

---

Subject: FUZIX on the Mini-M68k (68008) board  
Posted by [jcoffman](#) on Sat, 31 Dec 2022 23:06:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I have posted a version 1 of Fuzix for the Mini-M68k board 2.0. Look in the software:firmware:68000 part of the Wiki if you are interested. The kernel boots from the BIOS accessible FAT16 file system on CF card. Partition 2 is used for the fuzix file system. Contact me through direct e-mail (found on the Board Inventory page) if you have questions.  
--John

---

---

Subject: Re: FUZIX on the Mini-M68k (68008) board  
Posted by [etchedpixels](#) on Tue, 31 Jan 2023 21:09:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

As I was playing with this and didn't have a Mini68K I've added a mini68k emulator to my emulator set <https://github.com/EtchedPixels/EmulatorKit>

It'll run a fair amount of stuff but I'm still finishing the interrupt controller.

---

---

Subject: Re: FUZIX on the Mini-M68k (68008) board  
Posted by [will](#) on Sat, 04 Feb 2023 14:47:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Ha! This is quite fun. I thought I'd post some more detailed instructions here since there was a bit of figuring out involved in making this all work.

I'm running a recent Debian Linux but this all looks pretty portable.

First you need to download Alan's EmulatorKit software, make a tiny patch so the CTS line for the emulated UART appears to be asserted (the Mini68K BIOS expects this, it seems), and build it:

```
$ git clone https://github.com/EtchedPixels/EmulatorKit.git
$ cd EmulatorKit
# if you want to use exactly the same revision that I tested with, do "git checkout d06bfbc" now
$ vim mini68k.c # or other editor of your choice
  Find the device_init function around line 950
  Add "uart16x50_signal_event(uart, 0x10);" as the last line of this function
  Save the file and exit
$ make
```

Now you need to prepare a disk image. John has provided a raw hard disk image, but it needs to be patched with an updated Fuzix kernel which he provided separately. To do this we set up a

"loop" device attached to the disk image, mount the FAT filesystem on the first partition, update the kernel, then unmount and destroy the loop device.

```
# Download and decompress the fuzix.img disk image file:
```

```
$ wget -q
```

```
https://www.retrobrewcomputers.org/lib/exe/fetch.php?media=software:firmwareos:68000:fuzix.01.tar.bz2 -O - | tar jxv -
```

```
# Attach it to a loop device and ask the kernel to parse the partition table
```

```
$ sudo losetup -f -P --show fuzix.img
```

```
# This will print out the device name, eg "/dev/loop0". Modify the following commands if  
# a different device is used on your system.
```

```
# Mount the first partition (which is FAT formatted) on the loop device:
```

```
$ sudo mkdir /mnt/target1
```

```
$ sudo mount /dev/loop0p1 /mnt/target1
```

```
# Download the new fuzix kernel and write it to the FAT filesystem:
```

```
$ wget
```

```
https://www.retrobrewcomputers.org/lib/exe/fetch.php?media=software:firmwareos:68000:kernel02.zip -O /tmp/kernel2.zip
```

```
$ sudo unzip -o -d /mnt/target1/fuzix/ /tmp/kernel2.zip
```

```
# Unmount the disk image, destroy the loop device:
```

```
$ sudo umount /mnt/target1
```

```
$ sudo losetup -d /dev/loop0
```

You will need a ROM image containing the BIOS. The ZIP file contains both 512KB and 128KB ROMs but the emulator only uses the 128KB version, which it expects to find in a file named "mini-128.rom".

```
$ wget
```

```
https://www.retrobrewcomputers.org/lib/exe/fetch.php?media=software:firmwareos:68000:mini_rom_10.3-1.zip -O minirom.zip
```

```
$ unzip minirom.zip mini-128.rom
```

Finally we have the complication that EmulatorKit does not use a raw disk image, instead it requires a special 1KB header to be present, so you need to use the "makedisk" program to prepare a compatible drive image that is large enough and copy the raw disk image into it. Look in "ide.h" for a list of drive types, we need at least 64MB so I suggest using type 5, the 128MB "ACME ACCELLERATTI INCREDIBILUS". I suggest making a backup copy in case your filesystems become corrupted.

```
$ ./makedisk 5 mini68k-hdd.img
$ dd if=fuzix.img bs=512 of=mini68k-hdd.img seek=2 conv=notrunc
$ rm fuzix.img # no longer required
$ cp mini68k-hdd.img mini68k-hdd.img.bak
```

Now you're ready to boot it up! On the first boot the BIOS will ask you to configure the NVRAM (The NVRAM contents is persisted in a file named "mini68k.nvram"). The BIOS defaults are all sensible choices for the emulator. To terminate the emulator send a SIGQUIT signal (typically Ctrl+^).

```
$ ./mini68k -i mini68k-hdd.img
```

Welcome to the MINI-M68000 System BIOS 10.3-1

Copyright (C) 2011-2022 John R. Coffman <johninsd@gmail.com>  
Copyright (C) 2015,2021 William R. Sowerbutts <will@sowerbutts.com>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License in the file COPYING in the distribution directory along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Licensed for hobbyist use on the RetroBrew MINI-M68000 CPU board.

To enter Setup type 's' during the memory test.

Found memory from 000000 to 080000

C: ACME ACCELLERATTI INCREDIBILUS v0.1

LBA sectors: 0x40000 = 262144

Boot at 15:43:41 on 2023-02-04 2459980J 1073701T

BIOS version 10.3-1 of 01-Jan-2023 Mon Jan 2 11:01:34 PST 2023

-C:/>

To boot Fuzix, type "boot" at the BIOS "-C:/>" prompt to run the "boot.cmd" script.

```
-C: /> boot
boot.CMD: 74 bytes, Command file.
REM
REM This is disk FUZIX1
REM
supv /fuzix/fuzix.elf hda2 console=tty1
/fuzix/fuzix.elf: 64312 bytes, ELF.
Loading 44040 bytes from file offset 0x1000 to memory at 0x1000
COMMAND LINE: || hda2 console=tty1||
Entry at 0x1000 in supervisor mode
FUZIX version 0.4pre1
Copyright (c) 1988-2002 by H.F.Bower, D.Braun, S.Nitschke, H.Peraza
Copyright (c) 1997-2001 by Arcady Schekochikhin, Adriano C. R. da Cunha
Copyright (c) 2013-2015 Will Sowerbutts <will@sowerbutts.com>
Copyright (c) 2014-2022 Alan Cox <alan@etchedpixels.co.uk>
Devboot
Motorola 68000 processor detected.
0kB total RAM, 65436kB available to processes (125 processes max)
Enabling interrupts ... ok.
IDE drive 0: hda: hda1 hda2
IDE drive 1: ide error, status=0000
Mounting root fs (root_dev=2, ro): OK
Starting /init
init version 0.9.0ac#1
Checking root file system.
Current date is Sat 2023-02-04
Enter new date:
Current time is 15:46:06
Enter new time:

^ ^
n n Fuzix 0.4pre1
>@<
  Welcome to Fuzix
m m

login:
```

---

Subject: Re: FUZIX on the Mini-M68k (68008) board  
Posted by [will](#) on Sat, 04 Feb 2023 19:42:11 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

With a bit of hacking on Alan's code I've got the start of a KISS-68030 emulator working ... nice!

Unfortunately the version of Musashi I am using doesn't fully support the PMMU. However I think I

can fix this.

```
btg@milo:~/projects/retrobrew/EmulatorKit$ make && ./kiss68030 -f
make: Nothing to be done for 'all'.
```

Welcome to the KISS-68030 System BIOS 10.3-1

Copyright (C) 2011-2022 John R. Coffman <johninsd@gmail.com>  
Copyright (C) 2015,2021 William R. Sowerbutts <will@sowerbutts.com>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License in the file COPYING in the distribution directory along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Licensed for hobbyist use on the RetroBrew KISS-68030 CPU board.

To enter Setup type 's' during the memory test.  
Found memory from 00000000 to 10000000

Start of Setup.  
The MF/PIC UART is 16450  
The contents of NVRAM are Invalid.  
The clock is running.

Serial port baud rate (Kbit/sec) [9600]:

---

Subject: Re: FUZIX on the Mini-M68k (68008) board  
Posted by [will](#) on Sun, 05 Feb 2023 01:23:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

With a newer M68K emulation jammed in there, I have my kiss68030 emulator at <https://github.com/willsowerbutts/EmulatorKit> working well enough to load and run the Linux kernel. It needs more work the other hardware emulation now - IRQ controller, UART, IDE, etc.

The CPU and RTC look to be working well, at least!

Welcome to the KISS-68030 System BIOS 10.3-1

Copyright (C) 2011-2022 John R. Coffman <johninsd@gmail.com>  
Copyright (C) 2015,2021 William R. Sowerbutts <will@sowerbutts.com>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License in the file COPYING in the distribution directory along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Licensed for hobbyist use on the RetroBrew KISS-68030 CPU board.

To enter Setup type 's' during the memory test.

Found memory from 00000000 to 02000000

C: ACME BIGGUS DISKUS v0.1

LBA sectors: 0x1000000 = 16777216

Boot at 1:19:35 on 2023-02-05 2459981J 90548T

BIOS version 10.3-1 of 01-Jan-2023 Mon Jan 2 11:01:34 PST 2023

-C:/> boot

boot.CMD: 74 bytes, Command file.

0:/vmlinux console=ttyS0,115200n8r root=/dev/hda2 mfpic=0x50,7,0,16000000

0:/vmlinux: 4997308 bytes, ELF.

Loading 3447776 bytes from file offset 0x1000 to memory at 0x1000

Loading 91136 bytes from file offset 0x34B000 to memory at 0x373000

Linux kernel detected: creating bootinfo at 0x38A000

Entry at 0x2000 in supervisor mode

ABcKGHIJK

[ 0.000000] Linux version 4.5.0 (btg@victor) (gcc version 5.3.1 20160205 (Debian 5.3.1-8) ) #32

Mon Mar 28 17:45:29 BST 2016

[ 0.000000] bootconsole [kissuart0] enabled

[ 0.000000] Built 1 zonelists in Zone order, mobility grouping on. Total pages: 8112

[ 0.000000] Kernel command line: console=ttyS0,115200n8r root=/dev/hda2

mfpic=0x50,7,0,16000000

[ 0.000000] PID hash table entries: 128 (order: -3, 512 bytes)

```

[ 0.000000] Dentry cache hash table entries: 4096 (order: 2, 16384 bytes)
[ 0.000000] Inode-cache hash table entries: 2048 (order: 1, 8192 bytes)
[ 0.000000] Sorting __ex_table...
[ 0.000000] Memory: 28748K/32768K available (2380K kernel code, 336K rwdata, 800K rodata,
92K init, 160K bss, 4020K reserved, 0K cma-reserved)
[ 0.000000] Virtual kernel memory layout:
[ 0.000000]   vector : 0x0034acbc - 0x0034b0bc ( 1 KiB)
[ 0.000000]   kmap   : 0xd0000000 - 0xf0000000 ( 512 MiB)
[ 0.000000]   vmalloc : 0x02800000 - 0xd0000000 (3288 MiB)
[ 0.000000]   lowmem  : 0x00000000 - 0x02000000 ( 32 MiB)
[ 0.000000]   .init  : 0x00373000 - 0x0038a000 ( 92 KiB)
[ 0.000000]   .text  : 0x00001000 - 0x002540fc (2381 KiB)
[ 0.000000]   .data  : 0x00256b40 - 0x00372f2c (1137 KiB)
[ 0.000000]   .bss   : 0x0034abe0 - 0x00372f2c ( 161 KiB)
[ 0.000000] NR_IRQS:24
[ 0.000000] ns32202 at 0xffff0040
[ 0.000000] unexpected interrupt from 268
[ 0.000000] ns32202: whoops! ISRV=0008 IMSK=fbff IPND=0000
[ 0.020000] Console: colour dummy device 80x25
[ 0.020000] Calibrating delay loop... 0.04 BogoMIPS (lpj=246)
[ 0.260000] pid_max: default: 32768 minimum: 301
[ 0.260000] Mount-cache hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.260000] Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.340000] devtmpfs: initialized
[ 0.360000] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
19112604462750000 ns
[ 0.380000] NET: Registered protocol family 16
[ 0.400000] MF/PIC at I/O address 0x40, IDE ECB IRQ 9, UART ECB IRQ 12, UART clock
1843200 Hz
[ 0.400000] MF/PIC at I/O address 0x50, IDE ECB IRQ 7, UART ECB IRQ 0, UART clock
16000000 Hz
[ 0.520000] SCSI subsystem initialized
[ 0.560000] VFS: Disk quotas dquot_6.6.0
[ 0.560000] VFS: Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
[ 0.740000] NET: Registered protocol family 2
[ 0.780000] TCP established hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.780000] TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.780000] TCP: Hash tables configured (established 1024 bind 1024)
[ 0.780000] UDP hash table entries: 256 (order: 0, 4096 bytes)
[ 0.780000] UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
[ 0.800000] NET: Registered protocol family 1
[ 0.820000] futex hash table entries: 256 (order: -1, 3072 bytes)
[ 0.820000] zbud: loaded
[ 0.900000] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 253)
[ 0.900000] io scheduler noop registered
[ 0.900000] io scheduler cfq registered (default)
[ 0.920000] Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
[ 0.940000] console [ttyS0] disabled

```

```

[ 0.940000] console [ttyS0] disabled
[ 0.960000] Uniform Multi-Platform E-IDE driver
[ 0.960000] ide0: KISS-68030 MF/PIC IDE controller
[ 1.560000] ide0 at 0xffff0044 on irq 17
[ 1.560000] ide1: KISS-68030 MF/PIC IDE controller
[ 2.160000] ide1 at 0xffff0054 on irq 15
[ 2.160000] ide-gd driver 1.18
[ 2.160000] mousedev: PS/2 mouse device common for all mice
[ 2.180000] rtc-ds1302 rtc-ds1302.0: rtc core: registered ds1302 as rtc0
[ 2.180000] ledtrig-cpu: registered to indicate activity on CPUs
[ 2.200000] NET: Registered protocol family 17
[ 2.200000] mpls_gso: MPLS GSO support
[ 2.200000] registered taskstats version 1
[ 2.200000] zswap: loaded using pool lzo/zbud
[ 2.220000] rtc-ds1302 rtc-ds1302.0: setting system clock to 2023-02-05 01:19:39 UTC
(1675559979)
[ 2.220000] Warning: unable to open an initial console.
[ 2.220000] VFS: Cannot open root device "hda2" or unknown-block(0,0): error -6
[ 2.220000] Please append a correct "root=" boot option; here are the available partitions:
[ 2.220000] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0)
[ 2.220000] CPU: 0 PID: 1 Comm: swapper Not tainted 4.5.0 #32
[ 2.220000] Stack from 01c2dee0:
    01c2dee0 002e32f7 0004d7d6 00008001 0009aaa4 00154f04 0000206c 01c2df50
    0004d998 003d5b70 ffffffff 003745d6 002c3779 01c2df50 002c3728 002c36f5
    0038602d 01c2df50 ffffffff 00000000 01c2df50 00000000 0000000c 0001cfe0
    00154e4c 0038875c 00385de4 00154d4a 756e6b6e 6f776e2d 626c6f63 6b28302c
    30290080 00000000 003746ac 002c37dd 003746d6 002c37dd 00008001 0038602d
    00000050 00374828 00000008 00097a14 0038875c 00385de4 003742fe 00000000
[ 2.220000] Call Trace: [<0004d7d6>] panic+0xa0/0x1fe
[ 2.220000] [<00008001>] fp_ueq+0x7/0x12
[ 2.220000] [<0009aaa4>] SyS_mount+0x0/0x8a
[ 2.220000] [<00154f04>] strlen+0x0/0x18
[ 2.220000] [<0000206c>] do_one_initcall+0x0/0x194
[ 2.220000] [<0004d998>] printk+0x0/0x1a
[ 2.220000] [<003745d6>] mount_block_root+0x1ac/0x240
[ 2.220000] [<0001cfe0>] parse_args+0x0/0x31c
[ 2.220000] [<00154e4c>] strncmp+0x0/0x34
[ 2.220000] [<00154d4a>] strcpy+0x0/0x1c
[ 2.220000] [<003746ac>] mount_root+0x42/0x74
[ 2.220000] [<003746d6>] mount_root+0x6c/0x74
[ 2.220000] [<00008001>] fp_ueq+0x7/0x12
[ 2.220000] [<00374828>] prepare_namespace+0x14a/0x186
[ 2.220000] [<00097a14>] SyS_dup+0x0/0x46
[ 2.220000] [<003742fe>] kernel_init_freeable+0x188/0x192
[ 2.220000] [<0024fa42>] kernel_init+0x0/0xcc
[ 2.220000] [<0024fa4a>] kernel_init+0x8/0xcc
[ 2.220000] [<0024fa42>] kernel_init+0x0/0xcc
[ 2.220000] [<00002810>] ret_from_kernel_thread+0xc/0x14

```



```
[ 2.220000]
[ 2.220000] ---[ end Kernel panic - not syncing: VFS: Unable to mount root fs on
unknown-block(0,0)
[ 82.560000] random: nonblocking pool is initialized
```

---

---

Subject: Re: FUZIX on the Mini-M68k (68008) board  
Posted by [will](#) on Sun, 05 Feb 2023 23:11:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Emulator is now working well enough to boot linux -- it required a few tweaks to get the UART and IDE working:

UART - it wanted control signals looped back when loopback mode was enabled.

IDE - it wanted the device/head register to have a couple of "obsolete" bits set to 1.

With those fixed it boots far enough to mount the root filesystem and then it seems to stop. My guess is that it is trying to run init in userspace and init wants to write to the console, but the UART isn't signalling that it's ready by raising an interrupt at the right time.

edit 2022-02-06: I think it's actually drowning in some sort of interrupt hell. Further debugging of the ns202 model is required.

I used to think debugging an interrupt service routine was hard, but now I know that's only because I hadn't yet tried debugging a software emulation of an interrupt controller...!

```
-C:/> boot
boot.CMD: 80 bytes, Command file.
0:/vmlinux console=ttyS0,115200n8r root=/dev/hda2 mfpic=0x50,7,0,16000000 debug
0:/vmlinux: 4997308 bytes, ELF.
Loading 3447776 bytes from file offset 0x1000 to memory at 0x1000
Loading 91136 bytes from file offset 0x34B000 to memory at 0x373000
Linux kernel detected: creating bootinfo at 0x38A000
Entry at 0x2000 in supervisor mode
```

ABCKGHIJK

```
[ 0.000000] Linux version 4.5.0 (btg@victor) (gcc version 5.3.1 20160205 (Debian 5.3.1-8) ) #32
Mon Mar 28 17:45:29 BST 2016
[ 0.000000] bootconsole [kissuart0] enabled
[ 0.000000] On node 0 totalpages: 8192
[ 0.000000] free_area_init_node: node 0, pgdat 0034b1c4, node_mem_map 0038d000
[ 0.000000] DMA zone: 80 pages used for memmap
[ 0.000000] DMA zone: 0 pages reserved
[ 0.000000] DMA zone: 8192 pages, LIFO batch:0
[ 0.000000] pcpu-alloc: s0 r0 d32768 u32768 alloc=1*32768
```

```

[ 0.000000] pcpu-alloc: [0] 0
[ 0.000000] Built 1 zonelists in Zone order, mobility grouping on. Total pages: 8112
[ 0.000000] Kernel command line: console=ttyS0,115200n8r root=/dev/hda2
mfpic=0x50,7,0,16000000 debug
[ 0.000000] PID hash table entries: 128 (order: -3, 512 bytes)
[ 0.000000] Dentry cache hash table entries: 4096 (order: 2, 16384 bytes)
[ 0.000000] Inode-cache hash table entries: 2048 (order: 1, 8192 bytes)
[ 0.000000] Sorting __ex_table...
[ 0.000000] Memory: 28748K/32768K available (2380K kernel code, 336K rwdata, 800K rodata,
92K init, 160K bss, 4020K reserved, 0K cma-reserved)
[ 0.000000] Virtual kernel memory layout:
[ 0.000000]   vector   : 0x0034acbc - 0x0034b0bc ( 1 KiB)
[ 0.000000]   kmap     : 0xd0000000 - 0xf0000000 ( 512 MiB)
[ 0.000000]   vmalloc  : 0x02800000 - 0xd0000000 (3288 MiB)
[ 0.000000]   lowmem   : 0x00000000 - 0x02000000 ( 32 MiB)
[ 0.000000]   .init   : 0x00373000 - 0x0038a000 ( 92 KiB)
[ 0.000000]   .text   : 0x00001000 - 0x002540fc (2381 KiB)
[ 0.000000]   .data   : 0x00256b40 - 0x00372f2c (1137 KiB)
[ 0.000000]   .bss   : 0x0034abe0 - 0x00372f2c ( 161 KiB)
[ 0.000000] NR_IRQS:24
[ 0.000000] ns32202 at 0xffff0040
[ 0.000000] unexpected interrupt from 268
[ 0.000000] ns32202: whoops! ISRV=0008 IMSK=fbff IPND=0200
[ 0.000000] Console: colour dummy device 80x25
[ 0.000000] Calibrating delay loop... 0.04 BogoMIPS (lpj=246)
[ 0.240000] pid_max: default: 32768 minimum: 301
[ 0.240000] Mount-cache hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.240000] Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.320000] devtmpfs: initialized
[ 0.340000] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
19112604462750000 ns
[ 0.360000] NET: Registered protocol family 16
[ 0.380000] MF/PIC at I/O address 0x40, IDE ECB IRQ 9, UART ECB IRQ 12, UART clock
1843200 Hz
[ 0.380000] MF/PIC at I/O address 0x50, IDE ECB IRQ 7, UART ECB IRQ 0, UART clock
16000000 Hz
[ 0.500000] SCSI subsystem initialized
[ 0.540000] VFS: Disk quotas dquot_6.6.0
[ 0.540000] VFS: Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
[ 0.740000] NET: Registered protocol family 2
[ 0.760000] TCP established hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.760000] TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.760000] TCP: Hash tables configured (established 1024 bind 1024)
[ 0.760000] UDP hash table entries: 256 (order: 0, 4096 bytes)
[ 0.760000] UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
[ 0.780000] NET: Registered protocol family 1
[ 0.800000] futex hash table entries: 256 (order: -1, 3072 bytes)
[ 0.800000] zbud: loaded

```

```
[ 0.900000] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 253)
[ 0.900000] io scheduler noop registered
[ 0.900000] io scheduler cfq registered (default)
[ 0.900000] Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
[ 0.920000] console [ttyS0] disabled
[ 0.920000] serial8250.0: ttyS0 at MMIO 0xffff0048 (irq = 20, base_baud = 115200) is a 16450
[ 0.920000] console [ttyS0] enabled
[ 0.920000] console [ttyS0] enabled
[ 0.920000] bootconsole [kissuart0] disabled
[ 0.920000] bootconsole [kissuart0] disabled
[ 0.940000] Uniform Multi-Platform E-IDE driver
[ 0.940000] ide0: KISS-68030 MF/PIC IDE controller
[ 0.940000] Probing IDE interface ide0...
[ 1.240000] hda: Raw disk image, ATA DISK drive
ide: hd0: 1: not present
ide: hd0: 1: not present
ide: hd0: 1: not present
ide: hd0: 1: not present
[ 1.960000] ide0 at 0xffff0044 on irq 17
[ 1.960000] ide1: KISS-68030 MF/PIC IDE controller
[ 1.960000] Probing IDE interface ide1...
[ 2.560000] ide1 at 0xffff0054 on irq 15
[ 2.560000] ide-gd driver 1.18
[ 2.560000] hda: max request size: 128KiB
[ 2.560000] hda: 16777216 sectors (8589 MB), CHS=8192/16/128
[Bug: request for sector offset 0].
[ 2.600000] hda: hda1 hda2 hda3
[ 2.620000] mousedev: PS/2 mouse device common for all mice
[ 2.640000] rtc-ds1302 rtc-ds1302.0: rtc core: registered ds1302 as rtc0
[ 2.640000] ledtrig-cpu: registered to indicate activity on CPUs
[ 2.660000] NET: Registered protocol family 17
[ 2.660000] mpls_gso: MPLS GSO support
[ 2.680000] registered taskstats version 1
[ 2.680000] zswap: loaded using pool lzo/zbud
[ 2.680000] rtc-ds1302 rtc-ds1302.0: setting system clock to 2023-02-05 23:07:53 UTC
(1675638473)
[ 2.700000] EXT4-fs (hda2): couldn't mount as ext3 due to feature incompatibilities
[ 2.700000] EXT4-fs (hda2): couldn't mount as ext2 due to feature incompatibilities
[ 2.790000] EXT4-fs (hda2): mounted filesystem with writeback data mode. Opts: (null)
[ 2.790000] VFS: Mounted root (ext4 filesystem) readonly on device 3:2.
[ 2.810000] Freeing unused kernel memory: 92K (00373000 - 0038a000)
```

---

Subject: Re: FUZIX on the Mini-M68k (68008) board

Posted by [e2k](#) on Mon, 06 Feb 2023 08:30:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Thanks a lot for this long write-up!

---

---

Subject: Re: FUZIX on the Mini-M68k (68008) board

Posted by [will](#) on Sun, 12 Feb 2023 22:41:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

If anyone is interested I now have my kiss68030 emulator at <https://github.com/willsowerbutts/EmulatorKit> working well enough to load and run the Linux kernel. It boots up all the way until it mounts the root filesystem and tries to load /sbin/init, at which point the wheels come off.

After (quite!) some investigation it turns out the problem is that the emulated 68030 PMMU cannot raise a bus error signal when a page lookup fails. It knows it needs to be done, but that part of the PMMU just isn't hooked up to the CPU. I think this is because the emulated CPU design has no way to abort the instruction and restart it later (after processing the exception). Instead it just ploughs ahead even though the MMU translation is wrong, happily scribbling over the wrong parts of memory!

I had a look at bringing in a newer version of the 68K emulator from MAME, but it looked like more trouble than it is worth.

The emulator is still useful. You can run the Linux kernel, just not the userspace. I've been using it to develop a port of Linux 6.1 to the KISS-68030. I have it booting, with UART, IDE and interrupts working. I'm figuring out how to hook up the RTC and then it should be functionally comparable to the older Linux 4.5 port.

---

---

Subject: Re: FUZIX on the Mini-M68k (68008) board

Posted by [tingo](#) on Tue, 14 Feb 2023 17:15:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Well done!

---

---

Subject: Re: FUZIX on the Mini-M68k (68008) board

Posted by [etchedpixels](#) on Wed, 15 Feb 2023 15:18:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Nice.. I'd expect the ns202 model to be wrong in places. I think the priority ordering may be backwards for one as well as not doing rotating priority.

For the Fuzix side of the min68k it should be sufficient when building from the main git to

```
set the platform in Makefile
make clean
make diskimage
```

to get a disk.ing (real disk) and emu-ide.img (emulator header + disk) that boot

The BIOS I was using just worked. Guess there are subtle differences somewhere in the one I tested. When it's ready please do send me patches. The original Fuzix image did have some issues with cts but those went away when I fixed up the serial code to only do CTS/RTS when told to (and the console defaults to no flow control, no carrier detect)

---

Subject: Re: FUZIX on the Mini-M68k (68008) board  
Posted by [etchedpixels](#) on Thu, 16 Feb 2023 11:48:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

For the MMU emulation it might be worth looking at something like Hatari instead, as that has full PMMU emulation, 68040 emulation etc.

---

Subject: Re: FUZIX on the Mini-M68k (68008) board  
Posted by [denis2342](#) on Wed, 26 Apr 2023 16:30:36 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Hi John,

some years ago you said you had the detection of the 16750 fixed but not included. I see that you did forget it again. Maybe you still have your workflow set up and can add that to 10.3.

Also I think putting all this stuff in a version control system would encourage people to participate (I'm guilty here as well, already forgot what I hacked together to make it all work...)

after some digging I found my old changes:

```
+ move.b #0x83,lcr(%a5) /* 8n1 and DLAB on */
  move.b #0xE7,fcr(%a5) /* enable all FIFO's */
+ move.b #0x03,lcr(%a5) /* 8n1 and DLAB off */
  move.b iir(%a5),%d0 /* IIR and FCR are the same */
  btst #6,%d0 /* test bit 6 */
  jbeq test16450
@@ -114,15 +118,15 @@
  btst #7,%d0
  jbeq fnd9 /* it is a 16550 */
/* we have at least a 16550A */
- move.l #6,%d1 /* set 16750 response */
- btst #5,%d0 /* test bit 5 */
```

```
- jrne fnd9 /* it is a 16750 */
  move.b #0x20,mcr(%a5) /* set AFC */
  move.l #4,%d1 /* say 16550A */
  btst #5,mcr(%a5) /* stuck on zero? */
  jbeq fnd9 /* it is a 16550A */
- move.l #5,%d1
- jbra fnd9 /* it is a 16550C */
+ move.l #5,%d1 /* set 16550C response */
+ btst #5,%d0 /* test for 64 byte FIFO */
+ jbeq fnd9 /* it is a 16550C */
+ move.l #6,%d1 /* set 16C750 response */
+ jbra fnd9 /* it is a 16C750 */
test16450:
  move.l #1,%d1 /* say 8250 */
  move.b #0x2A,scr(%a5)
```

Denis

---