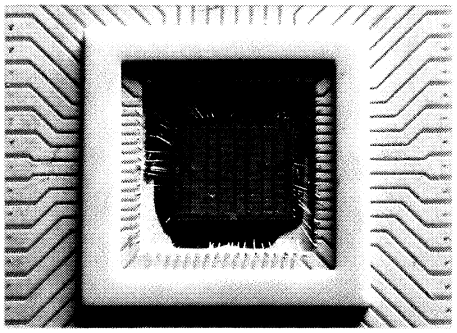
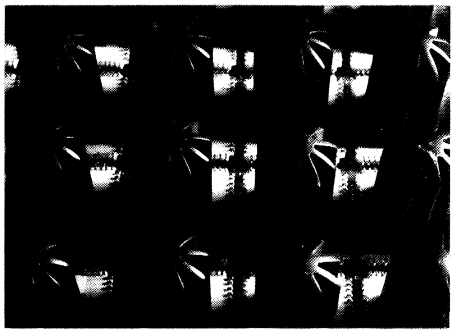
Star ST-100

ST-100

THE 100 MEGAFLOP ARRAY PROCESSOR

STAR TECHNOLOGIES, INC.





Star Technologies, Inc., a manufacturer of scientific array processors, is dedicated to meeting users' demands for higher speeds and enhanced systems throughput at lower processing costs!

The performance of array processors has not increased as rapidly as general scientific computers. No **significant** performance improvement has been made in either architecture or technology, **until now!**

A new generation of array processors is now available. By using VLSI technology and an advanced modular architecture, Star Technologies has developed an array processor that competes with super computers at a fraction of their cost.

Star Technologies management and technical staff are computer professionals with extensive array processor and scientific computer backgrounds in a variety of application areas. Our development facilities are in Minneapolis, Minnesota and our corporate office is in Portland, Oregon. Sales and support service offices serve Star Technologies customers worldwide.

Recent advances in computer aided design, VLSI and packaging technology have made possible a new generation of computer equipment. The advantage of this new technology is a dramatic increase in performance with moderate increase in cost. Star Technologies has utilized this advanced technology to build the ST-100. The ST-100 is constructed with 1200 gate bipolar VLSI circuits. These high density circuits have subnanosecond gate delays. Compared with conventional gate arrays, this logic yields substantial improvements in performance and greater utilization of on-chip components, thus permitting lower power dissipation.

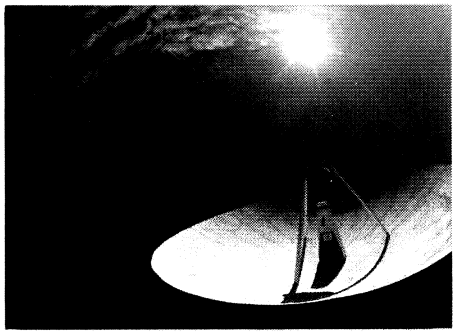
The availability of modern block level and gate level simulation permitted the thorough testing of VLSI circuits. Access to computer aided design routing and circuit placement software enabled the design of complex multilayered VLSI circuit boards.

Coupled with this advanced VLSI design technology, unique cooling and packaging concepts were developed. A cooling system to eliminate temperature gradients across circuit boards ensures the cooling of each individual circuit.

The ST-100 utilizes a unique packaging concept, integrating small modules of 64K bit dynamic random access memories into a large multi-million byte memory.

Dramatic improvements in hardware were not enough to ensure efficient system utilization. Today's software technology permits the sharing of multiple resources between multiple programs within a network of host computers. Star Technologies has used this advanced technology in the development of its ST-100 software system.

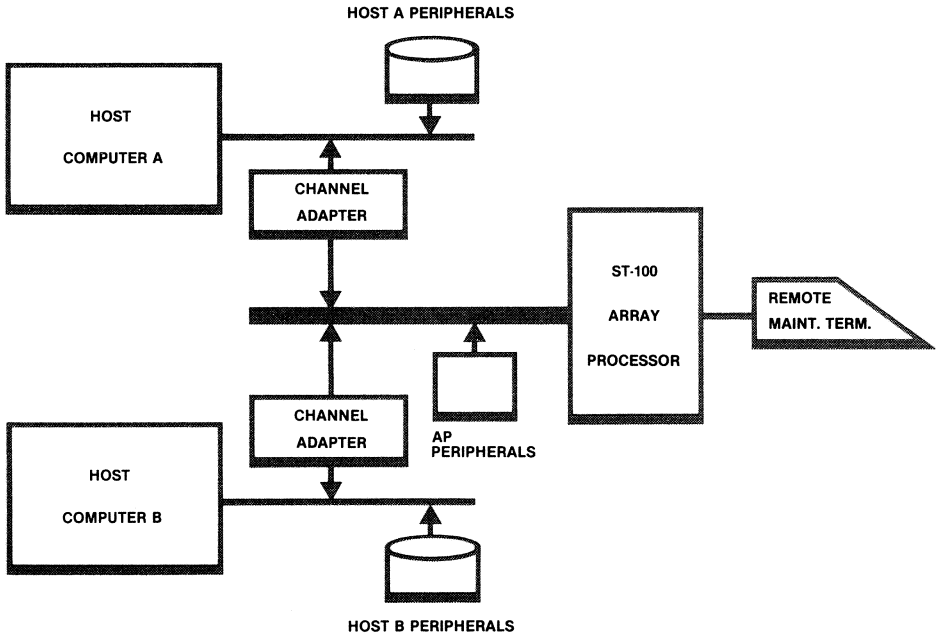
The end result of this technology and simulation capability is a 100 Megaflop Array Processor. **You be the judge – the ST-100 – designed with the user in mind.**

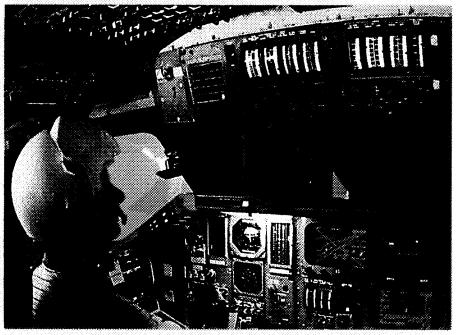


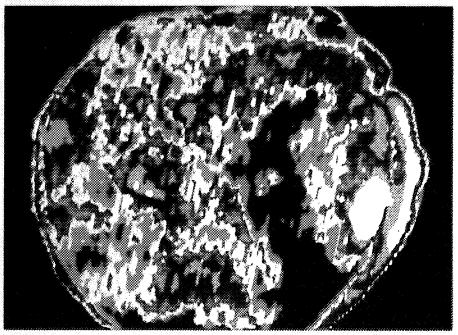
- Advanced VLSI technology
- Synchronous parallel, pipelined architecture
- Speed:
  - **40-nsec** machine cycle
  - **100 million floating point operations** per second (Megaflops)
  - 480-nsec hardware divide/ square root
  - 40-nsec integer, logical/ Boolean, or shift operations
- 32-bit word size
- Memory size:
  - Main memory up to **8 million words** (in 512K word increments)
  - 32 million bytes
  - Directly addressable to **1-Gigaword** (byte, half word, or word)
- Random Access data cache:
  - **48K words** in six 8192 word sections
- High speed input/output:
  - I/O channel supports up to 7 device adapters; 12.5 Mbyte/sec. data transfer rate
  - DMA for special devices; **100 Mbyte/sec.** data transfer rate
- Programmable hardware format conversion
- Multiple host interfaces
- Ambient air-cooled
- Remote maintenance capability

- Development Software
  - **FORTRAN-like** control language (ease of use)
  - macro assembler (simplifies microcoding)
  - library editor
  - simulator/debugger
- Production Software
  - **application library modules**
  - resource-sharing executive (supports multiple hosts connected with multiple array processors)
  - multi-user capability
- Maintenance Software
  - off-line and remote diagnostic capability
  - idle loop reliability tests
  - user confidence tests

SYSTEM BLOCK DIAGRAM







Unique architectural features combined with innovative packaging permits a level of performance previously unavailable to planners of medium sized systems.

The **ST-100** consists of multiple processors and memories interconnected to allow **independent data flow and arithmetic processing.**

Specially constructed processors enable a MIMD (Multiple Instruction-Multiple Data) architecture to solve problems in parallel.

The **ST-100** has four independent programmable processors. A separate processor is dedicated to each of the following functions: external data flow, internal data flow, arithmetic processing, and synchronization. A hierarchical memory system consists of external storage devices, a large main memory, a high speed random access partitioned data cache, and universal register set.

The **control processor** stores microprograms and parameters into the following processors' memories providing synchronization of their concurrent operation.

The **arithmetic processor** manages data flow between the data cache, the data interchange register set, and the arithmetic elements.

The **storage/move processor** manages movement of data between the main memory and the data cache.

The **input/output processor** manages data flow between external devices (disks, host memories, etc.) and the main memory via a device adapter and internal channel.

CONTROL PROCESSOR

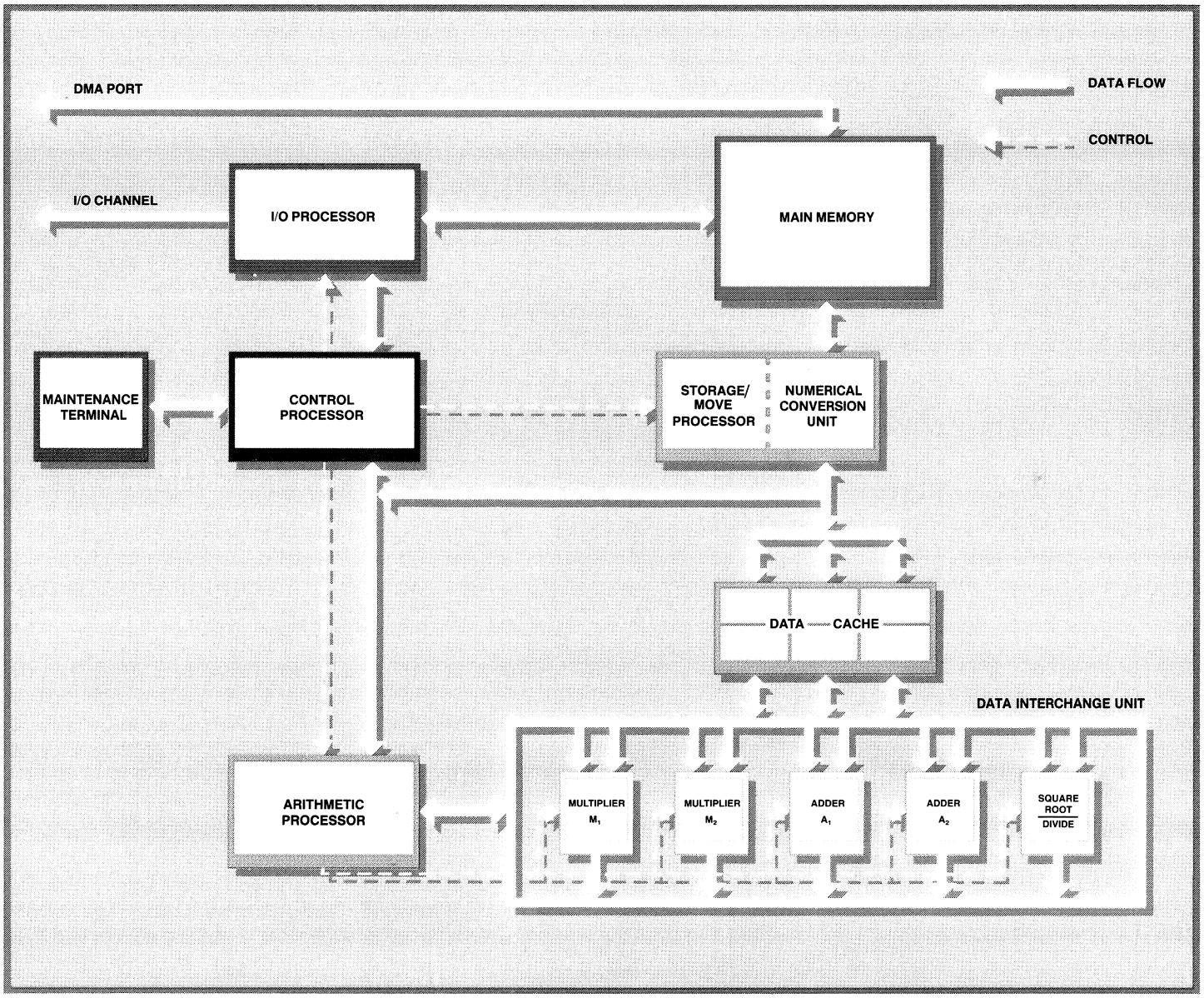
The control processor is an advanced microcomputer which manages resource allocation and data flow. This processor contains its own control and data memory, general purpose instruction set, hardware interrupt capability, interval timer, and direct memory access to the other three microprogrammable processors. The control processor is interconnected with the other three processors to manage microprograms and exchange control parameters. It initiates and monitors the other processors to synchronize their concurrent operations. This processor also manages overlapping data transfers and stages subsequent user jobs for execution without interferring with arithmetic processing. A maintenance terminal connects directly to the control processor via a local or remote communications link for diagnostics.

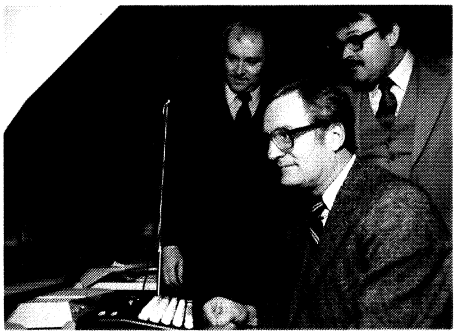
ARITHMETIC PROCESSOR

The microprogrammable arithmetic processor consists of a control unit, arithmetic elements, and a data interchange unit. The control unit generates cache addresses, selects arithmetic operands and operations, performs loop count control, and controls a parameter stack. The arithmetic elements comprise **two add/subtract units, two multiplier units,** and a **divide/square root** unit. The data interchange unit permits one of 16 operands to be selected for each arithmetic input register. During each machine cycle, three cache banks may be referenced, one loop control operation computed, four arithmetic operations started, and a conditional branch executed. This processor computes at a rate of **100 million 32-bit floating point operations per second.**

STORAGE/MOVE PROCESSOR

The microprogrammable storage/move processor initiates and controls data flow between the main memory and the data cache. Transfers and data format conversion can take place at full memory bandwidth (100 Mbytes/second) through the numerical conversion unit. Main memory addresses, cache memory addresses, and loop control are calculated at full memory bandwidth. The storage/move processor allows complex addressing sequences to be computed concurrently with data transfers. **Integer, shift, Boolean and logical 32-bit operations can be performed** on data from main memory or data cache.


DMA PORT
DATA FLOW
CONTROL
I/O CHANNEL
I/O PROCESSOR
MAIN MEMORY
MAINTENANCE TERMINAL
CONTROL PROCESSOR
STORAGE/ MOVE PROCESSOR
NUMERICAL CONVERSION UNIT
DATA CACHE
DATA INTERCHANGE UNIT
ARITHMETIC PROCESSOR
MULTIPLIER M1
MULTIPLIER M2
ADDER A1
ADDER A2
SQUARE ROOT / DIVIDE



INPUT/OUTPUT PROCESSOR

The input/output processor controls a 25 Mbyte channel and a 100 Mbyte Direct Memory Access (DMA) port. The 25 Mbyte channel supports seven device adapters. Device adapters connect host computers and peripherals to the array processor through the input/output processor. The device adapter can transfer at 12.5 Mbyte/second. High-performance devices can be connected directly to the Main Memory.

MAIN MEMORY

The Main Memory consists of eight 320 nsec cycle memories interleaved to provide an access rate of 40 nsec per 32 bit word. Single error correction and double error detection (SECDED) are implemented with seven additional bits. The Main Memory is **expandable to 8 million words** in increments of **512K words** by using 64K dynamic random access memory circuits. All Main Memory can be directly addressed as 32, 16 or 8 bit quantities. This memory can be partitioned and protected at multiples of 4096 words.

DATA CACHE MEMORY

The random access data cache memory consists of six banks of 8192 32-bit words with odd parity for a total of 48K words. During each machine cycle, four cache references are permitted: three by the arithmetic processor and one by the storage/move processor. These banks can be logically connected to main memory or the arithmetic elements under program control.

The ST-100 software system consists of three major segments: Development, Production, and Maintenance software.

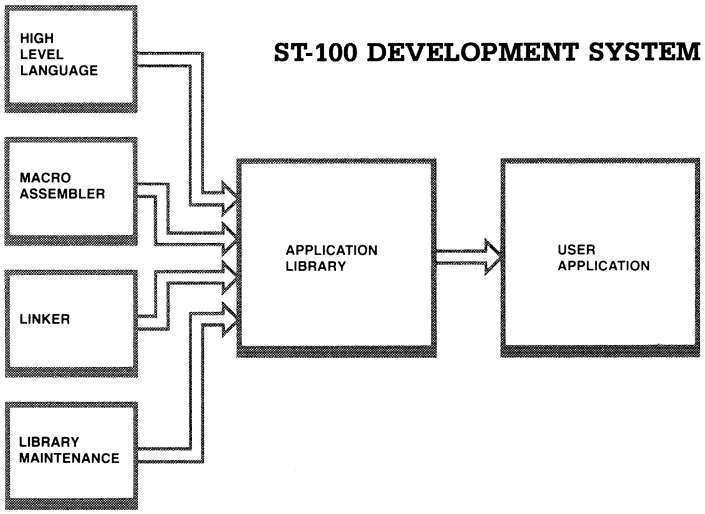
DEVELOPMENT SOFTWARE

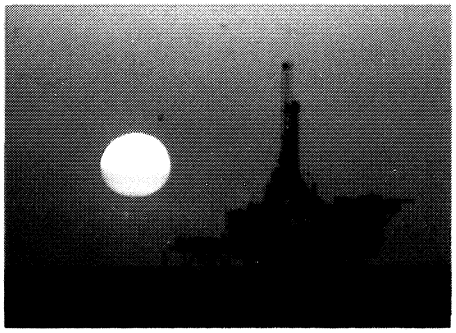
The ST-100 Development Software System is a set of FORTRAN programs residing in the host. It comprises a **High-Level control language, Macro Assemblers,** a **Simulator/Debugger,** a **Linker,** and a **Library Maintenance Program.** The Development Software System provides the ability to separately program all elements of the ST-100, allowing multiple levels of program optimization. These programs are used to create application library modules.

Application Library modules are written using the FORTRAN like high level control language. This module contains code that executes on the control processor which invokes macro calls to the arithmetic, storage/move and input/output processors. The **Macro Assembler** is used to generate these macros.

The **Simulator/Debugger** verifies the application modules and microprogrammed macros. The Linker then combines microprogrammed modules with object code created by the high-level language to produce host FORTRAN callable subroutines.

The **Library Maintenance Program** catalogues and maintains both the application library modules and the microprogrammed macro routines.





OPERATING SOFTWARE

The ST-100 Production Software system couples the array processor to the host application program. It consists of a **Host-Resident Array Processor Executive,** an **Array Processor-Resident Monitor,** and an **Application Library** that contains array processor executable code.

The **Array Processor Executive** is a set of FORTRAN subroutines that communicates with a device driver. This Executive manages the allocation of array processor resources and directs requests from the user's application program to the control processor.

The **Array Processor Monitor (APM)** schedules and controls requests from multiple users as well as from multiple hosts. The Monitor executes in the control processor.

The **Application Library** contains the programs which are executed in the array processor.

MAINTENANCE SOFTWARE

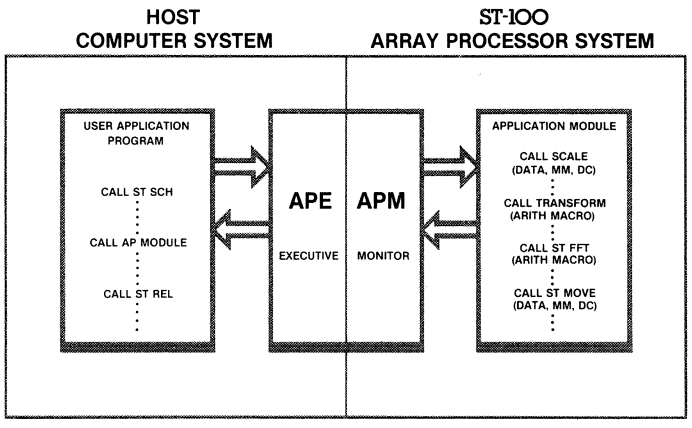
The ST-100 Maintenance Software system consists of a set of fault identification and isolation routines. These routines can be used in three ways: **as user-callable confidence library** modules; as array processor monitor **idle loop reliability** modules; and as diagnostic programs.

These **diagnostic programs** can be executed either on-line or off-line, and can be controlled either **locally** or **remotely.**

Star Technologies designed the ST-100 Array Processor to be a **total system solution.** Executing complete application programs within the ST-100 reduces unnecessary host-array processor communications. The ST-100's ability to stage multiple jobs for subsequent execution allows one array processor to serve multiple users from multiple hosts.

Staging of multiple jobs and data transfers can be overlapped without interferring with arithmetic processing. This activity is controlled by the array processor monitor.

HOST PROGRAM — ST-100 INTERFACE



STAR TECHNOLOGIES, INC.
Cory House
The Ring
Bracknell
Berks. RG12 1ES
Telephone: 0344 54471
Telex: 848204 STAR G

STAR TECHNOLOGIES WORLDWIDE

FIELD OFFICES

Southern California
Star Technologies, Inc.
24672 San Juan Ave., Suite 101
Dana Point, California 92629
(714) 661-0340

Northern California
Star Technologies, Inc.
231 Bernal Ave.
Pleasanton, California 94566
(415) 462-2481

Texas
Star Technologies, Inc.
10103 Fondren, Suite 110
Houston, Texas 77096
(713) 270-4261

Washington, D.C.
Star Technologies, Inc.
6206 Montrose Road
Rockville, Maryland 20852
(301) 984-9004

Florida
Star Technologies, Inc.
P.O. Box 1029
Maitland, Florida 32751-1029
(305) 788-1696

Western Europe and United Kingdom
Star Technologies International, Ltd.
7 Rue du Marche
1204 Geneve, Switzerland
Telephone 280496 or 280453
Telex 4-28-870

Canada
Star Technologies, Ltd.
300, 205 9th Ave. S.E.
Calgary, Alberta, Canada T2G OP8
(403) 269-6115
Telex 03-821519

DIVISIONS

Research and Development
Star Technologies, Inc.
511 11th Ave. South, Suite 255
Minneapolis, Minnesota 55415
(612) 332-0134

Engineering/Manufacturing/ Customer Service
Star Technologies, Inc.
101 International Drive
Sterling, Virginia 22170
(703) 471-9797

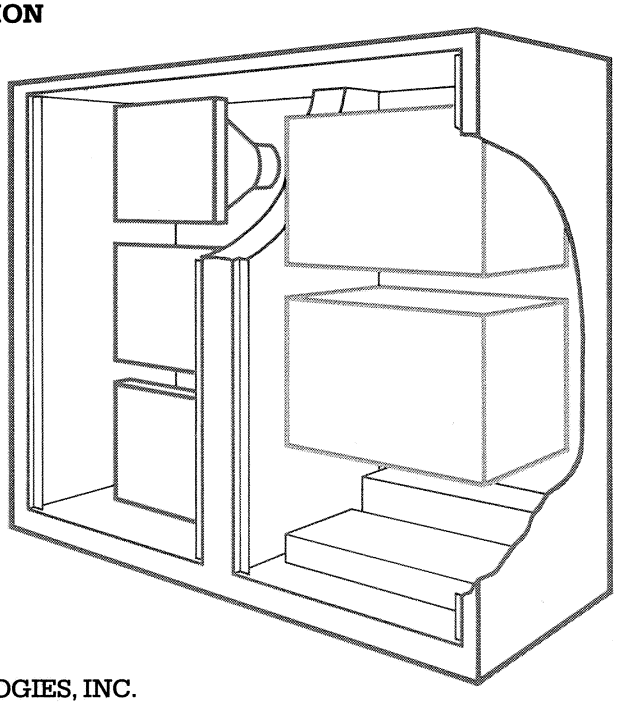
Corporate Offices
Star Technologies, Inc.
1200 Benjamin Franklin Plaza
One S.W. Columbia
Portland, Oregon 97258
(503) 227-2052

High-density VLSI circuitry enhances the ST-100's inherent reliability by reducing its component count. This reduction permits direct air cooling of individual circuits, which removes temperature gradients created by ordinary packaging. Extensive microprogrammed control permits a truly **synchronous architecture** that allows simpler clock and control circuitry. Additionally, the ST-100 has an **idle loop reliability** test that continually monitors the ST-100's operation.

The synchronous architecture also permits writing exact diagnostic microprograms. These diagnostic programs may be executed on-line or off-line through a maintenance terminal, either locally or by a remote modem. **SECDED** errors are logged to better diagnose the main memory.

Field maintenance is simplified by a minimum number of unique circuit board types in the ST-100. The physical placement of the boards, cooling ducts, and interconnection cables eases access for faulty board identification without using extender cards or disturbing cooling ducts.

CHASSIS CONFIGURATION



STAR TECHNOLOGIES, INC.
1200 Benjamin Franklin Plaza
One S.W. Columbia
Portland, Oregon 97258
503/227-2052

 PRINTED IN U.S.A. OCT. '82

ST-100

ARRAY PROCESSOR

STANDARD PRODUCT
INTERNATIONAL PRICE LIST

EFFECTIVE 1 APRIL 1983

PRODUCT

ST-100 ARRAY PROCESSOR
The **ST-100** is a high speed array processor completely packaged to include all hardware and software components necessary for efficient operation with a selected host computer. Each unit is provided with power supplies and equipment chassis mounted into a FCC regulated EMI/RFI cabinet.

Each **ST-100 basic configuration** includes:

- Host channel interface
- 512K 32-bit word main memory with SECDED
- 48K 32-bit word random access data cache
- I/O processor
- Storage/move processor
- Arithmetic processor
- Control processor
- Development software package
  - — Array Processor Control Language (APCL)
  - — Macro assembler
  - — Simulator Debugger
  - — Linker
  - — Library maintenance program
- Production software package
  - — AP resident monitor
  - — Host resident executive
  - — FORTRAN callable application library
- Maintenance software package
  - — Local/remote diagnostics
  - — User confidence tests
  - — Idle loop reliability tests
- System installation
- Documentation

Model No.	Description	Purchase Price
ST-100	basic equipment configuration	$325,000.
ST-MT100(1)	maintenance terminal	11,050.
Options:(2)		
Hardware:		
ST-512K	512K word memory increment	$ 66,560.
ST-VAX-I	additional host interface	13,000.
ST-IBM-I	additional host interface	13,000.
ST-PE-I	additional host interface	13,000.
ST-SEL-I	additional host interface	13,000.
Software:		
ST-VAX-S	additional host software	$ 16,250.
ST-IBM-S	additional host software	16,250.
ST-PE-S	additional host software	16,250.
ST-SEL-S	additional host software	16,250.
ST-VAST	vector and array syntax translator	26,000.
ST-VPSS	VPSS-3838 simulator	19,500.

(1) One ST-MT100 Maintenance Terminal is required for each ST-100 configuration.

(2) All additional host interface and additional host software option pricings include installation.

ORDER INFORMATION

TERMS
The **ST-100** array processors and associated options may be ordered through your local Star Technologies sales representative. All sales are subject to our standard terms and conditions. Terms are 1% 10, net 30 days. FOB point of origin. Shipment will be best available unless otherwise specified. Freight will be prepaid by Star and invoiced separately to the customer.

Contact your local sales representative for OEM pricing and other configuration options.

INSTALLATION
The basic system price includes hardware and software installation of the **ST-100** array processor to a supported host computer for end users. Additional services are available on a quotation basis.

WARRANTY
The **ST-100** is warranted against defects in materials and/or workmanship for one year for end users, 60 days to OEM customers. This includes both parts and labor. Star's sole liability is to repair or replace defects.

MAINTENANCE CONTRACT
Maintenance Contract is available to cover all normal service requirements for a fixed annual fee, plus, in some cases, a travel charge. Contact your local STAR representative for details.

STANDARD FIELD SERVICE
On-call service is available for on-site repair from your local service or sales representative.

TRAINING
Training courses which include tutorial hardware and software classes are available on a tuition basis. Detailed maintenance courses are also available. Your local sales representative will provide you with pricing and course scheduling information.

Information contained herein is subject to change without notice.

STAR TECHNOLOGIES, INC.
Cory House
The Ring
Bracknell
Berks. RG12 1ES
Telephone: 0344 54471
Telex: 848204 STAR G

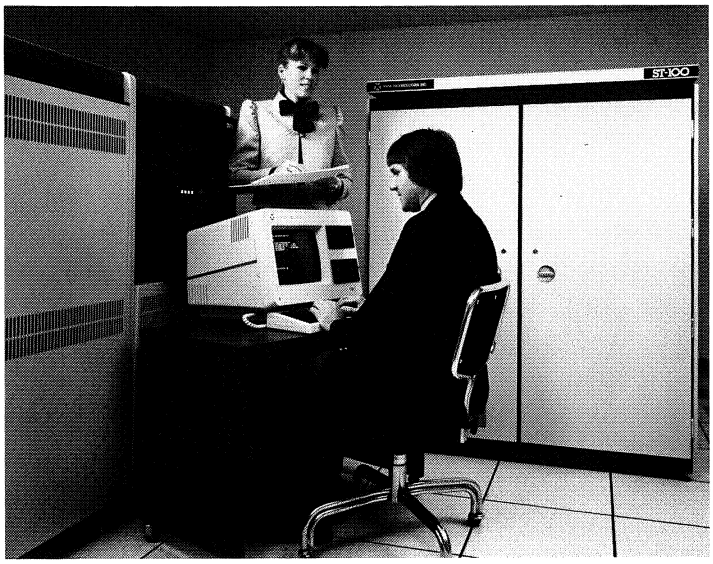
ST-100

ARRAY PROCESSOR

INSTALLATION SPECIFICATIONS

ST-100 ARRAY PROCESSOR

The **ST-100** Array Processor was designed to operate within the same environment as required for the proper operation of the host computers to which the ST-100 is connected. The environmental controls for electrical, temperature and humidity stability required by host systems such as the Digital Equipment Corporation's VAX-11/780, the IBM, Perkin-Elmer and Gould/SEL machines are sufficient to meet the requirements of the ST-100 Array Processor. Included below are power requirements for the ST-100, the heat load that the ST-100 will add to the host system and the interconnect cable lengths.



Power:

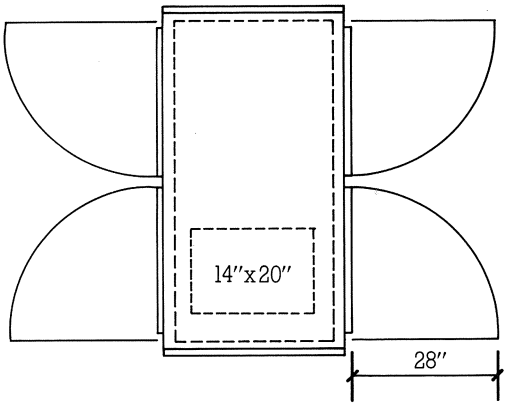
208V 30A 60Hz (230V 50Hz Optional)

Heatload:

17,000 BTU Per Hour

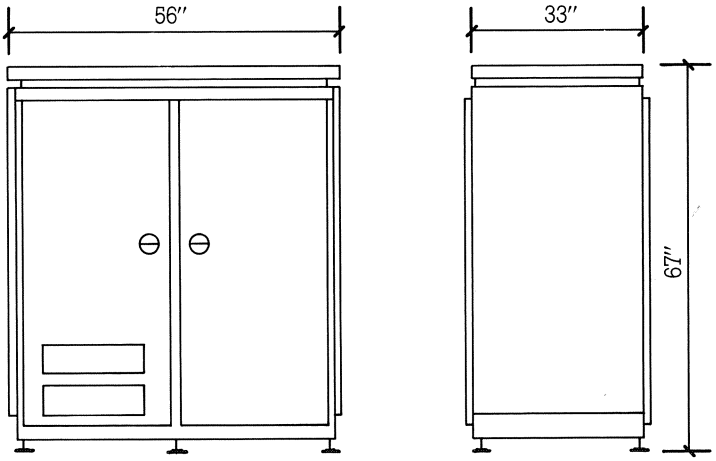
Maximum Cable Lengths Host to ST-100:

VAX-11/780—100'
IBM—100' (Customer Supplied)
Perkin-Elmer—100'
Gould/SEL—100'



Floor Loading:

Does not exceed 40 lb./sq. in. point loading
Total weight will not exceed 1,300 lb.



STAR TECHNOLOGIES, INC.
1200 Benjamin Franklin Plaza
One S.W. Columbia
Portland, Oregon 97258
503/227-2052

ST-VAST

VECTOR AND ARRAY SYNTAX TRANSLATOR FOR THE ST-100

VAST™ FOR THE ST-100

VAST™ (the Vector and Array Syntax Translator) is a software tool for developing efficient programs to be run on ST-100 array processor systems. VAST is a precompiler that analyzes DO loops in standard FORTRAN programs and converts those loops for which vectorization is possible into array operations. VAST creates a listing of the input program with diagnostic comments added to tell the user which loops were not vectorized and why. VAST also creates an enhanced version of the input program which includes efficient ST-100 processes in place of the vectorized loops.

ADVANTAGES

VAST provides several advantages over hand coding in an array syntax:

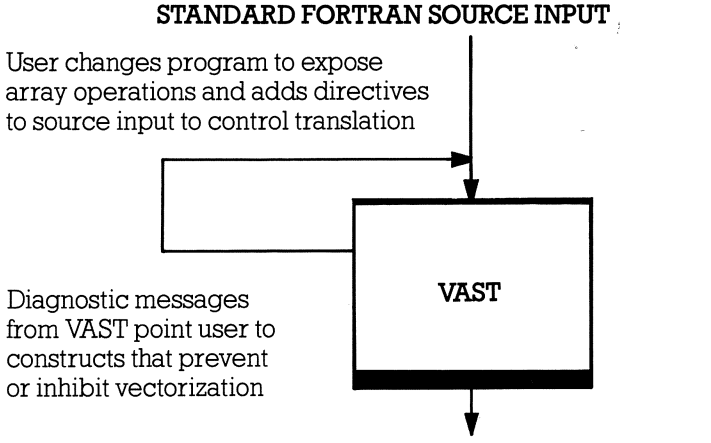
- Source code remains transportable (the program can be written in standard FORTRAN which will run on the host system and other machines).
- Conversion effort is much reduced (no need for the programmer to learn a new syntax; he needs to make fewer changes to his program, and diagnostic messages point him to areas requiring more work).
- Debugging is easier (it can be done in standard FORTRAN on the host computer).
- Source code remains readable (no strange syntax or mass of subroutine calls).

CAPABILITIES

VAST's abilities fall into three general areas: communication with the user, vectorization of DO loops, and bundling of operations so the ST-100 can operate on a large chunk of a program each time the array processor is invoked. Communication includes both the diagnostics by which VAST sends messages to the programmer and the user directives by which the user directs VAST to take certain actions; these features allow the user to know exactly what has occurred and to control the translation process as he chooses. The iterative process that results from this communication is shown in the figure.

In the area of vectorization, VAST is capable of recognizing the following FORTRAN DO loop situations and converting them into the appropriate array processor syntax:

- Propagation of scalar temporaries into temporary vectors.
- Recognition of reduction functions, including vector summation, dot product, miminum or maximum element, and vector product.
- Data dependency analysis to determine if loop is safe for vectorization.
- Both loop-dependent and loop-independent conditional assignment statements, forward transfers, and block IF's.
- Nontrivial subscript and index variable expressions.
- Indirect addressing.
- Statement function expansion.
- Use of an index as part of the calculation.
- Use of functions which have vector versions.



Iterating with VAST

In order to gather up the maximum amount of calculation for each access of the ST-100, VAST can bundle together:

- All the operations in one vectorized DO loop.
- Several consecutive vectorized DO loops (even if there are a few scalar statements between them).
- Tightly nested DO loops (outer loops are done in scalar mode on the ST-100).
- Non-tightly nested DO loops (controlled with a user directive).
- User ST-100 macros (subroutines with ST-100 macro versions are called on the ST-100 rather than forcing a return to a FORTRAN version on the host).

Finally, in generating code for the ST-100, VAST allocates temporary vectors in the cache memory of the array processor and outputs high-level vector primitive operations to ensure full utilization of the underlying hardware.

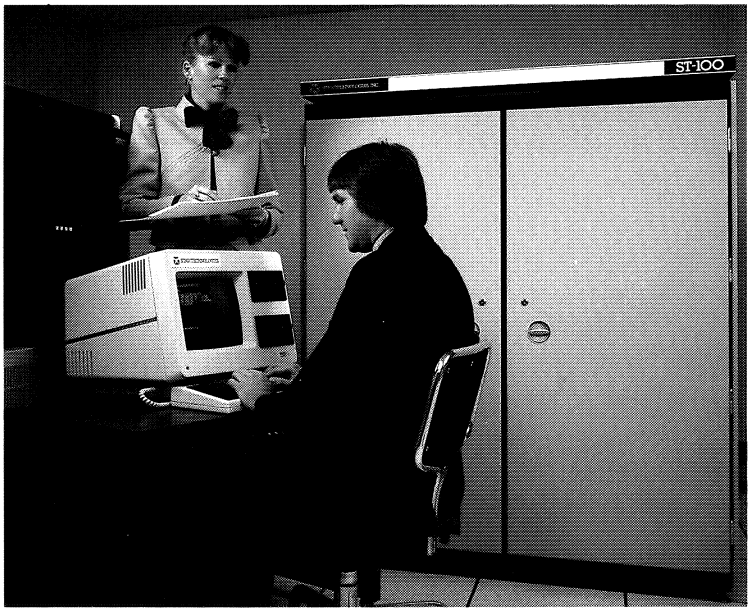
VAST bridges the gap between the optimal efficiency of hand-coded operations for the ST-100 and the total transportability and maintainability of standard FORTRAN on the host system. A programmer who uses this tool well can achieve impressive results with a minimum of effort.

NOTE: VAST is a trademark of Pacific Sierra Research Corporation.

STAR TECHNOLOGIES, INC.
1200 Benjamin Franklin Plaza
One S.W. Columbia
Portland, Oregon 97258
503/227-2052

ST-100

THE 100 MEGAFLOPS ARRAY PROCESSOR



STAR TECHNOLOGIES, INC.

Star Technologies, Inc., a manufacturer of scientific array processors, is dedicated to meeting users' demands for higher speeds and enhanced systems throughput at lower processing costs!

The performance of array processors has not increased as rapidly as general scientific computers. No **significant** performance improvement has been made in either architecture or technology, **until now!**

A new generation of array processors is now available. By using VLSI technology and an advanced modular architecture, Star Technologies has developed an array processor that competes with super computers at a fraction of their cost.

Star Technologies management and technical staff are computer professionals with extensive array processor and scientific computer backgrounds in a variety of application areas. Our development facilities are in Minneapolis, Minnesota and our corporate office is in Portland, Oregon. Operations facilities including engineering, manufacturing and customer training/customer service are located in Sterling, Virginia. Sales and support service offices serve Star Technologies customers worldwide.

Recent advances in computer aided design, VLSI and packaging technology have made possible a new generation of computer equipment. The advantage of this new technology is a dramatic increase in performance with moderate increase in cost. Star Technologies has utilized this advanced technology to build the ST-100. The ST-100 is constructed with 1200 gate bipolar VLSI circuits. These high density circuits have subnanosecond gate delays. Compared with conventional gate arrays, this logic yields substantial improvements in performance and greater utilization of on-chip components, thus permitting lower power dissipation.

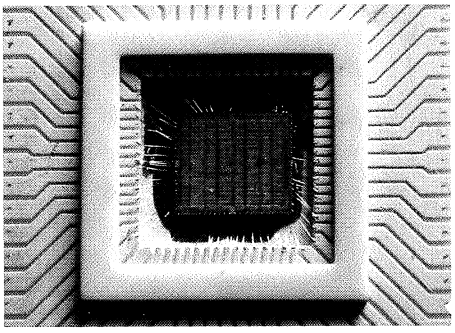
The availability of modern block level and gate level simulation permitted the thorough testing of VLSI circuits. Access to computer aided design routing and circuit placement software enabled the design of complex multilayered VLSI circuit boards.

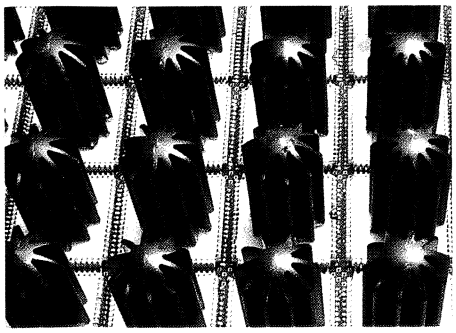
Coupled with this advanced VLSI design technology, unique cooling and packaging concepts were developed. A cooling system to eliminate temperature gradients across circuit boards ensures the cooling of each individual circuit.

The ST-100 utilizes a unique packaging concept, integrating small modules of 64K bit dynamic random access memories into a large multi-million byte memory.

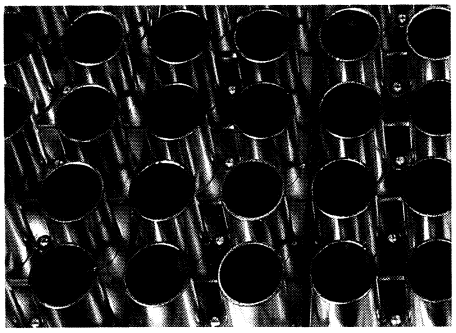
Dramatic improvements in hardware were not enough to ensure efficient system utilization. Today's software technology permits the sharing of multiple resources between multiple programs within a network of host computers. Star Technologies has used this advanced technology in the development of its ST-100 software system.

The end result of this technology and simulation capability is a 100 Megaflops Array Processor. **You be the judge – the ST-100 – designed with the user in mind.**

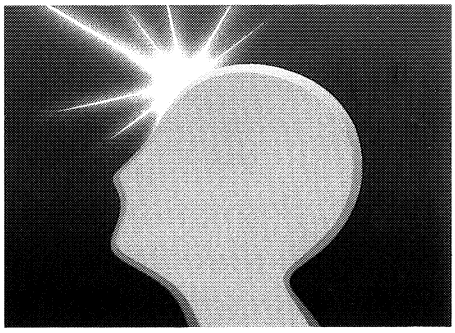




- Advanced VLSI technology
- Synchronous parallel, pipelined architecture
- Speed:
  - **40-nsec** machine cycle
  - **100 million floating point operations** per second (Megaflops)
  - 480-nsec hardware divide/square root
  - 40-nsec integer, logical/Boolean, or shift operations
- 32-bit word size
- Memory size:
  - Main memory up to **8 million words** (in 512K word increments)
  - 32 million bytes
  - Directly addressable to **1-Gigaword** (byte, half word, or word)
- Random Access data cache:
  - **48K words** in six 8192 word sections
- High speed input/output:
  - I/O channel supports up to 7 device adapters; 12.5 Mbyte/sec. data transfer rate
  - DMA for special devices; **100 Mbyte/sec.** data transfer rate
- Programmable hardware format conversion
- Multiple host interfaces
- Ambient air-cooled
- Remote maintenance capability

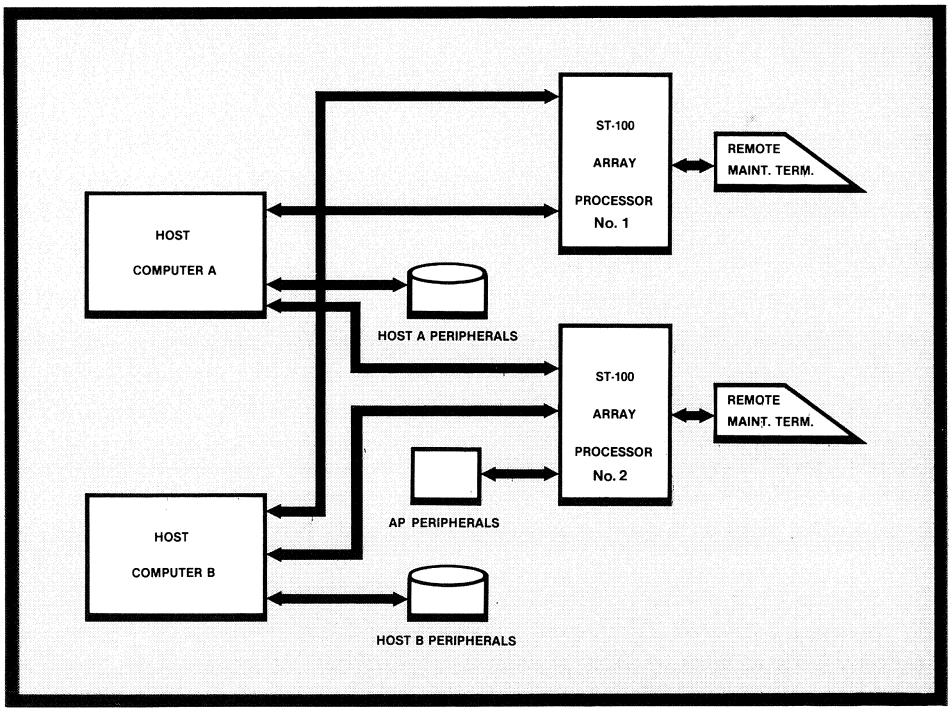


- Development Software
  - **FORTRAN-like** control language (ease of use)
  - macro assembler (simplifies microcoding)
  - library editor
  - simulator/debugger
- Production Software
  - **application library modules**
  - resource-sharing executive (supports multiple hosts connected with multiple array processors)
  - multi-user capability
- Maintenance Software
  - off-line and remote diagnostic capability
  - idle loop reliability tests
  - user confidence tests



IMAGINE: A new generation of array processors designed for your needs.

SYSTEM BLOCK DIAGRAM



Unique architectural features combined with innovative packaging permits a level of performance previously unavailable to planners of medium sized systems.

The **ST-100** consists of multiple processors and memories interconnected to allow **independent data flow and arithmetic processing.**

Specially constructed processors enable a MIMD (Multiple Instruction-Multiple Data) architecture to solve problems in parallel.

The **ST-100** has four independent programmable processors. A separate processor is dedicated to each of the following functions: external data flow, internal data flow, arithmetic processing, and synchronization. A hierarchical memory system consists of external storage devices, a large main memory, a high speed random access partitioned data cache, and universal register set.

The **control processor** stores microprograms and parameters into the following processors' memories providing synchronization of their concurrent operation.

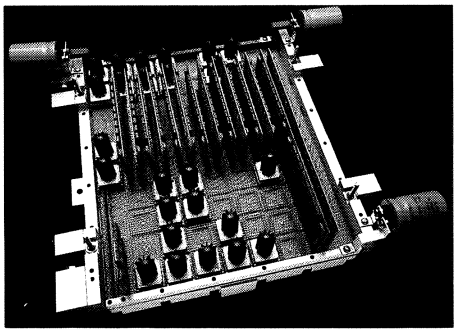
The **arithmetic processor** manages data flow between the data cache, the data interchange register set, and the arithmetic elements.

The **storage/move processor** manages movement of data between the main memory and the data cache.

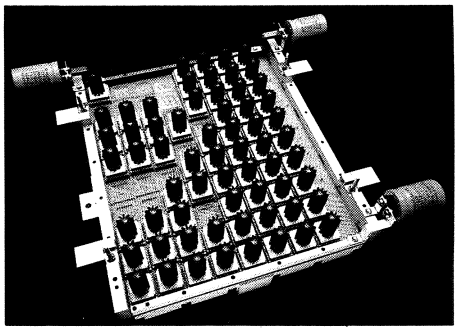
The **input/output processor** manages data flow between external devices (disks, host memories, etc.) and the main memory via a device adapter and internal channel.

CONTROL PROCESSOR

The control processor is an advanced microcomputer which manages resource allocation and data flow. This processor contains its own control and data memory, general purpose instruction set, hardware interrupt capability, interval timer, and direct memory access to the other three microprogrammable processors. The control processor is interconnected with the other three processors to manage microprograms and exchange control parameters. It initiates and monitors the other processors to synchronize their concurrent operations. This processor also manages overlapping data transfers and stages subsequent user jobs for execution without interferring with arithmetic processing. A maintenance terminal connects directly to the control processor via a local or remote communications link for diagnostics.



ST-100 Arithmetic Control Processor.



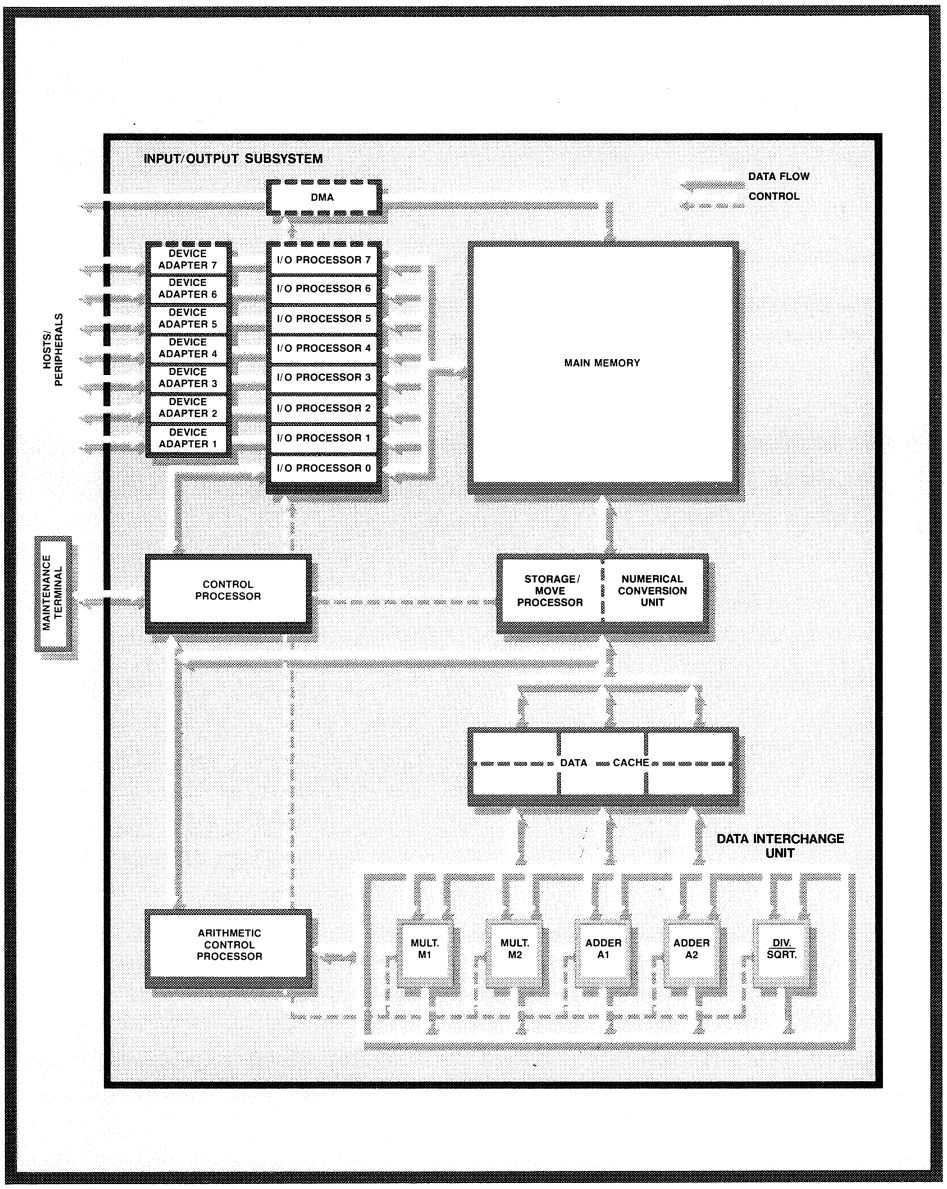
ST-100 Arithmetic Unit, showing gate array configuration with heat sinks.

ARITHMETIC PROCESSOR

The microprogrammable arithmetic processor consists of a control unit, arithmetic elements, and a data interchange unit. The control unit generates cache addresses, selects arithmetic operands and operations, performs loop count control, and controls a parameter stack. The arithmetic elements comprise **two add/subtract units, two multiplier units,** and a **divide/square root** unit. The data interchange unit permits one of 16 operands to be selected for each arithmetic input register. During each machine cycle, three cache banks may be referenced, one loop control operation computed, four arithmetic operations started, and a conditional branch executed. This processor computes at a rate of **100 million 32-bit floating point operations per second.**

STORAGE/MOVE PROCESSOR

The microprogrammable storage/move processor initiates and controls data flow between the main memory and the data cache. Transfers and data format conversion can take place at full memory bandwidth (100 Mbytes/second) through the numerical conversion unit. Main memory addresses, cache memory addresses, and loop control are calculated at full memory bandwidth. The storage/move processor allows complex addressing sequences to be computed concurrently with data transfers. **Integer, shift, Boolean and logical 32-bit operations can be performed** on data from main memory or data cache.



INPUT/OUTPUT PROCESSOR

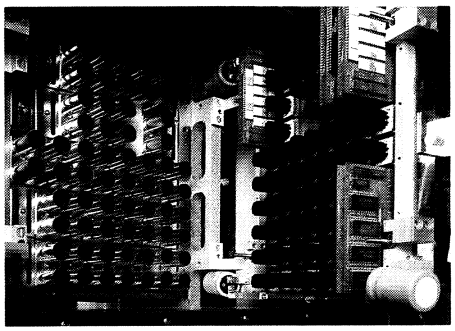
The input/output processor controls a 25 Mbyte channel and a 100 Mbyte Direct Memory Access (DMA) port. The 25 Mbyte channel supports seven device adapters. Device adapters connect host computers and peripherals to the array processor through the input/output processor. The device adapter can transfer at 12.5 Mbyte/second. High-performance devices can be connected directly to the Main Memory.

MAIN MEMORY

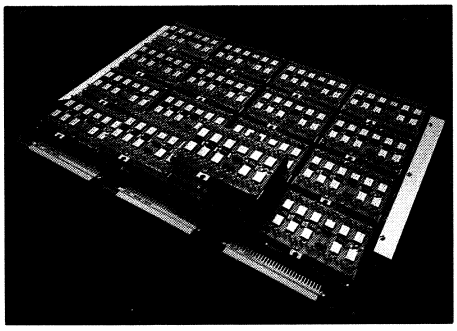
The Main Memory consists of eight 320-nsec cycle memories interleaved to provide an access rate of 40-nsec per 32-bit word. Single error correction and double error detection (SECDED) are implemented with seven additional bits. The Main Memory is **expandable to 8 million words** in increments of **512K words** by using 64K dynamic random access memory circuits. All Main Memory can be directly addressed as 32, 16 or 8-bit quantities. This memory can be partitioned and protected at multiples of 4096 words.

DATA CACHE MEMORY

The random access data cache memory consists of six banks of 8192 32-bit words with odd parity for a total of 48K words. During each machine cycle, four cache references are permitted: three by the arithmetic processor and one by the storage/move processor. These banks can be logically connected to main memory or the arithmetic elements under program control.



ST-100 Storage Move Processor, swung out from duct nozzles.



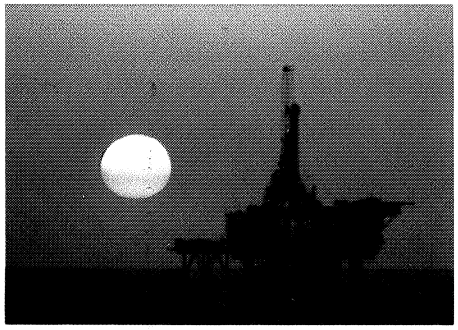
ST-100 Memory Board, showing maxi-and miniboards. A single layer produces 512K words. Four complete layers-2-million words. Four complete maxiboards-8-million words.

The ST-100 software system consists of three major segments: Development, Production, and Maintenance software.

DEVELOPMENT SOFTWARE

The ST-100 Development Software System is a set of FORTRAN programs residing in the host. It comprises a **High-Level control language, Macro Assemblers,** a **Simulator/Debugger,** a **Linker,** and a **Library Maintenance Program.** The Development Software System provides the ability to separately program all elements of the ST-100, allowing multiple levels of program optimization. These programs are used to create application library modules.

Application Library modules are written using the FORTRAN like high level control language. This module contains code that executes on the control processor which invokes macro calls to the arithmetic, storage/move and input/output processors. The **Macro Assembler** is used to generate these macros.



The **Simulator/Debugger** verifies the application modules and microprogrammed macros. The Linker then combines microprogrammed modules with object code created by the high-level language to produce host FORTRAN callable subroutines.

The **Library Maintenance Program** catalogues and maintains both the application library modules and the microprogrammed macro routines.

OPERATING SOFTWARE

The ST-100 Production Software system couples the array processor to the host application program. It consists of a **Host-Resident Array Processor Executive,** an **Array Processor-Resident Monitor,** and an **Application Library** that contains array processor executable code.

The **Array Processor Executive** is a set of FORTRAN subroutines that communicates with a device driver. This Executive manages the allocation of array processor resources and directs requests from the user's application program to the control processor.

The **Array Processor Monitor (APM)** schedules and controls requests from multiple users as well as from multiple hosts. The Monitor executes in the control processor.

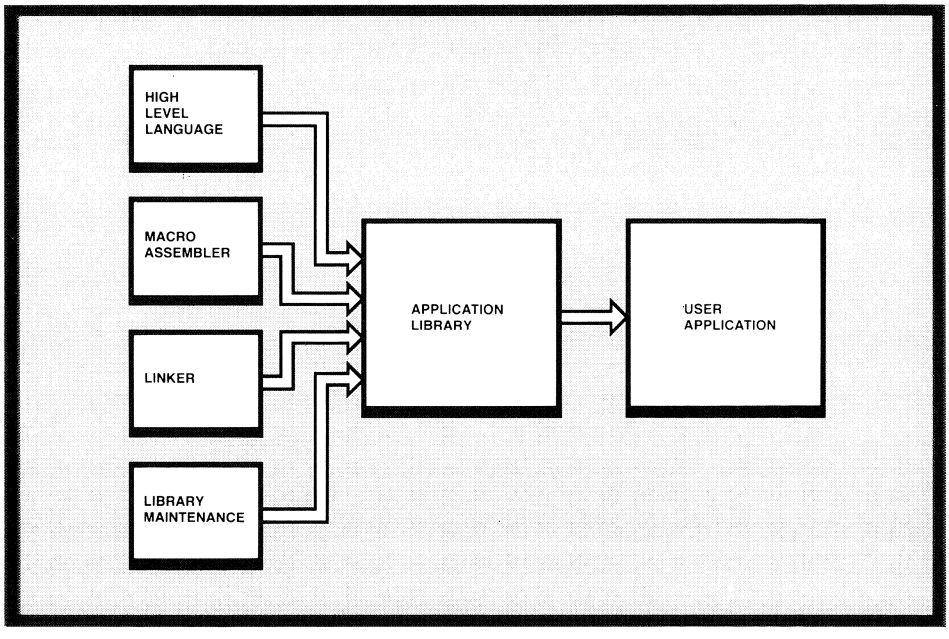
The **Application Library** contains the programs which are executed in the array processor.

MAINTENANCE SOFTWARE

The ST-100 Maintenance Software system consists of a set of fault identification and isolation routines. These routines can be used in three ways: **as user-callable confidence library** modules; as array processor monitor **idle loop reliability** modules; and as diagnostic programs.

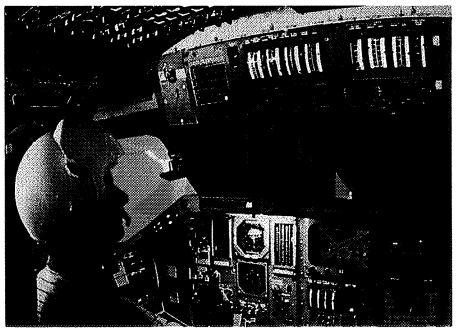
These **diagnostic programs** can be executed either on-line or off-line, and can be controlled either **locally** or **remotely.**

ST-100 DEVELOPMENT SYSTEM



Star Technologies designed the ST-100 Array Processor to be a **total system solution.** Executing complete application programs within the ST-100 reduces unnecessary host-array processor communications. The ST-100's ability to stage multiple jobs for subsequent execution allows one array processor to serve multiple users from multiple hosts.

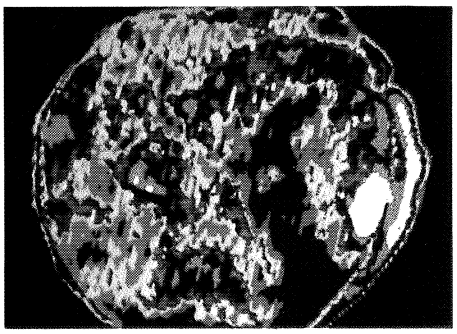
Staging of multiple jobs and data transfers can be overlapped without interferring with arithmetic processing. This activity is controlled by the array processor monitor.

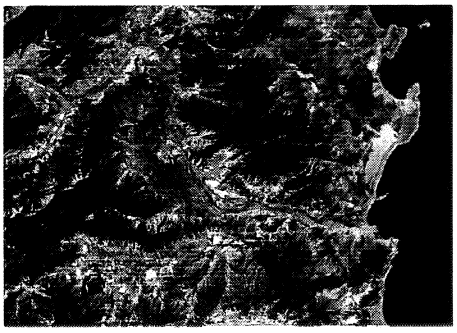


High-density VLSI circuitry enhances the ST-100's inherent reliability by reducing its component count. This reduction permits direct air cooling of individual circuits, which removes temperature gradients created by ordinary packaging. Extensive microprogrammed control permits a truly **synchronous architecture** that allows simpler clock and control circuitry. Additionally, the ST-100 has an **idle loop reliability** test that continually monitors the ST-100's operation.

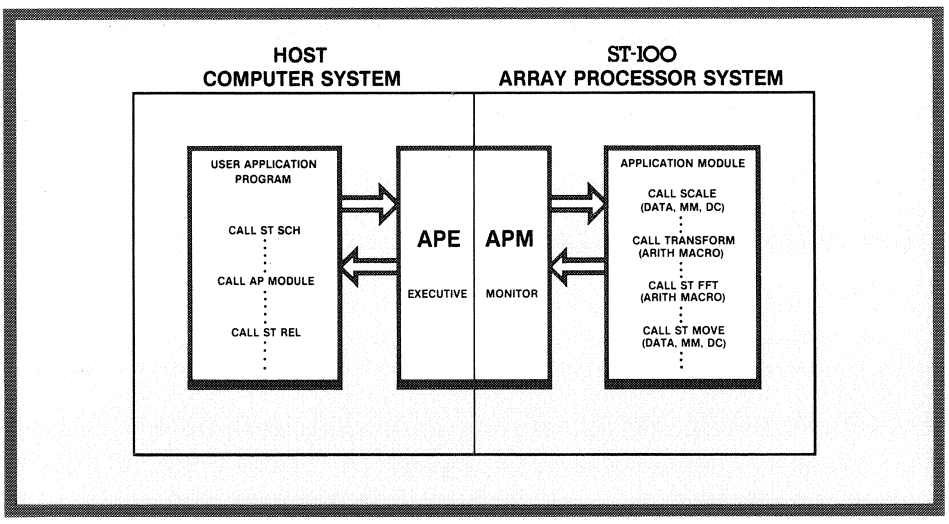
The synchronous architecture also permits writing exact diagnostic microprograms. These diagnostic programs may be executed on-line or off-line through a maintenance terminal, either locally or by a remote modem. **SECDED** errors are logged to better diagnose the main memory.

Field maintenance is simplified by a minimum number of unique circuit board types in the ST-100. The physical placement of the boards, cooling ducts, and interconnection cables eases access for faulty board identification without using extender cards or disturbing cooling ducts.





HOST PROGRAM — ST-100 INTERFACE



FIELD OFFICES

Southern California
Star Technologies, Inc.
24672 San Juan Ave., Suite 101
Dana Point, California 92629
(714) 661-0340

Northern California
Star Technologies, Inc.
231 Bernal Ave.
Pleasanton, California 94566
(415) 462-2481

Texas
Star Technologies, Inc.
10103 Fondren, Suite 110
Houston, Texas 77096
(713) 270-4261

Washington, D.C.
Star Technologies, Inc.
6206 Montrose Road
Rockville, Maryland 20852
(301) 984-9004

Florida
Star Technologies, Inc.
P.O. Box 1029
Maitland, Florida 32751-1029
(305) 788-1696

Oklahoma
Star Technologies, Inc.
6923 South Canton
Tulsa, Oklahoma 74136
(918) 492-2704

Colorado
Star Technologies, Inc.
Harlequin Plaza North, Suite 350
7600 E. Orchard Rd.
Englewood, Colorado 80111
(303) 850-9293

Illinois
Star Technologies, Inc.
4544 West 103rd St.
Oak Lawn, Illinois 60453
(312) 422-3224

Boston
Star Technologies, Inc.
7 Chestnut Street
Woburn, Massachusetts 01801
(617) 935-9290

Canada
Star Technologies, Ltd.
300, 205 9th Ave. S.E.
Calgary, Alberta, Canada T2G OP8
(403) 269-6115
Telex 03-821519

Headquarters for United Kingdom and Western Europe
Star Technologies International, Ltd.
7 Rue du Marche
1204 Geneve, Switzerland
Telephone 280496 or 280453
Telex 4-28-870

Germany
Star Technologies GmbH
Bussardstrasse 15
D-8311 Viecht, Germany
Tel. 08709-1651

France
Star Technologies France S.A.
C.E. 1430
Z.I. Petite Montagne Nord
7 Allée du Lubéron et du Vercors
91019 EVRY CEDEX
Tél: (33) 60 77 52 49
Télex: 691186

United Kingdom
Star Technologies, Inc.
Cory House
The Ring
Bracknell, Berkshire
RG 12-1ES
Tel. 344-54471

DIVISIONS

Research and Development
Star Technologies, Inc.
Suite 102
7101 Northland Circle
Brooklyn Park, Minnesota 55428
(612) 535-8100

Engineering/Manufacturing/ Customer Service
Star Technologies, Inc.
101 International Drive
Sterling, Virginia 22170
(703) 471-9797

Corporate Offices
Star Technologies, Inc.
1200 Benjamin Franklin Plaza
One S.W. Columbia
Portland, Oregon 97258
(503) 227-2052

STAR TECHNOLOGIES, INC.
Cory House
The Ring
Bracknell
Berks. RG12 1ES
Telephone: 0344 54471
Telex: 848204 STAR G

STAR TECHNOLOGIES, INC.
1200 Benjamin Franklin Plaza
One S.W. Columbia
Portland, Oregon 97258
503/227-2052

 PRINTED IN U.S.A. AUG. '83

ST-100
ARRAY PROCESSOR

SYSTEM OVERVIEW

STAR TECHNOLOGIES, INC.

PUBLICATION 90000000

REVISION RECORD	
REVISION	DESCRIPTION
01 (10-15-82)	Preliminary edition.

publication number: 90000000

Address comments concerning this publication to:
Star Technologies, Inc.
Publications Department
Suite 255
511 11th Avenue South
Minneapolis, MN 55415

PREFACE

This overview introduces the Star Technologies, Inc. (STAR) ST-100 Array Processor. Readers are assumed to have at least a general understanding of basic computer system concepts, including the functional relationships among hardware and software components and the general requirements and procedures for program generation and execution.

Section 1 of the overview is an introduction briefly defining what an array processor is and summarizing the important benefits and features of the ST-100 Array Processor. Section 2 explains the ST-100 system in terms of its functional characteristics, development facilities, and production (execution) facilities. Section 3 summarizes the functions and noteworthy capabilities of the ST-100 hardware and software components. The appendix includes detailed feature lists of the Array Processor Control Language (APCL), array processor executive (APX), and application support library, plus an example of an ST-100 process.

CONTENTS

FIGURES

SECTION 1
INTRODUCTION

The Star Technologies, Inc. (STAR) ST-100 Array Processor is a high-performance, cost-effective array processor. It attaches to general-purpose computer systems (figure 1-1) for use in signal processing, image processing, simulation, geophysical applications, and general scientific computing.

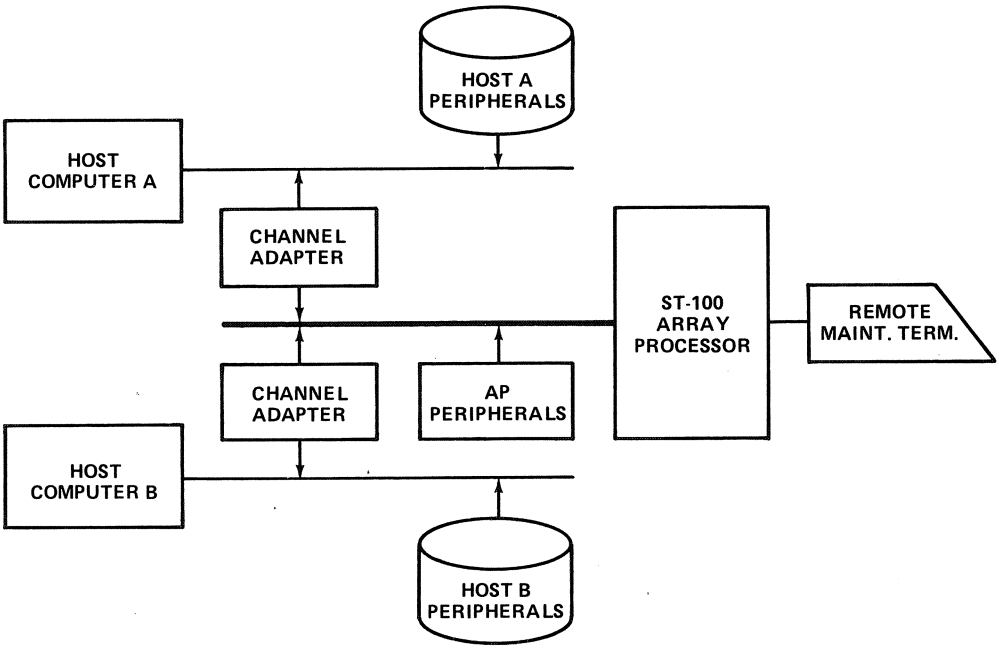


Figure 1-1. ST-100 Multiple Host Attachment

An array processor is designed to attach to a more general-purpose computer, or host computer. The combination of the host computer and array processor provides more cost-effective scientific computation. A significant feature of the ST-100 Array Processor is that it can attach to and service multiple host computers concurrently. Its relatively small size (about 5 ft. H x 4-1/2 ft. W x 2-1/2 ft. D) and air-cooled design offer flexibility in placing the ST-100 in a variety of computing environments. Yet, its large capacity and high speed computing capability at a reasonable price make general-purpose computers with an attached ST-100 Array Processor a viable and competitive alternative to replacing or supplementing those general-purpose computer facilities with more expensive conventional scientific computers.

The high-performance ST-100 Array Processor is flexible, efficient, and easy to use.

The high performance of the ST-100 Array Processor is due to its unique architecture, which employs multiple processors, a hierarchical high-capacity data memory system, VLSI (very large-scale integration) circuitry, and a combination parallel-pipelined arithmetic section. It can execute data movement operations and arithmetic operations concurrently, performing arithmetic operations at up to 100 megaflops (100 million floating point operations per second).

The ST-100 Array Processor is flexible. Its design enables it to support up to seven host computers, if necessary. Its input/output subsystem is expandable to include independent input/output processors for each host system. The ST-100 architecture also allows for the attachment of peripheral devices, such as disks and data acquisition devices, directly to the array processor.

The ST-100 Array Processor is efficient because of its unique multiple processor design, large memory system, and production software support. The programs that execute in the ST-100 Array Processor are called processes. The structure of processes takes full advantage of the multiple-processor design, consisting of more general-purpose computer code executing in a control processor and specialized microprocessor programs (microcode), called macros, executing in a storage move processor and arithmetic control processor. The multilevel program structure plus the hierarchical memory structure enable more efficient use of both host computer resources and array processor resources. They allow the array processor to perform a hierarchy of concurrent arithmetic and data movement operations without the need for host computer intervention. In addition, ST-100 production software allows concurrent staging and execution of processes, enabling subsequent processes to be loaded and readied for execution while the current process executes.

The ST-100 Array Processor is easy to use. It features the Array Processor Control Language (APCL), an extended subset of FORTRAN 77. APCL allows process programmers full manipulation of the unique features of the ST-100 architecture while retaining the ease-of-use inherent in the FORTRAN-like coding characteristics of the language. Macros, which control process data-movement operations and arithmetic operations, are executable by APCL calls. The ST-100 includes a large library of precoded macros to facilitate process development. It also includes the Macro Assembly language, a microprogramming assembly language, for users desiring to code their own macros to supplement the precoded library macros. This complement of APCL, support library, and Macro Assembly language enables users to program and implement the ST-100 Array Processor according to their own experience, capabilities, and needs. Added to this are simulation and debugging

facilities for macro and process debugging, plus a standard host computer interface, which allows transportability of ST-100 processes from one host computer to another without reprogramming.

SECTION 2
SYSTEM DESCRIPTION

Physically, the ST-100 Array Processor attaches to a host computer like any other peripheral device. Logically, however, the ST-100 acts as an extension of the host computer processor, executing compute-intensive applications. The applications that execute in the ST-100 Array Processor are called processes. The ST-100 Array Processor, therefore, is a system of hardware and software components for both developing ST-100 processes in the host system and executing the processes in the array processor. Understanding the full function and use of the ST-100 system evolves around the definition, development, and execution of ST-100 processes; that is, what a process is, how it relates to the ST-100 hardware and software components, and how a host FORTRAN application program executes it.

ST-100 FUNCTIONAL CHARACTERISTICS

The functional characteristics of the ST-100 Array Processor hardware determine the structure and operation of the ST-100 processes. The ST-100 Array Processor hardware is in itself a system of processor elements and data storage (memory) elements housed in a single cabinet with an attached maintenance terminal (figure 2-1). The processor elements include an input/output subsystem, control processor, storage move processor, and arithmetic control processor with associated arithmetic units (multipliers, adders, and divide/square root unit). The data storage (memory) elements are main memory and the data cache.

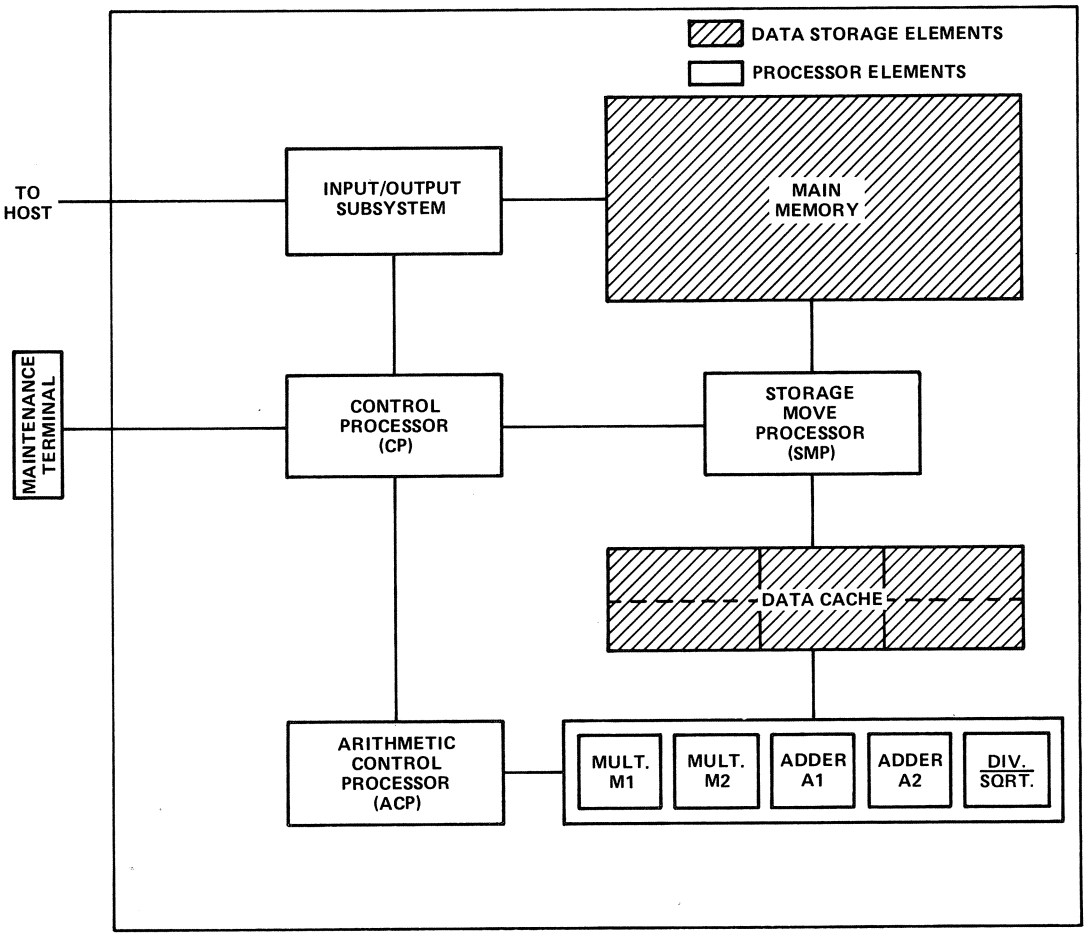


Figure 2-1. ST-100 Hardware Elements

ST-100 Major Flows

The major function of an array processor is to efficiently perform compute-intensive applications, which generally require large memory capacities. The large memory, multiple-processor structure of the ST-100 Array Processor is designed for that purpose.

Data from the host computer enters the array processor through the input/output subsystem into main memory (figure 2-2). Main memory is the bulk data storage element of the array processor with a capacity of up to eight million 32-bit words available. The storage move processor (SMP) then moves the data from main memory to the data cache, where the data is accessed by the arithmetic control processor (ACP) for arithmetic data processing. The ACP returns processed data to the data cache, from which the SMP moves the data to main memory. The processed data is then transmitted to the host computer through the input/output subsystem. The control processor provides overall processor coordination.

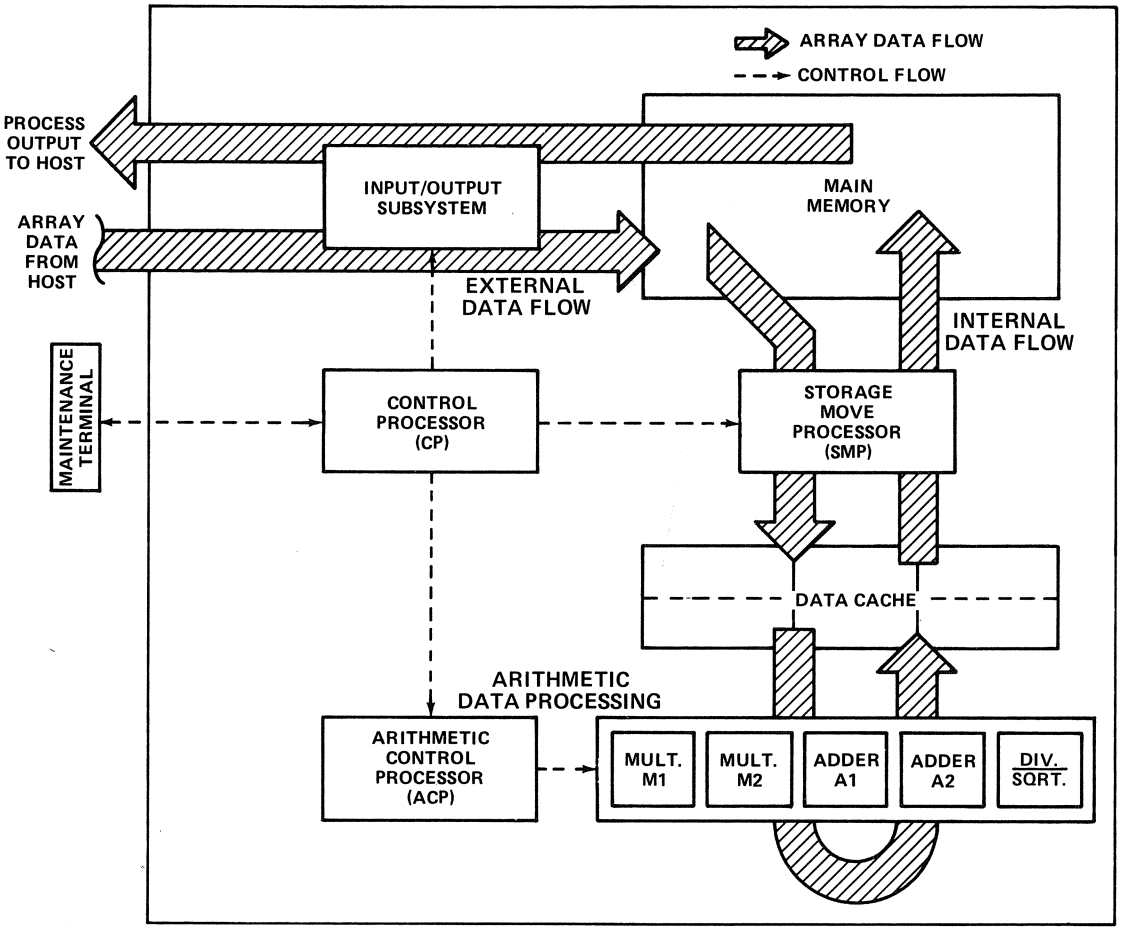


Figure 2-2. ST-100 Major Flows

ST-100 Process

Data flow into and out of the array processor is initiated by the host system. In that sense, the array processor is a slave device to the host computer, providing extended computer resources to the user. Data movement within the array processor and arithmetic data processing is controlled by the ST-100 process (figure 2-3). An ST-100 process consists of general-purpose control-oriented program code executing in the control processor (CP) plus specialized microprocessor code (macros) executing in the SMP and ACP. The CP controls SMP and ACP operation using program calls to SMP and ACP macros to initiate operations. ST-100 processes and the data upon which they operate originate from the host computer system, which loads the processes and data into the array processor for execution.

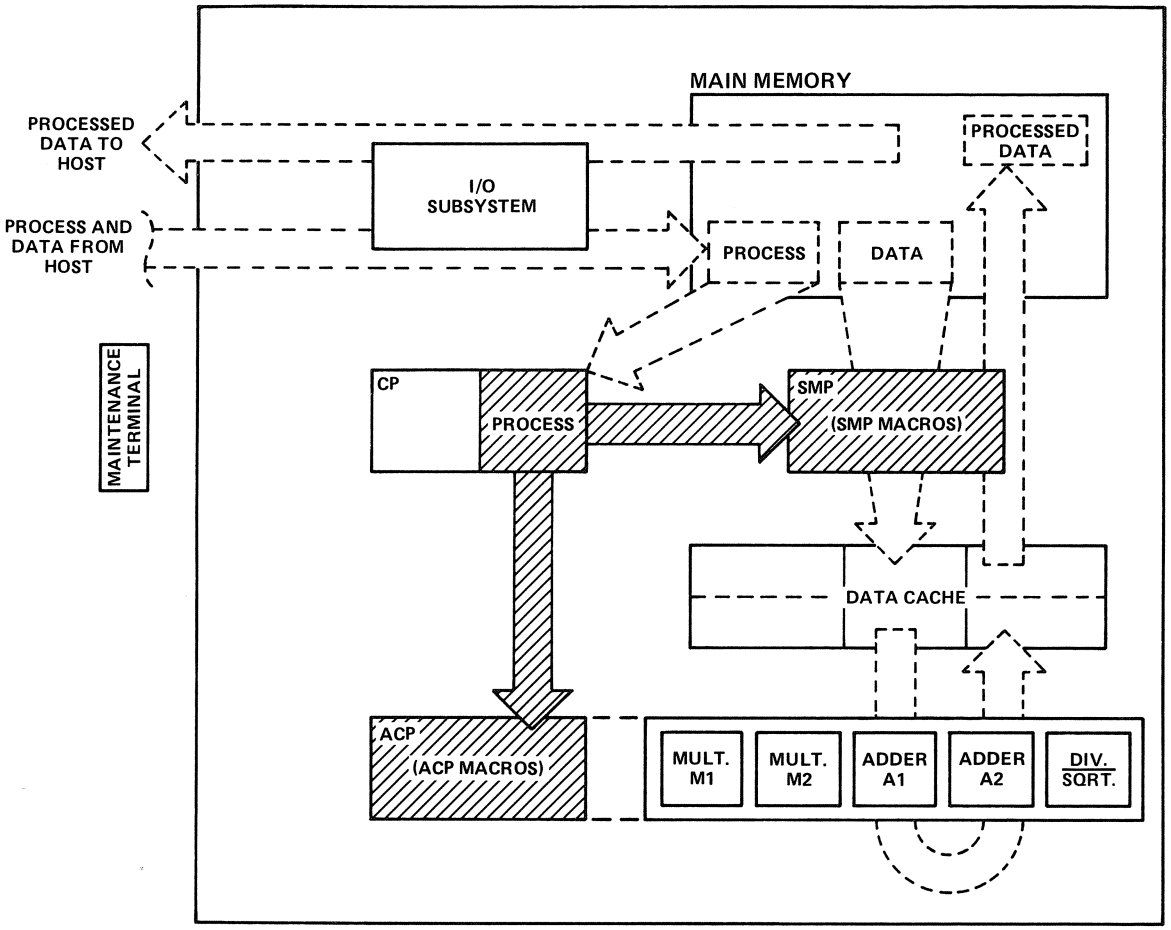


Figure 2-3. ST-100 Process

Concurrent Staging and Execution

The control processor consists of two Motorola 68000 microprocessors, which can operate simultaneously in the system. One microprocessor executes the ST-100 process controlling the internal movement and arithmetic processing of data. The second microprocessor contains the array processor monitor (APM), which controls the loading of processes and data for execution and retrieval by the host system. While one process executes, the APM stages any other processes being transmitted to the array processor by the host system (figure 2-4). The APM is capable of loading processes from multiple host systems, establishing process execution priorities, and keeping track of which host system process is currently executing.

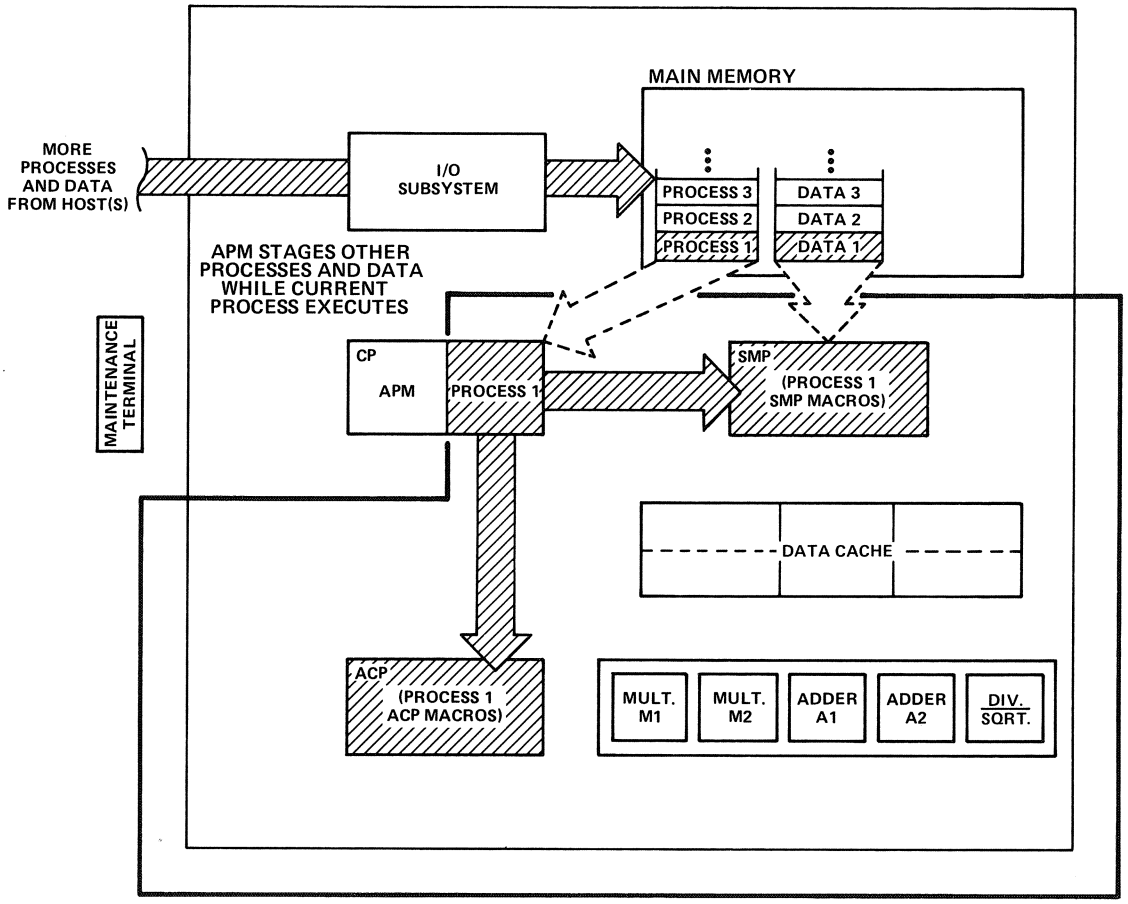


Figure 2-4. Concurrent Staging and Execution

ST-100 DEVELOPMENT FACILITIES

Users program and debug ST-100 processes on the host computer system using program tools supplied with the ST-100 system. Processes are multilevel program structures consisting of general-purpose program code executable in the control processor within the array processor plus macros that execute in the storage move processor (SMP macros) and arithmetic control processor (ACP macros). The general-purpose code executing in the control processor is coded in a FORTRAN-like language called Array Processor Control Language (APCL). SMP macros and ACP macros are coded in Macro Assembly language. Process development, therefore, consists of at most two phases: APCL process development and, if necessary, macro development.

Macro Development

STAR provides a large library of SMP and ACP macros for use in process development. If the user requires additional macros, macro development tools provided as part of the ST-100 system are available for use. SMP and ACP macro development consists of coding the macro in Macro Assembly language, using a host system text editor program to put the source code on disk, assembling the macro using the STAR macro assembler, linking the macro using the STAR linker program, and then debugging the macro using the STAR simulator and debugger facility (figure 2-5).

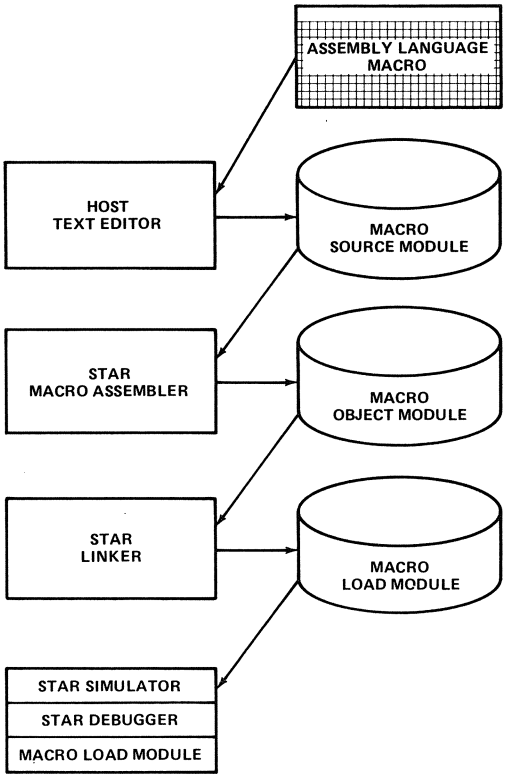


Figure 2-5. Macro Assembling and Debugging

The STAR simulator simulates ST-100 Array Processor operation, allowing debugging of the macro through the STAR debugger facility. Any errors found in the macro require the user to correct the macro source code and reassemble the macro (figure 2-6).

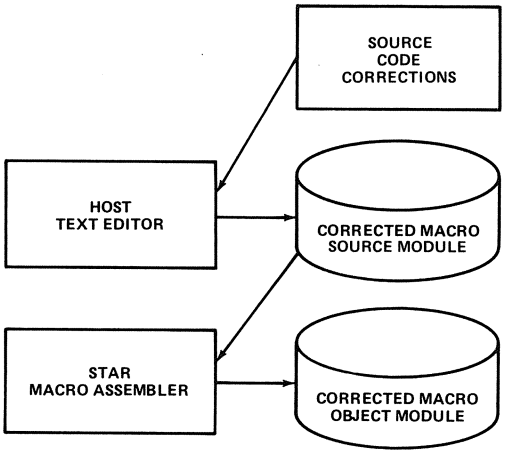


Figure 2-6. Macro Correction

The corrected macro assembler output (macro object module) is used in APCL process development.

APCL Process Development

The development of a process involves generating the process and developing a host FORTRAN test program to execute the process for debugging. The steps follow.

1. The user codes the process in APCL. The APCL process includes calls to the SMP and ACP macros used in the process. The user puts the source code on disk using a host system text editor, compiles the process using the STAR APCL compiler, and, using the STAR linker, links the compiled process with the assembled SMP and ACP macros to be included in the process (figure 2-7).

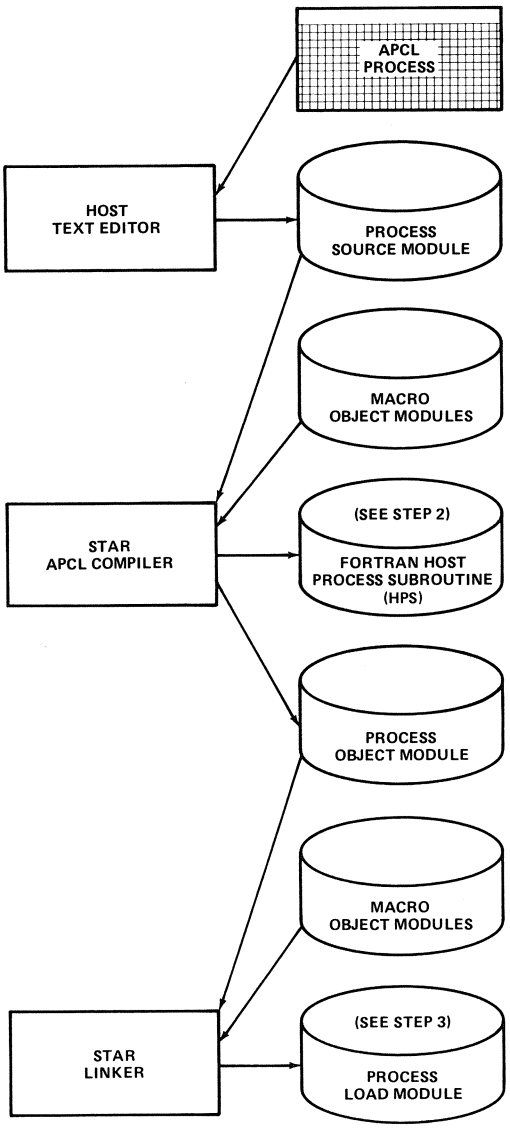


Figure 2-7. Process Compilation and Linking

Output from the STAR APCL compiler includes a special FORTRAN subroutine, called the host process subroutine (HPS). It must be compiled and then linked with the host FORTRAN application program to effect a proper link between the application program and the process it executes.

2. The user codes the host FORTRAN test program, complete with calls to the ST-100 processes it executes, compiles the program with the ST-100 host process subroutines for the processes called by the program, and links the compiled test program to prepare it for loading and execution (figure 2-8).

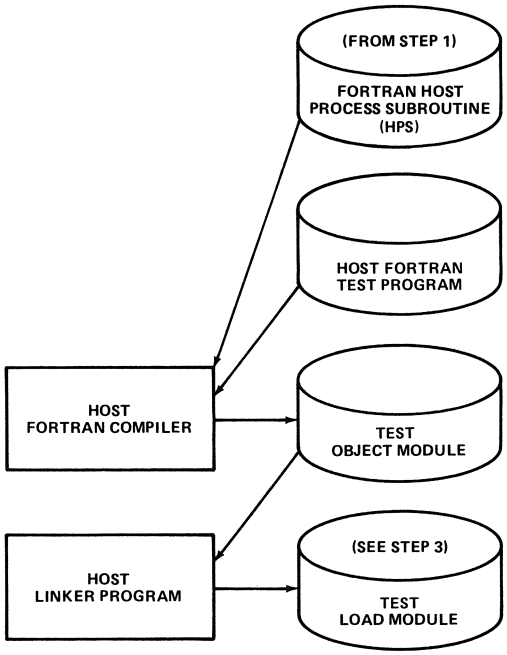


Figure 2-8. Test Program Generation

3. Finally, the user debugs the process using the STAR debugger facility (figure 2-9). The STAR array processor executive (APX) is loaded and executed in debugger mode along with the host FORTRAN test program, which calls the process for execution in the array processor.

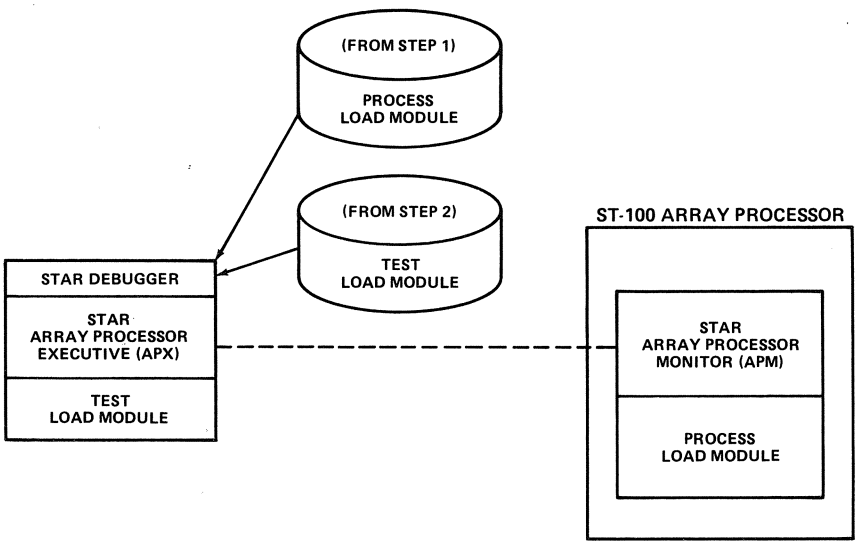


Figure 2-9. Process Debugging

Any ST-100 process errors detected during debugging require the user to correct the APCL process source code, recompile the process, and relink the process (repeat step 1). The final result of process development is a process load module ready for execution in the array processor. The load module includes the general-purpose control code for execution in the control processor (CP code) and the SMP and ACP macros called by the process (figure 2-10).

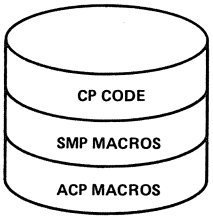


Figure 2-10. Process Load Module

Application Development

Host application development involves coding the FORTRAN programs that require ST-100 process execution, compiling the programs and linking them to the ST-100 processes they execute, and executing the applications as normal jobs or job steps in the host computer system. Linking a host application program to an ST-100 process consists of linking the program to the FORTRAN host process subroutine (HPS) generated for the process by the STAR APCL compiler, as previously described in figure 2-8 for the FORTRAN test program used in process debugging. FORTRAN application program coding involves coding a series of FORTRAN calls to the STAR array processor executive (APX) to prepare the array processor for use and call the ST-100 processes for execution. This is described in more detail in the following section.

ST-100 PRODUCTION FACILITIES

Once the ST-100 processes have been developed and the host FORTRAN application programs that call them have been generated, the ST-100 Array Processor is ready for production (use). ST-100 production software, consisting of the array processor executive (APX) and array processor monitor (APM), control communication between the host system and array processor, including the loading of processes for execution and the transfer of data between the host system and array processor.

All ST-100 processes are executed by calls to the APX from host FORTRAN application programs. The general sequence of activities in executing a process follows:

1. The FORTRAN application program is loaded for execution as a host job or job step. The application program issues a call to open the array processor and schedule an application partition in the array processor (figure 2-11). The application partition will be used to contain the process and data when they are loaded for execution.

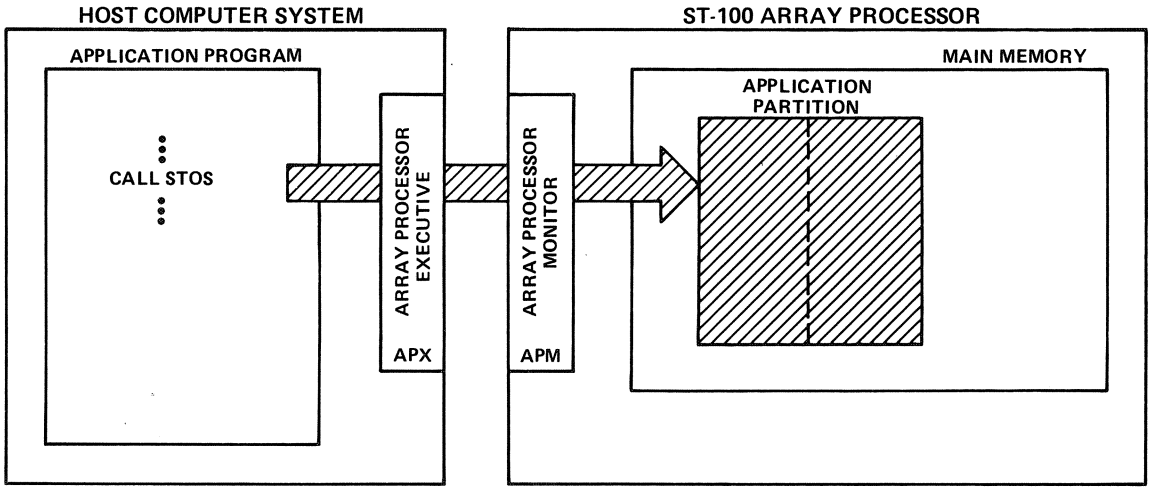


Figure 2-11. Scheduling an Application Partition

2. The application program issues calls to define the data areas in the application partition and transfer the data into these areas (figure 2-12).

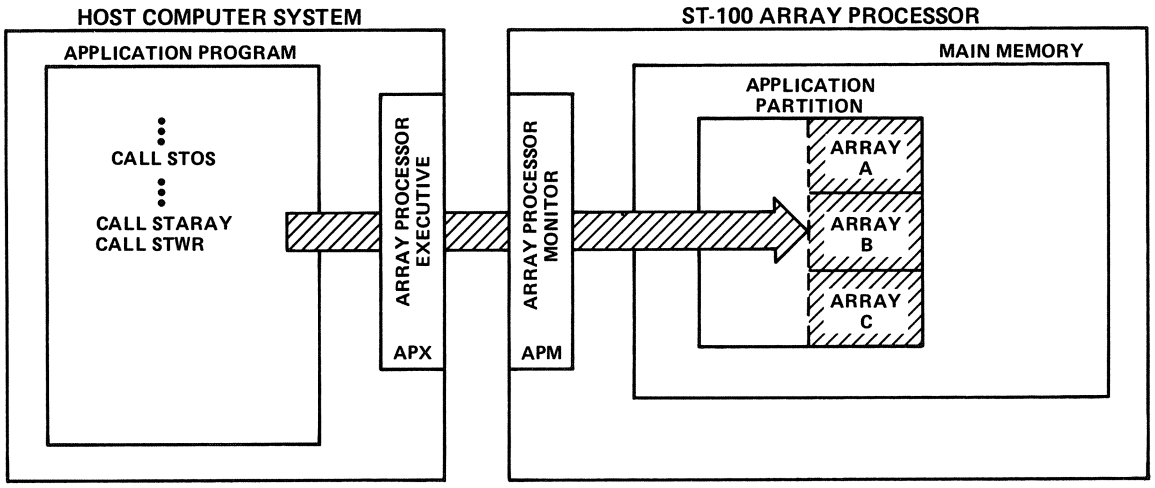


Figure 2-12. Loading Data

3. The application program calls the process for loading and execution (figure 2-13). Parameters are provided by the calling program to direct process execution.

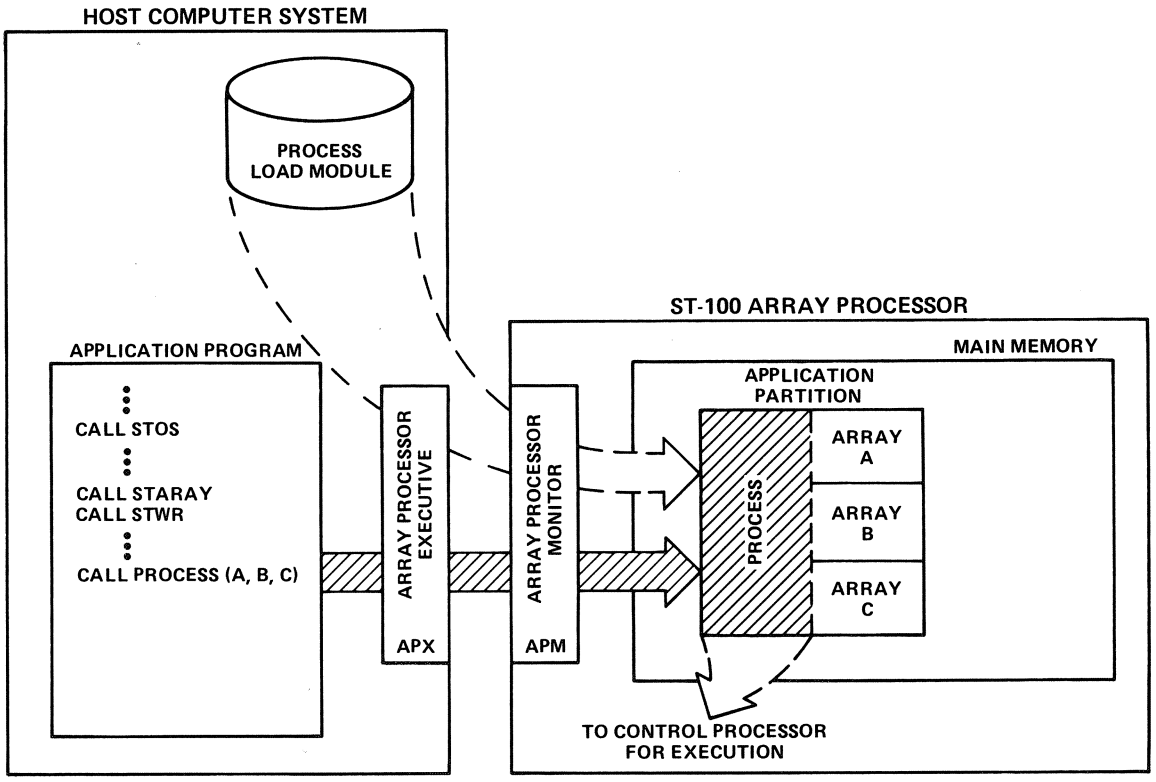


Figure 2-13. Executing the Process

4. Upon the array processor completing the process, the host application program reads the processed data from the array processor (figure 2-14). The processed data occupies space in the data area of the application partition.

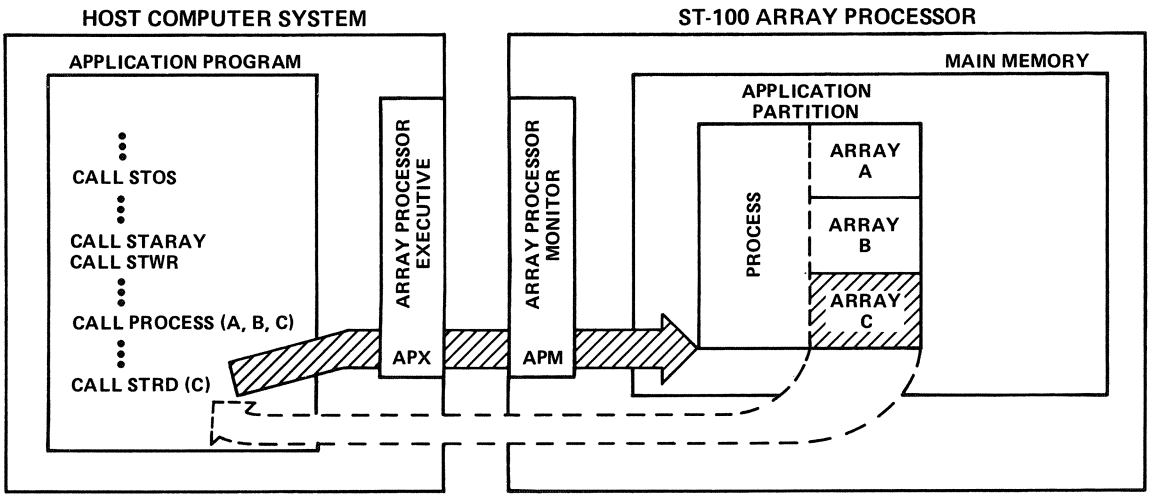


Figure 2-14. Reading the Processed Data

5. When array processor use is complete, the application program releases the application partition for use by other applications (figure 2-15).

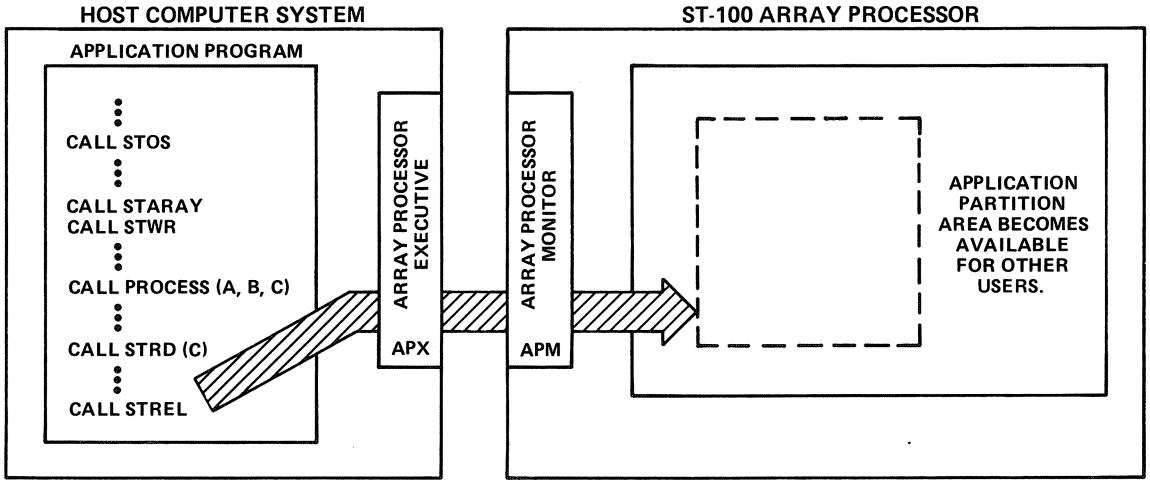


Figure 2-15. Releasing the Partition

The ST-100 Array Processor is capable of handling processes from multiple applications within the same host computer or from different host computers in a multiple-host configuration.

SECTION 3
COMPONENT DESCRIPTIONS

The ST-100 Array Processor is a system of hardware components and software components for the development and execution of ST-100 processes. This section describes each of the components.

HARDWARE COMPONENTS

The ST-100 Array Processor hardware consists of multiple processors and memories interconnected to allow the independent flow of data and arithmetic processing (figure 3-1). It has four independently programmable processors dedicated to external data flow (host/peripheral input/output), internal data flow, arithmetic processing, and resource synchronization and management.

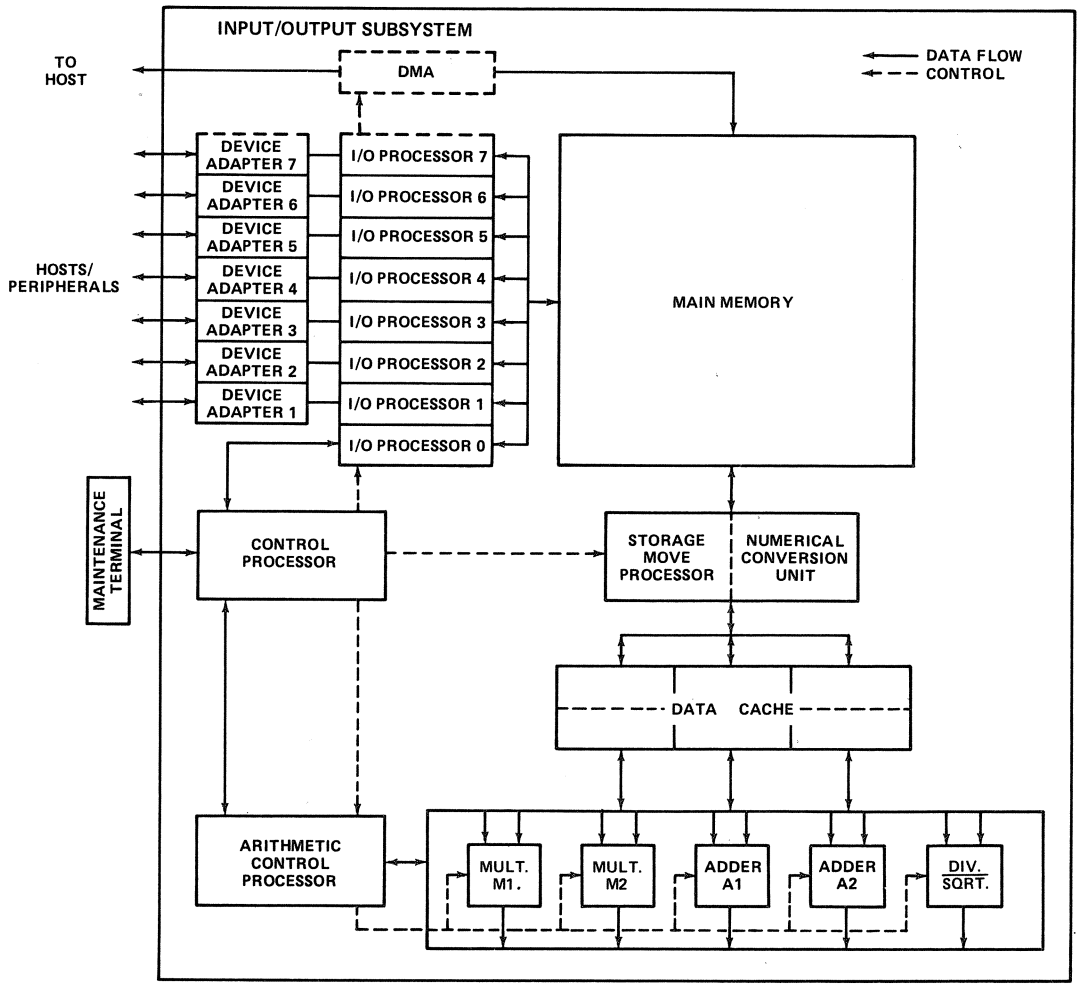


Figure 3-1. ST-100 Hardware Components

External data flow between the host or peripherals and the ST-100 main memory is controlled by input/output processors and device adapters comprising the input/output subsystem. Data movement within the ST-100 hierarchial memory system is managed by a user-programmable storage move processor. The ST-100 arithmetic section is controlled by a user-programmable arithmetic control processor. A hierarchial data memory system consists of external storage devices, a large main memory, and a high-speed, random-access partitioned data cache. In addition, each processor has its own register sets and program control memories.

An additional programmable maintenance terminal is provided to power-up, monitor, and initialize diagnostic programs for the ST-100 system.

Input/Output Subsystem

The input/output subsystem consists of up to eight input/output processors. One processor is dedicated to the system, while the remaining seven can be connected to external devices. These devices can be hosts or peripherals. Each of these processors can run at rates up to 12.5 Mbytes per second. Each external device is connected to the system through a device adapter, which matches the external-device hardware characteristics to those of the ST-100 processor. These input/output processors are controlled by the control processor. The device adapter can be used to connect to host computer channels, DMA devices, and other peripheral devices.

Control Processor

The control processor schedules all of the ST-100 activities. It initiates and synchronizes the loading and execution of all the other processors in the ST-100 system. This control is accomplished through the use of both software and special hardware features. The special hardware permits the chaining of commands to both the storage move processor and arithmetic control processor. This allows the control processor to synchronize and overlap the execution of these processors.

The control processor consists of two Motorola 68000 microprocessors clocked at 80 nsecs. These processors are connected to their own memory system, which consists of a shared and a non-shared portion. The shared portion is 192K bytes, while each processor has its own 32K-byte memory. The memory system can be accessed at 120 nsecs.

Each Motorola 68000 serves specific functions: one executes the monitor, controls the input/output subsystem, the maintenance terminal, dead start, and the other Motorola 68000. The second Motorola 68000 executes ST-100 processes. This processor can also communicate with main memory, the storage move processor, and the arithmetic control processor.

Main Memory

Main memory serves as the communication mechanism between external devices and the ST-100. It has three ports and an aggregate data rate of up to 100 megabytes per second. These ports are assigned to the input/output subsystem, the data cache, and a direct DMA. the input/output port can operate at 25 Mbytes per second, while the DMA and cache ports operate at a rate of 100 Mbytes per second. The data width of each port is 32 bits.

Main memory is eight-way interleaved to allow a 32-bit word transfer every 40 nsec. The memory has a 32-bit byte-address space, which gives a theoretical one billion word memory. However, with current chip and packaging technology, the ST-100 is capable of eight million words using 64K dynamic RAMS. Each 32-bit word has an additional 7 bits for error correction, which allows single-error correction and double-error detection.

Main memory can also be addressed in byte, half-word, and full-word modes through the storage move processor.

Data Cache

The data cache is the internal working data memory. It consists of three physical non-interleaved banks. Each bank can be divided into two logical sections by the control processor, thus producing six logical sections.

The control processor can assign any combination of these six logical sections to either the storage move processor or arithmetic control processor. Four cache memory accesses can be made every cycle (40 nsec), one by the storage move processor and three by the arithmetic control processor. This memory structure allows double buffering of data from main memory and also high operand flow to and from the arithmetic unit of the ST-100.

The data cache has 32 bits plus parity and cycles at a 40 nsec rate. Each physical bank consists of 16K words of 40-nsec-access memory. However, addressing has been provided to permit increasing these banks to 65K words each.

Storage Move Processor

The storage move processor controls data movement between main memory and the data cache. In addition, the processor controls format conversion between data in main memory and the data cache. Main memory can contain data in various formats, while the data cache operates in the ST-100 internal formats. The ST-100 internal formats are 2's complement 32-bit integer and 32-bit floating point (1 sign bit, 8 exponent bits, 24-bit fraction including the hidden bit). The storage move processor is a powerful integer (and logical) CPU that has the ability to create sophisticated address sequences for both main memory and the data cache. The processor is microprogrammed by STAR or the user through the Macro Assembly language.

The storage move processor contains three arithmetic and logical units (ALU) and a numerical conversion unit (NCU). The ALU's generate address and loop control for main memory and the data cache.

- Main memory address-generation is performed using a 32-bit ALU (can generate a 32-bit address). This unit also contains a 16-bit multiplier, 32-bit shifter, population count accumulator, and sixteen 32-bit general-purpose registers. Results of ALU operations can be tested for branch control.

- Data cache addressing is controlled by a 16-bit address for any of the three cache banks. This ALU has a bit-reversal shifter (for FTT address generation) and eight 16-bit general purpose registers. This ALU also controls bank selection of the data cache.

- The third ALU provides loop control for the two address generation ALU's. This ALU is 16 bits wide with eight 16-bit registers. Results of this ALU's operations can be tested for branch control.

- The NCU is a three-stage, pipelined format-conversion unit, which can reformat data as it is transferred between main memory and the data cache.

The storage move processor is controlled using an 80-bit wide micro word, which allows the three ALU's to run in parallel, thus maximizing data throughput. Processor memory capacity is 1K words, with addressing provided to permit increasing the capacity to 4K. As an additional feature, the storage move processor can be used as a powerful integer computer.

The processor is loaded and controlled by the control processor. Upon completion of a task, it issues an interrupt to the control processor. Any abnormal conditions within the storage move processor are also reported through interrupts to the control processor.

Arithmetic Section

The arithmetic section controls and performs floating point arithmetic in the ST-100. It operates only on data in the data cache, obtaining operands and returning results to the multiple cache memory banks. Thus, the arithmetic section only operates on the STAR internal 32-bit data formats.

The arithmetic section consists of a computation subsection coupled to the arithmetic control processor.

The computation subsection consists of data interchange and arithmetic units. The interchange unit facilitates univeral flow of operands between the data cache and arithmetic elements. The unit also allows for universal distribution of operands between the arithmetic elements without having to return intermediate results to the data cache.

The arithmetic units include two adders, two multipliers, and a divide/square root section. The adders and multipliers are three-step pipelines, with each step executing at the clock rate of 40 nsec. The divide/square root section is non-pipelined and requires 13 clock periods to complete. Thus, when the adders and multipliers are all in operation, the ST-100 can operate at 100 megaflops. The adders can perform fix-to-float/float-to-fix operations, subtract, add, and test-and-branch operations. These operations can be performed in signed or magnitude mode. Both the adders and multipliers can operate under program control, in either rounded or truncated mode.

The arithmetic control processor consists of four ALU's. Three are used for address generation to and from the data cache, while the fourth is used for loop control. Each ALU has eight 16-bit registers, and the loop control ALU has test-and-branch capability.

Re-entrant subroutine calling to any arithmetic macro is facilitated by the use of a 1K x 16-bit stack within the arithmetic control processor. This stack is used to save parameters and the program counter prior to calls to other macros. Thus, a user who is writing macros can use any existing macro in an efficient manner.

The arithmetic control processor is a mircoprogrammed device with a 4K x 128-bit microprogram memory with parity. Addressing has been provided to permit increasing this memory to 16K. The wide control word allows a high degree of parallel operation in each 40 nsec cycle. The unit can perform four integer ALU operations, four floating-point arithmetic operations, one test-and-branch operation, plus three memory references in each cycle.

The arithmetic control processor controls a crossbar switch between the data cache and the computation subsection. This switch enables a process or another macro to treat the data cache banks logically rather than physically. Thus, the memory banks can be assigned to the three address generators in any order. This facilitates the logical movement of data in the data cache for different calculations without time consuming physical movement.

Maintenance Terminal

A maintenance terminal with diskette-input capability attaches directly to the control processor. Its use includes ST-100 system loading and initiation (deadstart), ST-100 diagnostic program loading and execution independent from the host computer, and ST-100 error logging. The terminal can be accessed locally from the terminal keyboard or remotely through a modem to allow remote technical assistance.

Packaging

The ST-100 packaging system is designed with access to all the major components for ease of maintenance. The unit is designed for ambient temperature operation, requiring only normal computer room cooling (even for the VLSI logic).

The ST-100 uses a cabinet with the following approximate dimensions: 5 ft. H x 4-1/2 ft. W x 2-1/2 ft. D.

SOFTWARE COMPONENTS

ST-100 Array Processor software includes development software, production software, and maintenance software.

Development Software

Development software enables users to code ST-100 processes and macros, and generate process load modules ready for execution. It includes the following software products:

- Array Processor Control Language (APCL) and compiler
- Macro Assembly language and assembler
- Linker
- Debugger
- Simulator
- Library maintenance
- Application support library

All of the products except the application support library are coded in FORTRAN and execute in the host computer system. Macros in the application support library are object modules ready for linking with process object modules.

Array Processor Control Language (APCL)

Array Processor Control Language (ACPL) is used to write ST-100 processes, which are the program mechanisms for performing operations in the ST-100 Array Processor. Processes written in APCL are compiled by the STAR APCL compiler, which produces a process object module and related FORTRAN host process subroutine (HPS). The process object module is input for the STAR linker, which produces a process load module suitable for execution in the control processor in the ST-100. The FORTRAN HPS module may be linked to the host FORTRAN application programs that call the ST-100 process for execution. HPS provides the necessary interface to load the process for execution when the application program calls the process.

APCL statements are a Star Technologies, Inc. defined subset of ANSI FORTRAN 77 with extensions to control the unique architectural features of the ST-100 (see the appendix for a complete list of supported statements). The language has been extended with statements to support the ST-100 hierarchical memory structure. In addition, service requests to the array processor monitor can be made through subroutine calls. These service requests include macro initialization and synchronization, peripheral input/output, and host coordination.

The APCL compiler produces a listing of the compiled source input, host FORTRAN language process subroutine (HPS) source statements, and the process object code. The compiler places the HPS source module and process object module in appropriate host libraries according to user-supplied directives.

Macro Assembly Language

The Macro Assembly language is used to write macros for the arithmetic control processor and storage move processor. A meta-assembler concept is used, so that the same assembler can serve both processors. The assembler accepts free format input and directives. The directives include the ability to define text substitution macros.

Assembler output includes a source listing with errors and cross references, object code for the target processors, and information to define the relationship between cache memory banks. This cache memory relationship is used in the Array Processor Control Language to determine the cache cross-bar control.

Linker

The STAR linker collects relocatable modules for the control processor, arithmetic control processor, and storage move processor, and links them into a load module for execution in the ST-100. These modules are obtained from Star Technologies or user-created libraries. In the linking process, references between macros and processes or other macros are resolved.

The linker produces a memory map of the load module, diagnostic information, error messages, and information to direct the array processor monitor.

The process load module consists of Motorola 68000 program code, arithmetic control processor microprograms, and storage move processor microprograms. The linker also places these load modules in appropriate host libraries under user-controlled directives.

Debugger

The STAR debugger facility controls the execution of either the ST-100 or STAR simulator when debugging macros for the arithmetic control processor or storage move processor. The debugger operates in interactive or batch mode. It allows the user to examine, modify both data and program memories, set breakpoints, and request execution. When a breakpoint is encountered, the user can examine the contents of the memories and registers, and also request dumps of the contents of the array processor.

Simulator

The STAR simulator assists the user in the development of macros (microprograms) for the arithmetic control processor and storage move processor. The simulator runs on the host computer and does not require the ST-100 Array Processor hardware. The simulator is a bit-for-bit replica of the arithmetic control processor and storage move processor; thus both numerical results and program timing information can be obtained.

The simulator accepts load modules from the linker; however, these load modules are specifically linked for use in the arithmetic control or storage move simulator.

Library Maintenance

Library maintenance facilities are used to access, create, and maintain Star Technologies or user-developed libraries. These libraries can contain processes, macros, or load modules. The library facilities provide a common interface for the Array Processor Control Language, macro assembler, linker, host executive, and simulator to access the libraries. This capability also enhances the ability to transfer ST-100 applications between dissimilar hosts.

The library facilities have the ability to add, delete, and replace modules within libraries and also to create, merge, and catalogue libraries.

Application Support Library

The STAR application support library is a set of standard macros supplied by Star Technologies, Inc. for the arithmetic control and storage move processors. These macros are used in the development of ST-100 processes. A process, written using the Array Processor Control Language, calls these macros using the standard FORTRAN subroutine syntax.

Storage move macros accomplish the following functions:

- Move arithmetic macros from main memory to the arithmetic control processor microprogram-memory
- Move data between main memory and the data cache with format conversion.
- Integer and logical operations on data in main memory or the data cache.

Arithmetic control processor macros, which only operate on data in the data cache, will have the following functions:

- General vector arithmetic (real and complex)
- Vector-to-scalar arithmetic
- Vector logical operations
- Matrix operations
- Transform operations
- Filtering operations

A list of all of the macros is given in the appendix.

Production Software

Production software consists of the array processor executive (APX) and array processor monitor (APM). It links the host computer system and ST-100 Array Processor to execute ST-100 processes.

Array Processor Executive (APX)

The STAR array processor executive enables a host application program to communicate with the ST-100 Array Processor. The communication is via a set of FORTRAN-callable subroutines in the executive. These executive subroutines contain all of the logic necessary to interface the user application FORTRAN program with the array processor. The executive is bound and linked with the user application program and executed within the context of the user's address space.

These executive subroutines enable the user to request array processor resources, define ST-100 main memory arrays, control data movement to and from the ST-100, call processes, and release ST-100 resources.

The array processor executive supports concurrent access to the array processor by multiple application programs within a host computer.

Array Processor Monitor (APM)

The STAR array processor monitor resides in the ST-100 control processor and controls activities within the system. The monitor communicates through the St-100 input/output subsystem with multiple hosts. This mechanism

allows the monitor to control data transmission between host computers and the main memory of the ST-100. Main memory allocation and protection is supervised by this monitor.

Each main memory application partition contains data and process code. The execution of these processes is scheduled on a priority basis. The monitor loads the process into the control processor as soon as sufficient control processor memory is available. Execution of this process is then initiated upon completion of the currently executing process.

When a process is initiated, the monitor controls the activation of all the resources within the ST-100. These resources include the storage move processor, the arithmetic control processor, and any peripherals attached to the ST-100. Commonly-used process support routines are available through monitor calls. When an application partition is released, accounting information is returned to the host.

The monitor can communicate with the maintenance terminal to provide error logging and status information. This information is available for display in the terminal. The monitor also can initiate reliability verification tests for the ST-100 during idle periods.

Maintenance Software

Maintenance software includes idle loop reliability tests, user confidence tests, and diagnostic programs.

Idle Loop Reliability Tests

Idle loop reliability tests are a set of test programs residing in the ST-100. They automatically execute when the ST-100 is idle (no process activity) to verify the useability of the ST-100. The resultant status is provided to the host computer.

User Confidence Test

The STAR user confidence test is a user-callable process that verifies the usability of the ST-100 in a production environment. Host FORTRAN application programs may call the process to verify the ST-100. Upon completion of the process, a go or no-go status is returned to the application program.

Diagnostic Programs

STAR diagnostic programs are included in the ST-100 system to aid in fault isolation in the event of ST-100 failures. Diagnostic tests are provided for the following hardware elements:

- Control processor
- Storage move processor
- Input/output subsystem
- Arithmetic control processor
- Data cache
- Main memory
- Control processor random access memory (RAM)

Each test may be further divided into subtests that are selectable and repeatable. Operating procedures are implemented by menus on the ST-100 maintenance terminal. Directives, status, and fault isolation messages are in plain English. Hardware failures are isolated to the field replaceable unit. In addition, a packet of detailed error information is provided for further failure analysis.

APPENDIX A
ARRAY PROCESSOR CONTROL LANGUAGE FEATURES

The ST-100 Array Processor Control Language (APCL) is a STAR-defined subset of the American National Standard Programming Language FORTRAN, ANSI X3.9-1978 (FORTRAN 77) with extensions to control the unique architectural features of the ST-100 Array Processor.

The FORTRAN 77 features implemented in APCL include:

Language Element	Types Implemented
Constant	Integer
	Real
Data types	Integer
	Real
Variable	Scalar
	Array (main memory and cache memory only)*
Expression*	Arithmetic
	Relational
	Logical
Intrinsic functions	Convert a real number to its absolute value (ABS)
	Convert an integer number to its absolute value (IABS)
	Convert a real number to an integer (INT)
	Convert an integer to a real number (REAL)
	Convert a 32-bit real number to an integer (IFIX)
	Return the maximum value from a list of integers (MAX0)
	Return the maximum value from a list of real numbers (AMAX1)

*Subset FORTRAN 77 implementations

Language Element	Types Implemented
Intrinsic functions (cont.)	Return the minimum value from a list of integers (MIN0)
	Return the minimum value from a list of real numbers (AMIN1)
	Return the integer remainder of the division of one integer by another integer (MOD)
	Return the real remainder of the division of one real number by another real number (AMOD)
	Return the result of a bitwise logical AND of two integers (IAND)
	Return the result of a bitwise logical OR of two integers (IOR)
	Return the result of a bitwise one's complement of an integer (INOT)
	Return the result of a bitwise left logical shift of an integer (ISHFT)
Statements	INTEGER
	REAL
	DATA*
	Assignment
	CALL (macros and service requests only)*
	CONTINUE
	DO*
	GOTO*
	IF*
	RETURN
	END

*Subset FORTRAN 77 implementations

The major language extensions included in APCL are:

Extension	Description
PROCESS statement	Defines the process name and argument list
MAINMEMORY, CACHEMEMORY, and LOCALMEMORY statements	Define data in ST-100 main memory, data cache memory, and local memory within the control processor
Service requests	Calls to the array processor monitor (APM) to control various architectural features of the ST-100: • Wait for SMP queued requests to complete (WTSMP) • Wait for ACP queued requests to complete (WTACP) • Wait for both SMP and ACP queued requests to complete (WTAP) • Synchronize the ACP and SMP queues (SYNCAP) • Set bank selection (SYNCAP) • Voluntarily abnormally terminate the process (ABORT) • Read data cache memory into local memory (RCACHE) • Read main memory into local memory (RMAIN) • Read a storage move register into local memory (RSMREG) • Read an ACP register into local memory (RACREG)

APPENDIX B
ARRAY PROCESSOR EXECUTIVE FEATURES

The ST-100 array processor executive (APX) is the interface between host FORTRAN application programs and the ST-100 Array Processor. Application programs access the ST-100 and execute ST-100 processes via FORTRAN calls to the APX. The calls, which are host independent, are as follows:

Call Name	Description
STOPN	Open the array processor
STSCH	Schedule an application partition
STOS	Open the array processor, and schedule an application partition
STFLT	Select a fault processing mode in the array processor
STDUMP	Specify a file to receive an array processor memory dump in the event of an abort condition
STARAY	Allocate array data areas in the application partition
STEQU	Define a new array by equating it to an old (established) array in the application partition
STWR	Transmit data to the array processor
STRD	Read data from the array processor
STWIO	Wait for the completion of previous STWR and STRD operations
STDOM	Specify an array processor search domain (for use with STLOD)
STLOD	Load a process into the array processor
STRUN	Execute a process in the array processor
STWEX	Wait for completion of specific process executions previously initiated
STPURG	Deallocate processes or arrays within the application partition

Call Name	Description
STREL	Release the application partition
STCLOS	Close the array processor for the host application
STETXT	Expand an array processor return error status code into an 80-character message
STFTXT	Expand an STRUN fault status code into an 80-character message

APPENDIX C
APPLICATION SUPPORT LIBRARY

The ST-100 application support library is a library of macros and processes supplied and planned by Star Technologies, Inc. for use by user application programmers and process programmers. The standard macros follow:

Macro Name	Description	Type
SMM2C	Move main memory data to cache memory	SMP
SMC2M	Move cache memory data to main memory	SMP
SMXMC2	Move complex vector (main memory to two cache memory subsections)	SMP
SMXMC1	Move complex vector (main memory to one cache memory subsection)	SMP
SMXCM2	Move complex vector (two cache memory subsections to main memory)	SMP
SMCTMC	Move constants table (main memory to cache memory)	SMP
SMSTMC	Move sine/cosine table (main memory to cache memory)	SMP
SMMW2R	Move main memory word to main memory address-generator register	SMP
SMCW2R	Move cache memory word to main memory address-generator register	SMP
SIMUL	Integer multiply utility	SMP
SVAND	Vector logical AND	SMP
SVOR	Vector logical OR	SMP
SVXOR	Vector logical exclusive OR	SMP
SVEQV	Vector logical equivalence	SMP
SVNOT	Vector complement	SMP
SVCLR/AVCLR	Vector clear	SMP/ACP
SVFILL/AVFILL	Vector fill	SMP/ACP

Macro Name	Description	Type
SMC2C/AMC2C	Vector move (cache memory to cache memory	SMP/ACP
SVROM/AVROM	Vector reverse order move (cache memory to cache memory)	SMP/ACP
SPOPC	Bit vector population count	SMP
SDR2M	Dump SMP registers to main memory	SMP
SDR2C	Dump SMP registers to cache memory	SMP
ADR2C	Dump ACP registers to cache memory	ACP
AVNEG	Vector negate	ACP
AVABS	Vector absolute	ACP
AVNABS	Vector negative absolute	ACP
AVADD	Vector add	ACP
AVSUB	Vector subtract	ACP
AVMUL	Vector multiply	ACP
AVDIV	Vector divide	ACP
AVSQRT	Vector square root	ACP
AVSADD	Vector scalar add	ACP
AVSMUL	Vector scalar multiply	ACP
AVSSQ	Vector signed square	ACP
AVMSA	Vector multiply and scalar add	ACP
AVSMA	Vector scalar multiply and add	ACP
AVMA	Vector multiply and add	ACP
AVLN	Vector natural logarithm	ACP
AVEXP	Vector exponential	ACP
AVSIN	Vector sine	ACP
AVCOS	Vector cosine	ACP
AVTAN	Vector tangent	ACP
AVATAN	Vector arctangent	ACP

Macro Name	Description	Type
AVAT2	Vector arctangent of A/B	ACP
AVINT	Vector truncate to integer	ACP
AVRNUM	Vector random numbers	ACP
ASVE	Sum of vector elements	ACP
ASVESQ	Sum of vector element squares	ACP
ADOTP	Dot product	ACP
AMAXE	Maximum element in a vector	ACP
AMAXME	Maximum magnitude element in a vector	ACP
AVMAX	Vector maximum	ACP
AVMIN	Vector minimum	ACP
AVCLIP	Vector clip	ACP
ALVGT	Logical vector greater than	ACP
ALVGE	Logical vector greater than or equal	ACP
ALVEQ	Logical vector equal	ACP
ACVA	Complex vector add	ACP
ACVS	Complex vector subtract	ACP
ACVM	Complex vector multiply	ACP
ACVMCO	Complex vector multiply conjugate output	ACP
ACVSM	Complex vector scalar multiply	ACP
ACDOTP	Complex dot product	ACP
ACFFT	Complex FFT	ACP
ARFFT	Real FFT	ACP
AIRFFT	Inverse real FFT	ACP
ACONV	Convolution/correlation	ACP
AWLEV	Weiner-Levinson filter	ACP

Macro Name	Description	Type
ANMO	Normal moveout	ACP
ALINT	Linear interpolation	ACP
AQINT	Quadratic interpolation	ACP
ANZCN	nth zero crossing negative	ACP
ANZCP	nth zero crossing positive	ACP
APOLY	Polynomial expansion	ACP
ARECF	Recursive filter	ACP
AMXTRN	Matrix transpose	ACP
AMXMUL	Matrix multiply	ACP
AMVM3	Matrix vector multiply (3x3)	ACP
AMVM4	Matrix vector multiply (4x4)	ACP

In addition to SMP macros and ACP macros, the application support library contains a standard user confidence test process for use in verifying the operational status of the ST-100.

APPENDIX D
ST-100 PROCESS EXAMPLE

This appendix shows an example of an ST-100 process, the FORTRAN host process subroutine (HPS) generated by the STAR Array Processor Control Language (APCL) compiler, and a host FORTRAN application program that executes the process.

ST-100 PROCESS

The process is a simplified vector-add used for easy illustration in this appendix. Lines of coding preceded by the letter C are comment lines explaining aspects of the coding. Calls are made to the following macros: SMM2C, AVADD, and SC2MM. Process APCL coding follows:

```
      PROCESS VADD(A,IA,B,IB,C,IC,LENGTH)
C
C DEFINE MAIN MEMORY ARRAYS PASSED FROM HOST
C
      MAINMEMORY
      REAL A(LENGTH),B(LENGTH),C(LENGTH)
C
C DEFINE CACHE MEMORY ARRAYS
C
      CACHEMEMORY
      REAL (C1T,AC(8000)),(C2T,BC(8000)),(C3T,CC(8000))
C
C DEFINE LOCAL VARIABLES
C
      LOCALMEMORY
      INTEGER LENGTH,IA,IB,IC,ISUBA,ISUBB,ISUBC,N
      DATA ISUBA,ISUBB,ISUBC/1,1,1/
C
 10   N=MIN0(LENGTH,8000)
C
C MACRO CALL FOR ST-100 SYNCHRONIZATION AND BANK SELECTION
C
      CALL SYNCAP(000000)
C
C MACRO CALLS TO MOVE ARRAY-A FROM MAIN MEMORY TO CACHE MEMORY
C
      CALL SMM2C(A(ISUBA),IA,4,AC,1,N)
      CALL SMM2C(B(ISUBB),IB,4,BC,1,N)
      CALL SYNCAP(101010)
C
C MACRO CALLS FOR VECTOR ADD AND SYNCHRONIZATION
C
      CALL AVADD(AC,1,BC,1,CC,1,N)
      CALL SYNCAP(010101)
C
```

```
C MACRO CALL TO MOVE RESULTS FROM CACHE MEMORY TO MAIN MEMORY
C
      CALL SMC2M(C(ISUBC),(IC,4,AC,1,N)
C
      ISUBA=ISUBA+8000+IA
      ISUBB=ISUBB+8000+IB
      ISUBC=ISUBC+8000+IC
      LENGTH=LENGTH-8000
      IF(LENGTH.GT.0) GOTO 10
      END
```

FORTRAN HOST PROCESS SUBROUTINE (HPS)

The FORTRAN host process subroutine generated by the APCL compiler during VADDU process compilation follows. Lines beginning with the letter C are comments.

```
      SUBROUTINE VADDU(I1,I2,J2,I3,I4,I5,I6,I7,I8,I9)
      INTEGER I1,I2,J2,I3(3),I5(3),I7(3),I6,I8,I9
      INTEGER ILOCI,ILOCR,IMAINI,IMAINR
      SAVE IPID
C
C DEFINE ARGUMENT TYPE VARIABLES
C
      DATA ILOCI,ILOCK,IMAINI,IMAINR/257,258,1025,1026/
C
C LOAD PROCESS IF NOT PREVIOUSLY LOADED
C
      CALL STLOD(I1,I2,IPID,'VADD')
      IF(I2.NE.0) RETURN
C
C EXECUTE PROCESS PASSING ARGUMENTS
C
      CALL STRUNW(I1,I2,J2,IPID,IMAINR,I3,ILOGI,I4,IMAINR,I5,
     -            ILOCI,I6,IMAINR,I7,ILOCI,I8,ILOCI,I9)
      RETURN
      END
```

HOST FORTRAN APPLICATION PROGRAM

An example of a host FORTRAN application program that executes the VADDU process follows. Lines beginning with the letter C are comments.

```
      PROGRAM EXMPL1
      DIMENSION A(10000),B(10000),C(10000),LAID(3),LBID(3),LCID(3)
C
C THIS PROGRAM CALLS A VECTOR-ADD PROCESS WITHIN THE ARRAY
C PROCESSOR TO ADD ARRAYS A AND B, THEN PLACE THE RESULTS
C IN ARRAY C.
C
      INC=1
      ILUN=50
```

```
C
C INTERNALLY OPEN ANY AVAILABLE ARRAY PROCESSOR AND ESTABLISH
C AN APPLICATION PARTITION.
C
      CALL STOSW(ILUN,ISTAT,'DSIZE',40,'PSIZE',4,'ELIMIT',1,
     1'JLIMIT',10)
      IF(ISTAT.NE.0) GO TO 1000
C
C ILUN=LOGICAL UNIT NO. OF ARRAY PROCESSOR
C ISTAT=STATUS OF CALL
C DSIZE=MAIN MEMORY SIZE IN 1K INCREMENTS
C PSIZE=MAIN MEMORY PROCESS SIZE IN 1K INCREMENTS
C ELIMIT=SINGLE PROCESS EXECUTION LIMIT (SECS.)
C JLIMIT=TOTAL PROCESS EXECUTION LIMIT (SECS.)
C
C DEFINE MAIN MEMORY ARRAYS IN PARALLEL TO HOST ARRAYS
C A,B, AND C.
C
      CALL STARAY(ILUN,ISTAT,LAID,10000,'REAL')
C
C LAID=3-ELEMENT ARRAY THAT DEFINES THE LOGICAL ID
C AND STARTING SUBSCRIPT FOR SUBSEQUENT CALLS
C
      IF(ISTAT.NE.0) GO TO 1000
      CALL STARAY(ILUN,ISTAT,LBID,10000,'REAL')
      IF(ISTAT.NE.0) GO TO 1000
      CALL STARAY(ILUN,ISTAT,LCID,10000,'REAL')
      IF(ISTAT.NE.0) GO TO 1000
C
C TRANSMIT DATA FROM HOST TO MAIN MEMORY
C
      CALL STWRW(ILUN,ISTAT,A,10000,LAID)
      IF(ISTAT.NE.0) GO TO 1000
      CALL STWRW(ILUN,ISTAT,B,10000,LBID)
      IF(ISTAT.NE.0) GO TO 1000
C
C EXECUTE PROCESS (VADDU SUBROUTINE GENERATED BY APCL COMPILER).
C
      CALL VADDU(ILUN,ISTAT,JSTAT,LAID,INC,LBID,INC,LCID,INC,10000)
      IF(ISTAT.NE.0) GO TO 1000
C
C TRANSMIT DATA FROM MAIN MEMORY TO HOST.
C
      CALL STRDW(ILUN,ISTAT,C,10000,LCID)
      IF(ISTAT.NE.0) GO TO 1000
C
C RELEASE JOB PARTITION WITHIN THE ARRAY PROCESSOR.
C
      CALL STREL(ILUN,ISTAT,'ALL')
      IF(ISTAT.NE.0) GO TO 1000
```

```
C
C CLOSE THE ARRAY PROCESSOR.
C
      CALL STCLOS(ILUN,ISTAT)
      IF(ISTAT.NE.0) GO TO 1000
      STOP
C
C ERROR ROUTINE.
C
 1000  ...
      END
```

STAR TECHNOLOGIES, INC.
One S.W. Columbia, Suite 1212
Portland, Oregon 97258
503/227-2052