

Motorola 68000

Drake Vidkjer

November 29, 2017

The Motorola 68000 architecture, commonly referred to as the m68k or simply 68k is a historic CPU architecture first introduced in 1979 to compete with Intels x86 CPUs including the 8086, 8088, and 80286. Unlike its competitors which at the time were mainly 16 bit, the Motorola 68000 is a 32 bit CISC architecture.[1] While originally produced in HMOS technology, over the years CPUs in the family have been produced in CMOS by third parties, and at varying clock speeds such as 8, 10, 12.5, 16.67MHz and up to 50MHz on later models. The 68k family was originally intended as a desktop processor, appearing in early Atari, Amiga, and Macintosh computers through the 1980s and 90s but was replaced in favor of new architectures such as PowerPC, also developed by Motorola. While mainly being replaced in the desktop field, 68k processors are still found in embedded systems, and were also used in many printers throughout the 1990s including Adobes first post-script printers. The 68k architecture was also regarded as one of the best architectures for running C code, and as a result was very influential in the UNIX market being found in SUN and SGI IRIS workstations.[2] While still being produced today, development of the 68000 architecture ceased in 1994, with Motorola introducing their new architecture in conjunction with IBM and Apple, PowerPC.

The original 68000 is based on a 16/32 bit CISC instruction set, with a 32 bit instruction set, internal data bus, and registers in addition to a 16 bit ALU and 24 bit external data bus. This external data bus was expanded to 32 bits on later CPUs.[3] The 68000 started with 56 instructions with a minimum size of 16 bits each. The architecture developers attempted to create a nearly orthogonal instruction set design. Orthogonal instruction sets are designed to allow almost all address modes to be available to almost all instructions.[4] Meaning that an assembly programmer needed to only remember a relatively few instructions and few addressing modes instead of remembering multiple instructions that functionally do the same thing. For example, additions instructions where the operands are loaded from registers, and from memory are the same, the addressing mode is simply different. From an assembly level, this orthogonality holds true, but from a bit level it is not. This was a compromise implemented by the architecture designers to allow for easy assembly programming while also allowing them to fill in the instruction table to reduce size. Some idiosyncrasies about the 68k instruction set include that fact that it does not include an equivalent to x86s CPUID, and the fact that because the instruction set is orthogonal, the number of cycles for an instruction can depend on the source of the instructions data. Most instructions are also dyadic, meaning they have a source, and destination, and the result of the operation is stored in the destination.

The Motorola 68000 is a big endian architecture, with 8 32 bit data registers, 7 32 bit address registers, and a 32 bit program counter. The last address register also acts as the stack pointer as well.[5] For the time when the 68k architecture was initially released, this was considered a good number of registers. Allowing for fast interrupt response due to only 15 registers needing to be moved and the fact that most calculations could be done within the CPU not needing to go to main memory to retrieve data. The 68k architecture also includes a status register with bit flags such as carry, zero, extend, overflow, and negative.[5] Motorola 68k CPUs have two privileged levels when running programs, these being user and supervisor mode. The main difference between the two modes is that code being run in supervisor mode has access to interrupt level controls, while code running in user mode does not. While user code does not have access to interrupt controls, it does have access to the supervisor bit which is also stored in the status register. One additional difference between user and supervisor privilege modes is that supervisor has a separate stack pointer, permitting a small level of multitasking. Starting with the 68020, Motorola added a true cache of 256 bytes and a three stage pipeline not seen in previous processors in the family.

In regards to memory, the original 68000 had an external 24 bit address bus meaning that it can only address 16MB of memory. On later 68k CPUs the external address bus was expanded to 32 bits allowing for a theoretical 2GB to be addressed. While the external memory bus on the original 68000 was only 24 bits, it maintained a 32 bit internal address bus.[2] The architecture designers did this in order to allow for forward compatibility of software. While this was the hope, many software applications took advantage of the extra 8 internal address bits in the 68000, meaning that the software could potentially have problems running on later processors. The 68000 did not support virtual memory, though some UNIX workstations were produced that used multiple 68000s and interrupts to allow for working virtual memory. Starting with the 68020, the architects implemented support for multiple co-processors through TAS, CAS, and CAS2 with the 68020 being able to support up to 8 co-processors.[1]

When compared to the original Intel 8086 processor, even the 68000 can be subjectively described as better; having larger address widths, and generally being 32 bit. As the 68k family progressed, it stayed in line as a formidable adversary to IA-32 processors. Once the IA-32 family was introduced, Motorola did things such as expanding the external address bus width, and adding a 32 bit ALU along with virtual memory support among other features. While both architectures provide general purpose registers, the 68k generally provides more, with x86 tending more towards reduced registers and cache instead. Both architectures use CISC instruction sets and some sort of cache. While IA-32 started out as having a 32 bit external address bus, the 68k family did not, this was expanded later on. An ironic difference between the two architectures is the fact that x86 was initially developed for embedded systems, while the 68k was meant to be a desktop processor from its inception. Today though the roles are flipped, with x86 rarely being found in embedded systems and the 68000 architecture being almost completely relegated to this duty.[1] A philosophical design difference between the two architectures is the fact that the 68k was designed for forward compatibility, as seen in the sizes of the internal and external address bus while the x86 architecture has been designed for backwards compatibility. One final difference is that of endianness; the 68000 family is big endian while IA-32, like all of x86 is little endian.

While the Motorola 68k family may not be anything impressive by today's standards, when it was introduced the architecture was a major contender to x86. CPUs from the 68k family were found in many desktops of the 1980s and 1990s and still hold a spot in my heart today, as I continue to own, and refurbish Macintoshes from the 68k era. Even with the popularization of ARM, and its takeover of printer processors people still find uses for the 68000 in embedded systems and other applications.

References

- [1] *Motorola 68000*, Nov. 2017. [Online]. Available: https://en.wikipedia.org/wiki/Motorola_68000.
- [2] *M68000*, Oct. 2017. [Online]. Available: <http://segaretro.org/M68000>.
- [3] G. Shvets, *Motorola 68000 microprocessor family*, Oct. 2017. [Online]. Available: <http://www.cpu-world.com/CPUs/68000/index.html>.
- [4] E. MacDonald, *Instruction set architecture ee 3376*. [Online]. Available: http://www.ece.utep.edu/courses/web3376/Notes_files/ee3376-isa.pdf.
- [5] *68000 programmers reference manual*. [Online]. Available: <https://www.nxp.com/docs/en/reference-manual/M68000PRM.pdf>.
- [6] *Byte magazine volume 11 number 09 - the 68000 family*. [Online]. Available: https://archive.org/stream/byte-magazine-1986-09/1986_09_BYTE_11-09_The_68000_Family#page/n191/mode/1up.