# THE HISTORY OF THE 4004

**Federico Faggin**

*Synaptics, Inc.*

**Marcian E. Hoff Jr.**

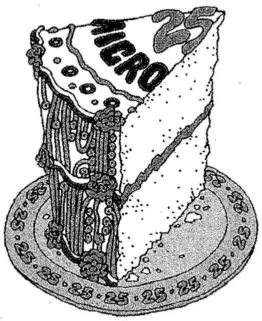
*Teklicon*

**Stanley Mazor**

*BEA Systems*

**Masatoshi Shima**

*VM Technology Inc.*



*The 4004 design team tells its story.*

Twenty-five years ago, in November 1971, an advertisement appeared in *Electronic News:* "Announcing a new era in integrated electronics, a microprogrammable computer on a chip." The ad was placed by Intel Corporation of Santa Clara, California, then just over three years old. From that modest but prophetic beginning, the microprocessor market has grown into a multibillion-dollar business, and Intel has maintained a leadership position, particularly in microprocessors for personal computers.

In 1968, Bob Noyce and Gordon Moore, who had both just left Fairchild Semiconductor, founded Intel Corporation, and operations began in September of the same year. The new company was committed to developing semiconductor mainframe memory products using both bipolar and MOS (metal-oxide-semiconductor) technologies. Bipolar processes offered faster access times, while MOS processes promised higher chip complexity—that is, more memory bits per chip. Rather than use the established technologies of the day, Intel was determined to use new bipolar and MOS processes similar to those Fairchild Semiconductor had just developed. For the MOS products, Intel chose a self-aligned P-channel silicon-gate process.

Intel intended to produce proprietary memory products, rather than a specific product for each customer. Though this strategy offered high potential sales volume, it increased the design time. To optimize its revenue stream, therefore, Intel remained open to limited custom work, hoping that customers would be ready to use the products as soon as they were working. The company did not project custom products to reach the high sales volumes it expected of the proprietary products, but it hoped they would provide an important source of revenue until the memory products were established.

## Busicom

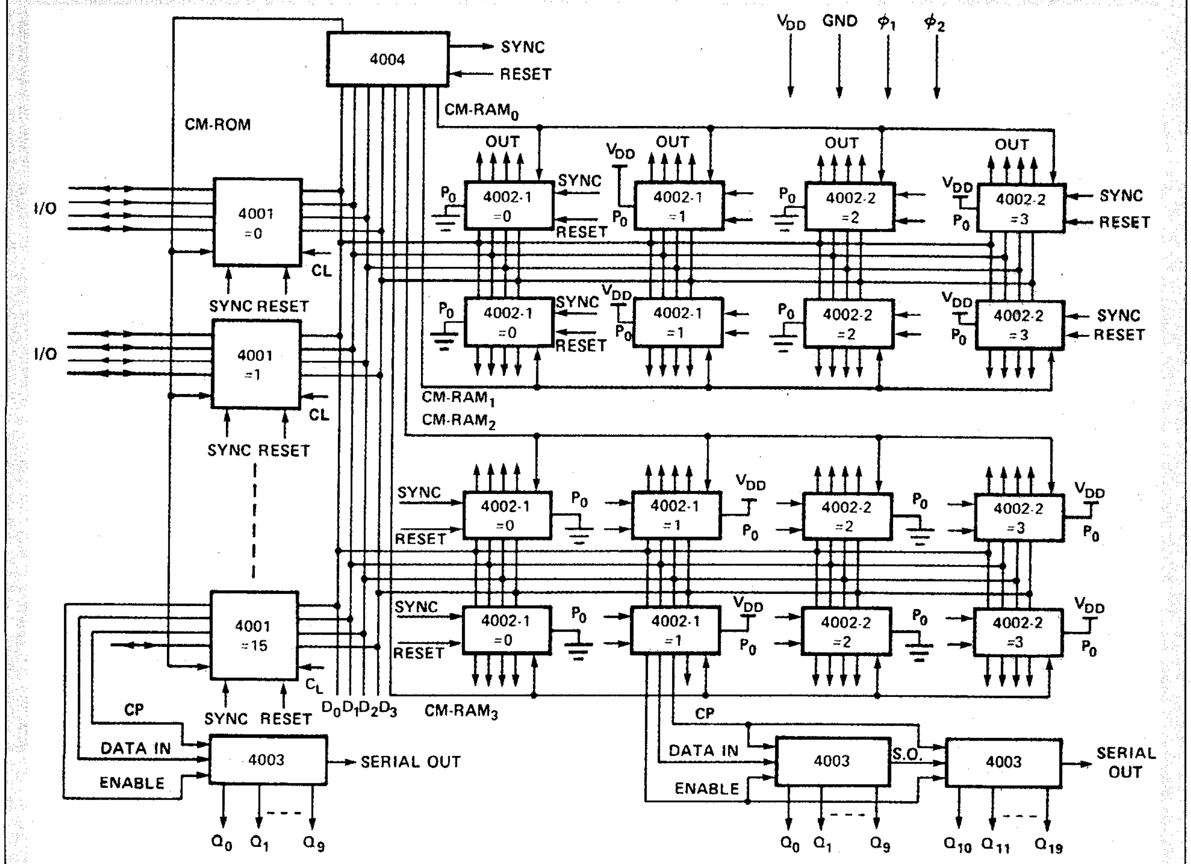
In April of 1969, Intel agreed to develop a set of calculator chips for a Japanese firm. The firm consisted of two companies: Electro-Technical Industries handled product development, and Nippon Calculating Machines Company handled marketing. The calculators bore the brand name Busicom. Busicom intended to use the chip set in several different models of calculators, from a low-end desktop printing calculator to calculator-like office machines such as billing machines, teller machines, and cash registers. The firm made arrangements for three of its engineers to come to Intel to finish the logic design for the calculator chips and to work with Intel personnel to transfer the designs into silicon. The three engineers from Japan—Masatoshi Shima and his colleagues Masuda and Takayama—arrived in late June.

Intel assigned Marcian E. Hoff Jr. to act as liaison to the Japanese engineers. Hoff had received his PhD from Stanford University in 1962 and had remained there as a research associate working on electronic neural networks until he joined Intel in September 1968. At Stanford, Hoff had programmed and built hardware interfaces for IBM model 1620 and 1130 computers. He was Intel's twelfth employee and received the title manager of applications research.

Hoff's duties were to help define Intel products, meeting with customers and marketing personnel as necessary. In addition, as Intel products became available, he would develop applications information to help customers use those products. However, because in early 1969 the products were still in development, and there were limited opportunities to question potential customers, Hoff had taken on several tasks peripheral to his primary duties.

Although Hoff was only supposed to act as liaison to the Busicom engineering team,

0272-1732/96/$5.00 © 1996 IEEE



## System interconnection

This figure shows the 4004 CPU and its main 4-bit multiplexed, tristate bus driving up to 16 ROM (4001) and up to 16 RAM (4002) chips. The 4004 sends six other control signals to all the 4001 and 4002 chips. Each 4002 contains four registers of 20 nibbles (16+4 × 4 bits) each. It also contains a 4-bit output port. Each 4001 contains 256 bytes of ROM and a 4-bit I/O port; ROM and ports are metal-mask programmable. The 4003 is a 10-bit serial-in, parallel-out and serial-out shift register used for keyboard scanning, printer control, and so on. The output ports of the 4001/4002 drive the 4003.

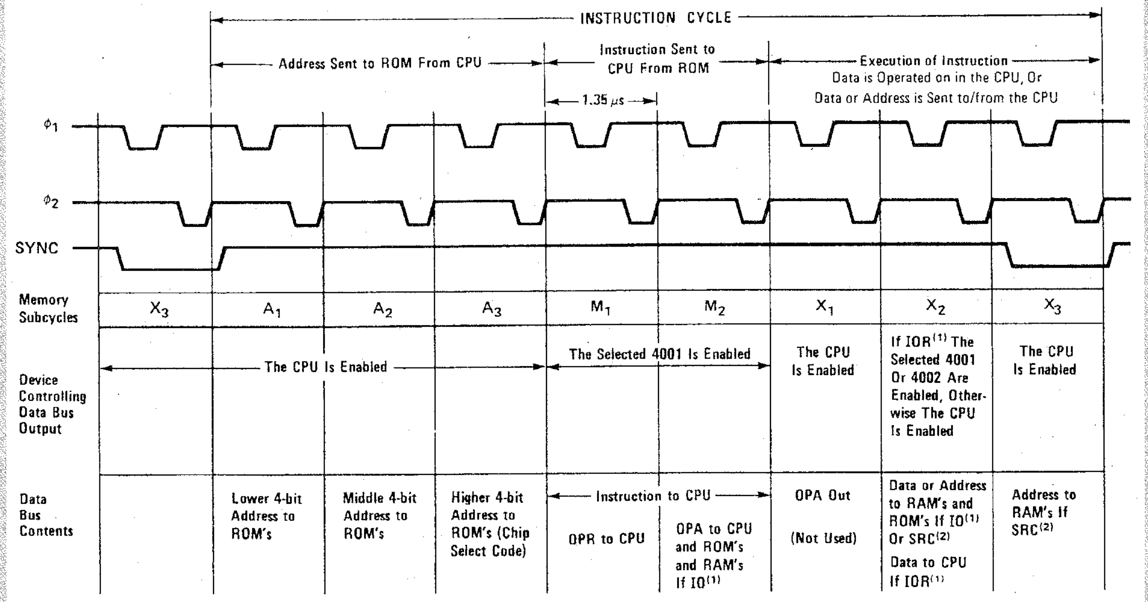
All the chips are packaged in 16-lead DIPs. The 4000 chip set can have up to 4 Kbytes of ROM (sixteen 4001 chips), 1,280 nibbles of RAM (sixteen 4002 chips), 32 directly addressable 4-bit I/O ports, and an unlimited number of output ports via the 4003. The addition of a few external gates doubles the amount of addressable ROM or RAM. (The RAM chip can store only data, not instructions.) The minimum system configuration consists of one 4004 (CPU) and one 4001 (ROM with 4-bit I/O port).

curiosity about the calculator led him to study the design. His first reaction was surprise at how complex the calculator logic was, particularly when compared to the general-purpose digital computers he had used. In addition, the interconnections between the various chips were extensive, requiring large and expensive packages. Having attended several meetings on the project's cost objectives, Hoff became concerned that the packaging requirements alone might prevent Intel from meeting those objectives.

Busicom had proposed a ROM-based, macroinstruction programmable decimal computer consisting of seven different LSI chips: program control, decimal arithmetic unit, timing logic, ROM, shift register, printer control, and output ports. Busicom had already successfully implemented this design in commercial products since 1968, using transistor-transistor logic (TTL) and ROM.

Hoff expressed some of his concerns about packaging and design complexity to Intel's upper management—designing that many chips could be a daunting task for the limited chip design staff. Bob Noyce particularly encouraged him to pursue an alternative design if one appeared feasible.

Hoff was initially reluctant to deviate too far from the original Busicom design. While some aspects of the proposed chip set were similar to those of other calculators of the day, it also included some novel capabilities. Most notable were the use of ROM for macroinstruction storage and a specialized instruction set that would allow various calculator-intensive machines to use the same chips. Another innovative


INSTRUCTION CYCLE
Address Sent to ROM From CPU
Instruction Sent to CPU From ROM
1.35 μs
Execution of Instruction
Data is Operated on in the CPU, Or Data or Address is Sent to/from the CPU
ϕ1
ϕ2
SYNC

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Memory Subcycles | $X_3$ | $A_1$ | $A_2$ | $A_3$ | $M_1$ | $M_2$ | $X_1$ | $X_2$ | $X_3$ |
| Device Controlling Data Bus Output | The CPU Is Enabled | | | | The Selected 4001 Is Enabled | | The CPU Is Enabled | If IOR(1) The Selected 4001 Or 4002 Are Enabled, Otherwise The CPU Is Enabled | The CPU Is Enabled |
| Data Bus Contents | | Lower 4-bit Address to ROM's | Middle 4-bit Address to ROM's | Higher 4-bit Address to ROM's (Chip Select Code) | Instruction to CPU<br>OPR to CPU | OPA to CPU and ROM's and RAM's If IO(1) | OPA Out<br>(Not Used) | Data or Address to RAM's and ROM's If IO(1) Or SRC(2)<br>Data to CPU If IOR(1) | Address to RAM's If SRC(2) |

## Timing diagram of the 4000 series

A single −15-V power supply powers the P-channel MOS chip set. (Alternatively, it could use −12 V and +5 V to allow TTL compatibility, a very useful feature in the design of microcomputer development tools.) It uses two interleaved clocks, $\phi_1$ and $\phi_2$ as shown. The clock frequency is 750 kHz, and the instruction cycle requires 8 or 16 clock cycles, depending on the instruction.

The 4004 generates a synchronizing signal, SYNC, which marks the beginning of the instruction cycle and which the 4004 sends to all the 4001 and 4002 chips. During the first three clock cycles, the 4004 sends a 12-bit address to the ROM chips; the selected 4001 outputs its 8-bit instruction opcode in the next two clock periods. The 4004 interprets and executes the instruction during the next three clock periods. A few instructions—like Jump, for example—require two bytes and execute in 16 clock periods. The typical instruction execution time is 10.8 μs, or 21.6 μs for the double-length instructions.

feature was the variable amount of shift register memory that the design could use, with the different calculator models having different numbers of memory registers.

Like many calculators of the day, Busicom's design used shift registers for memory. Shift registers are quite fast for the arithmetic calculations, display, and printing that calculators require, but are slow for operations requiring random access. Shift registers used six transistors per bit, like static RAM, but a shift register cell was smaller than the RAM cell. The shift register's size advantage, however, was offset by increased control logic complexity and slower speed for random access. Any access to even a portion of a memory register required a complete scan through that register. Such slow memory access would make a conventional CPU architecture too slow to be practical.

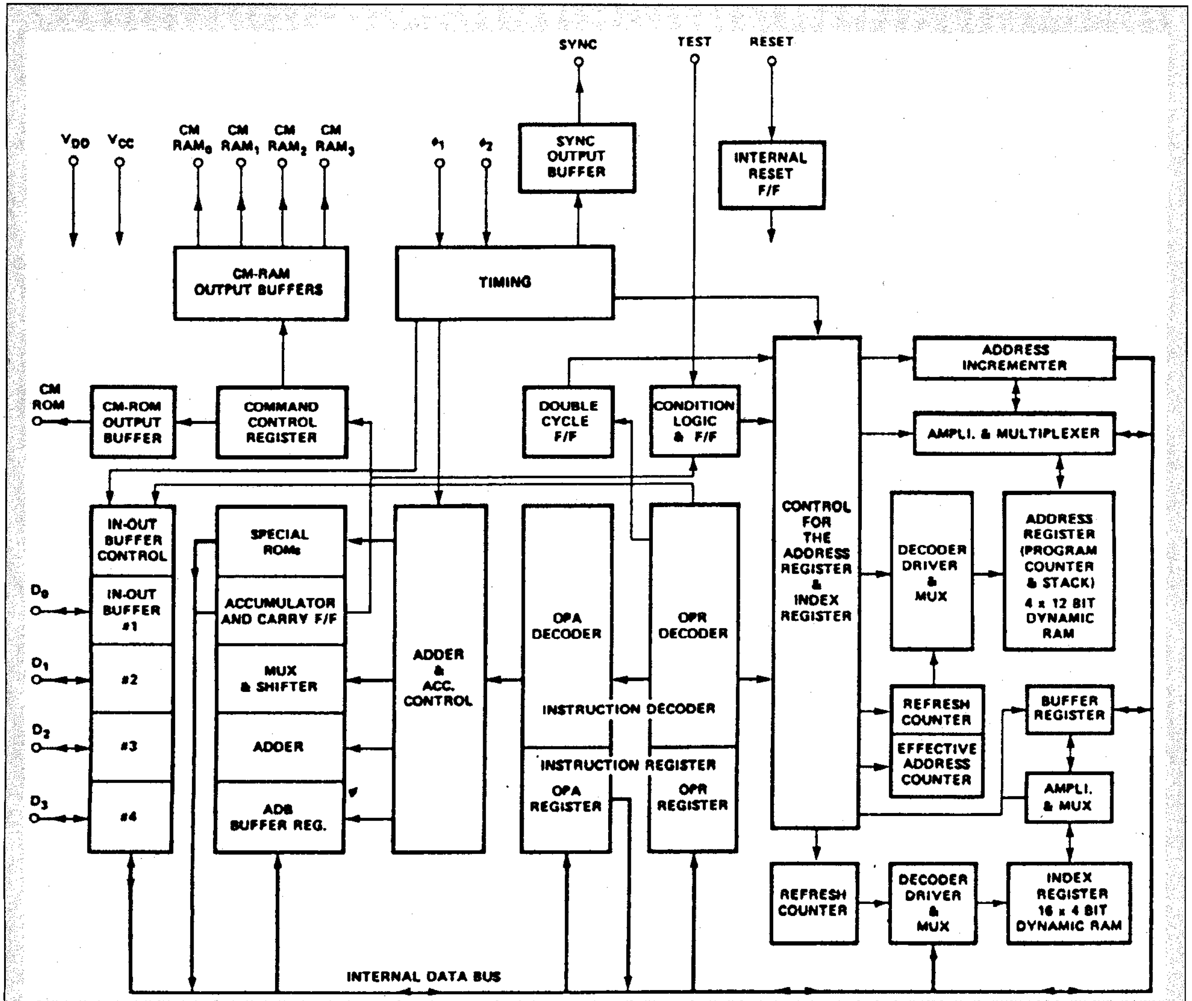
### The 4004 is conceived

Intel had just begun working on dynamic MOS RAM, using a three-transistor cell. Hoff, aware of that development, thought that if he could solve its refresh problem in the calculator environment, the DRAM would be an ideal alternative to the shift register memory. Unlike the shift register, the DRAM could be accessed in as small a quantum as desired. In addition, the three-transistor DRAM cell used even less silicon area than the shift register cell.

One of the first modifications to the Busicom design Hoff considered was adding subroutine capability to the instruction set. Subroutines of simple instructions could replace more complex instructions, which should allow simplification of the hardwired logic. Although the Busicom engineers appeared unreceptive to Hoff's proposals, with Noyce's encouragement, Hoff continued exploring options.

Hoff began to consider the design of a general-purpose computer that might be programmed to perform calculator functions. The computer would fetch program instructions from a ROM into an arithmetic chip. The arithmetic chip would interpret the instructions, reading and writing as necessary to DRAM chips. The arithmetic chip would also have local "scratch-pad" registers. During the time the arithmetic chip was fetching program instructions from ROM, the DRAMs could be refreshed, since no instruction execution would be occurring at that time. The data quantum would be 4 bits to allow binary-coded decimal (BCD) arithmetic.

Hoff performed these studies of a general architecture in July and August of 1969. During this time, he believed the Busicom team was unresponsive to his idea. On the contrary, Busicom engineers recognized that Hoff's proposal of a general-purpose CPU was more advantageous than their



## Block diagram of the 4004 CPU

The 4004 contains 16 general-purpose 4-bit registers; one 4-bit accumulator; a four-level, 12-bit push-down address stack containing the program counter and three return addresses for subroutine nesting; a binary and decimal arithmetic unit; instruction register, decoder, and control logic; timing logic; bus control; and miscellaneous control.

In addition to the pins required for the 4-bit tristate data bus, two-phase clock, power, and ground, the 4004 has a SYNC timing output pin and five control lines for addressing the 4001 and 4002 chips (CM ROM and CM RAM0-3; "CM" indicates command). There is also a reset pin to initialize the system and a test input pin. Test provides one of the conditions in the conditional jump instruction (JCN), allowing the 4004 to poll external devices. Later generation processors replaced Test with a much more convenient interrupt facility.

Although several prior computer architectures inspired the 4004 (PDP-8, IBM 1620, and so on), it is unique in many aspects. Its main value resides in its simplicity and economy of means—essential ingredients, given the limited capabilities of 1970 semiconductor technology.

design. The concept was still incomplete, however, and required additional features to function satisfactorily in Busicom's products. Certain calculator functions, such as decimal adjust and keyboard processing, required too many bytes of ROM, and there was no mechanism for real-time control to synchronize the CPU with external events. Also, the RAM chip's organization did not seem well suited for storing the decimal position, sign, and other data necessary for calculating a decimal string of digits.

In September 1969, Stanley Mazor joined Intel from Fairchild, where he had been since 1964, and where he had helped design the Symbol computer. After Mazor arrived, progress began to accelerate.

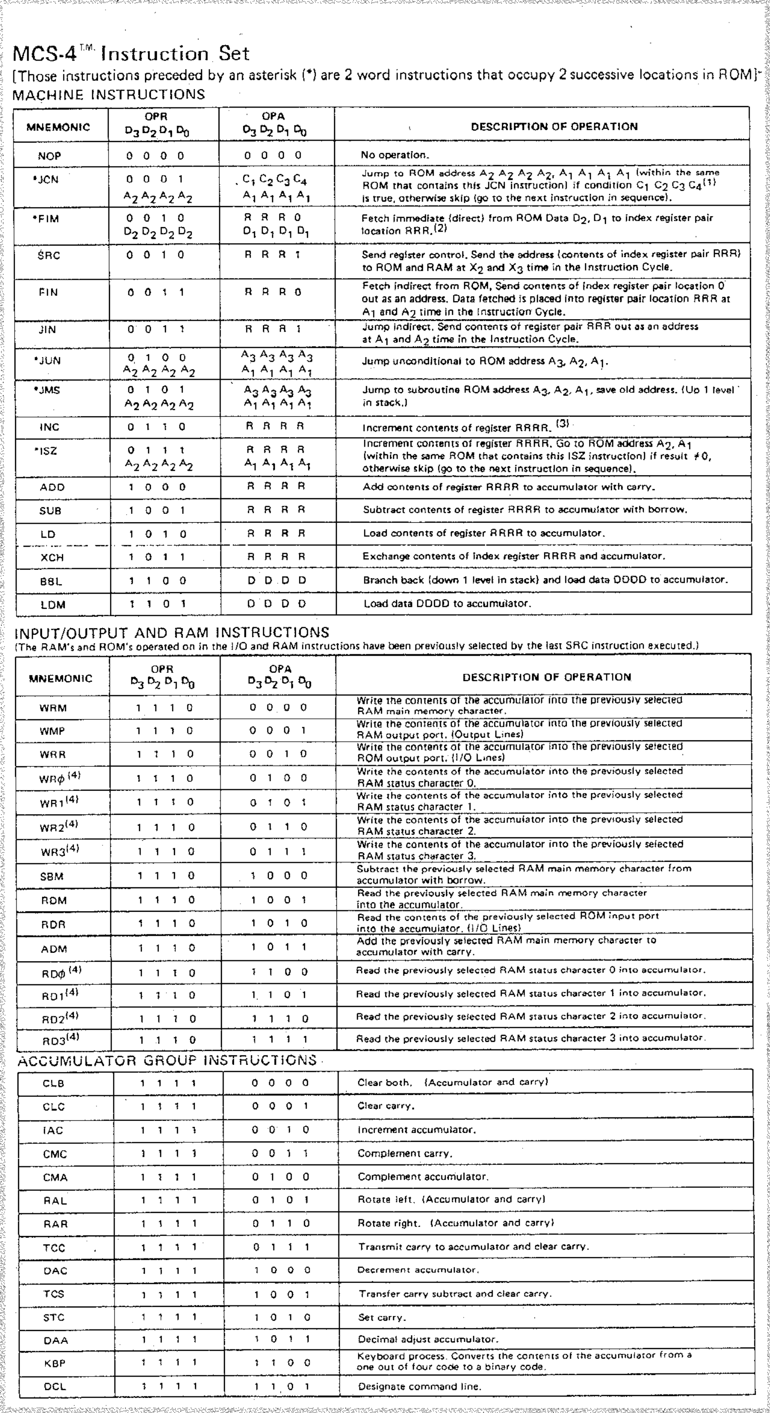
Working together, Mazor and Hoff further refined the idea of a general-purpose design and demonstrated its potential capabilities, addressing the objections raised by the Busicom team. In response to the Busicom engineers' need for macroinstruction capability, Mazor proposed adding a Fetch

## 4004 instruction set

The 4004 has 16 instruction types. The main group contains 14 general instructions: register instructions, conditional and unconditional jumps, increment and skip if zero, and so on.

The I/O and RAM group includes 15 types. RAM data addressing in the 4000 architecture works as follows. First, the DCL instruction selects a group of four RAM chips out of four groups. Then, the SRC instruction selects one 4-bit nibble out of the 256 regular nibbles contained in the selected group of four chips. Finally, one of the I/O and RAM instruction types performs the operation on the selected data. Each RAM chip contains 64 regular nibbles and 16 status nibbles (hence 80 nibbles); the I/O and RAM instructions address the status nibbles directly.

Finally the accumulator group includes 11 binary arithmetic instruction types, a decimal-adjust instruction, a code conversion instruction (KBP, keyboard process) that reduces the number of ROM bytes required for the keyboard-scanning operation, and a memory control instruction (DCL). Most of the instructions execute in 8 clock cycles (10.8 µs).



MCS-4™ Instruction Set

[Those instructions preceded by an asterisk (*) are 2 word instructions that occupy 2 successive locations in ROM]

MACHINE INSTRUCTIONS

| MNEMONIC | OPR<br>$D_3 D_2 D_1 D_0$ | OPA<br>$D_3 D_2 D_1 D_0$ | DESCRIPTION OF OPERATION |
|---|---|---|---|
| NOP | 0 0 0 0 | 0 0 0 0 | No operation. |
| *JCN | 0 0 0 1<br>$A_2 A_2 A_2 A_2$ | $C_1 C_2 C_3 C_4$<br>$A_1 A_1 A_1 A_1$ | Jump to ROM address $A_2 A_2 A_2 A_2$, $A_1 A_1 A_1 A_1$ (within the same ROM that contains this JCN instruction) if condition $C_1 C_2 C_3 C_4$(1) is true, otherwise skip (go to the next instruction in sequence). |
| *FIM | 0 0 1 0<br>$D_2 D_2 D_2 D_2$ | R R R 0<br>$D_1 D_1 D_1 D_1$ | Fetch immediate (direct) from ROM Data $D_2$, $D_1$ to index register pair location RRR.(2) |
| SRC | 0 0 1 0 | R R R 1 | Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at $X_2$ and $X_3$ time in the Instruction Cycle. |
| FIN | 0 0 1 1 | R R R 0 | Fetch indirect from ROM, Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR at $A_1$ and $A_2$ time in the Instruction Cycle. |
| JIN | 0 0 1 1 | R R R 1 | Jump indirect. Send contents of register pair RRR out as an address at $A_1$ and $A_2$ time in the Instruction Cycle. |
| *JUN | 0 1 0 0<br>$A_2 A_2 A_2 A_2$ | $A_3 A_3 A_3 A_3$<br>$A_1 A_1 A_1 A_1$ | Jump unconditional to ROM address $A_3$, $A_2$, $A_1$. |
| *JMS | 0 1 0 1<br>$A_2 A_2 A_2 A_2$ | $A_3 A_3 A_3 A_3$<br>$A_1 A_1 A_1 A_1$ | Jump to subroutine ROM address $A_3$, $A_2$, $A_1$, save old address. (Up 1 level in stack.) |
| INC | 0 1 1 0 | R R R R | Increment contents of register RRRR. (3) |
| *ISZ | 0 1 1 1<br>$A_2 A_2 A_2 A_2$ | R R R R<br>$A_1 A_1 A_1 A_1$ | Increment contents of register RRRR. Go to ROM address $A_2$, $A_1$ (within the same ROM that contains this ISZ instruction) if result $\neq 0$, otherwise skip (go to the next instruction in sequence). |
| ADD | 1 0 0 0 | R R R R | Add contents of register RRRR to accumulator with carry. |
| SUB | 1 0 0 1 | R R R R | Subtract contents of register RRRR to accumulator with borrow. |
| LD | 1 0 1 0 | R R R R | Load contents of register RRRR to accumulator. |
| XCH | 1 0 1 1 | R R R R | Exchange contents of index register RRRR and accumulator. |
| BBL | 1 1 0 0 | D D D D | Branch back (down 1 level in stack) and load data DDDD to accumulator. |
| LDM | 1 1 0 1 | D D D D | Load data DDDD to accumulator. |

INPUT/OUTPUT AND RAM INSTRUCTIONS

(The RAM's and ROM's operated on in the I/O and RAM instructions have been previously selected by the last SRC instruction executed.)

| MNEMONIC | OPR<br>$D_3 D_2 D_1 D_0$ | OPA<br>$D_3 D_2 D_1 D_0$ | DESCRIPTION OF OPERATION |
|---|---|---|---|
| WRM | 1 1 1 0 | 0 0 0 0 | Write the contents of the accumulator into the previously selected RAM main memory character. |
| WMP | 1 1 1 0 | 0 0 0 1 | Write the contents of the accumulator into the previously selected RAM output port. (Output Lines) |
| WRR | 1 1 1 0 | 0 0 1 0 | Write the contents of the accumulator into the previously selected ROM output port. (I/O Lines) |
| WR$\phi$(4) | 1 1 1 0 | 0 1 0 0 | Write the contents of the accumulator into the previously selected RAM status character 0. |
| WR1(4) | 1 1 1 0 | 0 1 0 1 | Write the contents of the accumulator into the previously selected RAM status character 1. |
| WR2(4) | 1 1 1 0 | 0 1 1 0 | Write the contents of the accumulator into the previously selected RAM status character 2. |
| WR3(4) | 1 1 1 0 | 0 1 1 1 | Write the contents of the accumulator into the previously selected RAM status character 3. |
| SBM | 1 1 1 0 | 1 0 0 0 | Subtract the previously selected RAM main memory character from accumulator with borrow. |
| RDM | 1 1 1 0 | 1 0 0 1 | Read the previously selected RAM main memory character into the accumulator. |
| RDR | 1 1 1 0 | 1 0 1 0 | Read the contents of the previously selected ROM input port into the accumulator. (I/O Lines) |
| ADM | 1 1 1 0 | 1 0 1 1 | Add the previously selected RAM main memory character to accumulator with carry. |
| RD$\phi$(4) | 1 1 1 0 | 1 1 0 0 | Read the previously selected RAM status character 0 into accumulator. |
| RD1(4) | 1 1 1 0 | 1 1 0 1 | Read the previously selected RAM status character 1 into accumulator. |
| RD2(4) | 1 1 1 0 | 1 1 1 0 | Read the previously selected RAM status character 2 into accumulator. |
| RD3(4) | 1 1 1 0 | 1 1 1 1 | Read the previously selected RAM status character 3 into accumulator. |

ACCUMULATOR GROUP INSTRUCTIONS

| | | | |
|---|---|---|---|
| CLB | 1 1 1 1 | 0 0 0 0 | Clear both. (Accumulator and carry) |
| CLC | 1 1 1 1 | 0 0 0 1 | Clear carry. |
| IAC | 1 1 1 1 | 0 0 1 0 | Increment accumulator. |
| CMC | 1 1 1 1 | 0 0 1 1 | Complement carry. |
| CMA | 1 1 1 1 | 0 1 0 0 | Complement accumulator. |
| RAL | 1 1 1 1 | 0 1 0 1 | Rotate left. (Accumulator and carry) |
| RAR | 1 1 1 1 | 0 1 1 0 | Rotate right. (Accumulator and carry) |
| TCC | 1 1 1 1 | 0 1 1 1 | Transmit carry to accumulator and clear carry. |
| DAC | 1 1 1 1 | 1 0 0 0 | Decrement accumulator. |
| TCS | 1 1 1 1 | 1 0 0 1 | Transfer carry subtract and clear carry. |
| STC | 1 1 1 1 | 1 0 1 0 | Set carry. |
| DAA | 1 1 1 1 | 1 0 1 1 | Decimal adjust accumulator. |
| KBP | 1 1 1 1 | 1 1 0 0 | Keyboard process. Converts the contents of the accumulator from a one out of four code to a binary code. |
| DCL | 1 1 1 1 | 1 1 0 1 | Designate command line. |

Notes:

(1) The condition code is assigned as follows:

$C_1$ = 1 Invert jump condition
$C_1$ = 0 Not invert jump condition
$C_2$ = 1 Jump if accumulator is 0
$C_3$ = 1 Jump if carry/link is a 1
$C_4$ = 1 Jump if test signal is a 0

(2) RRR is the address of one of eight index register pairs in the CPU.

(3) RRRR is the address of one of 16 index registers in the CPU.

(4) Each RAM chip has four registers, each with twenty 4-bit characters subdivided into 16 main memory characters and four status characters. Chip number, RAM register, and main memory character are addressed by an SRC instruction. For the selected chip and register, however, status character locations are selected by the instruction code (OPA).

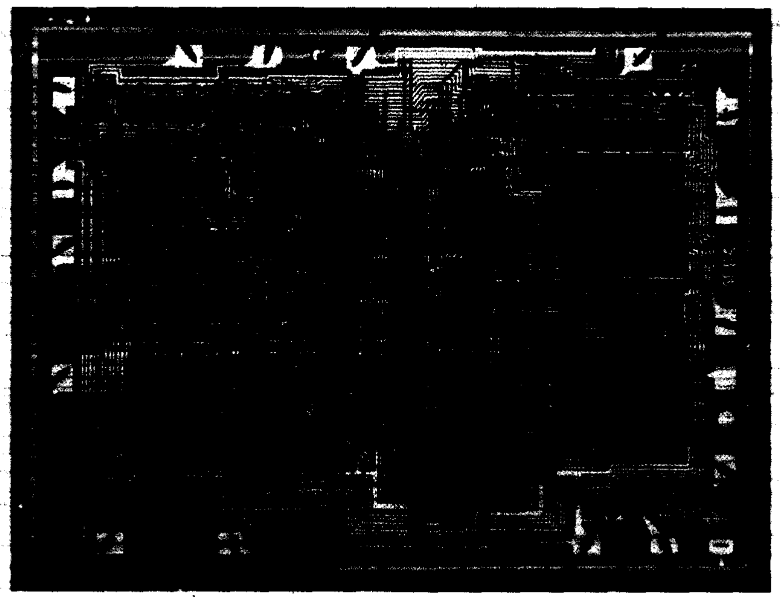
Indirect instruction and coded a 20-byte interpreter to execute 1-byte macroinstructions. Shima, the Busicom engineer in charge of programming, further refined Mazor's interpreter. In addition, Shima proposed including a conditional jump based on the status of an external input pin (test), adding an instruction that would simplify keyboard scanning, and modifying the Branch Back instruction.

It appeared at the time that a 1-MHz clock would be feasible for the processor's logic. To allow the use of small, inexpensive packages (16 leads), the design could include extensive multiplexing of interconnecting lines. Four data transfer lines would permit the processor to transfer one 4-bit quantum each clock cycle. With a 12-bit program address and an 8-bit instruction, it would take five clock cycles to address and fetch an instruction. Since most instructions would be simple, three cycles for execution seemed adequate. Those timing parameters and a 1-MHz clock would allow the processor to add multidigit BCD numbers at a rate of 80 microseconds per digit. This speed was comparable to that of the IBM 1620 computer Hoff had used in the early 1960s.

### The 4004 chip

The 4004 is the first example of a complex random-logic circuit built using silicon-gate MOS technology. Silicon gate was essential in obtaining the small size and the high speed (for the day) required by a general-purpose CPU. The chip measures 3.0 mm×4.0 mm and integrates approximately 2,300 transistors.

Under Federico Faggin's direction, three layout draftsmen drew the composite layout of the 4004 using colored lead pencils on mylar at 500 times the actual scale. The composite layout translated the abstract circuit diagram into the actual geometry of the transistors and their interconnections. Showing all the masking layers required for processing, the layout served as a template for the preparation of the "rubies." A rubylith consists of a mylar sheet with a thin layer of semitransparent, red material that can be cut and peeled off. The composite layout, placed underneath the ruby, guided the cutting and peeling operations. One ruby was prepared for each mask layer required in the wafer processing. The 4004 required six layers, including the scratch-protection layer; the other chips in the set required five. The ruby was then photoreduced to 10 times the 4004's actual scale to prepare the reticle. The reticle, in turn, was used to create the actual scale mask via a step-and-repeat optical process.



By mid September, Intel marketing was sufficiently confident of the new approach to suggest it to Busicom management as an alternate to the original design. In October 1969, Intel held a formal meeting with the Japanese firm's management, who had come to the US to discuss the project. Intel presented both approaches, with Hoff and Mazor arguing that the Intel architecture was much more flexible than the original. Busicom's managers, appreciating the architecture's increased simplicity and flexibility, chose the Intel design, and Intel became committed to build the first single-chip computer CPU. The Busicom engineers returned to Japan, except Shima. He stayed on at Intel until December to develop many of the calculator's key software programs, which he based on the new architecture and its instruction set.

When the companies signed a development contract, Hoff was disappointed to learn that, although Intel had developed the architecture and it differed markedly from the original, the contract gave exclusive rights to Busicom. Intel marketing explained that the project would not have proceeded without that concession.

Intel was now committed to develop the chips for the new architecture, but the company had a staffing problem. Neither Hoff nor Mazor had designed chips, and the proposed chips' complexity would require someone with extensive experience. Thus, the design would fall to a different department than Applications Research. Since MOS designers were in short supply, and all of those at Intel were already committed to memory projects, Intel would have to recruit someone to take over the project's logic and circuit design and the silicon implementation phase. That process would take months.

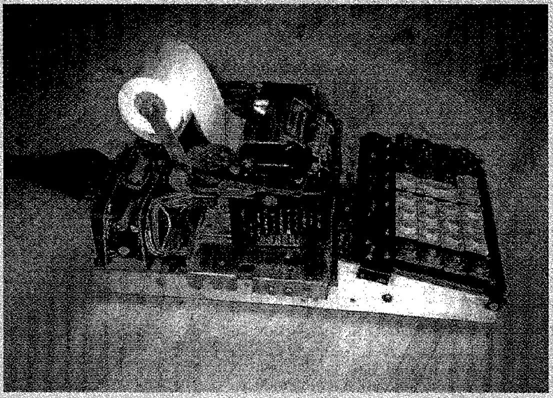
In the meantime, Hoff and Mazor had responsibility for generating applications information for the memory products that Intel was adding to its product line. One of the more successful memory products was a line of shift registers that quickly found a market in CRT (cathode-ray tube) computer terminals.

One of the customers for shift registers was Computer Terminals Corporation of San Antonio, Texas. In December 1969, an officer of that company inquired if Intel could modify an existing Intel static RAM (the I3101, a bipolar 64-bit RAM) to create a 4×16 stack memory for an intelligent terminal CTC was designing, the Datapoint 2200.

Mazor and Hoff studied the request and determined that the CTC processor did not appear much more complicated than the proposed 4004. They concluded that it would be feasible to make a single-chip, 8-bit microprocessor. They drew up a target specification, and CTC contracted with Intel for the development of what would be Intel's second microprocessor. By

### The Busicom calculator

Shown here is the engineering prototype of the Busicom calculator that opened the microprocessor application floodgates. This 14-digit, floating- and fixed-point, printing calculator, completed in April 1971, had memory and an optional square-root function. It was designed, built, and marketed by Busicom in Japan. Intel's 4000 series performed all the electronic functions of this calculator, except the discrete-transistor printer-driver circuitry, the clock generator, and miscellaneous lamp drivers. In all, the calculator used one 4004 CPU, four 4001 ROM chips, two 4002 RAM chips, and three 4003 shift register chips. (The model with the square-root function used one additional 4001.) Busicom sold this calculator worldwide beginning in July 1971.



early 1970, Intel was committed to produce two different single-chip computers, and still had no staff to do the design.

## The design of the 4004

Early in 1970, Leslie Vadasz, who headed Intel's MOS design group, announced he had found someone to do the design of the calculator chip set: Federico Faggin. Faggin worked at Fairchild, where he and Tom Klein had developed the original MOS silicon-gate process in 1968. He had also designed the first commercial circuit to use that technology (the 3708, an 8-bit analog multiplexer with decoding logic). Faggin also had experience with computer design, having codesigned and built a small computer for Olivetti in his homeland, Italy, in 1961.

Faggin joined Intel in April 1970, as the engineer in charge of the design of the calculator set. Internally called the 4000 family, the set consisted of four chips: the ROM program memory (4001), the RAM register memory (4002), an I/O expansion shift register chip (4003), and the CPU (4004). A couple of days after Faggin joined Intel, Shima arrived from Japan to check on the project's progress. Shima was very disappointed that no progress had been made since he left Intel in December 1969; according to him, the schedule for the project had been irreparably compromised. Because of this delay, Faggin began work at a furious pace, often far into the early morning hours, to make up as much of the lost time as possible. Shima stayed at Intel for six months to help Faggin with the project.

After resolving the few remaining architectural issues, Faggin laid down the foundations of the design methodology he was going to use for the chips. Random logic design with silicon-gate technology required a different methodology than metal-gate technology, and no one had ever designed a circuit of the 4004's complexity.

An important element of Faggin's methodology was its use of bootstrap loads. These circuits provided faster output voltage swings, switching to the full supply voltage instead of the supply voltage minus the transistor threshold voltage (augmented by the body effect). Bootstrap loads allowed him to use pass transistors, simplifying the circuit design and reducing the number of transistors necessary to perform the required logic functions. In those days, it was common belief that bootstrap loads were not feasible with silicon-gate technology, unless the design incorporated an additional masking step. Faggin, however, had figured out how to make bootstrap loads without modifying the process architecture. This circuit trick was essential to achieve the necessary speed and density without exceeding the power budget.

Faggin was also happy to find that Intel had adopted the "buried-contact" design. This technique, similar to the one he had developed at Fairchild, permitted direct connections between the polysilicon layer and the diffusion layer, allowing higher circuit densities. The buried contact was essential to achieve a manufacturable chip size for the 4004.

Faggin decided to design the 4001 first, followed by the 4003, the 4002, and finally the 4004. In those days, there was little automation of the design process. Although Intel had access to a time-shared mainframe computer for critical circuit simulation (via a 10-characters-per-second teletype), the company discouraged Faggin from using it because of its cost. So, Faggin did most of the circuit design with a slide rule and using graphical analysis based on measured static and dynamic transistor characteristics.

Designing a production integrated circuit took many steps, starting with the definition of the chip architecture and its basic specifications. For the 4000 set, Hoff and Mazor completed these initial steps, with contributions by Shima and the other Busicom engineers. Next came the logic design, circuit design, layout design, ruby-cutting, mask making, wafer processing, chip verification and debugging, characterization, production test-pattern development, and transfer to manufacturing. The entire process, starting from the logic design and ending with working samples, would take a minimum of six months for a simple chip, longer for a complex one.

At the peak of the project, Faggin and Shima worked simultaneously on all four chips at different stages of the development process. The 4004's detailed logic design, which Shima undertook, took place during June and July. Shima also did the logic simulation, while Faggin concentrated on the circuit design, layout, and overall supervision of the project.

## The 4004 comes to life

Intel processed the first silicon wafers of the 4001 in October 1970, and Faggin found the circuit fully functional. In preparation for receiving the chips, Shima returned to Japan to complete writing the software and to build the engineering prototype of the Busicom calculator. In November, 4003 and 4002 wafers came out of the processing line. The 4003 was fully functional, and the 4002 had a minor problem that was soon diagnosed and corrected.

Finally, at the end of December the big day arrived; Faggin received the first 4004 wafers, less than nine months after he had begun the project. Faggin's hands were trembling as he loaded the first wafer in the wafer prober to begin the test, and as he probed around the 4004, he found *absolutely no life*. He couldn't believe his eyes. Within half an hour, however—the longest half hour of his life—Faggin found that one masking step (the buried-contact layer) had been left out during wafer processing. This manufacturing problem explained why the 4004 was dead.

It wasn't until January 1971 that Intel processed a new run of 4004 wafers. Faggin received the wafers in the evening and, alone in the lab, tested them through most of the night. This time, everything worked as expected. That was the night the 4004 was born.

During the following days, Faggin continued verifying the 4004 and found two minor bugs that were soon diagnosed and corrected. As a result, he achieved fully functional 4004s on the next mask iteration, in March 1971.

After thoroughly testing the 4004, Faggin sent several samples to Busicom, where Shima was testing the calculator and debugging his software using a RAM-based emulator for the 4001.

The Busicom calculator used one 4004, two 4002, three 4003, and four 4001 chips; it used an additional 4001 for the optional square-root function. In other words, the system consisted of a 4-bit CPU running approximately 100,000 instructions per second, with 1 Kbyte of ROM, 80 bytes of RAM, and approximately 50 I/O lines. Today, using 0.35-micron lithography, the most advanced manufacturing technology in production, these functions, without the bonding pads, would occupy less than one tenth of a square millimeter. (Incidentally, the manufacturing cost would now be approximately half a cent.)

In April, word came that the Busicom calculator was fully functional. That was the final and essential proof that all the chips were working properly, individually and as a system. That same month, Shima sent Intel the final ROM patterns to generate the custom metal masks for the calculator's four 4001s. This was the last step preceding volume production, which was to start in June.

## Finishing touches

During the 4004 characterization, which began in March, Faggin observed a very disturbing phenomenon: At high temperature some 4004s were occasionally failing, but when he tested them again, they would pass. This problem was maddening, because the lack of repeatability and the lack of diagnostic tools made it very difficult to find the reason for such elusive failures. It took a few days to conclusively determine that the problem was caused by the corruption of some of the data stored in the DRAM registers. However, Faggin was at a loss to understand the mechanism responsible for it.

After a tense week of tests and analysis, however, he traced the problem to a weakness in the RAM decoder's design, which caused the injection of minority carriers in the substrate to leak away the electrical charge stored in the DRAM cell. (Intel had avoided a similar problem in its standard DRAM components by using an additional substrate bias, not desirable in the 4000 series.) Once Faggin understood the problem, he soon found a solution. Fortunately, there was enough room in the chip to make the necessary modifications to the decoder without a major redesign.

Faggin was surprised that no similar problem had ever been observed in the 4002, which had the same decoder design as the 4004. To make sure, Faggin created a special test sequence to see if the 4002 would also fail under properly adverse conditions. Indeed, the 4002 demonstrated such failures, validating the problematic-decoder hypothesis and leading Faggin to change its design in the 4002 as well. These were the last steps to ensure that the company would manufacture a quality product, averting potential problems in the field. Production could then start in earnest, and by August 1971, the 4000 series became a major source of revenue for Intel.

## Marketing the 4004

When Faggin found that the 4000 chip set was exclusive to Busicom, he was very disappointed because he saw the set's market potential reaching far beyond calculators. Though he started lobbying management to obtain the rights to sell the 4000 series to the general market, the sentiment at Intel was that the 4004 was good mostly for calculators and calculator-like products. In an effort to prove otherwise, Faggin decided to use the 4004 as the controller for the 4004 production tester he was designing. Conveniently, he was able to load the software into the new EPROM devices (electrically programmable, read-only memory, just invented at Intel by Dov Frohman-Bentchkovsky), instead of the mask-programmable 4001s. After successfully completing this project, Faggin used the example to prove to management that the 4004 was quite useful and thus marketable for applications besides calculators.

Hoff later found that the 4004 simplified the design of a unit for programming the EPROM devices while providing the ability for rapid upgrades. Because EPROM promised to be ideal for holding programs for the single-chip computers, Hoff and his group developed a circuit board containing interface circuits that would allow the EPROM to substitute for the 4001 ROM. Later, Intel developed a similar board for the 8-bit processor.

One day, talking over the phone with Shima in Japan, Faggin discovered that Busicom was having financial problems. To be more competitive in the marketplace, the firm needed lower prices for the chip set. Faggin and Hoff then pleaded with Noyce and marketing that Intel give a price concession to Busicom in exchange for nonexclusivity. By May 1971, Intel had obtained the right to sell the calculator chips to others, except for desktop calculator applications.

A brief new product announcement in *Datamation* magazine mentioned the chip series. However, even with the limited rights to sell the chips to other companies, Intel management was reluctant to announce the microprocessors officially. Marketing had deep concerns about the field sales staff's ability to properly support such complex products. Intel was developing a good reputation based on its memory products and its support of them, and marketing did not want to risk that reputation.

## Marketing the early microprocessors

*Hank Smith*

The first public announcement of a microprocessor by Intel in November 1971 was really a turning point in the history of the electronics industry and has continued to profoundly affect all our daily lives. The history of the first microprocessor design and development team—consisting of Federico Faggin, Ted Hoff, Stan Mazor, and Masatoshi Shima—is now well documented. But there was another integral part of the team whose contribution to the early success of the microprocessor has generally been ignored, and that was the marketing group. So, when Faggin asked me to describe the early days of marketing the microprocessor, I was delighted because so little has been written about this important part of microprocessor history.

Very early on, we realized that the microprocessor was very different from any other product Intel had introduced, and that we would have to market it very differently. First, most engineers were unfamiliar with programming and debugging software (particularly at the machine and assembly level), which was going to be necessary in designing systems with this product. Second, we felt that we could build a group of loyal customers because, once they designed applications using the microprocessor, their significant software investment would keep them from changing products or suppliers. Our primary objective was to get companies to design Intel's microprocessor into as many applications as possible, and as quickly as possible. To do this, we had to make it as easy as possible to use. Thus, we needed flexible, inexpensive design tools and development systems, and a group dedicated to the marketing and sales of the microprocessor. This, then, became the mission of the first Microcomputer Systems Group, formed in April of 1972.

We were pioneers. No one had ever marketed a product like this before, and we had nothing to guide us. Everything we did was a first, and we influenced the directions of the industry for many years with the design tools we introduced.

We were first with the following:

- *Microcomputer Systems Group.* For the first time a semiconductor company combined hardware engineering, software development, manufacturing, and marketing in a single marketing group.
- *Comprehensive documentation and manuals.* We marketed each microprocessor chip as part of a series (the MCS-4 and MCS-8 chip sets) and provided user's manuals and comprehensive documentation for using the products.
- *Simulation (development) boards.* The Sim4-01 and Sim8-01 were general-purpose microcomputer modules that customers could use for development, preproduction, and small production runs of microprocessor-based products.
- *PL/M high-level language.* PL/M was originally developed for the 8008 set by Gary Kildall. This language made it possible to write a program once and, by compiling it, have it run on all different kinds of 8008 and 8080 products and systems. We provided a compiler, cross assembler, and simulator, all written in Fortran IV, that could run on a general-purpose computer or one of our development systems.
- *Intellec development system.* The Intellec-4 and Intellec-8 development systems were self-contained, expandable systems complete with CPU, memory, I/O, clock, TTY interface, power supply, control and display modules, and standard software. These systems, which could be programmed in PL/M, were really the forerunners of the more sophisticated MDS development systems, and the personal computer.

No one could have forecast the microprocessor's unbelievable success, and I feel very fortunate to have been an important part of the original team responsible for the launch and success of this revolutionary product.

*After leaving Intel,* ***Hank Smith*** *spent 15 years in the venture capital industry as a general partner of Venrock Associates. He currently lives in Woodstock, Vermont, where he is an independent investor. He owns an antique car restoration business and a horse farm, and is a principal owner of the Norwich Navigators, a AA minor league baseball team affiliated with the New York Yankees.*

Another concern, one shared by Hoff and Mazor, was that customers accustomed to the power of minicomputers would be unable to adapt to the microprocessors' poorer performance. However, both felt that proper presentation would prepare customers for the limitations, and that microprocessors would still find many uses.

In the summer of 1971, major changes in Intel's marketing department brought in a new vice president of marketing—Ed Gelbach, formerly with Texas Instruments. Ed was much braver than his predecessors, and he arranged the formal announcement of the 4004 in November of 1971. Hank Smith, working for Gelbach, became the first microcomputer marketing manager.

## Market reactions

Intel had changed the chip set's name from 4000 to MCS-4, for Micro Computer System 4-bit; the response to the announcement of this first microprocessor was very encouraging. Marketing worked with Hoff, Faggin, Mazor, and Hal Feeney to provide support. The support items included data sheets with application information, user manuals, and printed circuit boards. Marketing released literature that also revealed the coming 8-bit processor, which Intel officially announced in April 1972 under the name 8008 (twice the 4004!) as the core of the MCS-8 series.

The 8008 could actually have been the world's first microprocessor. A few weeks before Intel hired Faggin to design the 4000 set, Hal Feeney had joined Intel to work on the 8-bit microprocessor for CTC. Feeney worked with Mazor and CTC to complete the specifications for the chip—internally called the 1201—and to modify the CTC architecture as necessary for silicon implementation. Financial problems at CTC, however, soon reduced the 1201's priority, so Feeney was diverted to other projects. Other potential customers kept the project alive, but the design did not proceed much past the first few months of work.

The CTC project remained dormant until January 1971, when Intel reassigned Feeney, now working under Faggin's supervision, to the project. The designers' recent experience with the 4004 provided a proven design methodology that paved the way for the 8008. Feeney did the detailed design of the 8008, and by March 1972, Intel was producing working chips.

Ironically, during 1970, CTC had also contracted with Texas Instruments to design the same processor using TI's MOS aluminum-gate process. TI's chip, heralded in the technical press in June 1971 as the first CPU on a chip, was more than twice the size of the 8008. CTC reported that it had never fully worked.

Intel promoted both the 4004 and the 8008, and in May of 1972, Hoff and Mazor presented several seminars around the country. The microprocessors generated much interest, and many of Intel's customers began to design products based on them. Of the two, the 4004 offered lower cost and a higher degree of integration for the resulting system, because the series offered RAM and ROM chips with I/O capability on the same chip. The 8008 could address a larger memory space (up to 16 Kbytes) and could use any mix of RAM or ROM for its memory. However, the 8008 required some 20 standard TTL integrated circuits to provide the interface between the processor, memory, and I/O. While the 8008 instruction cycle was actually somewhat slower than the 4004, most customers perceived it as the preferred processor for more complex applications.

The 4004 and the 8008 became archetypes for today's two primary markets for microprocessors: embedded applications and user-programmable computers. Most microprocessors used in embedded applications are now integrated with the memory and the I/O functions—true single-chip computers. Thus, a low-cost, single chip can typically do all the work required in many simple control applications. Such devices are called microcontrollers. Simple 4-bit and 8-bit microcontrollers control microwave ovens and computer keyboards, for example, while sophisticated microcontrollers drive cellular phones and laser printers.

Currently, the semiconductor industry manufactures a few billion microcontrollers worldwide per year. More than 50% of all microcontroller units manufactured in 1995 were still 4-bit devices with capabilities equivalent to those of the MCS-4 set. Nonetheless, the more expensive 8-bit microcontrollers have the majority of the market dollar volume.

The 8008's first application was a Seiko user-programmable, scientific calculator, and soon the 8008 led to the personal computer, the quintessential microprocessor application. In fact, many consider the personal computer's archetype to be the Micral, a French desktop computer using the 8008 CPU, sold in 1973. The 8008 evolved into Intel's 8080, the first high-performance microprocessor, conceived and designed by Faggin and Shima, with architectural contributions by Mazor and Hoff. This evolution has continued on to the present Pentium Pro, with a new generation, on average, every three years.

***The market directly related to the microprocessor is over $100 billion at OEM component prices. The market value of all the products incorporating microprocessors is many times that figure.***

THE PERSONAL COMPUTER has become an enormous market for microprocessors, and is considered by the popular media the microprocessor's primary use. When Intel originally announced the microprocessor, however, Faggin, Hoff, and Mazor considered its primary market to be control devices—applications now described as embedded control. While main microprocessors for personal computers do indeed represent a large market, with tens of millions of units sold each year, many more microprocessors and microcontrollers go into embedded control applications, with a typical microcontroller costing between 30¢ and $10.

From its modest beginning 25 years ago, the microprocessor industry has grown to such an extent that nearly 70% of all semiconductors sold worldwide are either microprocessors, microcontrollers, or other components used in conjunction with them, such as memory and I/O devices. Since the worldwide sales of semiconductor components in 1995 was approximately $150 billion, this means that the market directly related to the microprocessor is over $100 billion at OEM component prices. The market value of all the products incorporating microprocessors is many times that figure, of course—a truly staggering amount.

Over the last 25 years, there has been an explosion of applications. People carry microprocessors with them inside their watches, pocket calculators, organizers, and cellular phones; and microprocessors are all around them, in their homes, cars, offices, and laboratories. The microprocessor has improved the quality, cost, and functionality of traditional electronic equipment. But, most importantly, it has enabled literally thousands of new applications impossible before its advent. Amazingly, the pace of deployment of microprocessors and microcontrollers in new applications is still going strong, and we expect it to continue for the foreseeable future. Without question, the microprocessor reality has far exceeded even the most bullish expectations of its creators.

## Suggested readings

Sack, E.A., R.C. Lyman, and G.Y. Chang, "Evolution of the Concept of a Computer on a Slice," *Proc. IEEE,* IEEE, Piscataway, N.J., 1964. The authors project chips of 100 gates or more as being feasible, and predict appreciable computer subfunctions on a single slice.

Flynn, M.J., "Complex IC Arrays: The Promise and the Problems," *Electronics,* July 11, 1966. The author predicts 1,000 gates per chip and hypothesizes the one-chip computer and array processors.

Beelitz, H.R., and H.S. Miller, "Partitioning for Large-Scale Integration," *Proc. Int'l Solid-State Circuits Conf.,* IEEE Computer Soc., Los Alamitos, Calif., 1967. Discusses partitioning LSI for a large computer to improve gate/pin ratios.

Bairstow, J.N., "LSI Will Demand New Computer Architecture," *Electronic Design,* Jan. 4, 1968. Performance will improve, but specialists see no significant cost reduction for the user.

Faggin, F., and T. Klein, "A Faster Generation of MOS Devices with Low Threshold Is Riding the Crest of the New Wave, Silicon-Gate IC's," *Electronics,* Sept. 29, 1969. Describes the original MOS silicon-gate technology developed at Fairchild and the world's first commercial integrated circuit to use it, the Fairchild 3708, an 8-bit analog multiplexer with decoding logic.

Narud, J.A., C.D. Phillips, and W.C. Seelbach, "Complex Monolithic Arrays: Some Aspects of Design and Fabrication," *Microelectronics,* Jul. 1969. "Integration at the system level... too terrifying even to think about."

Hoff, M.E., "Impact of LSI on Future Minicomputers," *Proc. IEEE Int'l Conv.,* IEEE, 1970. States the feasibility of small, single-chip processors.

Hoff, M., S. Mazor, and F. Faggin, *Memory System for Multi-Chip Digital Computer,* US patent 3,821,715, to Intel Corp., June 28, 1974. Covers the novel, and thus patentable, architectural features of the 4000 chip series.

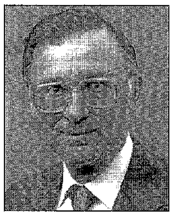
Faggin, F., *Power Supply Settable Bi-Stable Circuit,* US patent 3,753,011, to Intel Corp., Aug. 14, 1973. Covers a special circuit that provides the power-on reset function in the 4000 series.

Faggin, F., and M. Hoff, "Standard Parts and Custom Design Merge in Four-Chip Processor Kit," *Electronics,* Apr. 24, 1972, pp. 112-116. The first published article to describe the 4000 series.

Faggin, F., et al., "The MCS-4—An LSI Microcomputer System," *Proc. IEEE Region Six Conf.,* IEEE , 1972. Describes the 4000 series.

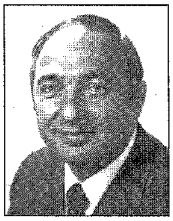
Faggin, F., M. Shima, and S. Mazor, *Computer Employing a Plurality of Separate Chips,* US patent 4,010,499 to Intel Corp., Mar. 1, 1977. Covers the novel architectural features of the 8080 CPU.

**Federico Faggin**'s photograph and biography appear on page 9.



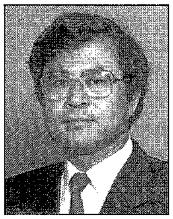
**Marcian E. (Ted) Hoff Jr.** is a design consultant for Teklicon Inc. and assists attorneys dealing with patent litigation. Formerly, he was vice president of corporate technology at Atari, Inc., Intel's manager of applications research, and the first Intel Fellow. He holds 17 US patents, including those for an electrochemical memory, the microprocessor, and the monolithic telephone codec.

Hoff holds a BEE degree from Rensselaer Polytechnic Institute and MS and PhD degrees in electrical engineering from Stanford University. For his role in the invention of the microprocessor, he has received numerous awards and was inducted into the Inventors Hall of Fame. An IEEE Fellow, Hoff received that organization's Centennial Medal. The Computer Society has recognized him as a computer pioneer.



**Stanley Mazor** is training director at BEA Systems. Formerly, he worked for Fairchild Semiconductor, an experience from which he shares patents on the Symbol computer. At Intel, after his work on the 4000 series (for which he shares patents), he proposed the 8008 CPU chip and later specified the 8080 CPU chip. He began his teaching career in Intel's Technical Training Group, and later taught classes at Stanford University, the University of Santa Clara, KTH in Stockholm, and Stellenbosh, S.A.

Mazor studied mathematics and programming at San Francisco State University. He has coauthored a book on a chip design language, *A Guide to VHDL,* and published 45 articles and papers on VLSI chips and design.



**Masatoshi Shima** is chair of VM Technology Inc. in Japan. Formerly, he was manager of the Intel Japan Design Center. While an employee at Busicom Corp. in Japan, he developed the desktop calculator, for which he introduced ROM-based stored programming technology with decimal-plus-binary computer architecture. After working on the development of the 4000 series products, he joined Intel, where he developed the 8080 and several peripheral chips as supervising engineer. As manager of high-end microprocessors at Zilog, he developed the Z80 and Z8000.

Shima received a BS in chemistry from Tohoku University and a DrEng from Tsukuba University, Japan.

Direct questions concerning this article to Federico Faggin, Synaptics, Inc., 2698 Orchard Pkwy., San Jose, CA 95134; federico@synaptics.com.

## Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 153 Medium 154 High 155