

the structure of the PES themselves is kept to a maximum simplicity and usually into operation units. For design simplification, there are few types of PES in the same system. For the same token, interconnections are kept to a nearest neighbor topology in order to minimize communication delays as well as power distribution issues. Note that topologies include two neighbors (linear arrays), four neighbors (square arrays), or six neighbors (hexagonal arrays) required by the problem to solve. This is notably due to the fact that scheduling mechanisms are based upon local criteria such as data availability. However, it should be noted that the same clock in all the computation cells. Linear systolic arrays can tolerate clock skews at the expense of multidimensional designs require slower clocks in order to compensate. In order to overcome these mechanisms, the design of a systolic array emphasizes an efficient mapping of the problem to the architecture. An example of a band matrix-vector multiplication is shown in Fig. 6. The synchronization of the processors and of the input data rate has been mapped to the requirements of the problem. Note that each processor is designed to operate upon two arguments. In summary, it should be noted that systolic arrays are very efficient at solving computationally intensive problems which involve many repetitive low-level calculations. Also, the simplicity of their design renders their function fixed at design time.”)

Gaudiot at Figure 6:

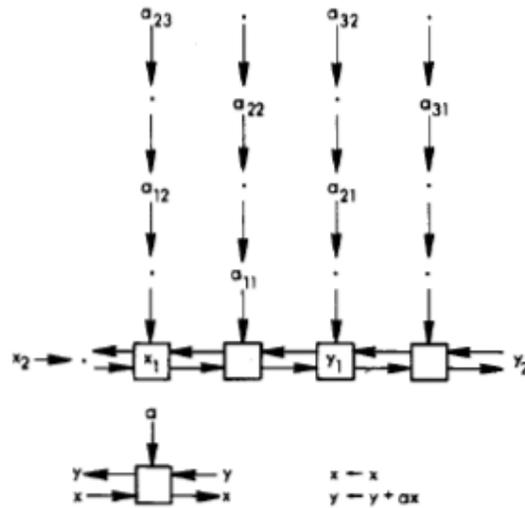


Fig. 6. A linear systolic array.

To the extent Plaintiff asserts this limitation is not expressly or inherently disclosed in the prior art, the apparent claim construction, or any other claim construction, the claimed subject matter would not have been obvious to a person of ordinary skill in the art considering this reference in combination with the knowledge of one of ordinary skill in the art at the time of the alleged invention. Plaintiff's disclosures in one or more of the references identified in Section I.B.2 of the cover sheet are hereby disclosed.

[1B] transforming an algorithm into a data driven calculation that is implemented by said reconfigurable computing system at the at least one reconfigurable processor;

At least under Plaintiff's apparent theories of infringement and interpretations of the claim limitation, alleging that any of Defendant's accused products satisfy this claim limitation, Gaudiot in combination with one or more references, discloses:

Gaudiot at Abstract ("New technologies of integration allow the design of powerful multiprocessors that may include several thousands of elementary processors. These multiprocessors may be used in a wide range of applications in signal and data processing. However, assuring the proper interconnection of a large number of processors and the ultimate safe execution of the user programs present a significant scheduling problem. The scheduling of operations upon the availability of their operations is a complex problem.")

termed the data-driven mode of execution and offers an elegant solution to the issue is described in this paper and several architectures which have been proposed or implemented (systolic arrays, data-flow machines, etc.) are examined in detail. The problems associated with data-driven execution are also studied. A multi-level approach to high-speed digital signal processing is then evaluated.”)

Gaudiot at 1220-21 (“The overall objective of this paper is to demonstrate the application of data-driven principles of execution to the design of high-performance signal and data processing architectures. Several approaches will be demonstrated and their particular domain of application will be contrasted. The description of low-level processing systems is beyond the scope of this paper and the interested reader is referred to an excellent survey by Allen [3]. Instead, we focus here on the issues related to building high-performance multiprocessors for signal processing applications. In Section II, we show the type of problems considered in signal processing. In Section III, data-flow principles of execution as they relate to digital signal processing problems are described in detail in Section III while several existing data-driven architectures are described in Section IV. In Section V, we analyze a multi-level data-driven architecture and examine its program execution environment. Conclusions are drawn in Section VI.”)

Gaudiot at 1222 (“C. Structure Handling

This is a crucial issue in signal processing for this kind of application requires that multiple elements which belong to the same structure be processed in a parallel or pipelined fashion. The basic premises of data-flow principles states that an output is a function of its inputs, regardless of the state of the machine at the time of execution.”)

Gaudiot at 1224 (“IV. DATA-DRIVEN ARCHITECTURES

We now describe in detail several systems which operate at runtime, compile-time, or design-time under data-driven execution. Although it is generally considered that data-flow principles of execution are in effect at runtime, we extend their domain of application to design-time systems and refer to them as data-driven systems. We thus initially examine multiprocessors where data dependencies have been frozen at design time (systolic arrays). We then consider reconfigurable programmable systolic arrays (the Wavefront Array Processor) and multiprocessors where compile time by the use of data-flow program graphs (the ESL polycyclic processor

Gaudiot at 1224-25 (“A. Systolic Arrays [32]

The primary goal of a systolic array is to make use of the large amount of processing available in VLSI technology by using repetitive circuitry to perform signal processing, matrix operations, image processing, etc. In summary, a systolic array is simply a collection of interconnected Processing Elements (PES). In order to incorporate as many processors as possible, the structure of the PES themselves is kept to a maximum simplicity and usually includes only a few operation units. For design simplification, there are few types of PES in the same systolic array. The same token, interconnections are kept to a nearest neighbor topology in order to minimize communication delays as well as power distribution issues. Note that topologies include two neighbors (linear arrays), four neighbors (square arrays), or six neighbors (hexagonal arrays), depending on the requirements required by the problem to solve. This is notably due to the fact that scheduling mechanisms are based upon local criteria such as data availability. However, it should be noted that the same clock is used in all the computation cells. Linear systolic arrays can tolerate clock skews at the expense of performance. Multidimensional designs require slower clocks in order to compensate. In order to solve these problems, synchronization mechanisms, the design of a systolic array emphasizes an efficient mapping of the problem to the architecture. An example of a band matrix-vector multiplication is shown in Fig. 6. The design of the systolic array, the synchronization of the processors and of the input data rate has been mapped to the requirements of the problem. Note that each processor is designed to operate upon two data items and two arguments. In summary, it should be noted that systolic arrays are very efficient at solving large scale, data intensive problems which involve many repetitive low-level calculations. Also, the fixed nature of their design renders their function fixed at design time.”)

Gaudiot at Figure 6:

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.