

Burroughs

BSP

FAULT-TOLERANT FEATURES

## CONTENTS

	<u>Page</u>
1. INTRODUCTION	C-1
2. SYSTEM DESCRIPTION	C-3
3. ERROR DETECTION	C-5
File Memory Control	C-5
Control Processor	C-7
Control and Maintenance Unit	C-9
Parallel Processor Control Unit	C-11
Parallel Processor	C-11
Arithmetic Element	C-12
Output Alignment Network	C-12
Memory Interface and Parallel Memory	
Address Parity Checks	C-12
4. MAINTENANCE LOG FEATURES OF THE BSP	C-15
5. INSTRUCTION RETRY	C-17
File Memory Control	C-17
Control Processor	C-18
Parallel Processor	C-18



## 1. INTRODUCTION

The Burroughs Scientific Processor (BSP) is one of the first machines of its class to employ the features of single-bit error correction of memory data and instruction retry. The system design embodies extensive error-detection capabilities that are the necessary foundations for effective retry. Additionally, all major data paths are protected with a modified Hamming code which provides single-bit error detection and correction and double-bit error detection. These features, coupled with the failure log capabilities, provide the scientific user and the Burroughs field engineer with state-of-the-art maintenance capabilities.

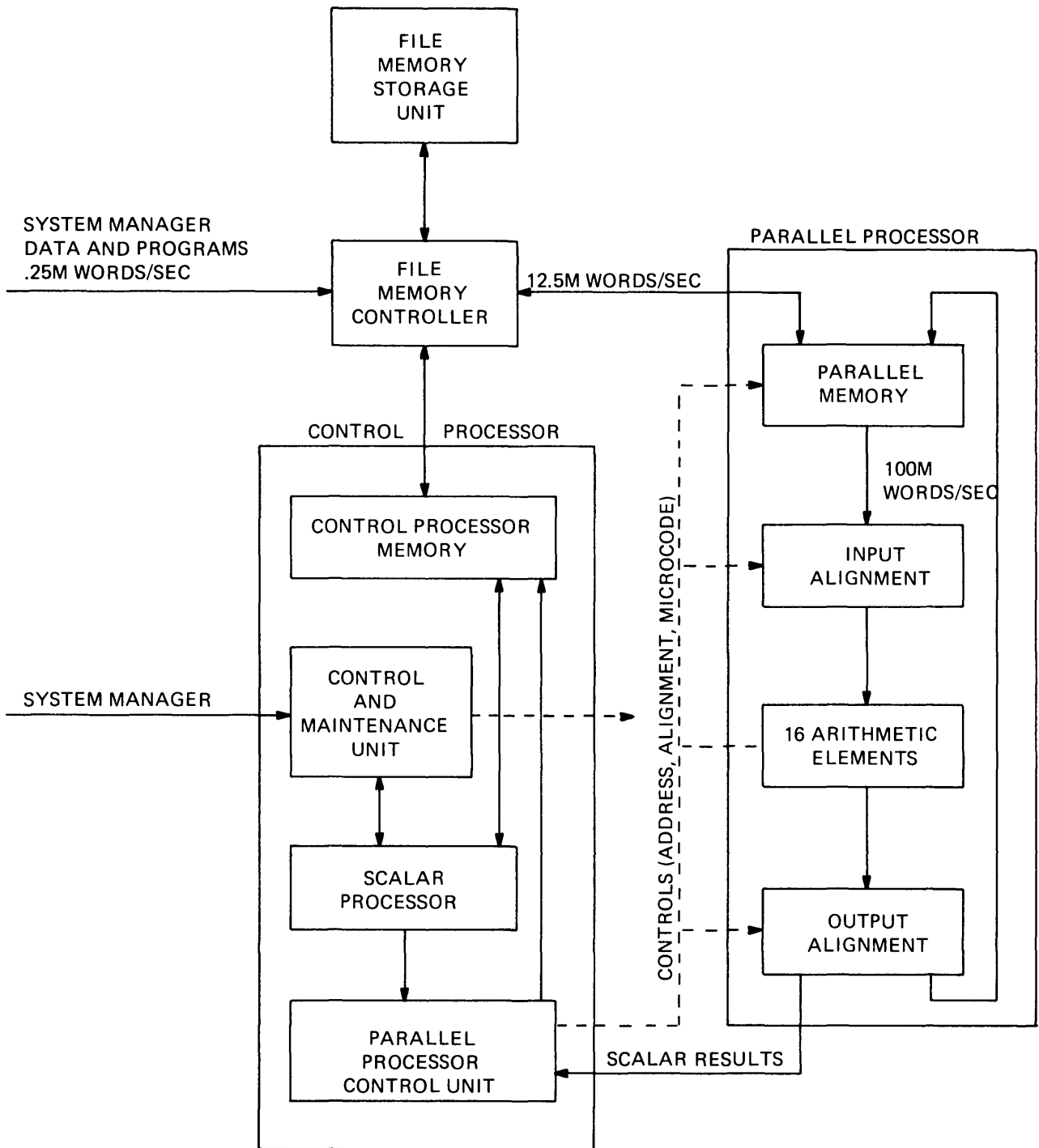


Figure 1. BSP System

## 2. SYSTEM DESCRIPTION

The Burroughs Scientific Processor (BSP) consists of a control processor (CP), 16 arithmetic elements (AE), a parallel memory (PM) consisting of 17 memory units, an alignment network to interface the AE's and PM, a file memory (FM), and a file memory control unit. The components are shown in Figure 1.

The control processor is a high-speed asynchronous element of the BSP that provides the supervisory interface to the system manager in addition to controlling the parallel processor and the file memory. The control processor consists of a scalar processor unit, a parallel processor control unit, a control memory, and a control and maintenance unit.

The control processor executes some serial or scalar portions of user programs utilizing an arithmetic element similar to the 16 arithmetic elements in the parallel processor.

The scalar processor unit processes all operating system and user program instructions, which are stored in control memory. It operates at a clock frequency of 12.5 MHz and performs up to 1.5 million floating-point operations per second. Array instructions and some scalar instructions are transferred to the parallel processor control unit, which queues them for execution on the parallel processor.

The parallel processor control unit receives array instructions from the scalar processor unit. The instructions are validated and transformed into microsequences that control the operation of the parallel processor.

The control memory is used to store portions of the operating system and user programs as they are being executed. It is also used to store program data values that are operands for those instructions executed by the scalar processor unit.

The control memory is a 4K-bit bipolar memory with a 160-ns cycle time. Four words can be accessed simultaneously. Capacity of the memory is 262K words; each word consists of 48 data bits and 7 bits for error detection and correction.

The control and maintenance unit (CMU) serves as the direct interface between the system manager and the rest of the control processor for initialization, communication of supervisory commands, and maintenance. It communicates with the input/output processor of the system manager.

The CMU has access to most data paths and registers of the BSP, so that it can perform state analysis and circuit diagnostics under control of maintenance software running on the system manager.

The parallel processor performs array-oriented computations at high speeds by executing 16 floating-point operations simultaneously in its 16 arithmetic elements. Data for the array operations are stored in a parallel memory consisting of 17 memory modules. Parallel memory is accessed by the arithmetic elements through a memory alignment network.

At any time, all of the arithmetic elements are executing the same instructions on different data values. The arithmetic elements operate at a clock frequency of 6.25 MHz and are able to complete the most common arithmetic operations in two clock periods.

The parallel memory consists of from .5 to 8 million words organized internally into 17 modules. Like the control processor memory, it is a 4K-bit bipolar memory. Each word contains 48 data bits and 7 bits for error detection and correction. The rate of data transfer between the parallel memory and the arithmetic elements is 100M words per second.

The file memory (FM) is the high-speed secondary storage device on the BSP system. It utilizes high-speed, charge-coupled devices (CCD) as its storage media – expandable from 4 to 64 million words. It is loaded by the system manager with BSP code and data files for execution of a task on the BSP. The FM is controlled by the file memory control unit, which provides queueing of I/O requests, priority operations, logical to physical address conversion and extensive error detection of data and of I/O descriptors.

### 3. ERROR DETECTION

#### FILE MEMORY CONTROL

The file memory control (FMC) is the controller for all BSP I/O operations. It performs single-bit correction and double-bit detection on data passing through it and extensive checking on descriptors issued to it (Figure 2).

Descriptors are received by the FMC from either the system manager or the control processor. System manager-issued descriptors are used to load BSP code and data files to the file memory. The descriptor specifies the logical file ID and the relative word address in the file. The descriptor and any subsequent data is transferred across a 17-bit wide data path which includes a parity bit. The FMC checks parity and, if an error is detected, it returns a result descriptor to the system manager indicating a parity error. The first three 16-bit transfers are interpreted by the FMC as the descriptor. All subsequent data is treated as specified by the descriptor (data). When three 16-bit data transfers are received correctly, the FMC generates a Hamming code for the word and transfers it to the file memory.

The actual physical location where the data is stored in the FM is a function of the dynamic address translator (DAT). The DAT is basically a software loadable table in the FMC that can be loaded only by the scalar processor unit (SPU). It is used to convert the logical address in the descriptor to a physical address in the file memory.

The DAT consists of two memories. The book descriptor memory is a 64-word by 28-bit memory; the page descriptor memory is a 2048-word by 12-bit memory. Each memory contains a parity bit.



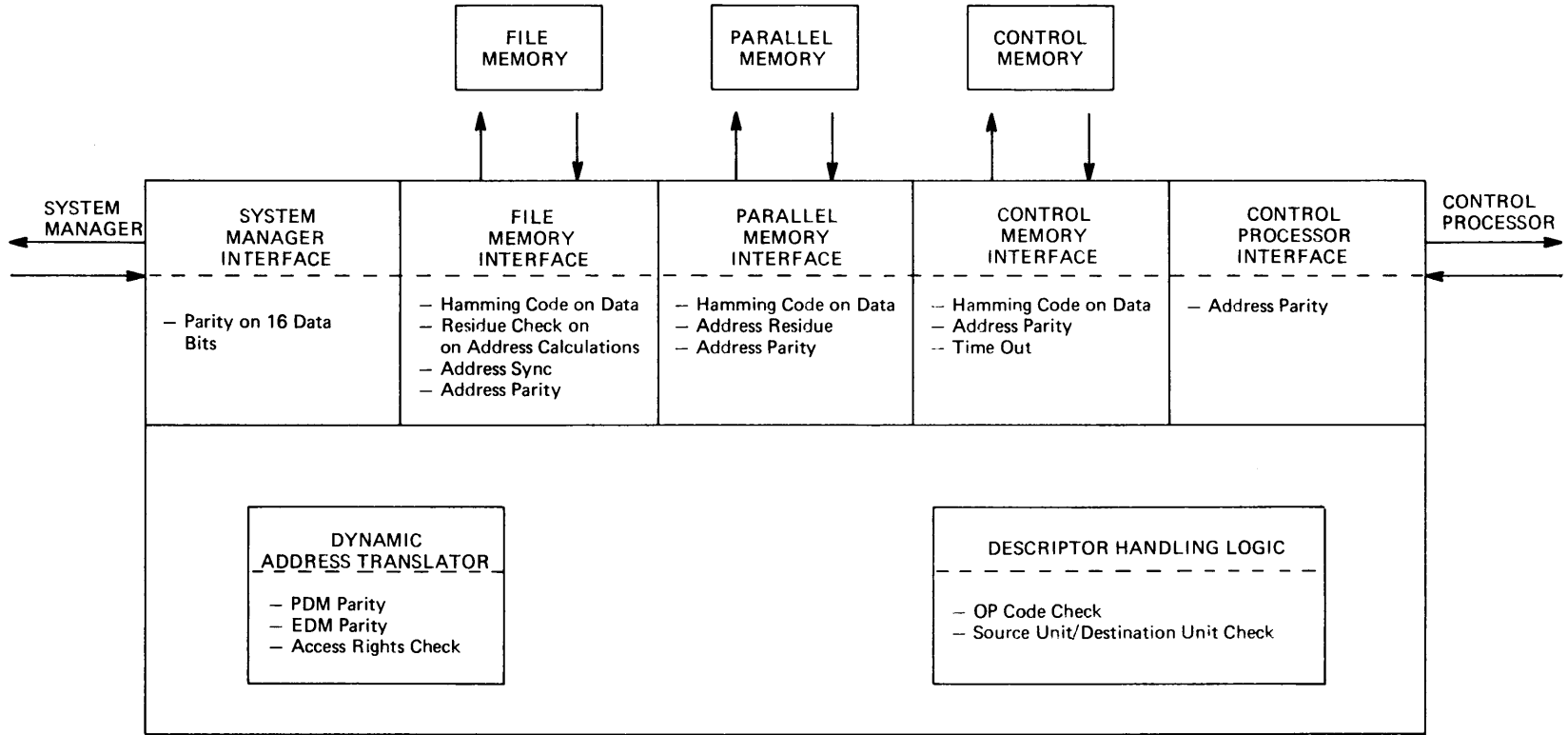


Figure 2. File Memory

If a parity error is detected during I/O operations between the system manager and the file memory, a result descriptor is returned to the system manager indicating the error.

Descriptors issued by the scalar processor control transfers between the file memory and parallel memory, control memory and file memory, and control memory and processor memory. Each of the memory units stores data with a Hamming code. As the data passes through the file memory controller, all single-bit errors are corrected and logged and double-bit errors initiate a retry and are logged.

Additionally, extensive checks are done on the I/O descriptor issued by the SPU. The illegal source/destination is a check for illegal transfers between the various memories. For example, it is not possible to transfer between the page descriptor memory and the book descriptor memory. Errors of this type cause a maintenance log entry, a descriptor retry and an error result descriptor to the SPU when retry has been exhausted.

Other checks by the FMC are an address parity check on addresses sent to the control memory, access rights and book limit checks on the DAT, charged-coupled device (CCD) address sync error, and others. Each of these, if detected, causes a maintenance log entry, initiates a retry, and if retry is unsuccessful (exhausted retry count), sends a result descriptor to the SPU.

## CONTROL PROCESSOR

Error detection in the control processor is shown in Figure 3. The scalar processor unit (SPU) receives both instructions and data from the control memory (CM). All instructions and data stored in the CM have a Hamming code attached.

During data fetches from CM to the SPU, single-bit errors are corrected, double-bit errors cause a system interrupt. Both types of errors cause maintenance log entries.

During instruction fetches from CM to the SPU, single-bit errors are corrected and double-bit errors cause a retry of the memory fetch cycle. Instructions are stored in the instruction file buffer (IFB) along with a parity bit for each 8-bit byte. As instructions are processed, they are transferred from the IFB to the pre-instruction register, where the parity is checked. Parity errors cause a retry of the instruction as well as a maintenance log entry. Instructions loaded in the preinstruction register are checked for illegal opcode, illegal variant field and privilege use. If any of these are detected, the instruction is retried and a maintenance log entry is created.

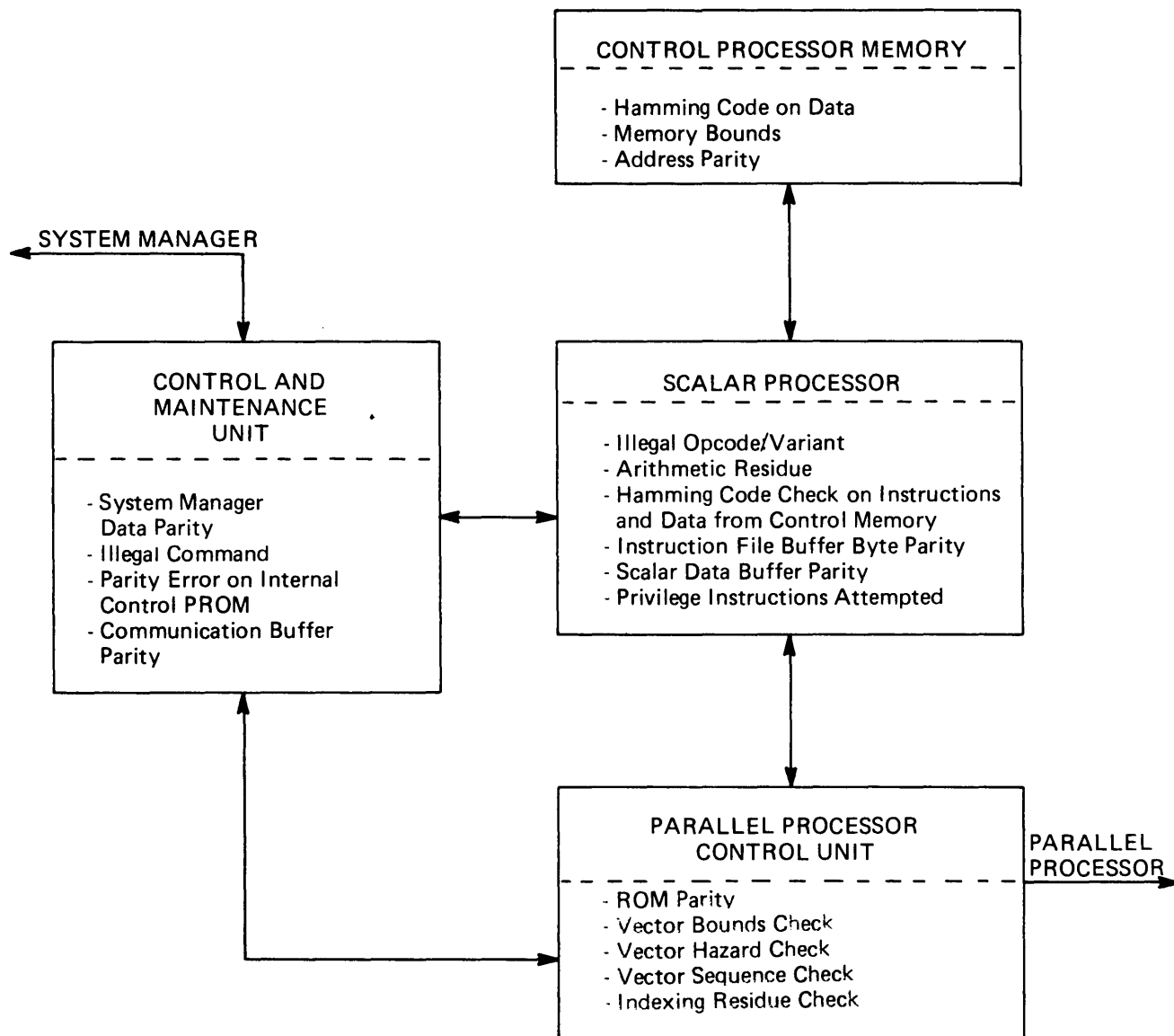


Figure 3. Control Processor

Various other checks are made throughout the control processor, all of which cause an interrupt and a maintenance log entry. For example, the arithmetic unit of the scalar processor is identical to the arithmetic units of the parallel processor. All arithmetic units use a residue check to verify their arithmetic computations; the instruction variant fields are checked for illegal conditions; the control memory instruction address calculations are checked for negative results or overflow conditions; the control memory data address calculations are checked that they are on 4-word boundaries; and various ROMs are checked for parity errors, etc.

The data path from the control memory to the scalar processor is protected with Hamming code. Data is transferred to the scalar data register (a set of 16 registers) along with an overall parity bit. Data fetched from the SDB is parity checked.

All failures detected in the control processor result in a maintenance log entry along with the appropriate interrupt.

#### CONTROL AND MAINTENANCE UNIT

The control and maintenance unit (CMU), as shown in Figure 3, is the interface unit to the system manager. During system operation, it provides the communication path between the operating system in the system manager and the operating system in the BSP. For maintainability purposes, it provides diagnostic access and control to the various subunits of the BSP.

The CMU receives instructions and data from the system manager via a 17-bit interface which includes parity. Detection of a parity error sends an error result descriptor to the system manager and generates a maintenance log entry.

Instructions are loaded into a command register where an illegal command check is done. The execution of commands is implemented with ROMs which include a parity bit. Illegal instructions or ROM parity errors generate a maintenance log entry and send an error result descriptor to the system manager.

The path for communications between the two operating systems is implemented by a communications buffer (CB). Data stored in the CB includes a parity bit. Errors detected during buffer transfers generate a maintenance log entry and send an error descriptor to the system manager or cause an interrupt to the operating system in the control processor.

The CMU does not contain retry features. In general, detected errors are reported to the system manager or control processor for processing by software. Additionally, a maintenance log entry is generated and sent to the system manager.

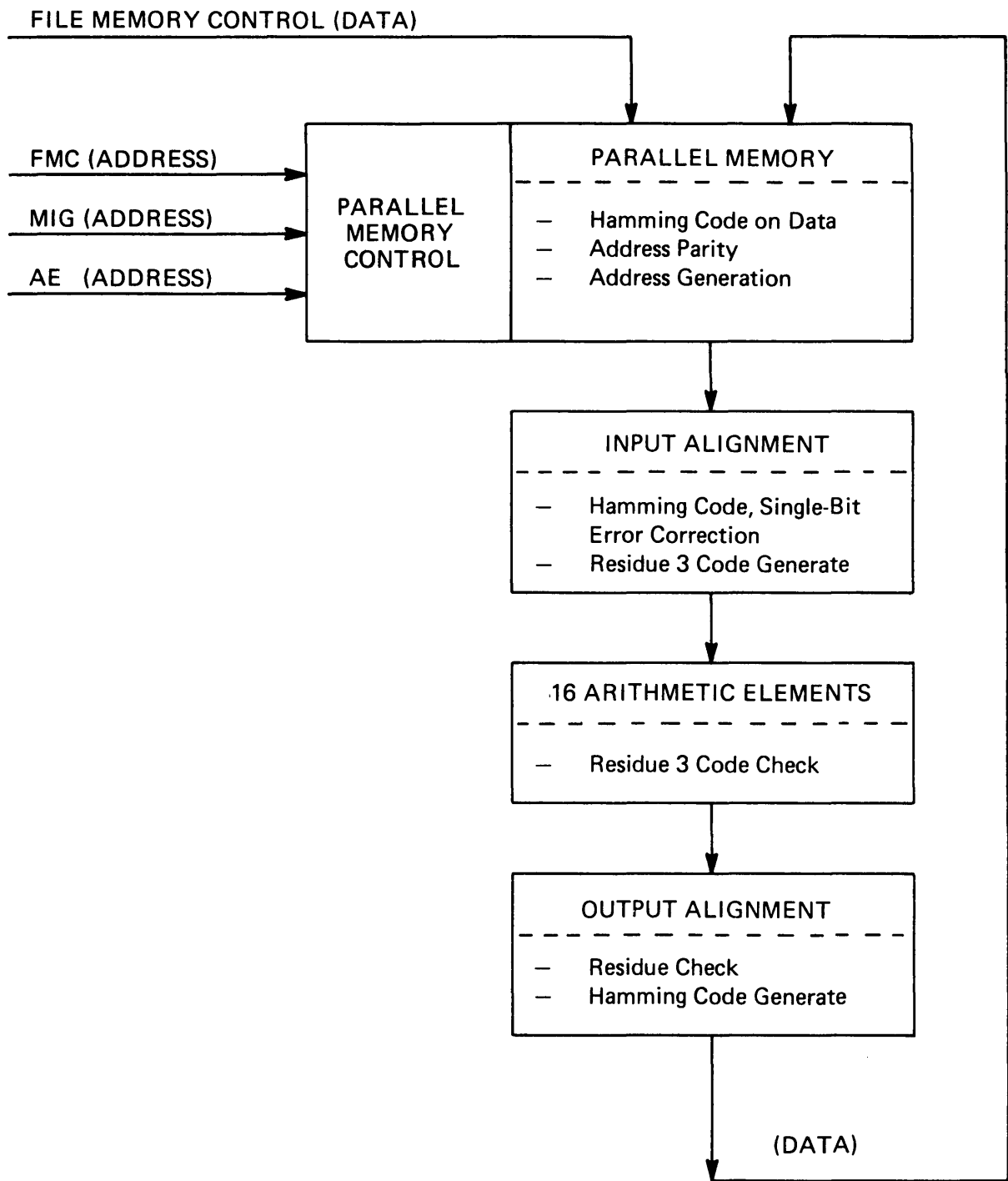


Figure 4. Parallel Processor

## PARALLEL PROCESSOR CONTROL UNIT

As shown in Figure 3, the parallel processor control unit (PPCU) receives vector instructions from the scalar processor. The vector instruction stream is first checked by the vector initialization and validation unit (VIV). The VIV checks the bounds (a check that vector results are in vector space), checks the vector sequence (a check that operations are set up correctly) and some internal operation checks (residue checks). Errors detected by the VIV are retried by the VIV and generate a maintenance log entry.

After the various checks, the vector instructions are queued in the vector parameter queue (VPQ) until they become operational. The VPQ is protected with parity. Detection of a parity error causes an interrupt to the operating system and a maintenance log entry. This particular failure is not retryable.

Vector instruction decode and operation is primarily implemented with ROMs. ROMs are used extensively to generate the control signals to each of the subunits of the array. Extensive parity checks are done on the ROM outputs. Errors detected on the ROMs are retried as a vector retry. Any errors detected generate a maintenance log entry.

## PARALLEL PROCESSOR

The parallel processor (Figure 4) is a pipeline processor that receives its instructions, memory addresses, and alignment controls from the parallel processor control unit (PPCU). Data in the parallel memory (PM) is referred to as vectors. Data is loaded from the file memory to the parallel memory under control of the FMC. The vector is transferred to the AEs, where computational work is done.

The vector to be transferred to the AEs is stored in the 17 memory units of the parallel memory. The PPCU calculates four initial address values, which are sent to the alignment network. The alignment network uses these values to compute the full set of 17 parallel memory addresses. The address calculation is checked by generating an extra set of values that are compared for equality against the initial values.

The input alignment network (IAN) receives a set of 16 tags from the PPCU. Each tag is the number of the MU that an AE is to receive data from during the alignment of a vector. The set of tags, along with two parity bits, is generated by the PPCU and checked by the alignment network.

After the data has passed through the IAN, the Hamming code is checked, and any single-bit errors are corrected. Double-bit failures are detected and result in a retry of the entire vector operation. After the Hamming code check, a residue 3 code is added to the data before it is transferred to the AE.

## ARITHMETIC ELEMENT

The arithmetic element (AE) receives data from the input alignment network (IAN) and microcode control from the parallel processor control unit (PPCU), as shown in Figure 4. The microcode is dispersed across three of the four AE logic islands. Each of the three islands receives a parity bit as part of the microcode. The AE checks the parity and reports any detected failures to the PPCU. If a failure is detected, the PPCU initiates a vector retry operation.

The data from the IAN includes a 2-bit, modulo-3 residue code for both the exponent and mantissa (a total of four bits of residue code). The residue code is generated by the IAN when it passes the data to the AEs.

During arithmetic operations, if the result is an intermediate value, then the residue is checked when the value is again used by the AE. If the result is a final value transferred to the output alignment network (OAN), then the OAN checks the residue and reports any failures to the AE, which in turn reports the failure to the PPCU, which in turn initiates a vector retry.

Any errors detected generate a maintenance log entry.

## OUTPUT ALIGNMENT NETWORK

Results calculated by the AE are transferred to the output alignment network (OAN) for alignment and transfer to the parallel memory. The OAN first checks the residue code and reports any failures to the AE, which in turn reports the failures to the PPCU, which initiates a vector retry. The OAN next generates the Hamming code for the 48 data bits and then transfers the data to the specified memory unit of the parallel memory. Certain types of vector operations (random fetch and random store) require that the AEs generate the memory address. In this case, the AE generated address is transferred to the OAN, which appends an overall parity bit to the address and then forwards the address to the parallel memory.

Any errors detected generate a maintenance log entry.

## MEMORY INTERFACE AND PARALLEL MEMORY ADDRESS PARITY CHECKS

The three units that generate addresses for the parallel memory (PM) are the FMC (for I/O operations), the memory index generators (MIG) (for run-of-the-mill vector operations), and the arithmetic elements (for special vector operations such as random fetch and random store). (See Figure 4.)

Addresses originating from the FMC or AE include a parity bit. The parallel memory control (PMC) receives the address and checks the parity. If an error is detected, the I/O operation will be retried or the vector operation in the array will be retried.

Assuming no errors are detected, the PMC sends the address to the PM without the parity bit and saves a copy of the parity bit. When the PM receives the address, it generates a parity bit which is returned to the PMC. The PMC compares the bit from PM with the saved parity bit.

If the requester was an AE and a failure is detected, and the request was for a memory read cycle, then the PPCU is notified and a vector retry is initiated. If the requester was the FMC and a failure is detected, and it was a read request, then the FMC is notified and an I/O retry is initiated.

If a failure is detected and the request is for a write operation, then regardless of who the requester is, the failure is considered nonrecoverable, and a retry is not possible because by the time the error is detected, the memory write cycle has already begun.

All addresses originating from the MIG have a parity bit generated and saved by the PMC. The PMC again transfers the address to the PM and receives a parity bit back from the PM, which is compared with the saved parity. If a failure is detected for read operations, a vector retry is initiated; for write operations, the failure is not retryable.





#### 4. MAINTENANCE LOG FEATURES OF THE BSP

The maintenance log features of the BSP provide a status recording of the pertinent hardware state when a failure is detected. For each of the error detection mechanisms in the system, a record of 47 bits is transferred to the system manager, where the data is stored on the system file. The data is then available as a history of detected failures on the BSP.

The logged information is extremely valuable for early identification of failure trends and, in particular, for identifying intermittent failures that have been successfully retried. An analysis of the data can identify which subsection of the system is suspect.

A typical log entry for a residue error in the array would identify the AE number, whether the residue error was a mantissa or exponent error, whether the failure was detected as internal to the AE or at the output of the AE, and whether the failure was an intermittent or solid failure. In the case of solid failures, a diagnostic routine can be used to isolate the failure to a small replaceable subsection of logic.

For intermittent failures, further analysis of the maintenance log will be necessary. The analysis function will be performed by the cumulative diagnostic error analyzer program (CDEAP). This program will run on the system manager and provide analysis of maintenance log entries to abstract intermittent errors and find commonality among intermittent failures. The program interacts with the system design data base to isolate failures to a replaceable subsection of logic.



## 5. INSTRUCTION RETRY

### FILE MEMORY CONTROL

File memory control (FMC) I/O operations can be initiated from either the system manager or the control processor. Those initiated by the system manager can only be transfers between the system manager and the file memory. They are not retryable, but any failure detected will cause an error result descriptor to be sent to the system manager.

The start of any I/O operations by the control processor (CP) begins with the CP transferring the address of the descriptor to the FMC. The address may be queued by the FMC, depending on priority and the busy state, or may start operation immediately.

For the FMC to start an I/O operation, it first sends the descriptor address to control memory (CM). The CM returns the descriptor to FMC, where it is loaded into the descriptor word memory. Next, various checks are made on the validity of the descriptor, such as illegal opcode, illegal source or destination unit. Any failures detected will cause a retry of the descriptor fetch operation.

When the descriptor has been decoded and the actual data transfer begins, numerous other checks are made — both in the data and control logic. For example, CM bounds check (bounds register used to delineate user area from supervisor area), page descriptor memory parity check, book descriptor memory parity check, file memory address parity check, double-bit Hamming code error, etc.

For any retryable error (there are some not retryable such as unit not-ready, single-bit errors that have been corrected), the operation in process is stopped. Then the descriptor is refetched from CM and the operation is retried.

## CONTROL PROCESSOR

The control processor retry capabilities are limited to the retry of the control memory instruction fetch cycle.

As instructions are fetched from control memory to the instruction file buffer (IFB), single-bit errors are corrected and double-bit errors cause a retry (refetch) of the suspect memory word. As the instructions are loaded into the IFB, a parity bit is appended to each 8-bit byte. As the instructions are transferred from the IFB to the preinstruction register (IRP), the byte parity is checked. Any failures detected here result in a reload of the entire IFB (refetch all instructions from memory to IFB) and a retry of the transfer from IFB to the IRP.

Once the instruction is in IRP, additional checks are made. These are an illegal opcode check, illegal variant field check, and a privilege execution attempted check. If any of these failures are detected, IFB is reloaded from memory, IRP is reloaded from the IFB, and another attempt (retry) is made to execute the instruction.

Once an instruction has been successfully loaded and decoded, any other failures detected during its execution (such as a residue error) cause an interrupt to the operating system in the BSP.

## PARALLEL PROCESSOR

Retry in the parallel processor begins with the detection of an error. The parallel processor control unit (PPCU) controls the retry operation for the array as well as certain failures detected within itself.

Throughout a vector operation, the PPCU keeps track of the progress of the operation. Essentially, it keeps a count of the number of elements in the vector that have been successfully processed. When a failure is detected, this count is saved. The PPCU then restarts the vector operation from the beginning, and notifies the parallel memory to inhibit the store of data into memory. When the count of the repeated operation reaches the saved count, the memory inhibit is removed and the vector operation continues.