HHC 2021

History of Floating Point Processors

Eric Smith

Note: this is a slightly updated version of the author's HPCC presentation from January 2021

History of Floating Point Processors

- Floating point arithmetic basics
- History of floating point arithmetic hardware
- Early floating point math coprocessors

Floating point arithmetic basics

- Floating point provides a way to represent approximations of real numbers
  - large range, typically spanning 75 or more orders of magnitude
  - limited precision, usually less than 20 decimal digits
  - represented using a relatively small number of bits, often 32 or 64

Floating point arithmetic basics

- The most common floating point formats use either radix 2 (binary) or radix 10 (decimal)
- Modern computers mostly use radix 2, which generally allows simpler and faster electronics
- Calculators mostly use radix 10

Floating point arithmetic basics

- In either radix, there are common fractions that cannot be precisely represented
- Negative powers of 10, and their multiples, e.g. 0.1, 0.2, 0.001, etc., cannot be precisely represented in binary floating point – **this is why calculators usually use radix 10**

Floating point arithmetic basics

“Floating point numbers are really just two small integers in a trenchcoat pretending to be a real number”

- @justarandomgeek

Floating point arithmetic basics

- A floating point number consists of three parts:
  - The sign (positive or negative) of the value
  - The **significand** (often incorrectly called “mantissa”) has deliberately limited bounds, [1.0, 2.0) for binary floating point, or [1.0, 10.0) for decimal, and represents the significant digits of the number
  - The exponent is an integer representing a positive or negative power of the radix

Floating point arithmetic basics

- The absolute value of the floating point number is the significand times the radix to the exponent power.
- In some cases, the sign and the significand may be stored in a joint signed representation, e.g., one's or two's complement for binary radix
- Zero may have a special representation, because the significand range doesn't include zero.
- Additional special representations may exist for infinities, denormalized numbers, and exception values (e.g., IEEE NaN, Not a Number)

Floating point arithmetic basics

- While floating point numbers are intended as a representation of real numbers, an IEEE single precision float can represent 0% of the real numbers within the range
- Since an IEEE double precision float can represent four billion times as many values as single precision, that brings us up to 0% of the real numbers within the range

Floating point history

The earliest computers to include a hardware implementation of floating point arithmetic were **electromechanical**:

- 1914: Leonardo Torres y Quevedo designed an electro-mechanical version of Charles Babbage's Analytical Engine, and included floating point arithmetic
  - Quevedo's paper contained the first published description of floating point arithmetic
  - Design was considered theoretical rather that practical, and was not built
- 1938: The Zuse Z1, designed and built by Konrad Zuse, used a 22-bit binary floating point format
- 1946: The Zuse Z4 was the first commercially sold computer with floating point arithmetic, and used a 32-bit binary floating point format

Floating point history

- Electronic computers were fast enough that it was possible to implement floating point arithmetic entirely in software, saving hardware cost, with adequate performance for many general-purpose tasks
- For computation-intensive tasks, it was still advantageous to have floating point hardware, which was typically one hundred times faster than software floating point

Floating point history – big iron

- 1954: The IBM 704 electronic (vacuum tube) computer was the first mass-produced electronic computer with hardware floating point (radix 2)
- 1959: The IBM 1620 is an early example of a computer for which floating point hardware (radix 10) was **optionally** available
- 1970s and 1980s: many minicomputers (e.g., DEC PDP-11) offer optional floating point processors, but they consist of many circuit boards, so are fairly expensive

Floating point history – Early HP Calculators

- 1968: The HP 9100A was HP's first desktop calculator, and used radix 10 floating point, implemented by a combination of hardware and microcode
- 1972: The HP 35 was the first handheld scientific calculator, and used radix 10 floating point, implemented mostly by microcode, using radix 10 integer arithmetic hardware

Floating point history – PC era

- 1978: Intel introduces the 8086 microprocessor, forerunner of modern “x86” personal computers
- 1980: Intel introduces the 8087 numeric coprocessor, using draft standard IEEE 754 binary floating point
- 1981: IBM introduces the IBM Personal Computer, based on the 8088 (an 8-bit bus version of the 8086). A socket on the motherboard allowed installation of an optional 8087 numeric coprocessor
- 1989: Intel introduces the 80486 processor, the first x86 family microprocessor to include on-die floating point

Floating point history – standards

- 1985: The IEEE 754 standard provided the first vendor-independent standard for representation of binary floating point numbers, and is now used by almost all modern computers.
- IEEE 754 single precision (“binary32”) has about 7 decimal digits of precision, and a range of about $\pm 10^{38}$
- IEEE 754 double precision (“binary64”) has about 15 decimal digits of precision, and a range of about $\pm 10^{1022}$

Floating point history - standards

- 1987: The IEEE 854 Standard for Radix-Independent Floating Point provided the first vendor-independent representation for decimal floating point (or other non-binary bases)
- 1984: The HP-71B handheld computer implemented decimal floating point using the IEEE 854 draft standard, with 12 significant digits, and a range of $\pm 10^{499}$
- Many HP-engineered handheld calculators since 1984 use arithmetic subroutines from the HP-71B, but do not implement the full standard as the 71B did.
- 2008: The IEEE 754 standard is revised to include decimal floating point

Early floating point coprocessors

- Some of the earliest general-purpose floating point coprocessor integrated circuits (mid-1970s) were based on calculator chips
  - code in masked ROM modified to interface to a host processor instead of (or in addition to) a keyboard and LED display
  - **very slow**, on the order of 10 ms for arithmetic, to close to 1 second for transcendental functions

Early floating point coprocessors

- Calculator-chip-based floating point coprocessors:
  - 1976: Texas Instruments TMS1018 “Number Cruncher”
    - no technical documentation known to still exist
    - presumed to be a mask-programmed TMS1000 series microcontroller
    - PMOS (very slow)

Early floating point coprocessors

- Calculator-chip-based floating point coprocessors:
  - 1977: National Semiconductor MM57109 “Number Oriented Processor”
    - based on MM5799 microcontroller
    - PMOS
      - made interfacing to NMOS microprocessors relatively difficult
        - not all signals were TTL compatible
        - some timing requirements were tricky

Early floating point coprocessors

- Calculator-chip-based floating point coprocessors:
  - 1977: National Semiconductor MM57109 “Number Oriented Processor” (continued)
    - very slow, slower than software floating point on some contemporary microprocessors
    - despite slowness, may still have been cost effective, as it reduced the amount of memory needed on the main processor (in 1977, memory was expensive)

Early floating point coprocessors

- Calculator-chip-based floating point coprocessors:
  - 1977: National Semiconductor MM57109 “Number Oriented Processor” (continued)
    - was described in electronics hobbyist magazines of the time, e.g. Byte and Radio Electronics
    - hobbyists built their own interfaces to microcomputers
    - SWTPC offered a “MP-N” kit for their MC6800-based microcomputer

Early floating point coprocessors

- Calculator-chip-based floating point coprocessors:
  - 1980: National Semiconductor MM57409 “Super Number Cruncher”
    - based on COP440 microcontroller
    - NMOS
      - could potentially have been faster than PMOS, but in this case wasn't
      - made interfacing easier than the PMOS MM57109, due to improved electrical characteristics and timing
    - last dying gasp of calculator-chip based floating point processors

Early floating point coprocessors

- 1976: DEC KEV11-A for PDP-11/03, LSI-11
  - is a microcode ROM, not really a coprocessor
  - microcoded floating point is faster than software floating point, but not as fast as floating point hardware
  - all earlier DEC floating point “processors” consisted of multiple printed circuit boards and hundreds of integrated circuits, making this DEC's first low cost floating point hardware option

Early floating point coprocessors

- 1978: AMD Am9511
  - designed from the ground up as a floating point chip, not derived from a calculator chip
  - microcoded
  - single-precision 32-bit binary floating point, but not IEEE compatible as it predates the standard
  - includes transcendental functions (logarithms and exponentials, trigonometric and inverse trigonometric functions)
  - much faster than the calculator chips (typ. 50x)
    - arithmetic usually under 100 us
    - sin, cos, tan usually under 2.5 ms

Early floating point coprocessors

- 1978: AMD Am9511 (continued)
  - interfacing still a bit tricky
    - NMOS electrical characteristics are fairly easy
    - Timing is still tricky (chip select setup time, /PAUSE output)
  - third-party add-on cards were available for microcomputers, e.g. Apple II
  - second-sourced by Intel as 8231

Early floating point coprocessors

- 1980: AMD Am9512
  - similar to Am9511, but uses IEEE floating point
  - single and double precision
  - no transcendental functions (perhaps not a big enough microcode ROM?)
  - includes integer conversions
  - NMOS, slightly easier interfacing than Am9511
  - second-sourced by Intel as 8232

Early floating point coprocessors

1980: Intel 8087

- glueless interface to 8086 or 8088, difficult to use with any other main processor
- purpose-built for IEEE floating point
- HMOS process (high-density NMOS)
- microcode ROM stores two bits per cell (like modern MLC NAND flash)

Early floating point coprocessors

1980: Intel 8087 (continued)

- fastest floating point chip at the time
- addition in 17 us
- single-precision multiplication in 19 us
- double-precision multiplication in 27 us
- tangent in 90 us

Summary

Evolution of floating point hardware in general-purpose computers:

- 1940s through 1960s: originally a standard feature of “big” computers
- 1960s through 1970s: became optional, but expensive due to use of many circuit boards and integrated circuits
- mid 1970s: single-chip floating-point coprocessors based on microcontrollers
- late 1970s through 1980s: purpose-designed floating point coprocessors
- 1990s to present: floating point included as a standard feature of most general-purpose microprocesors