The M88 Computer Manual

Matthew Iselin

NOTE: This document is a work in progress. It is not yet complete, and some sections may be missing, incomplete, or outright inaccurate.

Contents

I. The M88 Computer

1 What is the M88 Computer?

The M88 Computer is a computer designed around the Intel 8088 processor. It is loosely based on the IBM PC, but with a few modern touches and simplifications to make it easier to build and understand.

The computer was designed with a single goal in mind: to run Sid Meier's Civilization. I achieved that goal, and now I'm sharing the design with the world.

2 Getting Started

The M88 Computer is optimized for production using JLCPCB's PCB Assembly services and LCSC's component sourcing. For this reason, many of the passive components are surface-mount rather than through-hole. Even so, it is possible to hand-solder the motherboard if you are so inclined.

Because the design is optimized for PCB assembly, values are not annotated on the silkscreen. It is recommended to keep the motherboard's KiCad project close by to verify values and connections. You can obtain the KiCad files from the project GitHub repository at `https://github.com/miselin/M88`.

A fully assembled PCB will include DIP sockets for all ICs, and all other components will be present (including ISA slots and PS/2 ports). You will need to populate the board with the required ICs. A system ROM is required and can be downloaded from the same GitHub repository.

The ATTiny system management controller will also need to be programmed.

3 Required ICs

The M88 Computer requires the following ICs:

- 2x Intel 8284 clock generator
- 3x 74LS573 or equivalent
- 1x 74LS688 or 74F521
- 1x 7406
- 1x 75477
- 1x Intel 8088 processor, or equivalent (e.g. NEC V20)
- 2x 74LS125
- 1x 74LS244
- 1x 74LS32
- 1x 74HC245
- 1x 74LS175
- 1x 74LS04
- 1x Intel 8259 Programmable Interrupt Controller
- 2x 74LS247
- 1x Intel 8254 Programmable Interval Timer
- 3x 72LS138
- 1x 74LS02
- 1x 74HC74
- 1x 74LS00
- 1x 74LS08
- 1x ATTiny13A
- 1x W27C512 EEPROM

- 1-5x AS6C1008-55PCN SRAM
- 1x VIA VT82C42 PS/2 Controller

You may use HCT or ACT series chips instead of LS series chips, but the design has not been tested with them. If the motherboard was not pre-populated with crystals, you will also need one 24 MHz crystal and one 14.31818 MHz crystal. You may experiment with the value of the 24 MHz crystal. The system may not operate correctly if the 14.31818 MHz crystal value is changed, as this drives time-keeping and the ISA bus clock.

3.0.1 Crystal Selection

The default crystals of 24 MHz and 14.31818 MHz provide a CPU clock of 8 MHZ and an ISA bus clock of 14.31818 MHz.

Generally, it is not recommended to change the 14.31818 MHz crystal, as this drives the ISA bus and Programmable Interval Timer. Changing its frequency may require software changes to handle the different clock.

The 24 MHz crystal can be changed to provide a different CPU clock. The CPU operates at 1/3 the crystal frequency.

4 Required Peripherals

To use the M88 Computer, you will need the following:

- A Micro-ATX case is *strongly* recommended for mechanical stability, and to appropriately ground the motherboard
- An ATX power supply
- An 8-bit ISA VGA card. The computer has been extensively tested with a Trident TVGA8900C.
- An 8-bit ISA storage controller. The computer has been tested with an XT-IDE.
- A PS/2 keyboard

5 Features

- Built around the Intel 8088 processor, clocked at 8 MHz
- Up to 640 KiB of conventional RAM, using SRAM instead of traditional DRAM
- Four 8-bit ISA slots
- 2 PS/2 ports
- PC Speaker socket
- On-board 7-segment POST code display
- A Micro-ATX form factor motherboard for easy integration into modern cases
- ATTiny-based system manager to handle reset and ATX power supply control
- ATX fan headers for cooling

6 Differences from the IBM PC

The M88 Computer is not a 100% clone of the IBM PC. It is missing a DMA controller, which limits the ability to use certain peripherals. The lack of a DMA controller is an intentional simplification.

The M88 Computer uses SRAM instead of DRAM for the main memory. This further simplifies the design, but also makes it more expensive.

The M88 Computer uses a VIA VT82C42 PS/2 controller instead of the original AT keyboard interface. The mouse IRQ is also connected to IRQ2, instead of the more conventional IRQ12, as the M88 only has one 8259 interrupt controller.

7 I/O Map

FFFF	0100	Available for Peripherals
00FF	00B0	Reserved
00AF	00A0	8259 Interrupt Controller (Secondary)
009F	0090	Reserved
008F	0080	POST Codes
007F	0050	Reserved
004F	0040	8254 Programmable Interval Controller
003F	0030	Reserved
002F	0020	8259 Interrupt Controller (Primary)
001F	0000	Reserved

8 Memory Map

The M88 Computer has a maximum of 640 KiB of conventional memory. At least one SRAM chip is required in Slot #1 to provide the first 64 KiB of memory. SRAM should be loaded in sequential slots. ”Skipping” a slot will result in the system reporting less available memory than is actually installed.

The recommended SRAM chip is the `AS6C1008-55PCN`, but any pin-compatible 128 KiB SRAM chips can be used.

The memory map for the M88 Computer is as follows:

`FFFF:FFFF` `F000:0000`	System BIOS	
`EFFF:FFFF` `A000:0000`	Available for Option ROMs and Peripheral Memory	

`9FFF:FFFF`
`8000:0000`
	System RAM Slot #5	Conventional Memory

`7FFF:FFFF`
`6000:0000`
	System RAM Slot #4	

`5FFF:FFFF`
`4000:0000`
	System RAM Slot #3	

`3FFF:FFFF`
`2000:0000`
	System RAM Slot #2	

`1FFF:FFFF`
`0000:0000`
	System RAM Slot #1	

9 Troubleshooting

If the system does not boot, the POST code display will show a two-digit code. See below for a list of POST codes and what they mean.

`00`	BIOS code has started to run
`01`	CPU registers are OK
`02`	First 64K of memory is OK
`03`	Interrupt handlers installed
`04`	Programmable interrupt controller configured
`05`	Programmable interval controller configured
«beep»	*see above*
`06`	CPU interrupts enabled
`07`	Video BIOS complete
`08`	Memory count complete
`09`	Stack relocated to top of conventional memory
`10`	Extended BIOS data area installed
`11`	PS/2 controller configured
`12`	Serial and Parallel ports configured (if present)
`88`	All Option ROMs successfully called, about to call INT19

10 Power Specifications

Current draws are controlled by PTC resettable fuses. If the system is behaving erratically, check the temperature of each PTC fuse. If a fuse is hot, it has tripped and is limiting current. The system should be powered off until the fuse has cooled and reset, and the cause for the overcurrent should be investigated.

Voltage Rail	Max Current	Purpose
+5V	2A	Main system power for all components.
+5V PS/2	1A	PS/2 power, derived from +5V with its own fuse.
+5V SB	0.5A	Always-on standby power (for ATX power signaling).
+12V	1A	ISA bus power rail.
-12V	1A	ISA bus power rail.

11 INT 88H: M88 BIOS Services

The M88 BIOS provides a number of services to the system. These services are accessed through the `INT 88H` software interrupt.

The value of the AX register, and any input registers, will not be preserved across M88 BIOS calls. All other registers will be preserved.

11.1 Function 24H: Write I^{2}C Byte

To call:

AH	= 24H
AL	= I^{2}C device address
BL	= I^{2}C data byte

Note: This service will block until the I^{2}C bus is idle.

Returns:

Nothing

11.2 Function 25H: Read I^{2}C Byte

To call:

AH	= 54H
AL	= I^{2}C device address

Note: This service will block until the I^{2}C bus is idle.

Returns:

AL	= I^{2}C data byte

11.3 Function 26H: Check I^2C Bus State

To call:

AH = 26H

Returns:

AL = Zero if the I^2C bus is idle, non-zero if busy.

11.4 Function 27H: Set I^2C Device Address

To call:

AH = 27H
BL = New I^2C device address for the controller

Returns:

Nothing

II. Peripherals

12 ISA-to-I^{2}C Bridge

12.1 Introduction

The ISA-to-I^{2}C bridge is a simple device that allows the M88 Computer to communicate with I^{2}C devices.

12.2 Configuration

The ISA-to-I^{2}C bridge card includes a DIP switch to configure the I/O port address. Ports in the range `0x00` to `0x3FF` can be configured. Your configuration needs may vary depending on other peripherals installed on the system, but generally port `0x3A0` should work.

Note that the DIP switch controls address bits 2 through 9, requiring the address to be aligned to a 4-byte boundary.

The DIP switch positions for port `0x3A0` are as follows:

DIP Switch	A9	A8	A7	A6	A5	A4	A3	A2
Position	1	1	1	0	1	0	0	0

12.3 Easy Mode

The M88 System BIOS offers a simple interface for easy I^{2}C communication. See Section 11 for details. This interface is recommended for most users rather than directly managing the PCF8584 IC on the card.

12.4 Low-Level Operation

The ISA-to-I^{2}C bridge card is based around the PCF8584 IC, and is designed to be operated as the I^{2}Ccontroller in the system. The card provides several 3-pin headers for easy access to the `SDA`, `SCL`, and `GND` lines for multiple peripherals. A 5V power line is also provided for convenience.

Throughout this section, the I/O port address is assumed to be `0x3A0`.

You should insert a short delay between I/O operations to allow the PCF8584 to complete its operation. A single `IN` or `OUT` instruction is sufficient, as these both consume 8-10 clock cycles on the M88's 8088 CPU.

12.4.1 Initialization

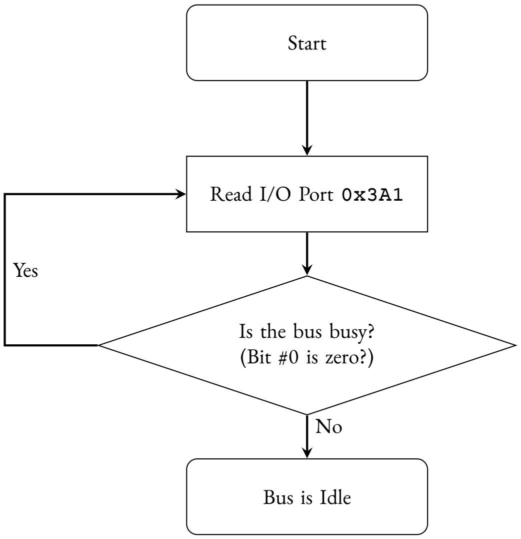
Generally, the M88 System BIOS will perform necessary initialization of the card during startup. However, you may still wish to perform your own initialization. Complete the following sequence of I/O writes to initialize the card:

Port	**Value**	**Description**
`0x3A1`	`0x80`	Select register S1 and disable the serial interface.
`0x3A0`	`0x55`	Set the card's I^{2}C address to `0xAA`.
`0x3A1`	`0xA0`	Select register S2.
`0x3A0`	`0x18`	Configure for a system clock of 8 MHz[1], `SCL` at 90 kHz.
`0x3A1`	`0xC1`	Enable the serial interface, shift the I^{2}C bus to idle.

12.4.2 Checking Bus State

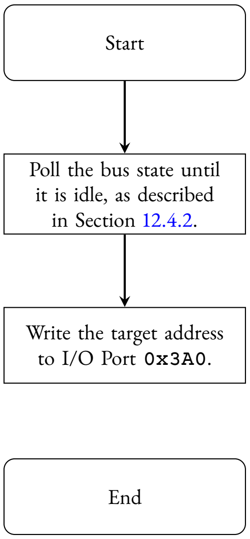
The PCF8584 provides a status register that can be read to determine the current state of the I^{2}C bus. To check that the bus is idle, and therefore ready for communication, read port `0x3A1` and check the lowest bit. If the bit is clear, the bus is busy and you should wait before attempting to communicate.

[1] If you have chosen a different crystal for the CPU clock, consult the `PCF8584` datasheet for the correct value for your resulting clock frequency.



12.4.3 Transmitting Data

To transmit data to an I^{2}C device, complete the following sequence of operations:


Start
Poll the bus state until it is idle, as described in Section 12.4.2.
Write the target address to I/O Port 0x3A0.
End

Port	Direction	Value	Description
			Poll the bus state until it is idle, as described in Section 12.4.2.
`0x3A0`	W	*target address*	Set the target device address.
`0x3A1`	W	`0xC5`	Generate the I^{2}C ”START” condition and clock the target address on the I^{2}C bus.

13 ISA Break-out

13.1 Introduction

blah