Intel Processor Family Evolution

Scaling from 4004 to Pentium 4

Courtesy of Intel Museum
• First microprocessor (1971)
  – For Busicom calculator
• Characteristics
  – 10 μm process
  – 2300 transistors
  – 400 – 800 kHz
  – 4-bit word size
  – 16-pin DIP package
• Masks hand cut from Rubylith
  – Drawn with color pencils
  – 1 metal, 1 poly (jumpers)
  – Diagonal lines (!)
8008

• 8-bit follow-on (1972)
  – Dumb terminals
• Characteristics
  – 10 \( \mu \)m process
  – 3500 transistors
  – 500 – 800 kHz
  – 8-bit word size
  – 18-pin DIP package
• Note 8-bit datapaths
  – Individual transistors visible
8080

• 16-bit address bus (1974)
  – Used in Altair computer
    • (early hobbyist PC)
• Characteristics
  – 6 µm process
  – 4500 transistors
  – 2 MHz
  – 8-bit word size
  – 40-pin DIP package
8086 / 8088

- 16-bit processor (1978-9)
  - IBM PC and PC XT
  - Revolutionary products
  - Introduced x86 ISA
- Characteristics
  - 3 \( \mu \)m process
  - 29k transistors
  - 5-10 MHz
  - 16-bit word size
  - 40-pin DIP package
- Microcode ROM
1M Total Min. Access
• 64K segments (16-bit)
• 4 segments (256K) at a time in 1
• CS pts to main mem. w/16-bit IP
• BIU computes 20-bit addr. using
  CS & IP & shifting CS 4 bits left, then add to IP reg.
Ex: (CS) = 456AH (16-bits)
     (IP) = 1620H (16-bits)
     CS ⇒ 456A0H (20-bits)
+ IP ⇒ 1620H (16-bits)
     46CC0H (phys. addr.)
Minimum Mode 8086 Typical Configuration
Maximum Mode 8086 Typical Configuration
80x87 Floating point coprocessor

When the 8086 CPU first appeared in the late 1970's, semiconductor technology was not to the point where Intel could put floating point instructions directly on the 8086 CPU. Therefore, they devised a scheme whereby they could use a second chip to perform the floating point calculations - the floating point unit (or FPU)[6]. They released their original floating point chip, the 8087, in 1980. This particular FPU worked with the 8086, 8088, 80186, and 80188 CPUs. When Intel introduced the 80286 CPU, they released a redesigned 80287 FPU chip to accompany it. Although the 80287 was compatible with the 80386 CPU, Intel designed a better FPU, the 80387, for use in 80386 systems. The 80486 CPU was the first Intel CPU to include an on-chip floating point unit. Shortly after the release of the 80486, Intel introduced the 80486sx CPU that was an 80486 without the built-in FPU. To get floating point capabilities on this chip, you had to add an 80487 chip, although the 80487 was really nothing more than a full-blown 80486 which took over for the "sx" chip in the system. Intel's Pentium/586 chips provide a high-performance floating point unit directly on the CPU. There is no floating point coprocessor available for the Pentium chip.
**80x87 FPU Data Regs**

The 80x87 FPUs provide eight 80 bit data registers organized as a stack. This is a significant departure from the organization of the general purpose registers on the 80x86 CPU that comprise a standard general-purpose register set. Intel refers to these registers as ST(0), ST(1), ..., ST(7). Most assemblers will accept ST as an abbreviation for ST(0).

Source: Art of Assembly: Chapter Fourteen
80286

- Virtual memory (1982)
  - IBM PC AT
- Characteristics
  - 1.5 μm process
  - 134k transistors
  - 6-12 MHz
  - 16-bit word size
  - 68-pin PGA
- Regular datapaths and ROMs
  Bitslices clearly visible
• The first complete redesign started with the 80286.
  – Although the processor was viable it is interesting to note the quick architecture design turn round Intel performed to superseded the 80286 by the 80386 and later the 80486 and Pentium.

• The most radical change at the 80286 was the introduction of REAL and PROTECTED modes.

• The overall processor design on the 80286 was radically reconsidered. The choice of basic architecture was still the Von Neuman structure, rather than the more RISC based Harvard architecture.
80286 Internal Block Diagram
80286 Virtual Address Translation
80386

- 32-bit processor (1985)
  - Modern x86 ISA
- Characteristics
  - 1.5-1 µm process
  - 275k transistors
  - 16-33 MHz
  - 32-bit word size
  - 100-pin PGA
- 32-bit datapath, microcode ROM, synthesized control
80386 Block Diagram
80386-486

- The 80386 is a full 32 bit implementation of the 80286 with the problems associated with the enhancements to REAL and PROTECTED mode removed.
- The 80846 built on the model derived at 80386, has all that processors major features, but these are complemented by an improvement in the instruction execution micro code, (driven partly by AMD's success in producing micro code for the clone 80386 which showed significant execution time improvement).
- Inclusion of a 8M byte internal cache, an internal enhanced floating point co-processor, it is generally accepted that at the same clock speed the 80486 showed a 50% improvement in performance over the 80386.
The i387 has a control unit for driving the bus and for controlling the numeric unit. The numeric unit carries out all calculations with floating-point numbers in an exponent and a mantissa module. Unlike the i386, the i387 has a register stack instead of discrete registers.
i387 Internal Registers

Register Stack

Control and Status Register

Instruction Pointer

Data Pointer

Tag

Status

Control
i386/i387 System Configuration
Internal Structure of the 386SL and 82360SL

SX versus SL
80486

• Pipelining (1989)
  – Floating point unit
  – 8 KB cache
• Characteristics
  – 1-0.6 μm process
  – 1.2M transistors
  – 25-100 MHz
  – 32-bit word size
  – 168-pin PGA
• Cache, Integer datapath, FPU, microcode, synthesized control
i486 Architecture
Embedded IntelDX2™ Processor

New Features vs. 386

- Integrated Floating-Point Unit
- Speed-Multiplying Technology
- 32-Bit RISC Technology Core
- 8-Kbyte Write-Through Cache
- Four Internal Write Buffers
- Burst Bus Cycles
- Dynamic Bus Sizing for 8- and 16-bit Data Bus Devices

- SL Technology
- Data Bus Parity Generation and Checking
- Boundary Scan (JTAG)
- 3.3-Volt Processor, 50 MHz, 25 MHz CLK
  — 208-Lead Shrink Quad Flat Pack (SQFP)
- 5-Volt Processor, 66 MHz, 33 MHz CLK
  — 168-Pin Pin Grid Array (PGA)
- Binary Compatible with Large Software Base
The i486 pipeline comprises one prefetch, two decoding, one execution and one write-back stage. The figure shows, as an example, the execution of an ADD eax, mem32 instruction.
Every instruction is broken down into partial steps for execution in the five-stage pipeline. The partial steps are executed within one single clock cycle. Thus, for example, the instruction $k$ needs five cycles to complete. But at the pipeline output an instruction result is available with each clock cycle.
Without pipelining and delayed branch, the processor would continue execution with the first instruction at the jump target (here: AND) immediately after the branch instruction. Because of instruction pipelining, several NOPs have to be inserted after the JMP instruction (four NOPs with the five-stage pipeline) so that the processor does not provide any incorrect result. Not before cycle $n + 4$ does the processor carry out the write-back of the operation result, that is, it writes the instruction pointer with the jump target address. Therefore, instruction AND at the target address cannot occur before cycle $n + 5$. 

Delayed Branch With Inserted NOPs

<table>
<thead>
<tr>
<th>Cycle n</th>
<th>Instruction Fetch IF</th>
<th>Decode D1</th>
<th>Operand Fetch OF</th>
<th>Execution EX</th>
<th>Write-back WB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>JMP</strong></td>
<td><strong>ADD</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cycle n+1</td>
<td><strong>NOP</strong></td>
<td><strong>JMP</strong></td>
<td><strong>ADD</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cycle n+2</td>
<td><strong>NOP</strong></td>
<td><strong>NOP</strong></td>
<td><strong>JMP</strong></td>
<td><strong>ADD</strong></td>
<td></td>
</tr>
<tr>
<td>Cycle n+3</td>
<td><strong>NOP</strong></td>
<td><strong>NOP</strong></td>
<td><strong>NOP</strong></td>
<td><strong>JMP</strong></td>
<td><strong>ADD</strong></td>
</tr>
<tr>
<td>Cycle n+4</td>
<td><strong>NOP</strong></td>
<td><strong>NOP</strong></td>
<td><strong>NOP</strong></td>
<td><strong>NOP</strong></td>
<td><strong>JMP</strong></td>
</tr>
<tr>
<td>Cycle n+5</td>
<td></td>
<td><strong>AND</strong></td>
<td><strong>NOP</strong></td>
<td><strong>NOP</strong></td>
<td><strong>NOP</strong></td>
</tr>
</tbody>
</table>
I486 TAP Structure

The input TDI and output TDO can be connected to the output of a circuit in a preceding stage and the input of a circuit in a successive stage, respectively.
Pentium

- Superscalar (1993)
  - 2 instructions per cycle
  - Separate 8KB I$ & D$
- Characteristics
  - 0.8-0.35 μm process
  - 3.2M transistors
  - 60-300 MHz
  - 32-bit word size
  - 296-pin PGA
- Caches, datapath, FPU, control
Pipeline

The Pentium's basic integer pipeline is five stages long, with the stages broken down as follows:

1. Prefetch/Fetch: Instructions are fetched from the instruction cache and aligned in prefetch buffers for decoding.

2. Decode1: Instructions are decoded into the Pentium's internal instruction format. Branch prediction also takes place at this stage.

3. Decode2: Same as above, and microcode ROM kicks in here, if necessary. Also, address computations take place at this stage.

4. Execute: The integer hardware executes the instruction.

5. Write-back: The results of the computation are written back to the register file.
The Pentium's U and V integer pipes were not fully symmetric. U, as the default pipe, was slightly more capable and contained a shifter, which V lacked.

Floating-point, however, simply went from awful on the 486 to just mediocre with the Pentium — an improvement, to be sure, but not enough to make it even remotely competitive with comparable RISC chips on the market at that time.
Pipeline Instruction Pairing Rules

- Both instructions must be simple
  - Hardwired, no microcode support
  - Must execute in 1 clock cycle

- No data dependencies between instructions (either memory or regs)

- Neither instruction may contain both a displacement and an immediate value

- Instructions w/prefixes can only be issued in the U-pipe
  - Branches can only be the 2nd of a pair
  - Must execute in V-pipe
Pipeline Instruction Pairing Rules

Pseudocode:

IF I1 is simple
  AND I2 is simple
  AND I1 is not a jump
  AND dest. of I1 is not source of I2
  AND dest. Of I1 is not dest of I2
THEN  issue I1 to U-pipe
       Issue I2 to V-pipe
ELSE issue I1 to U-pipe
Efficiency of Instruction Pairing Rules

C-code:

```c
for(k = l + prime; k <= SIZE; k += PRIME)
    Flags[k] = FALSE
```

Compiler assembly:

```
; PRIME in ecx, k in edx, FALSE in al
Inner_loop
```

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
<th>Execution cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVE byte ptr flags[edx], al</td>
<td>1 paired</td>
<td>80486 1</td>
</tr>
<tr>
<td>ADD edx, ecx</td>
<td>2 1</td>
<td>80486 1</td>
</tr>
<tr>
<td>CMP edx, FALSE</td>
<td>1 paired</td>
<td>80486 1</td>
</tr>
<tr>
<td>JLE Inner_loop</td>
<td>2 1</td>
<td>Pentium 6 2</td>
</tr>
</tbody>
</table>
The Pentium FDIV bug is the most famous (or infamous) of the Intel microprocessor bugs. It was caused by an error in a lookup table that was a part of Intel's SRT algorithm that was to be faster and more accurate.

With a goal to boost the execution of floating-point scalar code by 3 times and vector code by 5 times, compared to the 486DX chip, Intel decided to use the SRT algorithm that can generate two quotient bits per clock cycle, while the traditional 486 shift-and-subtract algorithm was generating only one quotient bit per cycle. **This SRT algorithm uses a lookup table to calculate the intermediate quotients** necessary for floating-point division. Intel's lookup table consists of 1066 table entries, of which, due to a programming error, **five were not downloaded into the programmable logic array (PLA).** When any of these five cells is accessed by the floating point unit (FPU), **it (the FPU) fetches zero instead of +2**, which was supposed to be contained in the "missing" cells. This throws off the calculation and results in a less precise number than the correct answer (Byte Magazine, March 1995).

At its worst, this error can occur as high as the fourth significant digit of a decimal number, but the possibilities of this happening are 1 in 360 billion. It is most common that the error appears in the 9th or 10th decimal digit, which yields a chance of this happening of 1 in 9 billion.

Intel has classified the bug (or the flaw, as they refer to it) with the following characteristics:

- On certain input data, the FPDI (Floating Point Divide Instructions) on the Pentium processor produce inaccurate results.
- The error can occur in any of the three operating precisions, namely single, double, or extended, for the divide instruction. However, it has been noted that far fewer failures are found in single precision than in double or extended precisions.
- The incidence of the problem is independent of the processor rounding modes.
- The occurrence of the problem is highly dependent on the input data. Only certain data will trigger the problem. There is a probability that 1 in 9 billion randomly fed divide or remainder instructions will produce inaccurate results.
- The degree of inaccuracy depends on the input data and upon the instruction involved.
- The problem does not occur on the specific use of the divide instruction to compute the reciprocal of the input operand in single precision.

Furthermore, the bug affects any instruction that references the lookup table or calls FDIV. Related instructions that are affected by the bug are FDIVP, FDIVR, FDIVRP, FIDIV, FIDIVR, FPREM, and FPREM1. The instructions FPTAN and FPATAN are also susceptible. The instructions FYL2X, FYL2XP1, FSIN, FCOS, and FSINCOS, were a suspect but are now considered safe.
A 3-D plot of the ratio 4195835/3145727 calculated on a Pentium with FDIV bug. The depressed triangular areas indicate where incorrect values have been computed. The correct values all would round to 1.3338, but the returned values are 1.3337, an error in the fifth significant digit.

Byte Magazine, March 1995. Intel has adopted a no-questions-asked replacement policy for its customers with the Pentium FDIV bug. It has also done statistical reasearch and provided information on the bug at its site at intel.com
Pentium Pro / II / III

- Dynamic execution (1995-9)
  - 3 micro-ops / cycle
  - Out of order execution
  - 16-32 KB I$ & D$
  - Multimedia instructions
  - PIII adds 256+ KB L2$

- Characteristics
  - 0.6-0.18 µm process
  - 5.5M-28M transistors
  - 166-1000 MHz
  - 32-bit word size
  - MCM / SECC
Pentium Pro Architecture
The P6 Pipeline

The P6 has a 12-stage pipeline — considerably longer than the Pentium's 5-stage pipeline.

BTB access and instruction fetch: The first three and a half pipeline stages are dedicated to accessing the branch target buffer and fetching the next instruction. The P6's two-cycle instruction fetch phase is longer than the Pentium's 1-cycle fetch, but it keeps the L1 cache access latency from holding back the clock speed of the processor as a whole.

Decode: The next two-and-a-half stages are dedicated to decoding x86 instructions and breaking them down into the P6's internal, RISC-like instruction format.

Register rename: This stage takes care of register renaming and logging instructions in the ROB.

[Reservation Station]

Write to RS: Moving instructions from the ROB to the RS takes one cycle, and occurs here.

Read from RS: It takes another cycle to move instructions out of the RS, through the issue ports, and into the execution units.

Execute: Instruction execution can take one cycle, as in the case of simple integer instructions, or multiple cycles, as in the case of floating-point instructions.

Retire: These two final cycles are dedicated to writing the results of the instruction execution back into the ROB, and then retiring the instructions by writing their results from the ROB into the architectural register file.
Pentium II
Single Edge Cartridge (SEC)

Product Highlights

- Available in speeds from 233 MHz up to 450 MHz.
- Utilizes Intel’s 0.25 micron manufacturing process for increased processor core frequencies and reduced power consumption.
- Includes MMX media enhancement technology.
- The Pentium(r) II processor at 450 MHz delivers 32% more integer performance (as measured by SPECint95), 30% more multimedia performance (as measured by Norton Media Benchmark), and 35% more floating point performance (as measured by SPECfp95) than the 333 MHz Pentium(r) II processor.
- Dual Independent Bus (DIB) architecture increases bandwidth and performance over single-bus processors.

This is why they need a cartridge design.
Pentium III
Single Edge Cartridge (SEC)

Product Highlights

- The Pentium® III Processor is available at speeds ranging from 450 MHz to 1.13 GHz.
- Versions available with either a 133 MHz or a 100 MHz system bus that are designed to support the Intel® 840, 820, 810e, 440GX and 440BX chipsets.
- Versions that incorporate 256 KB Advanced Transfer Cache (on-die, full-speed level 2 (L2) cache with Error Correcting Code (ECC) or versions that incorporate a discrete, half-speed, 512 KB in package L2 cache with ECC.
- 32 KB (16 KB/16 KB) non-blocking, level 1 (L1) cache.
- P6 Dynamic Execution microarchitecture including multiple branch prediction, data flow analysis and speculative execution.
- Internet Streaming SIMD Extensions, consisting of 70 new instructions that enable advanced imaging, 3D, streaming audio and video, speech recognition and an enhanced Internet experience.
- Intel® MMX™ media enhancement technology.
- Dual Independent Bus (DIB) architecture increases bandwidth and performance over single-bus processors.
- Memory cacheability up to 4 GB of addressable memory space and system memory scalability up to 64 GB of physical memory.
- Both dual-processor capable and uni-processor only versions.
- Data integrity and reliability features such as Error Correction Code, Fault Analysis and Recovery for both system and L2 cache buses.
- Intel® processor serial number, designed to improve asset management, platform identification and
Pentium 4

- Deep pipeline (2001)
  - Very fast clock
  - 256-1024 KB L2

- Characteristics
  - 180 – 90 nm process
  - 42-125M transistors
  - 1.4-3.4 GHz
  - 32-bit word size
  - 478-pin PGA

- Units start to become invisible on this scale
The Intel® Pentium® 4 Processor
Intel® NetBurst™ Micro-Architecture

42 million transistors

400 MHz System Bus
Advanced Dynamic Execution
Rapid Execution Engine

Advanced Transfer Cache
Hyper Pipelined Technology
Streaming SIMD Extensions 2

Execution Trace Cache
Enhanced Floating Point / Multi-Media

Intel

Copyright © 2000 Intel Corporation
Intel NetBurst Microarchitecture Overview

• Designed to achieve high performance for integer and floating point computations at high clock rates
• Features:
  – hyper-pipelined technology that enables high clock rates and frequency headroom (up to 10 GHz)
  – a high-performance, quad-pumped bus interface to the Intel NetBurst microarchitecture system bus
  – a rapid execution engine to reduce the latency of basic integer instructions
  – out-of-order speculative execution to enable parallelism
  – superscalar issue to enable parallelism
  – Hardware register renaming to avoid register name space limitations
  – Cache line sizes of 64 bytes
  – Hardware pre-fetch
  – A pipeline that optimizes for the common case of frequently executed instructions
  – Employment of techniques to hide stall penalties such as parallel execution, buffering, and speculation
Pentium 4 Hyper-Threading Technology

- Enables software to take advantage of both task-level and thread-level parallelism by providing multiple logical processors within a physical processor package.
Hyper-Threading Basics

• Two logical units in one processor
  – Each one contains a full set of architectural registers
  – But, they both share one physical processor’s resources

• Appears to software (including operating systems and application code) as having two processors.

• Provides a boost in throughput in actual multiprocessor machines.

• Each of the two logical processors can execute one software thread.
  – Allows for two threads (max) to be executed simultaneously on one physical processor
Hyper-Threading Resources

• Replicated Resources
  – Architectural State is replicated for each logical processor. The state registers control program behavior as well as store data.
    • General Purpose Registers (8)
    • Control Registers
    • Machine State Registers
    • Debug Registers
  – Instruction pointers and register renaming tables are replicated to track execution and state changes.
  – Return Stack is replicated to improve branch prediction of return instructions
  – Finally, Buffers were replicated to reduce complexity

• Partitioned Resources
  – Buffers are shared by limiting the use of each logical processor to half the buffer entries.
  – By partitioning these buffers the physical processor achieves:
    • Operational fairness
    • Allows operations from one logical processor to continue on while the other logical processor may be stalled.
  – Example: cache miss – partitioning prevents the stalled logical processor from blocking forward progress.
  – Generally speaking, the partitioned buffers are located between the major pipeline stages.

• Shared Resources
  – Most resources in a physical processor are fully shared:
    • Caches
    • All execution units
  – Some shared resources (like the DTLB) include an identification bit to determine which logical processor the information belongs too.
Summary

• $10^4$ increase in transistor count, clock frequency over 30 years!

<table>
<thead>
<tr>
<th>Processor</th>
<th>Year</th>
<th>Feature Size (µm)</th>
<th>Transistors</th>
<th>Frequency (MHz)</th>
<th>Word size</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>4004</td>
<td>1971</td>
<td>10</td>
<td>2.3k</td>
<td>0.75</td>
<td>4</td>
<td>16-pin DIP</td>
</tr>
<tr>
<td>8008</td>
<td>1972</td>
<td>10</td>
<td>3.5k</td>
<td>0.5–0.8</td>
<td>8</td>
<td>18-pin DIP</td>
</tr>
<tr>
<td>8080</td>
<td>1974</td>
<td>6</td>
<td>6k</td>
<td>2</td>
<td>8</td>
<td>40-pin DIP</td>
</tr>
<tr>
<td>8086</td>
<td>1978</td>
<td>3</td>
<td>29k</td>
<td>5–10</td>
<td>16</td>
<td>40-pin DIP</td>
</tr>
<tr>
<td>80286</td>
<td>1982</td>
<td>1.5</td>
<td>134k</td>
<td>6–12</td>
<td>16</td>
<td>68-pin PGA</td>
</tr>
<tr>
<td>Intel386</td>
<td>1985</td>
<td>1.5–1.0</td>
<td>275k</td>
<td>16–25</td>
<td>32</td>
<td>100-pin PGA</td>
</tr>
<tr>
<td>Intel486</td>
<td>1989</td>
<td>1–0.6</td>
<td>1.2M</td>
<td>25–100</td>
<td>32</td>
<td>168-pin PGA</td>
</tr>
<tr>
<td>Pentium</td>
<td>1993</td>
<td>0.8–0.35</td>
<td>3.2–4.5M</td>
<td>60–300</td>
<td>32</td>
<td>296-pin PGA</td>
</tr>
<tr>
<td>Pentium Pro</td>
<td>1995</td>
<td>0.6–0.35</td>
<td>5.5M</td>
<td>166–200</td>
<td>32</td>
<td>387-pin MCM PGA</td>
</tr>
<tr>
<td>Pentium II</td>
<td>1997</td>
<td>0.35–0.25</td>
<td>7.5M</td>
<td>233–450</td>
<td>32</td>
<td>242-pin SECC</td>
</tr>
<tr>
<td>Pentium III</td>
<td>1999</td>
<td>0.25–0.18</td>
<td>9.5–28M</td>
<td>450–1000</td>
<td>32</td>
<td>330-pin SECC2</td>
</tr>
<tr>
<td>Pentium 4</td>
<td>2001</td>
<td>0.18–0.13</td>
<td>42–55M</td>
<td>1400–3200</td>
<td>32</td>
<td>478-pin PGA</td>
</tr>
</tbody>
</table>
Link to all Intel processor data sheets: