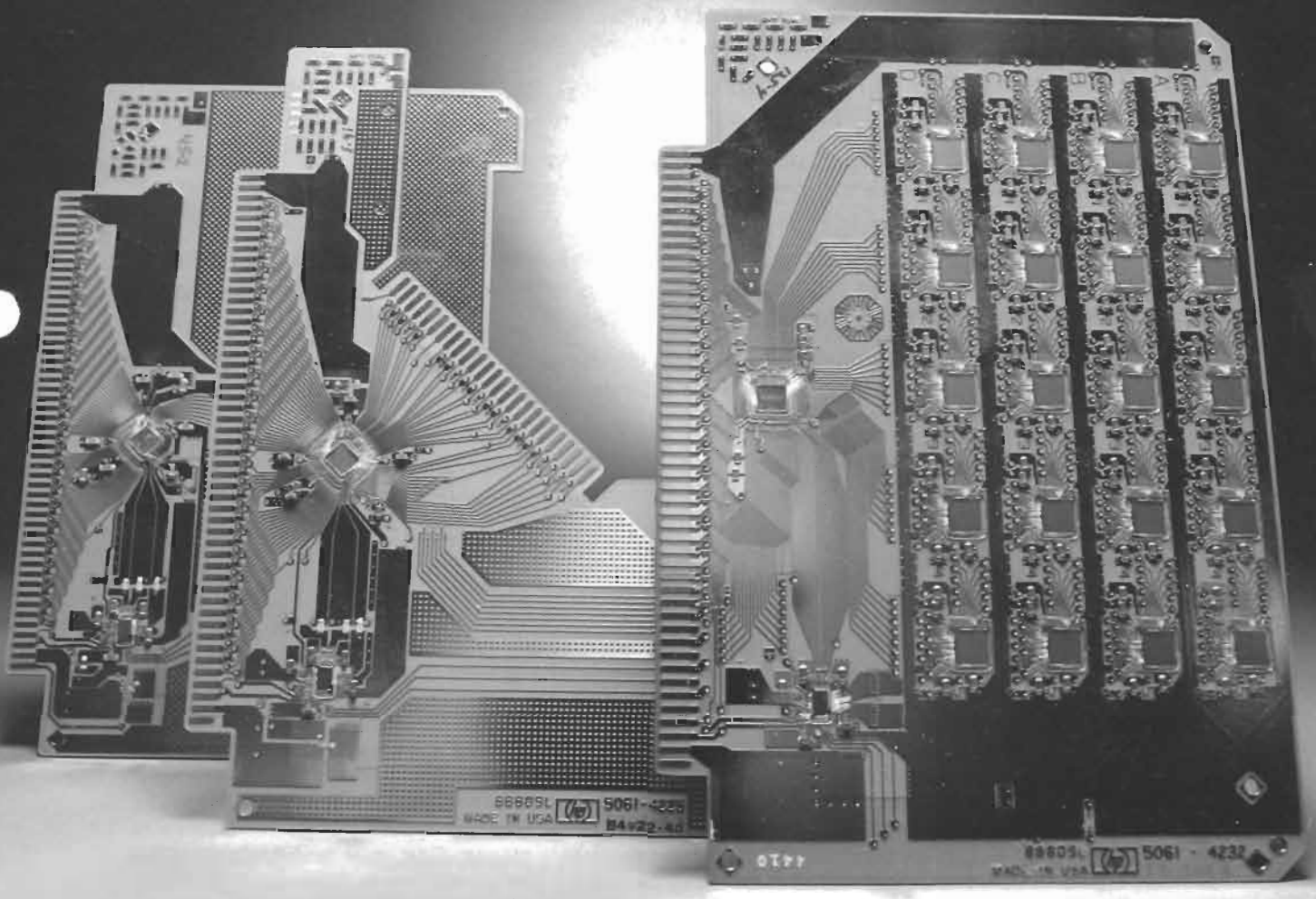
AUGUST 1983

HEWLETT-PACKARD JOURNAL



HEWLETT-PACKARD JOURNAL
Technical Information from the Laboratories of Hewlett-Packard Company

AUGUST 1983 Volume 34 • Number 8

Contents:

In this Issue:

Engineering productivity is under close scrutiny at many companies. Management wants more results for the money spent on R&D. Perhaps you remember, as I do, how the productivity of our engineers and scientists fairly spurted a few years ago when integrated circuit technology put scientific calculators in their pockets and powerful, interactive computers on their desktops. Can we do that again? The developments described in this issue are an attempt to do just that. A single integrated circuit chip packed with 450,000 transistors is the central processing unit (CPU) of a 32-bit computer system that fits in the category known as mainframes—the largest computers around. Four other equally dense chips support the CPU. A brand-new method of mounting the chips is given a new name—finstrates (see cover). A new process, NMOS III, uses pure tungsten to connect elements on the chip, a major departure from current IC practice. Test systems are designed into some of the chips because it's impractical to test anything so complex from the outside. Several issues from now we'll carry the story of the new HP 9000 Computer, a mainframe on a desktop for individual engineers and scientists that may give major impetus to the use of computers for design, engineering, simulation, and complex mathematical problem-solving. In this issue, you'll read about the IC and packaging technologies that make the HP 9000 possible. It's a remarkable achievement and a remarkable story.

-R.P. Dolan

VLSI Technology Packs 32-Bit Computer System into a Small Package

The new HP 9000 Computer is a compact, highly capable 32-bit computer system that incorporates five very dense integrated circuits made by a highly refined NMOS process.

by Joseph W. Beyers, Eugene R. Zeller, and S. Dana Seccombe

HOW DOES ONE GO ABOUT PACKING the power of a large mainframe computer into a desktop computer? Answering this question was only one of the many problems facing the HP design team given the assignment of developing a personal engineering design station with enough computing power to allow the entire design process to take place on an engineer's bench. Their answer is a fully integrated 32-bit processing system based on five custom VLSI circuits. This required the development of three key technologies:

- A 32-bit system architecture realized by using advanced circuit design techniques
- A state-of-the-art NMOS VLSI* process optimized for density and performance
- A new circuit board to dissipate the heat generated by the VLSI circuits and allow high-speed signal propagation.

System Overview

A block diagram of the 32-bit processing system is shown in Fig. 1. The system uses five different NMOS circuits operating at 18 MHz. These chips include a 32-bit CPU, an I/O processor, a memory controller, a 128K-bit RAM, and a clock driver (Fig. 2).

CPU. This single-chip 32-bit processor contains 450,000 transistors.[1] It is microprogrammed and has 9K 38-bit words of resident control store. It has twenty-eight 32-bit registers, a 32-bit ALU (arithmetic/logic unit) with multiply and divide logic, an N-bit shifter for bit extraction and alignment, and a seven-register port to the memory processor bus. The stack-oriented instruction set contains floating-point, string, and compiler optimization instructions. A 32-bit load instruction (including complete bounds checking) takes 550 ns and a 64-bit floating-point multiply takes 6 μs. Microinstructions can execute in 55 ns.

I/O Processor (IOP). The IOP is also microprogrammed and contains 4.5K 38-bit words of control store. It handles eight DMA (direct memory access) channels with a data rate of up to 5M bytes/s. It has sixteen software-programmable interrupt levels and can independently execute command sequences from memory.

Memory Controller. The memory controller chip can control 256K bytes of RAM, perform byte, half-word, word, and semaphore** operations, and do single-bit error correction and double-bit error detection of memory without any performance penalty. It can also heal up to 32 faulty locations and map logical to physical addresses in 16K-byte blocks.

RAM. The 16K×8-bit RAM chip contains 128K bits of random access memory organized with redundant rows and columns. It is pipelined and has a 165-ns access time and a 110-ns cycle time.

Clock. The clock chip generates two nonoverlapping 18-MHz clock signals from a 36-MHz sine wave. It can drive a 1500-pF load with a 6-ns rise time.

Memory Processor Bus

The CPU, IOP, and memory controller communicate via the memory processor bus (MPB). The protocol of this 44-line, 36M-byte/s bus can support up to seven CPUs or IOPs and fifteen memory controllers. This precharged dynamic bus is multiplexed between 29-bit addresses and 32-bit data words on alternating 55-ns clock cycles. The memory accesses are pipelined to allow sending up to two new addresses while the first data word is fetched from memory.

Because the 36M-byte/s data rate far exceeds the data requirements of one CPU, additional CPUs can be added and/or independent IOP operations can occur without a significant reduction in performance. Fig. 3 shows relative system performance as CPUs are added to the bus. Computation-intensive examples tend to approach the "ideal" line while heavy string operation performance tends to be lower than the average. The 32-bit CPUs are designed so that additional CPUs can be transparently added to the system. New tasks are usually assigned to

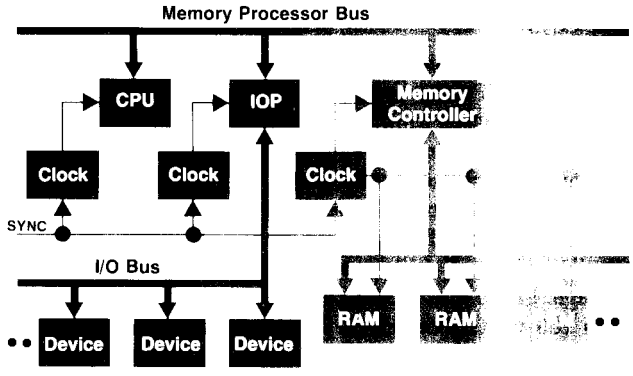


Fig. 1. *Block diagram of 32-bit computer system based on five VLSI circuits: CPU, IOP, clock, memory controller, and RAM.*

*n-channel metal oxide semiconductor, very large scale integration.

**Used to control accesses in a multiple processor system.

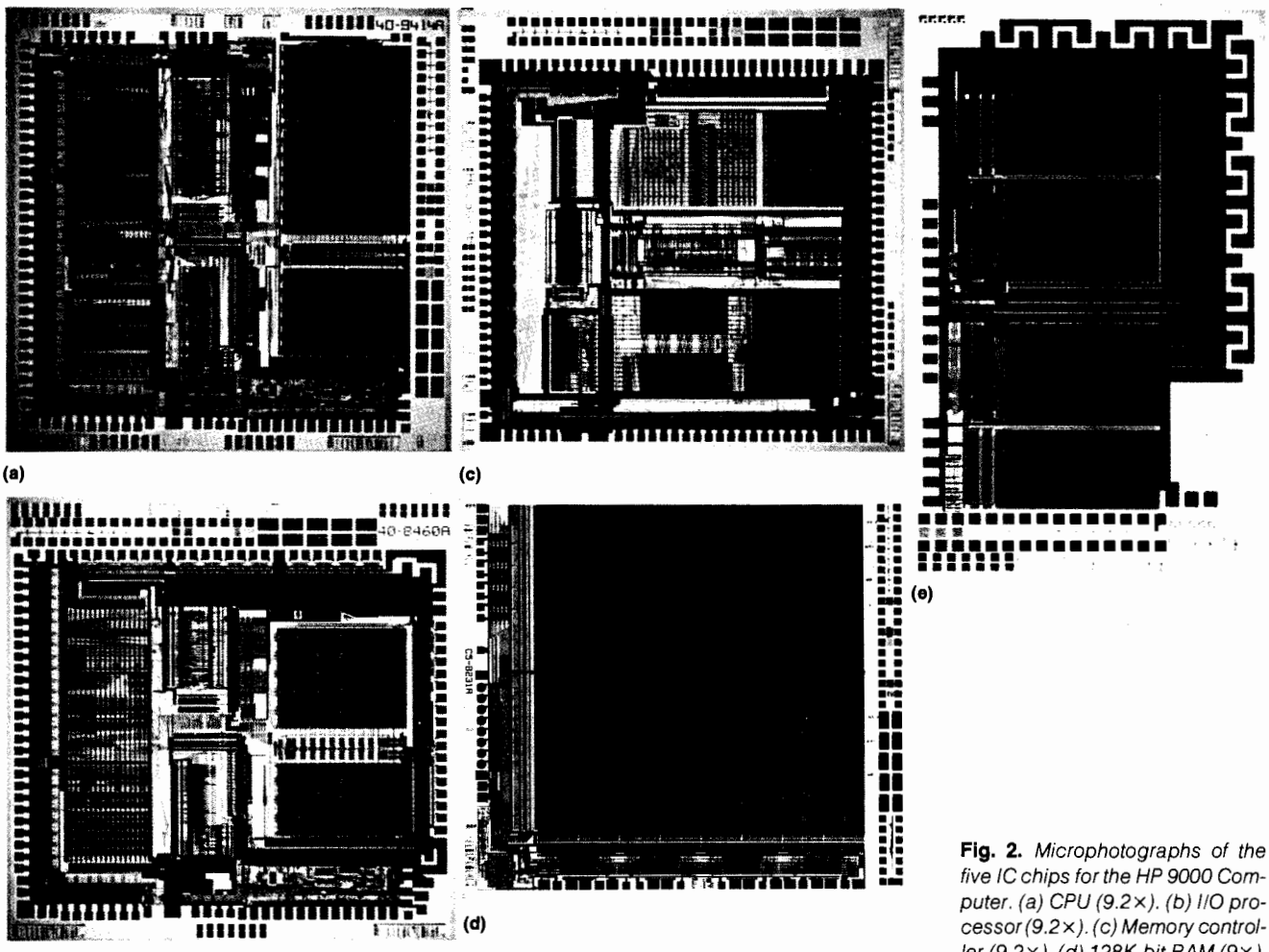


Fig. 2. *Microphotographs of the five IC chips for the HP 9000 Computer. (a) CPU (9.2×). (b) I/O processor (9.2×). (c) Memory controller (9.2×). (d) 128K-bit RAM (9×). (e) Clock (13×).*

whichever CPU on the bus is free. However, a CPU can be dedicated to specific tasks. For example, the I/O processors can be programmed to send interrupt requests to either a specific CPU or all CPUs.

Packaging

Fig. 4 shows a picture of how the above system is packaged for the HP 9000 product line. The package, called the Memory/Processor Module, can hold up to twelve circuit boards. This allows a system configuration of up to 2.5 megabytes of memory with one CPU and one IOP. Up to three CPUs and three IOPs can be used for increased performance by sacrificing some of the memory. Power is supplied through two connectors on the bottom of the package. In the worst-case configuration, the system dissipates 185 watts. Forced air flow is used to cool the VLSI circuits to below a worst-case junction temperature of 90°C.

The standard I/O bus exits the package through a slot in the bottom and two optional I/O buses exit through connectors on the module's door.

The VLSI chips are mounted on "finstrates." This name was coined from this circuit board's dual role as a cooling fin and chip substrate. The Teflon™ dielectric covering provides for low-capacitance interconnections and the finstrate's copper core spreads the heat away from the chips. The three types of finstrates used in the module are shown in Fig. 5. In the center is the CPU board, which contains the CPU chip and a clock chip. On the right finstrate are an IOP chip and a clock chip. The inset on the upper right side of this IOP finstrate is where a small printed circuit board containing a set of TTL buffers for driving the I/O bus is attached. The memory finstrate on the left contains a memory controller, a clock, and twenty 128K-bit RAM chips to provide 256K bytes of single-bit-error-correcting memory with a 36M-byte/s data rate.

The high-speed MPB exists only on the edges of the finstrates and the module's motherboard. The Memory/Processor Module also contains a small printed circuit card that generates a 36-MHz master clock sine wave driven to each of the twelve finstrate slots.

The system is mechanically self-contained. Electromagnetic interference (EMI) is suppressed by using spe-

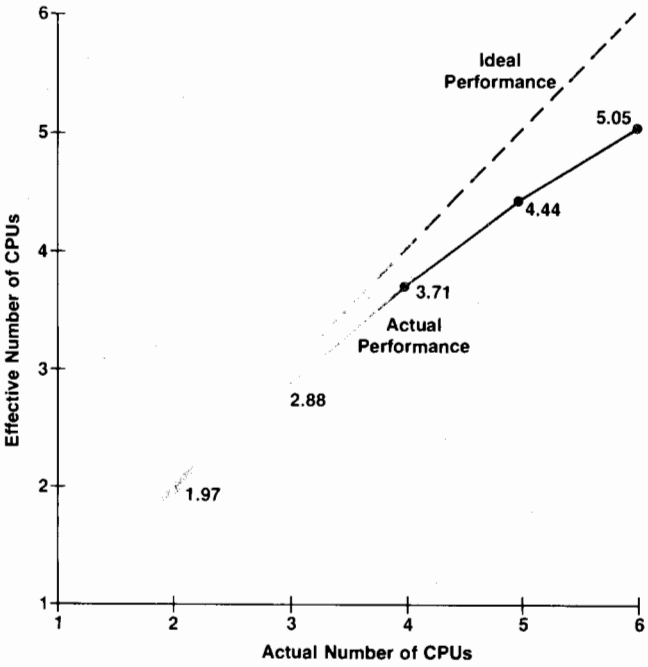


Fig. 3. *Multiple 32-bit CPU performance for Whetstone B1D benchmark.*

cial filters on the power supply connectors, honeycomb air filters at both ends, conductive door gaskets, and shielded I/O cables.

VLSI NMOS Process

The major process design goal was to develop a high-density, high-performance, highly reliable, production-volume, VLSI process. These goals were realized by the use of a modified n-channel silicon-gate MOS process featuring 3½ levels of interconnect: diffusion, polysilicon with buried contacts, and two levels of refractory metal.[2]

Integrated circuit densities are determined primarily by the minimum feature size. Lithographic considerations set this limit and resulted in layout rules and process capabilities that enable transistors to be fabricated with a minimum pitch of 2.5 μm (1.5-μm-wide lines spaced 1.0 μm apart). The unconventional contact-over-gate device structure allowed even tighter layout rules.

Another technique used to achieve the high circuit density is the use of two interconnect levels of refractory metal. Tungsten metallization was chosen because of its high conductivity and its resistance to electromigration.[3]

In addition to high density, transistor performance was emphasized. Special transistor characteristics (e.g., gate-to-drain overlap capacitance and threshold voltage versus backgate voltage dependence) required the use of self-aligned gates, shallow source and drain regions, and transistor threshold voltage implants.

Reliability and Self-Test

Besides maximum performance, high reliability and easy serviceability were also key design goals. These goals were achieved through several approaches. First, the NMOS process was designed for high reliability—silicon gates, refractory metal (no metal migration problems), and conservative design specifications that protect against gate hot-electron injection. Second, except for the I/O drivers and clock circuit, the system is fully integrated—92 integrated circuit chips for a system with one megabyte of memory. Third,

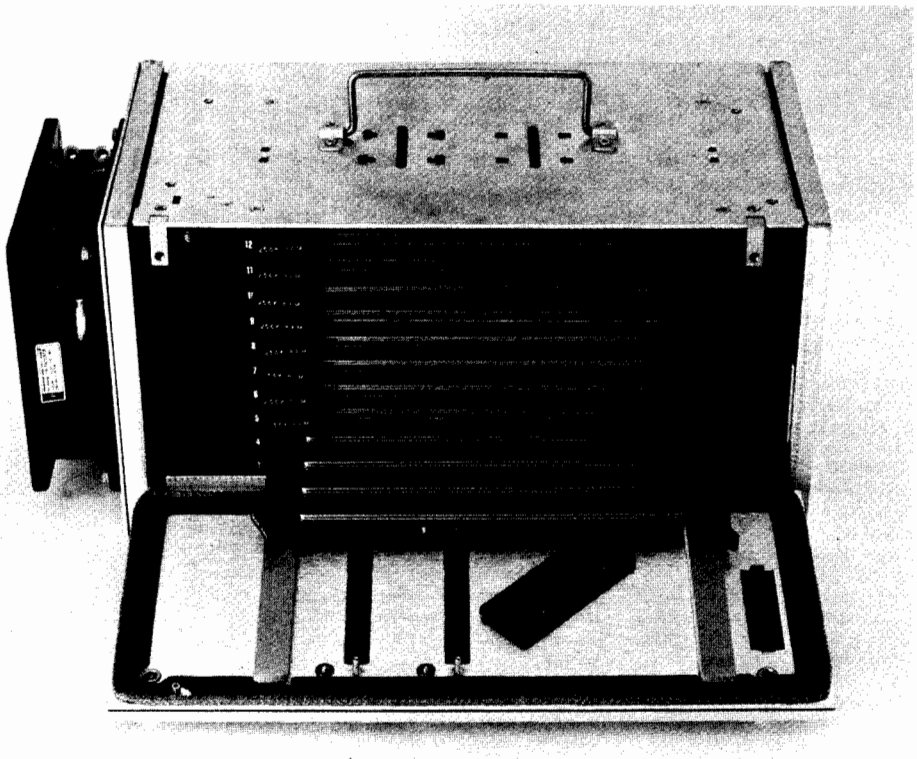


Fig. 4. *This package, called the Memory/Processor Module, can contain a complete 32-bit computer system with 2.5 megabytes of RAM. Performance can be enhanced by exchanging some of the memory boards for additional CPU and/or I/O processor boards.*

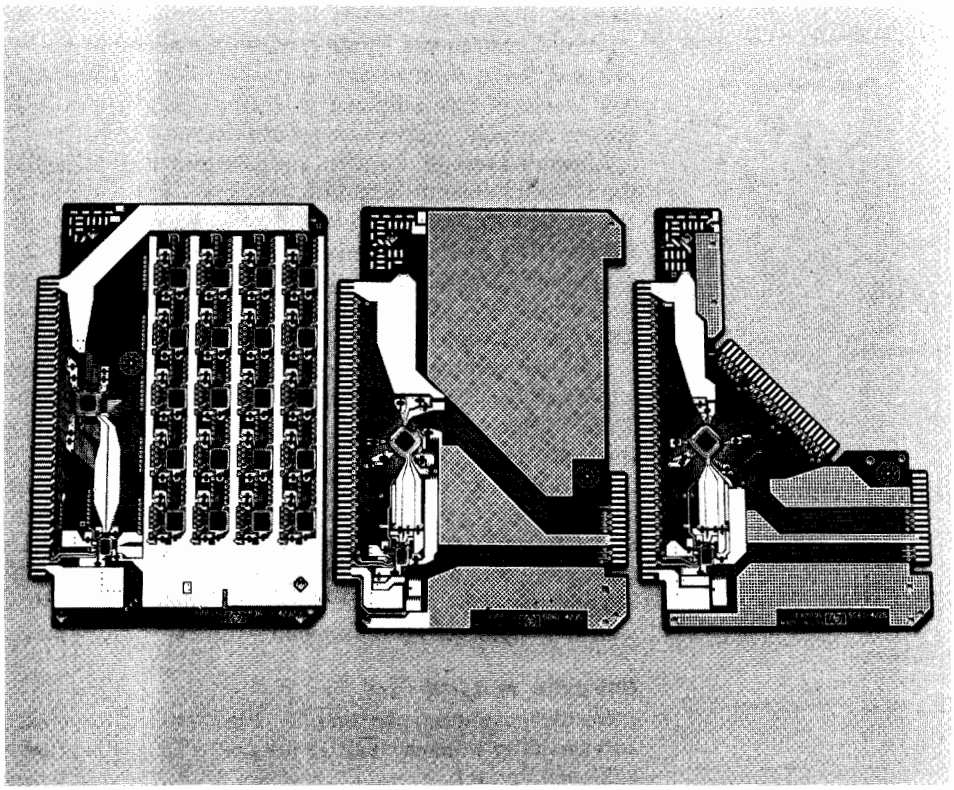


Fig. 5. *To dissipate the heat generated by the very dense VLSI chips, special boards, called finstrates, were developed. Shown from left to right are the 128K-byte RAM, CPU, and IOP finstrates.*

special reliability features such as single-bit error correction and double-bit error detection were incorporated into the system's architecture. Up to 32 faulty locations per memory finstrate can be healed by redirecting their contents to registers on the memory controller chip, and memory size can be degraded in 16K-byte blocks (the address space can be assigned arbitrarily within or between memory cards in 16K-byte increments). Any card can be easily removed and the system will still operate, assuming, of course, that there are still at least one CPU, one I/O processor, and one memory card left in the system.

In addition, the thorough internal self-test can quickly identify any bad cards. At power-on, each card is tested without the need for external software and if a fault is detected, appropriate LEDs (light-emitting diodes) are lit to indicate which card is defective.

References

1. J. Beyers, et al, "A 32b VLSI Chip," Digest of Technical Papers, 1981 IEEE International Solid-State Circuits Conference, THAM 9.1.
2. J. Mikkelson, et al, "An NMOS VLSI Process for Fabrication of a 32b CPU Chip," Digest of Technical Papers, 1981 IEEE International Solid-State Circuits Conference, THAM 9.2.
3. P.P. Merchant, "Electromigration: An Overview," Hewlett-Packard Journal, Vol. 33, no. 8, August 1982.

Acknowledgments

Bringing these complex technologies to production in late 1982 was the result of the determination and dedication of many people. Listed below are key contributors who transformed the initial design goals into a production reality.

The CPU chip design team included Joe Beyers, Kevin Burkhart, Dave Conner, Mark Forsyth, Mark Hammer, Tony Riccio, Harlan Talley, and Darius Tanksalvala. The CPU microcode was written by Jim Fiasconaro, Lee Gregory, Mike Kolesar, Bill Kwinn, Donovan Nickel, Rand Renfroe, and Larry Rupp. Fred Gross and Ed Weber wrote the IOP microcode and Mark Canepa, Ken Holloway, Bill Jaffe, Rich Kochis, Dave Maitland, Gary Taylor, and Don Weiss designed the IOP chip. The memory controller chip was developed by Joe Fucetola, Cliff Lob, Mark Ludwig, Bill Olson, Mark Reed, Tom Walley, and Jeff Yetter. Alexander Elkins designed the clock chip and Dale Beucler, Doug DeBoer, Lou Dohse, Charlie Kohlhardt, John Spencer, Bill Terrell, and John Wheeler designed the 128K-bit RAM chip.

Hai Vo-Ba was responsible for the layout of the finstrates. The Memory/Processor Module's mechanical design and the internal boards were designed by Madi Bowen, Jerry Kaufman, John Moffatt, Severt Shands, Gary Taylor, and Guy Wagner. Craig Mortensen and Ed Weber developed chip design tools which were run by the computer operators—Bev Raines, Binh Rybacki, and Kathy Schraeder. Special test hardware and software were developed by Brady Barnes, Richard Butler, Doug Fogg, Bob Miller, and Walt Nester.

The NMOS process development team included Rod Alley, Jim Barnes, Jeff Brooks, Doug Crook, Fung-Sun Fei, Barry Fernelius, Dave Forgerson, Tony Gaddis, Larry Hall, Norm Hendrickson, Ulrich Hess, Gary Hong, Dan Kessler, Rajendra Kumar, Fred LaMaster, Zemen Lebne-Dengel, Rick Luebs, Bob Manley, Carol McConica, Jim Mikkelson, John Moffatt, Ken Monnig, Don Novy, David Quint, Jim Roland, Dana Seccombe, Jodi Riedinger Smith, Paul Uhm, and Gene Zeller.

The photolithographic technology was developed by Howard Abraham, Skip Augustine, Keith Bartlett, Gary Hillis, J. L. Marsh, Rob Slutz, Mark Stolz, Rick Tsai, and Marty Wilson. Dave Allen, Kevin Funk and Glen Leinbach were responsible for the chip assembly process and the finstrate process was developed by Rick Euker, Deri Pratt, and Jeff Straw. The reliability of the chips and the system was the responsibility of David Leary, Arun Malhotra, and Henry Schauer.

This HP Journal issue focuses on the R&D portion of the technology development. However, the successful fabrication of VLSI chips in volume is equally determined by the manufacturing organization that supports it. We would like to thank our manufacturing organization for their enthusiastic support, especially Ray Cozzens, Cliff Doyle, Jim Drehle, Gary Egan, Jerry Harmon, and John Mahorney.

Special recognition and thanks go to our secretaries Carol Miller and Lavonne Gardner.

HP's Cupertino Integrated Circuits Operation and Hewlett-Packard Laboratories helped us solve some of our process development problems. In addition, special recognition should go to the key managers who continually supported this development. They include Jack Anderson, Doug Chance, Chris Christopher, Don Schulz, and Fred Wenninger.

An 18-MHz, 32-Bit VLSI Microprocessor

by Kevin P. Burkhart, Mark A. Forsyth, Mark E. Hammer, and Darius F. Tanksalvala

THE HEART OF HP's new 32-bit VLSI computer system is the Memory/Processor Module. The central processing unit in this module is an NMOS circuit containing 450,000 transistors on a single chip operating at a clock frequency of 18 MHz.[1] This compact CPU chip, which implements a 32-bit version of the HP 3000 Computer's stack-based architecture, is designed and microprogrammed to support multiple-CPU operations within a single Memory/Processor Module. Each CPU is capable of one-MIPS (million instructions per second) performance with very little performance degradation in multiple-CPU configurations.

Chip Organization

Fig. 1 shows the layout of the major functional components on the CPU chip. The data path area containing the ALU, register stack, and memory processor bus (MPB) interface is devoted to user- and system-level information processing. Two data buses within the data path link the ALU and the general-purpose register stack.

Register Stack. The register stack contains 31 registers used for machine instruction handling, general-purpose data storage, system addressing, and system status. Three registers are devoted to the machine instruction pipeline where special logic is included to predecode opcodes. Several registers in the stack hold the base and limit addresses for the data and program stacks in memory. Circuits are included to select the appropriate address base register automatically when address offsets are computed.

Four registers locally store the top values in the current data stack to allow fast access to often-used operands. Special-purpose hardware monitors one data bus for certain conditions such as zero, positive, and negative, and drives branch qualifier lines to the test-condition multiplexer. This data bus connects the register stack to the MPB interface, which handles all data transfers between the CPU and the memory and other processors.

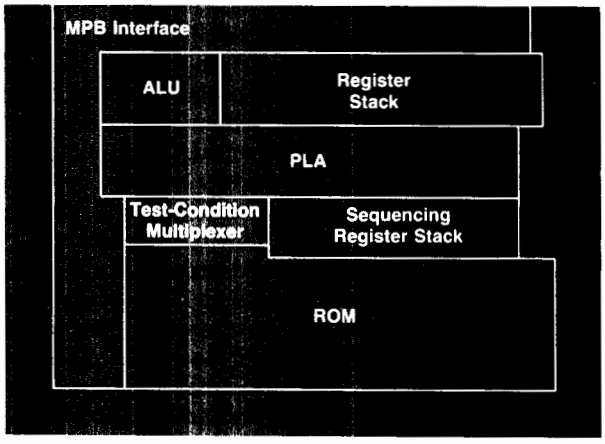


Fig. 1. *Outline of the 32-bit CPU chip indicating major sections of the chip's architecture.*

Since the MPB interface has its own data registers and control logic, internal CPU processes can initiate transfers and continue operation while the interface handles the MPB's complex synchronous protocol. The interface has dual-channel capability so that two completely different bus transactions can be in progress simultaneously.

ALU. The arithmetic logic unit provides a wide range of single-state, 32-bit arithmetic, logic, and shift operations. Operands can be selected from the main data path or the ALU's internal buses, and one of the operands can be complemented. The shifter provides up to 31-bit right/left arithmetic or logical shifting during one clock cycle. The logic function unit performs the OR, AND, and XOR operations on the operands and the adder provides their sum with carry-out and overflow bits. Master/slave result latches store intermediate results and return data to the register stack buses.

Sequencing Register Stack. Control circuits dominate the center area of the CPU chip. This control area contains a programmable logic array (PLA) microinstruction decoder, a test-condition multiplexer, and a 14-bit sequencing register stack which generates the 14-bit microinstruction addresses going to the control store. Address capabilities of the sequencing stack include short and long jumps, subroutine jumps and returns, traps to subroutines, address incrementing, and skips.

A mapping ROM generates microcode start addresses for all machine instruction formats and opcodes. The CPU machine instruction mapper includes an opcode PLA that can be programmed to select opcodes from any combination of bits in a 16-bit opcode. By altering the opcode PLA programming, the CPU can be remicroprogrammed to execute other stack architecture instruction sets. The output of the opcode PLA is an address into one of the 256 14-bit locations in the mapping ROM.

PLA. The central PLA decodes microinstructions and sends control signals to the data registers, ALU, MPB interface, and sequencing register stack. Microinstructions are divided into fields, each field specifying control for a different section of the CPU.

Test-Condition Multiplexer. An integral part of the PLA is the test-condition multiplexer. This multiplexer uses one microinstruction field to select a control qualifier from the data path registers, ALU, or MPB interface. Conditional microinstruction branches are taken by using the qualifier to control the address issued by the sequencing stack.

ROM. CPU control store consists of a 9216-word ROM organized in 32-word pages. During each clock state, the micropage address selects one page. A word address is issued during the following clock state to select one of the words on this page. With this ROM design, branches such as skips and short jumps on the current page execute with-

out interrupting the pipelined flow of microinstructions. Any jump off the current page results in one NOP clock state while the new page is selected. Microinstructions are transferred from this ROM to the central PLA by a 38-bit bus.

Typical Operation

The execution of a machine instruction begins when it is prefetched from memory and placed in the machine instruction pipeline registers (see Fig. 2). As the currently executing instruction completes, this prefetched instruction is moved up the pipeline into the next-instruction register and copied into the decoder-mapper in the microsequencing hardware. Meanwhile, execution of the immediately preceding instruction is initiated and another instruction is prefetched. Finally, this instruction is transferred to the current-instruction register and the appropriate starting microcode address is issued from the decoder-mapper. Instruction fetch, decode, and execution are performed in parallel except when a branch occurs.

Microcode from the control store ROM implements all machine instructions and performs the prefetch to keep the instruction pipeline full. The fields in each microinstruction are decoded by the central PLA, which sends control signals to the registers and ALU to move and process data. The MPB interface's dual-channel capability allows the currently executing instruction to fetch data on one channel while the instruction prefetch is in progress on the other channel.

Data fetched from memory is stored in general-purpose data registers in the CPU data path. Two parallel data buses within the data path simplify the transfer of operands to the ALU and the MPB interface. During each clock cycle, the ALU selects its operands and then performs an arithmetic operation and a logic or shift operation in parallel. Either or both results can be saved in the result registers, returned to the register stack, and/or used as operands during the next cycle. More complex operations such as multiplication or floating-point arithmetic are accomplished by sequences of microcode.

Features and Performance

The internal CPU data paths and registers, which carry and store user data and instructions, have full 32-bit widths. The CPU implements a stack-based architecture with a machine instruction set consisting of 230 instructions in 16-bit and 32-bit formats. Two 32-bit buses link the 31 general-purpose registers with the ALU and the MPB interface. The ALU has two internal 32-bit registers and three internal buses. Typical IEEE-standard floating-point execution time is 5.94 μs for a 64-bit addition and 10.34 μs for a 64-bit multiplication.

The CPU sends and receives data on the MPB which links the CPU to the other CPUs, I/O processors, and main memory in the Memory/Processor Module. The basic data word is 32 bits, but byte, half-word, and double-word load and store instructions are supported within a direct 500-megabyte address range.

The microinstruction bus linking the control store ROM and the PLA decoder transfers one 38-bit microinstruction every 55 ns. The control store ROM on the CPU chip contains 350K bits divided into 288 pages. Each page contains 32 words, each 38 bits wide. This ROM has a 70-ns access time, which includes a 20-ns final word select time.

Special microinstruction sequencing hardware provides addresses to the control store every 55 ns and minimizes the use of microcode fields for address control. Conditional jumps and subroutine calls in microcode are handled by the sequencing hardware to off-load these tasks from the ALU's main data path. The sequencing stack contains six registers, three incrementers, a comparator, a 10-by-14-bit trap ROM, a 256-by-14-bit mapping ROM, and an opcode PLA with 16 inputs, 120 product terms, and 8 outputs. The sequencing stack is interconnected by two 14-bit buses and one 5-bit true-complement word-select bus.

The pipelined microinstructions are divided into seven fields of five or six bits. Different microinstruction formats multiplex the different fields and constants into a single 38-bit word to enhance the efficiency of microcoded routines. These formats are decoded in the PLA based on the opcode in the 'special' field. The PLA microinstruction decoder, which consists of 55 inputs, 508 product terms, and 326 outputs, performs two-level decode logic in 55 ns.

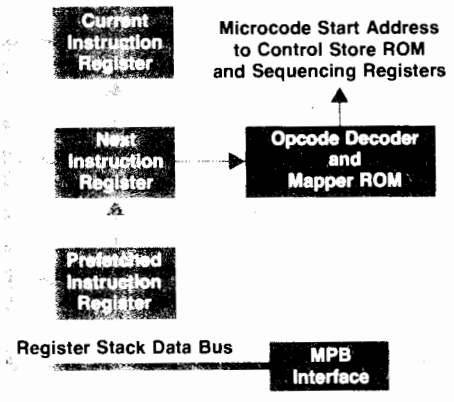


Fig. 2. *Block diagram showing arrangement of the machine instruction pipeline registers.*

Design for Testability

The complexity of the chip presented some very difficult testing challenges—fault coverage of a 450,000-transistor circuit with nearly 300,000 nodes, characterization at clock rates up to 24 MHz, verification of 350K bits of on-chip firmware, and providing process feedback from the first design in a new IC technology. Relying on commercially available LSI testers to solve these problems was not feasible because of their high cost, limited interactive diagnostic capabilities, and performance limitations. To provide a fast screen and detailed diagnostics under realistic operating conditions at low cost, it was necessary to incorporate most of the needed test capability into the chip's design.

Several key concepts are involved in the built-in testability of the CPU chip. A structured design methodology and a bus-oriented architecture allow substantial partitioning. Since all of the inputs and outputs of the individual circuits are connected to at least one of the major internal buses, every circuit can be individually controlled and observed by communicating with only a small number of data and control buses. A structured design separates circuits into distinct functional blocks, and a building-block approach

Instruction Set for a Single-Chip 32-Bit Processor

by James G. Fiasconaro

Fitting the entire CPU for a powerful 32-bit computer on a single manufacturable IC was a formidable task by any standard. This task was accomplished, in part, by encouraging the engineers who were designing and implementing the hardware and the instruction set to make the necessary tradeoffs between the two, but always with a thought towards the performance of the resulting chip. The present design is the result of many optimizing iterations. The hardware contains thirty-nine 32-bit registers, a 32-bit shifter, a 32-bit ALU, and 9K 38-bit words of microcode control store. It executes microcode at an 18-MHz rate.

The instruction set is stack-oriented. Each program has its own execution stack for allocating local variables, passing parameters to other procedures, saving the machine state on procedure calls, and evaluating expressions. There are instructions for pushing data onto the stack from memory, and for popping data from the stack and storing it in memory. Arithmetic instructions operate on the uppermost data words in the stack and leave their results on the stack. Instructions that operate from a set of parameters get these parameters from the top of the stack.

Segmentation is used to support virtual memory in the CPU instruction set. Every program can use up to 4096 code segments and 4096 data segments, and must use at least three segments—a code segment, a stack segment, and a global data segment (Fig. 1). Three pairs of 32-bit registers on the CPU point to the start and end of each of these three segments. These are the base and limit registers shown in Fig. 1. Another register, the program counter, points to the current instruction in the code segment, and two other registers point into the stack segment. The Q register points to the most recently pushed stack marker and the S register points to the uppermost 32-bit word in the stack. Four other registers on the CPU are used as a cache memory for the top four words in the stack, greatly reducing the number of reads and writes necessary to maintain the stack in memory. The information required to manage the segments used by each program is maintained in memory-resident tables. Each program has its own code and data segment tables and one common set of system code and data segment tables is shared by all programs.

Each code segment table entry contains the location and length of the segment, an absence bit, a privileged mode bit, a reference bit, and a use count. The use count indicates how many CPUs in the system are using the code segment at each point in time. Two primary instructions using the code segment tables are PCL (procedure call) and EXIT. PCL pushes a four-word stack marker, which contains the index register, the status register, the offset to the preceding stack marker, and the return address, onto the stack and transfers control to the new procedure. EXIT does the reverse and returns to the calling procedure. Both instructions also do a considerable amount of error checking.

Each data segment table entry contains the location and length of the segment, an absence bit, a privileged mode bit, a reference bit, a dirty bit, a write enable bit, a paged bit, a page-size field, link information, and a use count. Unlike code segments, data segments can be paged (with the exception of a program's stack and global data segments). Each program can access up to 4096 data segments through an external data segment pointer. If the segment is not paged, this pointer is interpreted as a 12-bit segment number and a 19-bit offset within the segment. (Segment length can be up to 2^{29} bytes.) If the segment is paged, this pointer is interpreted as a 31-bit virtual address with a 12-bit segment number, a page number, and an offset within the page. The page size can be chosen by the operating system in powers of two up to 2^{15} bytes. For paged segments, the data segment table entry points to a page table that contains a two-word entry containing location and status information for each page. Unpaged segments can be linked together and treated as a single logical entity, either by allocating the individual segments in consecutive data segment table entries, or by letting each data segment table entry point to the next entry in the chain.

Because the instruction set is stack-oriented, many instructions (e.g., ADD, SUB, AND, and OR) operate on the uppermost words in the stack and do not require any source or destination specification. Instructions that push information onto the stack and pop information from the stack use direct, direct indexed, indirect, or indirect indexed addressing. Direct addressing uses a base register and an offset specified in the instruction. Direct indexed addressing is similar except that the index register (a 32-bit two's-complement byte offset) is also added. Indirect addressing starts with the direct addressing calculation and fetches the indicated word from memory. This word is interpreted as a stack segment pointer, a global data segment pointer, or an external data segment pointer. Stack and global data segment pointers are simply offsets from the stack base and data base registers. External data segment pointers are evaluated through the data segment tables as described earlier. Indirect indexed addressing is like indirect addressing except that the index register is added after the indirect pointer is evaluated.

The instruction set provides a full repertoire of load and store instructions for bit, byte, half-word, word (four bytes) and double-word quantities using the addressing modes just described. All memory accesses using these instructions are bounds checked against program base and program limit, stack

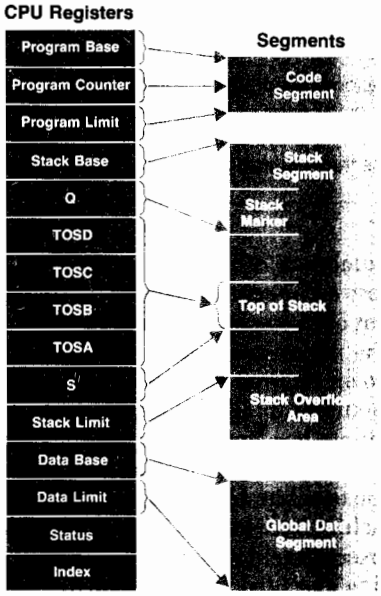


Fig. 1. *Use of the CPU registers for the code, stack, and global data segments.*

base and S register, data base and data limit, or the location and length information in a table as appropriate. A bounds violation causes a trap to the operating system. Stores into code segments are not allowed. In unprivileged mode, a user can access only the user's own code, stack, global data and external data segments. The instruction set also provides a set of privileged load and store instructions which use absolute addresses instead of segment base and offset information to access memory.

The primary data types supported by the instruction set are integers, floating-point numbers, and byte strings. Integers can be either 16-bit or 32-bit two's-complement numbers, 32-bit unsigned integers, or eight-digit unsigned decimal integers. The basic operations for add, subtract, multiply, divide, negate, compare, shift, and rotate are provided along with provisions to facilitate multiprecision (i.e., greater than 32-bit) integer arithmetic. Instructions that use one integer from the stack and an 8-bit immediate operand in the instruction are also provided.

Two types of floating-point numbers are supported. The first type includes 32-bit and 64-bit IEEE-standard binary floating-point numbers. The standard is met by supporting performance-critical operations directly in microcode and all other operations either directly by the operating system or by traps from microcode to the operating system. The second type is a 17-digit decimal format; only conversions between this format and the 64-bit IEEE-standard format are supported.

Both structured and unstructured byte string operations are supported. Unstructured strings are simply byte arrays. A set of move, scan, and translate instructions is provided to support this data type. Structured byte strings correspond to the string data types found in most high-level programming languages. These strings are accessed through a four-word string descriptor containing a pointer to the string, its maximum length, an index to the first byte of interest, and the number of bytes of interest. The current length of the string is stored in the first four bytes of the byte array containing the string. Instructions to load, concatenate, validate, and assign structured byte strings are supported.

The instruction set interacts with the operating system in two primary ways. The first way is through traps to code supplied by the operating system. When the microcode encounters a situation that it cannot handle, it traps to a prearranged entry point in a prearranged code segment. There are two broad categories of traps. The first category consists of error conditions. Examples include segment bounds or table length violations, privileged mode violations (attempts by unprivileged programs to execute privileged instructions or access privileged information), integer divide by zero, and system errors. The second category consists of situations that require operating system intervention. Examples include absent segments, pages, and page tables, stack overflow, floating-point mathematics traps, attempts to execute unimplemented instructions, and traps to support a set of high-level language debugging aids.

The second way the microcode interacts with the operating system is through a set of instructions. These instructions are primarily involved with task dispatching and I/O. This approach supports getting to and from the dispatcher and I/O driver code, assists some of the low-power operations which the dispatcher and I/O drivers must perform, and provides a special stack for the dispatcher and I/O drivers. The details of the algorithms used in the dispatcher and I/O drivers were left for the operating system to implement in machine code. This approach provides a good tradeoff between speed and flexibility.

The I/O interrupt handler provides sixteen I/O interrupt levels. At each level, I/O interrupts are handled on a first-come-first-serve basis. This is accomplished in cooperation with the I/O processor (IOP) chip by maintaining a linked list of all of the devices waiting for service at each priority level. The IOP logs devices at the end of each list and the CPU removes devices from the head of each list. Finally, provisions are made so that any CPU in a multiple-CPU system can handle any I/O interrupt.

Table I lists typical instruction times for a few CPU instructions. However, these times do not tell the whole story because up to three CPUs can be included in each Memory/Processor Module. Support for multiple-CPU systems was built into the instruction set from the very beginning. This support occurs primarily in the areas of dedicated memory locations, interrupt handling and manipulation of the code and data segment tables in memory. This support guarantees exclusive access to system information when necessary and facilitates implementation of efficient memory management in the operating system.

Table I
Typical Instruction Times

Instruction	Time (μs)
Direct Load	0.56
Integer Add	0.28
Integer Multiply	2.9
Integer Divide	9.4
64-Bit Floating-Point Add	6.0
64-Bit Floating-Point Multiply	10.4
64-Bit Floating-Point Divide	16.0
Procedure Call (to same segment)	3.3
Procedure Call (to different segment)	7.8

limits the number of different blocks.

To use these architectural features for testing purposes, a small amount of diagnostic support circuitry was added to the chip. The microinstruction register, one data register connected to an internal data bus, and an internal opcode bus were modified to allow loading or dumping serially through a single bus line. These registers can directly or indirectly control all of the internal data, address and control signals on the chip. Modifications to the microsequencing state machine provide the ability to halt or single-step microcode execution in a manner transparent to the microprogram being executed. This is done by using latches on all test qualifiers and recirculating data on internal buses.

A diagnostic interface port was added to facilitate control of the internal test features. This port consists of seven of the CPU chip's wire-bond pads: four opcode-bit pads, a serial I/O pad, an output pad, and a synchronizer input pad. The four opcode bits are connected to PLA inputs and used to control the serial shift registers and alter normal microinstruction sequencing. Data is loaded into and dumped from the internal registers via the serial I/O pad. The output of the test multiplexer can be observed via the output pad. The synchronizer input pad allows asynchronous communication between the diagnostic port and an external tester. The opcode bits are only executed once each time the synchronizer input is pulsed, which enables a relatively slow computer to communicate with and control a CPU running at a much higher clock frequency.

These features form an extremely powerful set of diagnostic tools. Operations that can be controlled through the

diagnostic port include setting microcode breakpoints, single-stepping microcode, loading and examining internal registers, and executing externally supplied microcode. The chip partitioning allows testing and characterization of a single circuit regardless of whether other circuits on the chip are functional. This capability proved to be essential in verifying a design of this complexity.

Testing chips in a production environment requires a high-speed pass/fail screen. To do this, a 100K-bit self-test microprogram was encoded into the CPU's ROM. This microprogram executes in twenty million clock cycles and outputs a series of pulses through the diagnostic port to indicate functionality of each major section of the chip. In addition to the standard instruction set, the self-test uses a set of microinstructions designed specifically for testing. Greater than 95% coverage of 'stuck-at.....' faults is achieved, and a variety of other potential defects such as storage node leakage, pattern sensitivities, and timing problems are covered as well. The self-test microprogram is executed whenever the chip is powered up, so it can be used for system verification and field tests besides wafer tests.

A feature of the architecture allows the CPU to communicate with itself via independent pad drivers and receivers connected to each of the MPB interface pads. Functional pad testing can thus be accomplished without the need for external, expensive, high-speed test electronics. However, if required, various loads can be connected to the circuit's pads during testing to simulate a system environment.

System-level hardware and software verification are also addressed by the built-in test features. A flip-flop controlled through the diagnostic port can put the CPU in a mode where it enters a transparent idle state at the completion of each machine instruction. This allows instructions to be single-stepped. Special microcode routines to provide breakpoint, variable tracing, and other software verification features are programmed into the CPU's ROM. Low-level system debugging can be done by executing microinstructions supplied through the diagnostic port to drive and monitor the system bus.

Acknowledgments

The CPU development effort was carried out at Hewlett-Packard's Systems Technology Operation in conjunction with the Engineering Systems Division. The initial phase of development was carried out under the guidance of Chris Christopher and Joe Beyers, and later development efforts were managed by Craig Mortensen and Lou Dohse.

Reference

1. J. Beyers, et al, "A 32b VLSI Chip," Digest of Technical Papers, 1981 IEEE International Solid-State Circuits Conference, THAM 9.1.

VLSI I/O Processor for a 32-Bit Computer System

by Fred J. Gross, William S. Jaffe, and Donald R. Weiss

HP'S 32-BIT VLSI computer system requires a high-performance input/output data path. The design objectives for the I/O path were to provide high data rates to peripheral devices to match the high performance of the CPU and to minimize the design effort. An I/O processor (IOP) able to control most I/O transactions without interfering with the CPU was chosen because it met the performance objective, and by using the same circuits and basic structure as the CPU chip, also met the second objective. As a side benefit, the first production runs of each chip served to test the other chip's design and establish a common reliability record for the shared circuits.

The I/O processor has an I/O bus bandwidth of 5.1M bytes/s when transferring at maximum rate. The IOP is capable of addressing eight device adapters, also known as I/O cards. Each device adapter has its own DMA (direct memory access) resource. There are sixteen levels of interrupt assignable to device adapters. The IOP is also capable of independently executing simple channel programs.

A microcode-controlled state machine gives the I/O processor enough power to perform all of its I/O tasks. A 38-bit microcode word with eight subfields allows simultaneous control of the I/O processor's internal registers and external control lines.

Operation

Operation of the I/O processor is directed by the CPUs in the computer system. The IOP alternately checks for a command from any CPU in the system or for a valid service request from any enabled device adapter. A CPU communicates with an IOP by sending it a command word and a data word. Embedded in the command word is the requesting CPU's return address. This allows all CPUs in a system to use any IOP. Commands sent from a CPU can set up DMAs, read registers on the IOP, or do direct I/O with a device adapter. Complex tasks that an IOP cannot handle independently result in CPU interrupts.

The IOP is connected to other processors and memory via

the memory processor bus (MPB). The MPB interface is a 32-bit pipelined interface with a synchronous protocol that allows overlapped memory fetches. It has its own registers and control logic, which hide its complex protocol from the IOP's register stack and control logic. This improves performance by allowing internal operation in parallel with memory operations. The MPB interface on the IOP chip is identical to the one on the CPU chip.

The IOP's I/O bus connects it to the device adapters. The new I/O bus protocol for the IOP is called HP-CIO for Hewlett-Packard Channel Input Output. The protocol was defined during the development of the IOP to provide a processor-independent, message-oriented bus.

During an IOP poll cycle, a device adapter enabled for service requests asserts a data line corresponding to its assigned address. The IOP latches the I/O bus responses, masks out any disabled devices, priority encodes the results, and then services the highest numerical address.

Service consists of transferring bytes or half-words (two bytes) which can be either data or commands. During the transfer, the address lines select one device adapter and the data direction line indicates who will be the data sender. The end of a transfer is signaled by the trailing edge of the I/O strobe IOSB, which the device adapter uses as a clock when receiving data or as a signal to assert the next data when sending data.

A poll cycle on which a single data transfer occurs is called a multiplex cycle and a cycle on which multiple data transfers occur is called a burst cycle (Fig. 1). A burst cycle increases I/O bus bandwidth because more bytes are transferred per poll cycle. The bandwidth reaches a maximum of 5.1M bytes/s in burst mode and 973K bytes/s in multiplex mode. When a poll is won by a device adapter for DMA, it has the option of asserting burst request BR. A device adapter in burst mode can take any number of transfers between two and thirty-two by asserting and then unasserting BR at the appropriate times. To reduce the lock-out time to an acceptable level, the IOP limits the number of transfers per poll cycle to no more than thirty-two.

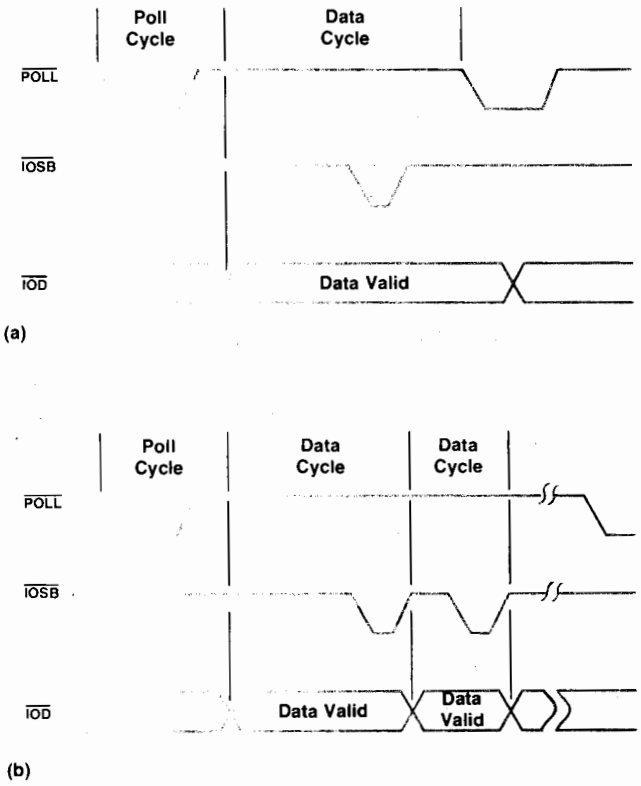


Fig. 1. *Timing diagram for (a) multiplex and (b) burst I/O cycles in the I/O processor.*

The width of the data word (byte or half-word) on the I/O bus is determined by the data sender. If the IOP is transferring in byte mode, channel byte CB is asserted to indicate to the device adapter that only the least-significant eight bits of the data bus are valid. If the device adapter is transferring in byte mode, device byte DB is asserted.

CPU interrupts are usually the result of either a DMA termination or a device adapter service request that cannot be handled by the IOP. When an interrupt occurs, the IOP records a device adapter interrupt request at the end of a linked list in memory for the interrupt level it is on. (This level is assigned by the CPU and stored in the status register for a particular device.) A message is then sent to the CPU to indicate that an interrupt was recorded for that particular interrupt level. When the CPU completes its current instruction, it services the highest-level list, starting at the list's beginning.

The CPU can configure itself to accept all interrupts, all interrupts above a certain level, or no interrupts. Each IOP has a register for enabling interrupts for any or all device adapters. A register on the IOP determines whether a particular CPU gets the interrupt request, or if all CPUs in the system get the interrupt request. In the latter case, the first eligible CPU available services the interrupt.

CPU commands not requiring a response can be placed in a list in memory for the IOP to execute. These lists are called channel programs. Each entry consists of a command word and a data word. The fourth word of the device reference table contains a pointer to the next executable command in the channel program. Each device adapter for every IOP has its own unique table in memory. When a status bit is enabled for a particular device adapter, the IOP executes one command per poll cycle when there are no CPU commands or service requests. A typical channel program allows multiple data transfers from different memory addresses to take place without interrupting the CPU. The logical completion of a channel program usually results in an interrupt.

I/O Processor Design

The IOP consists of a microcoded control section implemented with an internal ROM, an address sequencer, and a PLA decoder, a register stack of 44 registers connected by a common bus, the MPB interface, and an I/O interface. A block diagram of these sections on the IOP chip is shown in Fig. 2.

The control store is a 4608-by-38-bit, series-FET ROM with two-state pipeline access. In the first state of the pipeline, a page address is issued to select one 32-word page of the 144 possible pages. In the second state, the word address selects one of the 32 words on the selected page to be transferred to the PLA via a 38-bit bus. Branches within the current page do not interrupt the pipeline timing be-

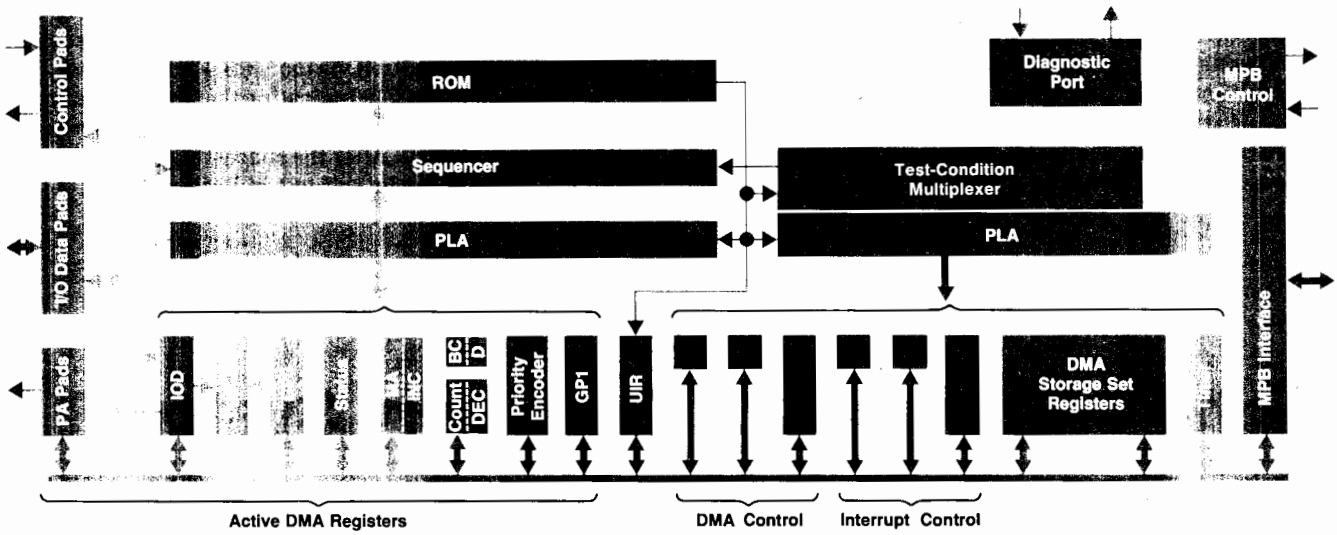


Fig. 2. *Block diagram of the I/O processor chip.*

cause the new word address is selected in the first state. Only jumps off the current page cause the pipeline to be restarted. The IOP only needs a ROM one-half the size of the CPU ROM. Structuring the CPU ROM into two equal arrays simplified the conversion to the IOP design.

The ROM address sequencer computes the 13-bit address of the next location to be fetched from ROM. In normal sequential access the previous address is incremented, but nonsequential addresses can be selected from either the previous instruction's branch target, the top of a subroutine stack, or a trap ROM. The address sequencer circuits are the same as those used for the CPU, but to conserve space, the opcode mapper circuit is deleted.

The PLA decodes the microcode words from the control store ROM and generates over 230 signals to control the IOP. The PLA is implemented with dynamic NOR-NOR logic for high performance, high density, and low power consumption. The IOP PLA is specially programmed for the IOP architecture, but the low-level building-block circuits for its design are identical to those used in the CPU's PLA.

The test-condition multiplexer controls conditional branching in the ROM address sequencer. It consists of static latches to sample and hold status information or external qualifiers, and a series-FET multiplexer to select the proper qualifier for the conditional branch.

The eight subfields in the 38-bit microcode word are classified as test, special, bus drive, bus receive, and four I/O controls. This word structure allows handling the unusual task of simultaneously directing data flow internal to the IOP while providing the appropriate I/O timing signals. In many cases I/O timing is adjusted by merely adding or deleting NOP (no operation) words to the microcode.

The register stack is made up of registers from 4 to 32 bits in length and a logic unit. The registers are divided into an active set that contains information about the DMA currently in progress, and a storage set that holds DMA information for all eight device adapters when DMA is not active. The active DMA registers consist of a memory address register with an incrementer, a count register with a decrementer, a burst count register with a decrementer, a status register with bits testable by the test-condition multiplexer, and an I/O data transfer register. When a DMA becomes active, the memory address, count, and status register values are transferred from the storage set to the active set. Each device adapter has a data register on the IOP chip to eliminate the need for a memory access before a data transfer. Response time to a DMA request is greatly improved since all information is contained on the IOP. When the transfer or transfers are completed, the new register values are stored in the storage set and the data buffer is filled or emptied.

The logic unit on the IOP replaces the powerful ALU found on the CPU. The bit-set/clear function performs a logical AND, OR, or exclusive-NOR between the IOD and GP0 registers and places the results on the common bus. The constant function can set any of the sixteen least-significant bits on the common bus. The compare function compares 32, 16, or 8 bits of the IOD and GP0 registers and sends the results to the test-condition multiplexer.

The I/O bus hardware consists of data drivers and receivers, address register and drivers, control drivers, and input qualifiers. The data drivers buffer the IOD register contents for output and are in a high-impedance third state during input operation while data is latched in the IOD register. The pad driver has a push-pull output stage designed to drive a load of 15 to 20 pF. The lines are buffered by external high-speed bipolar devices. This design has the advantages of being able to drive a large bus capacitance quickly without requiring a large IC chip area and of isolating the MOS lines from damage caused by static discharge. Input receivers consist of a protection device and a regenerative latch. The latch ensures proper system operation by resolving the input level before the internal test-condition multiplexer tests it.

The diagnostic port is a seven-line serial interface used for testing and diagnostics. It is identical to the one used on the CPU and allows the CPU and IOP to be tested on the same custom tester.

Self-Test

When power is applied, the IOP turns on its self-test

indicator and performs a self-test of its hardware using microcode routines programmed in its ROM. The routines are classified as internal test, I/O interface test, channel-to-channel test, and memory test. The internal test functionally tests all registers, operations, and sequencing. The I/O interface test sequentially sets and clears all output control lines and tests their level by a separate input. All inputs are driven high and low by special outputs and tested to ensure that they are functional. The channel-to-channel test causes the IOP to send data to itself via the MPB. Finally, if the CPU sends a message to the IOP indicating that there is working memory in the system, the IOP tests its ability to write to and read from memory. After the successful completion of all tests, the IOP turns its self-test indicator off.

High-Performance VLSI Memory System

by Clifford G. Lob, Mark J. Reed, Joseph P. Fucetola, and Mark A. Ludwig

IMPLEMENTING A HIGH-PERFORMANCE memory for HP's new 32-bit VLSI computer system requires the achievement of several important design goals to realize the full potential of this VLSI architecture. A dense resident memory and a large virtual address capability is desirable. A large memory bandwidth is needed to support multiple CPUs and I/O processors (IOPs) without creating bottlenecks. Also needed is the ability to do flexible memory operations such as byte, half-word, word, semaphore transfer, and refresh functions that are transparent to the CPU, IOP, and operating systems.

Fig. 1 shows a block diagram of a memory card for the 32-bit VLSI computer system. The key elements are the memory processor bus (MPB), MPB interface, memory controller chip, 128K-bit dynamic RAM chips, and clock chip.

Each memory card has twenty RAM chips organized in four rows of five chips each. Each RAM chip supplies 128K bits of memory storage and the memory card provides 256K bytes of total storage. Thus, a Memory/Processor Module can contain up to 2.5 megabytes of memory if it uses only one CPU and one IOP, and memory cards are inserted in all of the module's ten remaining empty slots.

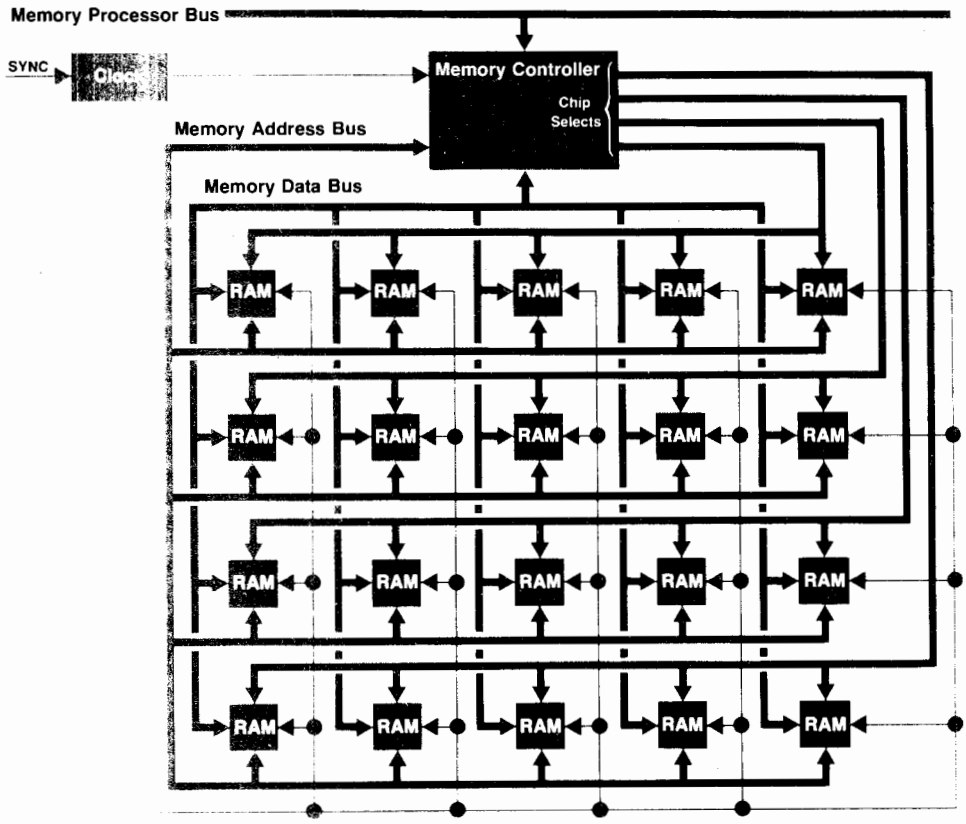


Fig. 1. *Block diagram of high-performance VLSI memory system.*

To achieve a large virtual address space, the 32-bit address has three bits of format control, allowing the remaining 29 bits to be used for addressing up to 2^{29} bytes. In addition, virtual memory support is provided by the CPU's microcode and instruction set.

The memory controller chip and the RAMs communicate via an 8-bit memory address bus, the 39-bit memory data bus, and a chip select (CS) line. Each row of five RAM chips has its own CS line, and the memory controller chip is connected to each row. Except when doing a refresh, the chip asserts only one CS line at a time. The memory address bus (MAB) and CS lines are driven only by the memory controller chip. The memory data bus (MDB) is bidirectional; write data is driven by the memory controller chip and read data is driven by the RAMs.

A large memory bandwidth is achieved through the MPB interface protocol, the pipelined nature of the RAM chip, and 18-MHz operation. Fig. 2 shows the timing for three read cycles. The internal pipeline design of the RAM allows it to accept a second address before handling data for the first address, and to issue read data nine million times per second. This allows the processors to issue three nonsequential data address requests without waiting for the first data word. Multiple processors, through a priority polling scheme, can interleave data.

After the polling sequence is completed, a memory address is sent on the bus and a read operation is indicated. The memory controller issues an 8-bit X address and a 6-bit Y address in succession on the memory address bus and generates the appropriate chip select CS. The RAM then decodes the address and outputs data onto the memory data bus. The memory controller corrects and aligns the 39-bit data word from the RAM row and outputs a 32-bit data word to the memory processor bus.

A single CPU can use no more than 65% of the bandwidth of the memory system. During normal operation, a CPU uses 30% of the bandwidth. In a system with multiple CPUs or CPUs with IOPs, the full bandwidth of 36M bytes/s can be completely used.

Important to packaging of the memory system is the finstrate board onto which the memory controller, RAM, and clock chips are mounted. Using forced-air cooling, the junction temperatures of the RAMs on an active memory card will not exceed 90°C, even under the following worst case conditions: 55°C ambient, 15,000 ft altitude, low fan voltage, and a fully loaded Memory/Processor Module. These low junction temperatures contribute to the excellent reliability of this memory system.

Flexible memory operations and high reliability and availability are implemented in the memory controller chip. This chip is controlled by a PLA (programmable logic array) for speed. It contains three separate synchronous state machines that control self-test, 'healing,' and normal memory controller operations. The chip dissipates up to five watts and has a total of 119 wire-bond pads.

In addition to refreshing the RAM chips, the memory controller performs the following functions:

- Aligning (reading and writing) of bytes and half-words
- Implementing semaphores by using the RAM capability of reading and writing in the same cycle
- Mapping logical addresses to physical memory
- Correcting single-bit errors and detecting double-bit errors on the fly
- Healing bad memory locations by replacing them with other on-chip memory locations
- Testing itself and the RAM chips.

Memory Controller Chip

Fig. 3 shows a detailed block diagram of the memory controller chip. The MPB interface handles the MPB protocol and routes addresses and data into and out of the chip. The mapper contains 32 CAMs (content addressable memories) and issues chip selects and part of the Y address. The MAB/CS drivers and multiplexer handle time multiplexing of X (row) and Y (column) addresses and read and write chip select signals. The MDB drivers and multiplexer handle time multiplexing of read data from, and write data to the RAMs.

The healer block also contains 32 CAMs. When an error is detected in memory, the healer places the physical address of that location in one of its CAMs so that substitution will be made for all subsequent accesses to that address.

The data manipulation section contains a Hamming encoder which attaches seven check bits to the 32-bit write data, a Hamming decoder which detects the position of a single-bit error and the existence of a double-bit error, a data corrector which corrects the single-bit error, and byte aligners which extract bytes and half-words from memory in nonword read operations and place bytes and half-words into memory for nonword write operations.

The MPB protocol is based upon polling for address/data bus cycles, and a master-slave synchronous handshake. During power-on, each CPU and IOP is assigned a nonzero channel number based on its physical position in the Memory/Processor Module. This channel assignment can be altered by the operating system to give highest priority to

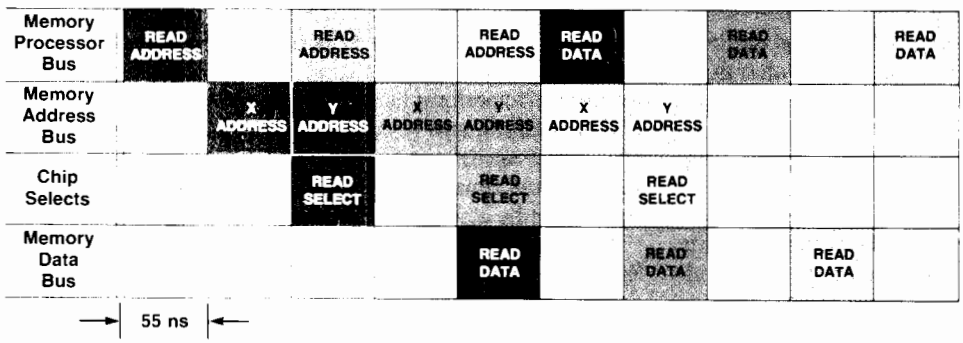


Fig. 2. *Memory system timing. The different shades indicate three separate, but interleaved read cycle operations.*

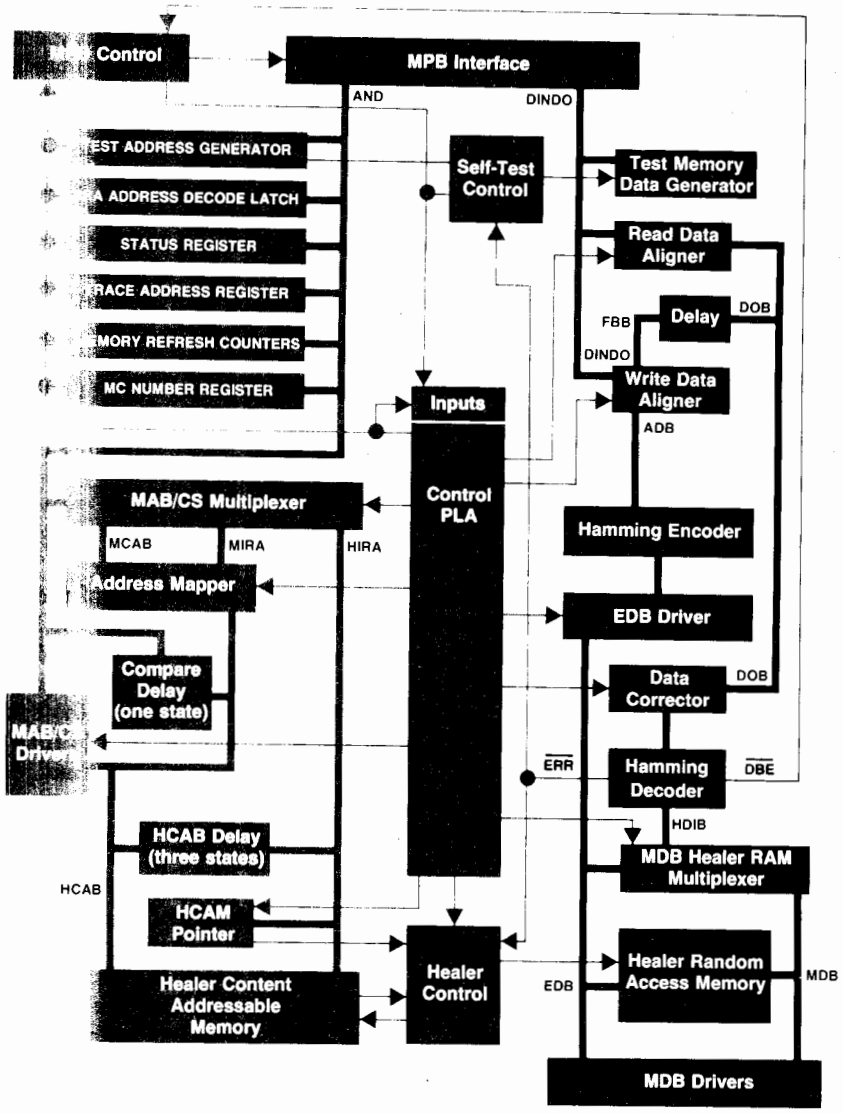


Fig. 3. *Block diagram of memory controller chip architecture.*

the processors requiring the most bandwidth for their tasks. The protocol allows for eight priority-assigned channels with 0 the highest priority and 7 the lowest priority.

Each memory controller is hardwired to channel 0, and is given a unique number (MC#) by the power-on procedure.

Figures 4 and 5 show protocol timing for read and write operations. The highest-priority channel responding to a poll wins the bus cycle, asserts the address on the next state,* asserts MCTL (master control) indicating a valid address, and, if this is a write operation, asserts $\overline{\text{WDBE}}$ (write/double-bit error). Two states later, the addressed slave asserts SCTL (slave control) to signify that it recognizes the address and currently is not busy. Three states after that, the data is asserted on the bus—by the slave if the transaction is a read, and by the master if it is a write. The slave asserts SCTL to signify that it can complete the transaction and the master asserts MCTL to signify that it can complete the transaction.

As a processor on the MPB, the memory controller chip has many characteristics very different from the CPU and IOP chips. Its master functions are simply to broadcast a message to the system, and to grab bus cycles for refreshing memory and for write operations. These chips are resident on channel 0 to guarantee that they win the bus poll cycle for these operations. In designing the chip, it was considered important that any master or slave processor functions interleave cleanly with pipelined memory accesses.

Each row of RAM chips on a memory finstrate provides 16K words of 39 bits each. Each word consists of 32 data bits and seven check bits which make up a modified Hamming code to allow single-bit and double-bit error detection and single-bit error correction. The 40th bit is not used.

A read address asserted on the MPB causes data to be returned five states later (Fig. 2). This includes time needed by the chip to perform its mapping functions, error detection, and data alignment. RAM access time is three states. However, a new access can be initiated every two states to give a 110-ns cycle time. The RAM is pipelined so that a second access can be started while another is still in progress.

Read Memory Operation

Fig. 4 shows timing for memory read operations. When

*In this article, one bus cycle is equal to two states.

an address is placed on the MPB, it is automatically placed on the AND (address, $\overline{\text{data}}$) bus during P1D (ϕ1 data). Parts of the address on the AND bus are processed simultaneously by many parts of the memory controller chip. An internal register access decode section checks to see if the channel field (bits 3 to 5) equals 0. It also captures address bits 6 to 10 and 23 to 29, which are pertinent to determining which memory controller chip register is accessed. The CAMs in the mapper compare bits 3 to 17 to their contents. Meanwhile, bits 18 to 29 go to the MAB/CS section and the healer, and bits 0, 1, 2, 30, and 31 go to the control PLA. In the MAB/CS section, bits 22 to 29 go out immediately to the

18-MHz Clock Distribution System

by Clifford G. Lob and Alexander O. Elkins

Designing the high-frequency distribution system to allow HP's new 32-bit VLSI processor to operate at 18 MHz proved to be a significant design challenge. The chips required 6V, two-phase, nonoverlapping clocks with rise times less than 6 ns and overshoot/undershoot less than 1V. It was decided early in the project that, because of area constraints, the processor chips would not buffer their clocks. However the RAM chips do provide some buffering. Hence the capacitive loading components vary from approximately 300 pF per phase for a CPU chip to approximately 30 pF per phase for a RAM chip. In addition, the capacitive loading presented is highly variable because of the dynamic circuits used and depends on which circuits are active. Worst-case tolerances produce capacitive specifications that can vary ±30% and cause unbalanced loads on each phase.

The first step in the design of the clock distribution system was the clock buffer chip. The clock buffer chip divides a 36-MHz signal and produces the two-phase, nonoverlapping clocks ϕ1 and ϕ2. Large capacitive drive is required since the RAM finstrate can load the clocks with 1500 pF per phase. In addition, the clocks are required to use a system sync signal to ensure that ϕ1 occurs on all finstrates simultaneously.

The chip size is 5.84 by 3.65 mm. Each large output transistor on the chip has a channel approximately 55,000 μm wide by 2.1 μm long and an output impedance of 0.5 ohm.

Fig. 1 shows a clock chip bonded to a finstrate and surrounded by chip capacitors used to reduce inductance and to bypass the supplies and ground. Peak currents of 2 to 3A occur when the clock switches. Multiple bonds interleaved with power supply and ground signals and multilayer chip metallization are used to reduce inductive and resistive effects.

Strip-line and microstrip techniques are used to distribute the clocks to the other chips on the finstrate. Careful attention was given to minimizing inductance because to achieve the clock specifications under worst-case variations, there must be less than 12 nH in series from the buffer to any chip. For comparison, a single wire bond contributes about 4 nH, and a 2-inch loop of wire is about 160 nH. Another inductance-reducing technique is the use of multiple taps per clock phase on the processor chip.

Fig. 2 shows actual clock waveforms as distributed with an earlier straightforward wiring approach, and as achieved with the tuned high-frequency design currently in use.

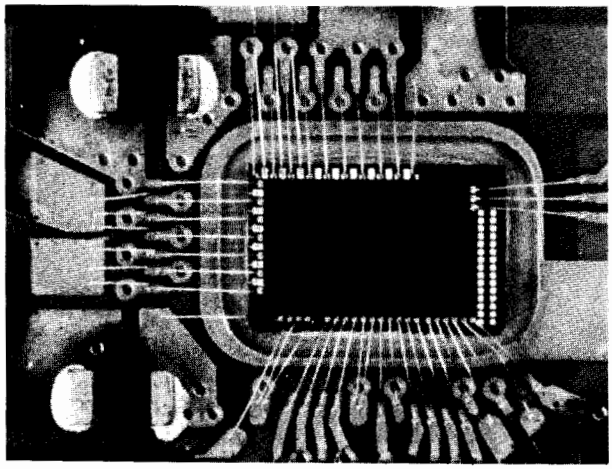


Fig. 1. *Photograph of 18-MHz clock buffer chip mounted in its cavity on a finstrate.*

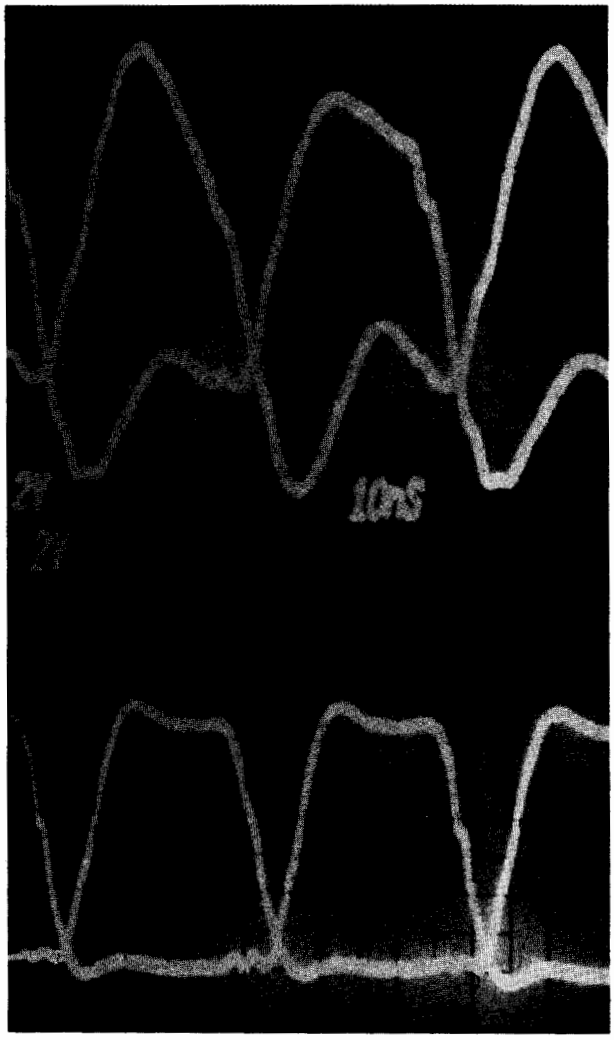


Fig. 2. *Clock waveforms using (a) an earlier straightforward wiring approach and using (b) the present tuned high-frequency design.*

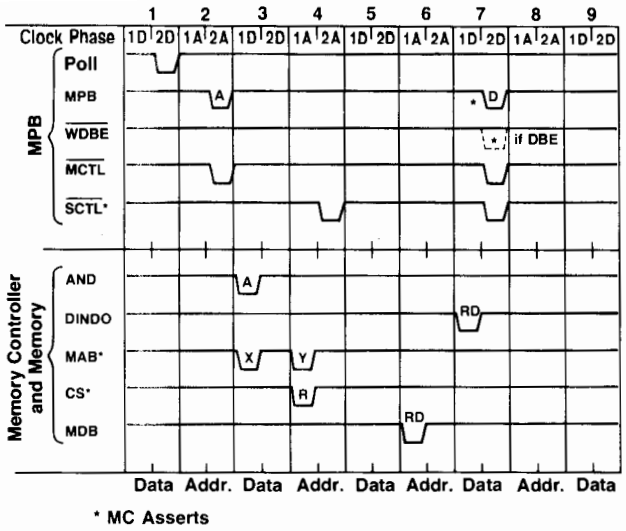


Fig. 4. *Memory read timing.*

RAMs as the X address (see Fig. 6).

The mapper's CAM outputs drive the mapper ROM, which generates chip selects and three bits of Y address during P1A (ϕ1 address). The operating system must ensure that logical-to-physical mapping assignments are unique because these outputs are wired-OR lines. Simultaneous matches in more than one mapper CAM can cause false physical addresses. An output by any mapper CAM causes a MY (my memory) condition to be sent to the control PLA and the MPB interface. An SCTL will be given on P2A (ϕ2 address) if this MY condition occurs and the control PLA determines that this is a memory operation.

The chip select and mapped Y address go to the healer and the MAB/CS section where they go immediately to the RAM as the read CS and as the Y address on MAB 1 to 3 (MAB 0 is not used in the Y address). Bits 18 to 21 from the original address were delayed and are now issued on MAB 4 to 7 to complete the Y address.

Available on the next P1A (state six), 39 bits of data from memory go to the seven decoding trees in the Hamming decoder. The decoder delays 32 of the bits for one clock phase while it generates seven syndrome bits. The syndrome bits are true during P2A when they are presented to the data corrector along with the 32 delayed bits of read data. If the syndrome bits are zero, the data corrector puts the delayed read data, unchanged, on the data output bus (DOB) during P1D.

If the syndrome bits are not zero, six of them provide a binary pointer for the data corrector to use to invert one of the read data bits. In the case of a single-bit error, the seventh syndrome bit (parity check of all 39 data bits) is a one and the bad bit is corrected. Should the parity check be a zero while the others indicate there is an error, a double-bit error has occurred. The bit inverted by the data corrector is neither of the error bits, so a signal ($\overline{\text{DBE}}$) about this is sent to the chip's MPB interface. Error detection and correction are accomplished in 40 ns.

Data on the data output bus goes to the read data aligner, and is also delayed one state to the fast-byte bus (FBB). In the read data aligner, signals from the control PLA select and right-justify bytes or half-words for nonword operations, or pass through the whole word for word operations. The read data aligner output goes to the data-in/data-out (DINDO) bus, which is connected to the MPB interface. The MPB interface now places the data, the second SCTL, and the $\overline{\text{DBE}}$ signal (if present) on the MPB.

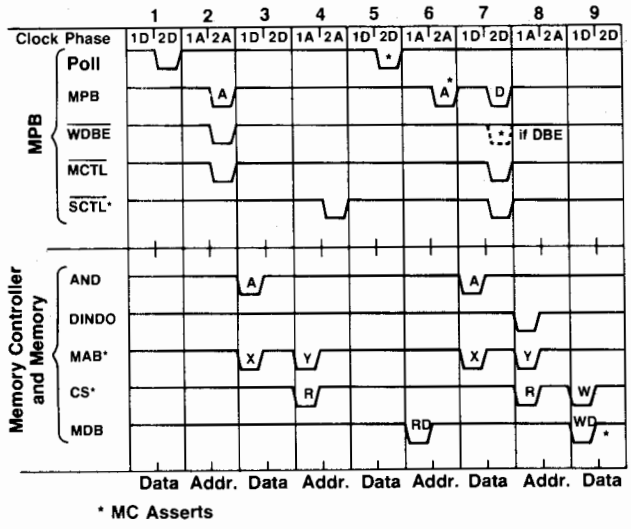


Fig. 5. *Memory write timing.*

Write Memory Operations

Fig. 5 shows some timing for memory write operations. If the memory operation is a write, the control PLA directs its MPB interface to poll for a bus cycle during state five. This obtains the bus cycle needed to put the write data into memory. In state six, it repeats on the MPB the original address, which was in a delay pipeline in the interface. Most of the read memory cycle is then repeated to accomplish the second half (completion) of the write.

Of course, several things are different from read operations. First of all, in state seven, the read data is not placed on the MPB, but rather the write data from the master processor is latched by the interface. On P1A during state eight, that data goes via the DINDO bus to the write data aligner. For nonword operations, the write data aligner merges the rightmost byte or half-word with the read data on the fast-byte bus by substituting it as specified by the address. In word operations, the read data is ignored and

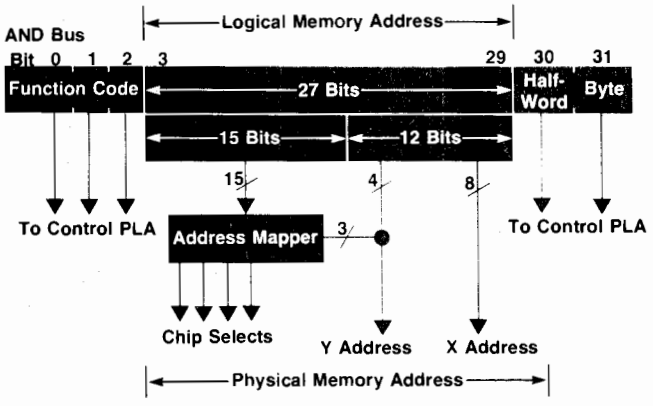


Fig. 6. *Address mapping process.*

the write data is passed through.

The output of the write data aligner goes to the Hamming encoder. The Hamming encoder delays the data by one-half state while it generates seven check bits from it. The check bits are appended to the 32 data bits. Its P2A output is sent to the encoded data bus. This bus goes to the memory data bus section, which presents the 39 bits to the RAM as write data during P1D in state nine. Also during P1D in state nine, the PLA has the MAB/CS section repeat the chip selects to the RAMs (write CS).

Semaphore Operation

A semaphore operation reads data from a memory location and sends it to the master processor while a minus one is written to that location. The master processor uses this to obtain control of a process. The semaphore operation follows the read operation with a few differences. First, the output of the Hamming encoder is turned off, so the encoded data bus and thus the write data on the memory data bus is left precharged (all ones, which is minus one in the signed integer format). Then during state five, the control PLA has the MAB/CS section repeat the chip selects as write CS. This makes the RAMs accept the minus one from the memory data bus as write data for that location.

Healer Operation

In the healer, bits 18 to 29 of the address on the AND bus are delayed one state, concatenated with the output of the mapper ROM, and presented through the healer cam address bus (HCAB) to the healer's CAMs (HCAMs) and to a pipeline that delays the bits for three states. The output of the HCAMs goes to the healer control PLA. A match by an HCAM causes a substitute memory location (an HRAM) to dump its contents to the HRAM output bus while the input to the Hamming decoder is switched from the memory data bus to the HRAM output bus.

The healer has a significant effect on system reliability and availability. Up to 32 words per memory finstrate can have hard errors without either shutting down the system because of known memory problems (uncorrectable hard errors) or potential memory problems (hard single-bit errors increasing the likelihood of uncorrectable errors).

Healing on the Fly

Healing on the fly is transparent to system performance. It improves system integrity by healing known memory errors as they are detected, without affecting the current transaction or bus bandwidth as a correction and write-back scheme would. It also provides a log of the error addresses, which is useful in the repair or replacement of a card.

A nonzero set of syndrome bits sends a signal $\overline{\text{ERR}}$ to the healer. $\overline{\text{ERR}}$ causes the HCAM pointer to increment during the next state. As $\overline{\text{ERR}}$ comes true, the address in the HCAB delay pipeline is dumped on the healer's internal register access (HIRA) bus while the HCAM indicated by the HCAM pointer is set from the HIRA bus. When the HCAM pointer is incremented, the next address goes to the next HCAM, leaving the error address in the previous HCAM—thus the error is healed.

Meanwhile, read data from memory is going to the HRAM input bus and being set into the HRAM corresponding to the HCAM indicated by the pointer. When the HCAM pointer is incremented, the read data is similarly captured in the HRAM, allowing the healer to have the same data in its substitute memory as was in the bad memory location. When the healer pointer count goes from 31 to 32, the healer is filled, a status bit is set, and a message is sent to the system.

Internal Register Access

To manage the healer and mapper, the system must be able to access their CAMs. It must also access the MC# (memory controller chip number) and status registers to turn on the system and the trace register for the system's debug aid. This is done with a channel access to channel 0. As previously mentioned, the address on the AND bus goes to an internal register access (IRA) decode section. This section checks the MC# field of the address (bits 6 to 9) against the memory controller chip's MC# and signals the control PLA if it matches. Memory controller chip IRA operations are handled with data going directly between the register and MPB interface. The main pathway is the data time (P1A) of the AND bus. The AND bus is connected via a multiplexer to the HIRA and mapper IRA (MIRA) buses in the healer and mapper.

Refresh

Since the NMOS RAM is dynamic, it must be refreshed. This is accomplished by having synchronized refresh counters on each memory controller chip. A refresh occurs every 16 bus cycles (32 states). The X address is changed for each refresh, but the CS signal is given each time to all RAMs. The MPB address time for the refresh cycle is normally wasted, so it is used as the time when a memory controller chip sends its messages to the CPU.

Memory Management

Also important to the system is being able to map and unmap memory blocks or to heal and unheal HCAMs. Thus each mapper CAM has a MAPOUT bit which disables that CAM no matter what the other contents of the CAM are. Each healer CAM has a HEALED bit, which when not set, disables that healer CAM.

Self-Test

The self-test section of the memory controller chip is almost as complex as microprocessors of six years ago. Occupying 5% of the chip's die area and containing about 7000 transistors, it does a 99% confidence test on the internal circuitry and the chip's MPB interface. Self-test on a good chip completes in less than 1.5 ms.

Self-test simulates the memory controller chip's MPB interface receiving addresses, data, and control signals from another processor. It does this by placing signals on the buses and control lines from the MPB interface to the internal registers, control PLA, and data manipulation section. Thus, the circuitry of the chip is tested as a functional unit rather than testing sections of circuitry separately.

Any failure halts the self-test. If no failure occurs, a column-march test is done on the RAM addresses controlled by each CAM in the mapper. Should Hamming decoding detect any error or the data be incorrect (as in the case of

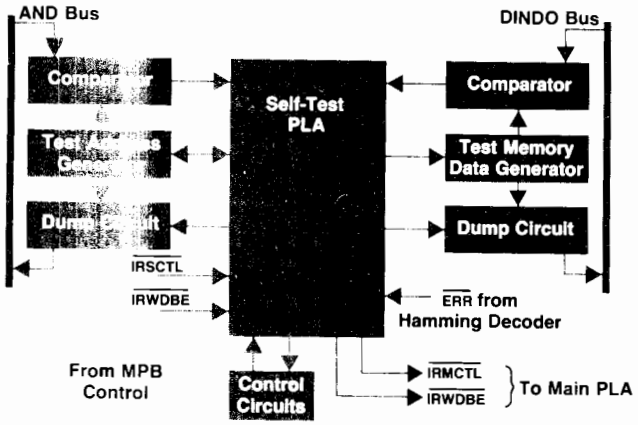


Fig. 7. *Block diagram of self-test system incorporated onto the memory controller chip.*

an addressing failure), the mapper CAM is loaded with a MAPOUT condition as a message to the operating system that the memory is not 100% good. The memory test takes less than 500 ms to complete.

Self-test then allows handshakes with the system turn-on procedure to test the MPB interface and its connection to the memory processor bus.

A block diagram of the self-test system in shown in Fig. 7. The core of this system is a 19-input, 68-output PLA with 272 terms. Many of its outputs are sent to two 32-bit PLA-like pattern generators. The patterns from the test address generator are used as addresses or data placed on the AND bus, or as data compared to the AND bus. The patterns from the test memory data generator are used as data placed on or compared to the DINDO bus.

Other terms control the counters and shift registers that generate patterns for the test-address or test-memory-data generators, and the control counters that sequence the PLA through each state of each test block.

The memory control chip self-test has no branching or subroutine capabilities. It is strictly a sequential machine. Thus, the main challenge in its design and implementation was to do the best test available while positioning the test blocks in a sequence that minimized terms in the PLA. This sometimes required inserting NOP test blocks, or repeating a test several times within a test block when once would have been enough.

To check the self-test implementation, a software simulator was built. Tied into the chip's software emulator, it helped check chip functionality. The emulator was also helpful in ironing out the complexities of healing on the fly in a pipelined system.

128K-Bit NMOS Dynamic RAM with Redundancy

by John K. Wheeler, John R. Spencer, Dale R. Beucler, and Charlie G. Kohlhardt

THE SEMICONDUCTOR random-access memory (RAM) chip is a basic building block of today's computer memory systems. Ideal memory chip characteristics in a high-performance multiprocessor system include fast cycle times (large bandwidth), high number of bits per chip (density), low cost, and low power dissipation. A VLSI NMOS RAM was designed and built by Hewlett-Packard to optimize the above characteristics for HP's new 32-bit computer system, the HP 9000.

The RAM chip, whose layout is shown in Fig. 1, is composed of a large dynamic memory array with supporting peripheral circuitry on the left and bottom sides. The number and complexity of the peripheral circuits are minimized by the use of a four-transistor memory cell. The performance and other characteristics of this RAM are listed in Table I.

The memory array contains 128K four-transistor cells organized to store 16K 8-bit words. Eight identical 16K-by-1-bit sections are arranged side by side. Each section has 256 rows and 64 columns. In addition, each section has eight redundant rows located in the upper half, and two redundant columns placed in the center.

The peripheral circuitry on the left side of Fig. 1 contains the X and Y address receivers and drivers, the row decoders and drivers, and the row redundancy circuitry. The X and Y addresses are multiplexed on eight address pads. Each address pad has an X address receiver, but only six of the eight address pads have Y address receivers. The X address is bused to the row decoders and row redundancy circuits via 16 true complement lines. The Y address is bused to the column decoders and column redundancy circuits via 12 true complement lines.

The peripheral circuitry for each section along the bottom of Fig. 1 contains a column decoder, an I/O multiplexer, a sense amplifier, an output driver, a write data receiver, and column redundancy. The wire-bond pads located along the bottom include eight data I/O pads, several column-redundancy testing and programming pads, and various power supply and clock pads. Timing and control circuits are found in the lower left corner.

Table I

128K-Bit RAM Performance and Characteristics

Technology:	NMOS, single-level polysilicon, 1.5-μm-wide lines, 1-μm spaces, two-layer metal
Operating Modes:	Read, read/write, standby
System Features:	Synchronous timing (18-MHz system clocks) Pipelined architecture Semaphore operations Multiplexed pads for chip select, data, and address
Organization:	16K×8
Memory Cell:	Four-transistor dynamic
Cell Size:	10.25 μm by 20.5 μm
Chip Size:	6690 μm by 7580 μm
Redundancy:	8 rows and 16 columns, using electrically programmed polysilicon links
Power Supplies:	−2V, 3.6V, 4.9V, 6.5V
I/O Levels:	Precharged bus scheme
Access Time:	165 ns
Cycle Time:	110 ns (includes read/write)
Power Dissipation:	450 mW, typical
Standby Power:	125 mW, typical (address receivers always active)
Refresh:	256 cycles, 1 ms
Package:	Finstrate. Copper-core printed circuit board with Teflon™ dielectric

Memory Cell

This dynamic RAM is somewhat novel in that it uses a four-transistor storage cell. Most MOS dynamic RAMs built today use a one-transistor storage cell, and most MOS static RAMs use either a six-transistor storage cell or a four-transistor storage cell with high-resistance polysilicon load resistors. The major characteristics of each design are summarized in Table II.

The desired speed and performance of the system prevented the use of the one-transistor storage cell. The high power dissipation of the six-transistor static cell was prohibitive and the four-transistor static cell could not be used because HP's VLSI NMOS process (NMOS III) does not incorporate high-resistance polysilicon.

The four-transistor dynamic cell shown in Fig. 2 provides high speed, low power dissipation, and a cell size smaller than most static cells, and is compatible with the NMOS-III process.

Functional Description

The RAM chip uses a nonoverlapping, two-phase system clock for synchronous operation. The clock period (one $\phi 1$ pulse and one $\phi 2$ pulse) is 55 ns. These pairs of $\phi 1$ and $\phi 2$ pulses are further organized as data cycles and address cycles. Synchronization of the RAM chip with the 32-bit computer system is done via the system-pop signal at power-up.

The RAM chip provides three modes of operation: read,

Table II

Storage Cell Characteristics

Type	Speed	Cell Size	Static Power
Four-Transistor Dynamic	Fast	Medium	None
One-Transistor Dynamic	Slow	Small	None
Six-Transistor Static	Fast	Large	High
Four-Transistor Static	Fast	Large	Low

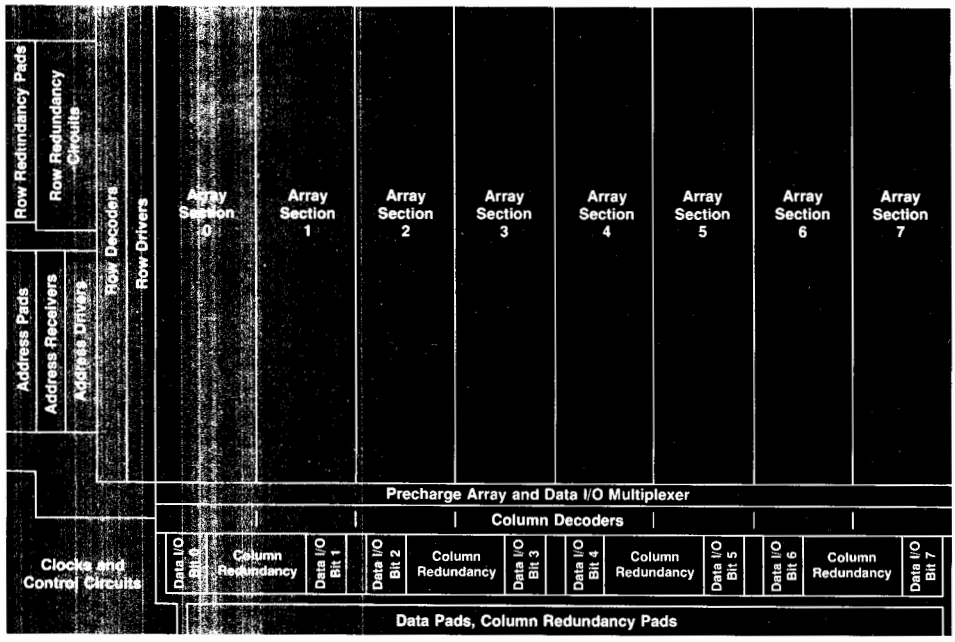


Fig. 1. *Physical layout of 128K-bit NMOS dynamic RAM.*

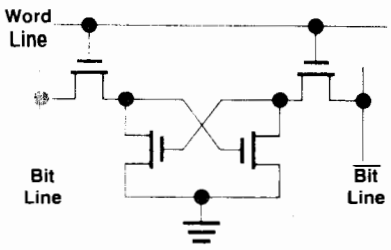


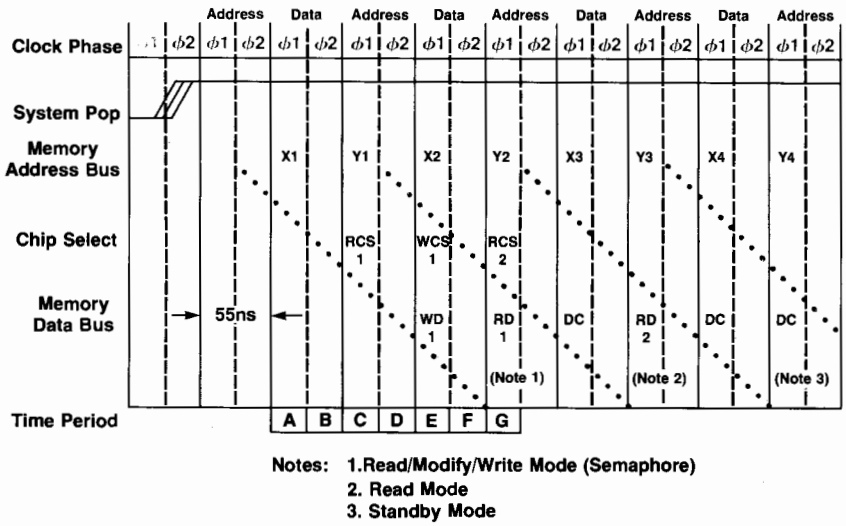
Fig. 2. *Four-transistor dynamic storage cell.*

in standby mode.

Time B. All pads are precharged.

Time C. The first Y address Y1 and read chip select signal RCS1 are received. RCS1 triggers the internal operation of the chip, and the chip goes from standby into active mode.

Time D. All pads are precharged. The decoded X1 address selects one of the 256 row lines to go high and all others remain low. All cells connected to this row line begin driving differential data on their respective precharged data/data' pairs.



Notes: **1.Read/Modify/Write Mode (Semaphore)**
2. Read Mode
3. Standby Mode

Fig. 3. *Timing overview for operation of 128K-bit NMOS dynamic RAM chip. (DC=Don't Care).*

read/write, and standby. See the memory system article on page 14 for a detailed discussion of address and data cycles.

A step-by-step description of timing periods A through G from the timing overview shown in Fig. 3 follows. Refer to the block diagram of the data path in Fig. 4 to follow the major internal events in the RAM.

Time A. The first X address X1 is received on the address pads and latched into the X address receivers. No other internal action takes place because, before this, the chip was

Time E. The write chip select and write data signals WCS1 and WD1 are received on the pads. The decoded Y1 address connects one of the 64 data/data' pairs in each section to the sense amplifier through the I/O multiplexer. Differential cell data is driven into the sense amplifier. The next X address X2 is received, initiating pipelined operation.

Time F. All pads are precharged. The sense enable clock signal SE isolates the sense amplifier from the I/O multiplexer. The sense amplifier completes the sense operation

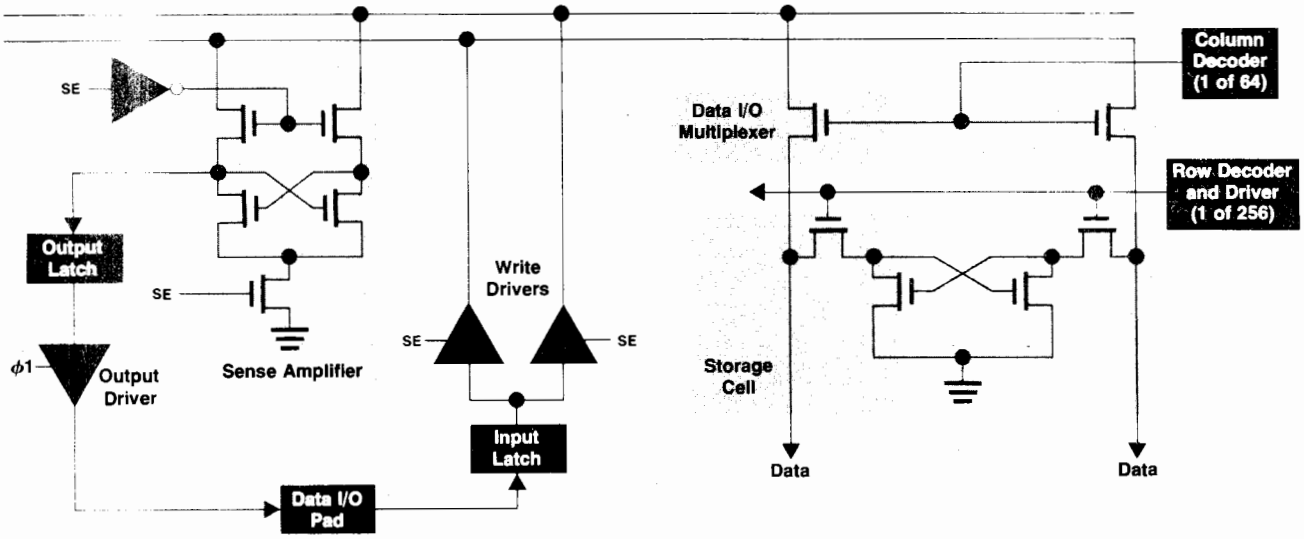


Fig. 4. *Block diagram of data path for 128K-bit NMOS dynamic RAM. Only one of the eight memory sections is shown.*

and sets up the output latch with the read data signal RD1. WD1 is now differentially written back through the I/O multiplexer to the currently addressed cell.

Time G. RD1 is driven from the output latch onto the data I/O pads. Pipelined operation continues with the reception of the next Y address Y2 and read chip select signal RCS2. This completes the read/write operation to the cell at address X1, Y1.

Six clock pulses occur from when the X1 address appears on the address pads to when RD1 is valid on the data pads. Thus the access time for this read is 165 ns. Note that the X2, Y2 address has been received and partially processed so that the read data signal RD2 will be valid four clock pulses after RD1, corresponding to a pipelined cycle time of 110 ns.

This timing scheme is different from that used for most commercially available dynamic RAMS, whose cycle time is longer than their access time.

Redundancy

One method of increasing RAM yields, and thus reducing chip cost, is the addition of redundant memory cells. These redundant cells are used to replace defective cells, thereby repairing some chips that would otherwise be rejected. By adding extra rows and columns to the RAM array, defects of various types and at various levels can be repaired.

To demonstrate the potential benefits of redundancy, a yield model was developed. Here, the good die (chips) per wafer are equal to

$$\left(\frac{\text{Chips}}{\text{Wafer}}\right) \times \left(\begin{array}{c}\text{Probability of Zero}\\ \text{Defects in the}\\ \text{Uncorrectable Area}\end{array}\right) \times \left(\begin{array}{c}\text{Probability That All}\\ \text{Defects in the Correctable}\\ \text{Area Can Be Repaired}\end{array}\right)$$

$$=\left(\frac{\text{Chips}}{\text{Wafer}}\right)\left(\exp(-\text{DUS})\right)\left[\sum_{k=0}^{R}\left(\frac{(\text{DC})^k}{k!}\right)\left(\exp(-\text{DC})\right)\text{P}^k\right]$$

where D=defect density, U=uncorrectable area, S=sensitivity of uncorrectable area to defects, R=amount of redundancy, C=correctable area, and P=probability that a given defect is correctable.

A Monte Carlo analysis was done, treating each parameter as a random variable with an assigned probability distribution. From this analysis the optimum numbers of redundant rows and columns for the 128K-bit RAM were determined to be eight rows and sixteen columns. In addition, the analysis indicated that a yield improvement greater than 4× could be achieved.

The four-transistor RAM architecture is well suited for redundancy. In the case of the 128K-bit RAM, 75% of the chip area is correctable. This correctable area includes not only the memory array, but also the I/O multiplexer, row drivers, and row and column decoders. Yield is limited by defects in the remaining chip area. However, because circuits along the periphery of this chip have a low percent-

Polysilicon Link Fusing and Detection Circuit

The redundant rows and columns on HP's 128K-bit NMOS dynamic RAM chip are programmed to replace defective rows or columns by fusing polysilicon links on the chip. Special circuitry is included on the chip to do this and to detect fused polysilicon links. This circuitry is illustrated in Fig. 1.

When fusing polysilicon links, a special power supply, V_{BLOW}, is connected to the fusing circuit, the link is addressed, and a voltage pulse is applied to the pulse pad. The resulting current through the link and FET Q3 fuses the link open. During normal operation, the pulse pad and V_{BLOW} are driven to ground by FETs Q1 and Q2 to disable the pulse circuitry.

To determine if a link is fused open or not, its resistance is compared to a polysilicon reference resistor. In the worst case, the link resistance must be only a factor of three different from the reference for reliable detection. The reference resistor is designed to be about five times the resistance of an unfused link, regardless of process variations. This design provides higher link fusing yield and greater reliability.

When power is first applied, POP (power on preset) becomes high, $\overline{POP}$ is low. The resulting voltage at node $\overline{F}_x$ is approximately equal to V_t (threshold voltage) if the link is intact, but is greater than V_t if the link is open. The currents through matched depletion FETs Q5 and Q6 depend strongly on the difference of resistances of the link and the reference resistor, and thus generate a corresponding voltage differential at nodes F_x and $\overline{F}_x$.

After system power up, POP goes low and $\overline{POP}$ goes high. The differential voltage between F_x and $\overline{F}_x$ is then amplified and the circuit latches. Complementary outputs are then present at nodes F_x and $\overline{F}_x$. Depletion capacitor C1 stabilizes the voltage at node 1 during the transition from POP to $\overline{POP}$ in case there is some deadtime or overlap between these signals. The resistances of the link and the reference become insignificant factors once the circuit latches.

-Douglas F. DeBoer

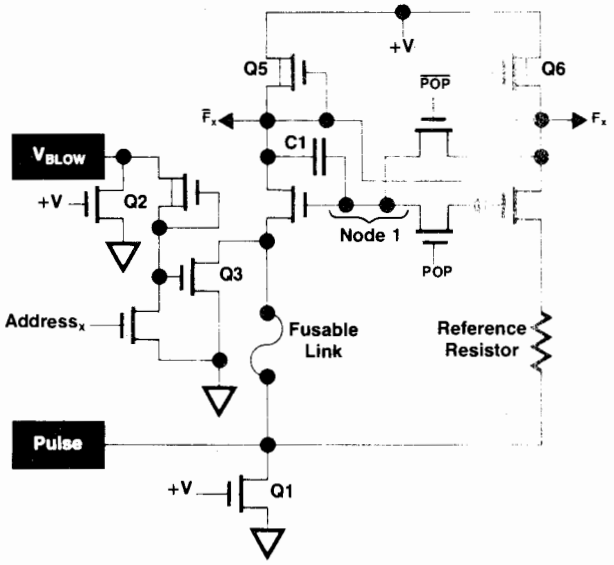


Fig. 1. *Link fusing and detection circuit used on the 128K-bit NMOS RAM chip.*

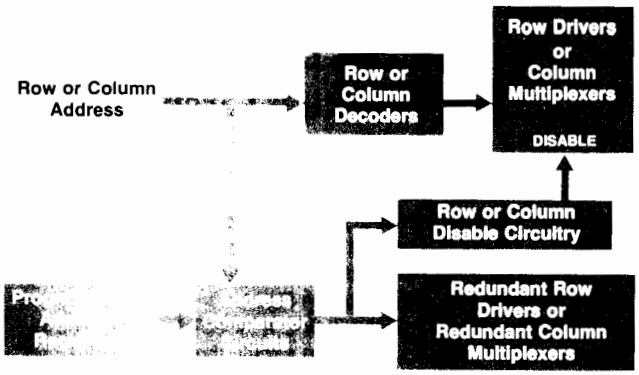


Fig. 5. *Block diagram of redundancy system used on the 128K-bit RAM chip.*

age of active device area and are relatively insensitive to leakage-type defects, this area is higher yielding. By using conservative design rules, the yield of this area is increased even further.

A block diagram of the redundancy system is shown in Fig. 5. The row and column redundancies are similar in design, but are separate circuits within the RAM. The operation is as follows. An incoming row or column address is compared to the addresses stored in the preprogrammed address registers. If a match occurs on any of the address comparators, the corresponding redundant row or column is enabled. In addition, the deselect circuitry is activated, which disables the nonredundant rows or columns. Redundant rows and columns are identical to other rows and columns. They share the same I/O data path, timing, storage cell pitch, and layout. The only exception is that the address decoders are replaced by address comparators for the redundant rows and columns.

The programmable address registers contain polysilicon links which are electrically programmed during wafer testing. During normal operation, the link resistance is compared to a reference polysilicon resistor through a comparator circuit (see box on page 23). This circuit provides both the true and complement outputs of the programmed address to the address comparator circuits.

Because of additional delays through the address comparators and disable circuitry, the disable signal becomes true after the normal address decoding is complete. So that redundancy does not degrade chip performance, the disable signal deactivates the final stages of normal row and column selection rather than disabling their decoders.

Finstrate: A New Concept in VLSI Packaging

Finstrate combines a copper fin for heat conduction and dissipation with a multilayer substrate for low-capacitance interconnection between ICs.

by Arun K. Malhotra, Glen E. Leinbach, Jeffery J. Straw, and Guy R. Wagner

EVEN THOUGH HP'S NMOS III technology has low power dissipation per gate, it also allows an IC designer to pack more than half a million transistors onto a single chip. The result is an average power density of 20 watts per square centimeter and power output up to 5 watts per chip. This degree of miniaturization also results in circuits with a large number of interconnection pads and high clock speeds. A 32-bit I/O processor chip, for example, has 122 pads and operates at 18 MHz.

Early in the design of the chip set for a 32-bit VLSI computer system, it became obvious that the speed, interconnect, and cooling requirements of the system could not be met by established packaging methods. An insulating material with a low dielectric constant is necessary to minimize line capacitance for high-speed operation. Fine line traces are needed for the dense interconnect pattern surrounding the ICs. The high power dissipation of the chips results in unacceptable junction temperatures without good heat dissipation.

The finstrate (fin-substrate) board was developed to meet these needs. It has a solid copper core, uses Teflon™ for the dielectric, and provides 0.125-mm-wide traces spaced 0.125 mm apart.

Fabrication

An array of finstrates begins as a single copper sheet. This sheet, roughly 1.0 mm thick, forms the heat dissipation path and electrical backgate connection at the center of each finstrate. Holes are drilled through this sheet to provide openings for insulated electrical connections between the outer layers of a completed finstrate. Following a surface treatment operation, a sheet of Teflon and a copper foil are laminated onto each side of the copper sheet. At this stage, the Teflon fills the drilled holes mentioned above as shown in Fig. 1a. The copper foil is converted to intermediate-layer circuits by a print-and-etch process, a technique borrowed from conventional printed circuit technology.

Teflon is a registered trademark of the DuPont Corporation.

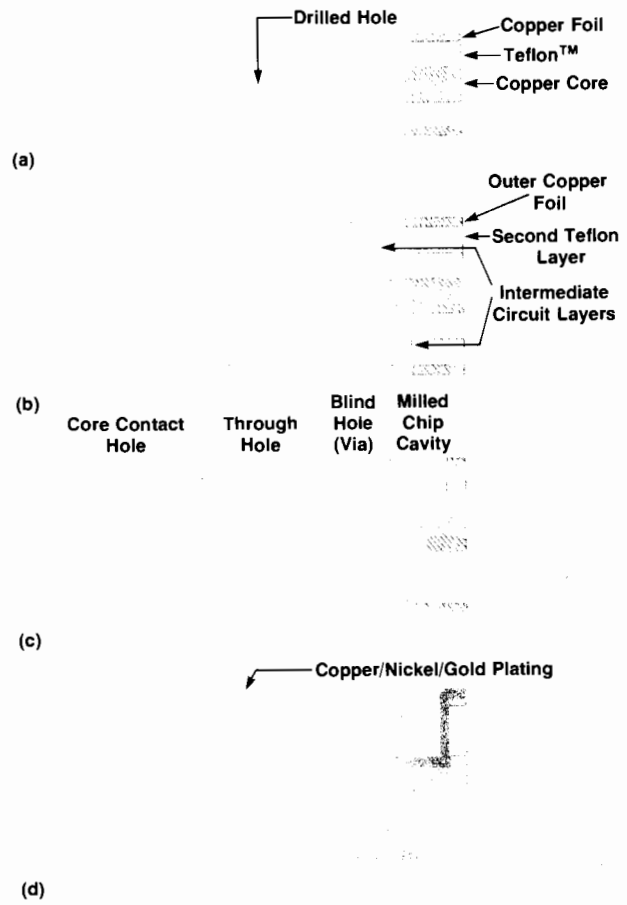


Fig. 1. *Cross sections during the finstrate fabrication sequence. (a) After lamination of the intermediate copper foil layer. (b) After definition of the intermediate layer circuit pattern and subsequent outer copper foil layer lamination. (c) After drilling and cavity milling steps. (d) Completed finstrate.*

The finstrate panels grow thicker (Fig. 1b) as a second layer of Teflon and another copper foil are laminated on each side. Interconnect features between the intermediate and outer copper foil circuit layers are defined next by a selective etch process. Blind holes (vias) connect the intermediate layer with the associated outer layer on each side of the copper core.

Next, cavities are milled through the laminated layers to the copper core at the locations where the ICs are to be attached. A second drilling operation also performed at this time serves two purposes. Relatively small bits drill through the center of the Teflon material filling the large holes through the copper core to form holes for plated-through connections. The other holes drilled in this operation contact the copper core to create outer-layer-to-core connections. Fig. 1c demonstrates these features. Following plating operations to build a conductive base coating over the entire panel surface, circuits are defined by a photoresist masking process which leaves the desired circuit pattern exposed. Electroplated copper, nickel, and gold increase the thickness of the exposed pattern. A further selective electroplating step leaves a high-purity gold layer on the edge connector fingers, wire bond pads, and chip cavities. All copper foil remaining between traces is etched away after the photoresist masks are stripped (the gold layer protects the desired circuit pattern). A blanking operation separates individual finstrates from the panel, and electrical and visual tests complete the finstrate fabrication. A completed finstrate, as shown in Fig. 1d, has a copper core, two copper-foil interconnect layers on each side of the core, Teflon as the dielectric, and gold-plated circuit patterns.

IC Assembly

Assembly of a large hybrid circuit (124×181 mm) with 22 integrated circuits, 92 passive components, and over 800 wire bonds is a challenge in itself. The refractory metallization used on the ICs and the finstrate's dielectric add even more constraints to the assembly process.

Finstrates are first mechanically scrubbed and rinsed in deionized water. This operation is essential for gold-wire bonding on finstrates. Chip capacitors are surface mounted using silver-filled epoxy. After a curing operation, a test is performed to check for epoxy bridging and to verify component values. Components are then coated with a nonconducting epoxy for protection from humidity.

The ICs for the finstrate are picked up with a vacuum collet and placed in the milled cavities which have undergone an epoxy stamping operation. The silver-filled epoxy, which is cured at 150°C, makes a good electrical and thermal connection between the finstrate and the IC. Special precautions are taken so that the IC's top surface is not touched when handling the chips. This minimizes mechanical damage and enhances the assembly yield.

The IC pads are electrically connected to the finstrate with 38-μm-diameter gold wires. Placing 4-mm-long wires on 0.16-mm centers using an automatic thermosonic wire bonder requires tight controls over the bonding process. Softening of the Teflon on the finstrates prevents the use of bonding temperatures greater than 100°C. The use of aluminum pads over silicon nitride on the IC and an extra ball bond over the tail bond to the finstrate gives the best results (see Fig. 2). After wire bonding, the ICs are coated with a polymer for alpha particle protection. Stainless-steel

Fig. 2. *Wire bonds from the IC to the finstrate use an extra ball bond over the tail bond as shown.*

Fig. 3. *The HP 9000 Computer uses HP's new VLSI and finstrate technologies to provide a desktop-size workstation low enough in price to let professional personnel have their own personal 32-bit mainframes.*

lids over the IC cavities are then attached to the finstrates with a nonconducting epoxy to provide mechanical protection for the ICs and the chip capacitors. After electrical tests and a burn-in cycle, the finstrates are completed.

Memory/Processor Module

Three types of finstrates (see Fig. 5 on page 6) are used in the HP 9000 Computer (Fig. 3). The CPU finstrate houses the CPU chip and a clock buffer chip. The IOP finstrate holds the I/O processor chip and a clock buffer chip and is connected to a printed circuit board containing TTL buffers. The 256K-byte memory finstrate contains twenty 128K-bit RAMs, a memory controller chip, and a clock buffer chip. All finstrates are housed in an enclosure called the Memory/Processor Module (see Fig. 4 on page 5). Finstrates in this module are located on 11.5-mm centers and are connected to a motherboard via edge connectors. A one-hundred-pin bus connects the finstrates together. In the center is the 32-bit memory processor bus (MPB) with interlaced ground traces and active termination provided by all inactive drivers. The system clock is routed to all finstrates simultaneously along traces of matched length. Self-test connections are included in the control signal portion of the bus, and power supplies use the remaining bus pins. A dc fan, chosen so that air velocity can be controlled as a function of ambient air temperature, is located on the end of the module to move cooling air across the finstrates.

Finstrate and Module Design

A considerable amount of time was spent on thermal analysis of the finstrates and the results were used to minimize the chip junction temperatures. A program that solves the steady-state Poisson heat conduction equation by using a finite-difference approximation was written for an HP 9845 Computer. Since air is used to cool the finstrate, a nodal network representing that moving fluid was added to the program. Fig. 4 shows a calculated result for the RAM finstrate. The program predicted a junction temperature of 81.5°C for the clock buffer chip, which was subsequently experimentally verified to be correct within ±2°C. By proper finstrate design it was found that under the worst-case operating conditions of 4572 meters altitude and 55°C ambient air temperature, no processor chip exceeded the maximum allowable junction temperature of 90°C.

For the clock speed and signal rise-time requirements of the HP 9000 processing system, special consideration of the electrical performance of packaged components was required. At the finstrate level, microstrip analysis of critical features was done. The choice of Teflon with its very low dielectric constant of 2.1 significantly reduces capacitive coupling when compared with other typical dielectric materials. This generally allows increased speed for a given output driver power level. Calculations also helped select appropriate trace shapes and sizes for the various interconnect requirements.

Acknowledgments

The authors would like to acknowledge the help from Walt Johnson and HP's Loveland Division printed circuit shop—without their support, this program would not have succeeded.

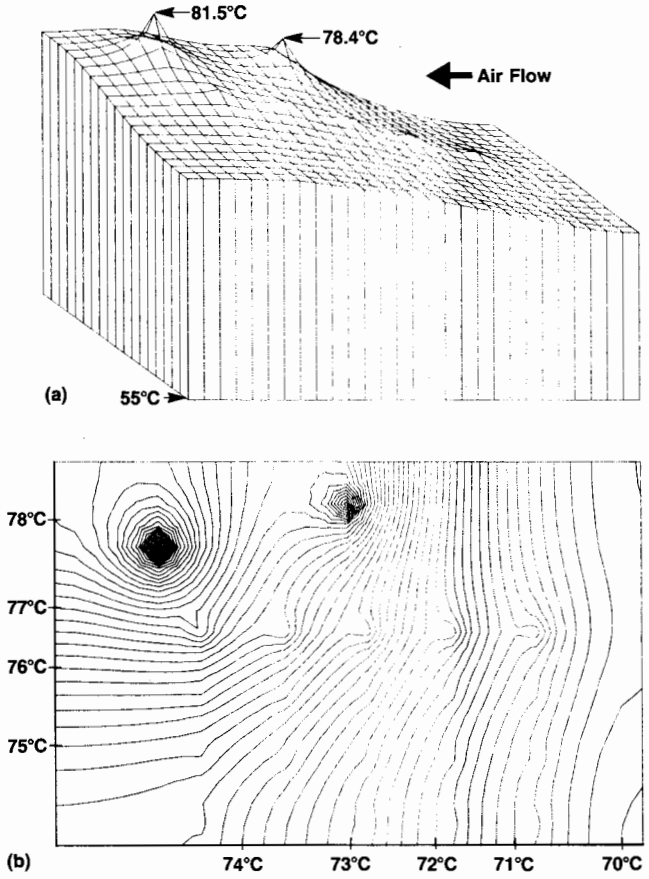


Fig. 4. *Three-dimensional (a) and surface contour (b) temperature plots calculated for the RAM finstrate at an altitude of 4572 meters and an ambient temperature of 55°C.*

NMOS-III Process Technology

by James M. Mikkelson, Fung-Sun Fei, Arun K. Malhotra, and S. Dana Seccombe

THE MAJOR TECHNOLOGICAL INNOVATION required for the design and manufacture of the 32-bit HP 9000 Computer System was the development of NMOS III, a high-density, high-speed IC process. This eight-mask, n-channel, silicon-gate process uses optical lithography to print minimum features of 1.5-μm-wide lines and 1.0-μm spaces on all critical levels. Both enhancement and depletion devices are available. The devices are fabricated with 40-nm-thick gate oxides and shallow implanted sources and drains to reduce short-channel effects. Major departures from conventional MOS processes include external contacts* to gates, drains, and sources, and two layers of refractory metallization for interconnecting devices.

Design Considerations

Significantly improved circuit performance can be obtained by reducing (scaling) the size of the geometrical features of an integrated circuit. Scaling provides increased speed and reduced power consumption for a given electrical function, and at the same time, it allows the fabrication of a greater number of circuits on a given silicon chip area. This increased packing density reduces cost and improves reliability of an electronic system.

But, to build a 32-bit VLSI computer system with the feature sizes used in NMOS III, simple scaling of conventional circuits or processes is not practical because several physical effects become significant circuit and fabrication limitations. Some important device limitations, such as electron velocity saturation, fringing capacitance, subthreshold current, substrate bias effects, and device variations caused by fabrication tolerances, must be properly modeled in the design rules and circuit simulations before a 32-bit VLSI chip can be designed successfully.

The importance of geometrical control is illustrated in Fig. 1. The variation of threshold voltage as a function of channel length is shown. The change in threshold voltage of a 1.5-μm-channel-length device caused by a photolithographic linewidth variation of ± 0.25 μm is 0.10V, which is about 10%. This change, in conjunction with a channel-length change of 1.4 to 1, causes an output current variation for the device of 1.6 to 1. These variations must be included in the worst-case design of circuits.

Circuit failure mechanisms caused by mobile ions, electron injection into the gate oxide, and metal electromigration or corrosion must be avoided. New process techniques are required to minimize operating margin loss and speed reduction caused by high-resistance interconnection between individual devices. In addition to addressing these problems, it was clear that to increase packing density significantly, new interconnection and contact techniques had to be developed.

*An external contact is a contact whose area is as large or larger than the device area to be contacted.

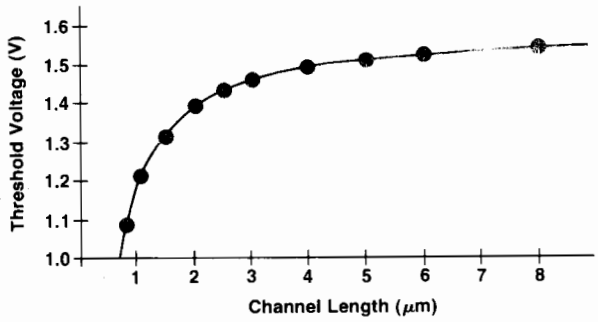


Fig. 1. *Effect of channel length on threshold voltage.*

The failure mechanisms are avoided by using clean processing techniques to prevent mobile ion contamination, minimizing electron trapping in the oxide, limiting the voltages applied to short-channel devices to prevent threshold shifts caused by electron injection, and using refractory metal to increase resistance to electromigration and corrosion.

Because two layers of refractory metal are provided, high-resistance polysilicon is not needed to interconnect devices over any significant distance. Besides eliminating the RC delays typically associated with high-resistance polysilicon interconnections, the two layers of metal help solve many of the topological problems associated with interconnecting a very large number of devices. The ability to run low-resistance interconnections in two directions

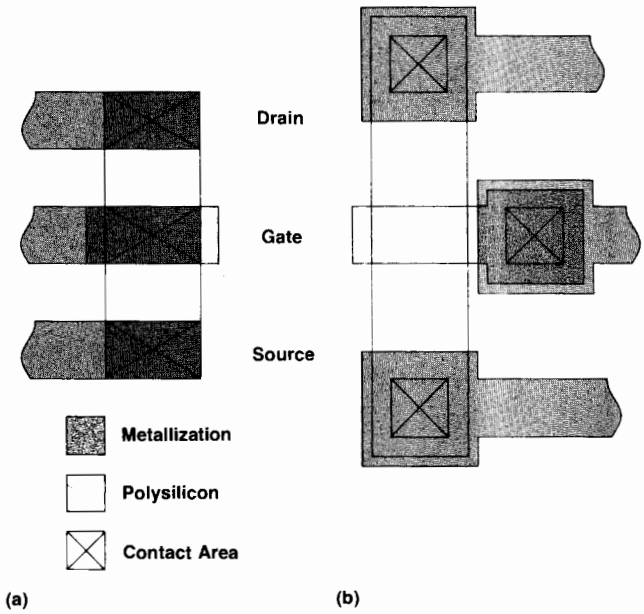


Fig. 2. *By using external contact structures (a), the area for a minimum device structure can be greatly reduced as compared with using internal contact structures (b).*

reduces the area of many circuits by a factor of two and significantly simplifies the design process.

In a typical MOS process, large areas are devoted to contacting the metallization to the gates, sources, and drains of the devices. Special processing techniques developed for NMOS III allow the use of external contacts as illustrated in Fig. 2. By allowing direct metal contact to the gate electrode over the gate oxide, and by allowing similar contacts to the source and drain, an area savings of up to 60% can be achieved.

Process Description

Fabrication begins with high-resistivity (20 Ω-cm) p-type substrates. After the growth of a 20-nm-thick thermal oxide buffer layer, a 160-nm-thick layer of silicon nitride (Si_3N_4) is deposited. The field oxide areas are patterned and the nitride and oxide layers are etched. The exposed silicon is anisotropically etched with potassium hydroxide and the field regions are implanted to provide a high parasitic threshold. A fully recessed field oxide is grown to a thickness of 600 nm using the nitride layer as a local oxidation mask.

The nitride and the oxide buffer layer are removed and all exposed silicon is implanted with boron to a depth of 0.3 μm and an average doping of $3\times10^{16}/cm^3$. The surface is masked and areas are opened for the depletion load implant. After these areas are implanted with phosphorus to a depth of 0.15 μm and an average doping density of

Polysilicon Link Design

The NMOS-III RAM required the development of an on-chip polysilicon link for the redundancy circuitry. Correctly designing this link required characterization of the physical fusing mechanisms, thermal properties, and electrical behavior of polysilicon. The resulting link can be electrically fused in a few microseconds with less than 200 mW. This link is shown in Fig. 1.

The electrical and thermal properties of polysilicon vary greatly with temperature as the link is heated to melting. For this reason, an electrical analog of the link's thermal characteristics was developed and simulated with the circuit analysis program HPSPICE. This model accurately predicts the voltage-current characteristics and thermal profiles for various links. Based on simulations, a link geometry was chosen that requires only low voltage and low power for fusing.

Simulation was also valuable in determining thermal profiles at the polysilicon-to-metal contacts found at each end of the link. It was discovered that the polysilicon could melt and cause the fuse gap to occur directly underneath the metal contact. This is highly undesirable from both a reliability and manufacturing yield standpoint. To control the temperature profile along the link, a contacting scheme was developed as shown in Fig. 2. At one end, the polysilicon makes contact with a diffused silicon region to create a heat sink to the substrate. This cools the region under the positive metal contact, forcing melting to occur only over field oxide and not near the contacting metal.

The physical mechanism of link fusing is by migration of ionized silicon atoms from the positive to the negative terminals of the link. This is why only the positive end of the link is connected to a diffusion for heat sinking. The fuse gap then occurs just beyond the diffusion contact. This mechanism was investigated and verified by cross-sectioning fused links and examining them with a scanning electron microscope.

The link's high reliability is attributed to the IC's top layer of insulating oxide softening and flowing into the fuse gap. The integrity of the IC's passivation layer is unaffected by fusing since the fusing power is so low. A guard ring was added during the design as an additional safeguard.

-William C. Terrell

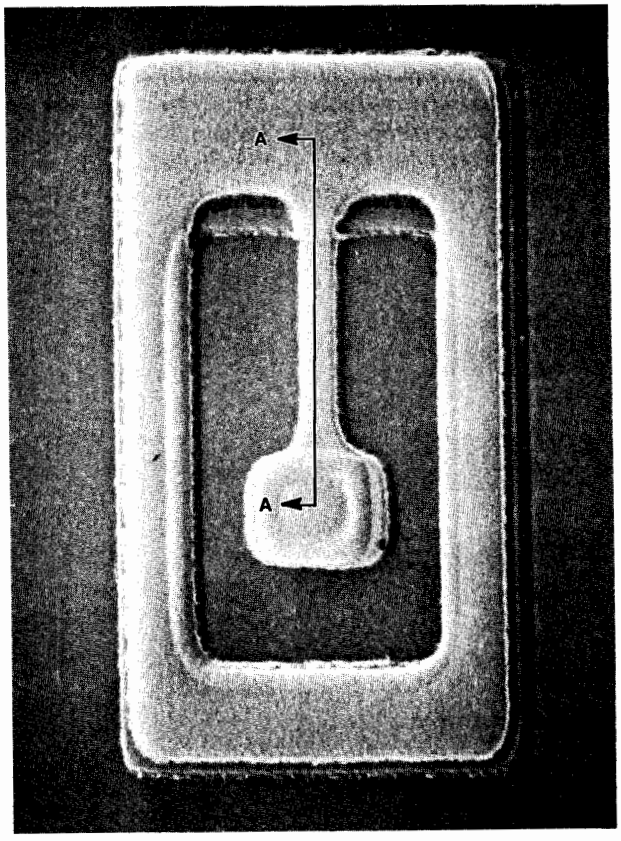


Fig. 1. *Microphotograph of a polysilicon link before deposition of CVD oxide and metallization. Cross-sectional view A-A is shown in Fig. 2 with CVD oxide and metallization added.*

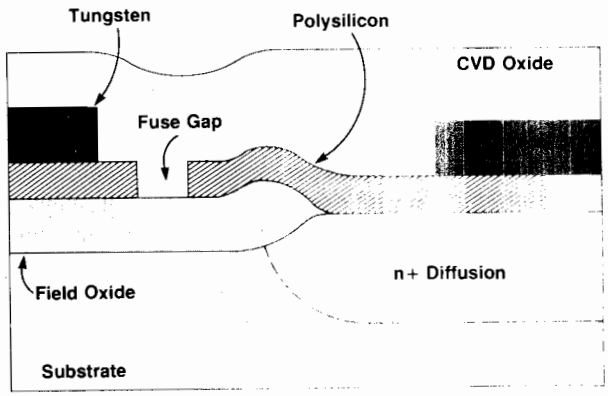


Fig. 2. *Cross section of polysilicon link structure showing metal contacts and fusing location.*

Automated Parameter Testing

With a process such as NMOS III that involves approximately 300 fabrication steps, it is essential that a performance measurement link be established between the process and the functionality of a chip such as the 32-bit CPU. This link should measure important parameters affecting the performance of a VLSI chip such as threshold voltage V_t, drain-to-source current I_{DS} versus drain-to-source voltage V_{DS}, and punchthrough voltage. The link should also indicate to the process engineers the steps to be controlled, including such parameters as oxide thicknesses, linewidths, and sheet resistances. In the early stages of process development, this link could be used as a process characterization and monitor tool. If the devices to be measured are included on wafers with the circuit chips (CPUs, RAMs, etc.), the link could also be used to screen wafers before functional testing. HP's System Technology Operation's implementation of this link is an automatic parameter tester coupled with a test section on each chip that enables 130 parameters to be tested in 2½ minutes.

The tester is composed of a variety of instruments, including four stimulus measurement units (SMUs) that are operated in a force-voltage/measure-current mode or force-current/measure-voltage mode, a high-voltage (±100V) power supply, a capacitance meter, an electrometer for low-current measurements, an HP 3455A Multimeter, and an HP 3437A High-Speed Voltmeter. All of these instruments are controlled by an HP 9845 Computer. The instruments are multiplexed through a 58-pin test head to an automatic prober with the wafer under test. Also included are an HP 7906 Disc Drive for temporary data storage and an RS-232-C/V.24-to-Factory-Data-Link interface for off-line data manipulation. The SMUs have a slew rate of 0.5V/μs at the end of any test pin. One reason for this remarkable performance is that all signal wires are guarded and driven by separate circuits throughout the test system. A block diagram is shown in Fig. 1.

In present production procedures, an operator loads a cassette of wafers onto the automatic prober, inputs the device type to the HP 9845, and the system takes over, automatically aligning, probing, testing, analyzing data, and then transmitting the results.

Many different device and parameter test patterns can be tested with this system, but one was specifically designed for the complete process/circuit monitoring mentioned above. It has evolved to include 14 gate-oxide FETs, including enhancement and depletion mode devices with various channel widths and lengths, two field-oxide FETs, devices for measuring lateral and vertical open-circuits and short-circuits at the polysilicon, first metal, and second metal layers, devices for measuring polysilicon and diffusion sheet resistances, and ten capacitors. The normal production flow involves sampling all test devices on five chips per wafer under various bias conditions and getting a real-time thermal printout of the results.

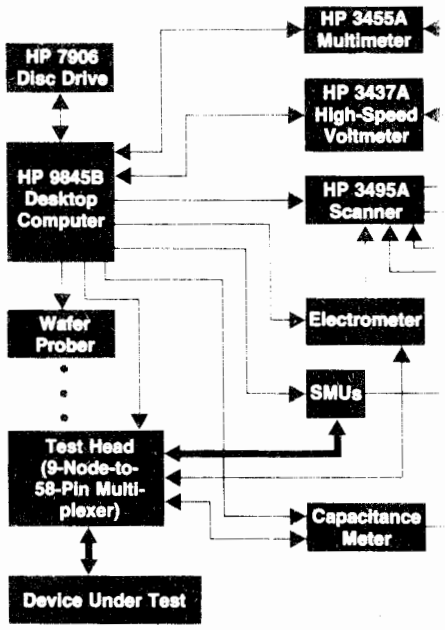


Fig. 1. *Block diagram of the automated wafer parameter test system used for the NMOS-III process.*

Options include making wafer maps of particular parameters and I_{DS} versus V_{DS} plots for specific FETs. This data is used to sort the wafers for functional testing. A one-to-two-page summary of all the data is also generated.

Acknowledgments

The test system was partially designed and constructed by Tracy Ireland, and the graphics for the summary were programmed by Richard Bettger.

-Fredrick P. LaMaster
-O. Douglas Fogg

$1.5\times10^{17}/cm^3$, the gate oxide is grown to a thickness of 40 nm. A cross section of the device structure at this point is shown in Fig. 3a on page 30.

The gate oxide is patterned and etched to expose areas for diffusions. LPCVD (low-pressure, chemical-vapor deposition) polysilicon is deposited and doped with phosphorus by a phosphine diffusion. During polysilicon doping, the diffused regions, which are 0.6 μm deep, are also generated by the diffusion of phosphorus through the polysilicon. The structure after polysilicon doping is shown in Fig. 3b.

A layer of Si_3N_4 is deposited on the polysilicon for use as an oxidation mask at a later step. But first, the nitride is patterned and etched for use as an etch mask for the polysilicon. The polysilicon is etched and the source and drain regions are implanted through the overlapping gate oxide with phosphorus to a depth of 0.3 μm. Buried contacts are formed at the same time as the polysilicon pattern. Fig. 3c shows the structure after source-drain implantation.

Next, the edges of the polysilicon features and the areas over implanted and diffused regions are oxidized using the Si_3N_4 layer as an oxidation mask to protect the polysilicon features. The nitride is removed and a layer of phosphorus-doped silicon dioxide is deposited by chemical vapor deposition. Self-aligned contact holes with minimum areas of 1.5 μm by 1.5 μm are defined and wet etched through the deposited oxide, making use of the different etch rates for phosphorus-doped oxide and thermal oxide. The first layer of metal (400-nm-thick tungsten)

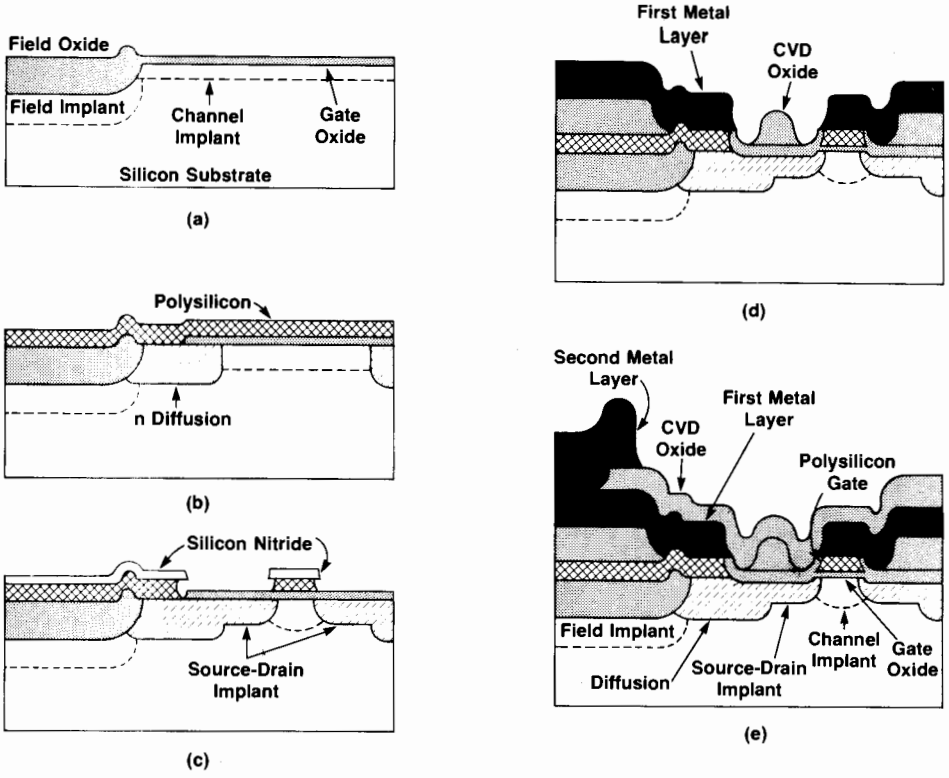


Fig. 3. *Cross sections of the device structure during various steps of the NMOS-III process. (a) After gate oxidation. (b) After polysilicon deposition and doping. (c) After polysilicon patterning and source-drain implantation. (d) After first-layer metal patterning. (e) Completed device structure.*

is deposited and patterned, leading to the structure shown in Fig. 3d.

Another layer of oxide is deposited to act as the insulator between the two layers of metal. After the definition and etching of this oxide layer to form contact holes between the metal layers, the 1.8-μm-thick second layer of metal (LPCVD tungsten) is deposited, patterned, and etched with typically 5-μm-wide lines and 3-μm spaces. The completed device structure is shown in Fig. 3e.

Two-Layer Refractory Metal IC Process

by James P. Roland, Norman E. Hendrickson, Daniel D. Kessler, Donald E. Novy Jr., and David W. Quint

THE ABILITY TO FABRICATE 500,000 devices on a single integrated circuit chip presents severe topological puzzles in interconnecting them. This task is further complicated by speed considerations that prohibit the use of a relatively high-resistance polysilicon layer for connections over any significant distance. Thus two layers of low-resistance interconnect are necessary for the practical design and operation of circuits using the NMOS-III technology. These two metal layers are constrained by the design rules shown in Table I.

The heavy emphasis on reducing device dimensions (scaling) affects not only the width of the metal lines, but also the material chosen and the processing used. Even though the total current through a minimum-dimension metal interconnect line is small in absolute terms, the current density in these lines is on the order of one million amperes per square centimeter because of their small cross-sectional area. This high current density can lead to electromigration failure.[1]

Because of its low resistivity and easy processing, aluminum is the most commonly used metal for integrated circuits. However, using typical values for current density and the elevated operating temperature (up to 90°C) of NMOS-III chips, electromigration calculations for aluminum predict a mean-time-before-failure on the order of weeks. The modeling of electromigration in tungsten is incomplete, but initial tests place the electromigration resistance of tungsten at about 1000 times that of copper-

Table I
NMOS-III Metal Interconnect Design Rules

Oxide	450-nm-thick silicon dioxide 1.5-μm×1.5-μm minimum contact area, zero overlap to polysilicon, zero overlap of first metal layer
First Metal Layer	1.5-μm-wide line/1.0-μm space 0.4 ohm/square sheet resistance
Intermediate Oxide	550-nm-thick silicon dioxide 1.5-μm×2.0-μm minimum contact area, zero overlap to first metal layer 2.0-μm overlap of second metal layer to via.
Second Metal Layer	5.0-μm-wide line/3.0-μm space 0.04 ohm/square sheet resistance
Lifetime	Median lifetime ≥10^4 hours at 85°C

doped aluminum.[1] For aluminum to function reliably in the NMOS-III process, its cross-sectional dimensions would have to exceed the high-density design rule specifications given in Table I.

As an example of this effect, Fig. 1 shows a large, copper-doped aluminum line connected to a small tungsten line after both were subjected to a high current density at an elevated temperature. The aluminum line is seen to have developed voids and hillocks, and exhibits some signs of melting. The tungsten line is unchanged. Because of electromigration considerations, as well as its etchability and chemical resistance, tungsten was chosen as the interconnect metal for NMOS III.

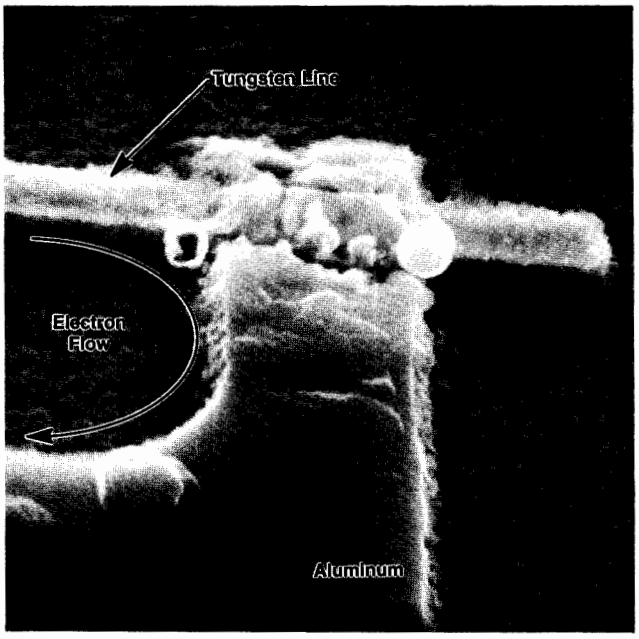


Fig. 1. *Microphotograph showing electromigration occurring in a wide aluminum line connected to an unaffected narrow tungsten line carrying the same current.*

Process Description

The choice of tungsten for integrated circuit interconnections is a major deviation from established practice and known mature technology. Furthermore, the dimensions and tolerances required by the NMOS-III process prevent the use of wet etching and demand that dry etching be used.[2] In addition, it has been repeatedly demonstrated that worst-case situations leading to yield loss coincide with surface variations of some sort. Therefore, care had to be taken in selecting the metal interconnect process sequence to control the shape of most features, avoid overhangs, and keep the surface as planar as possible.

The oxide below the first metal layer is deposited by an atmospheric CVD (chemical vapor deposition) process and doped with phosphorus for gettering mobile ions and to allow reflow.* The second oxide layer between the two metal layers is applied in a similar fashion, but is not reflowed. Oxide removal is accomplished in a plasma etcher designed to have a high level of vertical ion bombardment, which allows high and uniform etch rates. Results of etching the second oxide layer are shown in Fig. 2.

The deposition of tungsten can be accomplished either by a sputtering or a chemical vapor deposition process. Stress and conductivity in the deposited films are important parameters for a successful process. Because the first metal layer makes direct contact with polycrystalline silicon, and since the vias (vertical connections between interconnect layers) are of the zero-overlap type, silicon material is exposed to first-layer metal etch conditions at all vias. Under all plasma conditions tested, silicon etches considerably faster than tungsten. Therefore, the first metal layer is deposited with a 30-nm-thick etch-stop material under the 400-nm-thick layer of tungsten. The second metal layer is

*The process in which the oxide is heated to a temperature where it begins to soften and thus flows slightly to cover the underlying surface topography more evenly.

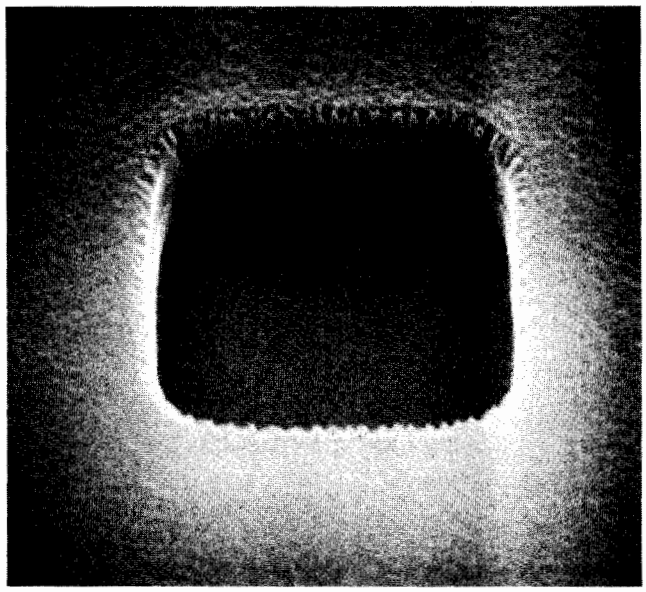


Fig. 2. *Microphotograph of a plasma-etched via between the first and second metal layers.*

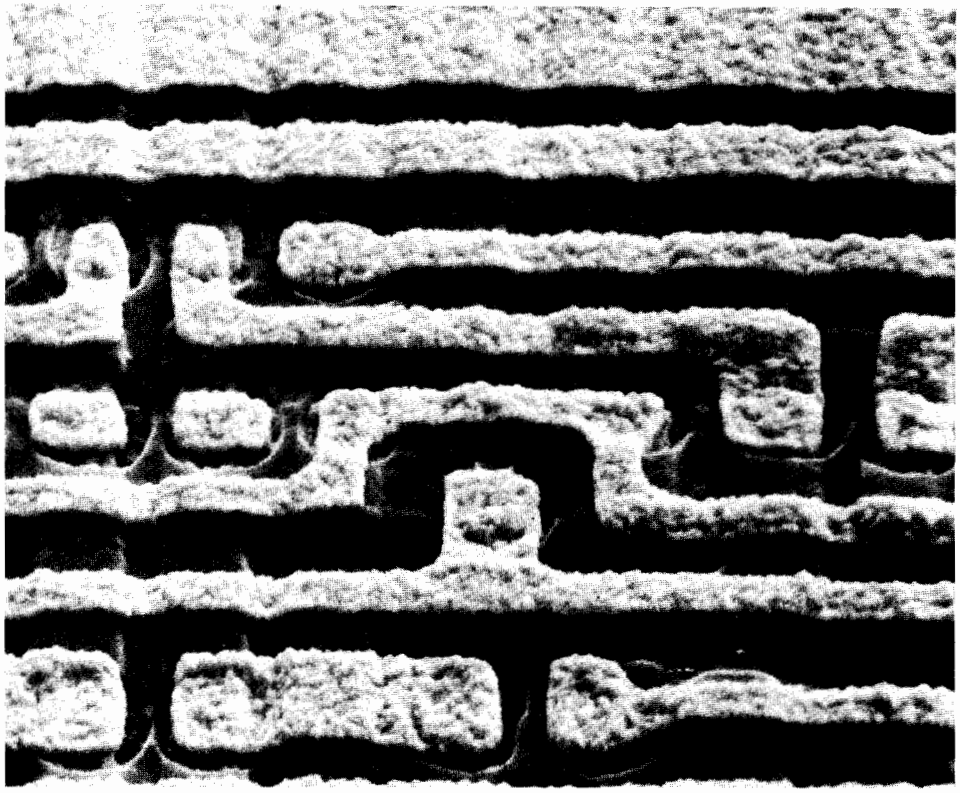


Fig. 3. *Microphotograph of NMOS-III first metal layer showing contacts to the underlying polysilicon layer (vertical lines in figure).*

1.8-μm-thick tungsten deposited by a low-pressure CVD process.

Etching of the tungsten layers is done in a parallel-plate plasma etcher using a fluorine-based gas composition. Worst-case conditions for metal shorts and opens were determined, and margins for underetching and overetching were established with regard to these limits. For instance, overetching failures in the first metal layer are often caused by a notch that occurs over certain features. Hence, the margin for overetching the first metal layer is determined by monitoring the area over this notch. A large-area test mask, a defect-density test mask, and run wafer data and analysis were used to define worst-case situations. Scanning electron microscope photographs of typical completed NMOS-III metal interconnects are shown in Fig. 3 and Fig. 4.

References

1. P.P. Merchant, "Electromigration: An Overview," Hewlett-Packard Journal, Vol. 33, no. 8, August 1982.
2. P.J. Marcoux, "Dry Etching: An Overview," Hewlett-Packard Journal, Vol. 33, no. 8, August 1982.

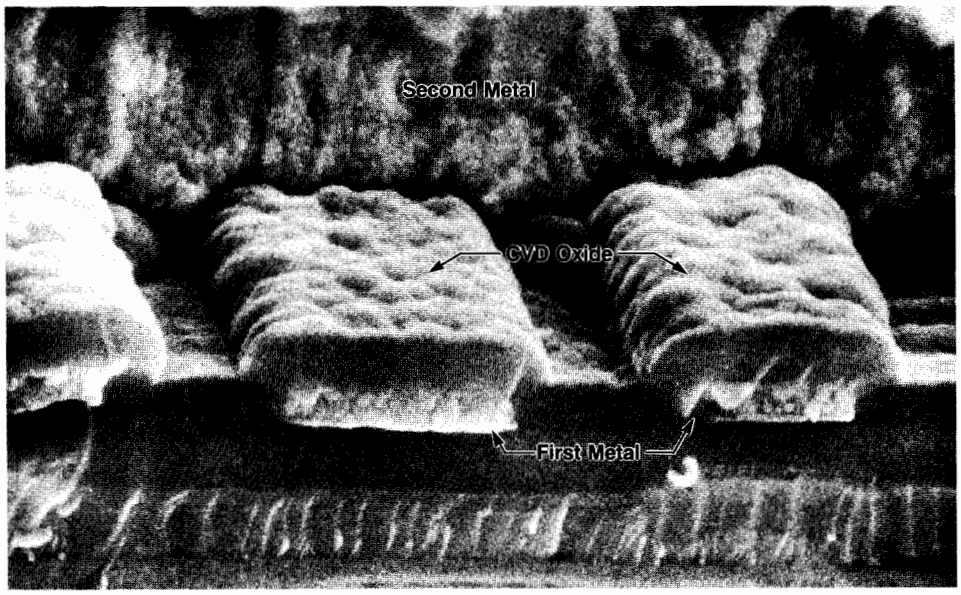


Fig. 4. *Microphotograph showing coverage of second metal layer over the first metal layer for the NMOS-III process.*

Defect Control for Yield Improvement

Defect control is a comprehensive term at HP's Systems Technology Operation (STO). It means reducing defects introduced by faulty etching, particles, or other process-induced problems, The defect control plan begins with understanding the failure mechanisms on real VLSI chips, (e.g., the 128K-bit RAM). Production runs are analyzed by subdividing each run into dead wafers, wafers with zones of defects, and random defects. Each category is extensively studied to determine the exact physical failure mechanism. Once the major process problems are found, the process engineers develop an improved process and usually establish a monitor for future control of this variable.

One typical example of comprehensive defect control relates to particles in the metal deposition system. The problems were traced to grinding chain mechanisms and a "big wind" effect during venting of the vacuum chamber to atmospheric pressure. Fig. 1 shows one of the many statistical control charts before and after the machine was improved.

Another example is the control of crystal defects. A densely packed 128K-bit RAM chip requires careful processing to avoid refresh problems. Refresh errors were found on the early RAMs and the problem was found to be junction leakage. Further investigations by STO and HP Laboratories personnel showed that the leakage was related to oxygen precipitation. Fig. 2 shows a RAM that was angle lapped, and Wright-etch decorated. The defects delineated are oxygen precipitates. Working with silicon vendors, the team solved the problem.

Acknowledgments

Thanks to Jim Barnes, Fred LaMaster, Rick Luebs, Tony Gaddis, Rajendra Kumar, Zemen Lebne-Dengel, Virgil Hebert and Pat DeBow for their ongoing contributions to these projects over the past few years. Also, thanks to HP Laboratories personnel, in particular Martin Scott and Julio Aranovich.

-Lawrence A. Hall

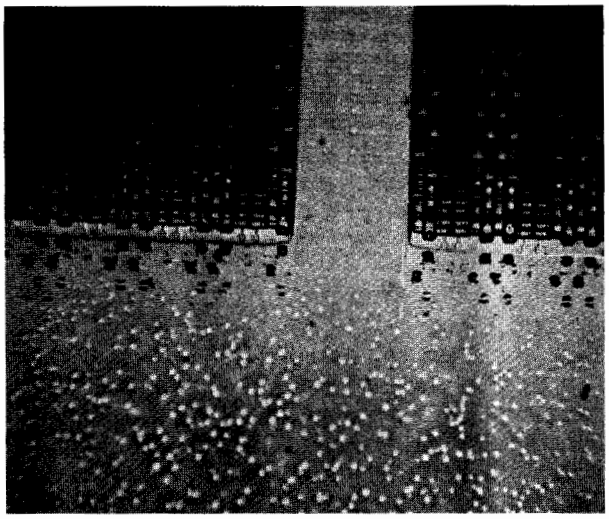


Fig. 2. *Microphotograph of angle-lapped and Wright-etched cross section of an early 128K-bit RAM wafer showing heavy concentration of oxygen precipitates that contributed to excess junction leakage.*

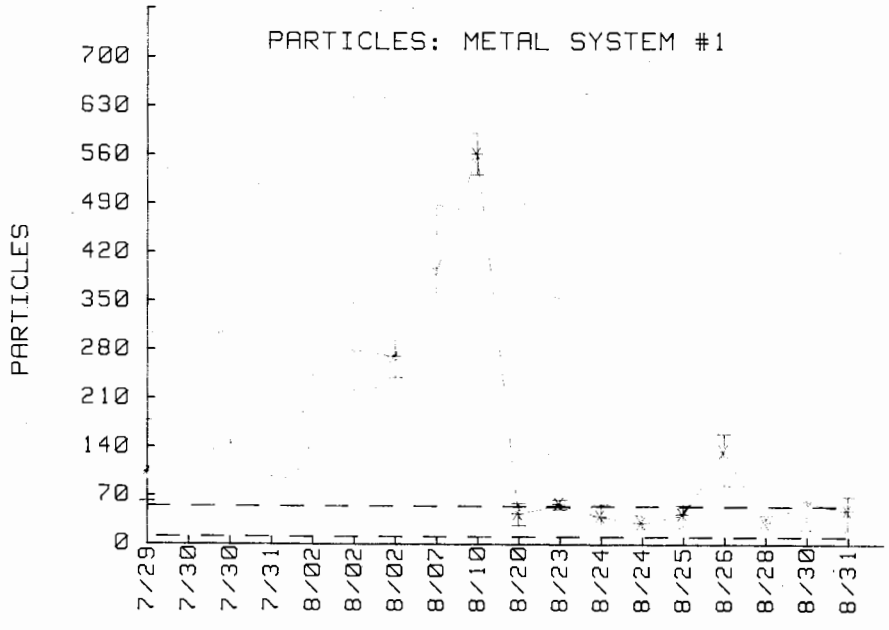


Fig. 1. *Control chart showing the reduction in particle count over a period of time as a metal deposition process was improved.*

NMOS-III Photolithography

by Howard E. Abraham, Keith G. Bartlett, Gary L. Hillis, Mark Stolz, and Martin S. Wilson

FROM EARLY FEASIBILITY STUDIES it was clear that the NMOS-III process would require revolutionary photolithography methods to produce chips in large volume. At that time, contemporary production processes achieved a minimum feature size of around 4 μm and level-to-level alignment within 0.75 μm. The corresponding feature size for the proposed NMOS-III process was to be 1 μm with ±0.25 μm alignment accuracy.

Early work using an optical aligner demonstrated that optical lithography could meet the requirements. At about the same time, the first step-and-repeat aligner with mass production capability was introduced to the marketplace. It was decided to use this machine for NMOS-III production. Initial development work was done using conventional photoresist processes, but it later became clear that the standard process lacked the necessary control for some levels because of exposure interactions with the substrate. A new multilayer photoresist process was developed to eliminate this problem.

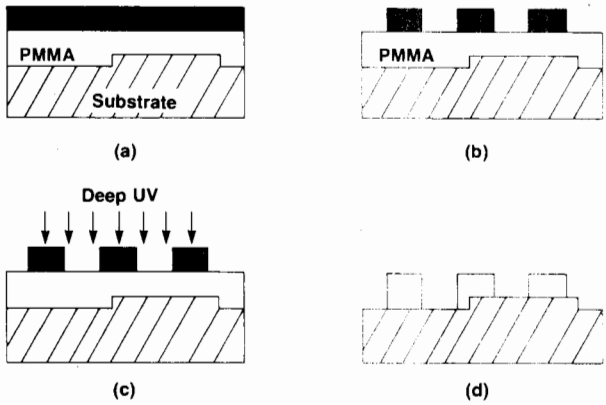


Fig. 2. *Two-layer resist process. (a) After application of PMMA bottom layer and positive-resist top layer. (b) After exposure and development of the top layer. (c) Flood exposure of bottom layer by deep-UV light. (d) After development of PMMA layer.*

Exposure System

The step-and-repeat optical aligner, shown schematically in Fig. 1, is the heart of the NMOS-III photolithography process. The light source consists of a HgXe bulb and an optical system for collecting, collimating, and filtering its output. Only radiation with a wavelength at the mercury G-line (436 nm) is used. A sensor measures the light output and provides information to a feedback system that adjusts exposure time to compensate for bulb aging and other variations. When the shutter is open, the light passes through the reticle, a glass plate with a 10× chrome circuit pattern. The reticle is precisely aligned to the optical column by using a dedicated microscope. The high-resolution reduction lens (numerical aperture = 0.28) projects a reduced image of the reticle pattern onto the photoresist-coated wafer. The wafer undergoing exposure is prealigned off-axis by the automatic alignment system and then moved into position under the lens. The entire wafer is exposed serially, in most cases one die at a time, under the control of the system computer. Positional feedback is derived from a laser interferometer system. Because the reduction lens has a shallow depth of focus (±2 μm), each image must be individually focused by the automatic focus system which uses reflected infrared light. Once the mask stepping is properly set up, the entire sequence proceeds automatically, cassette-to-cassette, without operator intervention. Because the system requires very uniform temperature for precision alignment and image accuracy, the aligner is enclosed in its own environmental chamber where temperature is controlled to ±0.1°C.

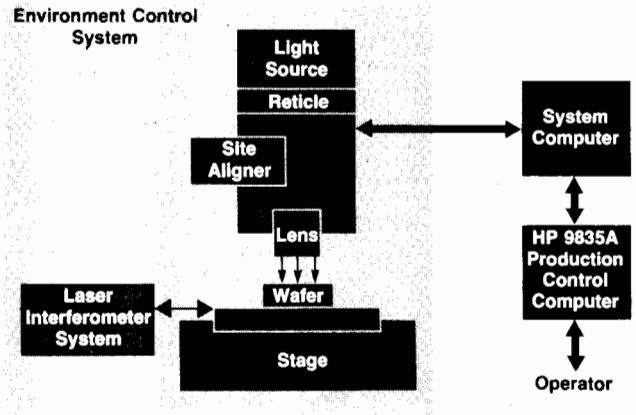


Fig. 1. *Block diagram of step-and-align optical wafer exposure system.*

Production control of the system is accomplished through an HP 9835 Desktop Computer. The 9835 is interfaced to the system's computer, which contains the master operating program. The production control program consists of approximately 4000 lines of BASIC code which provide a variety of data collecting and control functions. Among the more important are:

- The operator oversees the aligner's operation via the 9835 Computer. The system provides prompts and instructions to minimize operating complexity.
- Data files are maintained for each circuit and mask level. This data tells the system what stepping pattern to follow and what alignment offsets to apply.
- The system gathers and retains data for production control. For example, setup and run times, run identification, and focus and exposure settings are automatically recorded. Prealignment and alignment performance data is retained for engineering analysis.
- Using X and Y alignment data for each run level, the system automatically corrects wafer rotational error by

applying appropriate offsets to the stepping pattern. The system monitors optical column Z-axis movement to provide warning of potential poor focus caused by inadequate wafer flatness. This problem can arise if a particle becomes lodged under part of the wafer.

The wafer alignment system is capable of aligning each die individually. However, to improve throughput, the wafers are aligned globally and the pattern is stepped without aligning every die. Alignment performance with this approach is $\pm 0.20\ \mu m$ (2σ). The site alignment system uses a laser beam to illuminate a Fresnel-zone target on the wafer. The incident light is focused by the target and imaged through an optical system where detectors are used to determine the target's location. Position information is fed back to the controller which, with the aid of the interferometer system, accomplishes the final alignment.

The reticles used in the system are produced by electron beam lithography. Only electron beam generation can meet the reticle linewidth and runout control requirements. Also, the speed of electron beam generation is necessary because of the tremendous complexity of the chips used in HP's new 32-bit VLSI computer system. In some cases the pattern is constructed of over 3 million rectangles. The reticles are protected from contamination dust by pellicles as described in the box on page 36.

Photoresist Process

The mask alignment system selectively exposes a thin spun-on layer of photosensitive material (positive photoresist). Areas exposed to light are rendered soluble in a developer solution. After the photoresist pattern is developed, it becomes a mask for subsequent etching or ion implantation processes. It is extremely important that photoresist linewidths be well controlled, within $\pm 0.1\ \mu m$ for some NMOS-III mask levels. In the beginning, a severe problem in linewidth control was encountered as a result of light energy reflected from the underlying wafer surface. Since the exposing light is monochromatic, standing wave patterns exist in the resist because of interference between incoming and reflected wave fronts. Because of the standing wave, the amount of light energy coupled into the resist (i.e., the exposure dose) is a strong function of film thicknesses and substrate reflectivity. It was common to find that photoresist lines passing over a step on the wafer surface would be too wide on one side of the step and too narrow on the other. To solve this problem, a two-layer resist process was developed.

Fig. 2 shows the process steps. Two photoresist materials with very different properties are used. The bottom layer is PMMA (polymethyl methacrylate). This layer planarizes the wafer surface topography so that the top layer is uniform in thickness. The top layer is a standard positive photoresist which is sensitive to 436-nm-wavelength light (near-ultraviolet). The bottom layer is not sensitive to the imaging light and serves as a carrier for a dye which acts to absorb the imaging light. The dye was selected for its strong absorption at 436 nm as illustrated in Fig. 3. This characteristic, along with the relatively long path for light to be reflected by the wafer surface back to the top layer (two times the PMMA thickness), means that only a small amount (3%-by-weight) of dye must be added to the PMMA to decouple the imaging exposure from the wafer surface.

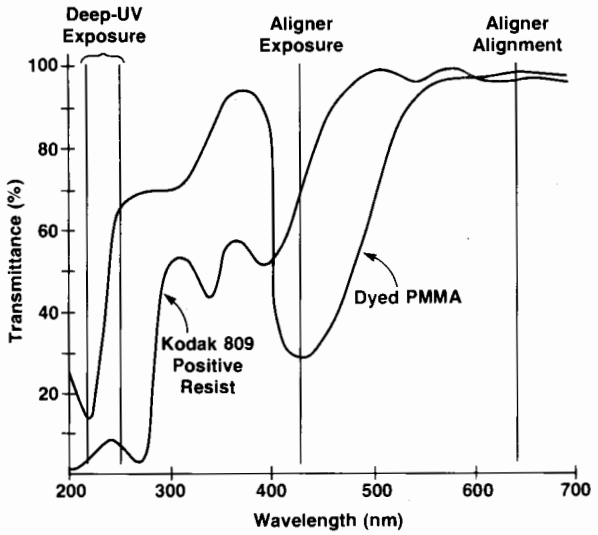


Fig. 3. *Light transmission characteristics of dyed PMMA and Kodak 809 photoresist as a function of light wavelength.*

After the top layer is exposed, it is developed. The patterned top layer then forms a mask for exposing the bottom layer. The wafer is blanket exposed with deep-ultraviolet light (wavelengths<250 nm), which induces rupture of the molecular chains in the PMMA layer, rendering the exposed area soluble. The top resist layer is opaque to these short wavelengths and therefore serves as an effective mask. The dye in the PMMA layer bleaches and does not strongly absorb the deep-UV light. Therefore, the thick layers of PMMA can be completely exposed in depth. Fig. 3 also shows the absorption and sensitivity characteristics of the positive photoresist used. PMMA is sensitive only to radiation wavelengths shorter than 250 nm. It has extremely good contrast, but low sensitivity. This implies excellent linewidth control, but long exposure time.

After the deep-UV exposure, the top layer of resist is removed. MX-931 developer is used to remove the layer of

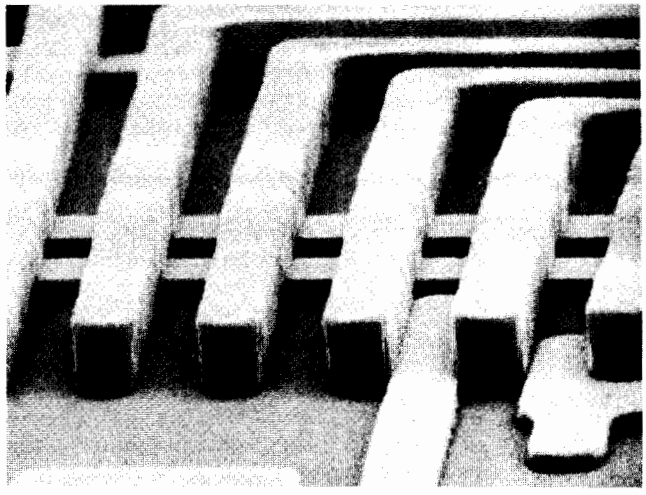


Fig. 4. *Microphotograph of exposed and developed PMMA pattern.*

intermixed material that forms where the two resist materials contact each other. The thickness of this interlayer is minimized by the use of Kodak 809 photoresist rather than other alternative resists which crosslink more readily. After interlayer removal, the PMMA layer is developed in MIBK (methyl isobutyl ketone).

Fig. 4 shows the typical vertical sidewalls of the PMMA lines produced by the process. Note the uniform linewidth independent of underlying topography. Linewidth uniformity and control are also evident in Fig. 5, which shows FET breakdown voltage (BV_{DSS}) measurements before and after the two-layer resist process was used to define the gate. The conventional process was characterized by lack of linewidth control, which resulted in extremely variable BV_{DSS} and other FET parameters.

A major emphasis in the two-layer process development has been manufacturability. The process involves no complex film deposition or etching. Application of both resist levels is accomplished cassette-to-cassette in an in-line coat/bake system. No operator intervention is required. Deep-UV exposure is done with a source that uses an RF-excited electrodeless Hg bulb and quartz optics to achieve wafer exposure about twenty times faster than the deep-UV sources available during the early development phase of the project. The deep-UV sources are mounted on cassette-to-cassette wafer handling systems to minimize the need for operator intervention.

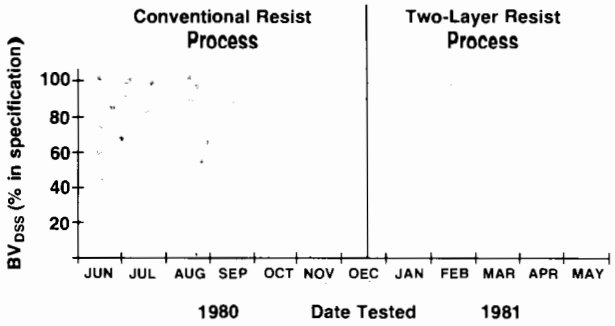


Fig. 5. *Comparison of the variation in breakdown voltage BV_{DSS} using a conventional resist process and the two-layer resist process.*

Emphasis was also placed on development of etches compatible with PMMA. PMMA has poor plasma etch resistance compared with many other resists and also has marginal substrate adhesion, which is detrimental for wet etching. By controlling plasma etching conditions, especially wafer temperature, successful etches have been developed

Yield Improvement by Use of Pellicles

Step-and-repeat photolithography has high yield potential because the reticle can be made perfect. However, the reticle must remain free of contamination in production or a serious defect may be replicated on every die printed with that reticle. To minimize or eliminate contamination, HP uses pellicles on all reticles used in the NMOS-III process.

Fig. 1 shows a pellicle mounted on an NMOS-III reticle. It consists of a thin nitrocellulose membrane stretched on a metal frame. The frame is bonded to the reticle to encapsulate the chrome pattern. One pellicle is used on each side of the reticle. The volume between frame, membrane, and reticle is cleaned of particles during assembly. In use, particles that fall on the membrane are out of focus and thus are not imaged unless they are large. Any large particles are easily detected and removed by the aligner's operator using a very simple in-situ detection system. The aligner's exposure source is turned on and the operator looks for light that is scattered from any particle present on the pellicle. During this observation, an optical filter is used to eliminate confusion from particles too small to matter.

The pellicle works like an optical coating tuned to high transmission at the operating wavelength (436 nm). To achieve transmission > 98%, which is required for exposure uniformity, the 865-nm membrane thickness must be uniform within ±10 nm over the entire field.

Success in eliminating repeating defects depends upon contamination-free attachment of the pellicles to the reticle. There can be no trapped particles larger than 3 μm in diameter. Thus, great care is taken in the assembly operation to avoid particles. All assembly is done in a bath of ionized laminar air flow. Reticles are stripped of any organic residue (resist, pellicle adhesive, etc.) in a mixture of sulphuric and chromic acids and then cleaned and dried automatically using a brush, detergent, and a high-pressure water jet. Pellicles are manually cleaned of particles by using a miniature air jet. Inspection of the completed assembly is done using a stereo microscope with illumination designed to highlight any particles and deemphasize the background.

The step-and-repeat optical aligners have been modified to accommodate the use of pellicle-protected reticles. To minimize dust accumulation and contamination, the reticles are stored in special filtered laminar-flow cabinets. Low-pressure airgun blowoff is the only method used to clean the assemblies in production.

-Robert Slutz

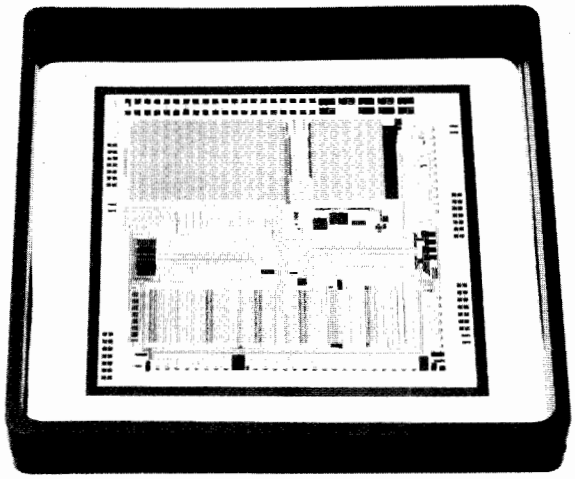


Fig. 1. *Photograph of pellicle used in the NMOS-III photolithography process.*

for patterning nitride (masks 1 and 4 in the NMOS-III process), first metal (mask 6), and vias (mask 7). PMMA also forms an effective ion implant stop for forming depletion loads (mask 2). A wet etch process was developed to pattern the gate oxide (mask 3) and contacts (mask 5). The etchant is NH_4F and water, buffered with citric acid. Other more conventional buffers such as acetic acid were found to induce unacceptable resist lifting at pattern edges.

Acknowledgments

In addition to the project team members mentioned on page 6, major contributions were made by Mike Bennett, Mark Anderson, and Mitch Weaver at HP's Systems Technology Operation, by Mung Chen and Rick Trutna at HP's Cupertino Integrated Circuits Operation, and by Mike Watts at HP Laboratories.

Authors

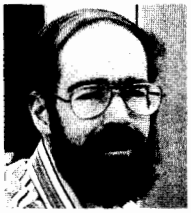
August 1983

3

Eugene R. Zeller

With HP since 1969, Gene Zeller has contributed to a number of HP's computer products, including the 9160A, 9101A, and 9102A peripherals for the 9100A/B Calculator, memory design for the 9810A and 9820A Computers, and logic design for the 9871A Printer. He was a project manager for NMOS-II and thin-film process development and the 9845B Computer, a section manager for NMOS-III chip design, and currently is section manager for NMOS-III process development. He is a coauthor of two papers related to NMOS-III chip and system design. Born in Rockford, Illinois, Gene received a BSEE degree in 1969 from the University of Wisconsin at Madison, and an MSEE degree in 1972 from Colorado State University. Married and the father of two daughters, he lives in Loveland, Colorado. He likes bicycle touring and camping in the Colorado Rocky Mountains.

S. Dana Seccombe



Dana Seccombe studied electrical engineering at the Massachusetts Institute of Technology, receiving the BS and MS degrees in 1971 and the Electrical Engineer degree in 1972. He then joined HP and has worked on numerous aspects of the NMOS-II and NMOS-III processes, first as an R&D engineer, and then with increasing management responsibility resulting in his current position as R&D Lab Manager for HP's Systems Technology Operation. His work has resulted in several patents and papers related to NMOS III. He is a member of the IEEE and has served on the program committee for the ISSC Conference and as guest editor for the IEEE's Journal of Solid-State Circuits. Outside of work, he enjoys sailing, woodworking, and skiing.

Joseph W. Beyers

Born in Pana, Illinois, Joe Beyers attended the University of Illinois, receiving a BS degree in computer engineering in 1973 and an MS degree in electrical engineering in 1974. He joined HP shortly thereafter and contributed to the design of the 9825 Computer's operating system. Part of his work on the 9825 resulted in a patent. Joe worked on the architectural definition of the 32-bit VLSI computer chip set before assuming his current responsibilities as a section manager. He is coauthor of two papers related to this chip set, one of which won the Best Paper Award at the 1982 ISSC Conference. He is an active member of the IEEE, serving as treasurer of the Denver Section in 1982, He is married (his wife works in marketing at HP), has two children, and lives in Fort Collins, Colorado. He enjoys hiking, downhill skiing, water skiing, and running (he ran in the Boston Marathon in 1981).

7

Kevin P. Burkhart

Kevin Burkhart is from Pueblo, Colorado and attended Colorado State University. With HP since receiving his BSEE degree in 1976, he contributed to the design of the 32-bit VLSI CPU chip and the test system for evaluating it and the other 32-bit computer chips. Currently a production engineer supporting these chips, he is a member of the IEEE, married, and lives in Loveland, Colorado. His outside interests include camping, alpine skiing, and fly-fishing using flies he has tied.

Darius F. Tanksalvala

Joining HP in early 1975 after receiving a PhDEE degree from the University of Wisconsin, Darius Tanksalvala also has a BTech degree in electronics awarded in 1967 by the Indian Institute of Technology at Kharagpur and an MTech degree in electrical engineering awarded in 1970 by the Indian Institute of Technology at Bombay. He contributed to the design of the 32-bit VLSI chip and now is working on other NMOS chip designs. He lives in Denver, Colorado and enjoys hiking, backpacking, and listening to music.

Mark A. Forsyth

A contributor to the design, software tool development, and test strategy for the 32-bit VLSI CPU chip, Mark Forsyth is now involved with NMOS-III chip characterization and yield improvement. Born in Minneapolis, Minnesota, he studied electrical engineering at the University of Minnesota. He received a BS degree in 1978 and joined HP shortly thereafter. Mark lives in Fort Collins, Colorado and is a member of the Larimer County Search and Rescue Team. He is married and enjoys hiking, running, and both cross-country and downhill skiing.

Mark E. Hammer

A native of Valier, Montana, Mark Hammer attended Montana State University, earning a BS degree in mathematics in 1974 and a BS degree in electrical engineering in 1976. He joined HP in 1976 and worked on the ALU portion of the 32-bit VLSI CPU chip and the related IC design tools. Mark is presently involved with developing further tools for IC design. He is married and the father of two sons, and lives in Loveland, Colorado. When not working on his house and yard, he enjoys running and hunting.

9

James G. Fiasconaro

Jim Fiasconaro has a BSEE degree awarded by Tulane University in 1968 and an MS degree (1970) and a PhD degree (1973) in electrical engineering awarded by the Massachusetts Institute of Technology. After working at an eastern U.S.

research facility for three years, he joined HP in 1976 and contributed to the definition and implementation of the instruction set for the 32-bit VLSI CPU chip. He is married (his wife is a librarian at HP), lives in Loveland, Colorado, and enjoys cross-country skiing, hiking, backpacking, and photography.

11

Donald R. Weiss

A project manager at HP's facility in Fort Collins, Colorado, Don Weiss is involved with increasing IC design productivity. He joined HP in 1976 and contributed to the hardware design of the 32-bit VLSI I/O processor chip before assuming his current responsibilities. A graduate of the University of Wisconsin at Madison, he received a BSEE degree in 1975 and an MSEE degree in 1976. Don was born in Monroe, Wisconsin, is married and the father of two daughters, and lives in Loveland, Colorado. His interests include running, bicycling, home computer programming, photography, and woodworking.

William F. Jaffe

Graduating from Arizona State University with a BSEE degree in 1973, Bill Jaffe worked for an avionics manufacturer until he decided to continue his studies at Stanford University in 1976. After receiving an MSEE degree in 1977, he joined HP and contributed to the development of the 32-bit VLSI I/O processor and RAM chips. Bill was born in New York City and is a member of the IEEE. He is married, lives in Fort Collins, Colorado, and is interested in piano, cycling, running, backpacking, skiing, and birding.

Fred J. Gross

Fred Gross joined HP in 1969. He contributed to the design of the processor for the 9810, 9820, and 9830 Calculators, the memory for the 9830, and the power supply and memory for the 9825 Calculator. Fred worked on the microcode for the 32-bit VLSI I/O processor chip and currently is working on microcode for future products. A native of Omaha, Nebraska, he studied electrical engineering at the University of Nebraska, receiving a BS degree in 1969. He continued his studies at Colorado State University and earned an MSEE degree in 1974. Fred is a member of the IEEE Computer Society and the Association for Computing Machinery. Living in Fort Collins, Colorado, he is married and has four children. His outside interests include motorcycle riding and home electronics and computing (he writes computer programs for his wife's business).

14

Clifford G. Lob

Cliff Lob is the coauthor of two papers related to the VLSI chips developed for the HP 9000 Computer. He started at HP in 1973 and worked on NMOS-II chip designs before his work on the memory controller chip. Cliff currently is a project manager responsible for several aspects of the VLSI memory system. He was born in Urbana, Illinois, and attended the nearby University of Illinois, earning a BSEE degree in 1973. Living in Fort Collins, Colorado, he is a member of the Ski Patrol at the nearby Winter Park ski area and is also interested in marathon running.

Mark J. Reed

Mark Reed was born in Aberdeen, Maryland, and attended Case Western Reserve University where he received a BSEE degree in 1974 and an MSEE degree in 1976. He joined HP that year and worked on a system to extract logic equations from flow charts and contributed to the design of the memory controller chip and a dedicated tester for the NMOS-III process. He is the author of a paper about a microprocessor-based control system. He is interested in skiing, gardening, and boardsailing, and lives in Fort Collins, Colorado.

Joseph P. Fucetola

Joe Fucetola received a BSEE degree from the New Jersey Institute of Technology in 1971 and an MSEE degree from the University of Illinois in 1972. He joined HP shortly thereafter and worked on the NMOS-II process for the 9825A Calculator. Joe was project manager for the 32-bit VLSI memory system before assuming his current position as process manager for NMOS-III products and wafer test. His work has resulted in four papers about the 32-bit VLSI computer system. He is married, has two sons, and lives in Fort Collins, Colorado. His interests include skiing, hiking, running, and woodworking.

Mark A. Ludwig

With HP since 1974, Mark Ludwig contributed to the design and production of the I/O controller chip for the 9825 Calculator and the 32-bit VLSI memory controller chip for the HP 9000 Computer. He studied electrical engineering at the University of Wisconsin at Madison and received a BSEE degree in 1972 and an MSEE degree in 1973. Born in Chilton, Wisconsin, he now lives in Loveland, Colorado. Outside of work, he enjoys downhill skiing, bicycling, and playing chess.

17

Alexander O. Elkins

Joining HP in 1978 after receiving a BS degree in electrical engineering from California State Polytechnic University, Alexander Elkins is an HP 3000 Systems manager at HP's Fort Collins, Colorado facility. He contributed to the design of the clock chip for the 32-bit VLSI computer system and currently is working on a buffer interface. Born in Fort Worth, Texas, he now lives in Fort Collins.

20

Dale R. Beucler

Dale Beucler joined HP in 1978 after completing his studies for a BSEE degree at California State Polytechnic University. He contributed to the design, testing, and production of the 128K-bit RAM. Dale grew up in West Covina, California, and now lives in Fort Collins, Colorado. He enjoys bicycle riding, running, backpacking, cross-country skiing, and bicycling with his wife on their tandem bicycle.

John K. Wheeler

With HP since 1976, John Wheeler worked on and then managed the 128K-bit RAM project before assuming his current responsibilities as a project manager for VLSI design and packaging. He was born in San Diego, California and holds a BS degree in electrical engineering awarded by the University of Colorado at Boulder in 1976. John is coauthor of one paper on the 32-bit VLSI computer system and a member of the IEEE. His outside interests include photography, volleyball, fishing, reading, and recreational mathematics when he is not involved with his primary interest, spending time with his wife and two children. He lives in Loveland, Colorado.

John R. Spencer

Born in Pampas, Texas, John Spencer attended Texas Tech University and received a BS degree in electrical engineering in 1976. Now a project manager responsible for future VLSI computer chips and laboratory productivity, he started at HP in 1976 and worked on the design of the 128K-bit RAM. He served four years in the U.S. Navy and is a member of the IEEE. Married

and the father of three children, he lives in Loveland, Colorado. He is actively involved with his church and the local chapter of Gideons International when not enjoying trout fishing, fly tying, motorcycle riding, and volleyball playing.

Charlie G. Kohlhardt

Graduating from the University of Wisconsin at Madison with a BSEE degree in 1978, Charlie Kohlhardt then joined HP and worked on the design, characterization, and production testing of the 128K-bit RAM. He currently is working on other NMOS-III designs. He is married and lives in an 80-year-old house in downtown Fort Collins, Colorado. When not busy remodeling his home, he enjoys downhill ski racing, water skiing, and other outdoor sports.

24

Glen E. Leinbach

Glen Leinbach started at HP in 1973 as a materials engineer at the Loveland Instrument Division. He contributed to the production of the 9825 Calculator and the design of the finstrate before assuming his current responsibilities for finstrate-IC assembly. A graduate of the University of Denver with a BSME degree awarded in 1973, Glen was born in San Diego, California. He is an author of a paper about IC assembly problems. He has a variety of outside interests that include restoring a 1950 DeSoto, skiing, whitewater rafting, volleyball, softball, singing tenor in a men's quartet, and playing trumpet in the local HP instrumental band (HPIB). Recently married, he lives in Fort Collins, Colorado.

Arun K Malhotra

Born in New Delhi, India, Arun Malhotra attended the India Institute of Technology at New Delhi, earning a B.Tech degree in 1970. He continued his studies at Purdue University and received an MSEE degree in 1972 and a PhDEE degree in 1974. After teaching as an assistant professor for two years, he joined HP in 1976. He was a project manager for NMOS-III and finstrate process development and for assembly and reliability related to the HP 9000 Computer. He recently left HP to manage technology development at a new microelectronics company. His work has resulted in twelve papers about thin-film and amorphous semiconductor technology. A member of the IEEE and the Electrochemical Society, he is married, has one son, and lives in Mountain View, California. When not busy with work, he enjoys jogging and playing racquetball.

Jeffery J. Straw

A native of Grand Rapids, Michigan, Jeff Straw studied electrical engineering at Michigan Technological University and received a BSEE degree in 1976 and an MSEE degree in 1978. He then joined HP and contributed to the development of the design and assembly process for the finstrates used in HP's 32-bit VLSI computer system. An author of a paper on computer-aided reliability screening, he is married and the father of two children, and lives in Loveland, Colorado. He recently completed building his home, sings bass in his church choir, and enjoys running and playing volleyball.

Guy R. Wagner

Joining HP in 1981 with several years of experience in designing PBX telephone systems, Guy Wagner contributed to several parts of the Memory/Processor Module design. He currently is working on VLSI packaging technology. Born in Dubuque, Iowa, Guy attended Iowa State University, earning a BSME degree in 1970 and an MSME degree in 1972. He is a member of the IEEE and the International Electronic Packaging Society and his work has resulted in one patent related to printed circuit mounting. He is married, has a daughter, and lives in Loveland, Colorado. His interests include photography, flying, camping, and radio-controlled model airplanes.

27

Fung-Sun Fei

Fung-Sun Fei received a PhD degree from the University of Virginia in 1976 and joined HP shortly after. Before assuming his current responsibilities as an R&D project manager for defect density reduction and reliability improvement, he managed the NMOS-III metallization process project. He is married, has a son, and lives in Fort Collins, Colorado. His interests include hiking, swimming, and bicycling.

James M. Mikkelson

Graduating from the Massachusetts Institute of Technology in 1972 with the BS, MS, and Electrical Engineer degrees in electrical engineering, Jim Mikkelson started at HP in early 1973 and worked on the development of the NMOS-II process before becoming an NMOS-III design and development project manager. He is coauthor of several papers related to the NMOS-III process (one of which received the Outstanding Paper Award at the 1981 ISSC Conference). Jim is a member of the IEEE and Sigma Xi. Born in Great Falls, Montana, he is married, has a daughter, and lives in Loveland, Colorado. Outside of work, he enjoys sailing, cross-country skiing, bicycling, woodworking, and touring on his motorcycle.

30

Norman E. Hendrickson

Norm Hendrickson graduated from the University of Minnesota in 1978 with an MSEE degree. He then joined HP and contributed to the development of the NMOS-III process at HP's facility in Fort Collins, Colorado. His outside interests include rock concerts, bicycle racing, and skiing over big moguls. He lives in Fort Collins.

Donald E. Novy, Jr.

Since joining HP in 1980, Don Novy has characterized and supported the production of several of the materials used in the NMOS-III metallization process. He was born in Hoffman Estates, Illinois and attended Purdue University where he earned a BSEE degree in 1979 and an MSEE degree in 1980. He is married, lives in Fort Collins, Colorado, and is interested in amateur radio and designing and building a home computer.

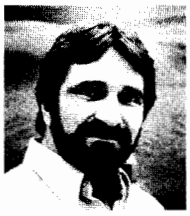
Daniel D. Kessler

An IC process engineer at HP's facility in Fort Collins, Colorado, Dan Kessler worked on several tungsten deposition processes for NMOS III and yield improvement for the 32-bit VLSI CPU chip. He joined HP in 1980 after receiving an MSEE degree from the Massachusetts Institute of Technology where he also earned a BSEE degree in 1978. Born in Lansing, Michigan, Dan is the author of a paper about a new photoconductive device. He has a daughter, lives in Fort Collins, and enjoys bicycling, camping, hiking, tennis, and racquetball.

David W. Quint

Dave Quint has the BS and MS degrees in electrical engineering awarded by the University of Wisconsin and a PhDEE degree awarded by the Massachusetts Institute of Technology. He started at HP in late 1979 and worked on NMOS III as a process engineer. Dave is the author of one paper and a coinventor for two patents, one related to CVD tungsten deposition. A native of Wisconsin, he served in the U.S. Air Force before beginning his college education. He is married, has three children, and lives in Fort Collins, Colorado.

James P. Roland



Jim Roland received a BSEE degree from General Motors Institute in 1970 and an MS degree from Purdue University in 1971. He worked for a research laboratory for two years and then returned to his studies at Purdue and earned a PhD degree in 1977. Jim then joined HP and worked on several aspects of the NMOS-III metallization process. He is the author of a paper about NMOS-III metallization and coinventor of a patent on a liquid-level sensor design. He was born in Detroit, Michigan, is married, and lives in Fort Collins, Colorado. He is vice-president of the local chapter of the Optimists and is interested in skiing, bicycling, racquetball, and camping.

Gary L. Hillis

A native of Kokomo, Indiana, Gary Hillis attended Purdue University and received a BSEE degree in 1978 and an MSEE degree in 1979. He then joined HP and was part of the initial team establishing the photolithographic facility at HP's plant in Fort Collins, Colorado. Gary is a coauthor of a paper on photolithography and coinventor of a patent related to the use of absorbing dyes in photoresist. He is married and the father of a daughter, and lives in Fort Collins. His interests include sailing, skiing, and basketball.

Howard E. Abraham

With HP since 1969, Howard Abraham has worked on a number of projects, including microwave device design, NMOS-II process and product development, and most recently, NMOS-III photolithographic technology. His work has resulted in two patents related to semiconductor device fabrication and a paper on transistor device design. Before joining HP, he was a flight control engineer for the Gemini and Surveyor space missions. Howard has a BSEE degree (1962) and an MSEE degree (1964) from the University of Wisconsin at Madison, and has done three years of graduate work at the University of California at Berkeley. He is a member of the IEEE and Sigma Xi. Married and the father of two children, he lives in Loveland, Colorado. Outside of work, he sings in a church choir, is interested in solar energy applications and classic automobiles, and enjoys motorcycling, computer programming, and flying.

Mark Stolz

Born in Presque Isle, Maine, Mark Stolz studied electrical engineering and computer science at the University of Colorado and received a BS degree in 1976. He then worked on high-speed data communication systems for a major defense contractor before joining HP in 1979. He installed the environmental control system for the NMOS-III process area and currently is responsible for the step-and-repeat aligners used for NMOS-III production. Mark is a member of the National Society of Professional Engineers. Living in Fort Collins, Colorado, he likes most outdoor activities, but particularly enjoys skiing, backpacking, bicycling, tennis, scuba diving, softball, and flying.

Keith G. Bartlett

Graduating from Colorado State University with a PhD degree in physics in 1977, Keith Bartlett worked on MOS memories for a major semiconductor manufacturer before joining HP in 1979. He worked on photolithography for the NMOS-III process and now is a production engineering project manager. His contributions have resulted in four patents and several papers related to optics and IC processing. He is married, has a daughter, and lives in Fort Collins, Colorado. When not training bird dogs, he enjoys fly fishing.

Martin S. Wilson

Before coming to HP in 1973, Marty Wilson worked on rocket and ramjet engine development. At HP, he contributed to the design of the thermal printer for the 9845 Computer, and currently is an NMOS-III photolithography project manager. He received the BS and MS degrees in aerospace engineering at the University of Colorado in 1967 and an MBA degree at California State University in 1972. Born in Denver, Colorado, he now lives in Loveland, Colorado, is married, and has two daughters. His interests include building projects, backpacking, hunting, fishing, and collecting Greek and Roman pottery.

HEWLETT-PACKARD JOURNAL

HEWLETT PACKARD

5953-8514