Advanced high resolution methods for real-time signal processing require fast multi-processor architectures, realized as Very Large Scale Integrated (VLSI) systems, for their implementation. A class of parallel architectures, 'systolic arrays', fulfills the demands of signal processing and at the same time conforms to the limitations of VLSI technology.

The design of efficient systems of systolic arrays is an iterative process in which the information gathered from mapping algorithms onto hardware is influential in the development of the algorithms. Presently, the absence of mapping tools makes it extremely difficult to find good systolic implementations for many important problems. Often, one cannot do better than implementing structurally simple but numerically inferior algorithms, which is undesirable for most signal processing tasks.

A methodology is proposed that automates the mapping of recurrence equations to processor arrays. Two aspects distinguish our methodology from extant work: (1) complex coupled systems of recurrence equations can be systematically treated and (2) the resulting...
19. Systolic systems are optimal.

The proposed methodology takes as input recurrence equations describing the algorithm, along with certain desirable hardware features. A sophisticated optimization procedure is then applied to generate the description of optimal arrays that implement the algorithm. If there is no satisfactory implementation, the algorithm will have to be revised, in order to improve pipelinability, without sacrificing numerical stability.

Several classes of applications will significantly benefit from this approach: fast analysis of parallel algorithms, efficient partitioning of algorithms, and determination of optimal architectures for the implementation of multiple algorithms.
Fast Design of Good Systolic Systems

FINAL REPORT
J.-M. Delosme, I. Ipsen
January 12, 1989

U. S. ARMY RESEARCH OFFICE
DAAL03-96-K-0158
YALE UNIVERSITY

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED
A. Statement of Problem Studied

Systems for such applications as radar and sonar detection and identification, robot vision, medical imaging and geophysical exploration have to extract features from or make decisions based on large amounts of data. In order to minimize the computation time or to keep storage low, such systems often process data in real time and therefore consist of many processors. To be sufficiently fast, the processors are special purpose arithmetic computers, presently realized as VLSI chips. Because of their impact on the whole system, some limitations of the VLSI technology must be taken into account at the outset of the design process. The most important are the high design cost for developing novel processor architectures and the limited input/output capabilities of chips due to small pin counts. These constraints combined with the desire for efficient pipelining have led to the concept of 'systolic arrays'.

Systolic arrays consist of a few types of processors that perform elementary operations (such as additions or multiplications) selected by a simple control unit. The pattern of interconnections among processors is regular and each processor is connected to only a few other processors. Since the types of operations performed are essentially data independent, the processors can be made to operate synchronously without affecting the global performance; this simplifies the control of the processors and reduces the area needed to implement them in silicon.

The process of designing such systems can be decomposed into three steps:

- Development of signal processing techniques.
- Development of fast, stable and pipelinable algorithms that perform the optimization.
- Scheduling the algorithms onto minimal cost systolic architectures that deliver the desired throughput and comply with the current technological constraints. The best architecture may be a combination of several systolic arrays with different processors or interconnection patterns.

Clearly the three design steps are not completely independent and the best designs will be obtained through an iterative procedure; if there is no satisfactory implementation of an algorithm it will have to be revised. To make such iterations practically feasible the time consuming scheduling process must be automated and efficient schedules must be determined at every iteration. The class of algorithms to be implemented on systolic arrays is large, yet most of them can be expressed as systems of recurrence equations. The goal of the scheduling method is to construct a realistic systolic array model and to find optimal schedules (in this model) for any system of recurrence equations. By applying the method, while still under development, to the design of parallel implementations for algorithms, we can in turn refine the model of systolic algorithms and the objective functions that characterize good systolic systems.

B. Summary of the Most Important Results

Referring to the issues in Section 5 of our proposal, we can sum up our contributions supported under this ARO contract as follows:

1. Preprocessing of Recurrence Equations. We exploit the structure of an algorithm by decomposing its dependence graph into strongly connected components. By introducing the concept of dependence mapping, we generalize the established concept of dependence vector and can so represent many more signal processing algorithms.

2. Determination of the Class of Local Transformations. We have developed notions and procedures (e.g. time cones and computability analysis) to determine whether indeed an algorithm is well-defined and to provide a compact description of all systolic schedules. By applying
a combination of transformations from a restricted class (affine, folding and propagation transformations) we can transform the class of affine recurrences to the simpler class of uniform recurrences (with conditionals), which is directly amenable to systolic implementation.

The introduction of dependence mappings is a crucial first step in developing a systematic and efficient method for transforming broadcast dependences to propagation dependences that require a minimal communication bandwidth.

3. Solution of the Global Optimization Problem and Determination of Local Transformations. The time cone and the removal of broadcast dependences make it finally possible to solve the difficult problem of minimizing the computation time, more precisely: the product of systolic cycle time and number of cycles. Tools from integer linear programming theory, which we explored in the context of time optimization, turn out to provide tight bounds on the search space for optimal processor allocation schemes; hitherto no bounds were available and good solutions could be missed.

The problem of finding a processor allocation that balances the load on each processor and minimizes the amount of interprocessor communication is highly combinatorial, especially for multistep algorithms. We have developed procedures for generating many possible processor allocation schemes; the problem left is to explore these possibilities and to select an (almost) optimum one: use of the simulated annealing method results in very high quality solutions. A long term investigation of simulated annealing resulted in a novel annealing schedule, significantly faster than any other in the literature. Using this schedule optimal allocations can now be obtained in a reasonable amount of time.

In order to assure that the systolic array satisfies the required performance specifications (e.g. matching the specified data rate for real-time applications), we have explored ‘partitioning strategies’ for grouping computations to be mapped to the same processor. These were applied to schedule several algorithms onto the CMU WARP.

4. Algorithm Design. As our motivation is the development of parallel algorithms and efficient implementations, we have used the above methods to analyze the structure of algorithms suitable for systolic implementation. Conversely, the study of difficult algorithms has helped us to advance our work on scheduling and processor allocation. For example, we have investigated the parallel implementation of singular value and symmetric eigenvalue problems, Toeplitz system solution, computation of partial correlations, and mesh refinement for partial differential equations.

Further details can be found in the progress reports.

C. List of All Publications and Technical Reports

\begin{itemize}
\end{itemize}


D. Scientific Personal Supported

Jean-Marc Delosme
Ilse Ipsen
Ata Arjomand
Jimmy Lam
Liz Jessup
Theo Omtzigt
Yi-Wan Wong
Jin-Hong Ma

Advanced Degrees Earned by Scientific Personnel:

Jimmy Lam: Ph.D. in Computer Science, December 1988