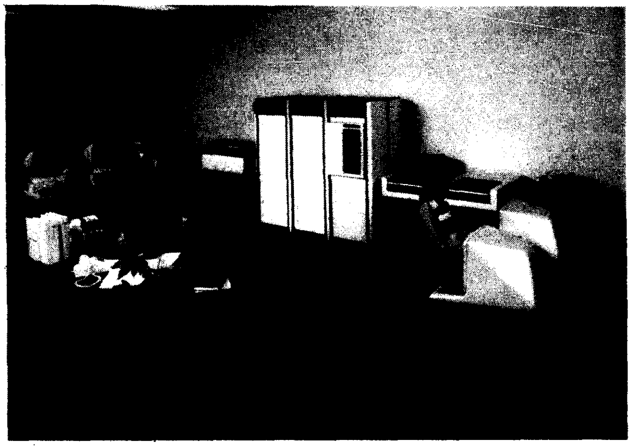
BTI 8000



The photo shows a BTI 8000 32-bit system with an optional 9-track magnetic tape drive in the rightmost of the three cabinets, a 900-lpm printer (left of the cabinets), two 254M-byte disk drives, and four terminals. (BTI does not include terminals in its product line.)

The BTI 8000, announced in June 1978, is a 32-bit, multiprocessor, multi-user, multi-language, and multifunction system. Modular in configuration, the 8000 is designed around a central bus with a 32-bit-wide data path and 16 slots for plug-in attachment of system resource modules. By varying the number and the mix of resource modules attached to the bus, a configuration can be selected that will closely match the price and performance requirements of an application. The system uses 32-bit architecture throughout and will support as many as 512 interactive users. Software for the 8000 provides a high level of security for users sharing a system.

MANAGEMENT SUMMARY

Even in the early days of marketing the 4000, BTI felt that to reach the general EDP marketplace, a totally new system with substantially greater processing power and I/O capacity was needed. Therefore, while continuing with the development of the 4000 and the 5000 (see Report M11-089-101), BTI embarked on a program which culminated in the announcement of the BTI 8000 at the 1978 National Computer Conference.

The BTI 8000 is based on a 32-bit, 67-nanosecond bus serving multiple processors, controllers for main memory modules, and peripheral processors which in turn control I/O device controllers, an approach which BTI calls "Variable Resource Architecture." Computational Processing Units (CPU's), Memory Control Units (MCU's), and Peripheral Processing Units (PPU's) can be plugged into the bus in any mix to match the requirements of the application. A minimum configuration requires at least one of each. The system's computational power, memory, and I/O resources are automatically pooled for efficient processing of the overall workload. If an application subsequently requires more computational power, memory, or I/O capacity, additional modules are plugged into the bus. Conventional manual regeneration of the monitor system (Sysgen) is not required when changing the hardware configuration—the system does it automatically under control of one front-panel switch. The 8000's hardware configuration is totally isolated from all user software, preserving a user's software investment as a system grows.

Background

BTI Computer Systems started out in 1968 as Basic Timesharing, Inc., a time-sharing service company serving

CHARACTERISTICS

MANUFACTURER: **BTI Computer Systems, Inc., 870 West Maude Avenue, Sunnyvale, California 94086. Telephone (408) 733-1122.**

BTI Computer Systems started in the San Francisco Bay area in 1968 as a time-sharing service company under the name Basic Timesharing, Inc., and took its present name in 1978. From this time-sharing experience, the company developed a series of interactive systems initially based on a modified Hewlett-Packard minicomputer. Today the company manufactures two product lines: a 16-bit single-processor system capable of supporting up to 32 users and a 32-bit modular multiprocessor system which can support over 500 users. BTI's manufacturing facility is in Sunnyvale, California, and U.S. sales offices are in Piscataway and Cherry Hill, New Jersey; Braintree, Massachusetts; Dallas, Texas; Minneapolis, Minnesota; Chicago, Illinois; St. Louis, Missouri; and Sunnyvale and Anaheim, California. BTI also has a European subsidiary, BTI Computer Systems (UK), Ltd., with a sales office in Slough, England, and a service office in Birmingham, England.

MODEL: **8000.**

DATE ANNOUNCED: **June 1978.**

DATE OF FIRST DELIVERY: **Scheduled for late 1979.**

NUMBER INSTALLED TO DATE: **—.**

DATA FORMATS

BASIC UNITS: **32-bit word and 8-bit byte.**

FIXED-POINT OPERANDS: **Operands can be single or double words, a character, or a field of from 1 to 32 bits.**

FLOATING-POINT OPERANDS: **Sixteen floating-point instructions deal with 64-bit, double-word operands, which include 11-bit biased exponents (10^{-154} to 10^{154}) and 52-bit mantissas (over 15 decimal digits). Double-precision floating-point operands of 128 bits are generated and manipulated by software with a precision of over 38 decimal digits.**

INSTRUCTIONS: **Machine instructions are all one word in length and reside on memory word boundaries. There are 174 machine instructions available in user mode (as opposed** the San Francisco peninsula. The experience gained in the next two years led to the development of a packaged proprietary time-sharing system capable of accommodating up to 16 users.

BTI's first formal product line, the BTI 3000 Series, an outgrowth of the company's assembled systems, was introduced in November 1972. This new system was based on a Hewlett-Packard 2100 minicomputer, chosen primarily because its user-microprogrammability enabled implementation of an efficient time-sharing facility in a system of relatively modest cost. In addition to the 48K bytes of 980-nanosecond core memory and the on-line disk storage, the basic 3000 system included 8 user ports, expandable to 16 ports, and the ability to extend the disk storage capacity beyond 9.6 megabytes by adding more disk controllers as required.

The 4000 Series, introduced in January 1975, was a continuation of the total hardware and software system established by the 3000 Series. The 4000 Series was based on the newer, more cost-effective, Hewlett-Packard 21MX minicomputer (Report M11-472-201) and initially consisted of three models: the 4000/10, 4000/20, and 4000/30. The new CPU's incorporated many system functions implemented in microcode. All of the 4000 Series models had 64K-byte core memories and differed in the type and amount of mass storage offered with each system. The 4000/10 used the same 2.4-megabyte disk drives as the 3000/20 and 3000/30, while the 4000/20 used the same 49-megabyte disk pack drives as the 3000/40. The 4000/30 featured 73-megabyte disk drives. All models were supplied initially with ports for up to 16 users, with the 4000/20 and 4000/30 having expansion capabilities to 32 ports.

In March 1976, the 4000 Series was upgraded and redesignated Models 4000/15, 4000/25, and 4000/35. The most visible difference between the new 4000's and the old was the cabinetry. Using a special modular packaging technique, the equipment mounting chassis were stacked together. Decorative skins were then added to lock the stack together and form an integrated cabinet.

Less visible differences included 650-nanosecond MOS memory instead of the 980-nanosecond core, more internal functions implemented in microcode, and 7.5-megabyte disk drives substituted for the 2.4-megabyte drives on the low-end model. Sources within Hewlett-Packard regarded the BTI product line as one in which the microprogramming capabilities of the 21MX were most extensively exploited.

One essential component of all time-sharing systems, user terminals, was not (and still is not) supplied by BTI. The company recognized that many terminals are available directly to users and passed on to its customers the potential savings of direct procurement. Any asynchronous terminal with a data rate between 100 and 2500 bits per second and a standard RS-232C interface could be used on any 4000 Series system. BTI made no to Monitor mode). The lowest 22 bits of most instructions specify an operand, while the next three higher bits are sometimes used to specify a register. Different methods of referencing operands are provided by the "address mode" field and 54 addressing modes. Indirect addressing further involves special one-word structures called pointers, which themselves contain address mode fields and parameters for operand specification.

Instructions provided for subroutine linkage check entry points and provide parameter-type checking for the subroutine. The calling sequence and the entry sequence are executed part by part, passing one parameter at a time with the pass parameter instructions on the calling side and corresponding store parameter instructions on the subprogram side. The instructions specify the parameter type, whether the parameter is being passed by location or value, and whether this is the last parameter in the protocol.

INTERNAL CODE: ASCII.

MAIN STORAGE

TYPE: Core.

CYCLE TIME: 670 nanoseconds per 32-bit word, full cycle.

CAPACITY: Up to 16 million bytes per Memory Control Unit (MCU) in increments of 128K or 256K bytes. Each MCU occupies 1 of the 16 basic module slots.

CHECKING: Parity checking by byte.

PROTECTION: Security mechanisms have been designed into the system, including its hardware, to enable functioning in a multi-user on-line environment. The account structure is closed and secure in that all operations and data remain private within account boundaries unless explicit action to grant foreign access is taken. All password requirements are stored in encrypted form only, and there is no way to decrypt stored passwords. Users can share files on a read-only basis, can limit writing privileges to "append-only," or can grant full access to a file. All removable data storage media, including on-line disk packs and all backup media, are automatically protected by encrypted control information.

RESERVED STORAGE: The lowest n+5 pages (4096 bytes per page) on the system, where n is the number of Computational Processing Units present, are reserved for resident Monitor use.

CENTRAL PROCESSOR

The major resource modules and all peripheral controllers are special-purpose microprogrammed processors, which in turn use microcomputer-based submodules for many service fucntions. The foundation of the system is the Variable Resource Architecture (VRA) bus, a distributed-logic, passive, synchronous bus with a 32-bit-wide data path and 16 slots for the attachment of major modules in priority order. All data transfers between major modules take place through the VRA bus at 67 nanoseconds per 32-bit word (14.9 million words per second or 119 million bytes per second).

The four major modules are the System Services Unit, Computational Processing Unit, Memory Control Unit, and Peripheral Processing Unit. A system must include at least one of each, but no more than one System Services Unit is required.

The System Services Unit is internally cabled to the operator's panel, which is mounted on the top of the left-most system cabinet. The panel contains a readout of 10

BTI 8000

PERIPHERALS/TERMINALS

DEVICE	DESCRIPTION & SPEED	MANUFACTURER
MAGNETIC TAPE UNITS		
8330	9-track, 800/1600 bpi, switch-selectable density, IBM/ANSI-compatible, 45 ips	Amcomp
8310	Cartridge, 3M-type, 10 million bytes/cartridge, 6400 bpi	D.E.I.
PRINTERS		
8420	Line printer, 64- or 96-character set, 136 cols., 300 lpm	Dataproducts
8425	Line printer, 64- or 96-character set, 136 cols., 600 lpm	Dataproducts
8430	Line printer, 64- or 96-character set, 136 cols., 900 lpm	Dataproducts

specific endorsement or recommendations in favor of any particular vendor's terminals, but instead supplied a list of terminals known to have been used successfully with its equipment. Similarly, any modem with compatible transmission specifications and an RS-232C interface could be used for remote terminals.

BTI chose not to develop application software, but reached end users requiring such software through an informal alignment of its computer systems with application software furnished by independent vendors. BTI was able to offer a unique advantage to the application software supplier; protection for his software comparable to that of BTI's proprietary operating system. With his software protected by an exclusive "proprietary" screen initially set up by BTI, the vendor was able to install his software on BTI systems of his choice where he could support his software over the telephone, much as BTI supported its own software. This exclusive proprietary software protection facility is also available with BTI's current products. The 4000's special protection for added-value software made it easier for the company to establish joint selling arrangements with independent software suppliers, including OEM purchasers. This feature was also an advantage in selling to service bureaus because it enabled the service bureau to become an OEM supplier to clients whose billings had grown to the point of justifying the acquisition of an in-house system.

In September 1978, BTI introduced the 5000, which superseded the 3000 and 4000 and become BTI's mainstay 16-bit product line. The 5000 uses an upgraded version of the operating system proven on the 4000, and the CPU enables BTI to perform automatic remote fault diagnosis. The 5000's design allows it to be called and tested by a computer at BTI's service center without on-site assistance by the customer. BTI's diagnostic computers are also used to monitor a customer's system after a repair has been made and to carry out periodic "health checks," looking, for example, at the incidence of soft (disk-read) errors which might later lead to a hard failure. Another advantage of computer-to-computer communication is the ease with which patches can be inserted into an operating system. If a bug is discovered, possibly on just one system, a patch can be made automatically and quickly by BTI's service computers (usually overnight) on *all* Model 5000 installations.

alphanumeric characters for reporting system status and exception conditions, an alarm light, and 8 rocker switches. The switches include the main power switch, a switch to disable BTI remote maintenance access, a switch to select between normal startup and dedicated diagnostic startup, the run/halt switch, and four switches to select from 16 variations of startup or diagnostic operation.

CONTROL STORAGE: Although the BTI 8000 CPU is microprogrammed, the user cannot modify control storage.

REGISTERS: Eight 32-bit general-purpose registers, a program counter, and a processor status register are available for machine-language programming.

ADDRESSING: Fifty-four addressing modes reference operands in registers, in memory, and instructions themselves. Addressing modes directly support compiler data structures, including stack, queue, and linked-list, with data elements of arbitrary size. Virtual to physical memory address conversion is performed in two bus cycles (134 nanoseconds) in parallel with instruction execution.

INSTRUCTION REPERTOIRE: 174 user-mode instructions, each one word long, include fixed-point, floating-point, double-precision floating-point, and Boolean arithmetic; subroutine linkage; character string manipulation, and address mode instructions.

CONFIGURATION RULES

The 8000 is completely modular in configuration and is designed around a Variable Resource Architecture bus with a 32-bit-wide data path and 16 slots for the attachment of major modules in priority order. At least one each of the four major modules (System Services Unit, Computational Processing Unit, Memory Control Unit, and Peripheral Processing Unit) must be included. Only one System Services Unit is necessary, but multiples of the other three may be attached as needed to increase memory size, to add peripherals, or to increase computational power.

The Model 8205 disk controller used in the 8000 supports up to eight 34-, 67-, 135-, or 254-megabyte disk drives in any mix. One Peripheral Processing Unit supports up to four controllers.

Each Memory Control Unit supports up to 16 million bytes of memory in units of 128K bytes or 256K bytes. Memory connected to a given MCU must be of one type, but different MCU's can control different types of memory as BTI makes future memory offerings available.

Virtually any terminal with a standard RS-232C interface can be used with the system. Any modem with facilities for the RS-232C interface can be used for remote applications.

BTI 8000

The 8000 Monitor creates private virtual machine environments for each process, independent of and isolated from the hardware configuration. Users may therefore develop application programs without reference to the specific system's hardware, and reprogramming is unnecessary as a system is expanded or otherwise changed in configuration.

BTI's Variable Resource Architecture also makes the 8000 fail-soft. In a multi-module configuration, the loss of a processor or memory bank merely reduces the resource pool. The operator removes or replaces the faulty module—identified by built-in diagnostics—and resumes system operation with a one-button restart.

The BTI 8000's bus provides a 32-bit-wide data path and uses distributed logic to achieve a data transfer rate between resource modules of 60 megabytes per second. Up to 16 resource modules can be plugged into the bus. In addition to the CPU, MCU, and PPU modules, the system requires one System Services Unit (SSU), a microprogrammed processor that provides system control.

The SSU includes the system's operator control panel, with pushbuttons for various system operations and a 10-character alphanumeric display. The display informs the operator of normal and exception status conditions and the results of self-test diagnostics. Each resource module automatically runs a self-test at start-up. On completion of the self-tests, the SSU initiates Sysgen, which, if all modules are operative, configures the Monitor. Any module not in working order is identified by the SSU display. Sysgen also checks the resource modules present and, if the hardware configuration has been expanded or reduced by the operator since the previous start-up, automatically reconfigures the Monitor to match the resources available. The SSU contains a program-accessible system ID, a permanently assigned number which identifies the system in which the SSU is installed, which permits vendors of proprietary software packages control of which systems can run their packages by checking the system ID before executing.

The Computational Processing Unit is a microprogrammed processor which uses 32-bit architecture throughout. Integer arithmetic is 32 or 64 bits; floating-point arithmetic is 64 bits, with 128-bit double precision floating-point available through system software. Fifty-four addressing modes directly support compiler data structures, including stack, queue, array, and linked-list structures, with arbitrary size data elements. System computation is performed by one or more Computational Processing Units, operating in parallel. To gain more computational power, additional CPU's are plugged into the bus. The Monitor software assigns tasks equally among available CPU's to achieve true concurrent processing. As an indication of the CPU's speed, a fully-configured, multiple-CPU system performs floating-point multiplication with 64-bit operands in an average time of approximately

A 9-track, 800/1600-bpi, 45-ips magnetic tape drive, a 3M-type high-density cartridge tape unit, and three line printers with rates from 300 to 900 lines per minute are offered.

MASS STORAGE

34-, 67-, 135-, AND 254-MEGABYTE DISK SYSTEM: One 8205 disk controller supports up to eight 8210, 8215, 8220, or 8225 disk drives in any mix, with overlapped seeks. Disk drives are storage module type. Access times are identical for all four sizes. Average seek time is 30 milliseconds. Data transfer rate is 1.25 million bytes per second, and transfers to and from memory occur one full page (4096 bytes) at a time.

COMMUNICATIONS CONTROL

The BTI 8000, in its largest configuration, can support a practical limit of 512 interactive users.

8510 ASYNCHRONOUS COMMUNICATIONS CONTROLLER (ACC): Supports up to eight 8515 8-port (RS-232C) interfaces for a total of 64 ports. One Peripheral Processing Unit can control four ACC's (a total of 256 ports per PPU). Data rates can be set individually to any standard rate from 110 to 19,200 bps. ACC includes internal buffering to accommodate full-screen (1920 characters), block-mode, interactive terminals.

SOFTWARE

OPERATING SYSTEM: The BTI 8000 operating system (Monitor) pools and coordinates physical machine resources, including processors, to provide a secure environment for each user of the system. The Monitor shields all users from actual hardware configurations, creating a virtual machine for each user process, and is itself protected from violation by user processes. The Monitor is also responsible for automating as much of the system operation as possible.

When the system is started, either from the operator's panel or through the remote maintenance facility, the System Services Unit sends a start signal through the bus, causing all units to run self-contained diagnostics. Upon successful completion of this stage of system start, the first Computational Processing Unit to become ready temporarily takes over the system. It locks out other CPU's so that it can control system initialization, reads resident Monitor code from a known location on the system disk volume into the low pages of physical memory, and then executes that code. This is the only circumstance in which one CPU assumes control of the system to the exclusion of other CPU's.

When the other CPU's are unlocked, the system immediately enters its normal run mode. At the start, there are no users on the system (assuming a cold start), and all CPU's run that portion of Monitor code (from a fixed physical memory location) which investigates a task assignment table elsewhere in memory; at this point there will be no tasks, so all CPU's will go idle. When a device (particularly a communications controller) signals the beginning of what might be a user log-on activity, the associated Peripheral Processing Unit places an interrupt signal on the bus. The first CPU to respond will handle the interrupt and post to the appropriate Monitor tables.

In the steady state of system operation, when there are more processes than processors, each CPU requests an interrupt from the SSU every 100 milliseconds after it "switches in" to any task to see if another task should be executed. Periodic interruption to run Monitor task dispatching code does not require full context switching.

The memory tables used to direct and coordinate the

BTI 8000

three microseconds. Hardware address mapping in the CPU converts virtual memory addresses to physical memory addresses in parallel with instruction execution.

The 8000 uses core memory with a 670-nanosecond full cycle time (including Memory Control Unit or MCU operation). Read access time is under 400 nanoseconds. Memory is furnished in 128K-byte increments and is interfaced to the system through MCU's. Minimum memory is 256K bytes, and the system will support over 100 megabytes of memory. All memory present is treated by the system as an entity (even if interfaced through more than one MCU) organized in pages of 4096 bytes. In the event of a memory malfunction, the system eliminates the bad area on a page basis and reconfigures the available memory.

Mass storage for the BTI 8000 is provided by disk drives in formatted capacities of 34, 67, 135, and 254 megabytes. All drives use high-density, removable storage modules (packs). One disk controller can control up to eight drives in any mix and can provide for overlapping seeks to minimize access times. A special error-correction technique substantially reduces the risk of data loss in the event of disk read problems. Mass storage data transfers take place one page at a time, and each page occupies one "block" of disk capacity. Blocks are stored on disk in a number of segments in such a way that an entire block can be reconstructed even if a segment becomes totally unreadable. The system "remembers" bad areas and dynamically reassigns block placements to work around them.

All system peripherals, including disk drives, are controlled by device controllers. Each Peripheral Processing Unit can support up to four controllers in any mix, and controllers can handle as many as eight devices. Peripherals currently available for the 8000 include serial magnetic tape cartridge drives, 9-track open-reel magnetic tape drives, and three line printers with print rates of from 300 to 900 lines per minute.

Terminals and modems are interfaced to the 8000 through an Asynchronous Communications Controller (ACC), which is in turn controlled through a PPU channel. One PPU can control up to four ACC's, and each ACC can control up to 64 ports in increments of 8 ports. BTI considers a practical maximum for the 8000 to be 512 ports. The Asynchronous Communications Controller is a microprogrammed unit with internal buffering to accommodate full-screen (1920 characters), block-mode interactive terminals. Any or all ports can be used at rates up to 19,200 bits per second. To allow users flexibility in the type of asynchronous terminal or other asynchronous device to be used with the system (BTI does not include terminals in its product offerings), user programs have full control over interface pins, selection of terminating characters, and input and output buffers.

The Monitor provides private virtual machine environments in which the system manager, operator, and all other users operate. The virtual system shields all users from the actual hardware present in any given configura-

activities of multiple CPU's are read and updated using software lockout. The lockout algorithms and the CPU instructions used to implement them are the same as those used in non-Monitor software to coordinate any set of co-operating simultaneous processes. A given memory location is chosen by mutual agreement to contain a "lock" word. Before proceeding through a critical region of code to be entered and executed completely by only one process at a time, the process executes a noninterruptible instruction that sets a special locked value into the public lock word while simultaneously bringing the previous value of that word into private storage for examination. If the retrieved value is other than locked, the process continues through the critical region, unlocking it when done. If, on the other hand, the retrieved value is locked, then the process waits, since this indicates that some other process has entered the critical region.

Even though memory modules can be physically interfaced through separate Memory Control Units, the system treats all of memory as a single continuous resource. The low n+5 pages (n is the number of resident CPU's) are unavailable for paging, since they contain resident Monitor code and tables. The rest of memory is used on a page basis for temporary location of code and data transferred in from mass storage, with no pre-assigned boundaries or regions.

When a Monitor routine executing in a CPU instructs a PPU to transfer a page into memory from mass storage, the PPU is given two memory addresses. One is the location of the page itself; the other is the address of a Monitor table element for storage of the structural information included in every mass storage block. In this way, programs can make use of the full 1024 words in every page, since pointers, flags, and other maintenance information are kept externally to the page contents. A similar procedure is used to write to disk.

The Monitor keeps track of the logical status of all pages in memory, including their "home" addresses on mass storage. If a user requests execution of a program, the Monitor will search its lists before executing a disk-read request and will take advantage of memory residency of any of the program pages to avoid disk access; any number of users can share any number of pages. This list searching takes place with every page-read request, including those for file data blocks.

Access control flags associated with each page indicate whether the page is read-only or writable, and, if writable, whether it has been altered during its residency. This information allows pages of writable program data or file data to be shared among multiple users. They will share the same physical memory page initially, but the Monitor will create a private copy of a shared writable page for any process that issues an instruction that would alter the page contents.

The access control and status flags, including a "page referenced" flag, are carried into the page files of the CPU's, so that the system need not make an extra memory reference merely to update or examine them. The page referenced flag is used to identify the working set of a process as it executes, for scheduling purposes.

The system is disk-based in the sense that structural information and operating parameters are ultimately entrusted to mass storage. Main memory is treated as a temporary area for process operation, with any structural or parameter changes written to disk. System restart presumes no information in memory. Thus the main concern in mass storage management is maintaining the integrity of its structures.

Disk drives, disk modules, and disk volumes (the logical contents of packs) are all identified separately, so that,

BTI 8000

The hardware configuration of a BTI 8000 can be expanded by attaching additional "resource modules" to a high-speed bus. In the photo, a Computational Processing Unit is being plugged into the system cabinet. The bus circuitry is on the panel forming the back of the open area.

tion. Any program will execute regardless of the number of CPU's, amount of physical memory, or even the specific peripherals connected to the system. The user can make I/O assignments externally to his program to suit his convenience. The Monitor is protected to ensure inviolate system operation despite any possibly harmful activities attempted by any user process or the system operator. Each user program runs in a virtual work space of 512K bytes regardless of the actual amount of physical memory present or the number of other users sharing that memory. The Monitor itself uses address space separate from the user's work space for all I/O services associated with the user's process.

Memory is organized in pages of 4096 bytes, all of which is available to the programmer. Real memory is dynamically allocated to users' processes on a demand-paged basis in a manner transparent to users. User processes can generate other concurrent processes to handle heavy work loads in parallel.

Software emphasis is on the commercial DP market stressing data protection and on-line terminal access. Software bundled with the system includes the Monitor, COBOL (ANSI 74), Codasyl DBMS, Control Mode (the system's command language), BTI 8000 PASCAL-X, and utilities. PASCAL-X, an extended version of the PASCAL language, was chosen as the 8000's system programming language because BTI feels its structure lends itself to efficient implementation of system software in a multiple-processor environment. While a single-pass assembler is included in the 8000's unbundled software, BTI claims that object code generated by application programs written in PASCAL-X will execute as fast as

for example, volumes can be copied from module to module. Files and libraries of files reside on individual mass storage volumes, so that volumes may be dismounted either logically or physically without halting system operation or destroying the integrity of structures. The system volume, containing the Monitor's operational tables and routines as well as other data, cannot be dismounted.

Internal system tables that are critical to operation or to the use of an entire volume are recorded redundantly in the interests of protecting operations and data. During a structural update, the more junior table is created first and removed last. Even relatively complex structures are handled in a crash-resistant manner by using the worst-case technique of creating an entirely new structure containing the new information and a copy of any previous information to be retained, updating the block that points to it, and finally freeing the old structure space.

Since the purpose of the BTI 8000 is to support many simultaneous processes, it is properly described as a multi-tasking system as well as a multiprocessor system. A "process" is the distinct invocation or separate execution of a program. Each process on the system is usually, but not always, associated one-for-one with an on-line interactive user. An interactive user process may generate other concurrent processes. Programs executed from batch queues are processes, and invocations of Monitor routines are also processes.

The Monitor creates a basically private, but identical, virtual machine for each process; one of its aspects is the process address space, or virtual memory. Any and every program on the BTI 8000 may be written to address a continuous virtual memory of 128 pages (512K bytes) as if it were the only program executing on a private computer with that much physical memory. The Monitor creates and maintains the correspondence between each page of every process' virtual memory and some page in physical memory; this is what is loaded into a CPU's page file when a CPU runs a process.

Processes on the BTI 8000 may be running in some CPU, runnable but waiting for a CPU to become available, or waiting for some other resource, including a page of virtual memory which is not yet resident in physical memory. Every process has all of its required virtual pages represented on blocks of a mass storage volume, but normally not all of them will be represented in physical main memory. When a running process references a virtual page that is not resident, as indicated by the page file, the process becomes suspended and the Monitor assumes the responsibility of loading the page from mass storage into some page of physical main memory.

Three characteristics of this technique should be noted. First, the pages of a given process may be placed anywhere in paged memory. Second, a memory page that the Monitor chooses to overlay with a new page will not first be written back to disk if it has not been altered since it was loaded in from disk. Third, frequently referenced pages, including pages referenced by more than one process, tend to remain resident, since the Monitor's replacement algorithm tries to minimize disk access.

Demand-paging systems normally operate with a least-recently-used replacement algorithm; that is, the page chosen for overlaying is that which has "aged" the longest since being referenced by any process. This algorithm is entirely reasonable with a moderate load on a system but invites "thrashing" when the load grows too large. The pages used by a given process during a specified period of time are its working set for that period. In a demand-paging system, the relationship between the total pages required to hold all active working sets and the total number of memory pages available determines the amount of disk activity on the

BTI 8000

that produced by well-written assembly language programs, offering a clear advantage in development time and effort for applications programmers. PASCAL-X and other BTI-supplied compilers use standardized interfaces to allow intermixing of their object routines with each other and with assembler object.□

system. As the page load grows, disk transfers become more frequent until all processes are reduced to their minimum working sets, below which they are incapable of executing any instructions without demanding a new page. Thereafter, any increase in load causes the system to spend almost all of its time in disk transfers. At this point, essentially no work is performed. The BTI 8000 modifies the conventional demand-paging algorithm to prevent thrashing. When a demand occurs, the Monitor selects the "least valuable process," based on a number of criteria, including the distinction between interactive and batch processes, and strips this process of the least recently used page of its working set, overlaying that memory page with the one demanded. As the overall load grows, this procedure is repeated until all processes are reduced to working sets close to minimum. At this stage, prior to the thrash point, the Monitor identifies the process that is the most critical "troublemaker"—normally the one with the largest current working set. The Monitor then suspends this process for a certain period of time, rolling out its entire working set to free up memory for the rest of the load. The BTI 8000 process management algorithm avoids thrashing by making a dynamic transition from demand-paging to a modified multiprogramming technique.

There are no preset, conventional priorities in this scheduling technique, although the system operator can modify certain scheduling parameters (e.g., to favor batch processing). The Monitor automatically favors processes that are currently interactive on the assumption that a user at a terminal requires service as soon as possible after entering a message. Processes that are not currently interactive can relinquish their demands for system resources, including CPU's, in favor of interactive processes, although a "fairness" algorithm ensures that batch processes are not totally locked out of execution. Process scheduling operates with dynamic priorities according to the recent behavior and current characteristics of the processes. On a heavily loaded system, with all other considerations equal, the most efficiently written programs—those with compact working sets—will be favored for execution over potential troublemakers.

LANGUAGES: The BTI 8000 supports six programming languages: COBOL, FORTRAN, PASCAL-X, BASIC-X, RPG II, and Assembler. All six have the following concepts in common:

- Program development may occur in an interactive mode. Programs may be written, compiled, and linked from a terminal, and test files can be defined, built, and dumped from a terminal. Programs can be tested at a terminal with the aid of an interactive, symbolic level debugging facility.
- All languages support terminals as standard I/O devices. A terminal can be accessed by the Monitor without special telecommunications software.
- All language run-time systems, except BASIC-X, are based on a common object program format.
- Any file created by any language, except BASIC-X, can be read by any language, except BASIC-X.

BTI 8000 COBOL includes Level 2 ANSI COBOL X.3.23-1974, except Segmentation is not supported, an interactive symbolic Debug replaces the standard Debug, non-standard Interprogram Communications is provided, non-standard Communications capabilities are supported, and the Report Writer module is scheduled for delivery in late 1980.

BTI 8000 COBOL includes the following enhancements to the standard:

- Support of the discontinued 1968 COBOL constructs NOTE, EXAMINE, and REMARK.
- DBMS-X Data Base Management System verbs are fully supported by the COBOL compiler; no precompiler is necessary, nor is it necessary to employ the CALL verb for data base access.
- Fully compatible data types between COBOL and FORTRAN.
- Symbolic interactive debug facility.

BTI 8000 FORTRAN is a full implementation of FORTRAN-77, ANSI standard FORTRAN X3.9-78. The compiler also accepts programs written in compliance with ANSI FORTRAN 66 and commonly used extensions.

BTI 8000 FORTRAN allows the programmer to concentrate on the algorithm instead of its implementation. Unlimited statement length and support of upper/lower case symbolic data names are two of its features. The compiler has a three-level error and warning system for diagnostics, allowing compilation to continue in spite of minor errors. Debugging statements (identified by a D in column 1) can be included in the compilation or interpreted as remarks, depending upon a selected compiler option. A fully interactive debugging facility aids in program test and verification.

BTI 8000 FORTRAN places the full power of the operating system in the hands of the programmer through the use of extended I/O facilities. Files may be created, attached, interrogated, and destroyed under program control using the OPEN, CLOSE, and INQUIRE statements. Data transfer to and from files may be formatted, unformatted, or list-directed. Files may be direct access or sequential and may contain variable length records. In addition, data may be transferred to and from character strings by using statements which are similar to regular I/O statements. Character strings may be concatenated and assigned to variables. Strings may be compared with other strings. Substrings may be extracted with a convenient subscript-like notation. Numeric data types (Real, Integer, and Complex) have over 15 digits of significance; Double Precision Real supports 34 digits of significance. Variables of any type may be subscripted, and an array may have up to seven dimensions with no restrictions on upper and lower bounds.

BTI 8000 FORTRAN extends the FORTRAN-77 standard with the following features:

- Debugging statements can be easily eliminated for compilation of production program version.
- Array subscript and computed GO TO expressions of real, double precision, or integer type.
- Interactive debug support.
- Symbolic names, 1 to 8 characters in length, in upper and lower case.
- Subexpression optimization.
- Variable length record I/O support.

BTI 8000 PASCAL-X includes all the features of standard PASCAL and is a valid superset of standard PASCAL.

BTI 8000

BTI 8000 PASCAL-X adds the following features to standard PASCAL:

- Full support for STRING data type.
- Conditional error recovery through the "ON error-condition DO" construct. In keeping with the block-structured nature of PASCAL, PASCAL-X includes a "TRY... ELSE" construct for structured error handling.
- Full access to all aspects of the BTI 8000 I/O and file system.
- Run-time adjustable arrays.
- Compile-time expression evaluation for "CONST" declarations.
- Spawning and management of concurrent processes and generation and control of underprograms.
- Inter-process communication capability.

PASCAL-X has full access to the entire BTI 8000 virtual machine. Efficiency-critical portions of a solution can be coded in assembly language, but the assembly-language routines are treated as PASCAL procedures. The block structure of the language is maintained, and machine-dependent code is isolated for ease of maintenance and conversion. All BTI 8000 systems software is written in PASCAL-X. PASCAL is also a feasible production language, providing structured programming capabilities.

BTI 8000 BASIC-X is a totally compatible implementation of BTI's previous BASIC-X. The primary design objective was to provide BTI customers with a conversion-free growth path from the BTI 3000/4000/5000 systems to the BTI 8000.

BTI 8000 BASIC-X is implemented using a technique which has characteristics of both a compiler and an interpreter. As source statements are presented to BASIC-X, during either keyboard entry or retrieval from a source library, each is translated on a line-by-line basis, as with an interpreter. Once the execution requirements of the line have been determined, however, the incremental compiler generates machine object code which is stored for later execution. When the program is run, the machine object code is executed in approximately the same manner that a compiled program executes; the interpreter-like features result in certain efficiency trade-offs. Further, the quasi-compiled object code can be stored and retrieved for later use without further translation.

As implemented on the BTI Model 3000/4000/5000, BASIC-X provides a fully interactive program development cycle. The incremental compiler implementation of BASIC-X on the BTI 8000 maintains this same user environment while approaching the execution-time efficiency of a compiler.

BASIC-X includes the following extended language facilities:

- String arithmetic with 252-decimal-digit precision.
- Extensive string and substring manipulation facilities.
- Complex error handling under control of the user program.
- File creation/deletion within user program.
- Up to 64 files open concurrently in each program.
- Extensive file-sharing features in support of update and inquiry activities.
- Complete set of matrix operators.
- COMmon file declaration between CHAINed programs to eliminate redundant file linking.
- PRINT USING to a string variable as well as to an I/O device.

BTI 8000 RPG II offers many of the programming capabilities of machine-oriented programming languages and will accept programs written for other manufacturers' RPG with little or no re-coding. Some of the significant advantages of BTI 8000 RPG II include:

- The full program development cycle is supported from terminals.
- Application programs written in RPG II can use terminals as input/output devices without additional control software.
- Operation in a completely secure, multi-user environment is supported. Program development, production processing, and terminal-based application programs can proceed concurrently without special system scheduling considerations.

BTI 8000 RPG II allows current batch-oriented users to upgrade to terminal-oriented data processing. Existing RPG batch application programs can be moved to the BTI 8000 with a minimum of conversion effort, and experienced RPG programmers can be trained to exploit the terminal capabilities of the 8000 as part of the general introduction to the system. BTI 8000 RPG II will be available in late 1980.

BTI 8000 Assembler is an extremely fast assembly-language translator whose features and internal design resemble those of a compiler rather than a traditional assembler. Assembler relieves the programmer of the burden of selecting address modes for operands by providing for definition of data structures and then automatically generating the proper address modes for referencing elements of those structures. Operand-field entries refer to the operands themselves, not their addresses.

Although it is a single-pass translator, Assembler allows forward referencing by deferring code generation as appropriate. It includes a BOX statement to encourage documentation; BOX encloses comments with a box of asterisks on the listing. The INPUT statement allows the source of language input to switch among files, with nesting of sources allowed. Assembler produces detailed cross-reference listings, with each cross-reference indication annotated to show how the symbol was used.

CODASYL DATA BASE MANAGEMENT SYSTEM: DBMS-X is a general purpose data base management system supported by BTI for use on the 8000. It provides for the definition, creation, maintenance, restructuring, and backup of network data bases through the use of standard system functions and high level language interfaces and assures the integrity of information stored under its control in a multilingual, multi-user on-line environment. DBMS-X complies with and exceeds the 1973 CODASYL Programming Language Committee specifications for data base management systems.

BTI has implemented extensions to the standard which provide the following features:

- Privacy and security controls implemented at the data item level through the use of access control lists. Specific system users identified by their account ID's are granted access to particular portions of the data base by the Data Base Administrator.

- Logical integrity protection provided through the BEGIN-TRANSACTION/COMMIT-TRANSACTION language construct. The system assures that all operations specified between the BEGIN-TRANSACTION and COMMIT-TRANSACTION designators in an application program have been successfully completed before releasing the data base for modification by another user and/or transaction. Update sequences which fail or are aborted prematurely due to system failure, logical error, or operator intervention are completely backed out of the data base before the next user or transaction can gain control.

- A series of meta-commands in the Data Manipulation Language allows programmers to perform operations that involve *groups* of data sets. These commands dramatically simplify the programming tasks required to implement inquiry and reporting functions against the data base.

DBMS-X provides a multi-level security system which allows for the implementation of extremely tight information access controls. At the highest level, it is impossible for an individual to gain access to *any* information unless he is logged onto the system as a user. As part of the log-on procedure, the user must supply the operating system with proper account identification and password information, which identifies the individual to the system as well as to the data base management system. As part of the data base definition process, the Data Base Administrator specifies the users who are allowed to access the data base. The DBA also specifies which portions of the data base may be read or changed by each valid user. In the absence of permission to access a data base, a user is prohibited from any activity against the data base.

DBMS-X temporarily blocks all other programs from accessing those resources which are required to complete a transaction, while assuring that the transaction either processes to completion or does not process at all. It specifically prevents the system from processing only part of a logical transaction and thus protects the logical integrity of the data base. Upon either completion or abortion of the transaction, control over the locked portions of the data base returns to the system. If the application program detects that a transaction cannot or should not be completed, the program executes an ABORT-TRANSACTION statement, which releases all update pages created by this transaction, clears the Update Table, sets the Transaction Status Code to COMPLETE, and releases all locked resources. Thus, the data base appears as if the update sequence had never started processing. If the run-time control system associated with data base processing detects that a program using a data base has terminated abnormally, it triggers an ABORT-TRANSACTION against the data base. This procedure returns the base to its last consistent state, and access to the base is then terminated as if the program had relinquished control of the base normally.

DBMS-X is designed to allow recovery from "hard" system failures in which some portion of the physical data base has been destroyed. In the event of a hard system failure, a backup copy of the data base is restored to the system, and "shadow" data base entries are applied against it. At the end of the recovery operation, the data base is as current as the shadow data base.

DBMS-X uses the shadow data base concept to provide a Program Test Mode facility which allows programmers to test programs against live data bases without interfering with concurrent production processing. The execution of a BEGIN-TEST-MODE statement in an application program triggers the establishment of a shadow data base for the test environment. Any changes made to the data base by programs running in test mode are stored in the test shadow data base. The test mode data base remains in existence until specifically destroyed by a program command. Thus, it is possible to run exhaustive test and verification procedures without special control procedures.

DBMS-X is designed to guarantee utilization of allocated disk space of at least 50 percent with utilization levels of better than 90 percent for growing data bases. Search structures are automatically maintained to assure rapid retrieval. The Data Base Administrator is, however, provided with the ability to monitor performance and adjust the run-time environment to improve data base performance.

Programs which access DBMS-X can be written in COBOL, FORTRAN, or PASCAL-X. Each host language has been extended to allow data base functions to be accepted directly by the language compiler, eliminating the need for preprocessors. The syntax of the data base extension set conforms to the overall form and style of the host language.

APPLICATION SOFTWARE: BTI does not generate applications software, but assists in the marketing of selected user-generated packages. A unique feature of the BTI operating system permits applications software to be installed on a system where it may be used in read-only mode by the system owner. It can, however, be accessed for updates and maintenance by the software vendor through a special "proprietary software account." Under this feature, the system owner and users are permitted access to the program and to all system management privileges except the proprietary source code. The vendor, however, given telephone access to his proprietary software, can update and correct the package without the need of sending copies or interrupting users.

BTI provides a utilities package which includes editor, sort/merge, copy, debug, and masters' and operators' programs.

PRICING

POLICY: BTI offers the 8000 system on a purchase-only basis. The base system configuration is complete and includes the Monitor operating system, Control Mode, PASCAL-X, and the utilities package. BTI warrants all hardware for 90 days. Software is licensed for use on one system, but discounts are offered for multiple installations by one customer. BTI-furnished software is maintained free for one year, and continuing maintenance is available on a yearly contract basis. Upgrades of BTI software are offered for a nominal handling fee to customers using the existing software.

SUPPORT: BTI features a unique customer-participation service and support system that combines human resources and hardware features of the 8000 system. The 8000 contains integrated maintenance aids for automatic fault diagnosis by a remote computer located at BTI's factory service center. BTI customer engineers can gain access to the operating system through a user port and exercise various system components. Customer cooperation, in the form of a person standing by the system to perform specified actions, may be required to aid the BTI engineer in testing and evaluating a failed system. The customer's responsibility to provide such assistance is noted in BTI's corrective maintenance contract.

Under the contract terms, BTI furnishes both parts and labor to correct all failures and to provide 7-day, 24-hour telephone service. Replacement parts are shipped from the factory by air freight, scheduled airline, or package express service to users, who replace them and return the failed parts. A BTI systems engineer is dispatched from Sunnyvale to any site where telephone consulting and testing cannot correct the malfunction.

BTI 8000

BTI Corrective Maintenance Plan charges are based on the hardware configuration. Typical monthly charges covering both preventive and connective maintenance are less than 1 percent of the purchase price.

BTI also features a customer-participation Preventive Maintenance Plan that includes the periodic sending of kits to customer sites. These kits include necessary air filters, cleaning agents, supplies, and reporting forms. Customer personnel perform the stipulated activities and return the completed forms to BTI.

System purchase prices include training by BTI personnel. Training includes both operation and maintenance procedures.

SMALL BTI 8000 SYSTEM: Consists of one each of the four major modules, 512K bytes of memory, two 34-megabyte disk drives, 32 user ports, cabinet with operator control panel, and software including the Monitor, Control Mode, PASCAL-X, and utilities. Purchase price is $118,200.

MEDIUM BTI 8000 SYSTEM: Consists of one each of the four major modules, two extra Computational Processing Units, one extra Peripheral Processing Unit, 1536K bytes of memory, four 135-megabyte disk drives, one 9-track magnetic tape drive, a 900-lpm printer, 128 user ports, cabinet with operator control panel, and software including the Monitor, Control Mode, PASCAL-X, and utilities. Purchase price is $361,200.

EQUIPMENT PRICES

		Purchase Price
BASE SYSTEM		
8000	Includes Computational Processing Unit, Memory Control Unit, Peripheral Processing Unit, System Services Unit, 256K bytes of memory, 34-megabyte disk storage drive, magnetic tape cartridge drive, 8-port interface, and double bay cabinet	$86,850
SYSTEM RESOURCES MODULES		
8110	Computational Processing Unit (CPU)	12,000
8130	Memory Control Unit (MCU)	3,000
8170	Peripheral Processing Unit (PPU)	8,000
8190	System Services Unit (SSU)	4,000
MEMORY		
8132	Memory Module, 128K bytes	9,000
8133	Memory Module, 256K bytes	16,000
8151	Memory Power Supply, supports up to 256K bytes on one MCU	2,500
8152	Memory Power Supply, supports up to 512K bytes on one MCU	3,500
MASS STORAGE		
8205	Disk Controller for 8210, 8215, 8220, 8225 Drives; supports up to 8 drives in any combination	10,000
8210	34M-byte Storage Module Drive with one removable disk pack	9,850
8215	67M-byte Storage Module Drive with one removable disk pack	12,500
8220	135M-byte Storage Module Drive with one removable disk pack	18,000
8225	254M-byte Storage Module Drive with one removable disk pack	30,000
MAGNETIC TAPE EQUIPMENT		
8305	Magnetic Tape Controller for 8310, 8315 Drives, controls up to 4 drives	3,000
8310	Single Magnetic Tape Cartridge Drive with four magnetic tape cartridges	3,000
8315	Additional Magnetic Tape Cartridge Drive	2,500
8320	Magnetic Tape Controller for 8330 Drive, supports up to 4 drives	5,000
8330	9-Track Magnetic Tape Drive, 800/1600 bpi; includes single-bay cabinet extension	9,000
PRINTERS		
8415	Line Printer Controller for 8420, 8425, 8430 Printers	5,000
8420	Line Printer, 300 lpm	10,000
8425	Line Printer, 600 lpm	13,500
8430	Line Printer, 900 lpm, includes quietized cabinet	18,700
	96-character set for 8420, 8425, or 8430	1,500
	Quietized cabinet for 8420 or 8425	800
COMMUNICATIONS		
8510	Asynchronous Communications Controller, supports up to eight 8515 8-port interfaces	8,000
8515	8-port interface, EIA RS-232C, at rates to 19,200 bps	2,000

SOFTWARE PRICES

	License Fee
FORTRAN	2,000
BASIC-X	2,000
RPG II	3,000
COBOL	4,000
DBMS X	4,500
Assembler	4,000 ■