DESIGN OF SYSTOLIC BASED OPTIMIZATION TOOL FOR FIR FILTERS USING BINARY TOUR METHOD

CHANDRA SHEKAR P1, RAVI KUMAR M S2

1,2 Department of E&CE, VTU Belgaum, KVGCollege of Engineering, Kurunjibagh Sullia - 574 327, D'Kannada, Karnataka, India

1Chandrashekar_mtech.vlsi@rediffmail.com, 2perajravi@yahoo.com

ABSTRACT

This paper is concerned with the design of systolic array by using linear mapping techniques on regular dependence graph (DG), the mapping technique transforms a Dependency graph to a space-time representation, where each node is mapped to a certain processing element and is scheduled to a certain time instance. The systolic design methodology maps an N-dimensional DG to a lower dimensional systolic architecture. The basic vectors involved in the systolic array design should satisfy feasibility condition for designing the tool. MATLAB version 7.01 is the platform used to design the FIR tool for faster implementation, and to achieve low level designs for selected vectors. The tool designed can also be used in selection of Scheduling inequalities and projection vector to meet the feasibility condition, and to achieve 100% HUE using “Tournament Selection” Method. The Tournament selection typically used in Evolutionary Programming, allows for tuning the degree of stringency of the selection imposed. Rather than Selecting on the basis of each Solutions fitness or error in light of the objective function at hand, selection is made on the basis on the number of wins, earned in a competition.

Keywords: Systolic array, Dependence graph, Processing element, Tournament Selection method.

1. INTRODUCTION

Systolic is commonly referred to as ““pumping” action of a heart. In psychology, the term systolic describes the contraction (systole) of the heart, which regularly sends blood to all cells in the body. Analogously, a systolic computer performs operations in repetitive, rhythmic manner. The word “The term “systolic” was first used in by H.T. Kung. By analogy, systolic computers pump data through each processing elements (PE). In computer architecture, a systolic array is a pipe network arrangement of processing units called cells. It is a specialized form of parallel computing, where cells (i.e. processors), compute data and store it independently of each other. A systolic architecture is an array composed of matrix-like rows of cells. Here, the Processing Elements is similar to central processing units (CPUs). 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 Data Processing Units (DPUs), often with different data flowing in different directions. An SIMD array is a synchronous array of PEs under the supervision of one control unit and all PE receive the same instruction broadcast from the control unit but operate on different data sets from distinct data streams but in systolic arrays usually pipe data from an outside host and also pipe the results back to the host and Eliminating the Von Neumann’s Bottleneck by processing each input multiple times and keep partial results in the PEs. These architectures are well suited for a VLSI or FPGA network implementation.

In tournament selection a number Tour of individuals is chosen randomly from the population and the best individual from this group is selected as parent. This process is repeated as often as individuals must be chosen. These selected parents produce uniform at random offspring. The parameter for tournament selection is the tournament size Tour. Tour takes values ranging from 2 to N ind (number of individuals in population). This work presents in selection of Scheduling inequalities and projection vector to meet the feasibility condition, and to achieve maximum HUE.

2. SYSTOLIC ARCHITECTURE DESIGN

2.1 Basic principle of Systolic array

High performance, special-purpose computer systems are typically used to meet specific application requirements or to off-load computations that are especially taxing to general-purpose computers. As hardware cost and size continue to drop and processing requirements become well-understood in areas such as signals and image processing, more special-purpose systems are being constructed. However, since most of these systems are built on ad hoc and basis for specific tasks, methodological work in this area is rare. Because the knowledge gained from individual experiences is neither accumulated nor properly organized, the same errors are repeated. I/O and computation imbalance is a notable example- often, the fact that I/O interfaces cannot keep up with a devices speed is discovered only after construction a high speed, special-purpose device.
We intend to help correct this ad hoc approach by providing a general guideline—specifically, the concept of systolic architecture, a general methodology for mapping high-level computations into hardware structure. In a systolic system, data flows from the computer memory in a rhythmic fashion, passing through many processing elements before it returns to memory, much as blood circulates to and from the heart. The system works like an automobile assembly line where different peoples work on the same car at different times and many cars are assembled simultaneously. An assembly line is always linear, however, and systolic systems are sometimes two-dimensional.

### 1.1 Flow approach used in the Systolic Design

[Diagram of the systolic design flow]

The systolic architectural concept was developed at Carnegie-Mellon University and versions of systolic processors are being designed and built by several industrial and governmental organizations.

Instead of:

5 million operations per second at most

We have:

30 million operations per second possible:

### 2.2 Systolic Systems

Systolic systems consists of an array of PE (Processing Elements) processors are called cells. Each cell is connected to a small number of nearest neighbors in a mesh like topology. Each cell performs a sequence of operations on data that flows between them. Generally the operations will be the same in each cell; each cell
performs an operation or small number of operations on a data item and then passes it to its neighbor. Systolic arrays compute in “lock-step” with each cell (processor) undertaking alternate compute/communicate phases.

2.3 Processing element

Fig shows the processing element architecture. The IN input stores data coming from the other processor on to a dedicated I/O register file data. The CTE input is connected to a broadcast bus for receiving constant data from the outside world and from the memory. These data are stored in a specific constant register file before being used. As the arithmetic operations involved are very limited, two specific units are implemented, namely an adder and a minimize. These two units are pipelined due to the regular and repetitive structure of the computation. The accumulator can be loaded either from the adder or from the minimize.

The processors of the array are activated by micro-commands which control actions such as accumulator loading, I/O and CTE register selection, data acquisition, etc. These micro-commands are specified by an instruction which is received from the outside and decoded. In the same way, the memory actions are programmed depending on the calculation being performed.

Processor array, reference memory and data reference array thus operate synchronously since; they each execute one instruction every machine cycle. One instruction specifies actions to be realized concurrently on all these units. In that sense, the processor array can be considered to have a SIMD execution mode: all the cells are executing the same instruction at the same time.

2.4 Features of Systolic arrays

A Systolic array is a computing network possessing the some features, such as Synchrony, Modularity, Regularity, Spatial locality, Temporal Locality, Pipelinability, Parallel computing.

Synchrony means that the data is rhythmically computed (Timed by a global clock) and passed through the network.

Modularity means that the array (Finite/Infinite) consists of modular processing units.

Regularity means that the modular processing units are interconnected with homogeneously.

Spatial Locality means that the cell has a local communication interconnection.

Temporal Locality means that the cells transmits the signals from one cell to other which require at least one unit time delay.

Pipelinability means that the array can achieve a high speed.

2.5 Mapping to Systolic Array

Various approaches to map computational algorithms on to systolic array structures have been proposed[13]. Generally they can be classified in to three categories: functional transformation, retiming, and dependence mapping. The dependence mapping consists of several steps:

- Step1:- is to map an algorithm to a Dependence graph (DG),
- Step2:- is to map the dependent graph to a signal flow graph (SFG),
- Step3:- finally maps the SFG on to an array processor.

Dependency diagram is a visual representation of a dependency graph; in the case of a dependency graph without circular dependencies, it can be interpreted as a Hasse diagram of the graph. Dependency diagrams are integral to software development, outlining the complex, interrelationships of various functional elements. Typically in a dependency diagram, arrows point from each module to other modules which they are dependent upon, the mapping of Dependency in shown below Fig3.3

3. TOURNAMENT SELECTION

In tournament selection a number Tour of individuals is chosen randomly from the population and the best individual from this group is selected as parent. This process is repeated as often as individuals must be chosen. These selected parents produce uniform at random offspring. The parameter for tournament selection is the tournament size Tour. Tour takes values ranging from 2 to N ind (number of individuals in population).
In selection the offspring producing individuals are chosen. The first step is fitness assignment. Each individual in the selection pool receives a reproduction probability depending on the own objective value and the objective value of all other individuals in the selection pool. This fitness is used for the actual selection step afterwards.

3.1 Selection schemes
Throughout this section some terms are used for comparing the different selection schemes.

Selective pressure
- Probability of the best individual being selected compared to the average probability of selection of all individuals.

Bias:
- Absolute difference between an individual’s normalized fitness and its expected probability of reproduction.

Spread
- Range of possible values for the number of offspring of an individual.

Loss of diversity
- Proportion of individuals of a population that is not selected during the selection phase

Selection intensity
- Expected average fitness value of the population after applying a selection method to the normalized Gaussian distribution

Selection variance
- Expected variance of the fitness distribution of the population after applying a selection method to the normalized Gaussian distribution

Table and figure show the relation between tournament size and selection intensity.

<table>
<thead>
<tr>
<th>tournament size</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>selection intensity</td>
<td>0</td>
<td>0.56</td>
<td>0.85</td>
<td>1.15</td>
<td>1.53</td>
<td>2.04</td>
</tr>
</tbody>
</table>

Tab. 3-1: Relation between tournament size and selection intensity

3.2 Analysis of tournament selection
In an analysis of tournament selection can be found.

Selection intensity
$$SelInt_{Tournament}(Tour) \approx \sqrt{2 \cdot \ln(Tour) - \ln(4.14 \cdot \ln(Tour))}$$

Loss of diversity
$$LossDiv_{Tournament}(Tour) = \frac{-1}{\ln(Tour)} \cdot \frac{-Tour}{\ln(Tour)}$$

(About 50% of the population are lost at tournament size Tour=5).

Selection variance
$$SelVar_{Tournament}(Tour) \approx \frac{0.918}{\ln(1.186 + 1.328 \cdot Tour)}$$

Importance of Tournament Selection
- A small group of individuals are selected from the whole population.
- The best individual in this group is selected and returned by the operator.
- Tournament selection prevents the best individual from dominating.
- The chosen group size can decrease the selective pressure.
4. RESULTS

![Fig. 3.1: Properties of tournament selection](image)

5. APPLICATIONS

- Matrix Inversion and Decomposition.
- Faster approach can be met in the field of designing.
- Polynomial Evaluation.
- Matrix multiplication.
- Image Processing Convolution.
- Systolic lattice filters used for speech and seismic signal processing.
- Artificial neural network.
- Robotics
- Equation Solving
- Signal processing
- Image processing
- Solutions of differential equations
- Graph algorithms
- Biological sequence comparison
- Other computationally intensive tasks.

CONCLUSION

The systolic architecture is a massively parallel processing with limited I/O communication with host computer and suitable for many regular interactive operations. The programs written to design the TOOL by making use of “TOURNAMENT SELECTION METHOD” in Evolutionary programming, efficiently utilize engineering theory in order to optimize the Systolic array design. The optimized design is compared with the hand calculations done for 1D-FIR tap filter and Reduced Dependency Graph (RDG). The results are matching.
hence I conclude that the design of TOOL shows correct results and is verified. The TOOL designed in this project can be used by any third party for better understanding of “Other Selection Method”.

FUTURE WORK
The analysis of the tool helps in better understanding of the “Other selection Method” thereby further optimization of the given task can be overcome to achieve 100% HUE. The TOOL delivers idea to design Polyphase-FIR filter, DFT, Polyphase-DFT and IDFT-Polyphase function.

ACKNOWLEDGEMENTS
Graceful acknowledgements are an essential part of any successful work. Hence, the very purpose of this paper would be defeated without thanking people, involved in this venture either directly or indirectly. I consider myself proud to be a part of KVG College of Engineering family, the institute that stood by way in all endeavors. I profusely thank DR. RAVI KUMAR M S ,HOD of ECE department, KVGE, Sullia, for his guidance, constructive criticism and timely suggestion without which I could not have seen the light of success in my work, and MR. PRASHANTH Asst.Professor, ECE department, YDIT, Bengaluru for giving cooperation and source of inspiration. I wish to thank all our teachers, friends and family members for their continuous support and encouragement throughout the work. I would especially like to thank the reviewer committee those who my manuscript and offered many suggestions for this paper improvement and also thank for the publisher.

REFERENCES