4.1 Chapter Overview

This chapter discusses history of the 80x86 CPU family and the major improvements occuring along the line. The historical background will help you better understand the design compromises they made as well as understand the legacy issues surrounding the CPU s design. This chapter also discusses the major advances in computer architecture that Intel employed while improving the x86[1].

4.2 The History of the 80x86 CPU Family

Intel developed and delivered the first commercially viable microprocessor way back in the early 1970 s: the 4004 and 4040 devices. These four-bit microprocessors, intended for use in calculators, had very little power. Nevertheless, they demonstrated the future potential of the microprocessor — an entire CPU on a single piece of silicon[2]. Intel rapidly followed their four-bit offerings with their 8008 and 8080 eight-bit CPUs. A small outfit in Santa Fe, New Mexico, incorporated the 8080 CPU into a box they called the Altair 8800. Although this was not the world s first "personal computer" (there were some limited distribution machines built around the 8008 prior to this), the Altair was the device that sparked the imaginations of hobbyists the world over and the personal computer revolution was born.

Intel soon had competition from Motorola, MOS Technology, and an upstart company formed by disgrunteled Intel employees, Zilog. To compete, Intel produced the 8085 microprocessor. To the software engineer, the 8085 was essentially the same as the 8080. However, the 8085 had lots of hardware improvements that made it easier to design into a circuit. Unfortunately, from a software perspective the other manufacturer s offerings were better. Motorola s 6800 series was easier to program, MOS Technologies 65xx family was easier to program and very inexpensive, and Zilog s Z80 chip was upwards compatible with the 8080 with lots of additional instructions and other features. By 1978 most personal computers were using the 6502 or Z80 chips, not the Intel offerings.

Sometime between 1976 and 1978 Intel decided that they needed to leap-frog the competition and produce a 16-bit microprocessor that offered substantially more power than their competitor s eight-bit offerings. This initiative led to the design of the 8086 microprocessor. The 8086 microprocessor was not the world s first 16-bit microprocessor (there were some oddball 16-bit microprocessors prior to this point) but it was certainly the highest performance single-chip 16-bit microprocessor when it was first introduced.

During the design timeframe of the 8086 memory was very expensive. Sixteen Kilobytes of RAM was selling above $200 at the time. One problem with a 16-bit CPU is that programs tend to consume more memory than their counterparts on an eight-bit CPU. Intel, ever cogniscent of the fact that designers would reject their CPU if the total system cost was too high, made a special effort to design an instruction set that had a high memory density (that is, packed as many instructions into as little RAM as possible). Intel achieved their design goal and programs written for the 8086 were comparable in size to code running on eight-bit microprocessors. However, those design decisions still haunt us today as you ll soon see.

1. Note that Intel wasn t the inventor of most of these new technological advances. They simply duplicated research long since commercially employed by mainframe designers.
2. Prior to this point, commerical computer systems used multiple semiconductor devices to implement the CPU.

At the time Intel designed the 8086 CPU the average lifetime of a CPU was only a couple of years. Their experiences with the 4004, 4040, 8008, 8080, and 8085 taught them that designers would quickly ditch the old technology in favor of the new technology as long as the new stuff was radically better. So Intel designed the 8086 assuming that whatever compromises they made in order to achieve a high instruction density would be fixed in newer chips. Based on their experience, this was a reasonable assumption.

Intel s competitors were not standing still. Zilog created their own 16-bit processor that they called the Z8000, Motorola created the 68000, their own 16-bit processor, and National Semicondutor introduced the 16032 device (later to be renamed the 32016). The designers of these chips had different design goals than Intel. Primarily, they were more interested in providing a reasonable instruction set for programmers even if their code density wasn t anywhere near as high as the 8086. The Motorola and National offers even provided 32-bit integer registers, making programming the chips even easier. All in all, these chips were much better (from a software development standpoint) than the Intel chip.

Intel wasn t resting on its laurels with the 8086. Immediately after the release of the 8086 they created an eight-bit version, the 8088. The purpose of this chip was to reduce system cost (since a minimal system could get by with half the memory chips and cheaper peripherals since the 8088 had an eight-bit data bus). In the very early 1980 s, Intel also began work on their intended successor to the 8086 — the iAPX432 CPU. Intel fully expected the 8086 and 8088 to die away and that system designers who were creating general purpose computer systems would choose the 432 chip instead.

Then a major event occurred that would forever change history: in 1980 a small group at IBM got the go-ahead to create a "personal computer" along the likes of the Apple II and TRS-80 computers (the most popular PCs at the time). IBM s engineers probably evaluated lots of different CPUs and system designs. Ultimately, they settled on the 8088 chip. Most likely they chose this chip because they could create a minimal system with only 16 Kilobytes of RAM and a set of cheap eight-bit peripheral devices. So Intel s design goals of creating CPUs that worked well in low-cost systems landed them a very big "design win" from IBM.

Intel was still hard at work on the (ill-fated) iAPX432 project, but a funny thing happened — IBM PCs started selling far better than anyone had ever dreamed. As the popularity of the IBM PCs increased (and as people began "cloning" the PC), lots of software developers began writing software for the 8088 (and 8086) CPU, mostly in assembly language. In the meantime, Intel was pushing their iAPX432 with the Ada programming language (which was supposed to be the next big thing after Pascal, a popular language at the time). Unfortunately for Intel, no one was interested in the 432. Their PC software, written mostly in assembly language wouldn t run on the 432 and the 432 was notoriously slow. It took a while, but the iAPX432 project eventually died off completely and remains a black spot on Intel s record to this day.

Intel wasn t sitting pretty on the 8086 and 8088 CPUs, however. In the late 1970 s and early 1980 s they developed the 80186 and 80188 CPUs. These CPUs, unlike their previous CPU offerings, were fully upwards compatible with the 8086 and 8088 CPUs. In the past, whenever Intel produced a new CPU it did not necessarily run the programs written for the previous processors. For example, the 8086 did not run 8080 software and the 8080 did not run 4040 software. Intel, recognizing that there was a tremendous investment in 8086 software, decided to create an upgrade to the 8086 that was superior (both in terms of hardware capability and with respect to the software it would execute). Although the 80186 did not find its way into many PCs, it was a very popular chip in embedded applications (i.e., non-computer devices that use a CPU to control their functions). Indeed, variants of the 80186 are in common use even today.

The unexpected popularity of the IBM PC created a problem for Intel. This popularity obliterated the assumption that designers would be willing to switch to a better chip when such a chip arrived, even if it meant rewriting their software. Unfortunately, IBM and tens of thousands of software developers weren t willing to do this to make life easy for Intel. They wanted to stick with the 8086 software they d written but they also wanted something a little better than the 8086. If they were going to be forced into jumping ship to a new CPU, the Motorola, Zilog, and National offerings were starting to look pretty good. So Intel did something that saved their

bacon and has infuriated computer architects ever since: they started creating upwards compatible CPUs that continued to execute programs written for previous members of their growing CPU family while adding new features.

As noted earlier, memory was very expensive when Intel first designed the 8086 CPU. At that time, computer systems with a megabyte of memory usually cost megabucks. Intel was expecting a typical computer system employing the 8086 to have somewhere between 4 Kilobytes and 64 Kilobytes of memory. So when they designed in a one megabyte limitation, they figured no one would ever install that much memory in a system. Of course, by 1983 people were still using 8086 and 8088 CPUs in their systems and memory prices had dropped to the point where it was very common to install 640 Kilobytes of memory on a PC (the IBM PC design effectively limited the amount of RAM to 640 Kilobytes even though the 8086 was capable of addressing one megabyte). By this time software developers were starting to write more sophisticated programs and users were starting to use these programs in more sophisticated ways. The bottom line was that everyone was bumping up against the one megabyte limit of the 8086. Despite the investment in existing software, Intel was about to lose their cash cow if they didn t do something about the memory addressing limitations of their 8086 family (the 68000 and 32016 CPUs could address up to 16 Megbytes at the time and many system designers [e.g., Apple] were defecting to these other chips). So Intel introduced the 80286 which was a big improvement over the previous CPUs. The 80286 added lots of new instructions to make programming a whole lot easier and they added a new "protected" mode of operation that allowed access to as much as 16 megabytes of memory. They also improved the internal operation of the CPU and bumped up the clock frequency so that the 80286 ran about 10 times faster than the 8088 in IBM PC systems.

IBM introduced the 80286 in their IBM PC/AT (AT = "advanced technology"). This change proved enourmously popular. PC/AT clones based on the 80286 started appearing everywhere and Intel s financial future was assured.

Realizing that the 80x86 (x = "", "1", or "2") family was a big money maker, Intel immediately began the process of designing new chips that continued to execute the old code while improving performance and adding new features. Intel was still playing catch-up with their competitors in the CPU arena with respect to features, but they were definitely the king of the hill with respect to CPUs installed in PCs. One significant difference between Intel s chips and many of their competitors was that their competitors (noteably Motorola and National) had a 32-bit internal architecture while the 80x86 family was stuck at 16-bits. Again, concerned that people would eventually switch to the 32-bit devices their competitors offered, Intel upgraded the 80x86 family to 32 bits by adding the 80386 to the product line.

The 80386 was truly a remarkable chip. It maintained almost complete compatibility with the previous 16-bit CPUs while fixing most of the real complaints people had with those older chips. In addition to supporting 32-bit computing, the 80386 also bumped up the maximum addressablility to four gigabytes as well as solving some problems with the "segmented" organization of the previous chips (a big complaint by software developers at the time). The 80386 also represented the most radical change to ever occur in the 80x86 family. Intel more than doubled the total number of instructions, added new memory management facilities, added hardware debugging support for software, and introduced many other features. Continuing the trend they set with the 80286, the 80386 executed instructions faster than previous generation chips, even when running at the same clock speed plus the new chip ran at a higher clock speed than the previous generation chips. Therefore, it ran existing 8088 and 80286 programs faster than on these older chips. Unfortunately, while people adopted the new chip for its higher performance, they didn t write new software to take advantage of the chip s new features. But more on that in a moment.

Although the 80386 represented the most radical change in the 80x86 architecture from the programmer s view, Intel wasn t done wringing all the performance out of the x86 family. By the time the 80386 appeared, computer architects were making a big noise about the so-called RISC (Reduced Instruction Set Computer) CPUs. While there were several advantages to these new RISC chips, a important advantage of these chips is