

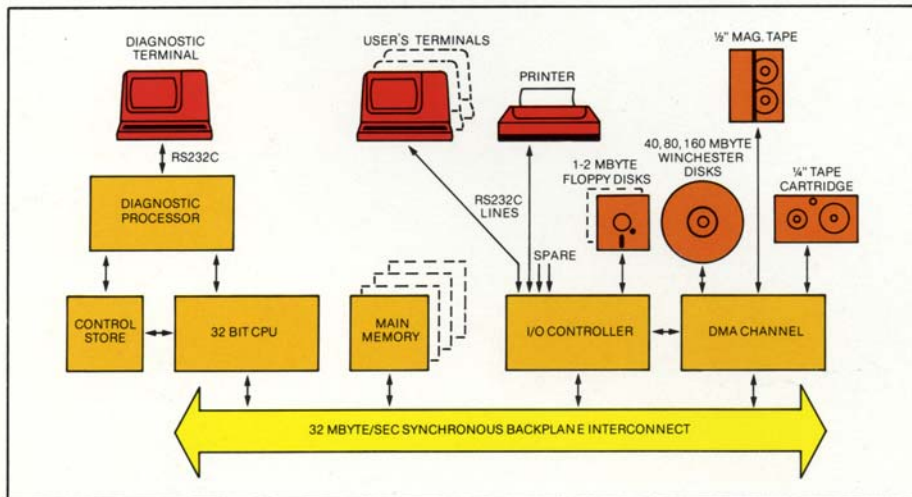
ORION

A HIGH PERFORMANCE PERSONAL COMPUTER

ORION A NEW CONCEPT IN PERSONAL COMPUTING

ORION is High Level Hardware's new 32-bit word length supermicrocomputer intended particularly for the scientific community. Designed primarily as a member of the new generation of high performance personal computers, the ORION machine will be of particular value to users whose work involves a large amount of computation such as in the fields of mathematical modelling, artificial intelligence and symbolic algebra. With its combination of high speed and architectural flexibility it will also be of interest to computer scientists as a powerful research tool for the development of the next generation of computer architecture.

Over the last twenty years there have been many developments in computer hardware technology at the component level but these have not been matched by corresponding advances in computer architecture. The ORION hardware attempts to redress this imbalance. ORION's architecture is designed to yield efficient and reliable implementations of high level languages, a feature previously restricted to expensive mainframe computers.



The ORION hardware is complemented by UNIX†, a portable operating system that brings with it the benefits of mature design, a variety of high level languages and an extensive selection of tools for program development and text processing. Unusually for a new machine, the adoption of UNIX means that ORION users gain immediate access to a wealth of existing software and become part of the large and rapidly growing international community of UNIX users.

† UNIX is a Trademark of Bell Laboratories.

The ORION computer is a compact unit mounted in a standard 19 inch rack built in to a desk; the rack also houses disk and tape drives.

The standard microcode implements a stack oriented machine. Rather than forcing all data through the bottleneck of a small number of high speed registers, a working stack is maintained in cache memory that is used both for expression evaluation and for storage of local variables. Measurements on ORION have shown that for programs written in high level procedural language, the majority of instructions need only access this working stack, giving fast overall operation. Operational speed is further enhanced by adopting a compact representation for the most frequently used instructions. Instruction latency is minimised by arranging for up to eight instructions to be pre-fetched from main memory into an opcode cache in the CPU and by overlapping instruction decoding and execution.

Central processing unit

The 32-bit ALU is built around Am2901C bit slice microprocessors. To these has been added a byte manipulation unit which can perform the shifting, rotating and masking operations required for handling eight and sixteen bit data. Additional logic is provided to support both signed and unsigned two's complement comparisons in a single operation, multiple precision arithmetic and floating point normalization. Most operations can be performed in 150 ns. The cycle time is variable from 125 ns to 200 ns under micro-program control so that timing can be optimised.

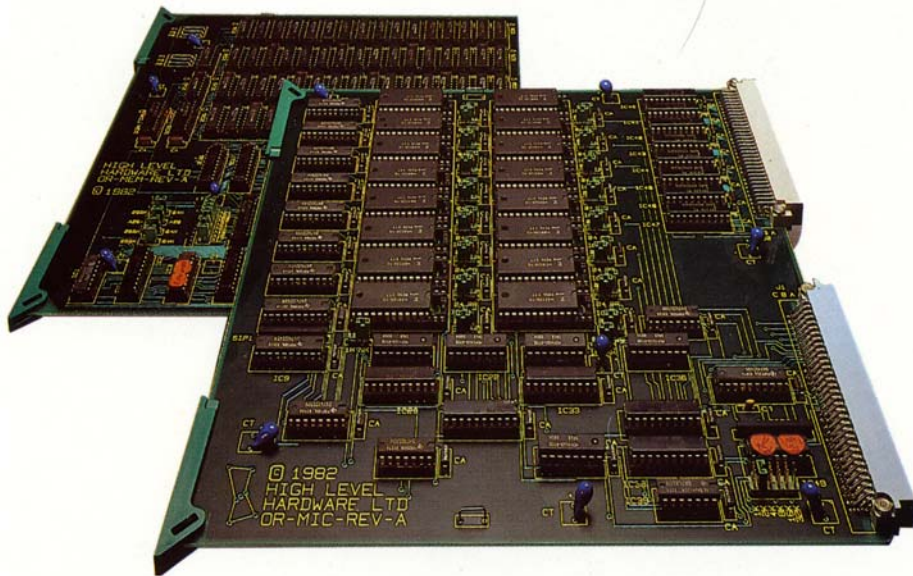
The micro-sequencer directs the flow of control through the microprogram. It can perform branches, loops and subroutine calls, most of which may be conditional on any of several CPU status conditions. The main

component is the Am2910, an LSI device designed specifically for this purpose.

The instruction decoder is the portion of the CPU responsible for decoding machine level instructions (as opposed to micro-instructions). This is done using map tables held in fast parity checked RAM which map one byte opcodes onto micro-instruction addresses. Control is transferred to these addresses using a special sequencer operation which is performed in parallel with other CPU functions; instruction decoding is thus normally fully overlapped with instruction execution.

An escape mechanism is provided to allow the instruction set to be expanded beyond the 256 entries selected by a one byte opcode. A further mechanism exists to switch between several sets of dispatch tables, allowing the machine to support multiple instruction sets concurrently. Using this mechanism a different instruction set can be selected each time a context switch occurs. The mechanism is also used to implement privileged instructions, dynamic profiling (for performance monitoring) and multiple CPU modes (e.g. User and Kernel).

The cache memory is totally independent of main memory. Its principle function is to hold the top of an evaluation stack for a procedure oriented language. The cache has a two cycle latency after which it can deliver one word per cycle and is divided into a number of pages each of 512 32-bit words with parity protection. The pages are grouped in pairs with an architectural maximum of 16 pairs, the current hardware having two such pairs. The second member of each pair is used typically as additional fast registers and scratch storage without affecting the stack page. The lower nine bits of the CPU register which addresses the cache is implemented with counters which allow increment and decrement operations (push and pop) as well as random access. By exploiting the structure of modern high level programming languages and keeping the current stack frame in the cache memory, a high hit rate is obtained using very simple hardware.



CONTROL STORE

MEMORY

CPU

I/O CONTROLLER

SYNCHRONOUS
BACKPLANE
INTERCONNECT

IC1 CA
IC2 CA
IC3 CA
IC4 CA
IC5 CA
IC6 CA
IC7 CA
IC8 CA
IC9 CA
IC10 CA
IC11 CA
IC12 CA
IC13 CA
IC14 CA
IC15 CA
IC16 CA
IC17 CA
IC18 CA
IC19 CA
IC20 CA
IC21 CA
IC22 CA
IC23 CA
IC24 CA
IC25 CA
IC26 CA
IC27 CA
IC28 CA
IC29 CA
IC30 CA
IC31 CA
IC32 CA
IC33 CA
IC34 CA
IC35 CA
IC36 CA
IC37 CA
IC38 CA
IC39 CA
IC40 CA
IC41 CA
IC42 CA
IC43 CA
IC44 CA
IC45 CA
IC46 CA
IC47 CA
IC48 CA
IC49 CA
IC50 CA
IC51 CA
IC52 CA
IC53 CA
IC54 CA
IC55 CA
IC56 CA
IC57 CA
IC58 CA
IC59 CA
IC60 CA
IC61 CA
IC62 CA
IC63 CA
IC64 CA
IC65 CA
IC66 CA
IC67 CA
IC68 CA
IC69 CA
IC70 CA
IC71 CA
IC72 CA
IC73 CA
IC74 CA
IC75 CA
IC76 CA
IC77 CA
IC78 CA
IC79 CA
IC80 CA
IC81 CA
IC82 CA
IC83 CA
IC84 CA
IC85 CA
IC86 CA
IC87 CA
IC88 CA
IC89 CA
IC90 CA
IC91 CA
IC92 CA
IC93 CA
IC94 CA
IC95 CA
IC96 CA
IC97 CA
IC98 CA
IC99 CA
IC100 CA
IC101 CA
IC102 CA
IC103 CA
IC104 CA
IC105 CA
IC106 CA
IC107 CA
IC108 CA
IC109 CA
IC110 CA
IC111 CA
IC112 CA
IC113 CA
IC114 CA
IC115 CA
IC116 CA
IC117 CA
IC118 CA
IC119 CA
IC120 CA
IC121 CA
IC122 CA
IC123 CA
IC124 CA
IC125 CA
IC126 CA
IC127 CA
IC128 CA
IC129 CA
IC130 CA
IC131 CA
IC132 CA
IC133 CA
IC134 CA
IC135 CA
IC136 CA
IC137 CA
IC138 CA
IC139 CA
IC140 CA
IC141 CA
IC142 CA
IC143 CA
IC144 CA
IC145 CA
IC146 CA
IC147 CA
IC148 CA
IC149 CA
IC150 CA
IC151 CA
IC152 CA
IC153 CA
IC154 CA
IC155 CA
IC156 CA
IC157 CA
IC158 CA
IC159 CA
IC160 CA
IC161 CA
IC162 CA
IC163 CA
IC164 CA
IC165 CA
IC166 CA
IC167 CA
IC168 CA
IC169 CA
IC170 CA
IC171 CA
IC172 CA
IC173 CA
IC174 CA
IC175 CA
IC176 CA
IC177 CA
IC178 CA
IC179 CA
IC180 CA
IC181 CA
IC182 CA
IC183 CA
IC184 CA
IC185 CA
IC186 CA
IC187 CA
IC188 CA
IC189 CA
IC190 CA
IC191 CA
IC192 CA
IC193 CA
IC194 CA
IC195 CA
IC196 CA
IC197 CA
IC198 CA
IC199 CA
IC200 CA
IC201 CA
IC202 CA
IC203 CA
IC204 CA
IC205 CA
IC206 CA
IC207 CA
IC208 CA
IC209 CA
IC210 CA
IC211 CA
IC212 CA
IC213 CA
IC214 CA
IC215 CA
IC216 CA
IC217 CA
IC218 CA
IC219 CA
IC220 CA
IC221 CA
IC222 CA
IC223 CA
IC224 CA
IC225 CA
IC226 CA
IC227 CA
IC228 CA
IC229 CA
IC230 CA
IC231 CA
IC232 CA
IC233 CA
IC234 CA
IC235 CA
IC236 CA
IC237 CA
IC238 CA
IC239 CA
IC240 CA
IC241 CA
IC242 CA
IC243 CA
IC244 CA
IC245 CA
IC246 CA
IC247 CA
IC248 CA
IC249 CA
IC250 CA
IC251 CA
IC252 CA
IC253 CA
IC254 CA
IC255 CA
IC256 CA
IC257 CA
IC258 CA
IC259 CA
IC260 CA
IC261 CA
IC262 CA
IC263 CA
IC264 CA
IC265 CA
IC266 CA
IC267 CA
IC268 CA
IC269 CA
IC270 CA
IC271 CA
IC272 CA
IC273 CA
IC274 CA
IC275 CA
IC276 CA
IC277 CA
IC278 CA
IC279 CA
IC280 CA
IC281 CA
IC282 CA
IC283 CA
IC284 CA
IC285 CA
IC286 CA
IC287 CA
IC288 CA
IC289 CA
IC290 CA
IC291 CA
IC292 CA
IC293 CA
IC294 CA
IC295 CA
IC296 CA
IC297 CA
IC298 CA
IC299 CA
IC300 CA
IC301 CA
IC302 CA
IC303 CA
IC304 CA
IC305 CA
IC306 CA
IC307 CA
IC308 CA
IC309 CA
IC310 CA
IC311 CA
IC312 CA
IC313 CA
IC314 CA
IC315 CA
IC316 CA
IC317 CA
IC318 CA
IC319 CA
IC320 CA
IC321 CA
IC322 CA
IC323 CA
IC324 CA
IC325 CA
IC326 CA
IC327 CA
IC328 CA
IC329 CA
IC330 CA
IC331 CA
IC332 CA
IC333 CA
IC334 CA
IC335 CA
IC336 CA
IC337 CA
IC338 CA
IC339 CA
IC340 CA
IC341 CA
IC342 CA
IC343 CA
IC344 CA
IC345 CA
IC346 CA
IC347 CA
IC348 CA
IC349 CA
IC350 CA
IC351 CA
IC352 CA
IC353 CA
IC354 CA
IC355 CA
IC356 CA
IC357 CA
IC358 CA
IC359 CA
IC360 CA
IC361 CA
IC362 CA
IC363 CA
IC364 CA
IC365 CA
IC366 CA
IC367 CA
IC368 CA
IC369 CA
IC370 CA
IC371 CA
IC372 CA
IC373 CA
IC374 CA
IC375 CA
IC376 CA
IC377 CA
IC378 CA
IC379 CA
IC380 CA
IC381 CA
IC382 CA
IC383 CA
IC384 CA
IC385 CA
IC386 CA
IC387 CA
IC388 CA
IC389 CA
IC390 CA
IC391 CA
IC392 CA
IC393 CA
IC394 CA
IC395 CA
IC396 CA
IC397 CA
IC398 CA
IC399 CA
IC400 CA
IC401 CA
IC402 CA
IC403 CA
IC404 CA
IC405 CA
IC406 CA
IC407 CA
IC408 CA
IC409 CA
IC410 CA
IC411 CA
IC412 CA
IC413 CA
IC414 CA
IC415 CA
IC416 CA
IC417 CA
IC418 CA
IC419 CA
IC420 CA
IC421 CA
IC422 CA
IC423 CA
IC424 CA
IC425 CA
IC426 CA
IC427 CA
IC428 CA
IC429 CA
IC430 CA
IC431 CA
IC432 CA
IC433 CA
IC434 CA
IC435 CA
IC436 CA
IC437 CA
IC438 CA
IC439 CA
IC440 CA
IC441 CA
IC442 CA
IC443 CA
IC444 CA
IC445 CA
IC446 CA
IC447 CA
IC448 CA
IC449 CA
IC450 CA
IC451 CA
IC452 CA
IC453 CA
IC454 CA
IC455 CA
IC456 CA
IC457 CA
IC458 CA
IC459 CA
IC460 CA
IC461 CA
IC462 CA
IC463 CA
IC464 CA
IC465 CA
IC466 CA
IC467 CA
IC468 CA
IC469 CA
IC470 CA
IC471 CA
IC472 CA
IC473 CA
IC474 CA
IC475 CA
IC476 CA
IC477 CA
IC478 CA
IC479 CA
IC480 CA
IC481 CA
IC482 CA
IC483 CA
IC484 CA
IC485 CA
IC486 CA
IC487 CA
IC488 CA
IC489 CA
IC490 CA
IC491 CA
IC492 CA
IC493 CA
IC494 CA
IC495 CA
IC496 CA
IC497 CA
IC498 CA
IC499 CA
IC500 CA
IC501 CA
IC502 CA
IC503 CA
IC504 CA
IC505 CA
IC506 CA
IC507 CA
IC508 CA
IC509 CA
IC510 CA
IC511 CA
IC512 CA
IC513 CA
IC514 CA
IC515 CA
IC516 CA
IC517 CA
IC518 CA
IC519 CA
IC520 CA
IC521 CA
IC522 CA
IC523 CA
IC524 CA
IC525 CA
IC526 CA
IC527 CA
IC528 CA
IC529 CA
IC530 CA
IC531 CA
IC532 CA
IC533 CA
IC534 CA
IC535 CA
IC536 CA
IC537 CA
IC538 CA
IC539 CA
IC540 CA
IC541 CA
IC542 CA
IC543 CA
IC544 CA
IC545 CA
IC546 CA
IC547 CA
IC548 CA
IC549 CA
IC550 CA
IC551 CA
IC552 CA
IC553 CA
IC554 CA
IC555 CA
IC556 CA
IC557 CA
IC558 CA
IC559 CA
IC560 CA
IC561 CA
IC562 CA
IC563 CA
IC564 CA
IC565 CA
IC566 CA
IC567 CA
IC568 CA
IC569 CA
IC570 CA
IC571 CA
IC572 CA
IC573 CA
IC574 CA
IC575 CA
IC576 CA
IC577 CA
IC578 CA
IC579 CA
IC580 CA
IC581 CA
IC582 CA
IC583 CA
IC584 CA
IC585 CA
IC586 CA
IC587 CA
IC588 CA
IC589 CA
IC590 CA
IC591 CA
IC592 CA
IC593 CA
IC594 CA
IC595 CA
IC596 CA
IC597 CA
IC598 CA
IC599 CA
IC600 CA
IC601 CA
IC602 CA
IC603 CA
IC604 CA
IC605 CA
IC606 CA
IC607 CA
IC608 CA
IC609 CA
IC610 CA
IC611 CA
IC612 CA
IC613 CA
IC614 CA
IC615 CA
IC616 CA
IC617 CA
IC618 CA
IC619 CA
IC620 CA
IC621 CA
IC622 CA
IC623 CA
IC624 CA
IC625 CA
IC626 CA
IC627 CA
IC628 CA
IC629 CA
IC630 CA
IC631 CA
IC632 CA
IC633 CA
IC634 CA
IC635 CA
IC636 CA
IC637 CA
IC638 CA
IC639 CA
IC640 CA
IC641 CA
IC642 CA
IC643 CA
IC644 CA
IC645 CA
IC646 CA
IC647 CA
IC648 CA
IC649 CA
IC650 CA
IC651 CA
IC652 CA
IC653 CA
IC654 CA
IC655 CA
IC656 CA
IC657 CA
IC658 CA
IC659 CA
IC660 CA
IC661 CA
IC662 CA
IC663 CA
IC664 CA
IC665 CA
IC666 CA
IC667 CA
IC668 CA
IC669 CA
IC670 CA
IC671 CA
IC672 CA
IC673 CA
IC674 CA
IC675 CA
IC676 CA
IC677 CA
IC678 CA
IC679 CA
IC680 CA
IC681 CA
IC682 CA
IC683 CA
IC684 CA
IC685 CA
IC686 CA
IC687 CA
IC688 CA
IC689 CA
IC690 CA
IC691 CA
IC692 CA
IC693 CA
IC694 CA
IC695 CA
IC696 CA
IC697 CA
IC698 CA
IC699 CA
IC700 CA
IC701 CA
IC702 CA
IC703 CA
IC704 CA
IC705 CA
IC706 CA
IC707 CA
IC708 CA
IC709 CA
IC710 CA
IC711 CA
IC712 CA
IC713 CA
IC714 CA
IC715 CA
IC716 CA
IC717 CA
IC718 CA
IC719 CA
IC720 CA
IC721 CA
IC722 CA
IC723 CA
IC724 CA
IC725 CA
IC726 CA
IC727 CA
IC728 CA
IC729 CA
IC730 CA
IC731 CA
IC732 CA
IC733 CA
IC734 CA
IC735 CA
IC736 CA
IC737 CA
IC738 CA
IC739 CA
IC740 CA
IC741 CA
IC742 CA
IC743 CA
IC744 CA
IC745 CA
IC746 CA
IC747 CA
IC748 CA
IC749 CA
IC750 CA
IC751 CA
IC752 CA
IC753 CA
IC754 CA
IC755 CA
IC756 CA
IC757 CA
IC758 CA
IC759 CA
IC760 CA
IC761 CA
IC762 CA
IC763 CA
IC764 CA
IC765 CA
IC766 CA
IC767 CA
IC768 CA
IC769 CA
IC770 CA
IC771 CA
IC772 CA
IC773 CA
IC774 CA
IC775 CA
IC776 CA
IC777 CA
IC778 CA
IC779 CA
IC780 CA
IC781 CA
IC782 CA
IC783 CA
IC784 CA
IC785 CA
IC786 CA
IC787 CA
IC788 CA
IC789 CA
IC790 CA
IC791 CA
IC792 CA
IC793 CA
IC794 CA
IC795 CA
IC796 CA
IC797 CA
IC798 CA
IC799 CA
IC800 CA
IC801 CA
IC802 CA
IC803 CA
IC804 CA
IC805 CA
IC806 CA
IC807 CA
IC808 CA
IC809 CA
IC810 CA
IC811 CA
IC812 CA
IC813 CA
IC814 CA
IC815 CA
IC816 CA
IC817 CA
IC818 CA
IC819 CA
IC820 CA
IC821 CA
IC822 CA
IC823 CA
IC824 CA
IC825 CA
IC826 CA
IC827 CA
IC828 CA
IC829 CA
IC830 CA
IC831 CA
IC832 CA
IC833 CA
IC834 CA
IC835 CA
IC836 CA
IC837 CA
IC838 CA
IC839 CA
IC840 CA
IC841 CA
IC842 CA
IC843 CA
IC844 CA
IC845 CA
IC846 CA
IC847 CA
IC848 CA
IC849 CA
IC850 CA
IC851 CA
IC852 CA
IC853 CA
IC854 CA
IC855 CA
IC856 CA
IC857 CA
IC858 CA
IC859 CA
IC860 CA
IC861 CA
IC862 CA
IC863 CA
IC864 CA
IC865 CA
IC866 CA
IC867 CA
IC868 CA
IC869 CA
IC870 CA
IC871 CA
IC872 CA
IC873 CA
IC874 CA
IC875 CA
IC876 CA
IC877 CA
IC878 CA
IC879 CA
IC880 CA
IC881 CA
IC882 CA
IC883 CA
IC884 CA
IC885 CA
IC886 CA
IC887 CA
IC888 CA
IC889 CA
IC890 CA
IC891 CA
IC892 CA
IC893 CA
IC894 CA
IC895 CA
IC896 CA
IC897 CA
IC898 CA
IC899 CA
IC900 CA
IC901 CA
IC902 CA
IC903 CA
IC904 CA
IC905 CA
IC906 CA
IC907 CA
IC908 CA
IC909 CA
IC910 CA
IC911 CA
IC912 CA
IC913 CA
IC914 CA
IC915 CA
IC916 CA
IC917 CA
IC918 CA
IC919 CA
IC920 CA
IC921 CA
IC922 CA
IC923 CA
IC924 CA
IC925 CA
IC926 CA
IC927 CA
IC928 CA
IC929 CA
IC930 CA
IC931 CA
IC932 CA
IC933 CA
IC934 CA
IC935 CA
IC936 CA
IC937 CA
IC938 CA
IC939 CA
IC940 CA
IC941 CA
IC942 CA
IC943 CA
IC944 CA
IC945 CA
IC946 CA
IC947 CA
IC948 CA
IC949 CA
IC950 CA
IC951 CA
IC952 CA
IC953 CA
IC954 CA
IC955 CA
IC956 CA
IC957 CA
IC958 CA
IC959 CA
IC960 CA
IC961 CA
IC962 CA
IC963 CA
IC964 CA
IC965 CA
IC966 CA
IC967 CA
IC968 CA
IC969 CA
IC970 CA
IC971 CA
IC972 CA
IC973 CA
IC974 CA
IC975 CA
IC976 CA
IC977 CA
IC978 CA
IC979 CA
IC980 CA
IC981 CA
IC982 CA
IC983 CA
IC984 CA
IC985 CA
IC986 CA
IC987 CA
IC988 CA
IC989 CA
IC990 CA
IC991 CA
IC992 CA
IC993 CA
IC994 CA
IC995 CA
IC996 CA
IC997 CA
IC998 CA
IC999 CA
IC1000 CA

© 1982
HIGH LEVEL
HARDWARE LTD
OR-MIC-REV-A

OR-MIC-REV-Y
HIGH LEVEL
HARDWARE LTD
© 1985

1680 HIGH LEVEL HARDWARE LIMITED
OR-MIC-REV-B
OR-MIC-REV-C

OPTO-PIC
OR-MIC-REV-D
HIGH LEVEL
HARDWARE LTD
1983

15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

Control store

The control store is built using high speed static RAM devices. This is normally loaded at bootstrap time, allowing the machine to be fully user microprogrammable. The control store cycle time is 125 ns, equal to the fastest CPU cycle. The architecture allows for up to 32 Kwords (64 bit wordlength) of control store, although current memory technology limits this to 8 Kwords. The standard configuration has 4 Kwords on a single circuit board. Parity checking is provided. To achieve the required speed at reasonable cost, a two-level pipeline is employed around the control store.

Main memory

Main memory is organised as 32-bit words with two-way interleaving, allowing 64 bits of data to be fetched or stored in one operation. In normal operation main memory is accessed via a virtual memory management unit.

Each main memory module contains 0.5 Mbytes of storage with parity protection, constructed using 64K dynamic MOS RAMs. Random access cycle time is 500 ns per 32-bit word but multi-word transfers, for example to and from the cache, yield an effective cycle time of 250 ns per 32-bit word (16 Mbytes per second). The memory modules decode 26-bit physical word addresses and within this limit total memory capacity is restricted only by the number of available system bus slots; depending on the I/O configuration of the system, up to 10 Mbytes of physical memory can be installed.

Logical to physical address translation is carried out using a set of address translation tables. Each process has access to three independently extensible regions of memory, used typically for program, heap and stack. A fourth region is normally reserved for the operating system. The tables also contain a set of rights bits for each memory page giving full protection and supporting the implementation of demand paged virtual memory. The translation tables are cached in the CPU resulting, in most cases, in an overhead of only

one microinstruction when performing address translation. The page size which is fixed by the hardware, is 4 Kbytes. Each logical region can be up to 256 Mbytes.

I/O subsystems

The ORION I/O subsystems include a number of attached microcomputers to perform low level tasks such as running diagnostics and managing terminals and disks.

The diagnostic microcomputer is embedded within the CPU. Its functions include running a system confidence check when power is first applied, bootstrapping the CPU, and taking control should an unrecoverable control store parity error be detected. It can also be used to load new microcode dynamically whilst the machine is running. An RS-232C interface is provided to which a terminal can be attached. Extensive diagnostics can then be run in conjunction with special microcode to perform fault analysis in the event of a system failure. Problems can usually be isolated to one or two integrated circuits.

One or more intelligent I/O channels control peripheral activity. Each of these includes a full function microcomputer which performs control functions and housekeeping. Data transfers to and from peripheral devices take place via a direct memory access (DMA) path itself constructed using bit-slice microprocessors. This allows the full performance of the ORION memory system and of the peripheral device to be exploited, with the microcomputer able to take corrective action on soft I/O errors. Software on ORION communicates with the microcomputer using a high level message passing protocol.

A variety of Winchester technology disk subsystems is available, with capacities ranging from 40 to 160 Mbytes. For backup or archiving, a ¼ inch cartridge tape drive capable of storing up to 23 Mbytes per cartridge is used. One or more eight inch 1.2 Mbyte floppy disk drives and an industry standard ½ inch reel to reel magnetic tape drive can also be supplied as options. Please refer to the price list for further details.

Upgrade path

The ORION hardware has been designed to take advantage of new developments in semiconductor memory technology as soon as they become available. In particular the main memory modules will accept 256K by 1-bit dynamic memory devices with a change of three jumpers and the control store, cache memory and virtual address translation buffer will accept 64K bit or larger static devices. These new devices are expected to become available at reasonable cost within a year and will allow up to a fourfold increase in the maximum storage capacity of each of the memory systems. High resolution bit-mapped graphics and local area networking are under development.

Performance

The execution speed of C programs on ORION approximates to that on the VAX 11/750 super minicomputer (Digital Equipment Co.).

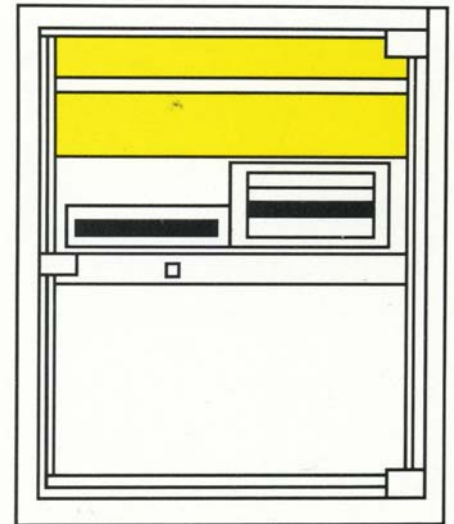
A local integer variable can be loaded or stored in as little as 600 ns. An ordinary branch instruction takes about 1.5 μ s including refilling the instruction cache. The high level statement $A := B * C$ is performed in approximately 7 μ s (32-bit integer operands) or 35 μ s (double precision floating point operands). A high level procedure call and return together take less than 3 μ s; note that this includes the time to create a new stack frame.

Although UNIX is a time sharing operating system it works well in a single user environment on a high performance processor. High Level Hardware have ported UNIX 4.1 bsd to ORION. This system, which was developed by the University of California at Berkeley, is derived directly from Bell Laboratories' UNIX 32V for the VAX 11/780, itself derived from UNIX version 7. 4.1 bsd is currently the only version of UNIX to include virtual memory management and is the primary system used at approximately 1000 VAX sites outside Bell Labs.

The nucleus of the UNIX operating system is memory resident and includes the code that supports system calls and maintains the file system. The file system is organised around a tree-structured directory hierarchy, with a similar method of access for devices, directories and data files. This structure makes it relatively easy for programs to perform input/output operations in a device-independent manner, buffering taking place invisibly within the operating system when required.

A distinguishing feature of UNIX is its powerful 'shell' command language interpreter. Using the shell the user can readily create

multiple processes that can run sequentially or in parallel and which may themselves create offspring processes. The processes can communicate with each other by means of asynchronous software interrupts (signals) and message buffers (pipes). The shell also supports redirection of program input/output. These features assist the user to break down the solution of his problem into a number of smaller parts, each of these smaller programs being run sequentially or concurrently. The ORION user has a choice of the standard Bourne shell or the Berkeley enhanced C shell.



The large number of UNIX utilities (tools) allows many jobs to be done by using a combination of existing tools without the need to 'descend' to a high level language. Among the available tools are: ed, a context-oriented text editor, and sed, a stream editor, both with a simple and regular command structure; vi, a screen editor; nroff, a comprehensive text formatter, and spelling and other utilities concerned with word processing; sort, a generalised sorting utility, and join and uniq which can be used with sort to manipulate relational data bases; grep, which efficiently searches a file for a pattern; diff, which compares text files; awk, a pattern scanning and processing language which is virtually a C interpreter; compiler writing aids such as the

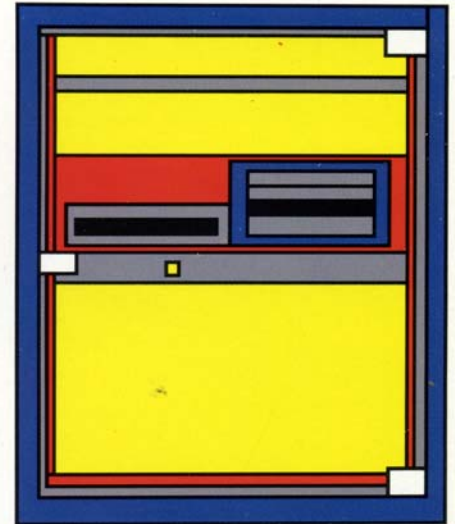
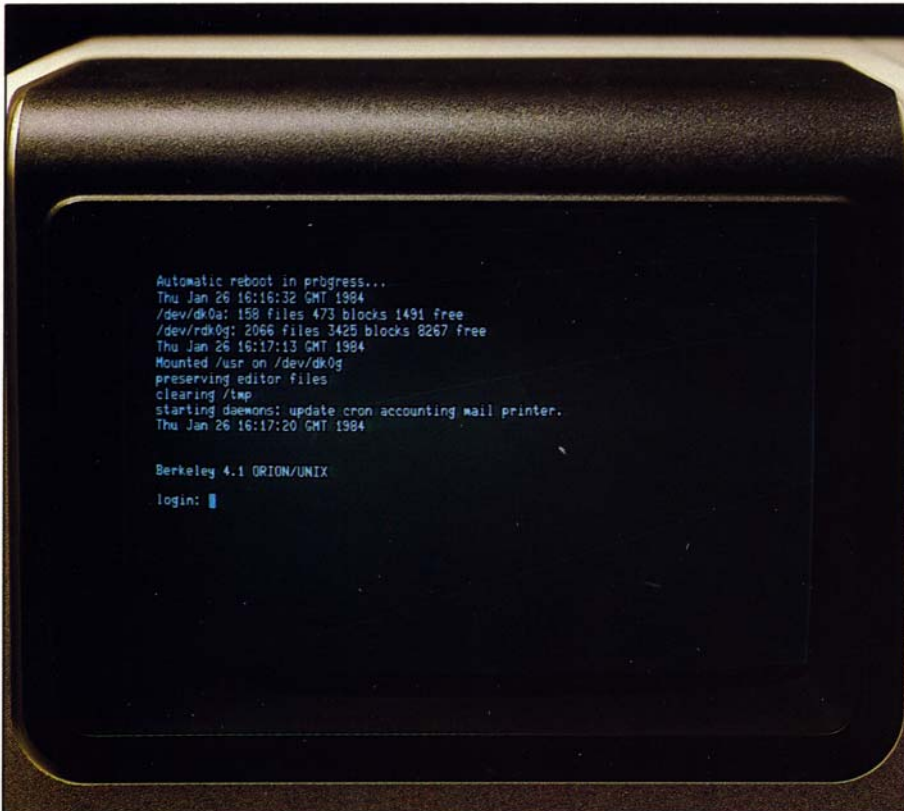
lexical analyser lex and the parser generator yacc; and many more.

The principal language available under UNIX is C, a general purpose systems implementation language that supports structured programming. The High Level Hardware C compiler is totally compatible with the UNIX standard C language. Particular effort has been expended to produce useful error diagnostics, an area greatly neglected in the past; the compiler includes many of the features of the lint C program verifier. A large number of other languages and utilities is also available, including BCPL and Fortran 77.

For specialized applications a range of microcode development tools is available. This

includes a reconfigurable micro-assembler, a micro-linker and a micro-librarian. Together they allow a modular approach to be used in micro-programming, and easy access to be gained to standard library routines. Since the ORION CPU supports multiple concurrent instruction sets, custom microcode can be developed, loaded, tested and used, entirely within the UNIX environment.

The ORION environment is attractive for implementing new languages. In particular, the task of writing a new code generator is relatively straightforward since the compiler writer is able to define an appropriate instruction set and the ORION assembler can easily be reconfigured to recognise the new instructions.



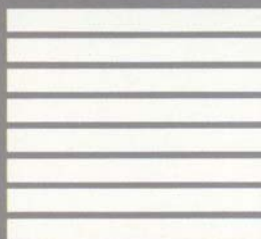
SPECIFICATION AT A GLANCE

PROCESSOR	
Type	32 bit CPU LSI bit slice technology User microprogrammable
Writable Control Store	64 bits by up to 32 Kwords
Granularity	4 and 16 Kwords
Cycle Time	125, 150, 175 or 200 ns, instruction dependent
Floating point	IEEE 32 and 64 bit formats
Decoding tables	Up to 16 independent instruction sets Pipelined instruction decoding
Cache memory	8 Kbytes expandable to 64 Kbytes, divided into 2 Kbyte segments 125 ns cycle time Stack or random access addressing

MEMORY	
Instantaneous virtual address space	1024 Mbytes
Physical address space	256 Mbytes
Page size	4 Kbytes
Translation buffer cache	2048 entries
Technology	MOS 150 ns, 64K or 256K dynamic RAM
Organisation	64 bits data plus 2 bits parity
Physical	Up to 40 Mbytes
Granularity	512 and 2048 Kbytes
Access length	4 and 8 bytes
Memory access bandwidth	16 Mbytes/sec sustained
Random access cycle time	500 ns (32 bits), 625 ns (64 bits)

SYSTEM INTERCONNECT	
Type	Synchronous, supporting multiple DMA channels
Data path	32 bits data, 1 bit parity
Clock	8 MHz
Bandwidth	32 Mbytes/sec sustained

I/O	
Type	Multiple intelligent microprocessor based peripheral controllers Microprogrammed bit slice DMA channels All data transfers via DMA to minimise CPU overhead



HIGH LEVEL HARDWARE

High Level Hardware Ltd.

PO Box 170, Windmill Road, Oxford OX3 7BN England
Telephone Oxford (0865) 750494