**MAIN PROJECT REPORT –SYSTOLIC PROCESSOR ARCHITECTURE IMPLEMENTION USING FPGA**

**PROJECT GUIDE : SUBMITTED BY :**

Deepak J Puthukkaden (20)

Giridhar A K (31)

Govindh B (32)

Rohit Sreekumar (53)

Mr. Rashid P.E Shine Ali (59)

**Chapter 1**

**INTRODUCTION**

The most widely known mode of information processing is sequential processing in which the operations that have to be performed to obtain the solution of a problem are carried out in a sequence by a single processing unit. In case the processor is computer it is termed sequential computing. The most common computer design is still based on a von Neumann architecture, which is composed of a single processor executing single sequence of instructions on a single stream of data known also under SISD machine. In von Neumann architecture both instructions and data come from a common memory, while in a Harvard architecture data and instructions are stored in different memories and are fed to the processor by two different paths . The conceptual simplicity of the sequential approach and associated technology at the time led to digital computer revolution.

For most applications sequential computers present an efficient solution providing the required processing power. Nevertheless, there were, are, and will be problems that require computational power that exceeds the resources of the most powerful sequential computers at the time for several orders of magnitude. These problems are not only those referred to as large-scale scientific computing problems, but also several real-time signal and image processing problems. Some of them were covered in the previous chapter.

In order to achieve radically higher computational throughput new approaches must be considered. One of already prominent approaches is solving problems in parallel. As opposite to sequential processing, parallel processing could be regarded as a mode of information processing in which at least two processing units cooperate while carrying out in parallel the information processing operations that belong to a problem. During the process each processing unit works on a different part of a problem.

Usually, multiprocessing and parallel computing are used as synonyms referring to a number of electronic processing units involved in parallel computing of a particular computational problem. If number of these processing units is large, in order of hundreds or thousands, the attribute massively is added.

The main reason for parallel execution of a problem is the acceleration of computations. If single processing unit can accomplish the task in time *T* then *n* processing units could ideally accomplish it in time *T/n*. This is termed a linear speed-up. In most cases the actual speed-up achieved on parallel computers is considerably smaller then the desired linear speed-up.

In parallel computer architectures, a systolicarray is a homogeneous network of tightly coupled Data Processing Units (DPUs) called cells or nodes. Each node or DPU independently computes a partial result as a function of the data received from its upstream neighbors, stores the result within itself and passes it downstream.

The parallel input data flows through a network of hard-wired processor nodes, resembling the human brain which combine, process, merge or sort the input data into a derived result. Because the wave - like propagation of data through a systolic array resembles the pulse of the human circulatory system, the name systolic was coined from medical terminology. The name is derived from Systole (medicine) as an analogy to the regular pumping of blood by the heart.

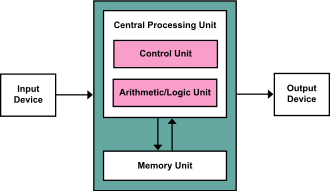
**Chapter 2**

**LITERATURE REVIEW**

**2.1 Comparison**

**2.1.1 Von Neumann Architecture**

This design architecture consists of a processing unit containing an arithmetic logic unit and processor registers, a control unit containing an instruction register and program counter, a memory to store both data and instructions, external mass storage, and input and output mechanisms. The meaning has evolved to be any stored-program computer in which an instruction fetch and a data operation cannot occur at the same time because they share a common bus.



**Fig 1 : Von Neumann Architecture**

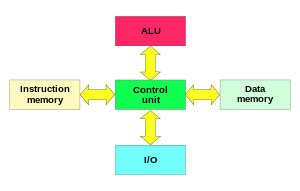
**Limitations and Overcoming Methods :-**

The main limitation of the von Neumann architecture is known as the "von Neumann bottleneck". This is due to the fact that all instructions and all data must pass through the same shared common multiplexed bus to get in or out of the processor, sooner or later things have to wait for other things to get access to this multiplexed bus and the processor gets starved for instructions and/or data. The result is the processor is unable to maintain its designed performance but waits idle instead of doing work.

A major benefit of systolic arrays is that all operand data and partial results are stored within (passing through) the processor array. There is no need to access external buses, main memory or internal caches during each operation as is the case with Von Neumann or sequential machines.

**2.1.2 Harvard Architecture**

The Harvardarchitecture is a computer architecture with physically separate storage and signal pathways for instructions and data.  These machines had data storage entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator; the processor could not initialize itself.



**Fig 2 : Harvard Architecture**

**Limitations and Overcoming Methods :-**

Free data memory cant be used for instruction and vice- versa. Production of a computer with two buses is more expensive and needs more time. Moreover this implementation requires more number of pins and it is not commonly used.  It is possible to access program memory and data memory simultaneously. Typically, code (or program) memory is read-only and data memory is read-write. Therefore, it is impossible for program contents to be modified by the program itself. All these can be overcome using Systolic Architecture.

**2.2 Theory**

Systolic systems consists of an array of Processing Elements. Each elements are called cells, and each cell is connected to a small number of nearest neighbours 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. It 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.

A Systolic array is a computing network possessing the following features:

• **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.

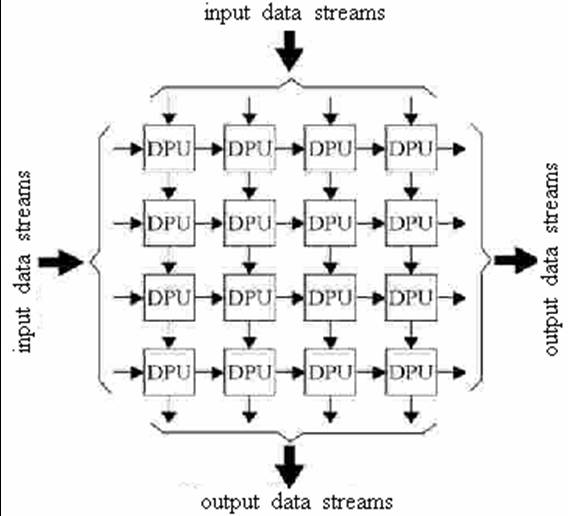
• **Regularity:-** means that the modular processing units are interconnected with homogeneously.

• **Spatial Locality:-** means that the cells has a local communication interconnection.

• **Temporal Locality:-** means that the cells transmits the signals from from one cell to other which

require at least one unit time delay.

• **Pipelinability:-** means that the array can achieve a high speed.



**Fig 3 : Systolic array network**

A systolic array is composed of matrix-like rows of data processing units called cells. Data processing units (DPUs) are similar to central processing units (CPUs), (except for the usual lack of a program counter, since operation is transport-triggered, i.e., by the arrival of a data object). Each cell shares the information with its neighbors immediately after processing. The systolic array is often rectangular where data flows across the array between neighbour DPU’s, often with different data flowing in different directions. The data streams entering and leaving the ports of the array are generated by auto-sequencing memory units, ASM’s. Each ASM includes a data counter. In embedded systems a data stream may also be input from and/or output to an external source. Systolic arrays are arrays of DPUs which are connected to a small number of nearest neighbour DPUs in a mesh-like topology. DPUs perform a sequence of operations on data that flows between them. Because the traditional systolic array synthesis methods have been practiced by algebraic algorithms, only uniform arrays with only linear pipes can be obtained, so that the architectures are the same in all DPUs. The consequence is, that only applications with regular data dependencies can be implemented on classical systolic arrays. Like SIMD machines, clocked systolic arrays compute in "lock-step" with each processor undertaking alternate compute | communicate phases.

A systolic array typically consists of a large monolithic network of primitive computing nodes which can be hardwired or software configured for a specific application. The nodes are usually fixed and identical, while the interconnect is programmable. The more general wavefront processors, by contrast, employ sophisticated and individually programmable nodes which may or may not be monolithic, depending on the array size and design parameters. The other distinction is that systolic arrays rely on synchronous data transfers, while wavefront tend to work asynchronously.

Unlike the more common Von Neumann architecture, where program execution follows a script of instructions stored in common memory, addressed and sequenced under the control of the CPU's program counter (PC), the individual nodes within a systolic array are triggered by the arrival of new data and always process the data in exactly the same way. The actual processing within each node may be hard wired or block microcoded, in which case the common node personality can be block programmable.

The systolic array paradigm with data-streams driven by data counters, is the counterpart of the Von Neumann architecture with instruction-stream driven by a program counter. Because a systolic array usually sends and receives multiple data streams, and multiple data counters are needed to generate these data streams, it supports data parallelism.

The actual nodes can be simple and hardwired or consist of more sophisticated units using micro code, which may be block programmable.

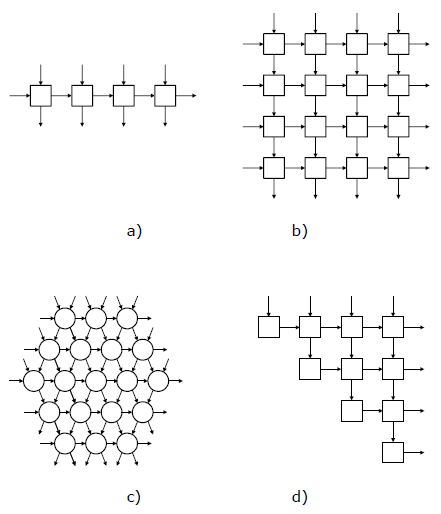
**2.2.1 Systolic Array**

The term systolic array was introduced in the computer science by H. T. Kung at the end of 70's. A systolic array typically consists of a large number of similar processing elements interconnected in an array. The interconnections are local meaning that each processing element can communicate only with a limited number of neighbouring processing elements. Data move at a constant velocity through the systolic array passing from one processing element to the next processing element. Each of the processing elements performs computations, thus contributing to the overall processing needed to be done by the array.

Systolic arrays are synchronous systems. A global clock synchronises the exchange of data between directly communicating processing elements. Data can be exchanged only at the ticks of the clock. Between two consecutive clock ticks, each processing element carries out computation on the data which it has received upon the last tick and produces data which is sent to neighbouring processing elements at the next clock tick. The processing element can also hold data stored in the local memory of the processing element.

The systolic arrays are usually represented as array of processing elements and array of interconnections connecting the PEs with some particular pattern. In other words, systolic array is a collection of PEs that are arranged in a particular pattern. The pattern and the PEs are defined with the implemented algorithm.

The PEs are composed of combinatorial part and/or memory. The combinatorial part is responsible for arithmetic operations required by the systolic algorithm. By memory (registers) we denote delay elements within the PE that hold data and thus control data flow into and out of the PE. In general, no large memory is associated with PEs.



**Fig 4 : Examples of Systolic Array a) linear systolic array, b) orthogonal systolic array,**

**c) hexagonal systolic array and d) triangular systolic array.**

• **Linear systolic array** :- Processing elements are arranged in one-dimension. The inter-connections between the processing elements are nearest neighbour only. Linear systolic arrays differ relative to the number of data flows and their relative velocities.

• **Orthogonal systolic array** :- Processing elements are arranged in a two-dimensional grid. Each processing element is interconnected to its nearest neighbours to the north, east, south and west. Again, the systolic arrays differ relative to the number and direction of data flows and the number of delay elements arranged in them.

• **Hexagonal Systolic array** :- Processing elements are arranged in a two-dimensional grid. The processing elements are connected with its nearest neighbours where inter-connections have hexagonal symmetry.

• **Triangular Systolic array :-** It refers to two-dimensional systolic array where processing elements are arranged in a triangular form. This topology is mostly used in different algorithms from linear algebra. In particular it is used in Gaussian elimination and other decomposition algorithms.

**2.2.1 Classification**

With the tremendous development of the VLSI technologies a wide variety of computer architectures have been proposed. In the last 20 years infinite number of new computer architectures for parallel processing have been innovated based on major approaches of parallel computing developed in 1960s and 1970s. With increasing number of computer architectures, it has become important to find a way for efficient classification. The classification should distinguish those structures that are significantly different and at the same time reveal the similarities between apparently divergent designs.

Taxonomies, or classification schemes, are widely used for classifying the world. For example, whales are grouped with other mammals rather than with fish. Their relationship with mammals is far more important than their relationship to fish even though it might be less obvious at the first glance .

Due to diversity of parallel architectures, diverse definitions have been proposed. Up to now many authors classified the computer architectures. One of the most widely adopted classifications is Flynn's taxonomy based on instruction and data stream organisation. essor architectures.

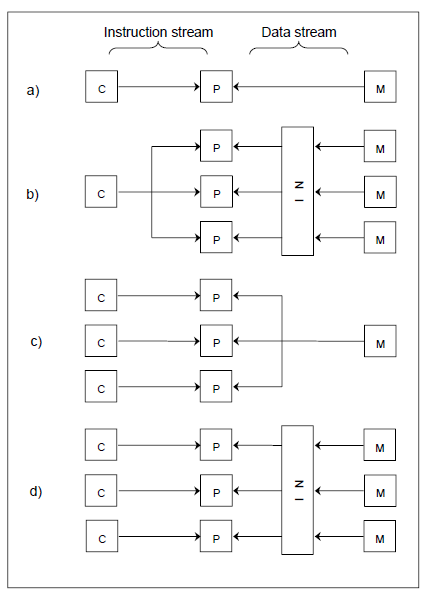
Flynn’s taxonomy divides the entire computer world into four groups: SISD, SIMD, MISD and MIMD. Flynn classifies architectures on the presence of single or multiple streams of instructions and data into:

• SISD (Single Instruction Single Data):- defines serial computers.

• SIMD (Single Instruction Multiple Data):- involves multiple processors simultaneously executing the same instruction on different data.

• MISD (Multiple Instruction Single Data):- involves multiple processors applying different instructions to a single datum; this hypothetical possibility is generally unrealistic but which Flynn affirms to include specialised streaming organisation.

• MIMD (Multiple Instruction Multiple Data):- involves multiple processors autonomously executing diverse instructions on diverse data.



**Fig 4 : Flynn’s taxonomy of computer architectures: a) SISD, b) SIMD, c) MISD, and d) MIMD (C: control unit, P: processor, M: memory, I N: interconnection**