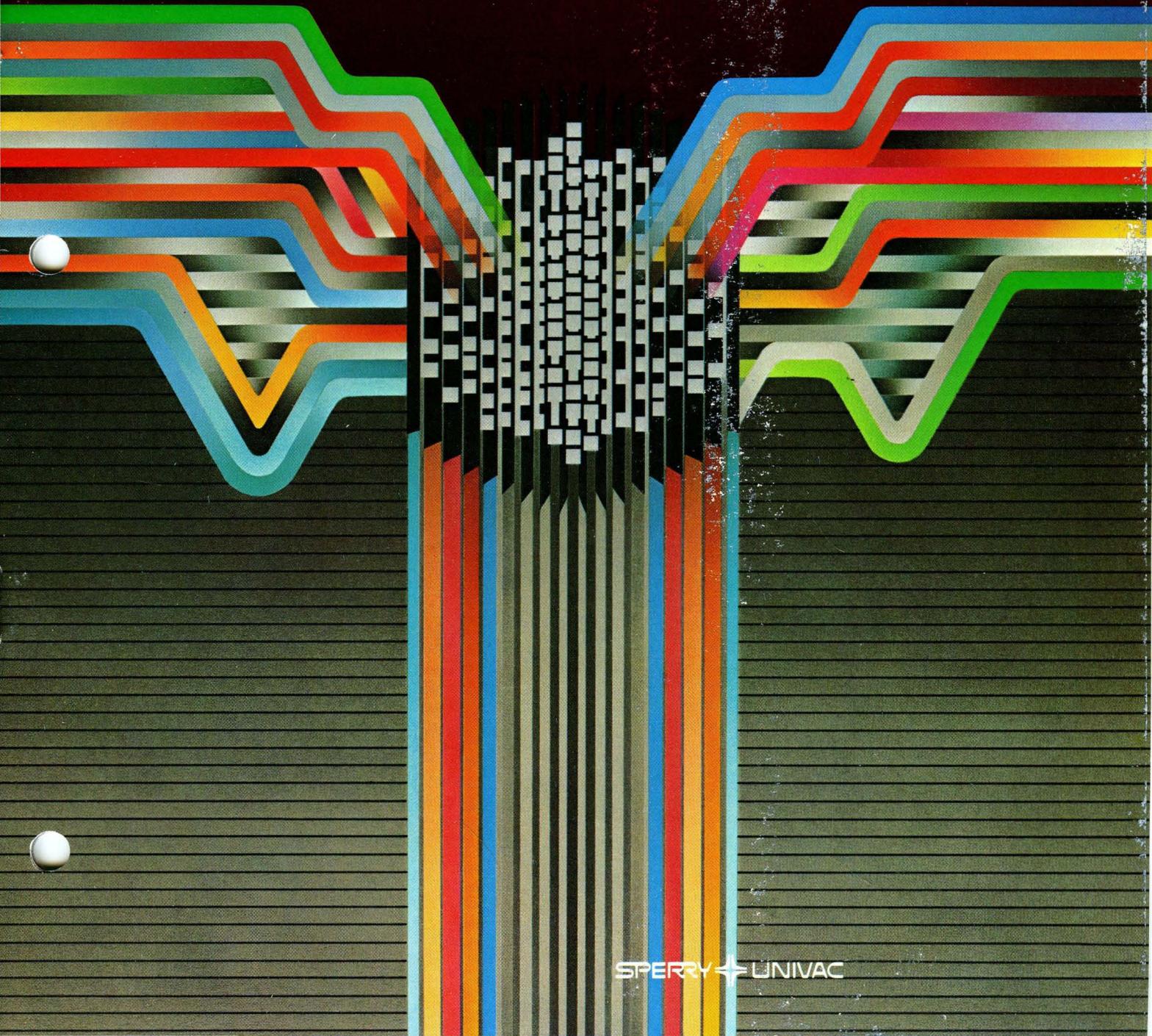
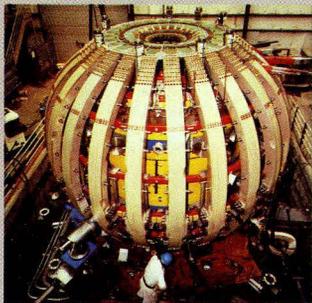


SPERRY UNIVAC
Series 1100 and
Floating Point Systems





Today, scientists, engineers and managers in a growing number of disciplines are turning to Array Processors for their high-speed calculations in numerically-intensive computing applications.

Both end users and the customers of systems integrators are discovering that Floating Point Systems (FPS) Inc.'s Array Processors, when attached to minicomputers or large, general-purpose mainframes, can dramatically increase numerical computing speeds at significantly lower costs per calculation.

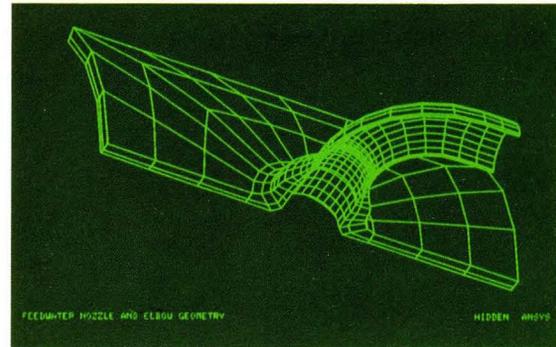
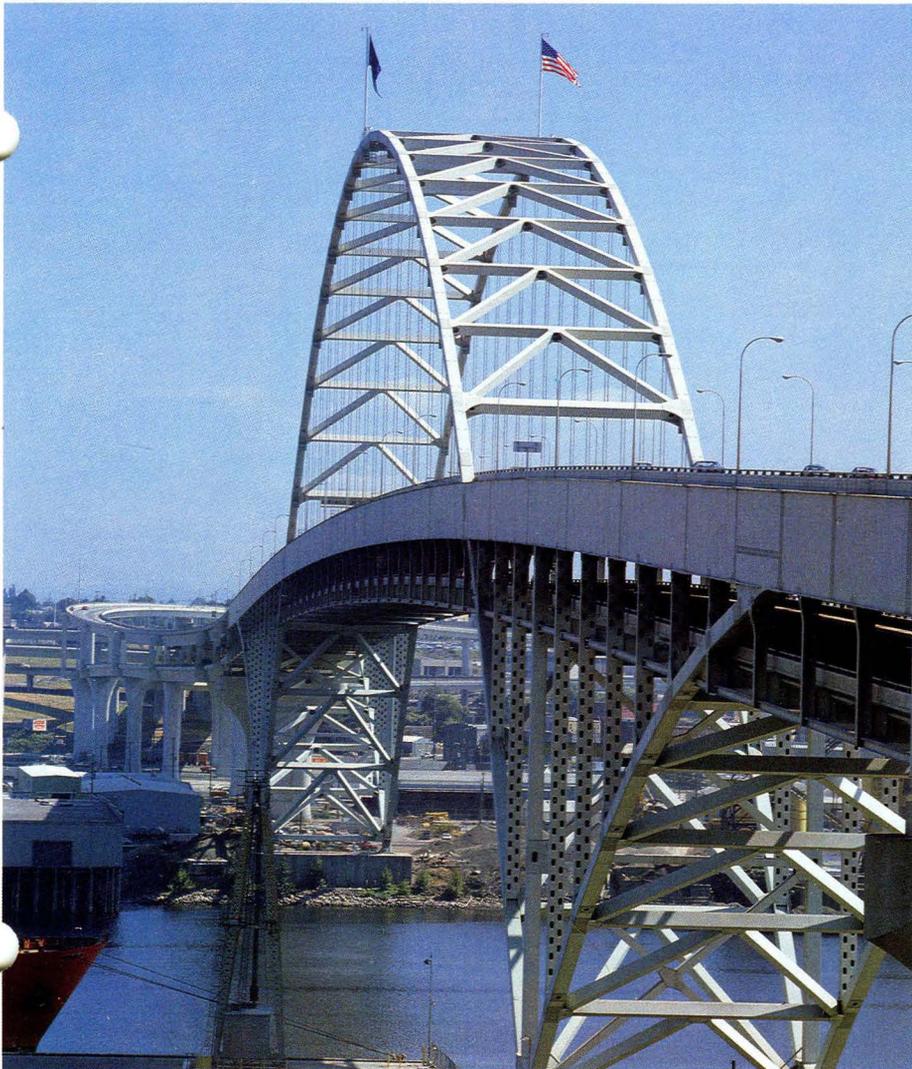
FPS Array Processors have become vitally important components of many scientific and engineering computing systems because they complement

the capabilities of general-purpose computers. Attached to the I/O channel of a host computer, they provide specialized high-speed numerical computing with 38- or 64-bit precision, with main memory sizes that range from 8K to 448K and from 64K words to 14 Megabytes, respectively, and with efficient and flexible programming, plus superior reliability.

Sperry Univac supports interfaces between the Series 1100 computers and two Floating Point System Inc.'s Array Processors, the AP-190L and the FPS-164. The AP-190L, a 38-bit Array Processor, is intended primarily for the Geoprocessor environment, encompassing geophysical exploration and production. The

FPS-164, whose data word size is 64-bits, is for the general scientific user community, supporting applications in areas such as structural analysis, numerical weather prediction and plasma physics.

Together, the FPS and SPERRY UNIVAC systems can solve simultaneous equations; multiply, normalize and invert matrices; perform signal, seismic and image processing operations like single and multi-dimensional Fast Fourier Transforms, digital filtering and convolutions; generate tabular functions by executing numerically intensive interpolation algorithms—to name but a few of the many calculations that are possible.



Arithmetic Precision and Dynamic Range

Precision and dynamic range define the accuracy and size limits of the numbers an array processor utilizes without introducing truncation and rounding errors. To optimize precision and dynamic range, Floating Point Systems employs binary-exponent floating-point data formats.

The AP-190L Array Processor uses a 38-bit format with a 28-bit mantissa (this determines the precision of the results) and a 10-bit exponent (the exponent determines the range of data values that can be processed). The FPS-164, used in higher-precision scientific computing, has a 64-bit format with 53-bit mantissa and 11-bit exponent.

The 38-bit format, with eight decimal digits of precision over the range of data values between $10^{** - 155}$ and $10^{** + 153}$, provides about two decimal digits more precision than is provided by the usual 32-bit floating-point format.

The 64-bit format provides 15 decimal digits over the range of data values between $10^{** - 308}$ and $10^{** + 307}$ —enough precision and dynamic range to execute most scientific calculations now being performed.

Speed

The AP-190L and FPS-164 are both nominally specified at up to 12-million floating-point operations per second (MFLOPS) maximum.

When these high speeds and moderate costs of FPS Array Processors are compared to the rest of the computing industry, you will find that Floating Point Systems cost-to-speed ratios are among the most impressive to be found anywhere.

Operational Flexibility

A special multi-bus architecture features a high level of parallelism, with seven independent data paths providing a separate bus for each floating-point arithmetic input, and a separate output bus from each arithmetic element to the other elements of the system.

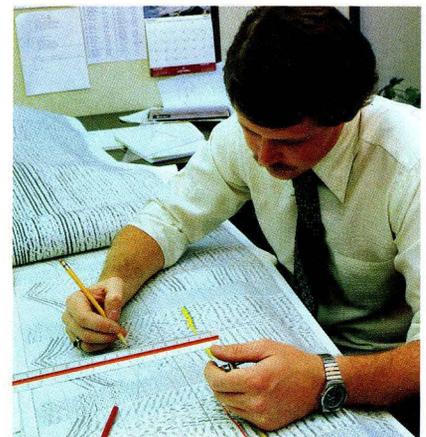
This parallelism allows for great flexibility: Multiple operands and resultants can be moved simultaneously from any element to any other. This minimizes access conflicts, and allows the FPS Array Processor to initiate multiple

commands within each cycle. As a result, specialized algorithms such as FFTs, matrix inversions and other repetitive calculations are processed at very high speeds, with FPS Array Processors also readily suited to less highly-organized computations.

Memory

Floating Point Systems offers array processors with large main data memory sizes that range from 8K to 448K words for the 38-bit processors, and from 64K words to 14 Megabytes for the 64-bit FPS-164. These sizes reduce the need for additional host memory, and increase the size of the largest calculation the array processor can perform as a single task, unlimited by disk speed.

Large array processor memory also minimizes programming complexity, because the host and array processor need perform fewer extra I/O operations to transfer data and results, and the program the array processor executes does not have to be broken up into separate overlays. This, in turn, speeds task executions. The lessened I/O time and programming effort further decrease total calculation cost.



SPERRY UNIVAC Interfaces

The AP-190L and FPS-164 can be interfaced to an 1100/60 or 1100/80 system. The AP-190L is attached to the SPERRY UNIVAC processor via a Multi-User Array Processor Handler (MUAPH). The MUAPH is designed to provide two to three times the throughput processing capability available with the standard FPS-supplied handler. MUAPH was originally designed by Digitech Systems, Ltd., a Geophysical Service Co. and a Sperry Univac user in Calgary, Alberta, Canada in cooperation with Sperry Univac and Floating Point Systems.

The FPS-164 is interfaced to the Sperry Univac host via a Multi-User Monitor (MUM). Like MUAPH, MUM increases throughput over the standard FPS interface (Single User Monitor or SUM) by allowing many users to share the Array Processor.

Programming Flexibility

Efficient and flexible use of FPS Array Processors is assured first through comprehensive software: APEX, for host/array processor transfer and data control operations; Program Development software including the FPS FORTRAN Compiler for running existing software on the array processor; Math, Signal, Image, and Seismic Processing, and Simulation Libraries of preprogrammed application routines.

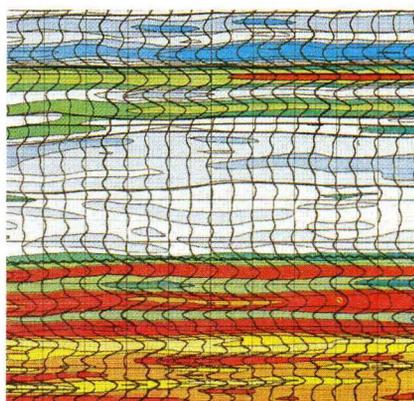
To further enhance the attractiveness of array processing in a FORTRAN oriented environment, SPERRY UNIVAC also offers an automatic vectorization capability. This facility called VAST™ (Vector to Array Syntax Translator) accepts an ordinary FORTRAN program as input, and determines, without user intervention, what computational sequences in the program (DO Loops) can be executed in vector mode on the array processor. Appropriate DO Loops are then automatically replaced by calls to microcoded array processor functions.

Reliability

FPS Array Processors have established an impressive reliability record. The average mean-time between failures (MTBF) for the AP-190L Array Processor based on customer service calls is well over 3,500 hours. High reliability reduces down-time as well as the risk that long calculations will be interrupted by hardware failures that invalidate results.

Manufacturing builds in reliability, starting with preconditioning and testing of individual integrated circuits. Testing continues at the board, subsystem, and system level all through the manufacturing cycle.

In addition, the FPS-164 provides state-of-the-art reliability and maintainability features. Error-correcting memory, internal diagnostic buses, and an internal diagnostic microprocessor (DMP) with floppy disk, all help to maximize up-time.



Configurability

Sperry Univac users can take advantage of the cost-effectiveness of FPS Array Processors whether they're planning a new installation, upgrading an existing one, or integrating a composite system for those customers who have specialized computing needs.

In new installations, an FPS Array Processor can often perform all the necessary computing at much less cost, even in those high-speed, high-precision applications that normally call for large scientific computers. Multiple FPS-190Ls or FPS-164s can be attached to an 1100 system. Only one copy of MUAPH (FPS-190L) or MUM (FPS-164) is required regardless of how many units are attached.

When upgrading a system to perform more numerically intensive applications, adding an FPS Array Processor can be much less expensive than buying another general-purpose CPU. Because the array processor attaches to a standard host computer I/O port, it requires little or no additional host hardware to integrate it into an existing system.

The FPS FORTRAN Compiler and Program Development Library, VAST, MUAPH, and MUM, also facilitate interfacing. An FPS Array Processor thus provides a way for the customer to increase system power without installing more complex hardware and software communication links.

FPS Array Processors fit quite well into packaged computer systems designed for specific single customers or customer groups with specialized applications. Many such customers require hardware that performs numerical calculations very rapidly, at a reasonable cost.

AP-190L Key Specifications

The AP-190L is a 38-bit array processor which attaches to mainframes.

Nominal Speed: 12,000,000 floating-point operations per second maximum (12 MFLOPS)

Word Size: 38-bit

Precision: 8 decimal digits

Dynamic Range: 10^{-155} to 10^{+153}

Memory Size

Main Data Memory: 8, 16, 32 or 64K 38-bit words: additional memory (optional) available in increments of 64K words, up to 448K words

Program Memory: 2K or 4K 64-bit program words

Table Memory: 2.5K or 4.5K words of ROM stores trigonometric and transcendental functions: 1K increments to 4K data words of RAM optionally available to store user-defined tables

Size: 31"L × 23.5"W × 61.25"H cabinet (78.8 × 59.8 × 155.8 cm)

FPS-164 Key Specifications

The FPS-164 is a large-memory, 64-bit scientific attached processor. It performs very high-precision scientific calculations at a cost that compares favorably with those of all other currently available processors.

Nominal Speed: 12,000,000 floating-point operations per second maximum (12 MFLOPS)

Word Size: 64-bit

Precision: 15 decimal digits

Dynamic Range: 10^{-308} to 10^{+307}

Memory Size

Main Data Memory: 64K data words optionally expandable to 128K, 256K, 512K, 1024K or 1856K data words

Program Memory: A four-region 1K word program cache controls transfer of 64-bit program words from main memory

Table Memory: 5.5K words of ROM for frequently used constants; 16K or 32K words of optional RAM available to store user-defined tables

Size: 64.5"H × 30"D × 83.5"W cabinet (164 × 76 × 212 cm)

Specifications subject to change without notice.

Programming ease and flexibility are enhanced by a comprehensive software library that includes over 450 routines for host/array processor communications and control, specific mathematical and engineering applications, and program development. The latter includes the FPS FORTRAN Compiler and Assembler, which allow users to create their own programs, a Linker and Debugger to debug programs in an actual or simulated array processor environment, and a Library Editor for library maintenance.

The Array Processor EXecutive (APEX) is a collection of host-resident run-time routines that controls the array processor and manages I/O activity between host computer and array processor. The FPS-164 also features the Multi-User Monitor (MUM) software. Residing in the array processor, MUM works with APEX to direct the flow of work through the array processor. MUM allows the FPS-164 to operate in a multiprogramming environment, allowing the overlap of I/O and CALL portions of different user jobs.

The AP-190L makes use of the Multi-User Array Processor Handler (MUAPH). With the multi-user hardware, Sperry Univac provides the first seismic processing system in the industry that can take full advantage of the high speed vector processing of the AP-190L. With this system data transfer, job set-up, and vector processing can take place concurrently by several users, thus effectively eliminating the channel conflict that prevents other systems from efficiently using the array processor. Extensive statistics and accounting capabilities are added to enable users to observe, analyze, and optimize system performance. With the multi-user handler, the 1100 system requires significantly fewer array processors to perform the same work as competitive systems.



For optimum programming ease, users may select from among the following preprogrammed applications routines:

The Standard Math Library features basic mathematical operations common to most array processing programs, as well as many supplemental routines.

The Advanced Math Library complements the Standard Library by providing for those applications with more advanced and complex mathematical requirements. Subroutine categories include: Integration, multivariable function generation, eigenvalues/eigenvectors, and more.

The Signal Processing Library includes subroutines used in signal processing application environments, such as: Histogram analysis, window functions, auto- and cross-spectrum operations, spectral averaging, correlation and convolution, filtering functions and spectral transforms.

Computational tools to filter and enhance images are provided by the Image Processing Library. It includes such routines as filtering functions, two-dimensional FFT's and image movement.

Geophysicists and data analysts in seismic exploration may use the Geophysical Processing Library—a menu of application routines that aids in geophysical computations. Some of these are: normal moveout routines, quadratic interpolation, recursive filter and rasterization.

For computationally intensive real-time mathematical modeling applications, users may select the Simulation Library (SIMLIB) available for both the AP-190L and the FPS-164.

Alternatively, users may program the array processor themselves, using the FPS FORTRAN Compiler. This enables an applications programmer to significantly increase processing throughput by running existing FORTRAN software on the array processor (often without recoding), and to write new software without learning a new programming language. This FORTRAN option is especially attractive on a Sperry Univac/FPS system due to the availability of VAST, the automatic vectorization facility. VAST allows the FORTRAN user to realize performance gains, while maintaining code transportability and development ease.

An assembler lets an applications programmer write a subroutine in Array Processor Assembly Language (APAL) that can run at an FPS Array Processor's maximum execution rate. Specifically, APAL allows this programmer to take full advantage of the parallel pipelined architecture of the FPS Array Processors, thus achieving the level of parallelism required for optimum efficiency. A programmer who can code in a mainframe or minicomputer assembly language can readily learn to write efficient APAL code.

In some cases, where a FORTRAN inner loop is limiting processing

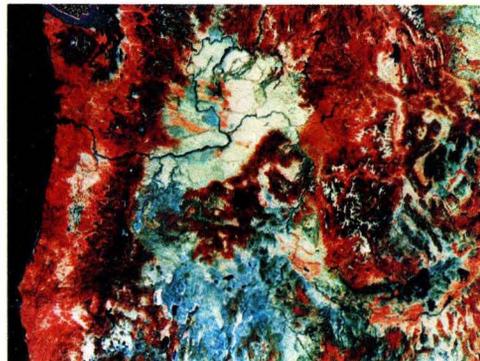
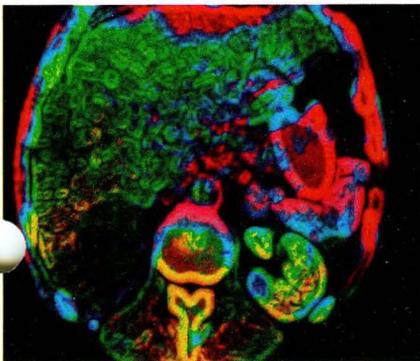
speed, coding the inner loop in APAL can achieve a significant improvement in throughput.

The Linker combines FORTRAN and APAL object modules together with referenced application library modules into a binary load module for execution on an FPS Array Processor. For each host-callable routine the Linker builds a Host-Array Processor Software Interface (HASI), a program which is compiled and linked with the host-resident software.

The HASI corresponding to a given array processor-resident routine takes care of all array processor synchronization as well as data and program transfers, and thus allows the Host/Array Processor partition to be invisible to the program logic. The HASI is called in the same way that the routine would be if it were host-resident.

The Array Processor Debugger (APDEBUG), a host-resident program, supports a set of commands that allows the user to interact with the control and execution of a program on the array processor, from the user's terminal. APDEBUG permits single-step operation.

On the FPS-164, APDEBUG supports full symbolic debugging of both FORTRAN and APAL programs. As both of the products are available as program libraries on the FPS-164, a user may execute/debug programs in the full host-array processor environment.



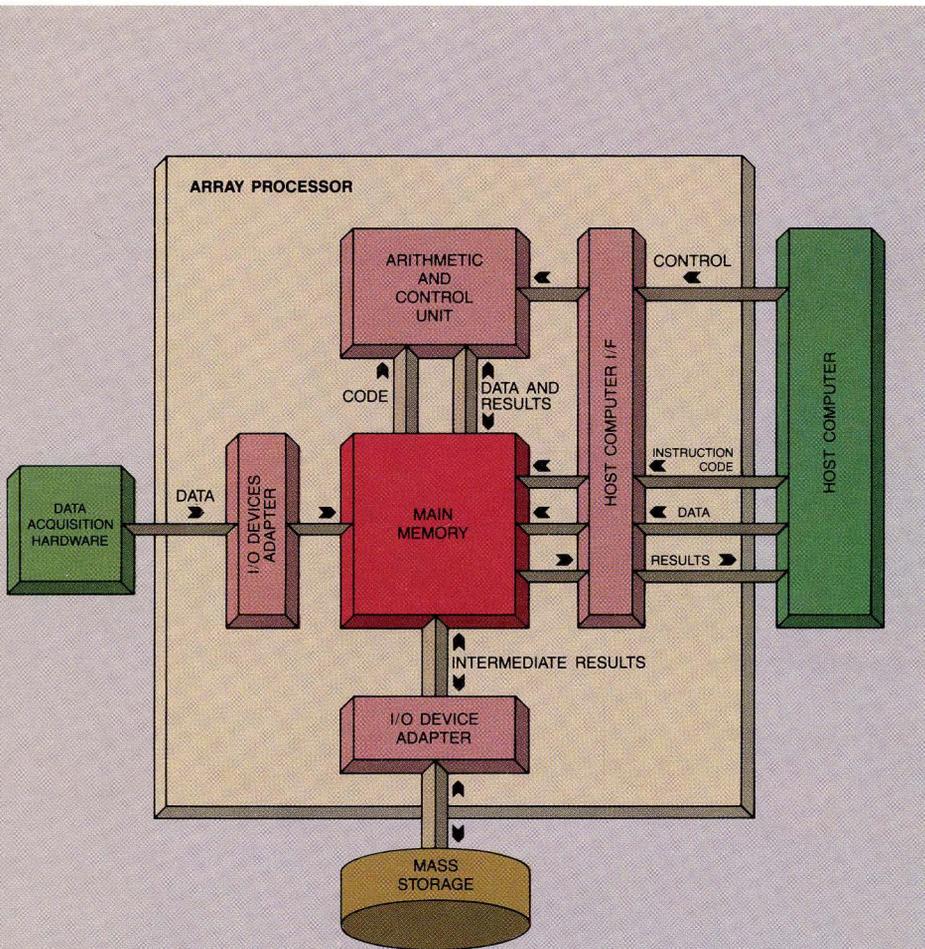


Figure 1. Host/Array Processor Systems

An FPS Array Processor connects to the Input/Output (I/O) channel of a host computer (Figure 1). Using the I/O channel, the host sends data to the array processor, which in turn processes the data and sends the results back to the host.

An FPS Array Processor may communicate with the host through the Host/Array Processor Software Interface (HASI) subroutine which resides in the host computer, and with an FPS library of host-resident routines called the Array Processor EXecutive (APEX). These transfer program code and data to the array processor, inform the host program when the array processor has finished a task, and return results to the host (Figure 2).

Here, in brief, is how the process works:

The host FORTRAN program calls the HASI, which in turn calls an APEX routine to transfer instructions and data to the array processor, and then continues executing its portion of the task. Later, it calls another APEX routine to verify completion of the transfer, and then instructs the array processor to start execution of the task. The host program once again resumes work on its portion of the task.

After that, the host program calls APEX routines to verify completion of the array processor task, to return results to the host, and to ascertain that the results have been returned. At that point the host program can perform any processing that depends on those results.

Using APEX calls, the host program can also transfer data for a second task to the array processor while the array processor is completing the first task. The APEX library thus gives the host/AP system a multitasking capability that results in extremely efficient use of the system's hardware resources.

The Sperry Univac/FPS interfaces, MUAPH and MUM, are unique in providing a multiprogramming environment for the AP-190L and

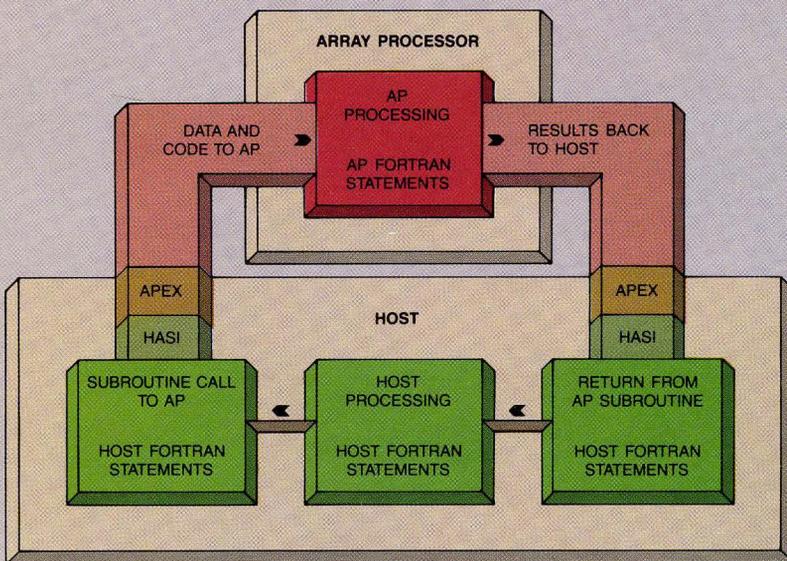


Figure 2. Processing Flow for Single User

FPS-164. This availability to overlap computation and data transfer for many users results in an increase in efficiency over and above that observed in other FPS computing environments.

Data acquisition hardware can use Direct Memory Access (DMA) to send data directly to array processor main memory. Signal processing systems often send data to the host through the array processor because the array processor can usually preprocess signal data much faster than the host.

The host can also use DMA to send preprocessed data to the array processor via the host/array processor interface. DMA allows both the array processor to compute and transfer data at the same time.

The array processor can also transfer data to and from direct access mass storage devices to store arrays that are too large for its main memory. These capabilities allow the array processor to store results and process large arrays without consuming additional I/O time on the host.

The Array Processor In Action

This sample calculation, chosen for its simplicity and brevity, indicates only one of the computational possibilities of an FPS Array Processor. As described by the FORTRAN source code used in FPS Array Processors, it is an excellent example of an application that maximizes the array processors cost-effective capability. The following FORTRAN source code performs a matrix multiplication:

```
DO 300 I=1,N
DO 200 J=1,N
C(I,J)=0.0
DO 100 K=1,N
C(I,J)=C(I,J)+A(I,K)*B(K,J)
100 CONTINUE
200 CONTINUE
300 CONTINUE
```

From this, three salient observations may be made:

First, if N is 100 or more, the code calls for a least a million floating-point multiplications and additions.

Second, the data points form a well-ordered array.

Third, the computer does not have to change a computing algorithm in response to interrupts or singularities in the data.

During this calculation, the host computer can be performing other tasks (multi-tasking) while the array processor makes the calculation, and the array processor can be overlapping. With overlapping, data are transferred for the next calculation before the array processor is finished processing the preceding data, and results of the first calculation are sent back while the array processor is performing the second one. Under these circumstances, both the host and array processor will then be operating at maximum efficiency (Figure 3).

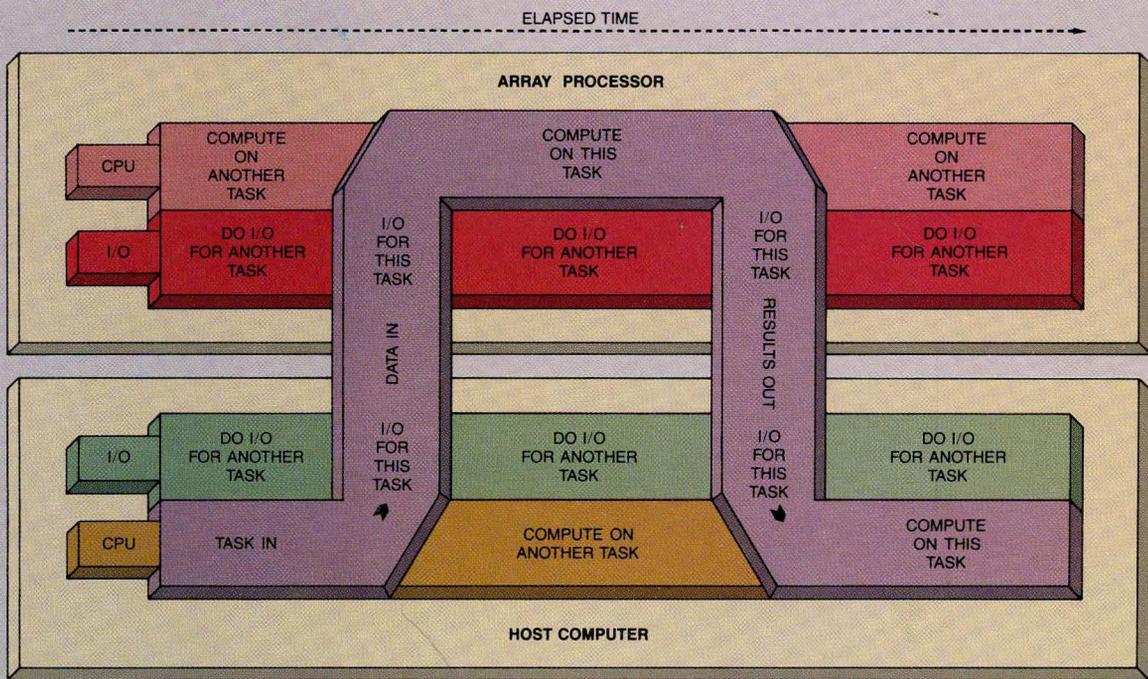


Figure 3. Multiple Processing

Inside The Array Processor

A block diagram of an FPS Array Processor (Figure 4) reveals multiple arithmetic, register, and memory units connected by parallel data paths. This arrangement allows FPS Array Processors to combine as many as 10 minor operations into one machine instruction.

The architecture of an FPS Array Processor incorporates hardware pipelining, a feature that enables it to operate several stages of computation simultaneously (paralleling operations in time). Each stage executes one of the operations the pipeline must perform to complete its task. At the end of each clock cycle, each stage of the pipeline passes a result on to the next stage. The initial stages accept new operands, and the final stages store them or pass them on to other functional units.

The code on the next page shows how this technique reduces the execution time of a vector/scalar multiply-add calculation of the form $Y_i = A \cdot X_i + Y_i$ for i from 1 to N .

The array processors multiple functional units allow it to compress the seven sequential instructions that consume 17 time units (cycles) on a sequential computer (Figure 5) into an overlapped pattern that takes only nine cycles to process (Figure 6). Specifically, the fetch of Y_i can take place in parallel with the multiply. The loop branch can take place in parallel with the store.

Because each iteration of the loop is independent of the next, the computation can be further reduced to only three cycles per loop. This is called pipelining, where in this case three separate loop computations are done in parallel (Figure 7).

At the beginning of the calculation, $X(1)$ and $Y(1)$ enter the nine-cycle pipeline formed by the overlapped microcode sequence. Three cycles later, $X(2)$ and $Y(2)$ enter the pipeline. And finally, after another three cycles, $X(3)$ and $Y(3)$ enter and fill the pipeline. From then on new data elements enter and leave the pipeline every three cycles until the pipeline has processed all the data.

The complete block of code contains a six-cycle startup sequence that fills the pipeline. The parallelogram shape of the code block is typical of the form of most pipelined processing loops. A complete program would also include short auxiliary sections that bypass the main loop to process one- and two-element vectors.

The net result is a processing rate nearly six times higher than that provided by the sequential instructions when operating on large vectors of data.

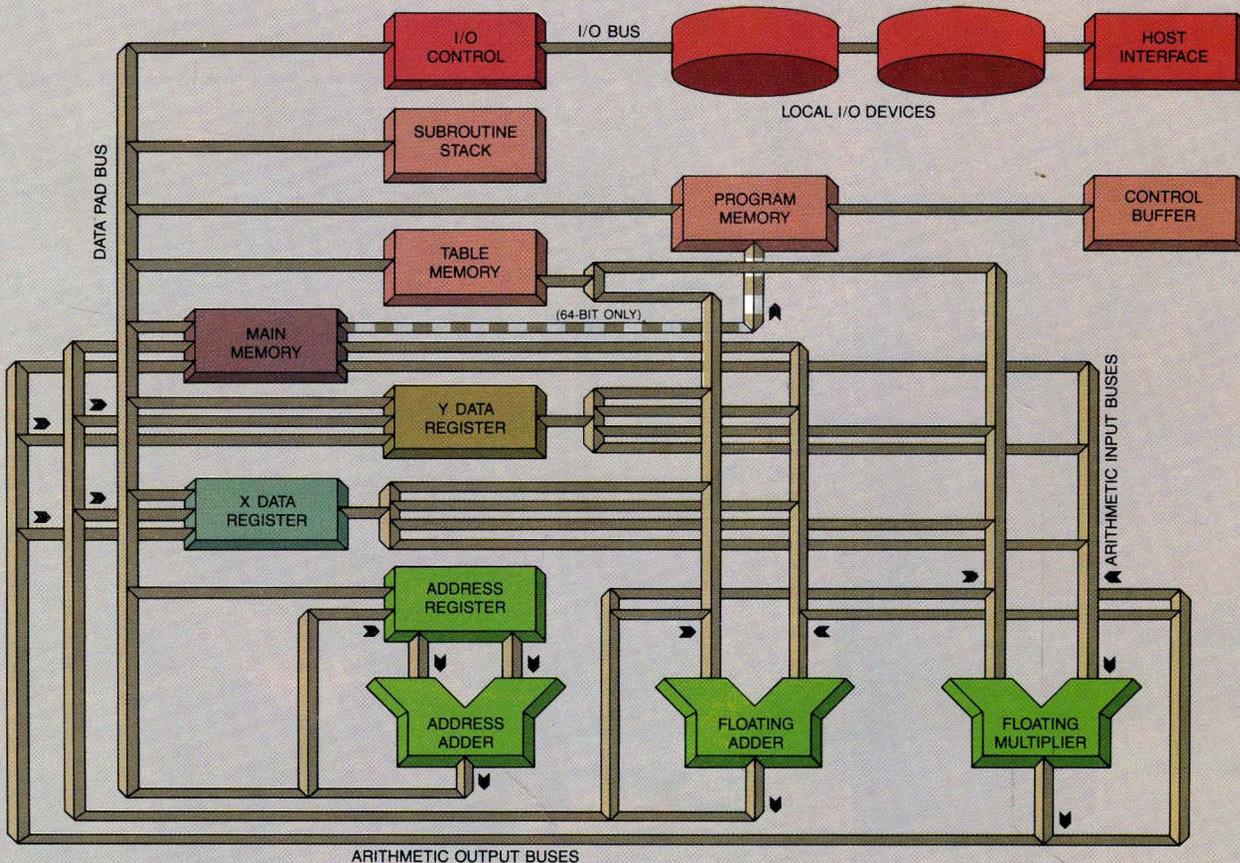


Figure 4. Array Processor Block Diagram

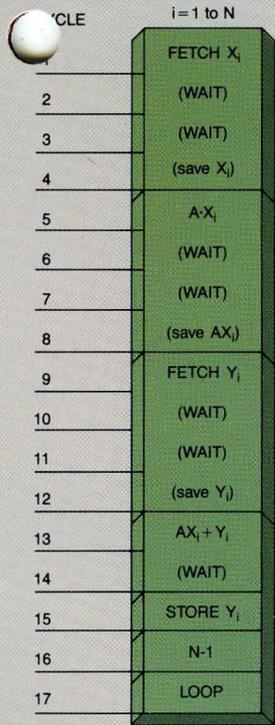


Figure 5. Sequential Computation

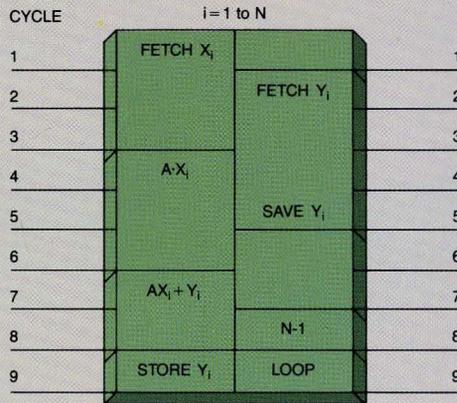


Figure 6. Overlapped Computation

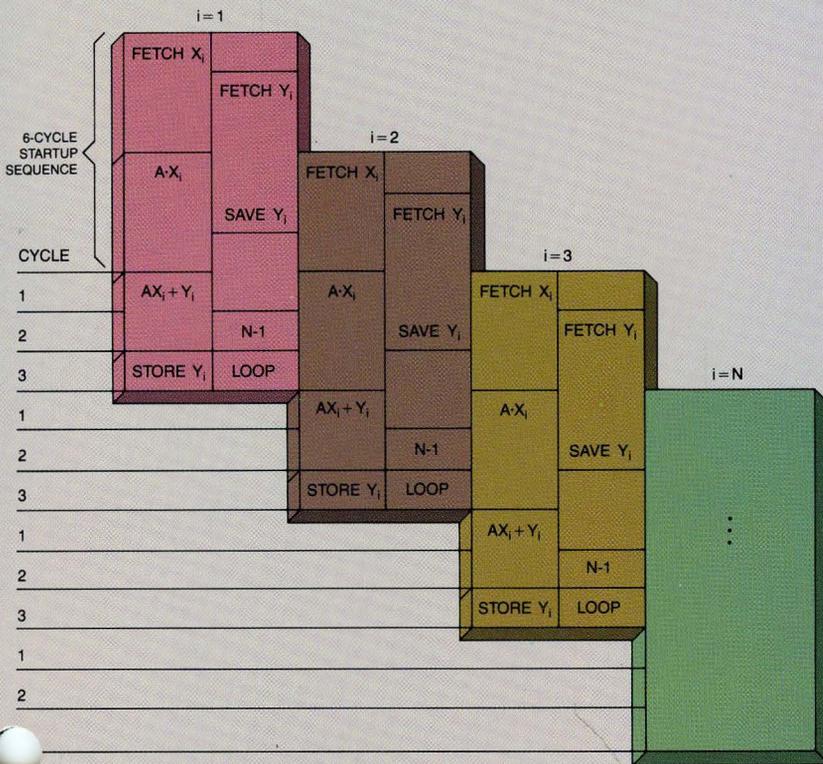


Figure 7. Pipelined Computation

Sperry Univac is a worldwide supplier of computer products with the second largest installed computer base in the business.

Our computer systems contain a number of diagnostics for both software and hardware. Corrective action and fault isolation are undertaken immediately when the system encounters problems. We also have a worldwide customer service organization ready to serve you.

For more information about SPERRY UNIVAC computers and Floating Point Systems Array Processors, contact your local Sperry Univac representative.

SPERRY  UNIVAC
The computer people who listen.