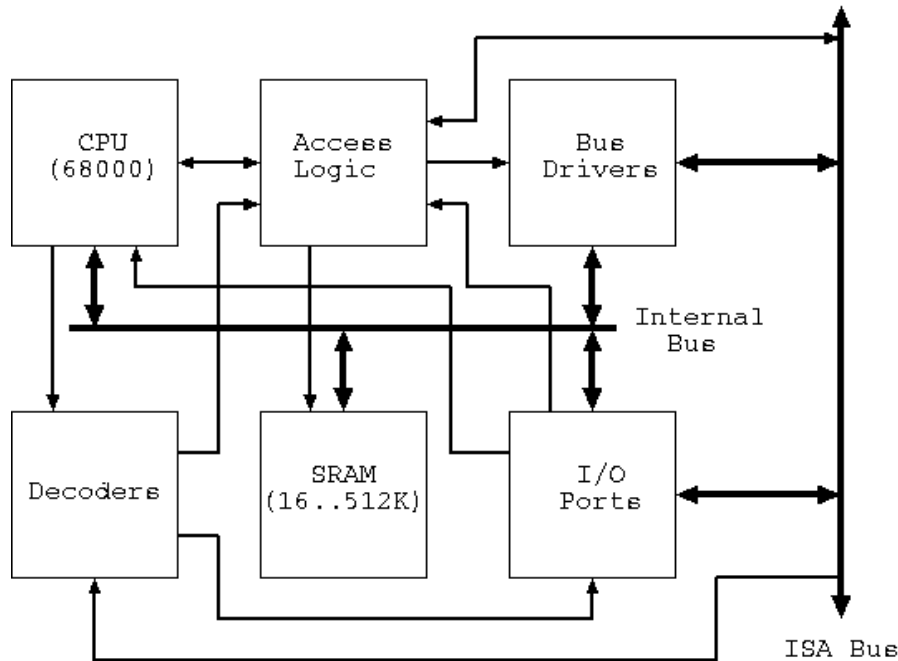


The PcPar68000

Parallelism for Everyone...

PcPar68000 is an MC68000-based parallel, PC-hosted computer whose concept was originally published by the german computer magazine **mc** in 1989. The idea was to place a simple 68000-based single board computer on a standard ISA board. You could place as much boards in your computer as you have free slots, thus achieving speedup by parallelism. The following figure shows the basic layout of such a board.



The heart of the board is (of course) the MC68000 microprocessor. It was chosen because of its good price/performance ratio and the fact that it was already well introduced. The 12MHz-version was used because 68000's rated at 16MHz were far more expensive at that time (about a factor of 3...). Nevertheless, overclocking the CPU to 16MHz wasn't a problem - Motorola CPUs traditionally seem to have a lot of headroom in this respect.

The CPU's address space is split into two areas by a simple address decoder. The bottom fifteen megabytes were assigned to RAM, and the topmost megabyte was assigned to special I/O ports. The address decoding was *very* simple - both the RAM and I/O-ports appeared multiple times in the 68000's address space.

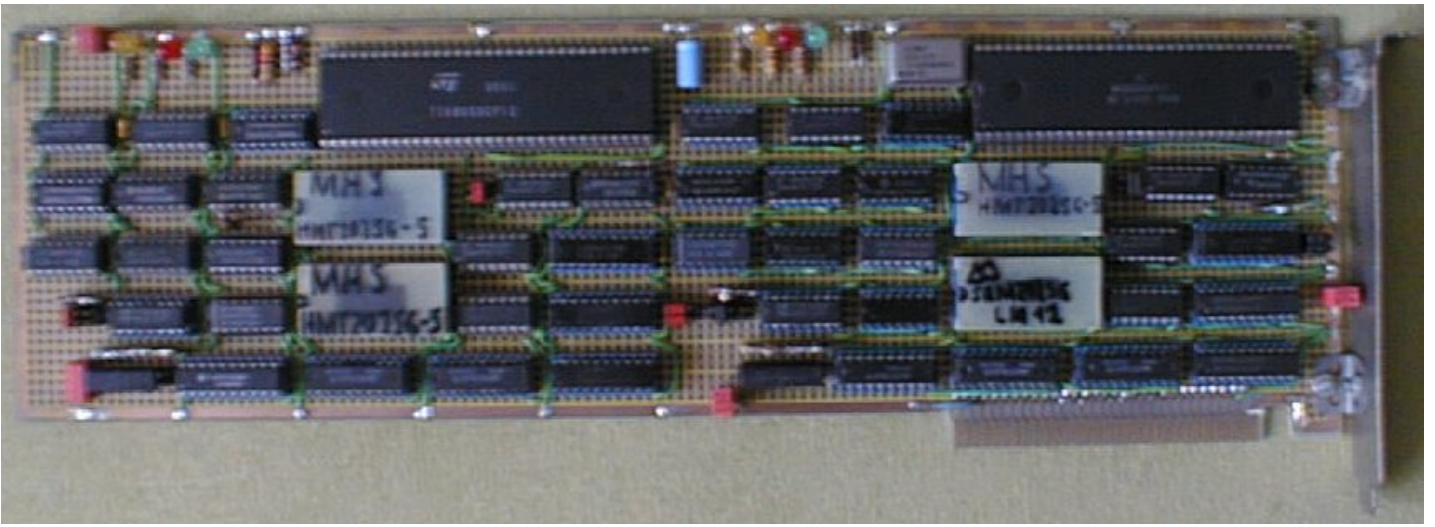
The board does not have a ROM to boot from. Instead, the 68000 is kept constantly in reset state after you turn the PC on. In this state, the PC has full access to the on-board RAM to write a program into it. It then releases the reset line (via an I/O port going from the ISA bus to the 68000) et voilà - we have a running 68000! Theoretically, the access logic allows memory accesses from the PC even when the 68000 is running. This however turned out not to work reliably enough, so the practical way of communicating data from the 68000 to the PC was to set a special value in an I/O port that is readable via the ISA bus. The PC periodically scans the I/O ports of all boards (interrupts would have been a nice extension...). If it finds a 68000 willing to communicate, it puts the 68000 into reset state, does the transfer and restarts the 68000. The 68000 previously had set the reset vector to a resume routine that would restore the register set and jump back to the application.

The PC itself sees the memory banked into a memory area between 640K and 1M with a bank size of 64K. The upper address bits are delivered by an I/O port that also holds the reset flag and a bit that allows to disable the mapping of the board's memory into the PC's address space. Since each board has such a mapping bit, you can e.g. simultaneously write the program to all boards and later read the results from individual boards.

The nice thing about the whole design was that it was completely built from standard components - the 68000, the SRAMs, and otherwise only simple TTL chips. No ASICs, no PALs, so the whole design was very cheap and components were easy to obtain.

However, as you all probably know, free ISA slots are a very scarce resource, and having only one board per slot puts a very quick limit to all dreams of a massively-parallel computer. So I had to reengineer things a bit...

The first step was to make a board that could hold *two* CPUs. The original PCB was a half-length card, so there was no problem placing the whole circuitry twice on a single full-length prototype board. The only shared component was the 12MHz crystal...



I explicitly apologize for my bad qualities as photographer ;-). This picture also shows that each board has three LEDs for diagnostic purposes:

- The green one is on when the 68000 is running, i.e. not in reset state.
- The red one is on when the board's memory is mapped into the PC address space.
- The yellow one may be freely turned on and off by the 68000 - I mainly used it to signal when the 68000 ran into an exception...

The next step was to go even further and place the CPU boards themselves in a separate 19 inch rack. Plugged into the ISA bus was only a simple host adapter holding bus buffers and part of the address decoders. A single CPU board now fitted on a single Euro card without programmable logic:



This board type again had the three LEDs. It has a 68010 (in case you couldn't read it ;-)) which gives a nice boost for loops and multiply/divides, but it was awfully difficult to obtain at clocks over 10MHz (and prohibitively expensive).

Finally, below is a photograph of the whole rack with one of the cards pulled out. I haven't turned on the machine for years, but it still has a place of honour on my desk next to the K6/200 PC. The PC is now probably faster than the 9 68000's altogether, but at that time it was a nice speedup compared to a 8088 with 8 MHz. Even with software emulation, the 68000 achieved almost a third of an 8087's floating point performance.



Pulled out is the last board I made for the PcPar68000: The CPU is in a PLCC case to save board space, and most of the random logic is shrunk into three 16V8 GALs. This gave space for a second pair of RAM sockets (this board has 320K SRAM) and for a 68881 coprocessor which was however never realized... The total power consumption, measured via a simple analog ammeter, was about 3A@5V.

© 1998 [Alfred Arnold, alfred@ccac.rwth-aachen.de](mailto:alfred@ccac.rwth-aachen.de)