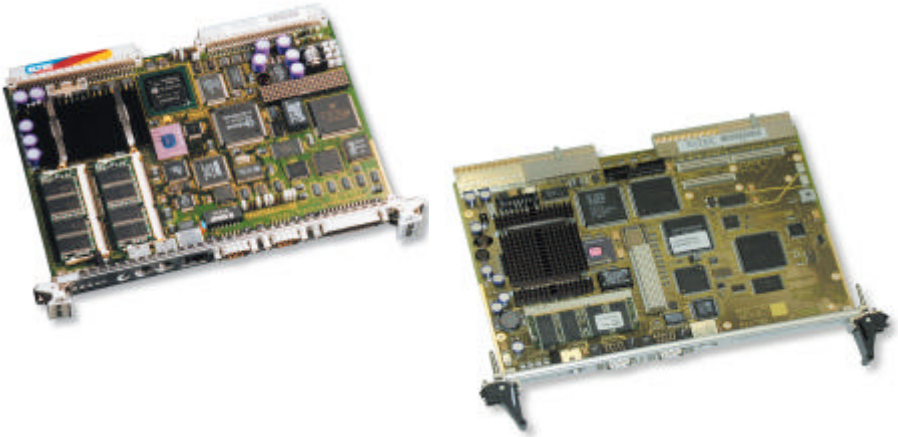
BAB740 / BAB750



nvRAM settings
Revision 1B

ELTEC
elektronik mainz

Revision History

Revision	Changes	Date
1A	First Edition, Valid for Software revision 2A,	09.05.2001 rae
1B	Valid for Software revision 3A (W-O9B7-106C) - Boot message changed -	26.09.2001 rae

Table of Contents

List of Tables

List of Figures

Conventions

If not otherwise specified, addresses are written in hexadecimal notation and identified by a leading "0x".

b	bit
B	byte
K	kilo, means the factor 400 in hex (1 024 decimal)
M	mega, the multiplication with 100 000 in hex (1 048 576 decimal)
MHz	1 000 000 Hertz

Software-specific abbreviations:

<BS>	Back Space (0x08)
<CAN>	Control-X (0x19)
<Ctrl>	Control
<CR> <ENTER>	Carriage Return (0x0D)
<ESC>	Escape Character (0x1B)
<LF>	Line Feed (0x0A)
<SP>	Space (0x20)
NMI	Non-maskable Interrupt

How to Use this Manual

Document Conventions

Font Types:

Font	Use
Arial, 8 Pt or 7 Pt	Tables and drawings
Arial, 10 Pt	Signal names
Times, italic	*Notes*
`Courier, bold`	`Program code, function names, commands, file names, module names`
Times, bold	**Emphasized text**

Other conventions:

Indicates information that requires close attention.

Indicates critical information that is essential to read.

Indicates information that is imperative to read. Skipping this material, possibly causes damage to the system.

1 Introduction

The non volatile RAM is used to store diverse configuration parameters. The size of this RAM is 8192 Byte and it content is mirrored in the system information. The content of the non volatile RAM is also available as the memory module 'SystemInfo' under OS-9. The structure of the data is described behind (section 7 on page 7-1).

After power-on or a system-reset the hex switch in front of the BAB-740 will be read, the system information will be load from the desired source and the universe-chip will be initialized. The available sources are:

Number	source of system information
0	factory setting
1	nvRAM setting
2	reserved
3	reserved
4	reserved
5	reserved
6	reserved
7	reserved
8	datamodule bootcnfg_8
9	datamodule bootcnfg_9
a	datamodule bootcnfg_a
b	datamodule bootcnfg_b
c	datamodule bootcnfg_c
d	datamodule bootcnfg_d
e	datamodule bootcnfg_e
f	datamodule bootcnfg_f

2 First steps

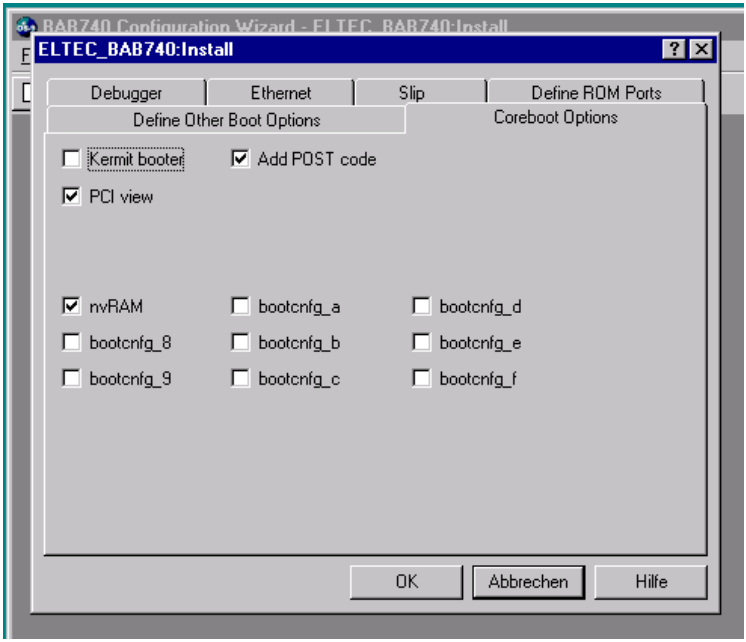
The following modules must be installed via the configuration wizard:

nvram	to handle the system information in the non volatile RAM
nvramsetup	to edit the settings
initUniverse	to initialize the universe during the coreboot
linkSystemInformation	to access the system information under OS-9
genBootcnfg	to generate modules with valid nvRAM-informations to store them in the flash EEPROM

The modules **`nvram`**, **`nvramsetup`** and **`initUniverse`** will be merged into the file coreboot, if the nvRAM option is checked, found in

Configure=>Coreboot=>Main Configuration=>Coreboot Options

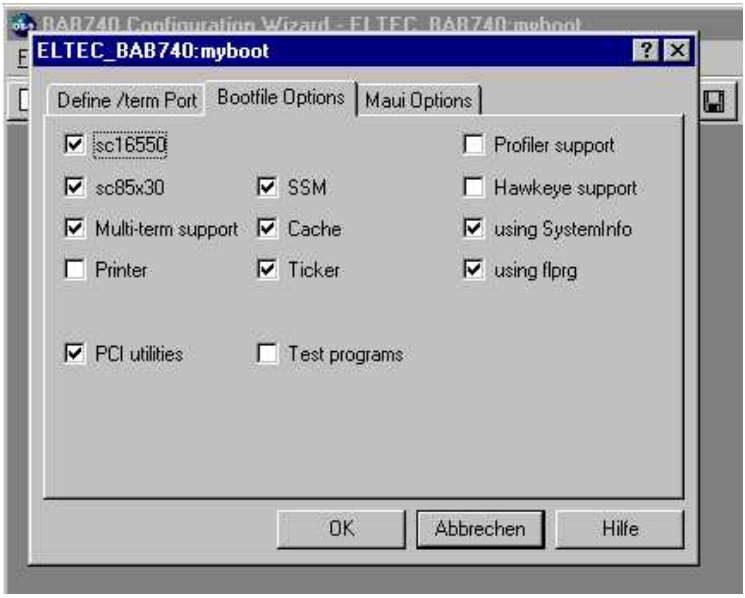
Figure 2—1: Coreboot Options



To use the system information within OS-9, the modules **`linkSystemInformation`** and **`genBootcnfg`** must be merged into the file bootfile. This happend, if the **`using systeminfo`** option is checked too, found in

Configure=>Bootfile=>Configure System Options

Figure 2—2: Configure System Options



3 Setup

To edit the internal used parameters, the **nvram**-utility can be called from the coreboot-menu. If the autoboot is activ, the menu can be reached by pressing the **<SPACE>**-key during the countdown of the autoboot (an example is shown in figure 3-1 on page 3-2).

Figure 3—1: Boot Example

```
*** ELTEC Elektronik AG, Mainz ***

BAB-PPC Monitor Version 1.2/3

Init MPU/MSR/FPU/Segment registers.
Init SuperIO (polled output on COM1).
Activating 1st level cache ---------------------------------------- OK
Setting MPC106 register-------------------------------------------- OK
Reading SPD of bank0/1 -------------------------------------------- OK
      RAM-Type: SDRAM
Reading SPD of bank2/3 -------------------------------------------- FAILED
Activating  32 MByte.

PowerPC 74x/75x Ver.0008 Rev.0202 at 267 / 66 MHz

PCI devices on local bus ...
```

No.	VendorId	DeviceId	Device Class	Sub-Class
00	1057	0002	Bridge device	00
0B	10AD	0565	Bridge device	01
0D	1000	0006	Mass storage controller	00
0E	1011	0019	Network controller	00
1E	10E3	0000	Bridge device	80

```
Press any key to skip memory test :    32768 KByte

OS-9 Bootstrap for the PowerPC(tm) (Edition 64)

nvram: edition 14, generating NVRamModule ...
nvram: get system information from non volatile RAM
initUniverse: initialize universe
ll21040: MediaSelect SYM (id = 0x0)
ll21040: Ethernet started

*** autoboot in  9 seconds; press <SPACE> to abort ...
```

BOOTING PROCEDURES AVAILABLE ---------- <INPUT>

Boot over Ethernet ---------- <eb>
Boot embedded OS-9000 in-place ---------- <bo>
Copy embedded OS-9000 to RAM and boot ---------- <lr>
PCI View Utility ---------- <pciv>
Enter system debugger ---------- <break>
Enter setup for nvRAM ---------- <nvram>
Restart the System ---------- <q>

Select a boot method from the above menu:

3.1 The Main Menu

In the main menu you can select a separate block to view and edit the corresponding parameters, reset the nvram to the manufacturer settungs, read the settings from the nvRAM or write the settings back to the nvRAM.

Figure 3—2: Main Menu of the nvRAM Setup

```
nvram setup:

configure boot parameters ------------------------------------------ <boot>
configure I/O parameters ------------------------------------------- <io>
configure special parameters --------------------------------------- <spec>
configure VME interface (Universe II)------------------------------ <univ>
get manufacturer setting ------------------------------------------- <init>
read settings from non volatile RAM -------------------------------- <read>
write settings to non volatile RAM --------------------------------- <write>
exit to mainmenu --------------------------------------------------- <exit>

Select an item from the above menu:
```

The commands of the main menu are case insensitive and must be written in the full length. The commands **boot**, **io**, **special** and **universe** calls sub menus, the command **init** sets the complete settings to the manufacturer setting, **read** reads back the nvRAM and the command **write** copy the actual settings back to the nvRAM. With **exit** the system goes back to the coreboot-menu.

3.2 The Submenus

All menu entries contains submenus like the menu shown behind, expect the universe menu, because the universe setting is too complex for an easy handling.

Figure 3—3: The Mostly Used Submenu

```
show parameters ------------------------------------------------- <show>
change parameters ----------------------------------------------- <change>
set local part to factory setings ------------------------------- <init>
read local part from non volatile RAM --------------------------- <read>
write local part to non volatile RAM ---------------------------- <write>
return to main menu --------------------------------------------- <ret>

Select an item from the above menu:
```


The commands **init**, **read** and **write** has the same meaning like the commands in the main menu, but they works only with the parameters, which can be edit in the respective submenu. The command **return** is used to go back to the mainmenu, **show** shows the actual setting and change is used to edit the settings.

If an value shall be change, a help line come out under the value. It includes some hints how the desired setting shall be entered. This line shows the useable keys and, if necessary, the area of values, which can be used.

3.3 Boot Parameters

The boot-parameters consist of two parts --- the first three entries are global settings for all boot devices, the last entries depend on the selected boot device.

The global entries are the bootdevice and the autoboot-delays for ELTEC-autoboot (interruptable) and for Microware-autoboot (not interruptable). The entries, which are depend of the boot device, will be explained behind.

Figure 3—4: Content of the Ethernet Boot Parameters

boot device	: Ethernet
auto boot delay for ELTEC	: 3
auto boot delay for MicroWare	: 3
net boot retry	: 3

3.3.1 Select the Boot Device

The boot device can be selected with the **<SPACE>**-key. Each hit selects an other device, a hit to the return-key ends the selection. The order of the boot devices is:

ethernet	BootP via ether-network
embedded OS-9	copies the kernel from ROM to RAM and boot
SCSI hard disk	boot kernel from an SCSI hard disk
SCSI floppy disk	boot kernel from an SCSI floppy disk
IDE hard disk	boot kernel from an IDE hard disk

3.3.2 Select the Autoboot Delay for ELTEC

The autoboot delay for ELTEC is a time, in which the boot process can be stopped by a hit with the **<SPACE>**-key. The time can be between 0 or 3600 seconds. 0 seconds means no auto boot.

This parameter is useable to delay the boot process of the BAB 740. One reason to do this can be, that the bootP-server needs some time to boot itself. If the bootP-server needs two minutes to boot, then the autoboot delay should be set to 120 seconds.

During the boot delay, booting can be abort by pressing the **<SPACE>**-key. In this case, the coreboot menu will be shown.

3.3.3 Select the Autoboot Delay for Microware

The autoboot delay for microware is a delay time for booting. The time can be between 0 or 3600 seconds. This boot delay has the same resons like the autoboot delay for ELTEC, but it is unbreakable.

3.3.4 Ethernet Depend Entries

3.3.4.1 Select the Netboot Retry

With the netboot retry the numbers of bootp-retrys can be selected. The value can be between 0 or 20 tries.

3.3.5 Embedded OS-9 Depend Entries

Figure 3—5: Content of the RAM Boot Parameters

boot device	:	embedded OS-9000
auto boot delay for ELTEC	:	10
boot delay for MicroWare	:	3
base address for romboot	:	0x0

3.3.5.1 Select the Base Address for ROMboot

This address means the start address of the kernel in the rom. The address can be set with decimal or hexadecimal numbers. If hexadecimal numbers are used, the value must start with **0x**.

3.3.6 SCSI Hard Disk Depend Entries

Figure 3—6: Content of the SCSI Hard Disk Boot Parameters

boot device	:	SCSI hard disk
auto boot delay for ELTEC	:	12
boot delay for MicroWare	:	3
SCSI ID of the hard drive	:	0

3.3.6.1 Select the SCSI ID of the Hard Drive

This entry contains the SCSI id of the boot harddisk drive. The ID can be between 0 or 6, but the ID 7 used for the internal SCSI controller.

3.3.7 SCSI Floppy Disk Depend Entries

Figure 3—7: Content of the SCSI Floppy Disk Boot Parameters

boot device	:	SCSI Floppy disk
auto boot delay for ELTEC	:	12
boot delay for MicroWare	:	3
SCSI ID of the floppy	:	3

3.3.7.1 Select the SCSI ID of the Floppy

This entry contains the SCSI id of the boot floppy drive. The ID can be between 0 or 6, but the ID 7 used for the internal SCSI controller.

3.3.8 IDE Hard Disk Depend Entries

Figure 3—8: Content of the IDE Hard Disk Boot Parameters

boot device	:	IDE hard disk
auto boot delay for ELTEC	:	3
boot delay for MicroWare	:	3
port address	:	0xfe0001f0

3.3.8.1 Select the Port Address

The port address is need to select the IDE device to boot. For example, if the system shall boot from an PC-Card, the port address must be the address of the PC-Card, because the PC-Card is handled from the system in the same manner as an IDE drive. The address can be set with decimal or hexadecimal numbers. If hexadecimal numbers are used, the value must start with **0x**.

3.4 I/O Parameters

This parameters defines the dataformat at the ethernet-device only.

Figure 3—9: Content of the I/O Parameters

ethernet format	:	half duplex
ethernet speed	:	10 MB/s
ethernet IP	:	0.0.0.0
gateway IP	:	0.0.0.0
subnet mask	:	0.0.0.0

3.4.1 The Ethernet Address

This address can be writen like a normal string. If the address is '0.0.0.0', the ethernet address of the board will be taken from an other source, in dependence of the boot device:

Ethernet	The address will be got from the a bootp-server
all others	The address will be taken from the datamodule **`cnfgdata`**

3.4.2 Select the Ethernet Speed

The ethernet speed can be selected with the **`<SPACE>`**-key. Each hit selects an other speed, a hit to the return-key end the selection. The order of the ethernet speeds is:

10 MB/s set ethernet speed to 10 MBit per second

100 MB/s set ethernet speed to 100 MBit per second

3.4.3 Select the Ethernet Format

The ethernet format can be selected with the **<SPACE>**-key. Each hit selects an other format, a hit to the return-key end the selection. The order of the ethernet formats is:

half duplex selects half duplex format

full duplex selects full duplex format

3.4.4 Ethernet IP

The ethernet IP is an IPv4-address separated by data. Each ethernet IP can be written with decimal or hexadecimal values. A hexadecimal value must start with a **0x**.

3.4.5 Gateway IP

The gateway IP is an IPv4-address separated by data. Each gateway IP can be written with decimal or hexadecimal values. A hexadecimal value must start with a **0x**.

3.4.6 Subnet Mask

The subnet mask is an IPv4-address separated by data. Each subnet mask can be written with decimal or hexadecimal values. A hexadecimal value must start with a **0x**.

3.5 Special Parameters

This submenu is used to set parameters for special purposes. At this time, the rombug can be enabled oder disabled.

Figure 3—10: Content of the Special Parameters

special flags:

enable debugger: ( )

3.5.1 Enabling the Debugger

The state of the debugger can be switched by pressing the **<SPACE>**-key. If the debugger is enabled, it will be called before showing the coreboot menu or booting the operating system. For further details about the rombug, please read the appropriate manual.

3.6 Universe Parameters

Before starting to configure the universe chip, the universe manual must be read, to inform about the features of this powerful chip. The settings described behind reflects not the whole settings of the universe chip, but only a small subset, which is needed for the base initialization of the universe.To use all features of the universe chip, the values can be edit directly in the system information.

The menu for the universe selection differs from the other menus:

Figure 3—11: Select a Kind of Parameters


```
configure universe parameters

select parameters ---------------------------------------------------- <select>
initialize universe with new settings ------------------------------- <reinit>
set local part to factory setings ----------------------------------- <init>
read local part from non volatile RAM ------------------------------- <read>
write local part to non volatile RAM -------------------------------- <write>
return to main menu ------------------------------------------------- <ret>

Select an item from the above menu:
```

The entries **show** and **change** of the other submenus are combined to **select**. The new command **reinit** is used, to set the universe to the new settings without a new system boot. The commands **init**, **read** and **write** are similar to the other subware.

Figure 3—12: Select a Kind of Parameters

select universe parameters:

PCI slave windows ---------- <PCIwin>

VME slave windows ---------- <VMEwin>

PCI interrupt ---------- <PCIint>

VME interrupt ---------- <VMEint>

miscellaneous parameters ---------- <misc>

PCI miscellaneous ---------- <PCImisc>

return to universe menu ---------- <return>

Select an item from the above menu:

3.6.1 Show the Parameters of all PCI Slave Windows

The PCI slave windows are need, to access to other boards on the VME-bus from the BAB 740. There are eight windows available (lsi0-lsi7), the first three are used for the access in the same manner as the access from 68k-boards. lsi0 is configured for long I/O, lsi1 is configured for standard I/O and lsi2 is configured for short I/O. The window lsi3 is reserved for special usage and the windows lsi4-lsi7 can be freely used from the user programms.

Figure 3—13: Show the Parameters of all PCI Slave Windows

PCI slave window

base register	base	size	target	Flags
0x100 - lsi0	0x80000000	0x10000000	0x80000000	0x80c20000
0x114 - lsi1	0xc0000000	0x1000000	0xff800000	0xc0410000
0x128 - lsi2	0xefff0000	0x10000	0xffff0000	0x80400000
0x13c - lsi3	0x0	0x0	0x0	0x0
0x1a0 - lsi4	0x0	0x0	0x0	0x0
0x1b4 - lsi5	0x0	0x0	0x0	0x0
0x1c8 - lsi6	0x0	0x0	0x0	0x0
0x1dc - lsi7	0x0	0x0	0x0	0x0

change parameters (y/n)?

If the parameters of any window shall be changed, the next question must be answered with **y**, otherwise with **n**. If the parameters of a window shall be changed, a question for the number of the window to change occur.After selecting a window between **0** and **7**, each parameter can be set by giving a numeral value, except the flags. This value can be a decimal or a hexadecimal value. A hexadecimal value must start with **0x**. The flags can be edited directly by their meanings. It compenents are:

image enable	=> disable/enable
posted write enable	=> disable/enable
VME maximum datawidth	=> 8/16/32/64 Bit
VME address space	=> 16/24/32 Bit
Program AM code	=> data/program
supervisor AM code	=> non-privileged/supervisor
BLT on VMEbus	=> no/yes
PCI Bus Space	=> memory/IO/configuration space

3.6.2 Show the Parameters of all VME Slave Windows

The VME slave windows are needed, to access the memory of the BAB 740 from outside via VME bus.

Figure 3—14: Show the Parameters of all VME Slave Windows

VME slave window

base register	base	size	target	Flags
0xf00 - vsi0	0xa0000000	0x10000000	0x0	0xe0f20080
0xf14 - vsi1	0x0	0x0	0x0	0x0
0xf28 - vsi2	0x0	0x0	0x0	0x0
0xf3c - vsi3	0x0	0x0	0x0	0x0
0xfa0 - vsi4	0x0	0x0	0x0	0x0
0xfb4 - vsi5	0x0	0x0	0x0	0x0
0xfc8 - vsi6	0x0	0x0	0x0	0x0
0xfdc - vsi7	0x0	0x0	0x0	0x0

change parameters (y/n)?

If the parameters of any window shall be changed, the next question must be answered with **y**, otherwise with **n**. If the parameters of a window shall be changed, a question for the number of the window to change occur. After selecting a window between **0** and **7**, each parameter can be set by giving a numeral value, except the flags. This value can be a decimal or a hexadecimal value. A hexadecimal value must start with **0x**. The flags can be edited directly by their meanings. It components are:

image enable	=> disable/enable
posted write enable	=> disable/enable
prefetch read enable	=> disable/enable
program AM code	=> dada/program/both
supervisor AM code	=> non-privileged/supervisor/both
VME address space	=> 16/24/32 Bit
64-bit PCI bus transactions	=> disable/enable
PCI bus lock of VMEbus RMW	=> disable/enable
PCI Bus Space	=> memory/IO space

3.6.3 The PCI Interrupt Registers

Figure 3—15: Content of the PCI Interrupt Registers

PCI interrupt registers

0x300 - enable	:	0x0
- enable own VME interrupt		( )
- enable VME interrupt level 1		( )
- enable VME interrupt level 2		( )
- enable VME interrupt level 3		( )
- enable VME interrupt level 4		( )
- enable VME interrupt level 5		( )
- enable VME interrupt level 6		( )
- enable VME interrupt level 7		( )
- enable DMA interrupt		( )
- enable LERR interrupt		( )
- enable VERR interrupt		( )
- enable SW_ACK interrupt		( )
- enable SW_INT interrupt		( )
- enable SYSFAIL interrupt		( )
- enable ACFAIL interrupt		( )
- enable mailbox interrupt 0		( )
- enable mailbox interrupt 1		( )
- enable mailbox interrupt 2		( )
- enable mailbox interrupt 3		( )
- enable local monitor interrupt 0		( )
- enable local monitor interrupt 1		( )
- enable local monitor interrupt 2		( )
- enable local monitor interrupt 3		( )
0x308 - map 0	:	0x77777777
0x30c - map 1	:	0x77770777
0x340 - map 2	:	0x77777777

change parameters (y/n)?

If the answer of the question is **y**, the parameters can be changed. The inputs of enable and map 0-2 are values, which can be given as a decimal or hexadecimal value. Hexadecimal values must start with a **0x**. To switch a flag, the **<SPACE>**-key must be pressed.

3.6.4 The Miscellaneous Registers

Figure 3—16: Content of the Miscellaneous Configuration Registers

```
miscellaneous configuration registers

0x400 - master control          :       0x80c00000
0x404 - miscellaneous control   :       0x12040000

change parameters (y/n)?
```

If the answer of the question is **y**, the parameters can be changed. The flags can be edited directly by their meanings. The components of the flags are:

3.6.4.1 The Components of the Master Control Flags

maximum number of retries	=> 64 960/retry forever
posted write transfer count	=> 128 4096/no BBSY
VMEbus request level	(3) => 0 3
VMEbus request mode	=> demand/fair
VMEbus release mode	=> release when done/on request
VMEbus ownership bit	=> release/acquire and hold
PCI aligned burst size	=> 32/64/128 Byte
PCI bus number	(0) => 0 255

3.6.4.2 The Components of the Miscellaneous Control Flags

VMEbus time-out	=> 16...1024usec/disabled
VMEbus arbitration mode	=> round robin/priority
VMEbus arbitration time-out	=> 16usec/256usec/disabled
Software PCI reset	=> release/acquire and hold
Software VMEbus SYSRESET	=> release/acquire and hold
universe is in BI mode	=> no/yes
enable global BI-mode initiator	=> release/acquire and hold
Universe is VMEbus system controller	=> no/yes
VME64 auto ID	=> release/acquire and hold

3.6.5 The PCI Miscellaneous Register

Figure 3—17: Content of the PCI Miscellaneous Registers

```
PCI miscellaneous registers

0x184 - PCI miscellaneous          :      0x10000000
0x188 - special PCI target image   :      0x0

change parameters (y/n)?
```

If the answer of the question is **y**, the parameters can be changed. The flags can be edited directly by their meanings. The components of the flags are:

3.6.5.1 The Components of the PCI Miscellaneous Flags

coupled window timer => 16...512 PCI clocks/disabled

3.6.5.2 The Components of the Special PCI Target Image Flags

image enable => disable/enable

posted write enable => disable/enable

VME maximum datawidth for each 16MByte region (0) => 0...15

Program AM code for each 16MByte region (0) => 0...15

supervisor AM code for each 16MByte region (0) => 0...1

base address for this image (0) => 0...63

PCI Bus Space => memory/IO/configuration space

4 Store and Select Different nvRAM Configurations

To store different nvRAM configurations, the actual nvRAM setting can be saved with the utility **genbootcnfg**. This utility generates a data module, that contains the current system informations, which are used to initialize the system. This module must be saved toat the host and merged to the file coreboot, behind the module **cnfgdata**. for example:

```
genbootcnfg -n=10 ;* generate the datamodule 'bootcnfg_a'
save bootcnfg_a  ;* store datamodule to disk
```

After transfering the file **bootcnfg_a** to the host (for example to the directory **<MWOS>/os9000/603/ports/bab740/cmds/bootobjs**), it must be included to the file **coreboot** (see figure 2-1).

If a datamodule shall be generated with the **genbootcnfg**-utility, a module with the same name is not be allowed to exist in the flash EEPROM. In the other case, the setting will not be saved to the datamodule.

5 Network Configuration

To initialize a network with a dynamic IP address via nvRAM or bootP, there must be changed the file netdb2, which will be generated from the configuration wizard. To change this, you must delete the IP related entries from the file **interfaces.conf**, generate an own **inetdb2** and replace the **inetdb2**, which are generated by the configuration wizard with your own file. To do this, you must perform several steps:

1. edit the file

   <port>/BOOTS/INSTALL/SPF/interfaces.conf
   and delete the IP related parts. For example:
   generated from configuration wizard:

   ```
   # Created by BAB740 Configuration Wizard program on 07.03.01
   hostname LocalHost
   enet0    address 0.0.0.0 broadcast 10.0.255.255 \
            netmask 255.255.000.000 binding /spde0/enet
   ```

   changed:

   ```
   # Created by BAB740 Configuration Wizard program on 07.03.01
   hostname LocalHost
   enet0    binding /spde0/enet
   ```

2. generate a new **inetdb2** with the **makefile**

   ```
   os9make makefile
   ```

3. rename the file **inetdb2** to **inetdb2.dynamic**

4. insert a new line within the **`makefile`** to replace the new **`inetdb2`** with **`inetdb2.dynamic`**

```
$(ODIR)/inetdb: $(SFILES)
$(CODO) $(ODIR)/inetdb
-$(DEL) $(ODIR)/inetdb
$(CODO) $(ODIR)/inetdb2
-$(DEL) $(ODIR)/inetdb2
$(IDBGEN) -to=$(OS) -tp=$(CPU) -d=. -d=$(SDIR) $(ODIR)/inetdb
copy inetdb2.dynamic inetdb2
$(ATTRE0) $(ODIR)/inetdb
$(ATTRE0) $(ODIR)/inetdb2
```

After this changes, the IP address will be set as follow:

system booted from	initial IP address come from
bootP server	respond bootP server
other device	entry from nvRAM

6 The Library nvramlib

To handle the interrupts of the universe chip, there are several functions in the library nvramlib. The usage of these functions are shown with the demo-utility si_universeint.

6.1 Universe_INIT()

Definition:

error_code universe_init()

This function initializes the universe with the nvRAM settings

6.2 Universe_IRQ_Enable()

Definition:

error_code universe_irq_enable(u_int32 pciInterrupts, u_int32 vmeInterrupts)

This function enables one or more PCI- or VME-interrupts.

6.3 Universe_IRQ_Disable()

Definition:

error_code universe_irq_disable(u_int32 pciInterrupts, u_int32 vmeInterrupts)

This function disables one or more PCI- or VME-interrupts.

7 Structure

4096 byte of the whole 8192 bytes can be used by the user, the other 4096 bytes are used for internals.

The intern used informations are separated with blocks, every block is checked with an CRC, expect the block for special parameters. Is the CRC of the data from the read block not equate to the CRC of the block, the parameters of this block will be reset to the manufacturer settings.

mirrored information , selected with hexswitch		
content of the block	**offset**	**size**
user data	0x0000	0x1000
reserved for future extensions	0x1000	0x0800
misc settings	0x1800	0x0120
boot settings	0x1920	0x0020
I/O settings	0x1940	0x0200
universe settings	0x1b40	0x0200
reserved	0x1d40	0x00b0
special boot settings	0x1df0	0x0200
reserved	0x1ff0	0x0010
extended system information		
mirrored contents of the EEPROM	0x2000	0x0400
netwerk related informations	0x2400	0x0200
checksum of the data area	0x2600	0x0004

7.1 Base Structures

7.1.1 Windows for the Universe

type	offset	description
unsigned int32	0x0	control register
unsigned int32	0x4	base address
unsigned int32	0x8	bound address
unsigned int32	0xc	translation offset

7.1.2 Serial Parameters

Type	offset	description
unsigned long	0x0	baud rate
unsigned char	0x4	bits per character
unsigned char	0x5	parity
unsigned char	0x6	number of stop bits
unsigned char	0x7	flow control
unsigned char	0x8	reserved for future extensions

7.1.3 IP Addresses

There are two layouts available:

- IPv4

type	offset	description
unsigned char	0x0	IP address
unsigned char	0x4	unused to fit to the IPv6 format

- IPv6

type	offset	description
unsigned short	0x0	IP address

7.1.4 Devices

There are two layouts available:

- **IDE**

Type	offset	description
unsigned int32	0x0	Port address
unsigned int16	0x4	Sector size
Int16	0x06	offset of logical sector 0
unsigned int8	0x08	logical unit number
unsigned int8	0x09	start index of partition
unsigned int8	0x0a	end index of partition

- **SCSI**

type	offset	Description
unsigned short	0x0	PORT address
Int16	0x04	offset of logical sector 0
unsigned int8	0x06	SCSI ID of the boot device
unsigned int8	0x07	SCSI ID of the SCSI controller
unsigned int8	0x08	logical unit number
unsigned int8	0x09	start index of partition
unsigned int8	0x0a	end index of partition
unsigned int8	0x0b	SCSI reset

7.2 The Misc Settings

Table 7—1: Description of the MiscType Structure

Type	offset	Description
unsigned long	0x000	checksum for the boot parameters
unsigned short	0x004	version of the boot parameters structure
unsigned short	0x006	revision of the boot parameters structure
unsigned char	0x008	8 bytes reserved for future extensions
Device	0x010	Parameters for boot via SCSI disk
Device	0x020	Parameters for boot via SCSI floppy
Device	0x030	Parameters for boot via IDE disk
Device	0x040	Parameters for boot via IDE floppy

7.3 The Boot Settings

Table 7—2: Description of the BootType Structure

Type	offset	description
unsigned long	0x000	checksum for the boot parameters
unsigned short	0x004	version of the boot parameters structure
unsigned short	0x006	revision of the boot parameters structure
unsigned long	0x008	Base address for RAM/ROM boot
unsigned long	0x00c	Delay until autoboot starts for ELTEC stuff
unsigned long	0x010	Delay until autoboot starts for MicroWare stuff
unsigned short	0x014	Retry counter for network boot
unsigned char	0x016	number of the boot device

7.4 The I/O Settings

Table 7—3: Description of the IOType Structure

Type	offset	description
unsigned long	0x000	checksum for the I/O
unsigned short	0x004	version of the I/O structure
unsigned short	0x006	revision of the I/O structure
unsigned char	0x0ce	speed for ethernet controler value: 0x01 = 10 MBit/s; 0x02 = 100 MBit/s; 0x03 = 1000 MBit/s
unsigned char	0x0cf	format of ethernet-transfer value: 0x01 = half duplex; 0x02 = full duplex
IP	0x0d0	ethernet IP address
IP	0x0f0	gateway IP address
IP	0x100	subnet mask

7.5 The Universe Settings

Table 7—4: Description of the UniverseType Structure

type	offset	description
unsigned long	0x000	checksum for the universe information
unsigned short	0x004	version of the universe structure
unsigned short	0x006	revision of the universe structure
UNIV_SI	0x008	PCI Slave windows 0 to 7
UNIV_SI	0x028	VME Slave windows 0 to 7
unsigned int32	0x048	pci_id;
unsigned int32	0x04c	pci_csr;
unsigned int32	0x050	pci_class;
unsigned int32	0x054	PCI Configuration Base Address
unsigned int32	0x058	PCI MISC0 Register
unsigned int32	0x05c	PCI MISC0 Register
unsigned int32	0x060	reserved for future extensions
unsigned int32	0x068	PCI Special Cycle Control Register
unsigned int32	0x06c	PCI Special Cycle Address Register
unsigned int32	0x070	scyc_en
unsigned int32	0x074	scyc_cmp
unsigned int32	0x078	scyc_swp
unsigned int32	0x07c	reserved for future extensions
unsigned int32	0x080	PCI Miscellaneous Register
unsigned int32	0x084	slsi
unsigned int32	0x088	DMA Transfer Control Register
unsigned int32	0x08c	DMA Transfer Byte Count Register
unsigned int32	0x090	dla
unsigned int32	0x094	dva
unsigned int32	0x098	dcpp
unsigned int32	0x09c	DMA General Control/Status Register
unsigned int32	0x0a0	DMA Linked List Update Enable Register
unsigned int32	0x0a4	reserved for future extensions
unsigned int32	0x0a8	PCI Interrupt Enable Register
unsigned int32	0x0ac	PCI Interrupt MAP Register 0
unsigned int32	0x0b0	PCI Interrupt MAP Register 1
unsigned int32	0x0b4	PCI Interrupt MAP Register 2
unsigned int32	0x0b8	VMEbus Interrupt Enable Register
unsigned int32	0x0bc	VMEbus Interrupt Map Register 0
unsigned int32	0x0c0	VMEbus Interrupt Map Register 1
unsigned int32	0x0c4	VMEbus Interrupt Map Register 2
unsigned int32	0x0c8	VMEbus Master Control Register
unsigned int32	0x0cc	Miscellaneous Control Register
unsigned int32	0x0d0	user defined address modifier
unsigned int32	0x0d4	reserved for future extensions
unsigned int32	0x0d8	Location Monitor Control Register
unsigned int32	0x0dc	lm_bs
unsigned int32	0x0e0	VMEbus Register Access Image Control Register
unsigned int32	0x0e4	vrai_bs
unsigned int32	0x0e8	VMEbus CSR Control Register
unsigned int32	0x0ec	vcsr_to
unsigned int32	0x0f0	VMEbus CSR Bit Clear Register
unsigned int32	0x0f4	VMEbus CSR Bit Set Register
unsigned int32	0x0f8	VMEbus CSR Bit Clear Register
unsigned int32	0x0fc	reserved for future extensions

7.6 The Special Boot Settings

Table 7—5: Description of the extendedParamsType Structure

type	offset	description
char	0x03c	ethernet internet addr
char	0x05a	backplane internet addr
char	0x096	gateway internet addr
int	0x1d0	configuration flags

7.7 Mirrored Contents of the EEProm

Table 7—6: Description of the EEPromType Structure

type	offset	description
char	0x000	Magic number
char	0x008	Revision of structure
unsigned short	0x00a	Size of CRC area
unsigned long	0x00c	CRC
char	0x010	Board Revision information
char	0x020	Option Revision information
char	0x060	Board serial number
char	0x068	Ethernet node addresse
char	0x070	Revision codes
char	0x07e	Category codes
char	0x080	Text field

7.8 Network Related Information

Table 7—7: Description of the NetworkType Structure

type	offset	description
unsigned char	0x000	Boot file address/name
unsigned char	0x080	name of server
IPType	0x0c0	IPv4/6 address of Host
IPType	0x0d0	IPv4/6 address of SubMask
IPType	0x0e0	IPv4/6 address of Ethernet
IPType	0x0f0	IPv4/6 address of Backplane net
IPType	0x100	IPv4/6 address of Gateway
IPType	0x110	IPv4/6 address of Broadcast
unsigned int32	0x120	number of the module, whitch has bootet via auto boot
unsigned short	0x124	Network boot timeout value
unsigned char	0x126	flag for autoboot active