple examples of systolic array models are shown in the fig.1



1D array



2D array



3D array

Fig.1 systolic array models.A set of simple processing elements with regular and local connections, which takes external, inputs and processes them in a predetermined manner in a pipelined fashion as shown in the fig.2.

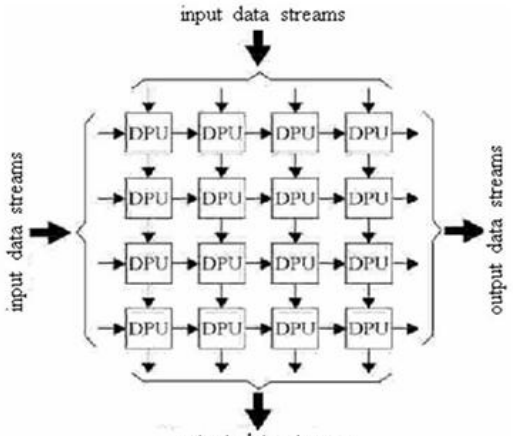


Fig.2 systolic array general model

A systolic array is composed of matrix-like rows of data processing units called cells (DPU). Each cell shares the information with its neighbors immediately after processing. The systolic array is often rectangular where data flows across the array between neighbor DPUs, often with different data flowing in different directions.

II.SYSTOLIC ARCHITECTURES

Systolic architectures, an architectural concept originally proposed for VLSI implementation of some matrix operations.

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”.

By replacing the single processing element with an array of PE’s as shown infig.4, a higher computation throughput can be achieved without increasing memory bandwidth. Being able to use each input data item a number of times (and thus achieving high computation throughput with only modest memory bandwidth) is one of the many advantages of systolic approach.

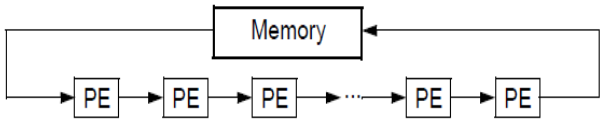


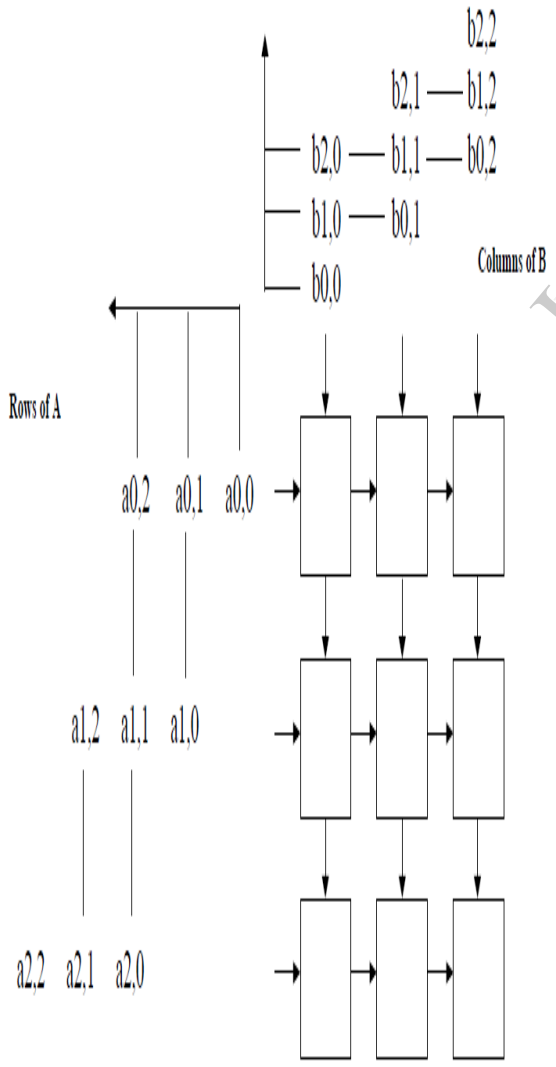
Fig.3 Basic Principle of Systolic Array

Typically all the PE's in the systolic array are uniform and fully pipelined, i.e. all communicating edges among the PE's contain delay element, and whole system usually contain only local interconnections.

An example of a systolic algorithm might be designed for matrix multiplication. One matrix is fed in a row at a time from the top of the array and is passed down the array; the other matrix is fed in a column at a time from the left hand side of the array and passes from left to right. Dummy values are then passed in until each processor has seen one whole row and one whole column. At this point, the result of the multiplication is stored in the array and can now be output a row or a column at a time, flowing down or across the array.

3x3 Systolic Array Matrix Multiplications

Processors arranged in a 2-D grid
Each processor accumulates one element of the product.



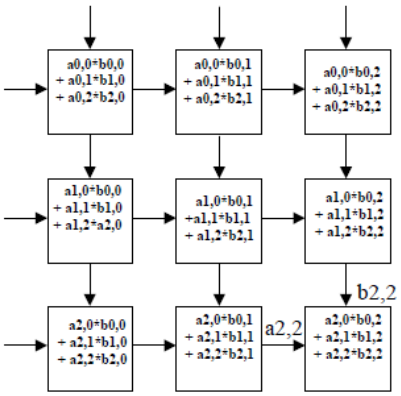


Fig 4 3x3 systolic array matrix multiplication

III.EVOLUTIONARYCOMPUTATION

Evolutionary computation uses computational models of evolutionary processes as key elements in the design and implementation of computer-based problem solving systems. There are a variety of evolutionary computational models that have been proposed and studied which we will refer to as evolutionary algorithms. They share a common conceptual base of simulating the evolution of individual structures via processes of selection and reproduction. These processes depend on the perceived performance (fitness) of the individual structures as defined by an environment.

More precisely, evolutionary algorithms maintain a population of structures that evolve according to rules of selection and other operators, such as recombination and mutation. Each individual in the population receives a measure of its fitness in the environment. Selection focuses attention on high fitness individuals, thus exploiting the available fitness information. Recombination and mutation perturb those individuals, providing general heuristics for exploration.

Although simplistic from a biologist's viewpoint, these algorithms are sufficiently complex to provide robust and powerful adaptive search mechanisms. Figure 3.1 outlines a typical evolutionary algorithm (EA). A population of individual structures is initialized and then evolved from generation to generation by repeated applications of evaluation, selection, recombination, and mutation. The population size N is generally constant in an evolutionary algorithm, although there is no *a priori* reason (other than convenience) to make this assumption.

IV.SYSTOLIC ARCHITECTURE DESIGN

We can design a systolic array for any regular iterative algorithms using: Linear mapping and projection techniques.

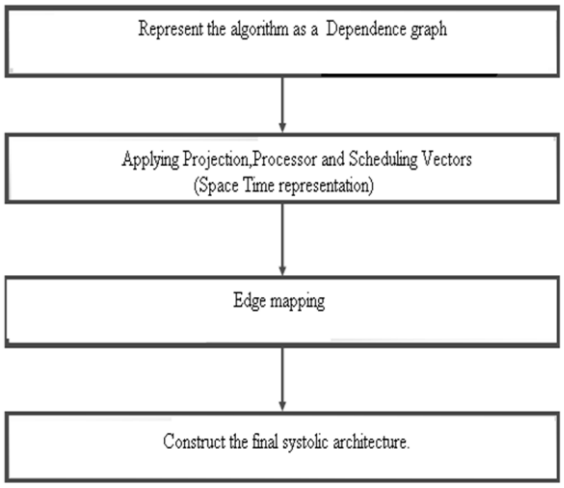


Fig.5: Algorithm for systolic array design

V. FIR SYSTOLIC ARRAYS

Consider the 3-tap FIR filter example.

$y(n) = \omega_0 x(n)+\omega_1 x (n-1)+\omega_2 x (n-2)$

$y(0) = \omega_0 x(n)$

$y(1) = \omega_0 x(1) +\omega_1 x(0)$

$y(2) = \omega_0 x(2) +\omega_1 (1) +\omega_2 x(0)$

$y(3) = \omega_0 x(3) +\omega_1 (2) +\omega_2 x(1)$

$y(4) = \omega_0 x(4) +\omega_1 (3) +\omega_2 x(2)$

$y(5) = \omega_0 x(2) +\omega_1 (4) +\omega_2 x(3)$

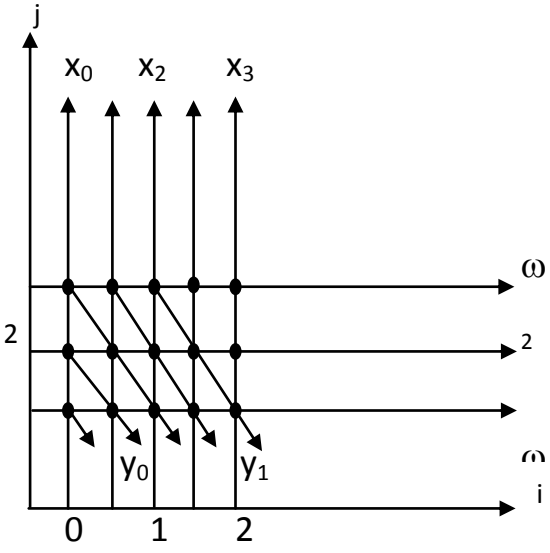


Fig. 6. Space Representation for FIR filtering.

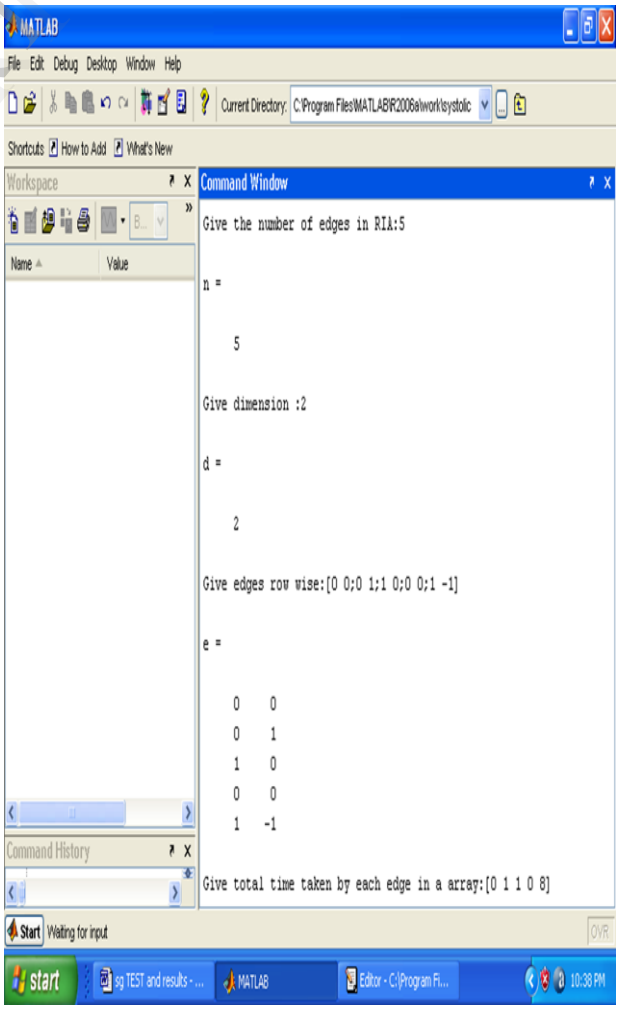
(0 , 0) (0 , 1) (0 , 2)
(1 , 0) (1 , 1) (1 , 2)
(2 , 0) (2 , 1) (2 , 2)
(3 , 0) (3 , 1) (3 , 2)
(4 , 0) (4 , 1) (4 , 2)

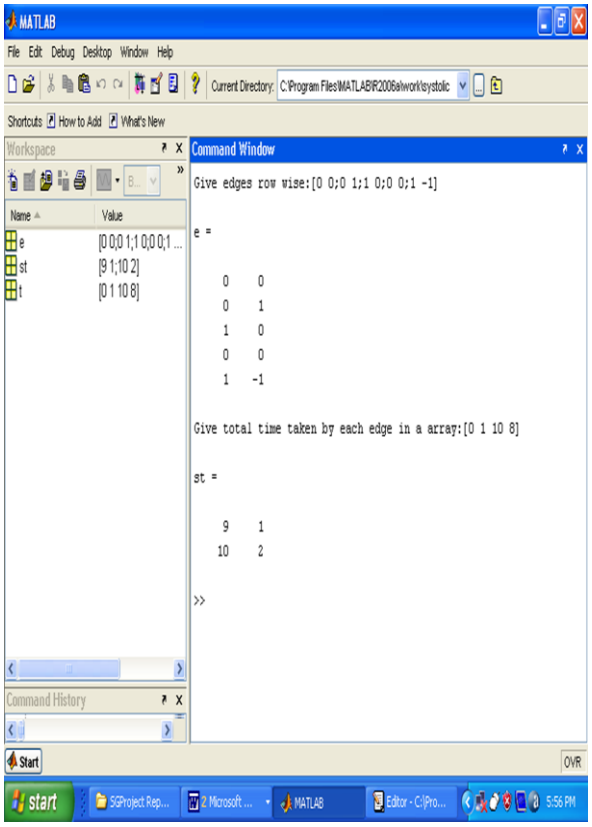
Processor

	P_0	P_1	P_2
0	$x_0\omega_0$		
1	$x_1\omega_0$	$x_0\omega_1$	
2	$x_2\omega_0$	$x_1\omega_1$	$x_0\omega_2$
3	$x_3\omega_0$	$x_2\omega_1$	$x_1\omega_2$
4	$x_4\omega_0$	$x_3\omega_1$	$x_2\omega_2$

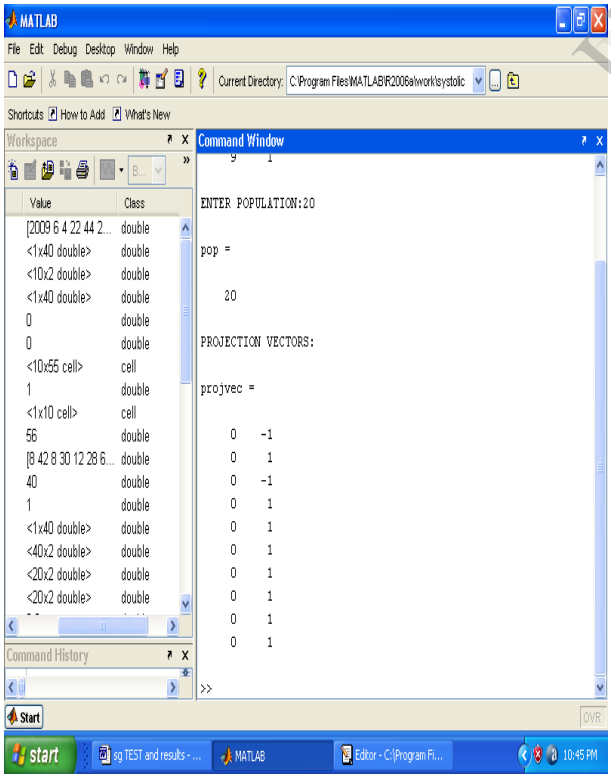
y at time

Verification of the result of design





Result window for scheduling vector



Output window of projection vector

VI. ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of any work would be incomplete without the mention of the people who made it possible. Their constant guidance and encouragement crowned my efforts to success.

. VII. REFERENCES

[1] "Systolic Algorithms & Architectures" by Patrice Quinton and Yves Robert, 1991 Prentice Hall International.
[2] "The New Turing Omnibus" by A.K. Dewdney, New York.
[3] Kalle Tammemäe, system on chip architecture, Dept. of CE, Tallinn Technical University 2000/02.
[4]Richard Hughey, Programming Systolic Arrays,Proc. Int. Conf. Application-Specific Array Processors, IEEE Computer Society, Aug. 4-7, 1992.
[5] VLSI signal processing, lecture 5, systolic arrayarchitecture,cwliu@twins.ee.nctu.edu.tw.
[6] Sun-Yuan-Kung, on supercomputing with systolic/wave front arrays, senior member, IEEE
[7] Systolic Architecture Design by Lan-DaVan, (Ph. D. Department of Computer Science, National ChiaoTung University, Taiwan, R.O.C., fall, 2010).
[8] "Getting started with MATLAB (A Quick Introduction for Scientists and Engineers) - Rudra Pratap", and "Digital Signal Processing Using Matlab V4 - Ingle & Proakis".