x87

x87 is a floating point-related subset of the x86 architecture instruction set. It originated as an extension of the 8086 instruction set in the form of optional floating point coprocessors that worked in tandem with corresponding x86 CPUs. These microchips had names ending in "87". Like other extensions to the basic instruction set, x87-instructions are not strictly needed to construct working programs, but provide hardware and microcode implementations of common numerical tasks, allowing these tasks to be performed much faster than corresponding machine code routines can. The x87 instruction set includes instructions for basic floating point operations such as addition, subtraction and comparison, but also for more complex numerical operations, such as the computation of the tangent function and its inverse, for example.

Most x86 processors since the Intel 80486 have had these x87 instructions implemented in the main CPU but the term is sometimes still used to refer to that part of the instruction set. Before x87 instructions were standard in PCs, compilers or programmers had to use rather slow library calls to perform floating-point operations, a method that is still common in (low-cost) embedded systems.

Description

The x87 registers form an 8-level deep non-strict stack structure ranging from ST(0) to ST(7) with registers that can be directly accessed by either operand, using an offset relative to the top, as well as pushed and popped. (This scheme may be compared to how a stack frame may be both pushed, popped and indexed.)

There are instructions to push, calculate, and pop values on top of this stack; monadic operations (FSQRT, FPTAN etc.) then implicitly address the topmost ST(0) while dyadic operations (FADD, FMUL, FCOM, etc.) implicitly address ST(0) and ST(1). The non-strict stack-model also allows dyadic operations to use ST(0) together with a direct *memory operand* or with an *explicitly* specified stack-register, ST(*x*), in a role similar to a traditional accumulator (a combined destination and left operand).

This can also be reversed on an instruction-by-instruction basis with ST(0) as the unmodified operand and ST(*x*) as the *destination*. Furthermore, the contents in ST(0) can be exchanged with another stack register using an instruction called FXCH ST(*x*).

These properties makes the x87 stack usable as seven freely addressable registers plus a dedicated accumulator (or as seven independent accumulators). This is especially applicable on superscalar x86 processors (such as the Pentium of 1993 and later) where these exchange instructions (codes D9C8..D9CF_h) are optimized down to a zero clock penalty by using one of the integer paths for FXCH ST(*x*) in parallel with the FPU instruction. Despite being natural and convenient for human assembly language programmers, some compiler writers have found it complicated to construct automatic code generators that schedule x87 code effectively.

The x87 provides single precision, double precision and 80-bit double-extended precision binary floating-point arithmetic as per the IEEE 754-1985 standard. By default, the x87 processors all use 80-bit double-extended precision internally (to allow for sustained precision over many calculations). A given sequence of arithmetic operations may thus behave slightly differently compared to a strict single-precision or double-precision IEEE 754 FPU.[1] This may sometimes be problematic for some semi-numerical calculations relying on knowledge of exact FPU precision for correct operation. To avoid such problems, the x87 can be configured via a special configuration/status register to automatically round to single or double precision after each operation. Since the introduction of SSE2, the x87 instructions are not as essential as they once were, except for high-precision calculations demanding the 64-bit mantissa precision available in the 80-bit format.

Performance

Clock cycle counts for examples of typical ***x87*** *FPU instructions (only register-register versions shown here).*[2]

The A~B notation (minimum to maximum) covers timing variations dependent on transient pipeline status as well as the arithmetic precision chosen (32, 64 or 80 bits); it also includes variations due to numerical cases (such as the number of set bits, zero, etc.). The L→H notation depicts values corresponding to the lowest (L) and the highest (H) maximum clock frequencies that were available.

x87 implementation	FADD	FMUL	FDIV	FXCH	FCOM	FSQRT	FPTAN	FPATAN	Max Clock	Peak FMUL/sec	Relative 5 MHz 8087§ FMUL
8087	70~100	90~145	193~203	10~15	40~50	180~186	30~540	250~800	5→10 MHz	34~55K → 100~111K	1.0 → 2.0 times as fast
80287 (original)	70~100	90~145	193~203	10~15	40~50	180~186	30~540	250~800	6→12 MHz	41~66K → 83~133K	1.2 → 2.4 times as fast
80387 (and later 287 models)	23~34	29~57	88~91	18	24	122~129	191~497	314~487	16→33 MHz	280~552K → 579~1100K	approx 10 → 20 × as fast
80486 (or 80487)	8~20	16	73	4	4	83~87	200~273	218~303	16→50 MHz	1.0M → 3.1M	approx 18 → 56 × as fast
Cyrix 6x86, Cyrix MII	4~7	4~6	24~34	2	4	59~60	117~129	97~161	66→300 MHz	11~16M → 50~75M	approx 320 → 1400 ×
AMD K6 (including K6 II/III)	2	2	todo	2	todo	todo	todo	todo	166→550 MHz	83M → 275M	approx 1500 → 5000 ×
Pentium / Pentium MMX	1~3	1~3	39	1 (0*)	1~4	70	17~173	19~134	60→300 MHz	20~60M → 100~300M	approx 1100 → 5400 ×
Pentium Pro	1~3	2~5	16~56	1 (0*)	1	28~68	todo	todo	150→200 MHz	30~75M → 40~100M	approx 1400 → 1800 ×
Pentium II / III	1~3	2~5	17~38	1 (0*)	1	27~50	todo	todo	233→1400 MHz	47~116M → 280~700M	approx 2100 → 13000 ×
Athlon (K7)	1~4	1~4	13~24	1 (0*)	1~2	16~35	todo	todo	500→2330 MHz	125~500M → 0.580~2.33G	approx 9000 → 42000 ×
Pentium 4	1~5	2~7	20~43	1 (0*)	todo	20~43	todo	todo	1.3→3.8 GHz	186~650M → 0.543~1.90G	approx 11000 → 34000 ×
Athlon 64 (K8)	1~4	1~4	13~24	1 (0*)	1~2	16~35	todo	todo	1.0→3.2 GHz	250~1000M → 0.800~3.2G	approx 18000 → 58000 ×

** An effective zero clock delay is often possible, via superscalar execution.*

[§] *The 5 MHz 8087 was the original x87 processor. Compared to typical software-implemented floating point routines on an 8086 (without an 8087), the factors would be even larger, perhaps by another factor of 10 (i.e., a correct floating point addition in assembly language may well consume over 1000 cycles).*

Manufacturers

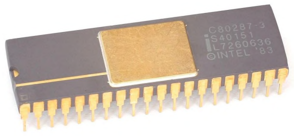
Companies that have designed and/or manufactured floating point units compatible with the Intel 8087 or later models include AMD (*287, 387, 486DX, 5x86, K5, K6, K7, K8*), Chips and Technologies (the *Super MATH* coprocessors), Cyrix (the *FasMath*, *Cx87SLC*, *Cx87DLC*, etc., *6x86*, *Cyrix MII*), Fujitsu (early *Pentium Mobile* etc.), Harris Semiconductor (manufactured *80387* and *486DX* processors), IBM (various *387* and *486* designs), IDT (the WinChip, *C3*, *C7*, *Nano*, etc.), IIT (the *2C87*, *3C87*, etc.), LC Technology (the *Green MATH* coprocessors), National Semiconductor (the *Geode GX1*, *Geode GXm*, etc.), NexGen (the *Nx587*), Rise Technology (the *mP6*), ST Microelectronics (manufactured *486DX*, *5x86*, etc.), Texas Instruments (manufactured *486DX* processors etc.), Transmeta (the *TM5600* and *TM5800*), ULSI (the *Math·Co* coprocessors), VIA (the *C3*, *C7*, and *Nano*, etc.), and Xtend (the *83S87SX-25* and other coprocessors).

Architectural generations

8087

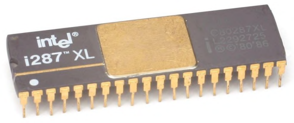
The *8087* was the first math coprocessor for 16-bit processors designed by Intel (the I8231 was older but designed for the 8-bit Intel 8080); it was built to be paired with the Intel 8088 or 8086 microprocessors.

80287



6 MHz version of the Intel 80287

The *80287* (*i287*) was the math coprocessor for the Intel 80286 series of microprocessors. Intel (and its competitors) later introduced an 80287XL, which was actually an 80387SX with a 287 pinout. The 80287XL contained an internal 3/2 multiplier so that motherboards which ran the coprocessor at 2/3 CPU speed could instead run the FPU at the same speed of the CPU. Other 287 models with 387-like performance were the Intel 80C287, built using CHMOS III, and the AMD 80EC287 manufactured in AMD's CMOS process, using only fully static gates.



Intel 80287XL.

The 80287 and 80287XL also worked with the 80386 microprocessor, and were initially the only coprocessors available for the 80386 until the introduction of the 80387 in 1987. Finally, they were also able to work with the Cyrix Cx486SLC. However, for both of these chips the 80387 was strongly preferred for its higher performance and the greater capability of its instruction set.

Intel's models included i80287 variants with specified upper frequency limits ranging from 6 up to 12 MHz. Later followed the i80287XL with 387 microarchitecture and the i80287XLT, a special version intended for laptops, as well as other variants.

80387

Intel 80287XLT.

The **80387** (**387** or **i387**) was the first Intel coprocessor to be fully compliant with the IEEE 754 standard. Released in 1987, a full two years after the 386 chip, the i387 included much improved speed over Intel's previous 8087/80287 coprocessors, and improved the characteristics of trigonometric functions. (The 80287 limited the argument range to plus or minus 45 degrees.)

Without a coprocessor, the 386 normally performed floating-point arithmetic through (slow) software routines, implemented at runtime through a software exception-handler. When a math coprocessor is paired with the 386, the coprocessor performs the floating point arithmetic in hardware, returning results much faster than an (emulating) software library call.

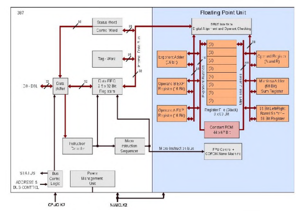
The i387 was compatible only with the standard i386 chip, which had a 32-bit processor bus. The later cost-reduced i386SX, which had a narrower 16-bit data bus, could not interface with the i387's 32-bit bus. The i386SX required its own coprocessor, the 80387SX, which was compatible with the SX's narrower 16-bit data bus.

16 MHz version of the Intel 80187

i387

i387SX

i387DX



i387 microarchitecture with 16-bit Barrel shifter and CORDIC unit

80187

The *80187* (*80C187*)[3] was the math coprocessor for Intel 80186 CPU. It is incapable of operating with the 80188, as the 80188 has a 8 bit data bus; the 80188 can only use the 8087. The 80187 did not appear at the same time as the 80186 and 80188, but was in fact launched after the 80287 and the 80387. Although the interface to the main processor was the same as the 8087, its core was that of the 80387, and was thus fully IEEE 754 compliant as well as capable of executing all the 80387's extra instructions.[4]

i487SX

80487

The **i487SX** was marketed as a floating point unit coprocessor for Intel i486SX machines. It actually contained a full-blown i486DX implementation. When installed into an i486SX system, the i487 disabled the main CPU and took over all CPU operations. The i487 took measures to detect the presence of an i486SX and would not function without the original CPU in place.[5]

80587

The **Nx587** was the last FPU for x86 to be manufactured separately from the CPU, in this case NexGen's Nx586.

References

[1] David Monniaux, *The pitfalls of verifying floating-point computations* (http://hal.archives-ouvertes.fr/hal-00128124/en), to appear in ACM TOPLAS

[2] *Numbers are taken from respective processors' data sheets, programming manuals, and/or optimization manuals.*

[3] CPU Collection - Model 80187 (http://www.cpu-info.com/index2.php?mainid=Overview&showm=8)

[4] http://www.datasheetcatalog.org/datasheet/Intel/mXryvuw.pdf

[5] Dictionary.com, ed. "Intel 487SX" (http://dictionary.reference.com/browse/Intel+487SX). . Retrieved 2010-10-17.

- Intel Corp., *IA-32 Intel Architecture Software Developer's Manual Volume 1: Basic Architecture*, order number 253665-017

External links

- Coprocessor.info : x86 Coprocessor development & history knowledge (http://www.coprocessor.info)
- Everything you always wanted to know about math coprocessors (http://wiretap.area.com/Gopher/Library/Techdoc/Cpu/coproc.txt)