

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

AIAA 82-4129

The Massively Parallel Processor

D.H. Schaefer,* J.R. Fischer,† and K.R. Wallgren‡
NASA Goddard Space Flight Center, Greenbelt, Md.

Introduction

THE massively parallel processor (MPP) is a single-instruction-stream/multiple-data-stream (SIMD) computer designed for the rapid and economical extraction of information from data, especially data in the form of images. The MPP consists of 16,384 processing elements (PE's) and supporting control and data management functions. The performance of the MPP is summarized in Table 1. The construction of a digital processor utilizing the very high degree of parallelism embodied in the MPP has not been previously attempted. The MPP is being built for the Goddard Space Flight Center by the Goodyear Aerospace Corporation. Delivery of the system is scheduled for 1983.

MPP Applications

The mid-1980's Earth application usage of spacecraft images will require detailed monitoring of world food supplies, atmospheric conditions, mineral resources, water and land resources, human pollution, and other physical features of the planet on which we live. The corresponding high information data rate cannot be processed with present, conventional computer systems. The need is for special image processing systems capable of real-time data processing of images.

If, for example, the thematic mapper instrument were operated 24 hours a day, this single instrument would produce 10^{13} bits of data per day to be potentially processed.

Typically, each image pixel (8 to 10 data bits) requires 10^2 to 10^4 computer operations for correlating and classification. This implies computer operating rates of 10^9 to 10^{11} operations/s on pixel data for 10^{13} bits/day. The maximum effective operating rate of existing computers is 10^8 operations/s or a factor of 10 to 10^3 short dependent on the number of operations/pixel. These figures are for just one instrument of the many image sensing spacecraft NASA will be orbiting in the 1980's.

The massively parallel processor will provide NASA with the ability to rapidly extract information from images sensed by applications and scientific spacecraft. Examples of such spacecraft are the Landsat series of spacecraft and the Large Space Telescope. The thematic mapper and the multilinear array are examples of sensors that produce data at very high rates. The MPP will perform radiometric correction, geometric correction, cross correlation, and classification tasks at speeds between one and two orders of magnitude faster than present capability. It will enable experimenters to

develop involved image classification algorithms, and to execute these programs in minutes or seconds, rather than hours as required on a conventional computer. Due to its high input rate, the MPP will be able to process images in near real-time as they are received from spacecraft.

The MPP will be used both as an element in an operational system and as a research tool. In the research mode, the MPP will be available for experimentation in such areas as image understanding, target recognition, and the production of imagery from three-dimensional software models. Experiments on the MPP will give insight into the strength of the present architecture, and provide guidelines for improved architecture. Initial work on hardware methods of producing compact massively parallel systems suitable for space use has been undertaken,¹ and the future direction of such work will be greatly influenced by experimentation on the MPP.

System Overview

The MPP system consists of four units. These units (shown in Fig. 1) are: 1) the Array Unit (ARU) consisting of 16,384 processing elements (PE's) arranged in a 128×128 array with nearest neighbor communication; 2) the Array Control Unit (ACU) that broadcasts control signals to all PE's of the ARU; 3) a conventional computer, the "Companion Processor," that manages overall data flow between MPP units; and 4) a staging buffer to provide buffering and reformatting of data as it enters and exists the ARU.

Array Unit

The Array Unit (ARU) is the 128×128 array of processing elements (PE's), and their associated 1024-bit random-access local-memories. Each PE is connected to its nearest neighbors—north, south, east and west. Opposite array edges can be connected together to form either a plane, a horizontal cylinder, a vertical cylinder, or a torus.

Arithmetic in each PE is performed in a bit-serial manner using a full adder and a shift-register to recirculate operands through the adder. The fundamental clock rate is 10 MHz for a cycle time of 100 ns.

The ARU inputs and outputs data over 128 parallel lines. The maximum transfer rate of 128-bit columns of data is 10 MHz for a maximum bandwidth of 1.28 billion bits/s. Input, output, or both can occur simultaneously with processing.

Table 1 Speed of typical operations

| Operation | Execution speed, MOPS ^a |
|-----------------------------------|------------------------------------|
| Addition of arrays | |
| 8-bit integers (9-bit sum) | 6553 |
| 12-bit integers (13-bit sum) | 4428 |
| 32-bit floating-point numbers | 430 |
| Multiplication of arrays | |
| 8-bit integers (16-bit product) | 1861 |
| 12-bit integers (24-bit product) | 910 |
| 32-bit floating-point numbers | 216 |
| Multiplication of array by scalar | |
| 8-bit integers (16-bit product) | 2340 |
| 12-bit integers (24-bit product) | 1260 |
| 32-bit floating-point numbers | 373 |

^a Million operations per second.

Received Jan. 1, 1981; revision received Nov. 19, 1981. Copyright © 1981 by D.H. Schaefer. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Presently, Adjunct Associate Professor, George Mason University, Fairfax, Va.

†Staff Engineer, MPP Project, Computer Development Section.

‡MPP Project Manager, Computer Development Section.

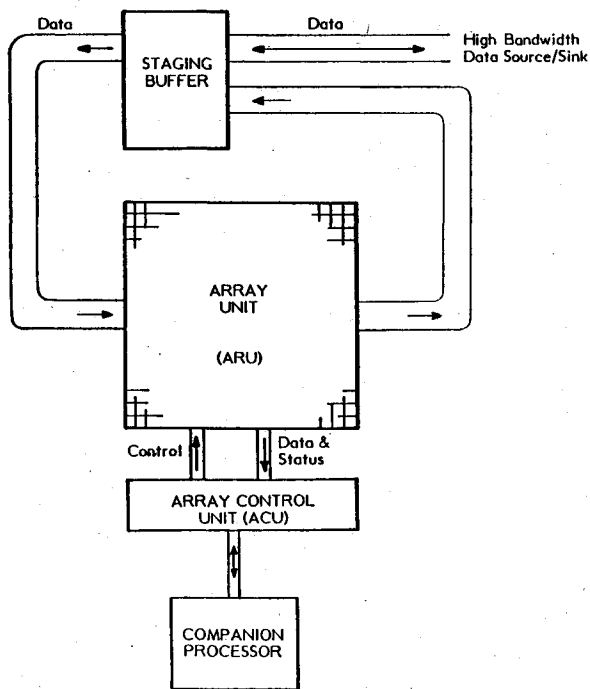


Fig. 1 The MPP system.

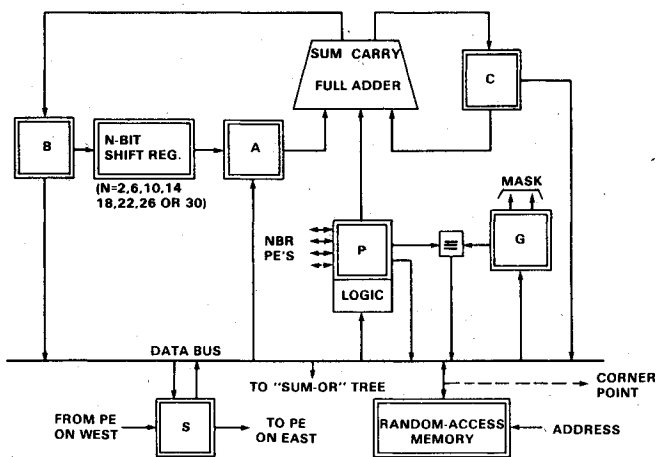


Fig. 2 Single processing element.

The data-bus states of all 16,384 PE's are combined in a tree of inclusive-or elements called the "SUM-OR tree." The single wire output from this tree is fed to the ACU, and used in certain operations such as finding the maximum or minimum value of an array.

MPP Processing Element

A single processing element is shown functionally in Fig. 2. The P-register, together with its input logic, performs all logic functions and can also receive data from any one of its four nearest neighbors. The A-, B-, and C-registers, the shift register, and associated logic form an arithmetic unit. The G-register controls masking of arithmetic logic and routing operations. (Unmasked operations are performed in all PE's. Masked operations are only performed in those PE's where the G-register equals 1.) The S-register is used to shift data into and out of the PE array without disturbing PE operations. The random access memory contains 1024 bits, and is on an IC chip separate from the PE's. The PE is interconnected with memory via a bidirectional data bus.

The P-register can be used for Boolean logic or for inter-PE routing. A logic operation combines the data bit on the data bus with the state of the P-register at the beginning of the machine cycle. The result of the logic operation is stored in

the P-register at the end of the machine cycle. All of the 16 Boolean functions of two variables are implemented. Any P-register can also read the state of the P-register in a neighboring PE (north, south, east or west in the square array) thus providing for the transfer of information from one PE to another.

The arithmetic unit consists of the full adder, the B- and C-registers, and a variable-length shift register.

The adder receives inputs from the A-register and the P-register. It adds these two bits to a carry bit stored in the C-register to form a two-bit sum. The least-significant bit of the sum is clocked into the B-register and the most significant bit of the sum is clocked into the C-register to become the carry bit for the next machine cycle.

The shift register has 30 states. Groups of stages can be bypassed so that the length of the register can be either 2, 6, 10, 14, 18, 22, 26, or 30 stages long. The A- and B-registers of the adder contribute two additional stages. Thus, the round trip path from the adder back to the adder is either 4, 8, 12, 16, 20, 24, 28, or 32 stages in length.

The S-register is used to input and output ARU data. While the PE's are processing data already present in their random-access memories, columns of additional input data are shifted into the west side of the ARU (Fig. 1) and through the S-registers, until a plane of 16,384 bits is loaded. This input plane is then stored in the random-access memories, in one 100-ns cycle, by stopping the processing momentarily in all PE's and moving the S-register values to the memory elements. Planes of data are output in a similar fashion through the east side of the ARU. Up to 160 Mbytes/s can be transferred through the ARU I/O ports.

Microcircuit Technology

All components of eight PE's, exclusive of random-access memory, are packaged on a custom microelectronic chip. This chip utilizes complementary mosfet (CMOS) technology.

This chip contains a two row by four column array of PE's, plus a parity tree and bypass gates.

Each processing element within the ARU is coupled to 1024 bits of local memory. These memories are commercial 1K × 4 bit static RAMS. Use of standard RAM integrated circuits for local memory will allow future MPP's to have expanded storage as advances occur in solid-state memory technology.

Array Control Unit

The Array Control Unit (ACU) broadcasts control signals and memory addresses to all PE's of the ARU, and receives ARU status bits. It is designed to perform bookkeeping operations (address calculation, loop control, branching, subroutine-calling, etc.), and control the ARU simultaneously. It contains three parts: 1) the PE Control Unit, 2) the Main Control Unit, and 3) the I/O Control Unit.

The PE Control Unit controls all ARU functions except the S-register functions. It executes routines stored in its program memory to perform all array operations required by applications programs.

The I/O Control Unit controls the shifting of I/O data through the ARU S-registers, and the transfer of I/O data between the S-registers and the ARU memory. It executes I/O programs stored in main control's program memory.

The Main Control Unit executes the application program stored in its program memory. It performs the scalar arithmetic operations required, calls the PE control unit for all array logic and array arithmetic operations, and calls for I/O operations. Both sets of calls are queued to await execution while the Main Control Unit moves on to generate other calls.

All three control units operate concurrently with a minimum of interference. Interference between the Main Control Unit and the PE Control Unit is minimized by using independent program memories and a queue for the PE Control Unit calls. The I/O Control Unit interferes with the PE Control Unit only when a bit-plane of data must be trans-

ferred between the S-registers and the ARU memory. The I/O Control Unit interferes with the Main Control Unit only when it reads an I/O-command from main control's program memory. One I/O command can move many bit-planes of data so this interference is minimized.

Companion Processor

A conventional computer manages data flow between MPP units, loads programs into the ACU, executes system-test and diagnostic routines, and provides program development facilities. This computer initially will be a DEC PDP-11/34. After system checkout, a DEC VAX 11/780 will assume the functions of the companion processor.

Staging Buffer

The MPP system will include a staging memory for buffering ARU data. This memory provides the "corner turning" function, which converts conventional byte-oriented data into the bit plane form needed by the ARU. The capacity of this memory will be 512 Kbytes when the MPP is delivered. It will be expandable to 16 Mbytes (using currently available memory chips) with a transfer rate to and from the ARU of 160 Mbytes/s. Later use of advanced memory chips will allow the capacity to increase to 64 Mbytes.

Software

System software for the MPP consists of program development software and diagnostic software. The program development software supports usage of the MPP's massive data processing capability while the diagnostics package supports checkout of the MPP system hardware. The user software interface to MPP will be via assembler language or Parallel Pascal. Parallel Pascal is an extension of standard Pascal which allows direct use of the MPP array architecture while programming in the high-level language. A user subroutine library will be provided.

Conclusion

Spatially parallel computers, such as the MPP, hold the promise of being the prevalent type of computer that will be used in making decisions based on real-time image inputs. Such on-line automatic decision making based on visual inputs is needed both for governmental and industrial uses. Future computers having over one million processing elements are envisioned. The MPP is an important initial step in the realization of computers that really can "see."

References

- ¹Schaefer, D.H., "Massively Parallel Processing Systems for Space Application," *Proceedings of 2nd AIAA Computers in Aerospace Conference*, Oct. 1979, p. 284.
- ²Batcher, K.E., "Design of a Massively Parallel Processor," *IEEE Transactions on Computers*, Vol. C29, Sept. 1980, pp. 836-840.

AIAA 82-4130

Deadband Control System Limit Cycles Analyzed by Fourier Series

Franklin C. Loesch*

The Aerospace Corporation, El Segundo, Calif.

Introduction

THE limit cycles of a missile's post-boost deadband pneumatic attitude control system are analyzed by calculating the switching time delays of the real system

relative to an ideal system in which the transfer function of each linear element is 1. These on and off delays are first derived by considering the geometric and timing relations between the missile's motion and the ideal switching lines in the missile's phase plane. The same delays are then calculated by representing the limit cycle rate vs time as a Fourier series; then deriving the deadband switch's input as another Fourier series in which the dynamic response of each linear element is accounted for. The limit cycle solution is obtained by a simple trial and interpolation method, which finds the maximum rate and duty fraction for which corresponding delays as calculated by both procedures are equal. Using 50 to 100 Fourier series terms, this method gives essentially exact results that agree with digital simulations within 1 to 2%. The accuracy of the method is easily assessed graphically, or by determining the changes in solution values as the number of Fourier terms and the size of the interpolation intervals are changed. Unequal solenoid valve on and off delays are accounted for easily. The method is limited to simple deadband systems for which the character of the limit cycle's rate time history is known beforehand. The method may become impractical for duty fractions less than about 1%, because the Fourier series may not accurately represent the system near the switching points with an acceptable number of terms.

Other methods for calculating limit cycle characteristics include Patapoff's,¹ which is restricted to small duty fraction cases for which all transients have essentially decayed prior to switch on. His method is not applicable to such cases as analyzed here, in which the off delay may even exceed the time to cross the deadband. Approximate methods include simple engineering estimates of the effective delays of the linear elements. It is shown here that the usual estimates of these delays result in overpredictions of duty fractions by 40 to 400%. The well-known describing function technique^{2,3} is a complicated approximate method, which is not able to account for the unequal on and off valve delays, and which has no intrinsic accuracy assessment capability. It appears that with the availability of computers for which the Fourier series and trial computations are trivial, the Fourier series method is superior to the describing function method because it is essentially exact rather than approximate, its accuracy is easily assessed, it accounts for unequal valve delays without resort to Padé³ approximations, and it appears to be simpler.

Phase Plane Delay/Rate/Duty Fraction Relations

Figure 1 shows one axis of the deadband system. The pneumatic thrusters impart a constant angular acceleration θ . Throughout this Note, all angular rates except ω are made nondimensional by dividing by $\lambda\theta$, where λ is the rate gain, and all angles except ωt are normalized by dividing by $\lambda^2\theta$. For brevity, these divisors are omitted; so B stands for $B/\lambda^2\theta$, etc. Figure 2 shows the missile's symmetric limit cycle in the nondimensional phase plane. The lines with slope of -1 are the ideal switching lines; i.e., these would be the switching lines if the transfer functions of all linear elements in the system were 1. The real transfer functions are not 1; hence, the real switch off occurs at point 0' rather than point 0. Nozzle 1 off occurs later than 0', due to the valve off delay; it may occur at point 1, before, or at alternate point 1, after crossing the No. 2 ideal on line.

The total on time is $4\theta_{\max}$. The total drift time is $4(\theta_4/\theta_{\max})$. Using the phase plane parabola relation to replace θ_4 by θ_{\max} , the limit cycle angular frequency ω and the duty fraction D are obtained:

$$(\omega\lambda/2\pi) = D/4\theta_{\max} \quad D = \theta_{\max}^2 / [\theta_{\max} + (\theta_{\max}^2/2)] \quad (1)$$

The relevant points in Fig. 2 are determined by intersections or time differences as shown in Table 1.

By defining these intersections and time differences analytically, using the equations for the No. 1 parabola, the ideal switching lines and the drift lines, together with sym-

Received April 24, 1981; revision received Jan. 4, 1982. Copyright © American Institute of Aeronautics and Astronautics, Inc., 1982. All rights reserved.

*Senior Engineer, Payload Integration and Mission Operations Directorate.