Burroughs

BSP

FILE MEMORY

BSP

BURROUGHS SCIENTIFIC PROCESSOR

FILE MEMORY

CONTENTS

ABSTRACT

The Burroughs Scientific Processor file memory exploits recently developed charge-coupled device memory technology to provide file access performance balanced to the needs of high-speed computation. It can transmit sequential file records at a sustainable rate of over 10 million words per second and can access a random record with an average delay of only 660 microseconds, thus exceeding conventional disk and drum speeds by an order of magnitude. It eliminates operating system softward overhead for routine record access by offloading these functions into the controller. As the only input-output device directly connected to the BSP, the file memory provides temporary storage for up to 64 million 48-bit words of program and data files, and serves as a staging area for files to be transferred to or from conventional peripherals and permanent storage.

1. INTRODUCTION

The speed of scientific computers is limited fundamentally by the rate at which data can be supplied to the arithmetic units. Although large, fast, random-access memories are attractive from performance considerations, their cost effectiveness declines for capacities above a few million words. However, many important scientific applications process data bases of tens of millions of words. Typically, the data represent values at points in a two-dimensional or three-dimensional grid, as in weather prediction and nuclear research, or coefficients of large matrices, as in structural analysis or linear programming. Algorithms typically access this data by one or more sequential sweeps across the grid or matrix.

Given these characteristic features of scientific data-base access, a two-level memory hierarchy, with a few million words of random-access memory backed up by an order of magnitude of more high-speed sequential access memory, could provide optimum performance at a reasonable cost. Unfortunately, the speeds of current generation supercomputers have considerably outstripped the speeds of sequential memory devices. For example, arithmetic speeds may approach 100 million operations per second (MOPS), but fast disks provide only 500 thousand operands per second, a compute/transfer ratio of 200:1. On the other hand, important algorithms have compute/transfer ratios in the range 20:1 to 5:1; indeed, the back substitution phase of linear equation solution has a ratio of only 1:1. Such algorithms would be severely I/O-bound by conventional devices.

A common pseudo-solution to the I/O problem has been to multiprogram an I/O-bound program with another program that is not I/O-limited. Although this uses the computing resources more efficiently, it does nothing to speed up the execution of the I/O-limited program. Thus, the multiprogramming approach is irrelevant to supercomputers, whose reason for being is to speed up the solution of problems consuming hours or days on conventional computers.

2. DESIGN GOALS

In order to provide balanced I/O performance for a scientific computer, the following design goals were established for the BSP file memory.

1. Since I/O delays are incompatible with the goal of maximum execution speeds for a single program, all BSP program I/O will be to a single secondary storage subsystem with balanced performance characteristics: the file memory. Files coming from or destined to lower speed peripherals will be independently spooled to/from file memory before the program starts or after it completes, concurrently with execution of other BSP programs.

2. The time to transfer one operand in and one operand out of central memory will not exceed the time for 10 floating-point operations. This is satisfied for a unidirectional transfer rate in excess of 10 million words per second, given the 50 million operations-per-second speed of the BSP.

3. File capacities of 8-16 times central memory size will be available. For a maximum main memory of 8 million words, this implies a maximum capacity of at least 64 million words.

4. The file system reliability must be consistent with continuous operations at maximum bandwidth. This implies file storage unit mean time to failure of at least 200 hours, with provision for transient error recovery and error logging for preventive maintenance.

5. Since system software necessary to issue input-output operations is scalar code that detracts from the floating-point payload of a number cruncher, the necessity for such code should be minimized. This suggests an asynchronous controller to perform routine "physical I/O" operations.

6. Since the use of assembly language contradicts the philosophy of Burroughs computers, the full efficiency of the I/O system must be available to the FORTRAN programmer. This suggests some extensions in BSP FORTRAN to permit unbuffered asynchronous I/O, as well as a storage and addressing structure compatible with FORTRAN record formats.

3. TECHNOLOGY

WHY NOT DISK?

Although off-the-shelf disk systems clearly have inadequate performance characteristics for supercomputers, a high-performance subsystem based on disk technology was considered.

One possible way of achieving the desired transfer rate is to operate many disks concurrently, as was done on the Burroughs ILLIAC IV. This approach requires sophisticated techniques for distributing related data among multiple disks and synchronizing the multiple transfers. These synchronization problems could be eliminated by connecting several read/write heads to a single disk, but a more fundamental problem remains.

The effective data-access rate from a disk-type device is ultimately limited by the mechanical access time, typically 10-40 milliseconds. Since the maximum size block transferred on any given access is limited by available memory, even an infinite number of disks or heads operated in parallel produces only a finite effective transfer rate. For example, even if the transfer rate were infinite, a disk with 33-millisecond access time would require two buffers of 0.3 million words each to provide a sequential throughput of 10 million words per second.

Another disadvantage of the multiple disk approach is the reliability requirement for a high bandwidth system. Current high performance disk technology attains error rates as low as one error per 10^{10} to 10^{12} bits transferred. But even at these rates, a 128-head system operating at 5×10^8 bits per second would fail about once per hour, which is intolerable for a system designed to service multi-hour jobs routinely.

CCD TECHNOLOGY

A solution to the performance requirements of the BSP secondary storage was found in an emerging semiconductor memory technology, specifically charge-coupled devices (CCD). The advantages of this technology include the following:

1. Availability. Competing technologies such as bubble memories or electron-beam storage tubes were not available in production quantities when the design was being settled. On the other hand, 9K-bit chips were in production from several vendors, with 16K-bit and 64K-bit versions in active development. Since CCD is based on well understood MOS semiconductor memory technology, this presented the least technical risk.

2. Economy. Cost projections indicated that CCD prices would drop dramatically as chip densities improved. And, although CCD would seem to be more expensive than bulk disk storage, when the engineering costs for a high-performance, relatively low-production-volume supercomputer I/O subsystem are included, CCD costs become competitive.

3. Easily paralleled. The transfer rate of a semiconductor memory can be made arbitrarily high just by operating multiple chips in parallel. Since chip transfer rates of about 5 million bits per second are available, it is practical to achieve the target transfer rates from a memory only a few words wide.

4. Low latency. Although high instantaneous transfer rate is easily achieved from parallelism alone, a high sustainable transfer rate requires buffers large enough to mask the access time. Since a CCD memory is essentially a large shift register, it shares with disk memories a rotational-type latency. However, access times below 500 microseconds are available with CCD technology. Thus, only moderate-sized buffers (24K - 64K words) are required to fully mask latency for double-buffered sequential transfers.

5. Reliability. Very good, chip-level reliability, coupled with easily implemented Hamming code single-error correction/double-error detection provides system reliability much superior to that available from off-the-shelf disks.

4. FILE MEMORY DESIGN

The file memory subsystem consists of two major sections, as illustrated in Figure 1. The file storage unit contains the CCD memory chips and addressing logic, and the file memory controller interfaces the storage system to both the BSP and the system manager.

FILE STORAGE UNIT

The initial production of the file storage unit (FSU) is being built for Burroughs by Fairchild Semiconductor, using their F464 65K-bit CCD chip. Each storage unit provides up to 16 million 56-bit words (48 data bits plus 8 bits for error correction), internally organized as 1 to 4 basic storage increments of 4 million words each.

The basic storage unit (BSU) operates at one of two clock frequencies. For access and transfer, the chips are operated at 3.1 megaHertz to provide two words every 160 nanoseconds with a maximum latency of 1.3 milliseconds. Nonaddressed modules are cycled at 1/4th that clock rate to conserve power while providing refresh necessary to retain data in this volatile storage.

The storage unit is designed to provide arbitrary length block transfers superimposed on a paged space allocation scheme. Consequently, the system can begin a sequential transfer at any address. When the memory address crosses a 16K-word boundary, the storage unit will continue transferring from the beginning of any other 16K-word block within the BSU without requiring an additional access delay.

FILE MEMORY CONTROLLER

The file memory controller (FMC) provides the programmability features of the file memory system, as well as the hardware interfaces. The data paths supported by the FMC are shown in Figure 1. There is a single half-duplex path (one direction at a time) between the FSU and the controller, operable at 12.5M words per second. The interface path to the system manager is buffered down to its 0.25M words per second maximum effective channel speed. Paths to the two BSP central memories operate at full FSU speed. Finally, the controller provides a utility block transfer path between BSP parallel memory and control memory, which can be operated concurrently with FSU to system manager transfers.

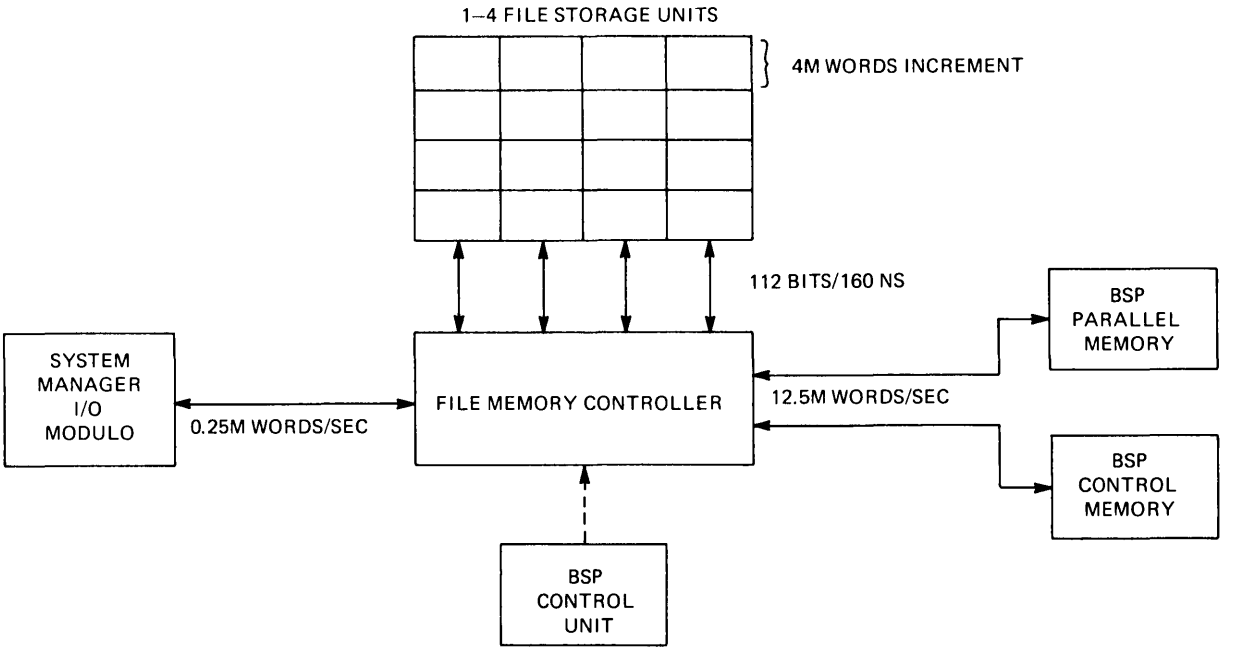


Figure 1. File Memory Organization

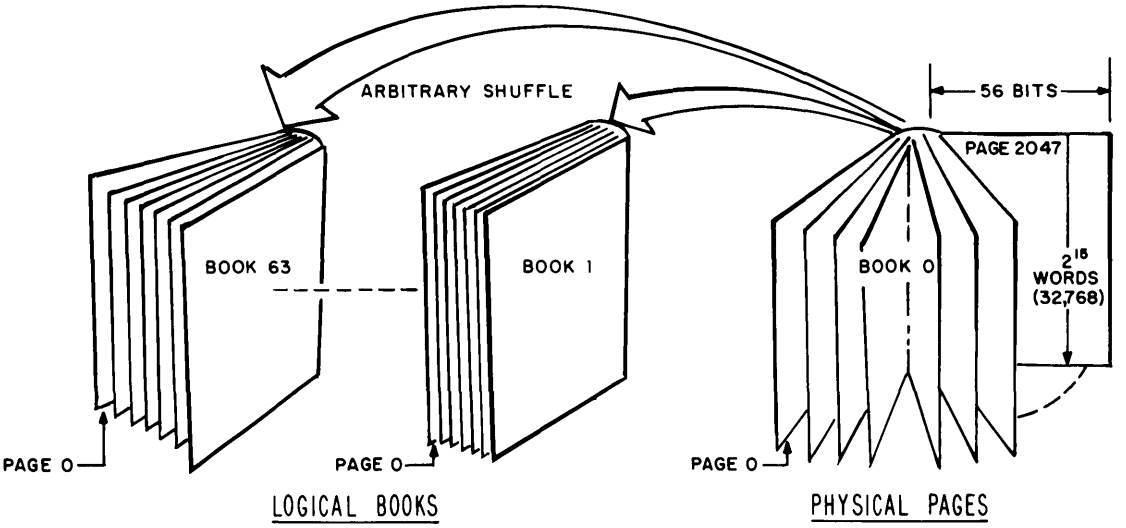
5. PROGRAMMABILITY CONSIDERATIONS

A major concern in the design of the file memory was to insure that the full I/O performance be made available to the FORTRAN programmer, without the necessity of complex assembly-language code or the overhead of serially-executed operating system software. As a result, the file memory controller provides hardware support for file storage space allocation, address translation, file protection, request priority queueing, error retry, and completion posting and synchronization.

SPACE ALLOCATION

Since a typical file is likely to require a relatively large fraction of file memory, the BSP system avoids the space fragmentation attendant with contiguous allocation by allocating file space in relatively small (16K word) discontiguous units called pages. Although this is similar to the multiple-extent allocation common with conventional disk systems, the BSP file memory offers significant enhancements.

Any paged allocation scheme requires an address translation or lookup to find the physical address associated with a given record of a given file. This lookup is typically performed by software on conventional systems, but on the BSP it is performed by hardware. Each file may be regarded as a book containing multiple pages. From the program's viewpoint, the pages are numbered consecutively, but from the file storage viewpoint, they may be physically scattered in a random way. Corresponding to each book is a table in the FMC local memory which associates a physical page address with each logical page in the book. This table is set up by the BSP supervisory software when the file is allocated; the FMC hardware then performs the logical-to-physical address translation for record accesses without further software intervention. (See Figure 2.)



A book contains an arbitrary collection of physical pages, logically renumbered from zero.

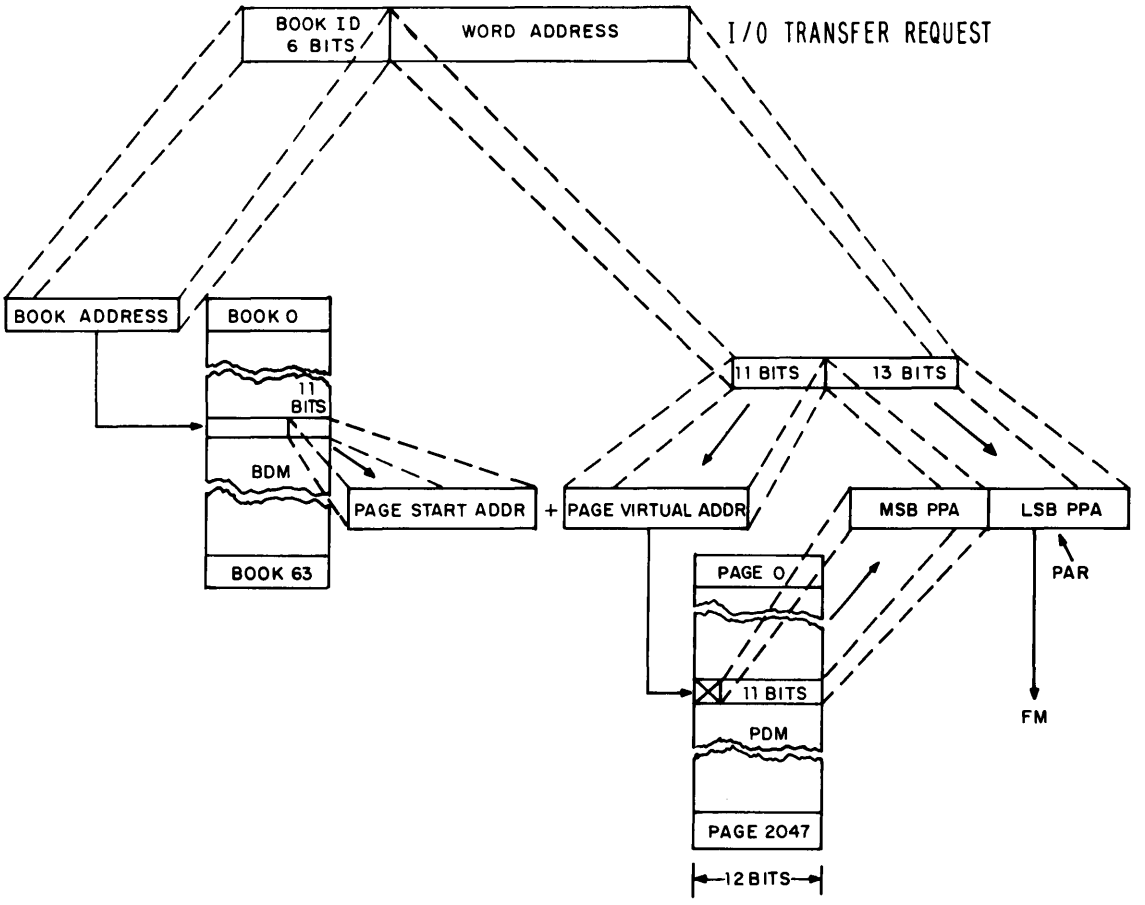


Figure 2. File Address Translation

FILE ADDRESSABILITY

Efficient sequential transfer requires that file memory transfers take place in fairly large blocks (24K - 64K words). On the other hand, programmer convenience dictates that the block size be related to natural problem dimensions, such as row size of a matrix, rather than to arbitrary hardware convenience. Consequently, the BSP file memory provides that the unit of transfer (block) be completely independent of the unit of allocation (page). In particular, a block transfer can begin at any word address and extend for an arbitrary number of words. The FMC hardware automatically switches to "next page" whenever the logical block crosses a physical page boundary. This mechanism allows a greatly simplified I/O request instruction; a program I/O request consists primarily of a book number (assigned to the file at the time it is allocated), a starting word address (relative to the logical beginning of the file), a block length, and a memory address.

FILE PROTECTION

A primary motivation for operating system intervention in I/O operations on conventional systems has been the need for software-implemented file protection. The BSP file memory, however, is equipped with a hardware file protection similar to the main memory protection mechanisms on conventional computers. The system provides that any combination of four access modes can be assigned on a file (book) basis: system manager read or write, BSP problem program read or write. If none of these are assigned, only the BSP supervisor (which runs in privileged mode) can access the files. Since the BSP runs only one program at a time, this provides effective isolation between files of the running program, files for other programs being copied to/from the system manager, and dormant files belonging to inactive programs.

REQUEST QUEUEING

Another significant source of operating system overhead on conventional machines is queueing and prioritizing I/O requests. The FMC again provides this function in hardware. Requests from either the system manager or a BSP program can be queued in the controller. System manager requests, having a much slower average transfer rate, receive priority but effectively interleave with BSP requests. BSP requests are normally honored first-in, first-out, but a high-priority mode is available for urgent requests such as true random access. The queue will retain up to 32 requests for asynchronous execution.

PROBLEM STATE I/O

As a result of FMC functional sophistication, the basic BSP I/O instruction can be executed safely and efficiently in prlblem mode, with no supervisor intervention whatever for error-free block transfers.

The BSP also provides a hardware synchronization mechanism, fully supported by the FMC, that also allows the complementary I/O completion posting to be accomplished without supervisor intervention. The BSP contains a number of bit registers called synchronizers, which function similarly to the semaphores introduced by Edsgar Dijkstra. Each I/O request has associated with it two such synchronizers, one of which is cleared to give the FMC permission to start the request (indicating that all processing of that buffer by the asynchronous parallel processor has completed) and another which is cleared by the FMC indicating that the I/O transfer has successfully completed. This latter synchronizer may be tested in problem mode when the program needs to reference that buffer. If it is already cleared, as will normally be the case for sequential buffered I/O, the program can continue without invoking the supervisor.

ERROR RECOVERY

An extensive component of conventional computing systems' input-output software is error-recovery procedures. The BSP file memory provides automatic retry on all errors not corrected by the error-correction code, and independently logs corrected as well as uncorrectable errors for future maintenance. Since all meaningful error-recovery procedures will have already been tried, the I/O error software needs only to classify the error and inform the running program, which will take whatever action was directed by the programmer.

6. SUMMARY

The BSP file memory is so called because it applies the performance and convenience associated with central memories to a file storage device. By this means, the performance bottleneck previously associated with record access in scientific programs has been effectively eliminated on the BSP. Thus, the BSP becomes the first supercomputer on which it is truly practical and efficient to process scientific programs with data spaces in the tens of millions of words.