

Asus K8S-MX / SiS 965L workarounds

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.hardware/2005-05/0707.html>

From: Juha Laiho (*Juha.Laiho_at_iki.fi*)

Date: 05/28/05

Date: Sat, 28 May 2005 10:31:48 +0000 (UTC)

I recently acquired an Asus K8S-MX -based system (which is based on the SiS 965L chipset). Looks like I completely managed to omit my homework; the system does have a set of serious issues with Linux - some of which I've managed to work around, thanks to various sources on the Net.

To help all those who have a board with this same chipset, I've collected here my findings and (non-production-quality) workarounds. Those who haven't done a purchase decision yet, I recommend to avoid buying a board with SiS 96x chipsets for Linux use (unless you're prepared to leave the on-board ATA (parallel) interfaces and on-board LAN unused). The SATA adapter on this chipset appears to be supported (I don't have a SATA disk to test this assumption, though).

The distribution I've been installing on the machine is SuSE 9.2 (from a downloaded ISO).

1. Failure to properly recognize and initialize the IDE chipset.

This made the system prone to hanging when doing multi-sector I/O to the HD (default behaviour on SuSE, at least). For me, this happened 100% of the time when attempting to mount a reiserfs filesystem. The last message printed by kernel was something about replaying transactions (but unfortunately I don't seem to find the exact message right now).

Additionally, the HDD transfer rates were extremely degraded, as the recognition problem prevented use of DMA transfers (4MB/s, as opposed to >50MB/s).

For this, the initial workaround (first, to be able to install the system at all, and later, to boot the installed system), was to disable multi-sector transfers. During installation this could be done (prior to partitioning the disk and creating/mounting the file systems) with command "hdparm -m 0 /dev/hda" given from the command prompt on tty2. Finding the correct place for this to get the installed system to boot was a bit trickier. I booted from cd to rescue mode; then gave the command manually once more. After that I mounted the installed system, unpacked the initrd image to the temporary directory, loop-mounted the unpacked

```
image, and edited the linuxrc contained in the image (phew!).
So, the above, in commands – after getting to the rescue prompt:
hdparm –m 0 /dev/hda
mount /dev/hdaX /mnt # Replace X with your actual root fs partition
gzip –dc /mnt/boot/initrd–2.6.8–24 > /tmp/initrd
mkdir /mnt2
mount –o loop /tmp/initrd /mnt2
vi /mnt2/linuxrc # Add a call to "/sbin/hdparm –m 0 /dev/hda"
cp –p /mnt/sbin/hdparm /sbin/hdparm
umount /mnt2
gzip –c /tmp/initrd > /mnt/boot/initrd–2.6.8–24
```

After this I was able to boot the installed system. Then, I could address the speed issue. The IDE controller logic in the system is a derivative of SiS 5513, with the actual circuitry embedded into the SiS 965L chip. However, the identifying information on SiS 965L is such that the stock sis5513 driver in Linux does not recognise it — the sis5513 driver fails to load. The following lines in "dmesg" output appear:

```
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
SIS5513: IDE controller at PCI slot 0000:00:02.5
SIS5513: chipset revision 1
SIS5513: not 100% native mode: will probe irqs later
... but nothing more. This indicates (though badly), that the driver
started to load, but failed. I made a small change [below] to somewhat
brute-force the recognition of the chipset, and it appears to work
(even though the solution most possibly doesn't use all of the 965L
capabilities). After the change, "dmesg" output will also contain
(right after the above dmesg output):
SIS5513: SiS 962/963 MuTIOL IDE UDMA133 controller
  ide0: BM-DMA at 0xffa0–0xffa7, BIOS settings: hda:DMA, hdb:DMA
  ide1: BM-DMA at 0xffa8–0xffaf, BIOS settings: hdc:DMA, hdd:DMA
```

After this change, also the multi-sector mode works, so the previous hack to disable multi-sector mode in the initrd can be removed (or, actually, you don't need to make sure you redo the change after each kernel install/initrd rebuild cycle).

Change needed was in file drivers/ide/pci/sis5513.c within the Linux kernel source; here is the change in unidiff format:

```
--- sis5513.c.orig 2005-03-29 15:26:09.000000000 +0300
+++ sis5513.c 2005-05-28 12:50:53.031716736 +0300
@@ -785,7 +785,7 @@
     pci_read_config_word(dev, PCI_DEVICE_ID, &trueid);
     pci_write_config_dword(dev, 0x54, idemisc);

- if (trueid == 0x5518) {
+ if ((trueid == 0x5518) || (trueid == 0x180)) {
     printk(KERN_INFO "SIS5513: SiS 962/963 MuTIOL IDE UDMA133 controller\n");
```

```
chipset_family = ATA_133;
```

... so, just change the (trueid = 0x5518) to ((trueid = 0x5518) || (trueid = 0x180)) . The 0x180 I obtained by printing out the trueid value that was obtained from the chip. I have no knowledge on how similar or different the 962/963 and 965L implementations are -- I just decided to make the guess that they might be similar enough to work with the same code, and blindly forced the use of the same code.

2. Driver for onboard LAN adapter completely missing.

The other big issue was that the onboard 10/100 LAN did not work. Like the IDE functionality, this is again circuitry embedded into the SiS 965L chip. The circuitry is apparently very close to what was SiS 190/191 network adapter chip. Funnily, there is next to no information about this chip anywhere. There has been a Linux driver for this chip, but it has been removed from the kernel at version 2.6.5 with comment:

<jgarzik@redhat.com>

Delete sis190 net driver.

The driver was copied from the very-buggy r8169 (pre-Francois), and is for hardware that isn't even out of the lab yet.

The good luck is, the driver compiles and "works" (such as it does) just by copying the source (sis190.c) from 2.6.5 sources and setting it up into the current source tree (for SuSE 9.2, current being 2.6.8-24.14). Setting up means:

- add a section with SIS190 into drivers/net/Kconfig (I just copied from SIS900 section)
- add a line with CONFIG_SIS190 into drivers/net/Makefile (again, copy and change from the CONFIG_SIS900 line)
- place a copy of the sis190.c (from 2.6.5 sources) into drivers/net
- configure and compile

Additionally, I found one small fix to the driver on the Net; (see <http://kerneltrap.org/node/4838>, a reply by Steve Tsai). Again, unidiff for the change:

```
--- sis190.c.orig 2004-04-04 06:38:27.000000000 +0300
+++ sis190.c 2005-05-28 13:13:41.131733920 +0300
@@ -1084,6 +1084,8 @@
         if ((status & (TxQ0Int | RxQInt)) == 0)
             break;

+ handled = 1;
+
+ // Rx interrupt
+ if (status & (RxQInt)) {
```

SiS190_rx_interrupt(dev, tp, ioadd

... so, add the 'handled = 1;' statement into the appropriate place.

After this, I could get the network going – but still with issues (and apparently, the removal of the driver was done for a reason; so this same driver shouldn't be placed back into the kernel).

Issues left:

- adapter connects in half-duplex mode; driver does not support configuration by ethtool (perhaps earlier mii-tool or mii-diag might work)
- disconnect/reconnect of the lan cable may freeze the machine
- driver reports (and uses) a very suspect MAC address, namely 00:11:11:11:11:00

These look like:

- the driver isn't properly communicating with hardware (at least reads some configuration data at a wrong place; thus there is also a risk that this driver may mess up the configuration data, if anything is written back improperly)
- the driver isn't properly communicating with Linux kernel, and is not using the current frameworks for services (like ethtool)

But so, this can be used in "emergencies" -- but be careful if you have two of these in the same LAN; I expect it'll cause a MAC address conflict. Years back, a change in Intel eeepro100 adapters (if I recall correctly) did cause a similar situation: Linux read MAC information from a wrong place, and when used in Linux, all adapters of the new board version showed identical MAC addresses.

3. I've yet to find any management bus on the board (so, anything readable by the lm_sensors or something comparable).

--

Wolf a.k.a. Juha Laiho Espoo, Finland
(GC 3.0) GIT d- s+: a C++ ULSH++++\$ P++@ L+++ E- W+\$@ N++ !K w !O !M V
PS(+) PE Y+ PGP(+) t- 5 !X R !tv b+ !DI D G e+ h---- r+++ y++++
"...cancel my subscription to the resurrection!" (Jim Morrison)