

TK
8315
V45
1996

PROCEEDINGS
EUROPTO
SERIES

Vision Systems: Sensors, Sensor Systems, and Components

Otmar Loffeld
Chair/Editor

10–12 June 1996
Besançon, France

Sponsored by

CNRS—Centre National de la Recherche Scientifique

Ministère de la Défense

Direction Générale de l'Armement

Direction de la Recherche et de la Technologie

Essilor Paris

Université de Franche-Comté, Besançon

EOS—The European Optical Society

SPIE—The International Society for Optical Engineering

The Commission of the European Communities, Directorate General
for Science, Research, and Development

RECEIVED

SEP 27 1996

UNIVERSITY LIBRARY



Volume 2784

PROCEEDINGS
EUROPTO
SERIES

***Vision Systems: Sensors, Sensor
Systems, and Components***

Otmar Loffeld
Chair/Editor

10–12 June 1996
Besançon, France

Sponsored by

CNRS—Centre National de la Recherche Scientifique

Ministère de la Défense

Direction Générale de l'Armement

Direction de la Recherche et de la Technologie

Essilor Paris

Université de Franche-Comté, Besançon

EOS—The European Optical Society

SPIE—The International Society for Optical Engineering

The Commission of the European Communities, Directorate General
for Science, Research, and Development

Published by

SPIE—The International Society for Optical Engineering

CLASS SEP.
SERIAL



Volume 2784

SPIE is an international technical society dedicated to advancing engineering and scientific applications of optical, photonic, imaging, electronic, and optoelectronic technologies.

Kurt F. Wendt Library
University of Wisconsin - Madison
215 N. Randall Avenue
Madison, WI 53706-1688



The papers appearing in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and are published as presented and without change, in the interests of timely dissemination. Their inclusion in this publication does not necessarily constitute endorsement by the editors or by SPIE.

Please use the following format to cite material from this book:

Author(s), "Title of paper," in *Vision Systems: Sensors, Sensor Systems, and Components*, Otmar Loffeld, Editor, Proc. SPIE 2784, page numbers (1996).

Library of Congress Catalog Card No. 96-67692
ISBN 0-8194-2170-7

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 360/676-3290 (Pacific Time) • Fax 360/647-1445

Copyright ©1996, The Society of Photo-Optical Instrumentation Engineers.

Copying of material in this book for internal or personal use, or for the internal or personal use of specific clients, beyond the fair use provisions granted by the U.S. Copyright Law is authorized by SPIE subject to payment of copying fees. The Transactional Reporting Service base fee for this volume is \$6.00 per article (or portion thereof), which should be paid directly to the Copyright Clearance Center (CCC), 222 Rosewood Drive, Danvers, MA 01923. Payment may also be made electronically through CCC Online at <http://www.directory.net/copyright/>. Other copying for republication, resale, advertising or promotion, or any form of systematic or multiple reproduction of any material in this book is prohibited except with permission in writing from the publisher. The CCC fee code is 0-8194-2170-7/96/\$6.00.

Printed in the United States of America.

Wendt
TK
8315
V45
1996

AVE 6057

Contents

v Conference Committee

SESSION 1 IMAGING SENSORS AND APPLICATIONS

- 2 **Log polar image sensor in CMOS technology [2784-02]**
D. Scheffer, B. Dierickx, F. Pardo, J. Vlummens, G. Meynants, L. Hermans, IMEC (Belgium)
- 12 **Coded-light setups with optimized transformations of code indices [2784-04]**
P. Lehle, H. Gärtner, H. J. Tiziani, C. Volland, Univ. Stuttgart (FRG)
- 21 **Structured light measurement by double-scan technique [2784-05]**
H. Gärtner, P. Lehle, H. J. Tiziani, C. Volland, Univ. Stuttgart (FRG)

SESSION 2 MULTISENSOR DATA FUSION

- 32 **Information gained by data fusion [2784-06]**
C. Arndt, O. Loffeld, Zentrum für Sensorsysteme (FRG)
- 41 **Information gained by pseudoredundant measurements [2784-07]**
C. Arndt, O. Loffeld, Zentrum für Sensorsysteme (FRG)
- 50 **Data fusing for optimization of spark ignition engine control [2784-08]**
M. Scherer, T. Ganser, Daimler-Benz (FRG); O. Loffeld, Zentrum für Sensorsysteme (FRG)

SESSION 3 SENSORS AND EFFECTS: TECHNIQUES AND APPLICATIONS

- 62 **Integrated intelligent sensor for the textile industry [2784-11]**
P. Peltié, D. David, CEA-LETI (France)
- 74 **Realization of a multisensor data fusion algorithm for spark ignition engine control [2784-20]**
T. Ganser, M. Scherer, Daimler-Benz (FRG); O. Loffeld, Zentrum für Sensorsysteme (FRG)
- 80 **Object recognition by the use of a hand scanner and Fourier transform [2784-14]**
N. Karlsson, Linköping Univ. (Sweden); L. Axelsson, Saab Military Aircraft (Sweden)
- 90 **Spatial processing of sky background radiation for remote sensing of atmosphere parameters [2784-15]**
R. R. Agishev, R. A. Bajazitov, M. M. Galejev, Kazan State Technical Univ. (Russia)

SESSION 4 INTERFEROMETRIC TECHNIQUES

- 98 **Birefringent fiber remote strain sensor with FMCW interferometry [2784-16]**
M. Campbell, G. Zheng, P. A. Wallace, Glasgow Caledonian Univ. (UK)
- 102 **Sagnac birefringent fiber strain sensor with FMCW technique [2784-17]**
G. Zheng, M. Campbell, P. A. Wallace, Glasgow Caledonian Univ. (UK)

- 106 **Robust unwrapper for two-dimensional images [2784-18]**
M. Arevalillo Herráez, D. R. Burton, M. J. Lalor, D. B. Clegg, Liverpool John Moores Univ. (UK)
- 111 **Optoelectronic/image processing module for enhanced fringe pattern acquisition and analysis [2784-19]**
G. Dymny, M. Kujawińska, Warsaw Univ. of Technology (Poland)

SESSION 5 SPECIAL SESSION ON COMPONENTS

- 120 **Parallel architecture dedicated to image component labeling in $O(n \log n)$: FPGA implementation [2784-21]**
E. Mozeš, S. Weber, J. Jaber, G. Prieur, LIEN/Univ. de Nancy I (France)
- 126 **Color line scan technology: new tool for industrial process and quality control applications [2784-22]**
G. F. Lemström, P. H. Pämänen, Temet Vision Industry Oy (Finland)
- 136 **Instruction systolic array in image processing applications [2784-23]**
M. Schimmler, ISATEC Soft- und Hardware GmbH (FRG); H.-W. Lang, Fachhochschule Flensburg (FRG)
- 145 **Network-based imaging system for the OMEGA laser system [2784-24]**
J. Z. Roach, S. W. Swales, Univ. of Rochester (USA)
- 154 *Author Index*

The instruction systolic array in image processing applications

Manfred Schimmmler

ISATEC Soft- und Hardware GmbH
Schauenburger Str. 114-116, 24118 Kiel, Germany

Hans-Werner Lang

Fachhochschule Flensburg
Kanzleistr. 91-93, 24943 Flensburg, Germany

ABSTRACT

The ISATEC parallel computer is the first implementation of an instruction systolic array for the commercial market. The goal of integration of 1024 processors on an add-on-board for PCs has been achieved by the development of a low-power/low-area processor architecture whose instruction set is suited particularly for image processing applications. The paper introduces the concept of the instruction systolic array, its implementation and some application examples in the field of image processing.

Keywords: Parallel processing, instruction systolic array, median filter, FFT, computer tomography

1. INTRODUCTION

Instruction systolic arrays (ISAs) have been developed to provide a programmable high performance hardware for specific computation intensive applications. Typically, such an array is connected to a sequential host, thus operating like a coprocessor which solves only the computationally intensive tasks within a global application. The ISA is a mesh-connected processor grid, where each processor is connected to its four direct neighbors by two unidirectional data wires. The processors are controlled by three streams of control information: instructions, row-selectors and column-selectors. Communication between adjacent processors is performed via designated communication registers. The whole system is globally clocked. Since data and control information is propagated only to physical neighbors, clock skew problems cannot occur.

This paper discusses the implementation of an instruction systolic array as a PC board with 1024 simple processors. The machine can be programmed in C/C++ and Pascal with an additional parallel interface language called LAISA. The parallel computer is particularly suited for image processing tasks since it achieves its peak performance in applications where identical operations have to be performed with large data sets.

The concept of instruction systolic arrays is introduced in the Section 2. Section 3 gives an overview over the architecture of the ISATEC parallel computer. It is documented how the ISA has been integrated on an add-on board for commercial PCs and how the different stages of cache memory are organized in order to meet the requirements of a large variety of application fields.

Several applications in the field of image processing are discussed in Section 4 which give insight into different techniques to parallelize computationally intensive standard routines.

The performance of the machine is presented in Section 5. Besides the theoretical peak performance values some benchmarks for image processing routines are given.

2. THE ISA CONCEPT

The ISA is a square array of identical processors, each connected to its four direct neighbors by data wires. The array is synchronized by a global clock. The processors are controlled by instructions, row- and column selectors.

The instructions are input in the upper left corner of the processor array, and from there they move step by step in horizontal and vertical direction through the array. This guarantees that within each diagonal of the array the same instruction is active during a single clock cycle.

The selectors also move systolically through the array: the row selectors horizontally from left to right, the column selectors vertically from top to bottom (Figure 1). The selectors mask the execution of the instructions within the processors, i.e. an instruction is executed if and only if both selector bits currently in that processor are equal to one. This construct leads to a very flexible structure which creates the possibility of very efficient solutions for a large variety of applications.

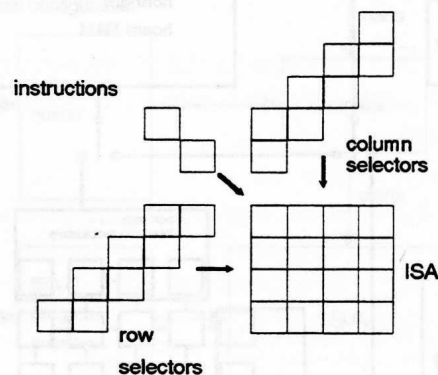


Fig. 1: Control flow in an ISA

Every processor has read and write access to its own memory. Besides that, it has a designated communication register that can also be read by the four neighbor processors. Two adjacent processors can exchange data by writing to their own communication register and afterwards reading the from communication register of the other in two subsequent clock phases. Within one clock phase the read access is always performed before the write access. This convention on the one hand avoids read/write conflicts, on the other hand it creates the possibility to broadcast information across a whole row or column with one single instruction.

The processors operate in a bit serial way. The execution of each instruction takes exactly one word clock which comprises 16 bit clock cycles. More complex operations (like multiplication) are composed of two or even more

subsequent instructions. Since every instruction is propagated to the neighbor processor directly after its execution, the processors do not require their own control memory.

Due to the following four properties, the ISA is superior to most of the comparable architectures:

- local communication for data and control flow,
- global clock without negative implications of clock skew,
- modularity, scalability
- wide applicability.

ISATEC holds the patents for the architecture in Europe and the US.

3. ARCHITECTURE OF THE PARALLEL COMPUTER

The ISA on the board is a 4×4 array of processor chips. Each chip contains 64 Processors, arranged as an 8×8 square. This provides 1024 processors on the board. In order to exploit the computation capabilities of this unit, it is necessary to provide data and control information at an extremely high speed. Therefore, a cascaded memory concept is implemented on board that forms a fast input and output environment for the parallel processing unit.

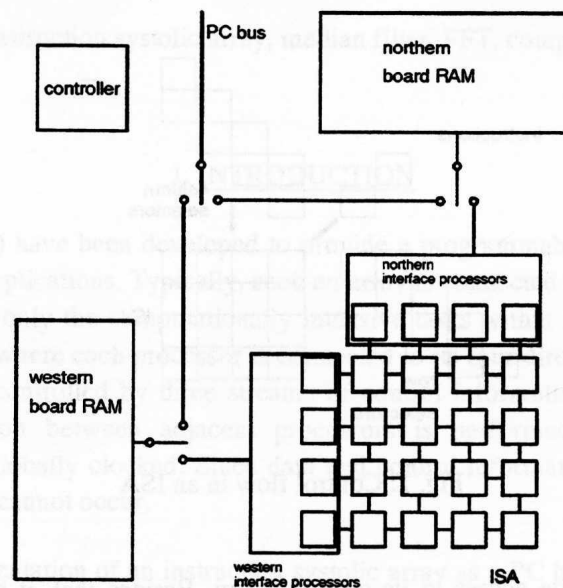


Fig. 2: Data paths in the parallel computer

For the fast data exchange with the processor array there are rows of intelligent memory units at the northern and western borders of the array. These units are called interface processors. Eight interface processors are integrated on one chip. Each interface processor is connected to its adjacent array processor by a single wire for data transfer in each direction.

The interface processors have access to an on-board memory by means of special fast data channels, those at the northern interface chips with the northern board RAM, and those of the western chips with the western board RAM. The northern and the western board RAM can communicate bidirectionally with the PC memory.

The data transfer between every two memory units within this hierarchy is controlled by an on-board controller chip. In particular, it controls the channels between the interface processors and the board RAMs as well as between the board RAMs and the bus of the PC (Figure 2). The controller receives its instructions either directly from the PC as board instructions, or it can operate autonomously. In the second case it receives controller instructions from an instruction queue, which is located on the board, too. The instruction queue is loaded before the execution of an application.

The ISA gets its instructions from the ISA program memory. It is also loaded before the execution with the desired application programs. The execution of such programs is started by the controller, as indicated in Figure 3.

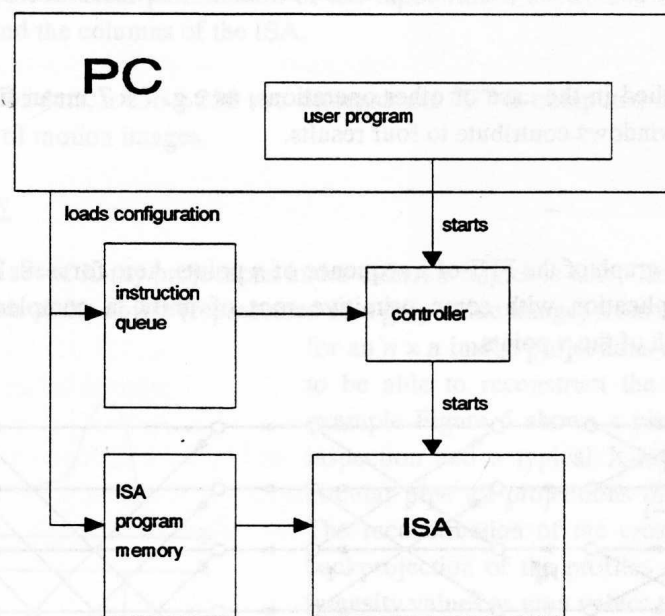


Fig. 3: Execution of a program

4. APPLICATIONS IN IMAGE PROCESSING

The natural way of mapping an image processing problem onto a two-dimensional array processor architecture is to associate each pixel of a 32×32 window of the image with one processor. However, there are a lot of other possibilities of such mappings. As an example, an entire row of 1024 pixels can be mapped onto the 1024 processors of the ISA in row-major way. Since it is possible to store more than one pixel in each processor, several rows of the

image can be processed simultaneously with the additional benefit that local operations in the columns of the image take place in on processor (i.e. no routing is necessary).

4.1. Image Filtering

In order to implement a two-dimensional filter operation, e.g. a 7×7 median filter, the pixels of the image have to be stored in the processor array in a way that the locality of the operation corresponds to local routing of data on the array.

The solution that we chose is to store all m pixels of a row of the image in adjacent $m/32$ rows of the ISA. The main reason for this is for better support of the row-wise input of image data from the camera.

In the case of a 7×7 median filter, the 49 pixels are sorted according to their grey value and the 25th value is taken as the result. The procedure can be optimized by taking into account the observation that the four overlapping 7×7 windows of every 2×2 adjacent pixels have 36 values in common. After pre-sorting of these 36 values the median 14 elements remain as candidates for the medians of each of the four 7×7 windows. The medians are computed by performing a merge operation between the 14 candidates and the $13 = 49 - 36$ elements not considered so far of each window.

Similar techniques can be applied in the case of other operations, as e.g. 7×7 mean filter, where the 36 common elements of four overlapping windows contribute to four results.

4.2. Fast Fourier Transform

Figure 4 shows the signal flow graph of the FFT of a sequence of n points, here for $n=8$. It consists of $\log n$ stages. In each stage, a complex multiplication with some primitive root of unity, a complex addition, and a complex subtraction is performed at each of the n points.

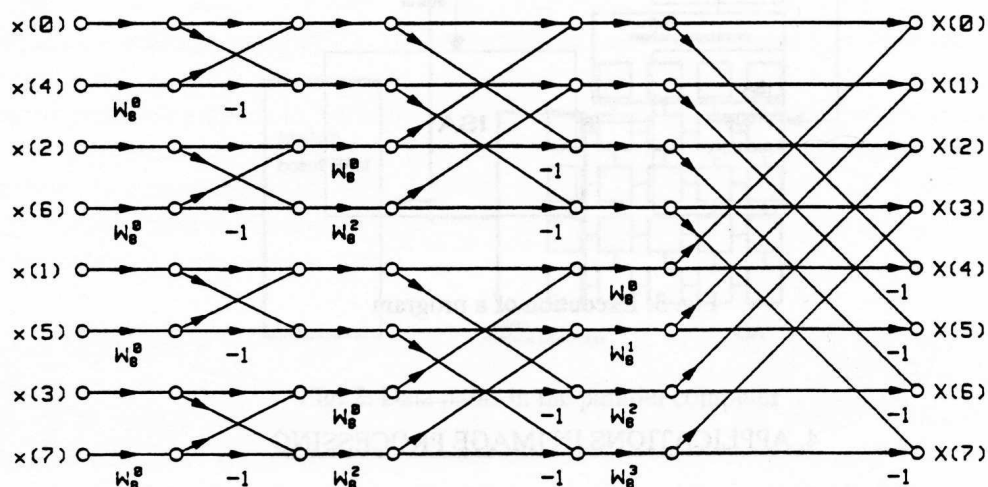


Fig. 4: Signal flow graph of an 8-point FFT

The output of each stage is permuted by interchanging the two halves of subsequences of length 2^k before being input to the next stage. At the beginning, the sequence of the n points is permuted according to the so called bit reversal permutation.

The implementation of a 2-dimensional FFT of size $n \times n$ on the ISA consists of two subsequent passes: first the rows are transformed and then the columns. Each row is mapped onto $n/64$ rows of the ISA such that each processor stores two values. In the case of $n=256$, for instance, four rows of the processor array are needed for one row of data.

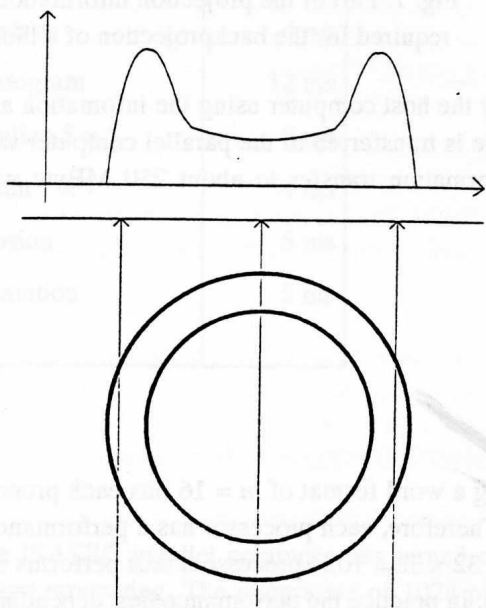
The complex roots of unity are precomputed and are loaded to the processor array when needed. This load operation takes only a small constant number of instruction cycles since the values can be broadcast along the columns of the ISA (on the ISA, broadcasting of a 16-bit word can be done within one instruction cycle). The computation in each stage requires 4 (real) multiplications and 6 (real) additions in each processor.

The implementation of the permutation between every two stages involves routing of data, which is performed efficiently either within each row of the ISA (in the case of interchanging the two halves of subsequences of length ≤ 64) or by interchanging complete rows of the ISA (in the case of interchanging the two halves of longer subsequences). The initial bit reversal permutation of the input values is realized in a similar way by subsequent permutations on the rows and the columns of the ISA.

Including input and output of data, a 256×256 two-dimensional FFT is computed on the ISA within 30 ms, which allows a real-time filtering of motion images.

4.3. Computer Tomography

In computer tomography, a set of X-ray projections under different angles is taken from some object. The problem is to reconstruct a cross section of the object (represented as a grey value image) from such a set of profiles. Typically,



for an $n \times n$ image projections of at least n angles are required to be able to reconstruct the object precisely. As a simple example Figure 5 shows a piece of pipe as the object under inspection and a typical X-ray projection. In this case of a circular pipe all projections of different angles are identical. The reconstruction of the cross section is now computed by backprojection of the profiles over the image and adding the intensity values as grey values to the individual pixels. Figure 6 shows the reconstructed image of the cross section if only six angles (0, 30, 60, 90, 120 and 150 degrees) are used.

The implementation of the reconstruction on the parallel computer is done by decomposition of the target image into small 32×32 -pixel tiles and subsequently mapping these tiles to the processor array. For each tile now the projection information of one projection is distributed to the processors and each processor interpolates the grey value to be added and performs this addition. After completion the next projection is broadcast etc.

Fig. 5: Projection profile of a cross section of a pipe

The problem of the implementation of this algorithm on the parallel computer is the amount of data required for the reconstruction step. E.g. for reconstruction of a 1024×1024 image there are 1024 profiles, each consisting of 1024 floating point numbers. These data are required 1024 times since this is the number of tiles of the decomposition of the whole image. The total amount of 4 GByte to be transferred between host computer and parallel unit would of course dominate the overall execution time.

The solution lies in the selection of profile information to be sent to the parallel unit. For each tile only a fraction of profile information is really required for the accumulation step (Figure 7).

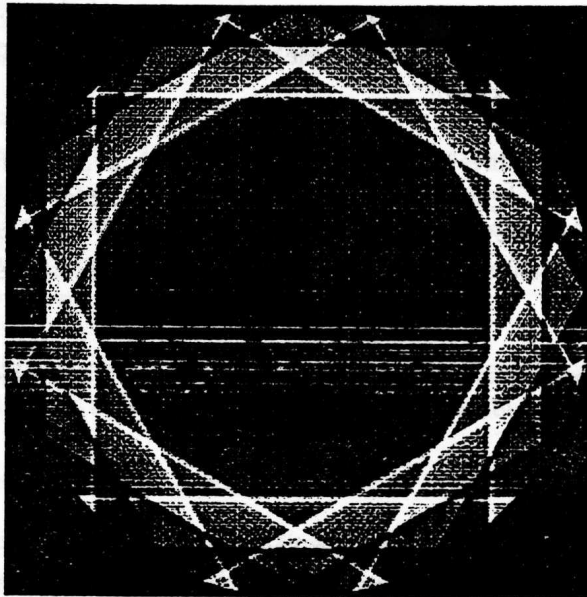


Fig. 6: Reconstructed cross section of pipe from 6 projections

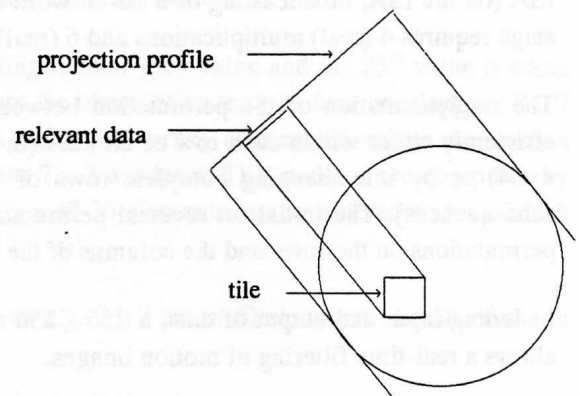


Fig. 7: Part of the projection information required for the backprojection of a tile

The selection of the interval of projection information is performed by the host computer using the information about the actual angle and the actual tile. Only a small subarray of the profile is transferred to the parallel computer which performs the reconstruction step. This optimization reduces the information transfer to about 250 MByte which requires about 4 seconds via the PCI bus of the PC.

5. PERFORMANCE

5.1. Peak performance

Fixed point arithmetic: At a clock frequency of $f = 50$ MHz and using a word format of $m = 16$ bits each processor can execute $f/m = 50/16 \cdot 10^6 = 3 \cdot 10^6$ word operations per second. Therefore, each processor has a performance of 3 MIPS (millions of instructions per second). The whole array with its $32 \times 32 = 1024$ processors thus performs up to 3125 MIPS. This value characterizes the theoretical peak performance. In practice the performance lies, dependent on the application, between 1000 and 3000 MIPS.

Floating-point arithmetic: The multiplication of two matrices may serve as an example for floating point performance and data rate of the parallel computer.

Using the double precision floating point format of IEEE (64 bit) one multiplication requires 16 machine instructions and one addition takes 19 instructions. For a scalar product operation 45 instructions are necessary (10 to fetch the operands and store them internally, 16 for the multiplication, and 19 for the addition). Each processor reaches a performance of 0,14 MFlops (millions of floating point operations per second). This leads to a value of 142 MFlops for the whole processor array.

Using single precision floating point format (32 bit), a scalar product operation requires 27 instructions (6 for fetching the operands, 7 for the multiplication, and 14 for the addition). This adds up to a performance of 237 MFlops.

Note that these numbers are *not* theoretical peak values, but *real* performance measurements of this particular application. The peak values are 183 MFlops for double precision and 305 MFlops for single precision.

5.2. Benchmarks for image processing

A number of standard functions frequently used in image processing have been implemented as parallel routines for C/C++ and Pascal. Here we give the benchmarks for these routines operating on 16-bit grey value images of size 256×256 .

Operation	Time
2D-FFT	30 ms
Laplace	4 ms
Histogram	12 ms
Median 5×5	6 ms
Mean 7×7	4 ms
Erosion	5 ms
Dilatation	5 ms

6. CONCLUSION

The ISATEC parallel computer has turned out to be a good architecture for high speed applications like on-line image processing. The integration of 1024 processors on an add-on board for PCs provides a parallel unit with an effective performance of up to 3000 MIPS. This allows to perform applications like an on-line 2-dim-FFT on a commercial PC.

Programming the parallel unit is supported by a comfortable programming environment as well as a rather complete library of efficient routines for standard tasks in several areas, in particular in the image processing field.

7. REFERENCES

1. Dittrich, A., Schmeck, H.: Given's Rotation on the Instruction Systolic Array. In: Wolf, G., Legendi, T., Schendel, U. (eds.): PARCELLA '88, Mathematical Research, Bd. 48, Akademie Verlag, Berlin, 340-346 (1988)
2. Kunde, M., Lang, H.-W., Schimmler, M., Schmeck, H., Schröder, H.: The Instruction Systolic Array and its Relation to Other Models of Parallel Computers, *Parallel Computing* 7, 25-39 (1988)
3. Lang, H.-W.: The Instruction Systolic Array, a Parallel Architecture for VLSI, *Integration, the VLSI Journal* 4, 65-74 (1986)
4. Lang, H.-W.: ISA and SISA, Two Variants of a General Purpose Systolic Array Architecture, 2nd Int. Conf on Supercomputing, 460-465 (1988)
5. Lang, H.-W., Maaß, R., Schimmler, M.: The Instruction Systolic Array - Implementation of a Low-Cost Parallel Architecture as Add-On Board for Personal Computers, *Proc. HPCN 94, München* (1994).
6. Riesemeier, H., Schimmler, M.: Der ISATEC Parallelrechner beim Einsatz für die Röntgen- und Computertomographie. *Proc. HPSC '95, Jülich, Germany*, 191-198 (1995).
7. Schimmler, M.: Fast Sorting on the Instruction Systolic Array, Technical Report No. 8709, Christian-Albrechts-University, Kiel (1987)
8. Schimmler, M.: Realisierung eines Instruction Systolic Array als Parallelrechner-Zusatzkarte für PC's, *Proc. HCF (Hochschul-Computer-Forum)*, Berlin (1993).
9. Schimmler, M.: Instruction Systolic Arrays - Experiences with a First Implementation. *ISCA '92, Hamilton Island, Australia* (1992).
10. Schimmler, M.: A bit-serial floating-point Processor below 10 mW, *Proc. 7. Fachtagung Mikroelektronik, Zwickau, Germany* (1996).
11. Schmeck, H.: A Comparison-Based Instruction Systolic Array, In: Robert, Y. et al. (eds): *Parallel Algorithms and Architectures*, North Holland (1986)
12. Strazdins, P.E.: Control Structures for Mesh-Connected Networks, PhD Thesis, The Australian National University, Canberra, Australia (1990).
13. Yuan, J., Svensson, C.: High Speed CMOS Circuit Technique, *IEEE Journal of Solid State Circuits*, Vol. 24, 62-70 (1989)