Intel 8080 CPU Chip Development

Stanley Mazor *Editor: Laurie Robertson*

Within a year of the introduction of Intel's first 8-bit CPU chip, the 8008, in April 1972 I was working on the design specification for a faster and more capable CPU chip, the 8080. Here, I describe the development of the 8080 chip that helped launch the personal computer industry. I will also address two common complaints about the 8080's architecture—its lack of register symmetry and the absence of an index register.

Background

Intel had declared its corporate objective to be "the most profitable semiconductor company, ever." To meet that objective, we focused on chips commanding higher selling prices rather than lower end "commodity" chips such as TTL (transistor-transistor logic) gates. When Intel started, our only business was memory chips.[1] We sold them in high volume to a few customers (which required only a small engineering staff and a small sales force) to meet our profit goal.

As a memory chip company, we used IC packages with few (16-18) leads, but with a lot of transistors in the chip: this represented a high gate-to-pin ratio. (It's noteworthy that if you double the number of bits within a shift register memory chip, its package leads are unaffected. If you double the number of bits in a RAM memory chip, you only need to add one more address pin.) So, memory chips efficiently manage the number of pins on the package. Saving pins lowered our costs and improved our profits. However, these memory chip packages were the only ones available to the Intel engineers and limited the number of input and output signals we could have on other types of chips such as the 8008 CPU chip.

Intel's strategy of pioneering the memory marketplace was risky. The adoption in the marketplace of RAM memory chips as a replacement for core memories was slow. As a consequence of slow sales of RAM chips (our standard products), we accepted a couple of custom chip projects to augment our standard part business. The development of the first 8-bit CPU chip at Intel was for Datapoint Corp.[2] Although it was a "failed" custom project (ultimately, Datapoint never used it), the 8008 was announced as a standard Intel product in April 1972.[3] We realized that selling CPU chips promoted memory chip sales, since at least 8 RAM chips were needed for every 8-bit CPU chip. Furthermore, users' main memory needs were always increasing because of application software demands.

As a technology-driven company, Intel had a good understanding of the memory chip market. The goal was to lower the cost per bit and improve data access times. During my 15 years at Intel, we often adopted newer semiconductor technologies to get higher chip densities that benefited these two market objectives. For example, during the two-year development period of the 8008 CPU chip, memory densities increased by a factor of four and speeds doubled, in accordance with Moore's law.[4]

A new CPU

Federico Faggin, who ran the ("small machine") non-memory chip design group at Intel, wanted to make better CPU chips. He hoped to double the performance of the 8008 by using a newer, n-type metal-oxide semiconductor (NMOS) process. However, the existing design masks couldn't be used without some re-engineering. Reluctantly, Intel management agreed, and authorized Faggin to make some "modest" improvements in the 8008 while re-laying out the new mask set. We were thereby given license to revisit the 8008 processor's instruction set and specification.

I understood the 8008's limitations and was interested in making improvements in the next-version CPU chip design. I'd programmed the 8008 myself, and also served as liaison to several contract programmers who developed software for our CPU product line. As an applications engineer, my job was to find new ways of using our technologies, to look for market opportunities, and to get customer feedback on our current products. Accordingly, I was in contact with some of our 8008 customers.

It so happened that one of the early Datapoint engineers, Harry Pyle, worked on the architectural specification of their 2200 computer and consequently on the 8008 CPU instruction set. Harry, who worked at Datapoint's Texas headquarters, and I regularly commu-

Editor's Note: Laurie Robertson

It is with sadness and much regret that we report the death of our friend and colleague, Laurie Robertson, who was the editor of the Anecdotes department. Laurie was one of those rare individuals who was both a practitioner of computer science and a student attending school. She was an employee of the MITRE Corporation and was about to take her doctoral qualifying exam in the Science, Technology, and Society Program at Virginia Tech. Because of this background, Laurie was especially good in working with Anecdote writers—in helping to shape a story, guide an argument, or find a gem of an idea that needed to be shared with the world. We will miss her.

Laurie's friend and editorial board member, Anne Fitzpatrick, has agreed to serve as an interim editor of this department. Current and potential authors of Anecdotes should correspond with her at the department address, annals-anecdotes@computer.org.

 Published by the IEEE Computer Society 1058-6180/07/$25.00

nicated by telephone to discuss architectural issues and enhancements. As the 2200 was built with TTL, it had a slightly different instruction set than the 8008; over time, the Intel and Datapoint's processors started to diverge. (I believe that a later model of the Datapoint computer system did use a Zilog Z-80.) As ''computer enthusiasts'' Harry and I spoke without concerns about patents, trade secrets, or competitive constraints, so I was also able to consider Datapoint's enhancements to their instruction set even though they weren't using our 8008 chip.

8008 issues

The primary weaknesses of the 8008 were its few package pins, its limited memory addressing modes, and its small on-chip stack.

18-pin package

The 8008's 18-pin package, (used for Intel's 1103 DRAM chip) reduced our manufacturing costs of this CPU chip, but customer systems containing the 8008 required a handful of chips surrounding it to interface to memory systems. The costs of these extra chips, the added assembly costs, and printed circuit board space for these chips probably were a net deficit compared with the cost savings of the cheaper CPU package itself. Additionally, having only 18 pins to bring out a 14-bit address and an 8-bit data bus required time division multiplexing of the 18 pins, and slowed the processor. However, by the mid-1970s, calculator chips were coming into high-volume production, and there was an abundance of 40-pin packages available that would benefit a single-chip CPU if it were redesigned with a bigger package.

Addressing modes

As a custom chip, the 8008 instruction set was 95 percent specified by Datapoint. My boss at Intel, Ted Hoff, added register Increment and Decrement instructions, but otherwise we accepted the instruction set more or less intact from Datapoint. The 8008 only had one way of getting to main memory—indirect addressing through the HL (High/Low) register pair, which was slow and cumbersome. The 8008 had neither index registers nor 16-bit address arithmetic.

PC stack

Both our 4004 (the first single-chip 4-bit CPU) and 8008 CPUs supported nested subroutine calls by saving return addresses in an internal stack memory within the CPU chip. This is useful when external main memory is ROM or Shift Register. But such an on-chip stack uses valuable CPU chip real estate, however, and its fixed size reduces the programmer's flexibility. The 4004 stack was 4-deep and the 8008's was 8-deep (for remembering return addresses). However, 8008-based systems usually had substantial RAM memory, and an improvement would be to delete the on-chip stack memory and instead use part of the computer's main memory as a last-in, first-out stack. (Later I would enjoy the irony of marketing this chip to our customers with the stack facility deleted from the chip, but touting it as a benefit—''Now the stack can be as large as you want.'')

Super 8: Highlights and constraints

As the chip architect of the 8008 and while still working for Ted Hoff, in the Applications Research Department, I started to evaluate improvements to the 8008. I named the 8080 project after the then-popular home movie format—Super 8.

Now, the dominant factors in IC chip design are the chip's size (transistor count), the power used, and the number of I/O pins on the package. Any CPU chip designer must consider these factors when defining the CPU's architecture and instruction set; these constraints determine the trade-offs that must be made.[5] The 8008 used about 2,000 transistors, and the follow-on (8080) chip would have a budget of about 5,000 transistors assuming Moore's law and using the latest NMOS process. We could move up to a 40-pin package and our power budget was under 1 watt.

A main-memory-based stack needed an additional stack pointer (SP) register within the CPU chip, and I had to determine the stack orientation in memory. Because most programming systems allocate the lower part of memory for the start-up/boot software, I imagined the stack at the high end of memory and running ''downhill''—in this way, memory was filled from both top and bottom ends.

Stacks weren't a new concept; I had used them in several computers, and I knew the memory stack would also be useful to a programmer for saving the CPU's registers. Saving such data was prohibitive with an on-chip stack but feasible with a stack in main memory. Push and Pop instructions were needed for each of the three register pairs. We also needed an instruction to load this new SP register and to read out its value. So, I immediately faced the issue of adding new operation codes to the 8008 instruction set.

Instruction compatibility

Adding new instruction op-codes meant breaking with strict machine code compatibility with the earlier 8008. Because I had done a lot of programming over the years, I knew that reassembling old 8008 source code for the new chip's instruction set was an automatic conversion solution. As our customer base and software pool was somewhat small, I favored making a major upgrade in the instruction set and taking a "hit" on machine code compatibility. Consequently, that's what we decided to do.

Symmetry

In 1965, while working as a programmer at Fairchild Semiconductor, I had programmed the IBM 360 and noted another phenomenon. Although the IBM 360 hardware designers created 16 identical registers within the CPU, the operating system programmers had specialized some of these registers for interrupt handling and OS communications. Hence a programmer didn't see them as a fully symmetric set anyway. A feature of the original 8008 was its symmetric use of the CPU's 6 registers: B,C,D,E,H,L (the HL register pair was used as the High/Low memory address). Because I had observed that programmers often specialize the register usage, I decided to "warp" the hardware architecture. Therefore, in our new CPU, I organized the registers into three tiered pairs: HL, DE, BC. HL would be the most capable, DE would be almost as capable, and BC would be the least useful. Later, the 8080 was often maligned for this lack of symmetry that I had implemented, which I'd done intentionally as a trade-off.

A specific example of a trade-off: at a hardware cost of just one flip/flop, the HL and DE register pairs could be swapped. (This was essentially a renaming trick that Hoff came up with; the data never moves.) To provide register swaps across all three register pairs would be desirable; however, it would have required about six times more hardware.

Index registers

The 8080 was also castigated for not having index registers—an intentional decision I had made, because I favored general-purpose 16-bit operations instead. By 1972, I had programmed about a dozen different computers. A characteristic of most fixed-word-length machines is their use of index registers for array addressing. However, in the older multibyte machines (such as the IBM 1620, the IBM 1401, and so on), programmers typically coded explicit-address arithmetic and indirect addressing. Because the 8008 was already using the HL register pair for the programmer's indirect addressing of data, I favored general-purpose 16-bit arithmetic (DE + HL → HL) rather than building in address arithmetic (indexing) hardware only. In fact, when completed, the new CPU chip provided about 18 new 16-bit data operations on the three register pairs—including 16-bit multiply and divide steps. (Interestingly, the competing Motorola 6800 did have an index register, but didn't provide 16-bit arithmetic to the user.)

Moving forward

In all, there were at least four different versions of the 8080 instruction sets proposed. For each of these, we considered the amount of engineering time required and the project's time budget. The final 8080 specification overcame most of the problems of the earlier 8008, and included the following:

- decimal arithmetic
- 16-bit data operation
- more addressing modes
- better/bigger external stack
- interrupt support
- chip hardware interfacing

On 8 September 1972, a proposal was circulated for management approval, with engineering cosigners: Federico Faggin, Marcian (Ted) Hoff, Stan Mazor, and Ben Warren. About a year later, Hal Feeney, an Intel marketing engineer, who had designed the original 8008 chip, conducted a customer survey concerning the new (8080) chip's specification. Nearly 100 customers responded, with 86 customers finding it adequate for their needs. Hank Smith, the marketing manager for microcomputers, forwarded the following customer's letter to me:

> It is with great pleasure that we noted that almost all of the objections we had to the instruction set of the 8008 have disappeared from the 8080. We are quite excited by the 8080 CPU and … it can be the basis of a system of great generality and power, and are looking forward to building and selling such a system.

Real chip design

Now we had real design work to do. Intel's first CPU chip, the 4004, had been a custom chip developed for Busicom of Japan.[6,7] Fag-

gin, the chip's design engineer, was assisted by our customer/engineer Masatoshi Shima. Shima had checked the logic and circuit details of the 4004 IC mask artwork. Although he had no IC layout experience, he quickly learned how to compare the circuit schematic against the layout, and was an asset to Faggin during this development of this first microprocessor chip.[8,9] Consequently, Faggin recruited Shima from Japan to join Intel to do the detailed circuit and layout of the 8080 chip. In early 1974, Shima completed the 8080; it ran 10 times faster than the 8008 and was a clear market winner.[10–12] A tripling of the clock speed from the better semiconductor process was to be expected, and a doubling of the performance from the added pin count bandwidth—the remaining gain was from improvements in the instruction set and architecture.

Conclusion

During the early years of Intel, there was a tight coupling between the engineering and applications research group that enabled rapid product specification and development. With a group of just 3-4 participants, engineering projects were completed with minimum interaction (interference) from management or marketing. The 8080 was a project that took the product line forward and influenced the development of Intel as a company and the microcomputer as a product. Dozens of customer systems such as the Altair and Imsai computers were soon announced using our 8080 as the CPU. Motorola followed the 8080 in 1974 with the M6800, and the microcomputer industry truly got its start. Shima and Faggin went on to found Zilog and produced the famed Z-80. By December 1975, about 3,000 small ''personal'' computers had been sold. In early 1976, the MOS Technology 6502 was introduced, and computer systems came out from Apple, Radio Shack, and Commodore.

Acknowledgments

I thank Marcian (Ted) Hoff, Federico Faggin, Masatoshi Shima, Gordon Moore, Robert Noyce, Hal Feeney, and Les Vasdasz for their original work and cooperation. See additional references 13-17.

References and notes

1. G. Bylinsky, ''Little Chips Invade the Memory Market,'' *Fortune*, Apr. 1971, pp. 100-104.
2. S. Mazor, ''8 Bits of Irony,'' *IEEE Annals of the History of Computing,* vol. 28, no. 2, 2006, pp. 73-76.
3. Intel, *MCS-8 User Manual*, 1972.
4. R. Noyce, ''A Look at Future Costs of Integrated Arrays,'' *Proc. Fall Joint Computer Conf.*, AFIPS Press,1966, pp. 111-115.
5. S. Morse et al., ''Intel Microprocessors 8008 to 8086,'' *Computer*, Oct. 1980, pp. 42-60.
6. M.E. Hoff, S. Mazor, and F. Faggin, *Memory System for a Multi-Chip Digital Computer*, US patent 3,821,715, Intel Corp., Patent and Trademark Office, June 1974.
7. Intel, ''MCS-4 Micro-Computer Set,'' data sheet 7144, 1971.
8. F. Faggin et al., ''The MCS-4: An LSI Micro Computer System,'' *Proc. IEEE Region 6 Conf.*, IEEE Press,1972, pp. 8-11.
9. M. Shima, *The Birth of the Microcomputer: My Recollections*, Iwanami Shoten, 1987.
10. M. Shima, F. Faggin, and S. Mazor, ''An N-Channel 8-Bit Single Chip Microprocessor,'' *Proc. IEEE Int'l Solid State Circuits Conf.* (ISSCC), IEEE Computer Soc. Press, Feb. 1974, pp. 56-57.
11. Intel, *MCS-80 User Manual*, 1974.
12. F. Faggin, M. Shima, and S. Mazor, *MOS Computer*, US patent 4,010,449, to Intel Corp., Patent and Trademark Office, Mar. 1977.
13. L. Vasdasz et al., ''Silicon Gate Technology,'' *IEEE Spectrum*, Oct. 1969, pp. 27-35.
14. R. Noyce and M. Hoff, ''A History of Microprocessor Development at Intel,'' *IEEE Micro,* vol. 1, no. 1, 1981, pp. 8-21.
15. S. Mazor, ''The History of the Microcomputer,'' *Readings in Computer Architecture*, M.D. Hill, N.P. Jouppi, and G.S. Sohi, eds., Morgan Kaufmann, 2000, pp. 60-67.
16. S. Mazor, ''Micro to Mainframe,'' *IEEE Annals of the History of Computing,* vol. 27, no. 1, 2005, pp. 82-84.
17. S. Mazor, ''Programming and/or Logic Design,'' *Proc. IEEE Computer Group Conf.*, IEEE Press, 1968, pp. 69-71.

Contact Stanley Mazor about this article at stanmazor@sbcglobal.net.

Contact the department editor, Anne Fitzpatrick, at annals-anecdotes@computer.org.