68040pc Specification

A Personal Computer Platform

Anthony Guerrero

Document Revision 0.0.1

1 System Overview

1.1 Design Goal

The primary objective of the 68040pc is to construct a personal computer platform that aligns with the capabilities and aesthetics of computing systems from the 1990s. The system is not merely a reflection of retro hardware but a modern interpretation emphasizing complete transparancy, user understanding, and modifiability. The design principles prioritize open-source documentation, source code, and hardware description language (HDL) to empower users and enthusiasts to understand, modify, and expand upon the system.

1.2 Key Features

- **Open Architecture:** An open-source ethos is at the core of the project. All documentation, designs, and code associated with this system will be freely accessible and modifiable by the end-users and community.
- **MC68040 Processor:** Serving as the heart of the system, this 32-bit microprocessor offers a blend of performance and functionality suitable for a wide range of computing tasks. It's Instruction Set Architecture (ISA) and built-in MMU lends itself well to implementing UNIX-like operating systems.
- **Flexible Expansion:** The platform is designed with expansion and modularity in mind, featuring an "Expansion Bus" architecture to facilitate the addition of new components. The bus dynamically assigns memory space to bus members, bypassing address conflicts and setup jumpers.
- **Integrated FPGA:** Utilizing the iCE40-HX4K FPGA, with full access to the CPU bus, most glue logic and peripheral controllers can be implemented and modified after the design is finalized.
- **Dynamic RAM Controller:** With four 72-pin SIMM slots, the platform supports up to 512MB of main system memory. (3.3v only!)
- **Standard Form Factor:** The PCB will be sized to fit in a pre-ATX form factor called "Baby AT". This allows for eight expansion slots, a DIN-5 sized cutout for a keyboard, and simple power management.

1.3 System Philosophy

Beyond the technical specifications, this project embodies a philosophy of transparency, education, and user empowerment. The system is not just a computing platform but a testament to the ethos that technology should be understandable, modifiable, and, above all, open. Whether for education, nostalgia, or innovation, this platform seeks to be a canvas for users to explore the intricacies of computing, reminiscent of an era when personal computing was just blossoming.

2 Hardware

2.1 CPU

The Motorola MC68LC040 was chosen because I found it in a box. The $\overline{\text{MDIS}}$ and $\overline{\text{IPLx}}$ lines are tied high by pull-up resistors during normal operation, which has the effect of enabling the small output buffers and non-multiplexed bus mode on the 68040 CPU. Designing the motherboard with this in mind makes the system also compatible with the standard 68040.

2.2 SRAM

There are 512KB of SRAM for the boot ROM to use. During boot, information about the hardware installed in the system is collected into a device attribute table and is communicated to the operating system.

2.3 DRAM

Four 72-pin SIMM slots provide the bulk RAM for the system. Each SIMM's $\overline{\text{RAS}}$ lines are tied together and sent to the memory controller, $\overline{\text{CAS}}$ lines are routed separately. The system should support commodity 5V 72-pin SIMMs which were available for the 486 and early Pentium platforms.

The SIMMs I have (Keystron MK16DS3232LP-50) have (16) 16M × 4-bit DRAMs per module. The module is layed out as listed in Figure 1, the DRAM chip numbers are listed from 0-7 for the front side and 8-15 for the back side. I've ordered the table so Bank 0 is listed first in order of Data Line. Next, Bank 1 is listed in the same order.

2.4 Boot ROM

The boot ROM program is stored on the same SPI EEPROM thats used to configure the FPGA. Accesses to ROM space are routed through the FPGA, which performs the required access to the SPI ROM and returns the data back to the CPU. This was done to facilitate rapid prototyping by providing a fast and available interface to update the ROM.

2.5 I2C Bus

I2C is used on-board for communication with the Real-Time Clock (DS1307) and other devices. (Temperature sensors/Fan controllers, etc.)

3 Memory Map

The memory map of the system is mostly organized into 512MB sections, using the `A31`, `A30`, and `A29` signals to select them. The system boot ROM is mapped at `$00000000-$1FFFFFFF`, followed by DRAM and SRAM areas.

Figure 1: Keystron MK16DS3232LP-50 module layout

DRAM Chip #	CAS Line	RAS Line	Data Lines
0	$\overline{\text{CAS0}}$	$\overline{\text{RAS0}}$	D0-D3
2	$\overline{\text{CAS0}}$	$\overline{\text{RAS0}}$	D4-D7
4	$\overline{\text{CAS1}}$	$\overline{\text{RAS0}}$	D8-D11
6	$\overline{\text{CAS1}}$	$\overline{\text{RAS0}}$	D12-D15
1	$\overline{\text{CAS2}}$	$\overline{\text{RAS2}}$	D16-D19
3	$\overline{\text{CAS2}}$	$\overline{\text{RAS2}}$	D20-D23
5	$\overline{\text{CAS3}}$	$\overline{\text{RAS2}}$	D24-D27
7	$\overline{\text{CAS3}}$	$\overline{\text{RAS2}}$	D27-D31
8	$\overline{\text{CAS0}}$	$\overline{\text{RAS1}}$	D0-D3
10	$\overline{\text{CAS0}}$	$\overline{\text{RAS1}}$	D4-D7
12	$\overline{\text{CAS1}}$	$\overline{\text{RAS1}}$	D8-D11
14	$\overline{\text{CAS1}}$	$\overline{\text{RAS1}}$	D12-D15
9	$\overline{\text{CAS2}}$	$\overline{\text{RAS3}}$	D16-D19
11	$\overline{\text{CAS2}}$	$\overline{\text{RAS3}}$	D20-D23
13	$\overline{\text{CAS3}}$	$\overline{\text{RAS3}}$	D24-D27
15	$\overline{\text{CAS3}}$	$\overline{\text{RAS3}}$	D27-D31

4 Expansion Bus

The system provides a Zorro III-inspired expansion slot bus, the main feature of which is dynamic memory mapping of add-in devices.

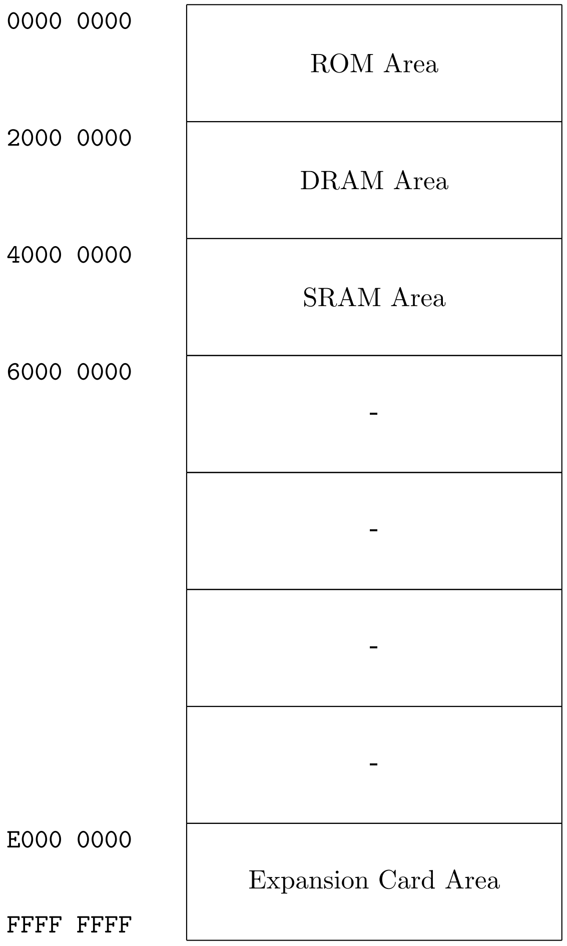
4.1 Design Goals

- **Expansion and Flexibility:** A primary aim of the expansion bus is to ensure that the design does not inherently limit future enhancements or adaptations.
- **Performance:** The bus shouldn't unnecessarily constrain the bandwidth of any peripheral that connects to it, including the bus-master. In practice, this means that it should pretty much be a minor extension to the local bus.
- **Support for Varied Devices:** The bus should accomodate any device which can interface with the MC68040's local bus. Unfortunately, this means 16 or 8-bit devices will need logic on the card to support "resizing" the accesses.
- **Reliability:** Ensure robust operation and avoid complications that might make the system prone to errors.

4.2 Bus Architecture

The design of the expansion bus draws heavily from the Zorro III expansion bus designed for the Amiga 3000 and 4000 series of computers, but differs in many key ways which make it incompatible.

Figure 2: System Memory Map



4.2.1 Card Configuration

Cards must ground the $\overline{\text{CDET}}$ line to be included in the bus, and therefore is a requirement for complying with this bus specification. If the card doesn't pull this line down, or there isn't a card plugged into an expansion slot, the $\overline{\text{CFGIN}}$ line will be bridged to the $\overline{\text{CFGOUT}}$ line for that expansion slot.

4.2.2 Bus Cycles

4.2.3 Arbitration/Mastering

4.2.4 Cache Support

While the expansion bus doesn't have any cache consistency mechanisms for managing caches between several caching bus masters, it does allow cards that absolutely must not be cached to assert a cache inhibit line, $\overline{\text{CI}}$, on a per-cycle basis. This cache management is mainly useful for support of I/O and other devices that shouldn't be cached.

4.2.5 Interrupts

A card supporting interrupts has on-board registers to store one or more vector numbers. The numbers are obtained from the OS by the device driver for the card and the card/driver combination must be able to handle the situation in which no additional vectors are available.

4.3 Signal Description

4.3.1 Power Connections

- **Digital Ground (GND)** This is the digital supply ground used by all expansion cards as the return path for all expansion supplies.
- **Main Supply (+5VDC)** This is the main power supply for all expanion cards. (Define power specification)

4.3.2 Clock Signals

- **TBD...**

4.3.3 System Control Signals

- **Hardware Bus Error**
- **System Interrupts**

4.3.4 Slot Control Signals

- **Configuration Chain ($\overline{\textbf{CFGIN\textsc{n}}}$, $\overline{\textbf{CFGOUT\textsc{n}}}$)** The slot configuration mechanism uses the bus signals $\overline{\text{CFGOUT\textsc{n}}}$ and $\overline{\text{CFGIN\textsc{n}}}$, where "N" refers to the slot number. Each slot has its own version of both signals, which make up the *configuration chain* between slots. Each subsequent $\overline{\text{CFGIN}}$ is a result of all previous $\overline{\text{CFGOUT}}$s, going from slot 0 to the last slot on the expansion bus.

During the autoconfiguration process, an unconfigured card responds to the address space X if its $\overline{\text{CFGINn}}$ is asserted. All unconfigured cards start up with $\overline{\text{CFGOUTn}}$ negated. When configured, a card will assert its $\overline{\text{CFGOUTn}}$ which results in the $\overline{\text{CFGINn}}$ of the next slot being asserted. Backplane logic automatically passes on the state of the previous $\overline{\text{CFGOUTn}}$ to the next $\overline{\text{CFGINn}}$ for any slot not occupied by a card.

- **Card Detect ($\overline{\textbf{CDET}}$)** This signal is to always be attached to ground by cards to allow them to participate in the expansion bus. This signal is part of the backplane circuitry that allows the configuration chain to function.

4.3.5 DMA Control Signals

- **TBD...**

4.3.6 Address and Related Control Signals

- **Address Bus ($\mathbf{A_0}$-$\mathbf{A_{31}}$)** This is the expansion address bus, which is driven by the bus master.

4.3.7 Data and Related Control Signals

- **Data Bus ($\mathbf{D_0}$-$\mathbf{D_{31}}$)** This is the expansion data bus, which is driven by either the master or the slave when "" is asserted by the master. It's valid for reads when $\overline{\text{DTACK}}$ is asserted by the slave.

- **Transfer Acknowledge ($\overline{\textbf{TA}}$)** This signal is used to normally terminate an expansion bus cycle. The slave is always responsible for driving this signal. For a read cycle, it asserts $\overline{\text{TA}}$ as soon as it has driven valid data onto the bus. For a write cycle, it asserts $\overline{\text{TA}}$ as soon as it's done with the data.

- **Transfer Burst Inhibit/Transfer Cache Inhibit ($\overline{\textbf{TBI}}$/$\overline{\textbf{TCI}}$)** This line is asserted at the same time as $\overline{\text{TA}}$ to indicate to the bus master that the cycle must not be cached, or that that the cycle can't be completed as a burst transfer.

4.4 Electrical Specifications

4.4.1 Standard Signals

The majority of signals on the bus are in this group. These are the bussed signals, driven actively on the bus by F-series (or compatible) drivers, usually tri-stated when ownership of the signal changed for master and slave, and generally terminated with a 220Ω/330Ω thevenin terminator.

A_0-A_{31}	D_0-D_{31}	R/$\overline{\text{W}}$
FC_0-FC_2		

4.4.2 Clock Signals

4.4.3 Open Collector Signals

Many of the bus signals are shared via open collector or open drain outputs rather than via tri-stated signals. A backplane resistor pulls these lines high, cards only drive the line low.

$\overline{\text{TA}}$
$\overline{\text{RESET}}$
$\overline{\text{TCI}}/\overline{\text{TBI}}$
$\text{R}/\overline{\text{W}}$
$\overline{\text{TEA}}$

4.4.4 Non-bussed signals

$\overline{\text{CFGIN}\textsc{n}}$
$\overline{\text{CFGOUT}\textsc{n}}$
$\overline{\text{IRQ}\textsc{n}}$
$\text{R}/\overline{\text{W}}$

4.4.5 Slot Power Availability

The system power for the expansion bus is based on the slot configurations. A backplane is always free to supply extra power, but it must meet the minimum requirements specified here. All cards must be designed with the minimum specifications in mind, especially the tolerances.

Pin	Supply
X, X	+5 VDC ±5% @ 2A
X	+12 VDC ±5% @ 500mA

4.5 Mechanical Specifications

4.6 Expansion Slot Pin Assignments

Table 1: Expansion Bus Connector Pinout

Pin	Name	Pin	Name	Pin	Name
a1	NC[1]	b1	D_{16}	c1	NC
a2	NC	b2	D_{17}	c2	NC
a3	D_0	b3	D_{18}	c3	NC
a4	D_2	b4	D_{19}	c4	NC
a5	D_4	b5	D_{20}	c5	NC
a6	D_6	b6	D_{21}	c6	NC
a7	D_8	b7	D_{22}	c7	NC
a8	D_{10}	b8	D_{23}	c8	NC
a9	D_{12}	b9	D_{24}	c9	NC
a10	D_{14}	b10	D_{25}	c10	NC
a11	GND	b11	D_{26}	c11	NC
a12	A_1	b12	D_{27}	c12	NC
a13	A_3	b13	D_{28}	c13	NC
a14	A_5	b14	D_{29}	c14	NC
a15	A_7	b15	D_{30}	c15	NC
a16	A_9	b16	D_{31}	c16	NC
a17	A_{11}	b17	A_0	c17	NC
a18	A_{13}	b18	A_0	c18	NC
a19	A_{15}	b19	NC	c19	NC
a20	A_{17}	b20	NC	c20	NC
a21	A_{19}	b21	NC	c21	NC
a22	NC	b22	NC	c22	NC
a23	NC	b23	NC	c23	NC
a24	NC	b24	NC	c24	NC
a25	NC	b25	NC	c25	NC
a26	NC	b26	NC	c26	NC
a27	NC	b27	NC	c27	NC
a28	NC	b28	NC	c28	NC
a29	NC	b29	NC	c29	NC
a30	NC	b30	NC	c30	NC
a31	NC	b31	NC	c31	NC
a32	NC	b32	NC	c32	NC

[1] This means it's not connected. You knew that though. This footnote is only here so I remember how to do it later.

5 Interrupt System

6 Special Thanks

Ben Eater for creating engaging videos on computer engineering, inspiring me to build a computer of my own.
Lawrence Manning MAXI030
Stephen Moody Y Ddraig(030)
Dave Haynie Author of “The Zorro III Bus Specification”