

# 68080 Core



## Overview

**68080** is a new CPU that is code-compatible with the **Motorola 68k** processors.

It is an affordable CPU for your classic Amigas (or Amiga compatibles) that is **much faster than the fastest 68060**. It is compatible with all CPUs and [FPUs](#) of the Motorola 68000 family. <sup>1)</sup> It is designed according to modern CPU standards, is fully written in **VHDL**, and is intended to run on [Altera Cyclone FPGAs](#).

**AmigaOS 3.x** currently recognizes this new CPU as a **68040**.

---

## Advantages

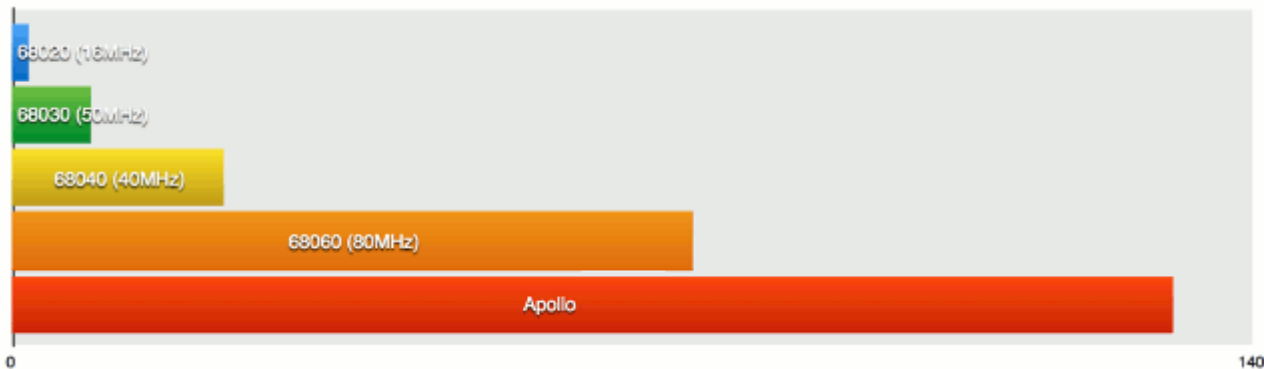


- Fast, available, affordable, compatible
  - Superscalar, Hyper-threaded
  - Executes up to 4 instructions per clock cycle
  - Two address calculation engines
  - Two integer execution engines
  - Market-leading code density
  - Optimal cache utilization
  - Separate data and instruction caches, supporting concurrent fetch/read/write per clock cycle
  - Automatic memory prefetching
  - Memory stream detection
  - Store buffer
  - Branch prediction
- 

## Performance



68080 CPU, when embedded on FPGA chips such as Cyclone III and Cyclone V, is much faster than the fastest 68060. This is possible because of the advanced design of the APOLLO Core, because of large and fast instruction/data caches, and because of the very fast read/write memory access.



Below are some benchmarks done with the [MiniBench<sup>2\)</sup>](#) tool for some classic Amiga machines. All tests show that the Core is much faster than a fast A1200 + 68060 @ 66Mhz - scores are all confirmed by [SysInfo](#), [SysSpeed](#) and [AIBB](#) benchmarking tools.

System	Accelerator	CPU	Frequency	Performance	MiniBench Points
Amiga 600	ACA620	68020	16 MHz	4	+
FPGA Arcade	TG68-based	68020	n/c	9	++
Amiga 1200	ACA1231	68030	41 MHz	11	+++
Amiga 1200	Blizzard1230	68030	50 MHz	11	+++
Amiga 4000		68040	25 MHz	13	+++
Amiga 4000		68040	40 MHz	25	++++++
Amiga 1200	MK2	68060	66 MHz	65	+++++
Amiga 600	Vampire 600 V2	SILVER6	x15	173	+++++

## Features



The 68080 CPU covers all the instructions and architectural designs of the classic MC680x0. It also brings some new features that make it faster. The table below shows the architectural improvements for previous 680x0 CPUs and this new one.

Feature	68000	68020	68030	68040	68060	68080
68K ISA	X	X	X	X	X	X
Extended EA-modes		X	X	X	X	X
BitFields		X	X	X	X	X
64Bit MUL		X	X	X		X
64Bit DIV		X	X	X		X
Instruction-Cache		256B	256B	4kB	8kB	16kB
Data-Cache			256B	4kB	8kB	32kB
Fully pipelined				X	X	X

Feature	68000	68020	68030	68040	68060	68080
<a href="#">Superscalar</a>					X	X
Store Buffer					X	X
Static Branch Prediction				X	X	X
Dynamic Branch Prediction					X	X
Branch Target Cache					X	X
Conditional Rewrite						X
Linkstack						X
<a href="#">Instruction Bonding</a>						X
<a href="#">Instruction Fusing</a>						X
64-Bit Support						X
3-opp instructions						X
Memory-Prefetch						X
Multimedia Extension						X
Selfmodify Support	X					X
Peak Inst/Cycle	0.25	0.5	0.5	1	2	4

## Supported instructions



All integer and floating-point instructions from MC68000 to MC68060 are supported.

- [68080 instructions](#)
- [68080 FPU instructions](#)
- [68080 asm code samples](#)

Unsupported instructions	Notes
TAS2	Never used on AmigaOS
MMU-related	Not implemented

## AMMX: New 64-bit instructions

These instructions are very similar to the SSE / AltiVec versions of them.

68080 lets you use EA as 1 source and to update 3rd registers - so normally not overwrite source registers.

- Quick Reference: [PDF](#)
- Full Reference: [TXT](#) / [AmigaGuide](#)

You are here: [start](#) » [apollo\\_core](#)

- <sup>1)</sup> Few rare instructions are currently not supported.
- <sup>2)</sup> Other than the latest version, you can also grab the [original MiniBench](#), from before the APOLLO-Team started maintaining it.

From:  
<https://wiki.apollo-accelerators.com/> - **Apollo Accelerators**

Permanent link:  
[https://wiki.apollo-accelerators.com/doku.php/apollo\\_core:start?rev=1582881296](https://wiki.apollo-accelerators.com/doku.php/apollo_core:start?rev=1582881296)

Last update: **2020/08/02 12:37**

