



# Frequently Asked Questions about Sun NVRAM/hostid

\$Date: 2004/02/02 07:28:26 \$

\$Revision: 1.69 \$

This document has been placed in the public domain by the author - [Mark Henderson <ratatosk@mail2me.com>](mailto:ratatosk@mail2me.com)

**Please note that newer M48T02 chips will not work properly in sun4c machines. Unfortunately, the SGS-Thomson has changed the specifications of the M48T02. More information is available at <http://www.sunhelp.org/pipermail/rescue/2001-May/024561.html>. Steve Powers has also provided a partial workaround [http://www.squirrelrel.com/powers\\_email.txt](http://www.squirrelrel.com/powers_email.txt)**

**If this information is abused, misinterpreted, or incorrect you can render your computer unbootable. The information in this document 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. You use this document at your own risk. The author disclaims responsibility for any damages that might result from the use of this document, even if they result from negligence or errors on the part of the author. Please do not use the information in this document to steal software or violate licence agreements with software vendors.**

**This is the new and hopefully improved Sun NVRAM FAQ. It is an attempt to answer the following frequently asked questions on comp.sys.sun.hardware and comp.sys.sun.admin.**

## **1. When I turn on my Sun I get output which looks something like:**

```
Sun Workstation, Model Sun-XXXXXX Series.  
ROM Rev X.X, XXMB memory installed  
ID PROM invalid.  
Testing 0 Megabytes of Memory ... Completed.  
ERROR: missing or invalid ID prom  
Requesting Internet address for 0:0:0:0:0:0
```

**or**

```
Sun Workstation, Model Sun-XXXX Series.  
Type 4 Keyboard  
ROM Rev X.X, XXMB memory installed, Serial #16777215  
Ethernet address ff:ff:ff:ff:ff:ff, Host ID ffffffff
```

```
Invalid format type in NVRAM  
The IDPROM contents are invalid
```

**How can I fix this? (*Probably Replace chip*)**

## **2. The clock on my Sun has stopped working. When I turn off the machine it loses track of the time and/or complains about invalid TOD when I turn it on. How do I get my clock to keep time when the system is off? (*Replace chip*)**

3. I get warnings about by NVRAM battery being low when I boot my machine but otherwise it works nicely. Should I be worried? (*Replace chip*)
4. When my Sun boots I get warnings about the IDPROM checksum being incorrect and/or Invalid format type in NVRAM. What does this mean and how can I fix it?
5. The hostid and/or ethernet address on my Sun have been corrupted. How can I restore it to a sane state?
6. How do I change the hostid and/or ethernet address of my Sun workstation?
7. My Sun is in full-security mode (can't even boot without password) and I don't know the EEPROM password. How do I fix this? (*Replace chip*)

The information in this document applies to the following Sun architectures: sun4c, sun4m, sun4d, sun4u, sun3x (but not to sun4, sun3)

All of these questions relate to the NVRAM chip in the Sun workstation. The contents of the NVRAM chip can become corrupted for a variety of reasons, most commonly, failure of the embedded battery. The battery embedded in the NVRAM chip keeps the clock running when the machine is off and also maintains important system configuration information. This FAQ tells you how to reprogram your NVRAM chip and where to buy a new one, should you need to replace your current NVRAM chip. If you have one of the questions marked (*Replace Chip*), you'll need to purchase a new NVRAM chip. The cost is about US\$20. For the other questions, as long as your machine retains its hostid and ethernet address when turned off, and the clock keeps time when the machine is turned off, reprogramming your NVRAM chip is enough.

One question is marked (*Probably Replace Chip*). Usually, when the NVRAM gets corrupted in this way, this is a symptom that the battery embedded in the NVRAM chip has run out and you need to replace the chip. If the machine is relatively new, you should try reprogramming the NVRAM chip with a hostid and ethernet address using the instructions below, then do a reset at the "ok" prompt to make sure the banner looks as expected. Next turn the machine off for a couple of minutes and turn it on again. If the machine retains its hostid and ethernet address, then you probably don't need to replace the NVRAM chip.

People can be asking two different questions when they are asking how to change the ethernet address of a workstation. If you merely want to change the ethernet address of an interface on your system, you can use the `ifconfig` command and forget about the procedures in this document. The ethernet address in NVRAM is the default address for all ethernet interfaces. If, for some reason, you want to change this default ethernet address in NVRAM, then the instructions in this document will tell you how to do this. Sun workstations are a little strange in this regard, they don't get their MAC addresses from the ethernet hardware itself.

The most up to date version of this document can be obtained from the following sites:

- <http://www.squirrel.com/squirrel/sun-nvram-hostid.faq.html>

Plain text versions of this document are available from the following:

- <http://www.squirrel.com/squirrel/sun-nvram-hostid.faq>

This FAQ is also distributed as part of a larger package for spoofing the hostid on Sun workstations called [change-sun-hostid](#). In particular, parts of [change-sun-hostid](#) can be used to modify the apparent hostid for some or all processes on a UNIX system without messing with the NVRAM. This package even provides a way to

make a host seem to have multiple hostids (different processes see different hostids). If you are interested in changing your hostid to deal with software licence issues, you should probably try the scripts/programs in this package first, as most of them don't make permanent changes to a chip on your motherboard. Changing the NVRAM should be a last resort. You can retrieve this package from:

- <http://www.squirrel.com/squirrel/sun-stuff/change-sun-hostid.tar.gz>

Some related links:

- <http://www.buildmonkey.com/download/hostid.tar.gz> is a software hostid changer by John Birtley that has been tested and verified to work with Solaris 8-10
- [http://www.jaist.ac.jp/~kosmic/private/data/sparc\\_nvram\\_change.html](http://www.jaist.ac.jp/~kosmic/private/data/sparc_nvram_change.html) is a Japanese language file with information on reprogramming and replacing the NVRAM in a sun4c machine.
- <http://www.portnet.co.jp/JIS/technical/WS/sunnvram.japanese.html> is a Japanese language eeprom-nvram FAQ
- For other Sun Hardware questions, check out The Sun Hardware Reference available from <ftp://ftp.picarefy.com/pub/Sun-Hardware-Ref>
- For those who want more technical information about the NVRAM chips themselves, SGS-Thomson has a WWW page with datasheets for the M48T02, M48T08, and M48T59Y on <http://www.st.com>
- Useful tools for Sun Workstations at <http://www.squirrel.com/squirrel/sun-stuff.html>

This document consists of the following parts.

1. [Introduction and Hardware](#)
2. [General sun4c, sun4m, sun4d, sun4u IDPROM Programming](#)
3. [A Quick-and-Dirty Guide to Restoring the NVRAM of a sun4c/m/u Machine](#)
4. [The Sun 3/80](#)
5. [Examples](#)
6. [Odds and Ends](#)
  - a. [Resetting the NVRAM when Stop-N doesn't work](#)
  - b. [Attaching a new battery onto an existing M48T02](#)
  - c. [How to start and stop the NVRAM clock on the Sun 3/80 and sun4c machines](#)
  - d. [Other more arcane methods for modifying the IDPROM](#)
  - e. [The hostid on Solaris 2.5 x86](#)
  - f. [The NVRAM in sun4 architecture machines](#)
  - g. [SparcClassic/Classic X NVRAM differences](#)
  - h. [Enterprise Server NVRAM Programming](#)
7. [Credits](#)

---

## Introduction and Hardware

Every Sun 3/80, sun4c, sun4u, and sun4m architecture machine contains an NVRAM chip (not to be confused with the NVRAM in Prestoserve). This NVRAM chip stores various configuration parameters (e.g. boot device, amount of RAM to test), maintains the clock, and also contains the IDPROM data, which is composed of the ethernet address, date of manufacture, hostid, a version number, and a checksum. The name IDPROM is historical. On older machines such as the sun2, sun3, and sun4 architecture machines the hostid and ethernet address were stored in a PROM

called the IDPROM.

The sun4d machines (e.g. SS1000) also have an NVRAM chip. However, the IDPROM information is stored in a flash EEPROM and is downloaded into the NVRAM during POST. You can also upload the IDPROM information from the NVRAM chip to the flash EEPROM, so you can change the value of IDPROM in the flash EEPROM by merely changing the NVRAM and uploading the new values.

This document is for people who want to accomplish one of the following tasks:

- Install a new NVRAM chip in a Sun 3/80, sun4c, or sun4m machine. Usually this will be due to NVRAM failure or loss of the NVRAM password in full security mode.
- Change the hostid/ethernet address of a Sun 3/80, sun4c, sun4m, sun4u or sun4d machine.
- Restore a machine with a corrupted NVRAM chip in a Sun 3/80, sun4c, sun4u, sun4m machine to working order.

I'm going to focus on methods that involve reprogramming the chip from the FORTH OpenBoot monitor ("ok" prompt). The OpenBoot monitor uses a modified version of FORTH, which is a language similar to that used on the HP calculators which use RPN. You don't need to know any FORTH to reprogram your IDPROM, but the instructions below might make a little more sense if you do know some FORTH.

The NVRAM chip will usually have a white or yellow barcode label on it (except for sun4d). Given the barcode, Sun can reconstruct your original hostid and ethernet address. On newer machines (some SS5, SS20, all Ultras) the number printed on the barcode is the last three bytes of the ethernet address and also the last three bytes of the hostid. The first three bytes of the ethernet address are always 8:0:20 and the first byte of the hostid is determined by the system type ([see table below](#)), so on these machines you can trivially reconstruct the hostid. I have no idea how to do it on the machines with the old style barcode label, but if nothing else, the label makes the NVRAM chip easy to identify.



The following table gives the location of the NVRAM chip for various models of Sun workstation.

3/80	U0205
4/60 (SS1)	U089
4/40 (IPC)	U0901
4/65 (SS1+)	U089
4/20 (SLC)	U1011
4/25 (ELC)	U0813
4/50 (IPX)	U0512

4/75 (SS2)	U0512
4E	U1101
4/10 (SPARCclassic X)	U0707
4/15 (SPARCclassic)	U0707
4/30 (LX/ZX)	U0707
SPARC Xterm 1	U1605
SS4	U1605
SS5	U1506
SS10	U1004
SS20	U1004
SS600MP	U2701
SS240 (Voyager)	U1506
SS1000/1000E	U1007
SS2000/2000E	U1205
U1/170	U2006

**The NVRAM chips are SGS-Thomson Timekeeper (formerly Mostek) chips. They contain an embedded battery with a fixed life. When the machine is off, the battery runs down. It is very common for the battery embedded in the Timekeeper chip in an older Sun (sun4c, sun3x) to fail. The following table lists the NVRAM chip type by machine architecture:**

sun4c	M48T02
sun4m	M48T08/18
sun4d	M48T08
sun4u	M48T59Y
sun3x	M48T02

**The NVRAM chips used to be designated Mostek MK48T02 etc. These chips come in various speeds. The newer SGS-Thomson part numbers are M48T02-200PC1 for the 200ns M48T02, M48T08-100PC1 for the 100ns M48T08. 200ns is adequate for any Sun, but often the faster chips are easier to come by, and there is no harm in buying them. Note that a number of people have reported problems using the 150ns chips in SS2/IPX machines (but 200ns or 120ns works just fine). If you have trouble tracking down a local SGS-Thomson distributor, Mouser electronics sells the SGS-Thomson Timekeeper chips (telephone: 800.346.6873, +1.817.483.5712). You can also order from Mouser on the web at <http://www.mouser.com/>. The Mouser stock numbers as of 1995 are slightly different than the usual part numbers and are as follows:**

SGS-Thomson part	Mouser stock number	speed	capacity
M48T02-200PC1	511-M48T0220PC1	200ns	2kB
M48T02-150PC1	511-M48T0215PC1	150ns	2kB
M48T02-120PC1	511-M48T0212PC1	120ns	2kB
M48T08-150PC1	511-M48T0815PC1	150ns	8kB
M48T08-100PC1	511-M48T0810PC1	100ns	8kB

M48T59Y-70PC1    511-M48T59Y70PC1    70ns    8kB

As of Oct 1995, the prices for these chips from Mouser range from US\$16.00-23.12. I have no affiliation with Mouser other than as a customer.

Dallas Semiconductor makes a clone of the SGS-Thomson M48T02 chips, the DS1642. I've had mixed experience using the Dallas chips in a Sun. My experience is that these chips do not work properly in the following machines: 3/80, SS2, IPX (they fail POST, but otherwise seem OK). However, the Dallas chips are reported to work in the following machines: SS1, SS1+, IPC. I have no idea whether the Dallas chips work properly in the following: SLC, ELC. You can order the DS1642 chips in 120ns or 150ns speeds (150ns is fine) directly from Dallas in quantities up to 10 by calling 1-800-336-6933 and giving Dallas a credit card number. Data sheets and distributor lists for Dallas are available from <http://www.dalsemi.com/>. Stanislav Sinyagin <stas@isf.ru> reports that the M48T12 is another acceptable alternative for the SS1.

Here are some other sources for M48T02/8/18 chips (if you know of sources not listed here, please [send me email](#)). Any SGS-Thomson distributor that will sell the chips in small quantities should do. There is a list of SGS-Thomson distributors at <http://us.st.com/stonline/address/distrib/>. Thanks to simonallen@cix.compulink.co.uk, Kenji Oshima <kosmic@jaist.ac.jp>, Michael J C Firth <mfirth@visual.bt.co.uk>, Bernd Wagner <bwagner@applix.de>, Werner Lappessen <wlappess@ford.com>, Mikael Bergqvist <mbq@aronnax.physto.se>, Klaus Koehnlein <klaus.koehnlein@uni-tuebingen.de>, NoRM <norm@mono.org>, H Richardson <harvey@tromba.demon.co.uk>, Jon Laughton <jon@eoin.demon.co.uk>, Jill Holliday <jill@lynxuk.demon.co.uk> for sending me the sources I have here:

- Farnell Electronic Components (Tel: 01132 636311) in the U.K.
- Maplin Electronics (Tel: 01702 554161) in the U.K. Maplin part number DC01B. Price as of Jan 1998 was GBP 18.99
- CPC in the U.K. Tel: 01772 654455. Ask for a service called "Part Finder". Price as of July 1997 was GBP 12.92
- Wakamatsu Tsusho Parts Shop, Mitsuwa Budg. 2F, 1-11-4 Soto-kanda, Chiyoda-ku, Tokyo 101 Japan tel. (+81)3-3251-4121 / fax. (+81)3-3251-7877
- Jermyn GmbH, Im Dachstueck 6, D-65449 Limburg, phone in Germany : 06431 5080
- MSC-Vertriebs GmbH, Industriestr. 16, D-76297 Stutensee, Tel. +49 (0)72 49 91 00, Fax +49 (0)72 49 79 93 (price as of July 1997 about DM 25)
- elpro, Am Kreuzer 13, D-64372 Ober-Ramstadt, phone in Germany : 06154/63570, fax: 06154/635749, email elpro@t-online.de
- ELFA (<http://www.elfa.se/>) is a Swedish company with branch offices in Norway, Denmark, and Finland. Part number for the M48T02 is 73-712-06 (part name in catalogue: M48T02B-15/ST) and the cost is Skr 182 (excluding VAT) as of April 1997. Phone numbers:
  - Sweden Order: +46 (0)8 735 35 35, Info 735 35 15
  - Finland Order: 08001 11 827
  - Denmark Order: 8001 20 02
  - Norway Order: 800 10 135

---

## General sun4c, sun4m, sun4d, sun4u



# IDPROM Programming

If you have a valid NVRAM chip installed (IDPROM is ok) then before doing anything else, write down a copy of the IDPROM information. You can get it under Sun OS 4.1.x by running `/usr/etc/devinfo -vp` or under Solaris 2.x by executing `/usr/sbin/prtconf -vp`. Alternatively you can get the IDPROM information at the OpenBoot monitor "ok" prompt by typing the command `.idprom` (yes the "." is part of the command). You'll need this information if the NVRAM gets screwed up and you need to try and back out.

## Step 1.

Go to the OpenBoot monitor (ok prompt). You can do this by turning on your machine, pressing L1/Stop-A to interrupt the boot sequence, and entering "new command mode". Alternatively, if your machine is running, just shut down your operating system.

If you are installing a new NVRAM type set-defaults followed by the <Enter> key and then `setenv diag-switch? false` followed by the <Enter> key.

Generally a machine will reset the NVRAM to the default values (excluding the IDPROM information) when it detects a new NVRAM. Still, it is good to do a set-defaults just in case this fails.

## Step 2.

The command to reprogram the IDPROM part of NVRAM is "mkp". The format for the command is

```
<value> <location> mkp
```

where location is read off the following table (all values are in hexadecimal). Note that some Sun clones (e.g. Tatung COMPstation 25 with TWS boot PROM REV 1.7) don't have an "mkp" command. If "mkp" doesn't work on your system, see the section below entitled [Other more arcane methods for modifying the IDPROM](#).

byte(s)	contents
0	always 01 - format/version number
1	first byte of hostid (machine type)
2-7	6 byte ethernet address (first three bytes should be 08,00,20)
8-b	date of manufacture (usually all 0s, doesn't really matter)
c	second byte of hostid
d	third byte of hostid
e	fourth byte of hostid
f	IDPROM checksum - bitwise xor of bytes 0-e

As mentioned above, you can look at the complete idprom by executing `.idprom`. Bytes c through e in the table above are collectively referred to as the serial number. If you convert the concatenation of bytes c-e to decimal, this is the serial number you see when you turn the machine on.

Alternatively, you can use the `idprom@` command to get a particular byte from the

**IDPROM**

**<location> idprom@**

**returns the IDPROM value of byte <location>**

**so**

**<location> idprom@ .**

**prints the IDPROM value of byte <location> because . is the FORTH command to print the value on the bottom of the stack.**

**Now make the changes using mkp. Be very careful and be sure to compute the checksum after making changes. If you don't, you'll get nasty warnings about an incorrect IDPROM checksum on boot. A quick and dirty way to compute and store the checksum in location f is to execute the following at the "ok" prompt after you have made your changes to locations 0-e.**

```
0 f 0 do i idprom@ xor loop f mkp
```

**Don't change the first byte of the hostid to something that doesn't correspond to your system type ([see table below](#)). Similarly, the first three bytes of the ethernet address should be (08,00,20). The first byte of the hostid is often used to determine the architecture when booting from CDROM on some Sun models. If you don't set the first three bytes of the ethernet address to 08,00,20 you might get message which says that you have a defective motherboard (but then I don't know of any other consequences of changing this to some reasonable value, especially avoid ff:ff:ff:ff:ff:ff !. On some systems you can get away with changing the first three bytes of the ethernet address to more or less anything you want).**

**If you'd rather not use the table below, you can find out what you should make the first byte of the hostid by typing real-machine-type . at the "ok" prompt. Note that real-machine-type may not be defined on some machines with older boot proms.**

**e.g. modify the hostid of an IPX to be 57c0ffee and the ethernet address to be 08:00:20:c0:ff:ee. At the OpenBoot monitor prompt (ok)**

```
1 0 mkp
real-machine-type 1 mkp
8 2 mkp
0 3 mkp
20 4 mkp
c0 5 mkp
ff 6 mkp
ee 7 mkp
0 8 mkp
0 9 mkp
0 a mkp
0 b mkp
c0 c mkp
ff d mkp
ee e mkp
0 f 0 do i idprom@ xor loop f mkp
```

**Step 3.**



**If you are on an SS1000, type update-system-idprom at the OpenBoot PROM "ok" prompt.**

**For any of the above machines, now type reset at the ok prompt. Your machine should then attempt to reboot with your new hostid/enet addr.**

The following table matches Sun system models with the first byte of the hostid. This is from a posting to comp.sys.sun.admin by Andy.Behrens@coat.com

01	2/1x0
02	2/50
11	3/160
12	3/50
13	3/2x0
14	3/110
17	3/60
18	3/e
21	4/2x0
22	4/1x0
23	4/3x0
24	4/4x0
31	386i
41	3/4x0
42	3/80
51	SPARCstation 1 (4/60)
52	SPARCstation IPC (4/40)
53	SPARCstation 1+ (4/65)
54	SPARCstation SLC (4/20)
55	SPARCstation 2 (4/75)
56	SPARCstation ELC
57	SPARCstation IPX (4/50)
61	4/e
71	4/6x0
72	SPARCstation 10 or SPARCstation 20
80	SPARCstation Classic, LX, 4, 5, SS1000, Voyager, Ultra 1

---

## **A Quick-and-Dirty Guide to Restoring the NVRAM of a sun4c/m/u machine**

**This is for folks who need to replace their NVRAM chip in a sun4c/m/u machine and don't want to bother with XOR calculations or the details above with mkp. I am assuming that you have a brand new NVRAM chip in hand. All numbers below are in hex.**

**First, decide what ethernet address you want to use and what you want for the last three byte of the hostid. The ethernet address should begin with 08:00:20. There are no restrictions on the last three bytes of the hostid. Say the ethernet address is**

**08:00:20:E3:E4:E5 and the last three bytes of the hostid are H1, H2, H3. The first byte of the hostid will automatically be set according to the system type (real-machine-type variable in the OpenBoot monitor).**

**Turn off the machine. Remove the old NVRAM chip after noting the orientation. Insert the new NVRAM chip. Be sure to insert it in the correct orientation, as installing it in the wrong orientation and powering on the machine will generally destroy the chip. Power up the machine and bring it to the "ok" prompt. At the "ok" prompt execute the following**

```
set-defaults
setenv diag-switch? false
8 0 20 E3 E4 E5 H1H2H3 mkpl
```

**mkpl expects some input (but it doesn't prompt you). The input is a Control-D followed by a Control-R. If mkpl does not print a copyright notice, then it changed the IDPROM. You should make sure by looking at the idprom after using mkpl by executing the .idpromcommand**

**e.g. 8 0 20 13 de ad c0ffee mkpl will set the last three bytes of the hostid to c0ffee and the ethernet address to 08:00:20:13:de:ad.**

***N.B.* mkpl will only work if the IDPROM checksum is invalid. Otherwise it will simply print a copyright notice after you type the Control-R. So, if you can't get mkpl to work, you can try making the IDPROM checksum invalid. You can invalidate the IDPROM checksum in an NVRAM with a valid IDPROM checksum by executing `f idprom@ 1 xor f mkp` (it seems that invalidating the version number will also do, e.g. `17 0 mkp`). If you still can't get mkpl to work, then you should try using mkp as described in the [above section](#). This note was added because on some machines, set-defaults will set the IDPROM checksum according to the other values in the IDPROM.**

---

## The Sun 3/80.

**Here's a way to modify NVRAM IDPROM info on a Sun 3/80 (people have to do this because of the usual battery problem, as with the sun4c machines) This may not be the most elegant procedure, but this should allow you to modify the NVRAM IDPROM info on a sun 3/80. At first glance you might think that the q command would work, but it doesn't seem to actually modify anything when given addresses `>= 0x7d8`**

**Press Stop-A after you turn the 3/80 on**

***N.B.:* ^t is a command in the steps below. Don't try control-T. This procedure has only been tested with 2.9.2 PROMS.**

- 1. At the > prompt execute the command `^t fef04000`. You'll get output which includes the following (samples included)**

```
TIA entry = 33FA500
TIB entry = 33FB80A
PTE=64000049
```

**Write these values down.**

- 2. Change the TIA entry for `fedfa000` to match the above**

```
> m a fedfa000
TIA MAP FEDFA000 [...] ? 33FA500
```

**Type Control-D at the next ? prompt**

**3. Change the TIB entry to match the above:**

```
> m b fedfa000
TIB MAP FEDFA000 [...] ? 33FB80A
```

**4. Now change the pagemap entry to match the above**

```
p fedfa000
PageMap FEDFA000 [...] ? 64000049
```

**5. Now enter the new IDPROM values. Use the following table and be sure to get the checksum right.**

```
o fedfa7d8
...
```

**Jay York <jyork@smart.net> has provided me with a great simplification of the above procedure that I have been unable to test. His procedure skips the steps preceeding this last step (o fedfa7d8). Instead, one can simply do o fef047d8 and enter the IDPROM values at that point. I am leaving the more involved procedure above, because I don't have a 3/80 to test this simplified procedure.**

Sun 3/80 IDPROM

byte	contents
fedfa7d8	01
fedfa7d9	42
fedfa7da-7df	6 byte ethernet address (first three bytes must be 08,00,20)
fedfa7e0-7e3	date of manufacture (all 0s is fine)
fedfa7e4	second byte of hostid
fedfa7e5	third byte of hostid
fedfa7e6	fourth byte of hostid
fedfa7e7	checksum - bitwise xor of bytes at fedfa7d8-fedfa7e6

- 6. Reset the computer by executing the command k 2. It should come back up with the desired hostid and ethernet address.**
- 7. If you are installing a new M48T02 chip, you'll also need to initialise the other values in NVRAM, e.g. boot device, memory, etc. Unlike the sun4c machines, the Sun 3/80 does not set these parameters to sane defaults when it detects a new M48T02. To modify the rest of NVRAM you can use the q command in the monitor. [See the table below.](#)**

**When Sun OS 4.1.x boots, it will kick-start the clock. The first time it boots it will complain that the TOD is not initialized. It shouldn't complain the second time the machine boots. However, if you are using Sun OS 4.1 a bug that was fixed in 4.1.1 will prevent the OS from being able to use the NVRAM clock. One can fix this. change the line in /usr/include/sun3x/devaddr.h and /usr/kvm/sys/sun3x/devaddr.h from**

```
#define V_CLK1ADDR 0xFEDFC7F8
```

to

```
#define V_CLK1ADDR 0xFEDFA7F8
```

and then patching the kernel by searching for 0xFEDFC7F8 in the kernel and changing it to be 0xFEDFA7F8. Thanks to [simonallen@cix.compulink.co.uk](mailto:simonallen@cix.compulink.co.uk) for this information.

The following table of memory locations in NVRAM for the Sun 3 series machines is from the Sun Hardware Reference by James W. Birdsall <[jwbirdsa@picarefy.com](mailto:jwbirdsa@picarefy.com)>. I provide it here so you won't have to dig up the Sun Hardware Reference to restore a 3/80 to working order. I've omitted a few rows that have nothing to do with the 3/80 (mostly 386i stuff)

0x14	Installed memory in MB.		
0x15	Tested memory in MB		
0x16	Monitor screen size	0x00	1152x900 (standard resolution)
		0x12	1024 x 1024
		0x13	1600 x 1280
		0x14	1440 x 1440
		0x15	1024 x 768
0x17	Watchdog reset action	0x00	invoke ROM monitor
		0x12	initate power-on reset (default)
0x18	Boot device	0x00	poll (default)
		0x12	boot from EEPROM/NVRAM specified boot device
0x19-0x1a	SunOS boot device name (in ASCII)	0x78 0x79	xy
		0x78 0x64	xd
		0x73 0x64	sd
		0x69 0x65	ie
		0x69 0x64	id
		0x6c 0x65	le

0x1b-0x1d	SunOS boot device controller, unit, partition numbers	0x00 0x00 0x00	(0,0,0) (default)
0x1f	Primary terminal	0x00	monochrome FB
		0x10	serial port A
		0x11	serial port B
		0x12	0x12 VMEbus and 3/60-P4 color frame buffers (configure locations 0x60c-0x613 when VX and MVX graphics options are installed)
		0x20	non-3/60 P4 color frame buffer
0x20	Power-up banner	0x00	Sun logo display
		0x12	custom banner stored in 0x68-0xb7
0x21	Keyboard click	0x00	off
		0x12	on
0x22-23	Diagnostic boot device name (in ASCII)	Used when NORM/DIAG switch is in DIAG position. On the 3/80 there isn't a switch, NORM/DIAG mode is controlled by location 0x70b. As in 0x1b-0x1d or 0x00, 0x00 to invoke the PROM monitor	
0x24-0x26	Diagnostic boot device controller, unit, partition number		
0x28-0x4f	Diagnostic boot path	Used when in DIAG mode. ASCII codes for path and filename to boot, or all zeros to invoke ROM monitor	
0x50	High resolution number of columns		
0x51	High resolution number of rows		
0x58	Serial port A default baud rate	0x00	9600
		0x12	use rate stored at 0x59-0x5a
0x59-0x5a	Serial port A baud rate	The baud rate as a 16-bit number, MSB first (e.g. 0x04 0xb0 for 1200)	
0x5b	Serial port A DTR/RTS	0x00	assert DTR and RTS
		0x12	do not assert DTR and RTS
0x60	Serial port B	0x00	9600

	default baud rate	0x12	use rate stored at 0x61-0x62. Note that when in DIAG mode, port B runs at 1200 baud and the settings in 0x60-0x62 are ignored
0x61-0x62	Serial port B baud rate	The baud rate as a 16-bit number, MSB first (e.g. 0x04 b0 for 1200)	
0x63	Serial port B DTR/RTS	0x00	assert DTR and RTS
		0x12	do not assert DTR and RTS
0x68-0xb7	Custom banner	ASCII codes for desired banner, padded with spaces and ending with 0x0D, 0x0A in locations 0xB6 and 0xB7	
0x18f	Logo type	0x00	normal Sun logo
		0x06	3D logo for CG6
		0x12	custom logo
0x492	Password mode select	0x01	command secure mode
		0x5e	fully secure mode
		other	non-secure mode
0x493-a	Password	Eight bytes of password in ASCII. If the ROM is 2.8, enter a '@' character before each letter of the password. Enter one letter per location, followed by Return. If the password is less than eight letters, enter 0x00 in the remaining locations. The hexadecimal values of the letters can also be used to enter the password.	
0x70b	3/80 power-on mode	0x06	normal boot
		0x12	diagnostic boot
		other	diagnostic boot

## Examples

***N.B. As I said above, you should be careful that the first byte of the hostid matches the system type.***

- Modify the hostid of an IPX to be 57c0ffee and the ethernet address to be 08:00:20:c0:ff:ee. At the OpenBoot PROM monitor prompt**

```
01 0 mkp
57 1 mkp
08 2 mkp
0 3 mkp
```



```
20 4 mkp
c0 5 mkp
ff 6 mkp
ee 7 mkp
57 8 mkp
0 9 mkp
0 a mkp
0 b mkp
c0 c mkp
ff d mkp
ee e mkp
29 f mkp
```

**Notice the simplification in the above example. If you make the ethernet address 08:00:20:H1:H2:H3 and the four bytes of the hostid ST,H1,H2,H3 where ST is the system type byte, and you put ST,0,0,0 in the date of manufacture field, then the IDPROM checksum will always be 29 (remember all of these numbers are hexadecimal). This make things a bit easier, you can, in general, just enter**

```
01 0 mkp
real-machine-type 1 mkp
08 2 mkp
0 3 mkp
20 4 mkp
H1 5 mkp
H2 6 mkp
H3 7 mkp
real-machine-type 8 mkp
0 9 mkp
0 a mkp
0 b mkp
H1 c mkp
H2 d mkp
H3 e mkp
29 f mkp
```

**and you don't need to calculate the checksum since it will always be 0x29.**

- 2. Change the hostid of an SS10 to be 72c0ffee and the ethernet address to be 08:00:20:c0:ff:ee.**

```
01 0 mkp
72 1 mkp
08 2 mkp
0 3 mkp
20 4 mkp
c0 5 mkp
ff 6 mkp
ee 7 mkp
0 8 mkp
0 9 mkp
0 a mkp
0 b mkp
c0 c mkp
ff d mkp
ee e mkp
```

```
0 f 0 do i idprom@ xor loop f mkp
```

3. Change the hostid of an SS1000 to 80c0ffee. Leave the ethernet address and the date of manufacture intact. Note that the system type byte for the SS1000 is 0x80

```
c0 c mkp
ff d mkp
ee e mkp
0 f 0 do i idprom@ xor loop f mkp
update-system-idprom
```

4. Install a new NVRAM in an IPX. Set the hostid to 57c0ffee and the ethernet address to be 08:00:20:c0:ff:ee .
  1. Turn the machine off.
  2. Remove the old NVRAM chip
  3. Install the new NVRAM chip. Be sure to get the orientation right.
  4. Turn the machine on.
  5. At the OpenBoot monitor prompt execute the following commands:

```
set-defaults
setenv diag-switch? false
8 0 20 c0 ff ee c0ffee mkpl
^D^R
```

where ^D represents Control-D, etc.

## Odds and Ends.

### Resetting the NVRAM (when Stop-N doesn't do it)

You might want to do this to recover from the loss of an NVRAM password (in full security mode) or if you mess up your nvramrc. I think that the safest thing to do is pay the \$20 for a new Timekeeper chip. But several people have reported to me success hot-swapping the NVRAM (i.e. removing and installing a new chip when the system is on).

dowdy@cs.colorado.edu (Stephen Dowdy) writes:

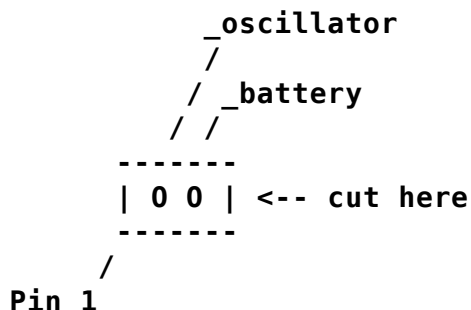
```
> (this may apply to other SPARC models.)
> IPC -- remove NVRAM, power-up without. *carefully* hot-plug it in when OK
> prompt comes up (after it says CHECKSUM failure). do:
>     OK set-defaults
>     OK set-defaults
> then power-cycle
>
> SS2 -- you need to boot from a good NVRAM, then hot-swap the "bad" one
> and "set-defaults". Only if the L1-N (or is it L1-D) thingy
> doesn't work for you.
```

### Attaching a new battery onto an existing M48T02

My take on this procedure is that it isn't worth trying to save the US\$20 it would take to buy a new M48T02. This procedure requires considerable care. I'm including

this excerpt from the old NVRAM FAQ for completeness.

The contents of the (nv)ram are backed up by a 3V lithium battery. It's located together with a quartz on top of the ram in a kind of backpack. The battery is on the side that's opposed to the dot marking pin 1, next to pin 12:



At the point marked above, some kind of nose is reaching down from the backpack over the resin. Carefully cut through the polyester resin filling the dimple. This works best with some kind of mini drill with a small milling head or a razor knife. Buried in the resin you'll find two small diagonal metal connectors :).

Be careful not to short-circuit them, or you'll loose the contents of your nvram (if it was still able to keep them). -That's why you should save them *\*before\** :)

The connector closest to pin 12 is ground, the other (opposing) one +3V. You can now solder some wire to them and connect them to a new 3V lithium battery.

## How to start and stop the NVRAM clock on the Sun 3/80 and sun4c machines

The following information on starting and stopping the clock in the NVRAM chip on the Sun 3/80 was contributed by [simonallen@cix.compulink.co.uk](mailto:simonallen@cix.compulink.co.uk). I've added the tables and the sun4c information below. As it says below, you don't normally need to do this. Sun OS will take care of starting the clock on a new NVRAM chip for you.

Procedure to 'kick start' a new MK48T02 NVRAM clock chip on Sun 3/80

**WARNING :** This procedure is intended to be used on new, blank NVRAM chips only. If you are using it on an existing NVRAM (for some unknown reason) you should follow the procedures described in 'Backing up your NVRAM'.

This procedure shouldn't normally be necessary as SunOS provides this functionality as part of the normal clock initialisation. However, it may prove useful in unforeseen circumstances.

The step numbers correspond to the step numbers in the SGS-Thomson data sheet procedure for 'kick starting' the clock.

The data sheet warns NOT to leave the kick start bit set to 1 for normal use as this will cause excessive current drain and shorten the life of the battery. It does not mention if this is dangerous or by how much the life of the battery is shortened...:-)

The kick start bit is the MSB in location 7FB

(The old NVRAM contents will be displayed by the monitor in the place marked XX in the steps below.)

First halt the machine and get to the monitor prompt.

```
> q 7f8
EEPROM 7F8: XX? 80          Step 1. Set write bit (enable write)
EEPROM 7F9: XX? 0           Step 2. Reset stop bit (not stopped)
EEPROM 7FA: XX? (Press Enter)
EEPROM 7FB: XX? 80          Step 3. Set kick start
EEPROM 7FC: XX? q
> q 7f8
EEPROM 7F8: XX? 0           Step 4. Reset write (disable write)
EEPROM 7F9: XX? q
                               Step 5. Wait for 2 seconds.
> q 7f8
EEPROM 7F8: XX? 80          Step 6. Set write bit
EEPROM 7F9: XX? q
EEPROM 7FA: XX? (Press Enter)
EEPROM 7FB: XX? 0           Step 7. Reset kick start
EEPROM 7FC: XX? q
> q 7f9
EEPROM 7F9: XX? 0           Step 8. Set a dummy time & date...
EEPROM 7FA: XX? 0
EEPROM 7FB: XX? 0
EEPROM 7FC: XX? 4
EEPROM 7FD: XX? 11
EEPROM 7FE: XX? 1
EEPROM 7FF: XX? 96
> q 7f8
EEPROM 7F8: XX? 0           Step 9. Reset write bit
EEPROM 7F9: XX? q
```

And that's it.

Double check that the Kick Start bit is set to zero by doing the following :

```
> q 7fb
EEPROM 7FB: XX? q
```

XX should be a value less than 80 Hex. If it isn't repeat the procedure again.

The related and opposite function to this is Stopping the Clock in the NVRAM. When the clock is stopped, the storage life of the battery is extended because the oscillator is not running. This might be useful if you intend to put the machine into storage or not use it for a while.

Again, halt the machine and get to the monitor prompt.

```
> q 7f8
EEPROM 7F8: XX? 80          Set write bit
EEPROM 7F9: XX? 80          Set stop bit
EEPROM 7FA: XX? q
> q 7f8
EEPROM 7F8: XX? 0           Reset write bit
```

EEPROM 7F9: XX? q

Simon Allen [simonallen@cix.compulink.co.uk](mailto:simonallen@cix.compulink.co.uk) 1996.

As Simon Allen mentions, the kick-start procedure is not normally necessary as Sun OS automatically starts the clock in an NVRAM with a stopped clock. However, if you are putting a machine into long term storage, it is nice to be able to stop the clock as doing this will significantly lower current drain on the battery. The following information from the M48T02 data sheet gives information on the clock.

The M48T02 Register Map

Address	Data								Function	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
7ff	-	-	-	-	-	-	-	-	Year	00-99
7fe	0	0	0	-	-	-	-	-	Month	01-12
7fd	0	0	-	-	-	-	-	-	Date	01-31
7fc	0	FT	0	0	0	-	-	-	Day	01-07
7fb	KS	0	-	-	-	-	-	-	Hours	00-23
7fa	0	-	-	-	-	-	-	-	Minutes	00-59
7f9	ST	-	-	-	-	-	-	-	Seconds	00-59
7f8	W	R	S	-	-	-	-	-	Control	

- ST is STOP BIT
- W is WRITE BIT
- R is READ BIT
- S is SIGN BIT
- FT is FREQ TEST BIT
- KS is KICK START BIT

## Instructions for starting the clock

1. Set the Write Bit to 1
2. Reset the Stop Bit to 0
3. Set the Kick Start Bit to 1
4. Reset the Write Bit to 0
5. Wait two seconds
6. Set the Write Bit to 1
7. Reset the Kick Start Bit to 0
8. Set the Correct Time and Date
9. Reset the Write Bit to 0

N.B. Leaving the KS bit set will cause the Clock to draw excessive current and will shorten the battery life.

## Instructions for stopping the clock

1. Set the Write Bit to 1
2. Set the Stop Bit to 1
3. Reset the Write Bit to 0

Given the above tables, Simon Allen's instructions for the 3/80, and the next section on which tells how to map the NVRAM to virtual memory on sun4c machines, it is a

**relatively easy matter to see how to stop and start the clock on a sun4c machine.**

### **Stopping the clock on a sun4c machine**

2000000 obio 0 map-page	map NVRAM to page 0
80 7f8 c!	set write bit
80 7f9 c!	set stop bit
0 7f8 c!	reset write bit

### **Starting the clock on a sun4c machine**

2000000 obio 0 map-page	map NVRAM to page 0
80 7f8 c!	set write bit
0 7f9 c!	reset stop bit
80 7fb c!	set kick start
0 7f8 c!	reset write bit
wait for two seconds	
80 7f8 c!	set write bit
0 7fb c!	reset kick start
0 7f9 c!	set dummy time and date
0 7fa c!	(if necessary)
0 7fb c!	
4 7fc c!	
11 7fd c!	
1 7fe c!	
96 7ff c!	
0 7f8 c!	reset write bit

## **Other more arcane methods for modifying the IDPROM**

mkp and mkpl are not the only way to modify the IDPROM. Before I discovered these commands, I used to use procedures like the ones below. The material in this section assumes that you have access to the OpenBoot PROM manual which is part of the Solaris 2.x Answerbook.

**You can use the OpenBoot monitor to find the virtual address of the NVRAM. The basic steps are:**

- 1. cd to the eeprom device (you can find its exact name via show-devs)**
- 2. execute .attributes or .properties depending upon whether you have V2 or V3 of the OpenBoot PROM.**
- 3. execute device-end followed by reset**

**For example**

```
ok show-devs
...
/obio/eeprom@0,200000
...
ok cd /obio/eeprom@0,200000
ok .attributes
address: ffee9000
```



```
...
ok device-end
ok reset
```

From this point, you can modify and look at the NVRAM by using the `c!`, `dump`, and `c?` commands in the OpenBoot PROM. You can also get the virtual address of the NVRAM by looking at the output of `/usr/etc/devinfo -vp` under Sun OS 4.1.x or `/usr/sbin/prtconf -vp` under Solaris 2.x. On sun4c machines, the IDPROM starts at offset 0x7d8 from the start of the NVRAM. On sun4m, sun4d, and sun4u machines the offset is 0x1fd8.

From here one can also find the physical address of the NVRAM using `pgmap?`. In the above example, if we type `ffee9000 pgmap?` at the OpenBoot prompt, we get a few lines of output, one of which is

```
Physical: 0.7120.0000
```

which means on this machine the physical address is 71200000 in address space 0.

You can also use `map-page` to map the physical address of the page containing the IDPROM to virtual address 0. To do this you need to know the physical address of the NVRAM. You can find it as above or just use the following table:

machine type	address	space	arch -k
SS1, SS1+, SS2, ELC, IPC, IPX, SLC	02000000	obio	sun4c
Classic, LX, SS5, SS4, Voyager	71200000	0	sun4m
SS10, SS20, 6x0/MP	f1200000	f	sun4m
SS1000	00280000	f	sun4d

For the sun4m and sun4d machines, the page size is 4kB (0x1000) so the simplest thing to do is map the second page of the NVRAM. This is because we're using `map-page` and it only maps one page at a time.

e.g. Modify the hostid of an IPX to be 57c0ffee and the ethernet address to be 8:0:20:c0:ff:ee

```
02000000 obio 0 map-page
1 7d8 c!
57 7d9 c!
08 7da c!
0 7db c!
20 7dc c!
c0 7dd c!
ff 7de c!
ee 7df c!
57 7e0 c!
0 7e1 c!
0 7e2 c!
0 7e3 c!
c0 7e4 c!
ff 7e5 c!
ee 7e6 c!
```

29 7e7 c!

**e.g. to modify the hostid of an SS10 to be 72c0ffee and the ethernet address to be 08:00:20:c0:ff:ee, do the following**

```
f1201000 f 0 map-page
01 fd8 c!
72 fd9 c!
08 fda c!
0 fdb c!
20 fdc c!
c0 fdd c!
ff fde c!
ee fdf c!
72 fe0 c!
0 fe1 c!
0 fe2 c!
0 fe3 c!
c0 fe4 c!
ff fe5 c!
ee fe6 c!
29 fe7 c!
```

**Note that we added 0x1000 to the physical address in the above table, as explained above.**

## **The hostid on Solaris 2.5 x86**

**Intel processor machines don't have an IDPROM. Sun uses a different mechanism to generate the hostid. When the operating system is initially installed a pseudo-random hostid is generated. It appears that this pseudo-randomly generated hostid will always be between 1 and 3b9aca00. The hostid is based on eight bytes of serialisation information in the kernel module /kernel/misc/sysinit. This is in contrast to the situation on SPARC machines where the hostid is based on the IDPROM.**

**/kernel/misc/sysinit contains code which initialises the variable hw\_serial in the kernel based on the serialisation information. On both SPARC and x86 versions of Solaris 2.5, hw\_serial stores the hostid as a decimal C string.**

**Other than the eight bytes of serialisation information the /kernel/misc/sysinit files do not differ between machines. Four of the serialisation bytes depend upon the other four bytes, so the hostid is somewhat tamper resistant. If the serialisation information is tampered with carelessly or the sysinit module fails to load for some other reason, the hostid of the machine will be 0. A little more obfuscation is done in the code, i.e. hw\_serial is not referenced directly in the module, but indirectly via the pointer \_hs1107.**

**This means that if you need to have two machines with the same hostid for some reason (say, to have a backup server with the same hostid in case your primary server malfunctions), you can just copy the /kernel/misc/sysinit file from one machine to another.**

**Moreover, it seems that initialising hw\_serial is the only function performed by the sysinit module. Hence, it is a simple matter to replace /kernel/misc/sysinit yielding a machine with whatever hostid one wants, by compiling a simple C program for a**

loadable kernel module which sets `hw_serial` to the desired value.

C code for a generic replacement `sysinit` module is included in [change sun hostid](#) which is available from the sites listed [near the beginning of this document](#). Replacing part of the operating system is probably not the best way to achieve this effect. In general, I'd recommend using one of the other modules in [change sun hostid](#) as there is less risk of damaging things and rendering the system unbootable, but a few people have asked for this.

## The NVRAM in sun4 architecture machines

The sun4 machines (e.g. Sun 4/1xx, 4/2xx, 4/3xx, etc.) also have M48T02 chips. These chips do not store the hostid and ethernet address, which are in an actual PROM, but they do keep track of the time of day and the system configuration information. Replacing the NVRAM chip is a relatively simple matter, the only trick is to set the appropriate values in the Sun PROM monitor using the `q` command then boot the operating system to kick-start the clock. You can use the [table for the Sun 3/80](#) to set the values in the old Sun PROM monitor. The only significant difference is that normal/diagnostic boot is controlled by a physical switch instead of the byte at location 0x70b. See the [Sun Hardware Reference](#) (referred to above), your machine documentation, or the Sun FE Handbook if you need more information on the Sun PROM monitor.

## SparcClassic /Classic X Terminal NVRAM differences

This section was supplied by [Gary Cook <gcook@netwiz.net>](mailto:gcook@netwiz.net). I have not been able to try this personally because I don't currently have access to this type of machine.

Change the following locations in NVRAM to switch between Classic and Classic

Address	ClassicX Value	Classic Value
71202004	ff	00
71202005	12	00
71202006	08	00
71202007	36	00

### Notes:

- Classic X requires all locations to be set to above values. If they are not, the system takes on the Classic personality.
- Bootprom for Classic X cannot be Version 2.9. This version will ignore the above NVRAM changes. Classic X requires earlier versions, such as 2.12.

### Procedure

Classic --> Classic X

```
ok
ff 71202004 20 spacec!
12 71202005 20 spacec!
```

```
08 71202006 20 spacec!  
36 71202007 20 spacec!
```

**Classic X --> Classic**

```
ok  
00 71202004 20 spacec!  
00 71202005 20 spacec!  
00 71202006 20 spacec!  
00 71202007 20 spacec!
```

## Enterprise Server NVRAM Programming

This section was supplied by [Gary Cook <gcook@netwiz.net>](mailto:gcook@netwiz.net). I have not been able to try this personally because I don't currently have access to this type of machine.

I had an opportunity to experiment with some Enterprise 3000/4000/5000/6000 systems and thought I would pass along what I found.

The procedure works the same as listed in the FAQ for other systems: <value>  
<location> mkp

80 is the system type for all Enterprise X000 systems. As has been the recent trend, the last 3 digits of the ethernet address are also the ones used by the hostid.

The checksum procedure works the same too

```
0 f 0 do i idprom@ xor loop f mkp
```

When writing to the Enterprise X000 NVRAM, you are writing to the one located on the clock board. Once it is programmed, you can copy it's contents to the NVRAM's on the I/O boards by using the command:

```
copy-clock-tod-to-io-boards
```

You can also program a blank or corrupt NVRAM on a board by copying a valid copy from another board. Only the clock board and I/O boards have NVRAMs. The CPU/Memory boards do not have NVRAMs.

**Example: Copy from an I/O board NVRAM in slot 1 to the clock board NVRAM.**

```
01 copy-io-board-tod-to-clock-tod
```

**NOTE: 01 above is the slot number of the I/O board.**

After self test, you may get a message saying the Clock TOD does not match any I/O boards. This may have been caused by swapping the clock or I/O board without keeping the original NVRAM. It can also be caused by a glitch in the Boot PROM code that causes the system to think there is a mismatch after you power off the system. Sun has a patch or fix for this problem. You can reprogram the Flash memory with an update.

To fix the mismatch, you have a choice of updating the clock NVRAM from an I/O board or the other way around. Keep in mind that the one displayed in the banner is the one on the clock board. If the clock board has the correct hostid, then update all the I/O boards with a single command:

## copy-clock-tod-to-io-boards

If one of the I/O boards has the correct hostid, then copy it's contents to the clock board with: (Change 01 to equal the slot # of the valid I/O board.)

01 copy-io-board-tod-to-clock-tod

You should be able to examine the contents of each I/O board's NVRAM to see which one has the correct hostid / Ethernet address.

---

## Credits

I'd like to thank the following people who have helped make this a better document. If I've left anyone off this list who should be on it please [drop me a note](#) and I'll add the name to the list.

- Kevin Murty for the use of an SS1000 for testing
- [simonallen@cix.compulink.co.uk](mailto:simonallen@cix.compulink.co.uk) for the kick-start information, information on the Sun OS 4.1 clock bug on the 3/80, a U.K. source of NVRAM chips.
- Michael Firth <[mfirth@visual.bt.co.uk](mailto:mfirth@visual.bt.co.uk)> for a U.K. source of NVRAM chips.
- [dowdy@cs.colorado.edu](mailto:dowdy@cs.colorado.edu) (Stephen Dowdy)
- [drzob@vectrex.login.qc.ca](mailto:drzob@vectrex.login.qc.ca) (Denis Solaro)
- [coughlin@cfa0.harvard.edu](mailto:coughlin@cfa0.harvard.edu) (Michael Coughlin) for numerous suggestions and bits of information.
- [sameer@c2.org](mailto:sameer@c2.org)
- Kenji Oshima <[kosmic@jaist.ac.jp](mailto:kosmic@jaist.ac.jp)> for the GIF image of a M48T02 and a source for replacement NVRAM chips in Japan.
- Eddie Berin
- Marek Jedryszek <[djslamm@pg.gda.pl](mailto:djslamm@pg.gda.pl)>
- Andy Michael <[andy@plumgate.demon.co.uk](mailto:andy@plumgate.demon.co.uk)>
- Bernd Wagner <[bwagner@applix.de](mailto:bwagner@applix.de)>
- Lonnie Borntreger <[67goat@exchange.wcc.att.com](mailto:67goat@exchange.wcc.att.com)>
- Jay York <[jyork@smart.net](mailto:jyork@smart.net)>
- James Lin <[linja@fasecon.econ.nyu.edu](mailto:linja@fasecon.econ.nyu.edu)>
- Werner Lappessen <[wlappess@ford.com](mailto:wlappess@ford.com)>
- Stanislav Sinyagin <[stas@isf.ru](mailto:stas@isf.ru)>
- Mikael Bergqvist <[mbq@aronnax.physto.se](mailto:mbq@aronnax.physto.se)>
- Klaus Koehnlein <[klaus.koehnlein@uni-tuebingen.de](mailto:klaus.koehnlein@uni-tuebingen.de)>
- NoRM <[norm@mono.org](mailto:norm@mono.org)>
- R. Blake Von Haden <[bvh@mspusa.com](mailto:bvh@mspusa.com)>
- Jon Laughton <[jon@eoin.demon.co.uk](mailto:jon@eoin.demon.co.uk)>
- Tatjana Heuser <[pierrot@sarastro.isdn.cs.tu-berlin.de](mailto:pierrot@sarastro.isdn.cs.tu-berlin.de)>
- Adrie Koolen <[adrie@ica.philips.nl](mailto:adrie@ica.philips.nl)>
- J"org Schilling <[js@cs.tu-berlin.de](mailto:js@cs.tu-berlin.de)>
- H Richardson <[harvey@tromba.demon.co.uk](mailto:harvey@tromba.demon.co.uk)>
- Alexander Shenkin <[ashenkin@wdl.lmco.com](mailto:ashenkin@wdl.lmco.com)>
- Jill Holliday <[jill@lynxuk.demon.co.uk](mailto:jill@lynxuk.demon.co.uk)>
- Tim Hogard <[thogard@abnormal.com](mailto:thogard@abnormal.com)>
- Steve Powers
- John Birtley
- Gary W. Cook <[gcook@netwiz.net](mailto:gcook@netwiz.net)> for the information on the Classic/ClassicX differences and on Enterprise server NVRAM programming