Special Feature

PC ENGINES - FROM 4004 to PENTIUM III

By ERNEST FLINT

Tracking the Intel microprocessor to its lair!

I'll see your Pentium II and Raise you Two Celerons!

In the not-so-distant past, very few people had the faintest clue as to which type of microprocessor was ensconced in their personal computer (PC). Instead, in the early days, consumers purchased computers by type, such as an IBM PC-AT or PC-XT. Even for those who did have a clue what was "under the hood", the playing field was fairly easy to understand, as things tended to progress in a fairly leisurely and obvious manner. Each new generation was faster and better than the one before and, more importantly, each new generation quickly *replaced* the previous one "on the store shelves".

By comparison, it's now hard to turn on the television without being accosted with yet another "Intel Inside" advert. In fact, those little rapscallions at Intel are rolling out new types of microprocessors so fast it makes one's head spin. Even worse, there are now multiple concurrent architectures targeted at different markets.

It was bad enough when all we had to contend with was the choice between a Pentium and a Pentium with MMX. Suddenly, as if from nowhere, we were being barraged with the additional options of Celerons, Pentium IIs, and Pentium II Xeons. And now we are staring Pentium IIIs and Pentium III Xeons in the face, desperately trying not to be the first to blink.

In fact there are now so many flavors of microprocessor on the loose (each of which is available in a range of clock frequencies), that even those of us who are "in the trade" are beginning to lose whatever tenuous grip we once had on reality. So in this article we will rend the veils asunder and expose the horrors within.

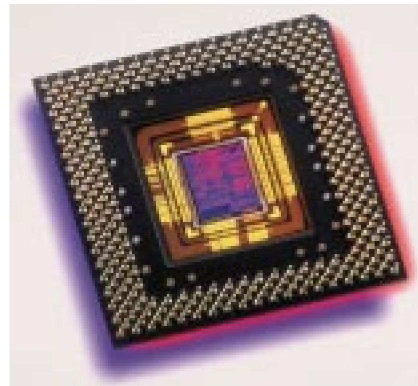


Fig.2. The Pentium processor in a Socket 7 configuration. (Photo Courtesy of Intel Corp.)

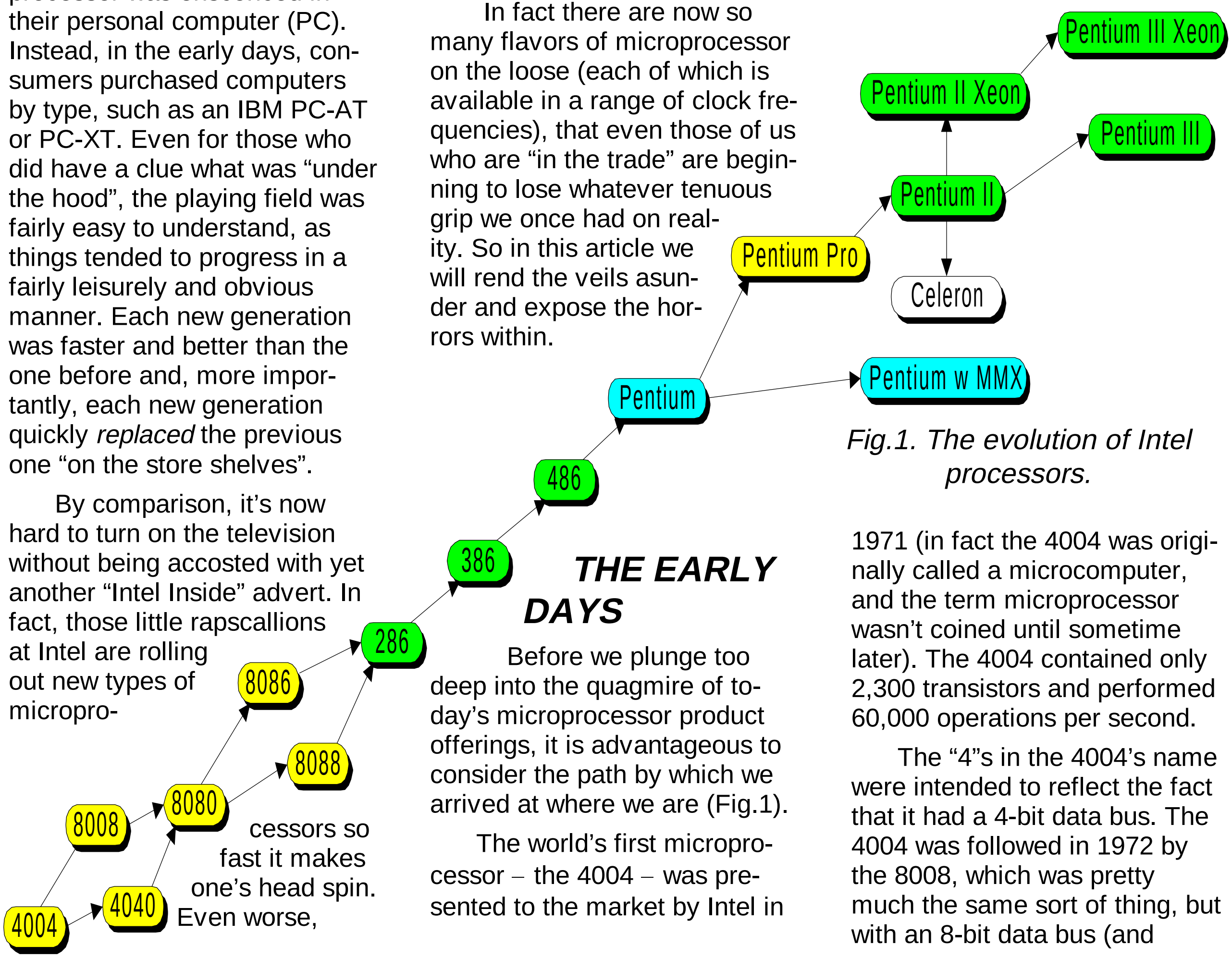


Fig.1. The evolution of Intel processors.

THE EARLY DAYS

Before we plunge too deep into the quagmire of today's microprocessor product offerings, it is advantageous to consider the path by which we arrived at where we are (Fig.1).

The world's first microprocessor – the 4004 – was presented to the market by Intel in 1971 (in fact the 4004 was originally called a microcomputer, and the term microprocessor wasn't coined until sometime later). The 4004 contained only 2,300 transistors and performed 60,000 operations per second.

The "4"s in the 4004's name were intended to reflect the fact that it had a 4-bit data bus. The 4004 was followed in 1972 by the 8008, which was pretty much the same sort of thing, but with an 8-bit data bus (and

3,300 transistors). The 4004 was also enhanced to form the 4040, which contained additional logical and compare instructions and a small internal stack.

The 4004, 4040, and 8008 were all interesting, but they were also all designed with specific applications in mind, and it wasn't until 1974 that the 4040 and 8008 evolved into the 8080, which contained 4,500 transistors and could perform 200,000 operations per second. The 8080 was the first truly general-purpose microprocessor, and was destined to become the central processor for many of the early home computers.

By 1978, the 8080 had evolved into the more sophisticated 8-bit 8088 and the 16-bit 8086. The 8088 was to become pivotal in the history of microprocessors as we know it, because IBM decided to use this as the CPU in what is now considered to be the first true PC, which was presented to the market in 1981.

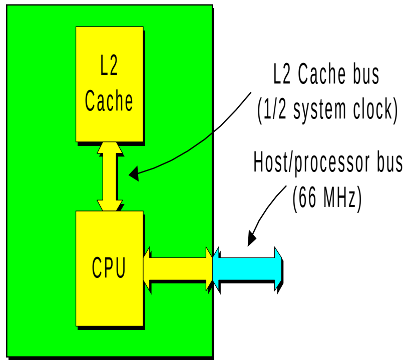


Fig.3. The Pentium Pro architecture (two chips in a multichip module).

THE MIDDLE KINGDOM

In 1982, Intel released the 16-bit 286 microprocessor (also known as the 80286). With 134,000 transistors, the 286 sported approximately three times the performance of other 16-bit processors of the time. Amongst other things, the 286 provided "backwards compatibility", which meant it could run programs that had been written for its predecessors. This was considered to be a pretty revolutionary concept at the time.

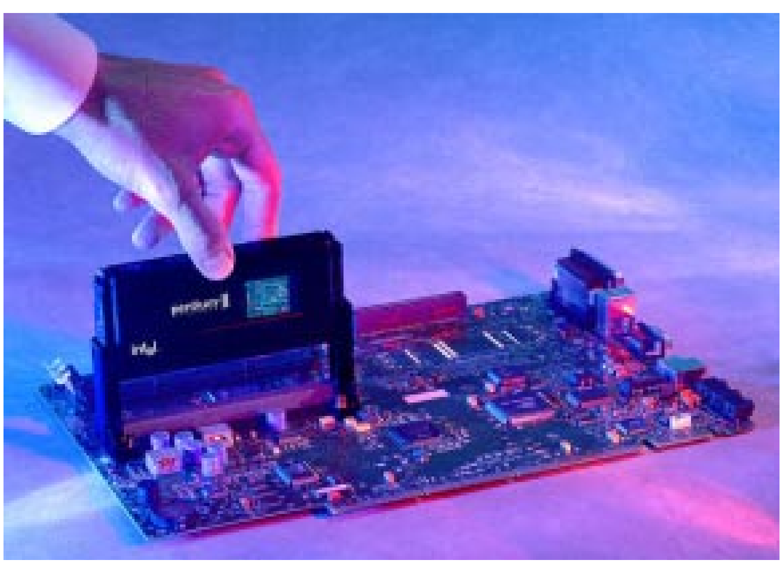


Fig.4. The Pentium II shown in a Slot 1 configuration. (Photo Courtesy of Intel Corp.)

A Brief Overview of Technical Terms

During the course of this article, one or two terms are mentioned, such as "cache", "host bus," MMX, and so forth. For those amongst us who aren't too familiar with these (and certain other) terms, a brief overview is presented here.

SRAM vs DRAM

Semiconductor memory comes in a variety of flavors. The two main categories are read-only memory (ROM) and random-access memory (RAM). (The purists amongst us would prefer to replace the term RAM with read-write memory (RWM), but realistically this is never going to happen).

ROM devices contain instructions and/or data that is "hard-wired" into them during their construction. By comparison, RAM devices only contain whatever data and instructions the computer last wrote into them (and they "forget" their contents when power is removed from the system).

Furthermore, RAM is itself split into two core technologies, which are known as Static RAM (SRAM) and Dynamic RAM (DRAM). Each memory cell in an SRAM device requires between 4 and 6 transistors, while each cell in a DRAM device requires only one transistor (so you can get more memory in a DRAM device). SRAMs are faster than DRAMS, but they use more power, run a lot get hotter, and are much more expensive. Thus, the bulk of a computer's main memory is formed from DRAM devices, and SRAMs are only used where extreme speed is required.

Cont...

The 16-bit 286 was followed by the 32-bit 386 in 1985. This device, which contained 275,000 transistors, was designed to support "multitasking" (running multiple programs at the same time). In turn, the 386 was followed in 1989 by the 486, which contained an amazing (for the time) 1.2 million transistors. In the early days, complex math functions (multiplication and division) were performed as a series of simple steps, such as "shift-and-add" algorithms. Later schemes employed external math coprocessor devices, which were special units dedicated to performing complex math functions as quickly as possible. The 486's 1.2 million transistors allowed it to include an on-chip math coprocessor as part of the main CPU (this was considered to be mega-cool by those of us who cared).

The first 486s had system clocks (see sidebar) running at 25 MHz. These were soon followed by 33MHz and 66MHz versions (and various "clock doublers" appeared later in the market).

THE GOLDEN AGE

The start of the current golden age of microprocessors (insofar as this article is concerned) occurred in 1993, when Intel first introduced the Pentium processor (Fig.2). With 3.1 million transistors, the Pentium was approximately five times faster than its 486 ancestor. The Pentium sported a 16KBL1 cache (see sidebar) and up to 512KB L2 cache (this L2 cache was mounted on the main motherboard). The first Pentiums started with clock speeds of 75 MHz, and over time additional clock speed options were added up to the 233 MHz versions of the present (note that even though the main clock frequency increased, the host bus speed remained at 66MHz – see sidebar). Also, the Pentium was presented in a pin grid array (PGA) package, which plugged into an Intel-defined pinout format called "Socket 7".

The next major development occurred in 1995 with the Pentium Pro, which sported a number of innovations and improvements (and a whopping 5.5 million transistors).

Quite apart from anything else, the Pentium Pro was pre-

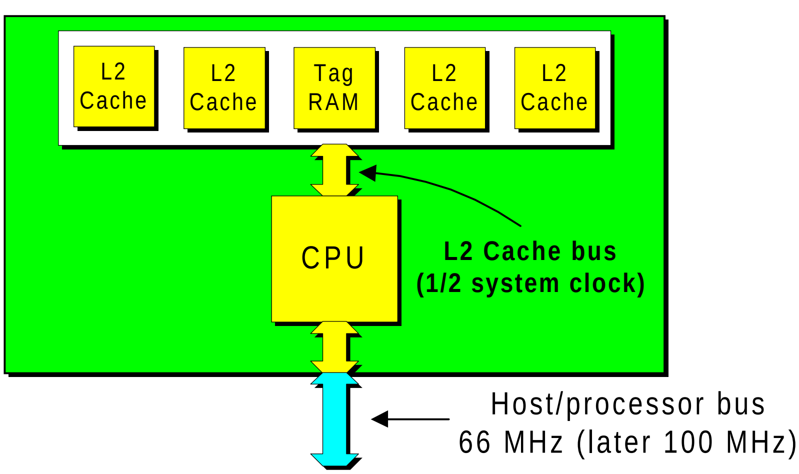


Fig.5. The Pentium II architecture.

SIMMs vs DIMMS

Multiple memory devices are typically mounted on small PCBs (known affectionately as "sticks" of memory). These are referred to as SIMMs (single inline memory modules) if they only have chips on one side, or DIMMS (dual inline memory modules) if they have chips on both sides.

FDO vs EDO vs SDRAM vs .....

As was previously noted, the bulk of a computers RAM is composed of DRAM devices, but these can be presented in different ways. When you purchased a computer in the not-so-distant past, it came with FDO (fast data out) memory. This was subsequently replaced in later systems by EDO (extended data out) memory, which was in turn superceded by SDRAM (synchronous DRAM) memory.

This is a bit confusing at first, but underneath it's really quite simple. These memory schemes are all basically formed around a core of standard DRAM chips, but each uses a different way of controlling and accessing the devices to squeeze more "throughput" out of them.

Also, note that the term ECC (error correcting and control) memory used in higher-end machines refers to the fact that these memory devices contain additional bits that can be used to detect (and correct) errors in the data.

L1 vs L2 Cache

Modern semiconductor memories are extremely fast,

Cont...

sented as a multichip module package containing two silicon chips: the Pentium Pro processor itself (with a 16-kilobyte on-chip L1 cache) and an L2 cache chip. The original Pentium Pros came equipped with 256KB L2 caches, and 512KB versions became available later. One really cunning innovation associated with the Pentium Pro was that it had two busses (an external processor-to-main-memory (host) bus and an internal processor-to-L2-cache bus) and the processor could access both busses simultaneously (Fig.3).

The original Pentium Pros supported a 150 MHz system clock, and this was quickly followed by 200 MHz versions. The multichip module of the Pentium Pro allowed the processor core and the L2 cache to be located very close to each other, which in turn facilitated the cache bus running at 1/2 the main system clock frequency. (This was much faster than the Pentium's L2 cache, which was located on the motherboard, and which was therefore obliged to run at the 66MHz host bus frequency.)

The end result of this dual-bus architecture (plus other cunning stuff that's too complex to go into here) made the Pentium Pro go "like a bat out of hell." The Pentium Pro module plugged into an Intel-defined pinout format called "Socket 8".

The pace really started to pick up in 1997, when Intel introduced two new developments. The first was the concept of MMX instructions (see sidebar), which first appeared in the "Pentium with MMX." The second was the Pentium II processor (with a mind-boggling 7.5 million transistors), which also included MMX instructions (Fig 4).

At a first glance, the Pentium II appeared to be radically different to the Pentium Pro, because it was presented in a relatively large package called a single-edge-connect (SEC) cartridge, which plugged into a new Intel-proprietary socket configuration called Slot 1 on the motherboard.

Internally, the Pentium II is constructed as a hybrid using a printed circuit board substrate. This circuit board contains the processor chip, which is essentially a Pentium Pro with a 32KB L1 cache and MMX instruction support. This board also contains four industry-standard burst-static cache RAM devices and a tag RAM chip, which together form the 512KB L2 cache (Fig.5).

(Note that Fig.5 is only an illustration. In reality, the processor core and two of the cache RAM devices are mounted on one side of the board, while the remaining cache RAMs and the Tag RAM are mounted on the other side.)

The first Pentium IIs supported clock frequencies of 233 and 266 MHz. These were quickly followed by 300 MHz, 333MHz, and 350 MHz versions, which were in turn superceded by 400 MHz and 450 MHz options. This obviously has a huge impact on processor operations, and also cache-intensive operations (because the Pentium II cache is running at 1/2 of the main clock frequency). Furthermore, the first Pentium IIs supported a host/processor bus frequency of 66MHz, but this was boosted up to100MHz in later versions, which dramatically improved access to the main memory.

Last but not least, the Pentium II supported Intel's AGP

but computer programs typically perform a humongous amount of memory accesses (reads and writes), so anything that can be done to speed things up is generally considered to be very desirable.

If we analyze programs, we tend to find that they are often composed of blocks of instructions, where each block may be performed multiple times before moving onto the next block. A key point is that these blocks of instructions tend to be relatively small compared to the size of the total program.

Computer designers can take advantage of this knowledge by creating a special "chunk" of high-speed memory called the cache. When the CPU starts running a program and it reads an instruction (or data) for the first time, in addition to processing that instruction it also stores it in the cache. The next time the CPU attempts to read that instruction (or data), it first checks to see if it's already in the cache – if so it can read it from the cache much faster than it could from the main memory.

The cache is formed from the fastest SRAM devices available, but the designer (who would ideally prefer as large a cache as possible) has to keep it smaller than he or she would like to balance cost, performance, and power consumption.

In fact computer designers typically use a hierarchy of memory. The L1 ("Level 1") cache is very small, but very, very fast, because it is constructed as an integral part of the CPU itself (on the same silicon chip). The L2 ("Level 2") cache is significantly bigger

Cont...

(advanced graphics port). Actually, to be more precise, it's the Pentium II's supporting chipset that provides AGP capability, but supporting chipsets are part of a completely different discussion, so we'll ignore them in this article (for which you can be truly thankful). The AGP is also outside the scope of this discussion, except to say that it is designed to streamline the transfer of data from the CPU to AGP-compatible graphics accelerator cards.

AND THEN …

For a long time things had appeared (at least on the surface) to be relatively well ordered. The 268 was followed by the 386, which was in turn followed by the 468. The 486 begot the Pentium, which led to the Pentium Pro, which in turn paved the way for the Pentium II. And then things began to get complicated …

In 1998, Intel introduced the Celeron processor, which was based on the Pentium II. The first Celerons didn't have any L2 cache at all, which caused some observers to refer to them as *"brain dead Pentium IIs."* Subsequent implementations had 128KB L2 caches, and it

than the L1 cache, but it's father away from the CPU and thus not as fast. (In fact some computers have L1, L2, and L3 caches between the CPU and the main memory!)

System Clock vs Host Bus

The main system clock is used to synchronize the internal actions of the CPU and also the rest of the system. The CPU communicates with the rest of the system by means of the main system bus, which may also be referred to as the "host bus", the "processor bus", or the "front bus."

Increasing the frequency of the system clock increases the rate by which the CPU performs its internal actions. However, the frequency of the host bus is not directly related to the frequency of the main system clock.

Overclocking

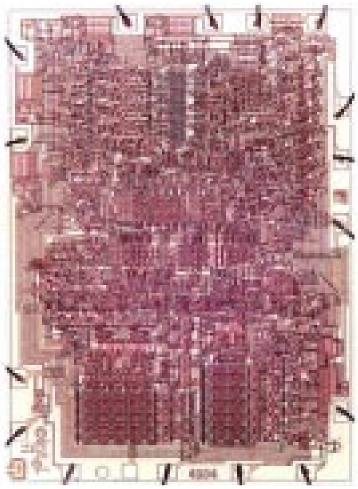
In order to understand the concept of overclocking, we first need to know that computer motherboards are designed to support a range of system clock frequencies. This allows a Pentium II motherboard, for example, to support say 350, 400, and 450 MHz versions of the processor. Theoretically, this allows end-users to upgrade their main processor (although this rarely happens in practice).

Most Intel processors are reasonably robust.

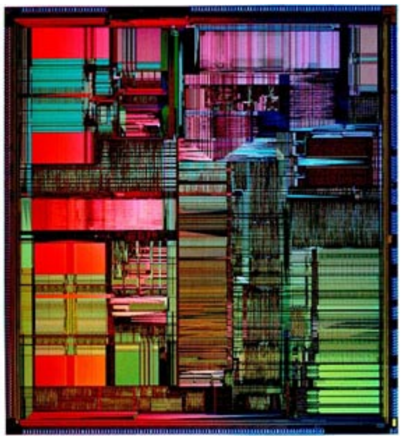
Cont..

Table.1. Summary of Intel microprocessor development

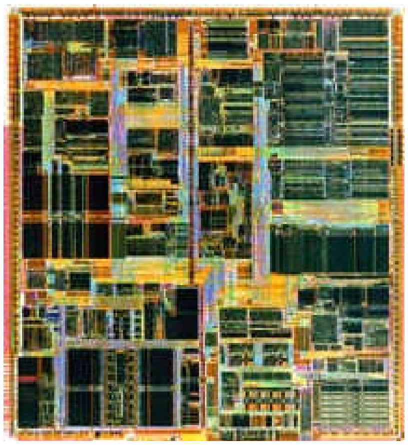
1971	4004	2,300 transistors, 4-bit, 60,000 operations per second
1972	8008	3,300 transistors, 8-bit
1972	4040	4-bit (like a 4004 but with additional instructions)
1974	8080	4,500 transistors, 8-bit, 200,000 operations per second
1978	8088/8086	8-bit/16-bit (the 8088 was used in the first IBM PC)
1982	286	134,000 transistors, 16-bit
1985	386	275,000 transistors, 32-bit (multitasking)
1989	486	1.2 million transistors, 32-bit (first math coprocessor) 25MHz system clock (-> to 66MHz over time)
1993	Pentium	3.1 million transistors, 32-bit, 16KB L1, 512KB L2 (external) 75MHz system clock (-> 233MHz), 66MHz host bus, Socket 7
1995	Pentium Pro	5.5 million transistors, 32-bit, 16KB L1, 256KB L2 (internal) 150MHz system clock (-> 200MHz), 66MHz host bus L2 Cache bus runs at 1/2 main system clock, Socket 8
1997	Pentium II	7.5 million transistors, 32-bit, 32KB L1, 512KB L2, MMX 233MHz system clock (-> 500MHz), 66MHz host bus (-> 100MHz) L2 Cache bus runs at 1/2 main system clock
1998	Celeron	7.5 million transistors, 32-bit, 32KB L1, 0KB L2 (later 128KB L2), MMX 266MHz system clock (-> 400MHz), 66MHz host bus L2 Cache (when provided) bus runs at 1/2 main system clock
1998	Pentium II Xeon	7.5 million transistors, 32-bit, 32KB L1, 512KB / 1MB / 2MB L2, MMX 400MHz system clock (-> 500MHz), 100MHz host bus L2 Cache bus runs at FULL main system clock
1999	Pentium III	7.5 million transistors, 32-bit, 32KB L1, 512KB L2, MMX II 500MHz system clock (-> 550MHz), 100MHz host bus L2 Cache bus runs at 1/2 main system clock
1999	Pentium III Xeon	7.5 million transistors, 32-bit, 32KB L1, 512KB / 1MB / 2MB L2 MMX II 500MHz system clock (-> 550MHz), 100MHz host bus L2 Cache bus runs at FULL main system clock



Bare (unpackaged) silicon chip (die) for the 4004 processor.



The Pentium die.



The Pentium II die.

(The die photos above are courtesy of Intel Corp.)

wouldn't be too surprising if versions with larger cache configurations started to appear.

Later on in 1998, just to make life interesting, Intel Introduced the Pentium II Xeon (Fig.6). This little rascal arrived in a yet bigger package, which plugged into a new Intel-proprietary socket configuration called Slot 2 on the motherboard.

The Pentium II Xeon contains two devices: the main processor and a honking big new L2 cache device. One key point about the Xeon (apart from its high price) is that its L2 cache, which is available in 512KB, 1MB, and 2MB options, runs at the ***same*** frequency as the main processor clock. (Remember that the cache bus on the standard Pentium II runs at *half* the main clock frequency.) This high-speed cache can have a significant effect for certain high-end applications such as database hosting and other server-bested applications.

And last but not least (as we pen these words), Intel have just announced the Pentium III (Slot 1)and the Pentium III Xeon (Slot 2) processors. These are essentially the same as their Pentium II forebears, but with higher clock frequencies (commencing at 500MHz and 550 MHz as they launched), including a special CPU ID, and supporting MMX II instructions (see sidebar).

With regard to the CPU ID, each Pentium III has a unique ID number hardwired into it, which is intended to allow querying software to positively identify the CPU (note that the CPU does not "broadcast" the ID, but instead programs have to actively read it). Intel's main (stated) goal was to provide in-

One by-product of this is that it is possible to take a processor rated for one clock frequency, say 350 MHz, and "tweak the jumpers" or otherwise modify the motherboard to clock the processor at a higher frequency.

There are a number of sites on the web that provide instructions on how to tweak different motherboards to overclock different processors. One downside of this is that the processor will run hotter than it's rated for (and than the computer's cooling system is expecting), which may cause problems. Another thing to bare in mind is that if you engage in overclocking, then you've just waved goodbye to any warranty that came with your system (assuming they find out what you did).

MMX vs MMX II

In 1997, Intel introduced the Pentium II and a new version of the Pentium, both of which boasted something called MMX. After wading through the "razzmatazz", we find that MMX Stands for "Multimedia eXtensions", which are actually a group of 57 special SIMD (single instruction multiple data) instructions.

To understand what this means, we first need to understand that many applications like to use data in small "chunks". For example, multimedia and image processing programs largely view the world as a series of picture elements (pixels), where each pixel consists of red, green, blue, and alpha (transparency) values (abbreviated to RGBA). Furthermore, many applications use 8-bits to represent each of these values, which makes it easy to pack all four values into a single

Cont...

creased security for electronic transactions (E-commerce) over the Internet. There are a number of other possible uses for such an ID, ranging from identifying stolen processors, to locking (licensing) applications to a specific machine.An unexpected result (at least, unexpected to Intel) is that a variety of civil liberties groups are currently "up in arms" campaigning about possible invasion of privacy issues. In response, Intel is providing a utility that will allow users to enable and disable the CPU ID at will, and the majority of machines will be shipped with the ID disabled by default (which sort of makes you wonder if it was all worth it in the first place).

WHICH SIZE FITS YOU?

For the remainder of this article, we will be concentrating on Intel's current product offerings: Celerons, Pentium IIs and IIIs, and Pentium II and III Xeons. It's certainly still possible to find systems sporting earlier processors, but these are now typically only available on the surplus market, as opposed to the "high street."

So which processor is right for you? This is a tricky one, because it can be difficult to separate the "hype" from the reality. Let's start with the Celeron, which is essentially a Pentium II with a smaller L2 cache *(whatever you do, don't buy one of the earlier Celerons without any L2 cache at all)*. Intel's marketing position is that the Celeron is intended for home use (games, entertainment, simple word processor-type applications) by the more cost-conscious buyer. In reality, Intel would probably have preferred not to have introduced the Celeron at all – they mainly did so in response to other processors from manufacturers like AMD, which were starting to "eat Intel's lunch" at the home consumer end of the market.

Pentium IIs (and now Pentium IIIs) are primarily targeted towards the high-end consumer and professional desktop (although television adverts often position them for home entertainment machines also). By comparison, Pentium II Xeons (and now Pentium III Xeons) are typically presented as being the processors of choice for high-end technical and professional applications. In reality, Xeons typically only deliver approximately 5% performance boost for many applications at a very significant cost premium over their non-Xeon counterparts. Where the Xeons really score (especially with the larger cache versions) is in server-type applications.

BOILING IT DOWN

Thus, for the majority of home users, the choice really boils down to one between a Celeron, a Pentium II, and a Pentium III. Let's see if we can reduce this choice a little further. If the truth be told, the vast majority of home users won't be able to spot any difference in performance between a Celeron (with 128KB L2 cache) and a Pentium II (with 512KB L2 cache) running at the same frequency. However, the Celeron will be several hundred dollars cheaper, and you can invest some of this money on additional RAM, which can provide a HUGE speed increase for memory-intensive applications. In fact some savvy users purchase a Celeron-based machine

32-bit word (the size currently used by the Pentium, Pentium Pro, Pentium II, and so forth).

The interesting point is that these applications often want to perform operations on pairs of 32-bit RGBA values, such as adding two of them together. However, the application doesn't simply want to add these two values as 32-bit quantities. Instead, it wishes to add the two 8-bit R values together, and likewise for the two 8-bit G values (and the B values and the A values).

The old way of doing this required multiple operations to separate out each of the 8-bit values, perform the requisite function, and then "munge" the results back together. Hence the introduction of MMX instructions, which direct the CPU to treat the data as four 8-bit chunks (but to still perform all four additions (or other operations) in a single clock beat!).

Of course these instructions are only of value if you have applications that make use of them. But once MMX became available, application developers began to take advantage of them, and MMX versions of certain programs run substantially faster than their non-MMX counterparts.

The original MMX instructions only worked on integer data representations. However, the new Pentium III and Pentium III Xeon processors have been enhanced to support MMX II. Also known as SSE (for "streaming SIMD extensions"), MMX II comprises around 70 additional instructions that allow operations to be performed on four floating-point numbers simultaneously.

Cont..

and then overclock it (see sidebar), which provides the increased performance associated with higher clock frequencies.

Of course, for those users who want (nay NEED) to live on the cutting edge of technology, only a Pentium III processor will suffice (and there aren't any Pentium III-based versions of the Celeron to play with). Sad to relate, however, the Pentium III typically won't actually perform any faster than a Pentium II running at the same clock frequency (at least not today). This is because the only real difference between the two processors (apart from the fact that the Pentium III has a CPU ID) is the presence of MMX II streaming SIMD instructions in the Pentium III.

What this means is that if you were to load one of your existing applications on a Pentium III, it wouldn't actually run any faster than on a Pentium II at the same clock frequency. If that application were subsequently recompiled by its developer using a compiler that was intelligent enough to take advantage of the Pentium III's MMX II instructions, then you would see some level of performance improvement. But in order to really shine, the application would have to be modified by the developer to take full advantage of these instructions. (Additionally, you can expect to see a significant performance boost for 3D graphics applications when the companies that supply graphics accelerator cards upgrade their drivers to take advantage of MMX II.)

The problem is that it will take some time for mainstream applications to be upgraded to take advantage of these instructions (and then you'll have to purchase these upgraded versions). Another consideration is that, in the same timeframe, you'll be able to get even faster processors for less money (see the next section). But it was always thus, and at the end of the day it's a case of *"you pays your money and you takes your choice"* as the old saying goes.

Finally, a note for those of you who are still struggling along by candlelight using your old 286 and 386 machines (*"it's still got a lot of life in it yet,"* you cry). Well take my word that the world has moved on, and it really is time for you to consider *"coming towards the light".*

AS FOR TOMORROW

"And what about tomorrow?" – you ask – "What new wonders are heading our way?" Well there are a whole host of exciting developments racing along towards us.

First, processor clock frequencies are going to rise yet higher. Internet newsgroups are confidently predicting 600 MHz clocks by the end of 1999, 700 MHz clocks by mid-2000, and 1 GHz clocks soon thereafter.

Next, Intel are working furiously on a 64-bit processor (the members of the current generation all work with 32-bit wide chunks of data), which should give another huge performance boost (when applications are rewritten and/or recompiled to take advantage of this feature, of course).

Also, new integrated circuit production technologies are allowing increasing numbers of transistors to be squeezed onto each chip. In the foreseeable future it will be possible to cram 100 million transistors onto a single device. What will all of

Future applications that take advantage ofthese instructions (such as analytical programs, graphics applications, and 3D games) will see a major speed increase.

Intel Who?

There are many fine microprocessors around, including those from Motorola (as found in Apple's Macintosh machines), AMD (K5, K6, K7, …), IBM (Cyrix 6x86), and others. So why has this article concentrated on those from Intel (you may ask)?

Strange to relate, Intel never really intended to manufacturer microprocessors – their original goal was to supply the world with semiconductor memory devices. So it rather took them by surprise when one of their design teams created the world's first microprocessor – the 4004. In fact the fate of the 4004 was precarious at best, and one of the main reasons that persuaded Intel's management to invest in continued development was the argument that the availability of microprocessors would help them to sell more memory devices.

In the mid-1970s, Intel considered adding a keyboard and a monitor to a board containing their 8080 microprocessor to sell it to the home user, but no one could think of a use for it (apart from *"something for housewives to keep their recipes on"*) so they dropped the idea.

Sometime later, IBM decided that there *was* a market for computers for the small business user, so they set about creating the world's first PC. At that time, the two main contenders to supply the CPU for

Cont...

these transistors be used for? Well in addition to making the processors more powerful, it will be possible to dramatically increase the sizes of the local memory caches and locate the caches on the core CPU device, which will significantly improve the processor's performance.

Meanwhile, processors will be able to handle ever-larger amounts of memory (which shall hopefully continue its trend to become even cheaper). Some high-end Intel-based professional workstations can already access 3GB of RAM or more (which makes the author's 64MB look positively measly by comparison). Also, Intel are working on bringing a next-generation memory system called Direct RAMBUS to the market, which will dramatically speed up processor-to-memory accesses.

It's also worth noting that, despite the Intel-centric focus of this article, Intel aren't the only game in town. There are a number of other players developing processors that look set to give Intel a run for their money.

Last but not least, there are a whole host of technologies that are outside the scope of this article, including optical fiber-based disk drives, and ….. but that would be telling, wouldn't it? Suffice it to say that, for those of us who love computers, the future promises nothing less than faster, better, and cheaper, and who amongst us can find fault with that?

FURTHER READING

The hybrid and multichip module packaging technologies mentioned in this article are described in detail in an amazingly useful book called *Bebop to the Boolean Boogie (An Unconventional Guide to Electronics)*. For those who want to understand more about how microprocessors work, *Bebop BYTES Back (An Unconventional Guide to Computers)* is jolly good. By some strange quirk of fate, both of these tomes are available from the *EPE Online Store* at **www.epemag.com**

For those who want the inside scoop on the latest breaking news in the processor wars, an excellent place to start is Tom Pabst's web site at **www.tomshardware.com**. There's also a lot of useful information put out by the guys and gals at UGEEK (**www.ugeek.com**), and Intel also provide a tremendous amount of interesting information (**www.intel.com**), although you sometimes have to root around a bit to find the interesting stuff.

this PC were Intel's 8088 and Motorola's 6800. Had IBM opted for the Motorola device, we would all now doubtless have "Motorola Inside" logos tattooed on our foreheads.

Why did IBM choose Intel over Motorola? Who knows? Who Cares? The decision was made, with the result that Intel Corporation are now arguably the best known manufacturer of computers on the face of the planet.

Go to next section