Integrator/PP1 and PP2

Getting Started Guide

ARM

ARM DUI 0169A

Integrator/PP1 and PP2
Getting Started Guide

Release Information

Date	Issue	Change
July 2002	A	First release

Proprietary Notice

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

This device is test equipment and consequently is exempt from part 15 of the FCC Rules under section 15.103 (c).

CE Declaration of Conformity

CE

The system should be powered down when not in use.

The Integrator generates, uses, and can radiate radio frequency energy and may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment causes harmful interference to radio or television reception, which can be determined by turning the equipment off or on, you are encouraged to try to correct the interference by one or more of the following measures:

- ensure attached cables do not lie across the card
- reorient the receiving antenna
- increase the distance between the equipment and the receiver
- connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- consult the dealer or an experienced radio/TV technician for help

Note

It is recommended that wherever possible Shielded interface cables be used.

Contents
Integrator/PP1 and PP2 Getting Started Guide

Preface

This preface introduces the Integrator/PP1 and PP2 and their user documentation. It contains the following sections:

About this book

This book provides getting started information for the Integrator/PP1 and PP2 porting platforms. It describes the systems and how to set them up ready for use.

Intended audience

This book is written for all developers who are using the ARM Integrator/PP1 and PP2.

Using this book

This book is organized into the following chapters:

Chapter 1 ***Unpacking and setting up***

Read this chapter for a description of the system components, and how to prepare the system for use.

Chapter 2 ***Architecture and Module Overview***

Read this chapter for an overview of the system architecture and the Integrator modules used in the system.

Typographical conventions

The following typographical conventions are used in this book:

italic	Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.
bold	Highlights interface elements, such as menu names. Denotes ARM processor signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that can be entered at the keyboard, such as commands, file and program names, and source code.
`mono`space	Denotes a permitted abbreviation for a command or option. The underlined text can be entered instead of the full command or option name.
`monospace italic`	Denotes arguments to commands and functions where the argument is to be replaced by a specific value.
`monospace bold`	Denotes language keywords when used outside example code.

Further reading

This section lists publications that provide additional information on the Integrator boards used in the system and tools for developing code for the ARM family of processors.

ARM periodically provides updates and corrections to its documentation. See `http://www.arm.com` for current errata sheets and addenda.

See also the ARM Frequently Asked Questions list at:
`http://www.arm.com/DevSupp/Sales+Support/faq.html`

ARM publications

The following documents provide information about related Integrator products:

- *ARM Integrator/AP User Guide* (ARM DUI 0098)
- *ARM Integrator/CM920T-ETM User Guide* (ARM DUI 0149)
- *ARM Integrator/CM9x0T and CM7x0T User Guide* (ARM DUI 0157)
- *ARM Integrator/LM-XCV600E+ LM-EP20K600E+ User Guide* (ARM DUI 0146).
- *ARM Integrator/IM-PD1 User Guide* (ARM DUI 0152).

The following publications provide reference information about ARM architecture:

- *AMBA Specification* (ARM IHI 0011)
- *ARM Architectural Reference Manual* (ARM DDI 0100)

The following publication provide information about ARM PrimeCell devices used to control the interfaces described in this manual:

- *ARM PrimeCell UART (PL011) Technical Reference Manual* (ARM DDI 0183*)*
- *ARM PrimeCell Synchronous Serial Port Master and Slave (PL022) Technical Reference Manual* (ARM DDI 0171)
- ARM PrimeCell *RTC (PL030) Technical Reference Manual* (ARM DDI 0140)
- ARM PrimeCell *KMI (PL050) Technical Reference Manual* (ARM DDI 0143).
- *ARM PrimeCell Advanced Audio CODEC Interface (PL041) Technical Reference Manual* (ARM DDI 0173).
- *ARM PrimeCell GPIO (PL061) Technical Reference Manual* (ARM DDI 0187)
- *ARM PrimeCell Color LCD Controller (PL110) Technical Reference Manual* (ARM DDI 0161)
- *ARM PrimeCell Smartcard Interface (PL130) Technical Reference Manual* (ARM DDI 0148)
- *ARM PrimeCell Vectored Interrupt Controller (PL190) Technical Reference Manual* (ARM DDI 0181)

- *ARM PrimeCell Multimedia Card Interface (PL181) Technical Reference Manual* (ARM DDI 0205).

The following publication provides information about the ARM Firmware Suite:

- *ARM Firmware Suite Reference Guide* (ARM DUI 0102)

The following publications provide information about ARM SDT 2.5:

- *ARM Software Development Toolkit User Guide* (ARM DUI 0040)
- *ARM Software Development Toolkit Reference Guide* (ARM DUI 0041).

The following publications provide information about the ARM Developer Suite:

- *Getting Started* (ARM DUI 0064)
- *ADS Tools Guide* (ARM DUI 0067)
- *ADS Debuggers Guide* (ARM DUI 0066)
- *ADS Debug Target Guide* (ARM DUI 0058)
- *ADS Developer Guide* (ARM DUI 0056)
- *ADS CodeWarrior IDE Guid*e (ARM DUI 0065).

The following publication provide information about the Multi-ICE:

- *Multi-ICE User Guide* (ARM DUI 0048).

Support and feedback

ARM Limited welcomes feedback on both the Integrator/PP1 and PP2, and its documentation.

Support for the Integrator/PP1 and PP2

If you have any problems with these systems, please contact your supplier. To help us provide you with a rapid response, please give:

- details of the release you are using
- details of the hardware platform, operating system type and version
- a small standalone sample of code that reproduces the problem
- a clear explanation of what you expected to happen, and what actually happened
- the commands you used, including any command-line options
- sample output illustrating the problem
- the version number and date of any tools.

Support for the operating system

If you have problems with the operating system, refer to the operating system vendor.

Feedback on this book

If you have any comments on this book, please send email to `errata@arm.com` giving:

- the document title
- the document number
- the page number(s) to which your comments apply
- a concise explanation of your comments.

General suggestions for additions and improvements are also welcome.

Chapter 1
Unpacking and setting up

The Integrator/PP1 and PP2 porting platforms provide you with a reference platform that allows you to develop applications on a system that closely models your end product. The systems are based on standard Integrator modules. This chapter introduces the Integrator/PP1 and PP2. It contains the following sections:

1.1 Unpacking the Integrator/PP1

The Integrator/PP1 product package comprises the following:

- System unit enclosed in an ATX-style case that contains:
  - Integrator/AP motherboard
  - Integrator/CM920T-ETM or CM920T core module with SDRAM DIMM
  - Imagination Neon-250 PCI graphics card
  - Intel Pro/100+ management adapter PCI Ethernet card or equivalent.
- PS2 keyboard and mouse
- *ARM Firmware Suite* (AFS) Version 1.4 or later with user guide
- *ARM Developer Suite* (ADS) Evaluation Version.
- Integrator user guides:
  - Integrator/AP User Guide
  - Integrator/CM920T-ETM or Integrator/CM9x0T and CM7x0T User Guide

Figure 1-1 shows the PP1 product package.

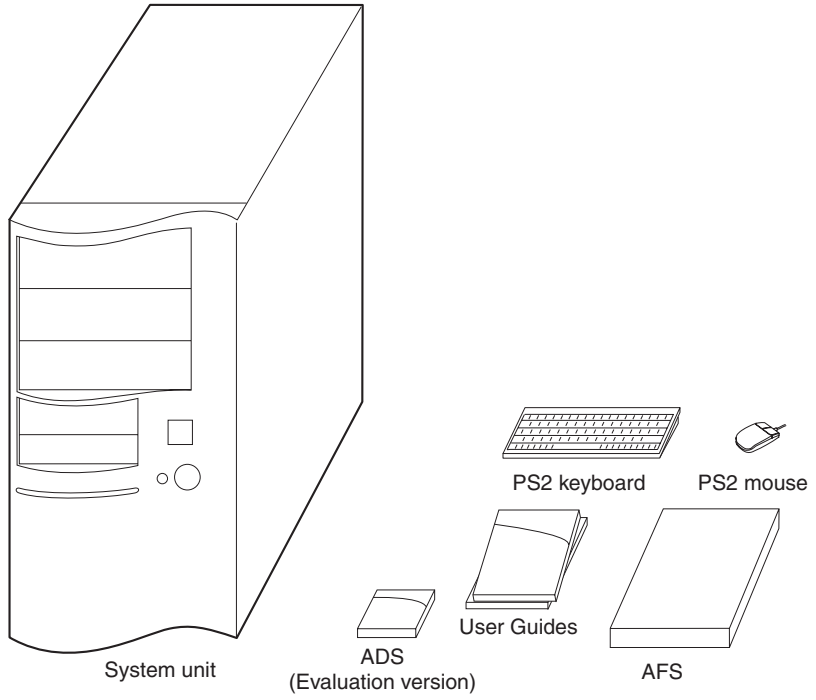

Figure 1-1 Integrator/PP1 product package

1.2 Unpacking the Integrator/PP2

As shown in Figure 1-2, the Integrator/PP2 product package comprises the following:

- System unit in a custom case that houses:
  - Integrator/AP motherboard
  - Integrator/CM920T-ETM or CM920T core module with SDRAM DIMM
  - Integrator/LM-XCVC600E+ logic module
  - Integrator/IM-PD1 interface module
  - Intel Pro/100+ management adapter PCI Ethernet card or equivalent.
  - PSU
  - LCD display.
- PS2 keyboard and mouse
- *ARM Firmware Suite* (AFS) Version 1.4 or later with user guide
- *ARM Developer Suite* (ADS) Evaluation Version.
- Integrator user guides:
  - Integrator/AP User Guide
  - Integrator/CM920T-ETM or Integrator/CM9x0T and CM7x0T User Guide
  - Integrator/LM-XCV600E+ User Guide
  - Integrator/IM-PD1 User Guide.

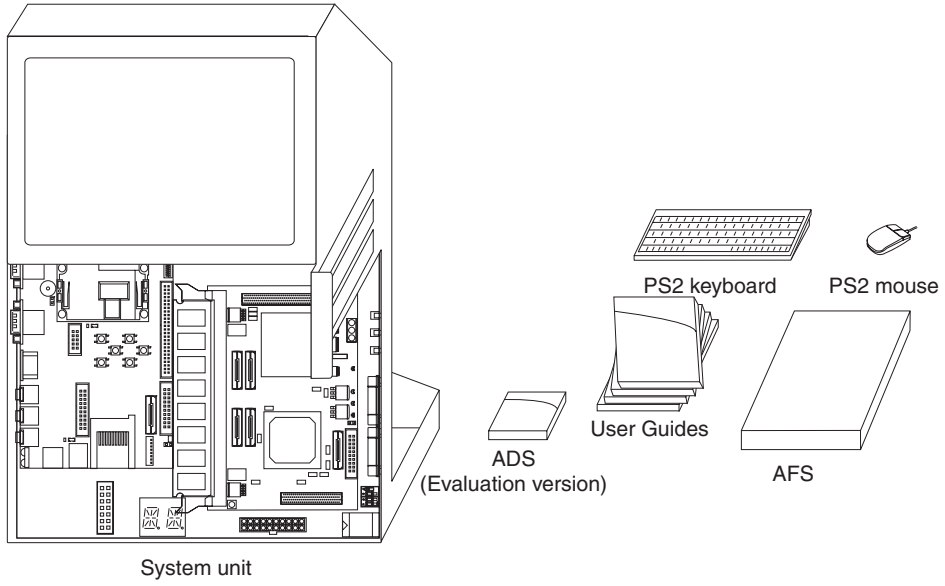

Figure 1-2 Integrator/PP2 product package

1.3 Setting up the Integrator/PP1

Figure 1-3 shows the locations of the connectors on the rear of the system unit.

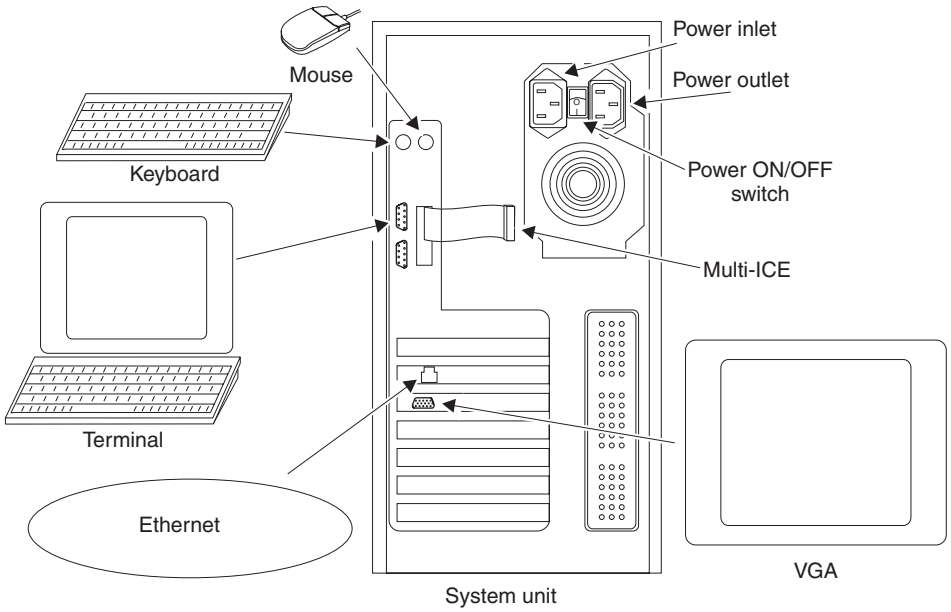

Figure 1-3 Integrator/PP1 system connections

Prepare the system for use as follows:

1. Connect the system components as shown in Figure 1-3:
   - the keyboard and mouse to the mini-DIN connectors
   - a terminal or PC running a terminal emulator to the 9-pin D-type connector using the null-modem serial cable supplied.
   - a display to the 15-pin D-type VGA connector shown
   - your Multi-ICE unit, if required to the trailing connector.
2. Connect the system to your LAN using the RJ45 connector.
3. Connect power and operate the power switch on the rear of the system unit.

   The system powers-up and the *Porting Platform Flash Utility* (PPFU) starts. You now need to load your operating system as described in *Loading and starting an operating system* on page 1-8.

1.4 Setting up the Integrator/PP2

Figure 1-4 shows the locations of the connectors and switches on the left side of the system unit.

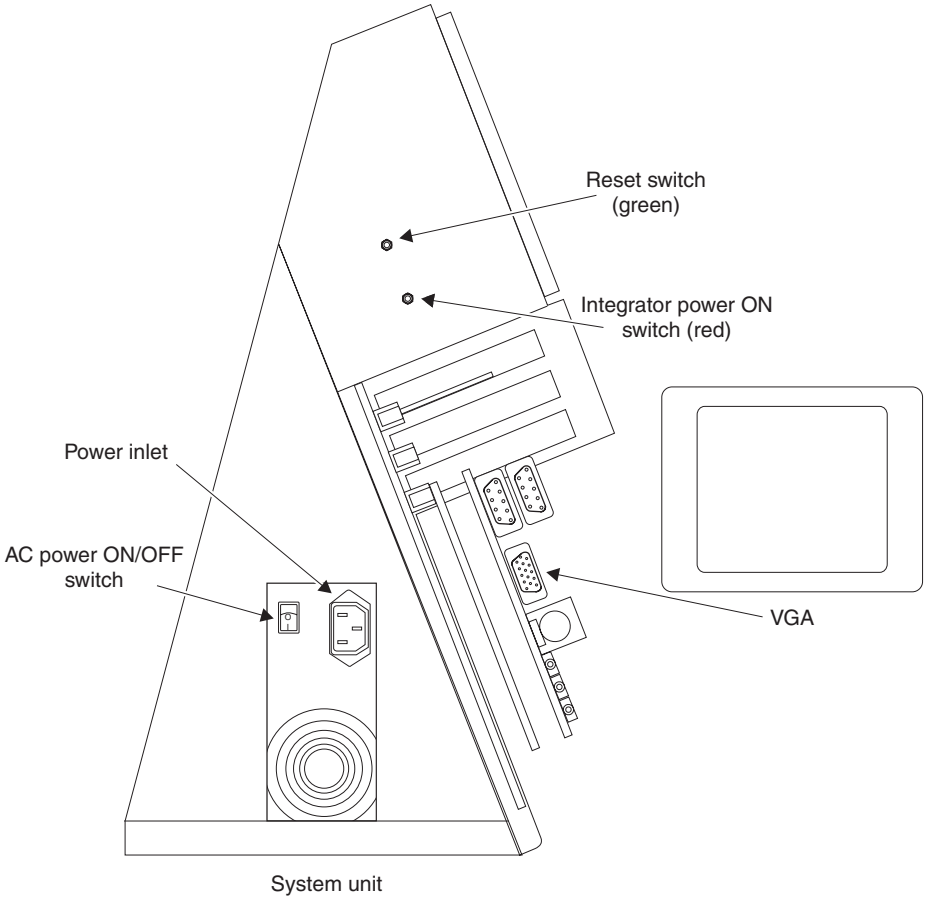

Figure 1-4 Integrator/PP2 connectors and switches (left)

Figure 1-5 on page 1-6 shows the locations of the connectors and switches on the right side of the system unit.

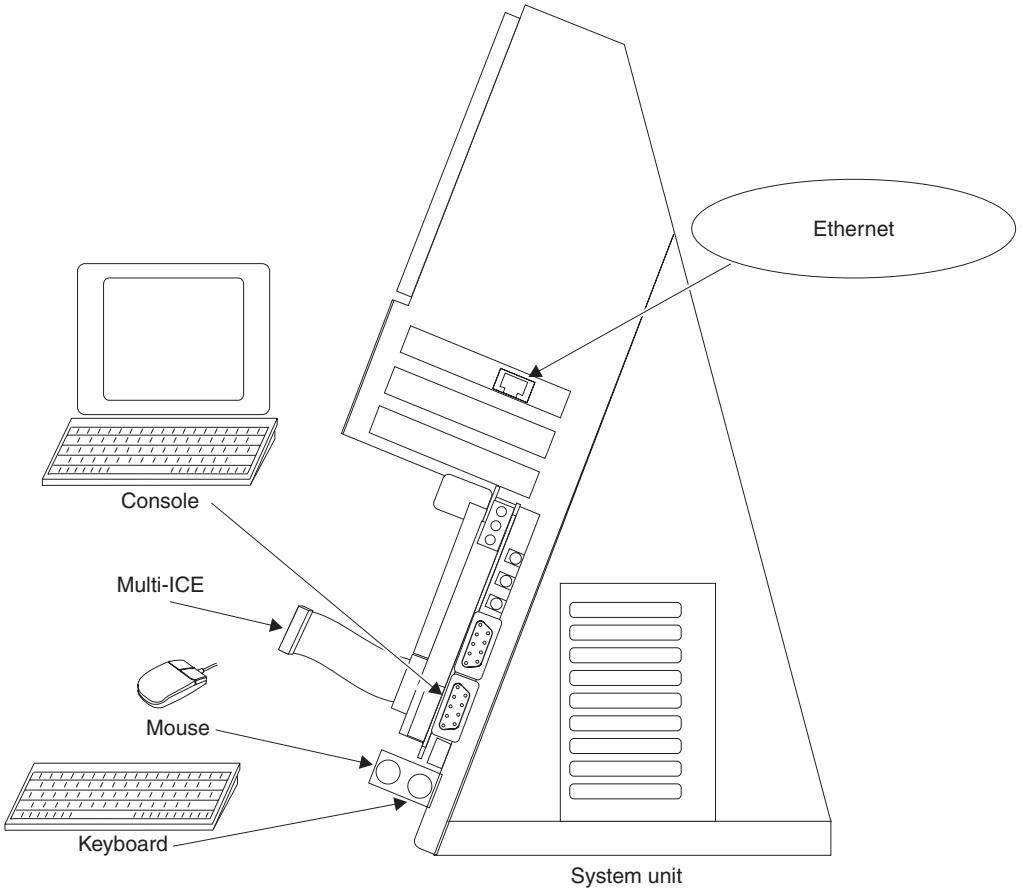

Figure 1-5 Integrator/PP2 connectors (right)

Prepare the system for use as follows:

1. Connect the system components as shown in Figure 1-4 on page 1-5 and Figure 1-5:
   - The keyboard and mouse to the mini-DIN connectors on the AP.
   - A terminal, or PC running a terminal emulator, to the 9-pin D-type connector on the AP using the using null-modem serial cable supplied.
   - Your Multi-ICE unit, if required, to the Multi-ICE connector on the core module, as shown in Figure 1-5.

- A VGA display, if required, to the VGA connector on the IM-PD1. To enable a VGA display output, use the LM_CONTROL register to select the correct display type (see the *Integrator/IM-PD1 User Guide*).

2. Connect the system to your LAN using the RJ45 connector.

3. Connect power to the power connector on the PSU, operate the AC power switch on the PSU and then the Integrator power ON switch on the left side of the system unit, as shown in Figure 1-4 on page 1-5.

   The system powers-up and the *Porting Platform Flash Utility* (PPFU) starts. You now need to load your operating system as described in *Loading and starting an operating system* on page 1-8.

1.5 Loading and starting an operating system

The porting platform is configured at the factory so that the boot monitor runs the *Porting Platform Flash Utility* (PPFU). This section describes how to load an image into flash using PPFU and then how to boot the image. The image could be an operating system or other program.

The PPFU commands are described in *About PPFU* on page 1-11.

1.5.1 Prerequisites

To load an image with PPFU you must:

- Connect the porting platform to a TFTP server using a LAN connection. Software is available from third-party vendors over the Internet to enable you to use a PC as a TFTP server.
- Connect a serial terminal to the Integrator/AP, as shown in Figure 1-3 on page 1-4 or Figure 1-4 on page 1-5. The terminal requires the following settings:
  - 38400 baud
  - no parity
  - 8 bits data
  - 1 stop bit
  - Xon/Xoff software handshaking.
- Ensure that the DIP switches on the Integrator/AP set as follows:
  - S1[1] = ON, boot switcher selects boot image according to S1-4.
  - S1[2] = don’t care.
  - S1[3] = don’t care.
  - S1[4] = OFF, boot the selected boot image from flash.

  These settings select the PPFU as the boot image for the porting platform.

1.5.2 Loading an image into flash

Load an image into flash as follows:

1. Power up your porting platform.

   The system boots the PPFU and displays start-up messages on the terminal. For example:

   ```
   Copyright ARM Limited 2002. All rights reserved.
   NexGenOS v1.3, NexGenIP v1.3, Copyright (c) 2001 WWW.NexGen-Software.fr
   Local IP address:  0.0.0.0
   Ticks per second:  1000
   Type help to get the list of commands.
   ```

   To display the syntax of a particular command, type `help` *`command`*.

   If your porting platform does not boot the PPFU, see *Selecting PPFU as the default boot image using the boot monitor* on page 1-11.

2. Set the IP address of the porting platform using one of the following:
   - If you have a DHCP server available, then:
     1. Enter the command `dhcpc start`. This requests an IP address from the DHCP server.
     2. Check the IP address that you have been assigned using the command `ifconfig`.
   - If you do not have a DHCP server, then set the IP address and netmask for the porting platform using the command `ifconfig` (see *ifconfig* on page 1-13).

   Consult your system administrator if you need help with any of these steps.

3. List the images in flash using the command `list`.

4. Download the image to the porting platform using the command `tftpread` *`filename IP_address size`*, where:

`filename`	is the file name to read (download) from the server.
`IP address`	is the IP address of the TFTP server.
`size`	is the size of the file to read.

5. Program the image into flash with the command: `program` *`imageNo image_name`*, where:

`imageNo`	is the number that you allocate to the image.
image_name	is the name of the image.

1.5.3 Setting a new default boot image

Set a new default boot image as follows:

1. At the PPFU prompt list the images in flash by entering the `list` command. For example:

```
>list
Listing images in Flash
Image   667 Blocks [   1,    1] address 0x24020000 exec 0x500000    - name milo
Image    66 Blocks [   2,    3] address 0x24040000 exec 0x2e6e4     - name PPFU
Image   666 Blocks [   4,   19] address 0x24080000 exec 0x24080000  - name kernel
Image   777 Blocks [  50,   75] address 0x24640000 exec 0x24640000  - name cramfs
Image    11 Blocks [  76,  167] address 0x24980000 exec 0x24980000  - name OS
  SIB    at   Block 255 End Block 255 address 0x25fe0000
```

2. Set the new boot image using the `setbootimage` command. In this example, the operating system is image number 11, so that the command to set this as the boot image is:

```
>setbootimage 11 0xfff00 0x100000
setting image 11 in SIB,
copy at 0xfff00 and run it from 0x100000
```

   A *System Image Block* (SIB) is created for this image.

3. Check this by using the list command again. For example:

```
>list
Listing images in Flash
Image    667 Blocks [   1,    1] address 0x24020000 exec 0x500000    - name milo
Image     66 Blocks [   2,    3] address 0x24040000 exec 0x2e6e4     - name PPFU
Image    666 Blocks [   4,   19] address 0x24080000 exec 0x24080000  - name kernel
Image    777 Blocks [  50,   75] address 0x24640000 exec 0x24640000  - name cramfs
Image     11 Blocks [  76,  167] address 0x24980000 exec 0x24980000  - name OS
  SIB   at   Block 254 End Block 254 address 0x25fc0000
  SIB   at   Block 255 End Block 255 address 0x25fe0000
```

   In this example, a new SIB has been created at block 254.

4. Reset the system.

   After a short delay, the porting platform boots the selected image.

   To escape from this boot sequence, press any key during the delay period and the PPFU image is booted instead.

1.6 About PPFU

The *Porting Platform Flash Utility* (PPFU) provides commands that you can use to:

- establish a network connection to a server
- download an image, program an image into flash, and select an image to boot.

1.6.1 Booting the PPFU

After power up or reset, the porting platform can boot the standard boot monitor, PPFU, or any other selected image. Boot image selection is controlled by the setting of the DIP switches.

If the boot monitor or another image is booted instead of the PPFU when you power up:

- check that the switches on the motherboard a re set with S1[1] to ON and S1[4] to OFF and reset the system.
- if the switch settings are correct select PPFU as the boot image as described in *Selecting PPFU as the default boot image using the boot monitor*.

For information about the DIP switches on the Integrator/AP board, see the *Integrator/AP User Guide*

1.6.2 Selecting PPFU as the default boot image using the boot monitor

Select PPFU as the default boot image as follows:

1. Check the contents of the flash using the v command. For example:

```
boot Monitor> v

There are 256 128KByte blocks of Application Flash:
Images found
============
Block  Size   ImageNo  Name      Compress
-----  ----   -------  ----      --------
    0     1         1  zygote    (0x24000000-0x2401FFEC)
    1     2        66    PPFU    (0x24020000-0x2405FFEC)
   32    88         2      OS    (0x24400000-0x24EFFFEC)

System Information Blocks
=========================
Address      Owner                   Size  Idx  Rev
-------      -----                   ----  ---  ---
0x25FE0000   ARM Boot Monitor         312    0    4
```

   In this example, the image named `PPFU` is listed as `ImageNo` 66.

2. Change the default boot image using the `bi` command. For example:

```
boot Monitor> bi 66
```

3. Reset the system. The following is displayed:

```
Copyright ARM Limited 2002.  All rights reserved.
NexGenOS v1.3, NexGenIP v1.3, Copyright (c) 2001 WWW.NexGen-Software.fr

Local IP address:  0.0.0.0
Ticks per second:  1000
Type help to get the list of commands.
```

1.6.3 PPFU command description

This section provides a description of the PPFU commands. To list the available commands, use the command `help`. For example:

```
>help
Supported commands:
help          ver          netstat          ifconfig
arp           route        ping             dhcpc
lsmod         tftpread     tftpsizeof       list
identify      testBlock    delete           deleteAll
deleteBlock   program      setbootimage
>
```

To display the syntax for a command, use the command `help` *command*.

ver

Displays the TCP/IP stack version

netstat

```
netstat [-a|m]
```

Displays network active connections. Where:

`-a` Display all information
`-m` Display multicast information
`-o` Display timers
`-i` Display interface table
`-r` Display routing table
`-s` Display statistics
`-b` Display buffers usage

ifconfig

`ifconfig [ifname [address|options]]`

Configure network interfaces where:

ifname is the name of the hardware Ethernet interface. By default this is eth0.

address is the IP address of the porting platform.

The options are:

`netmask mask` is the netmask.

`dstaddr addr` is the destination IP address.

`mtu n` is the maximum transfer unit.

`up` activates the interface

`down` shut down the interface.

The `ifconfig` command is used to assign a static IP address (if there is no DHCP server, for example). It can also be used also view the network interface setting. For example:

```
> ifconfig
Network Interfaces
eth0     Ethernet     HWaddr:00D0B7-A0A112
Addr:172.16.11.90 SubNet:172.16.0.0 Bcast:172.16.255.255
 UP BROADCAST RUNNING MULTICAST MTU:1500
 RX Pkts:37399 Mcast:37399 Bytes:2419164 Errs:0 Drops:1522 NoProto:1422
 TX Pkts:2 Mcast:0 Bytes:618 Errs:0 Drops:0
 Driver:i82559intAP Irq:14 IOBase:0x4800 Mem:0x44080000
 Multicast Filter
:
```

arp

`arp [-a] [-d hostname] [-s hostname_hw_addr]`

Displays the *Address Resolution Protocol* (ARP) host table. Where:

`-a` displays the host table

`-d hostname` deletes an entry

`-s hostname` **hw_addrs**

adds a static entry

route

`route [add [-net|-host] target [netmask_Nm] gateway] [del target]`

This command displays the routing table where:

`add`	adds a static route.
`del`	deletes a static route.
target	defines the target address. Specifies the default route if *netmask_Nm* and *gateway* are omitted.
netmask_Nm	is the netmask for the target network address.
gateway	specifies the gateway address. Must be reachable on a local network.

ping

`ping addr`

Send ICMP ECHO_REQUEST packets to a network host with the following argument:

addr	Specifies the IP address of the destination host.

For example:

```
> ping 192.16.100.93
32 bytes from 192.16.100.93: icmp_seq=0 ttl=128 time=4 ms
32 bytes from 192.16.100.93: icmp_seq=1 ttl=128 time=3 ms
32 bytes from 192.16.100.93: icmp_seq=2 ttl=128 time=3 ms
32 bytes from 192.16.100.93: icmp_seq=3 ttl=128 time=3 ms
>
```

dhcpc

`dhcpc [(start|release) [ifname]][inform [ifname IP_addr][sizeof]`

Use this command to:

- manage a DHCP client on a specified interface (defaults to first DHCP client)
- obtain an IP address from a DHCP server if one is accessible on the network.

Where:

`start`	starts the DHCP client and obtains an IP address from the DHCP server.
`release`	stops the DHCP client and allows the DHCP server to reallocate the IP address.
ifname	Network interface name, for example, eth0.
IP_Addr	IP address of DHCP server to send DHCPINFORM to

lsmod

```
lsmod
```

Display a list of modules present. For example:

```
Name          Type          Version     Date          Mount point
uHal Poll     OS Layer      1.30A       4/03/2001
i82559intA    Network
IP            Protocol
RAWIP         Protocol
UDP           Protocol
ARP           Protocol
DHCP          Protocol
```

tftpread

`tftpread` *`filename IP_address size`*

Use this command to retrieve a file from a specific machine running a TFTP server. The value of the size parameter is higher or equal to the size of the file to download. The `tftpread` command returns when the file has been downloaded and the file is present in the target memory.

The arguments are:

`filename`	File name to read (download) from server.
`IP_address`	The IP address of the TFTP server.
`size`	Size of the file to read.

tftpsizeof

```
tftpsizeof
```

Displays the size of NexGenBOOT TFTP internal structures.

list

`list`

Use this command to list the images in flash.

For example:

```
>list
Listing images in Flash
Image   667 Blocks [ 1,   1] address 0x24020000 exec 0x500000   - name milo
Image    66 Blocks [ 2,   3] address 0x24040000 exec 0x2ee18    - name loader
Image   666 Blocks [30,  44] address 0x243c0000 exec 0x243c0000 - name kernel
Image   777 Blocks [50,  75] address 0x24640000 exec 0x24640000 - name cramfs
SIB   at   Block 255 End Block 255 address 0x25fe0000
```

identify

`identify`

Us this command to identify flash type.

For example:

```
> identify
Current Active device is:
Flash (Intel 28F320S3) at address 0x24000000 : size 0x2000000
```

testBlock

`testBlock B`*`block_number`*

Use this command to write a test pattern to a particular flash block. You cannot write a test pattern to the boot SIB.

delete

`delete` *`image number`*

Use this command to delete an image in Flash. The logical image number is passed as argument.

deleteAll

`deleteAll`

Use this command to delete all blocks in flash.

deleteBlock

`deleteBlock B`*`block_number`*

Use this command to delete a block that appears not to be in an image.

program

`program` *`image_number image_name`* `[`*`address`* `|or| B`*`block_no`*`] [[noboot]|[z]]`

Use this command to program the an image into flash at by specifying an address `0x`*`hex_addr`* or block number.

This is used to program the image loaded in memory via tftpread into flash. For example:

```
> program 11 testImage
```

setbootimage

`setbootimage` *`image_number copy_address run_address`*

Use this command to create and write a *System Information Block* (SIB) at the end of the flash to store information for a binary image to boot (for example, an image that could not be loaded by boot monitor). The parameters are:

`image_number` The logical number of the image.

`copy_address` The address to where the image is to be copied in RAM.

`start_address` The address to where the image is to be started in RAM.

The SIB containing this information can be removed using the `deleteBlock` command, specifying the block number corresponding to that SIB.

Note

If, by error, the wrong SIB is removed (for example, the one used by boot monitor containing the default boot image), it can be recreated using `bi` command at the boot monitor prompt to set the default image to boot. See *Setting a new default boot image* on page 1-10.

Chapter 2
Architecture and Module Overview

This chapter describes how to set up and start using the PP1 and PP2. It contains the following sections.

2.1 System architecture

The porting platform design uses a modular architecture that enables you to easily configure the system for the requirements of your project. The components of the system architecture are shared between the Integrator modules. Figure 2-1 shows a block diagram of the Integrator/PP1.

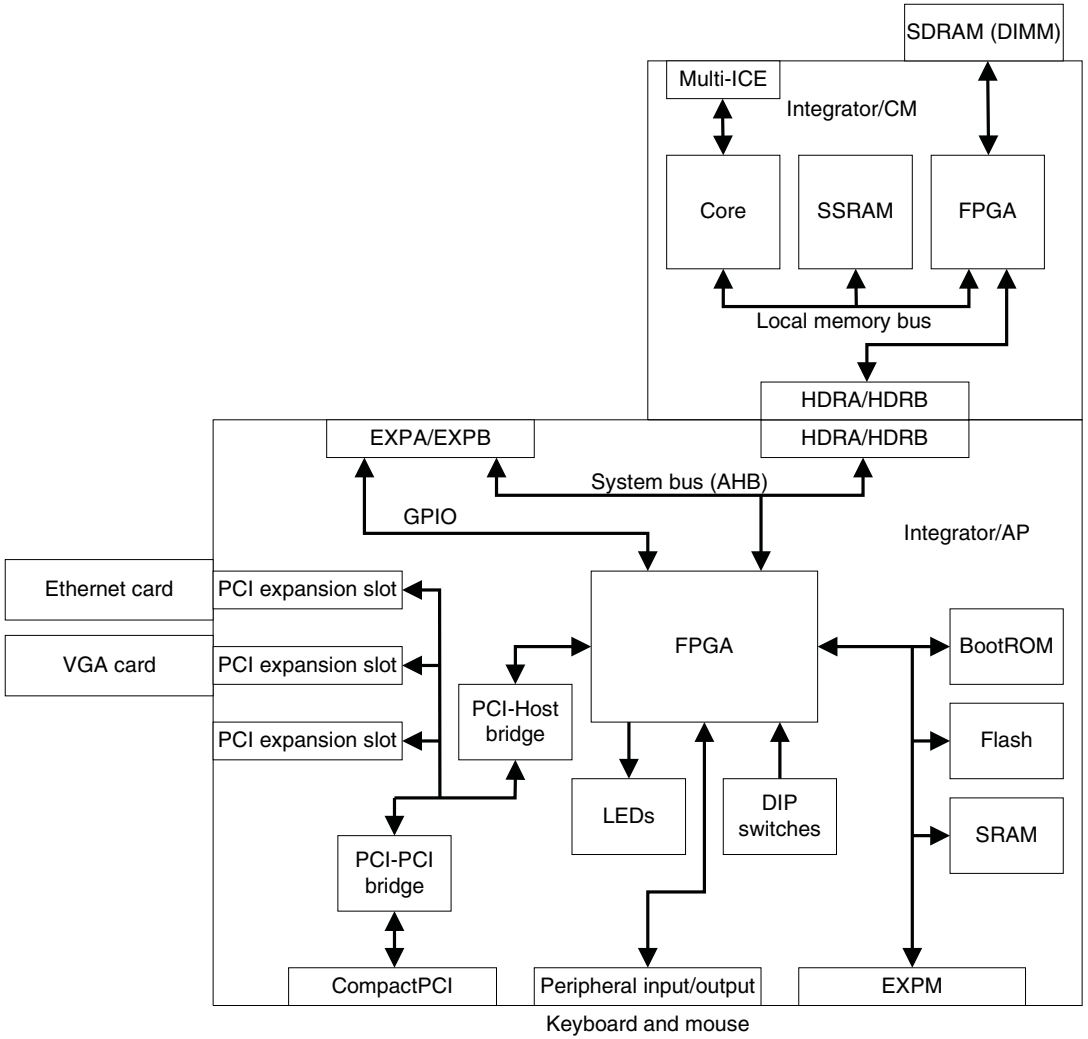

Figure 2-1 Integrator/PP1 architecture

Figure 2-2 shows a block diagram of the Integrator/PP2.

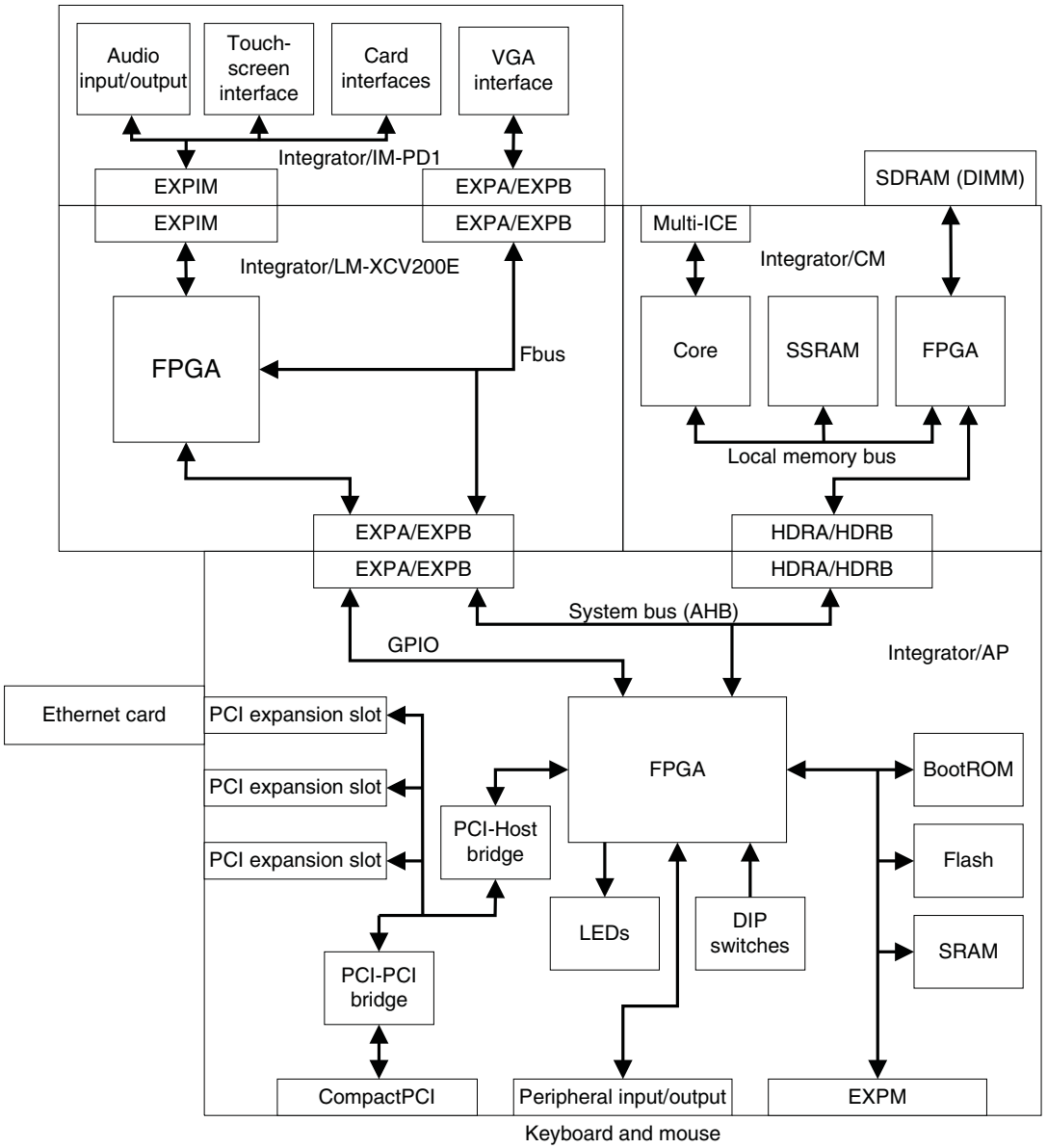

Figure 2-2 integrator/PP2 architecture

2.2 About the Integrator modules

The main features of the Integrator modules are described in:

- *Integrator/AP*
- *Integrator/CM920T-ETM or CM920T Core module* on page 2-5
- *Integrator/LM-XCV600E+ Logic module* on page 2-5
- *Integrator/IM-PD1 Interface module* on page 2-6.

2.2.1 Integrator/AP

The major features of the Integrator/AP are as follows:

- system controller *Field Programmable Gate Array* (FPGA) that implements:
  - system bus interface to the core and logic modules
  - system bus arbiter
  - interrupt controller
  - peripheral input and output controllers
  - three counter/timers
  - reset controller
  - system status and control registers
- clock generator
- 32MB flash memory
- 256KB boot ROM
- 512KB SSRAM
- two serial ports (RS232 DTE)
- expandable system architecture supporting additional core and logic modules (up to 5 in total)
- PCI bus interface
- *External Bus Interface* (EBI), supporting memory expansion.

2.2.2 Integrator/CM920T-ETM or CM920T Core module

The major features on the core module are as follows:

- ARM920T microprocessor core
- core module FPGA that implements:
  - SDRAM controller
  - system bus bridge
  - reset controller
  - interrupt controller
  - status, configuration, and interrupt registers.
- 1MB SSRAM.
- up to 256MB of SDRAM plugged into the DIMM socket
- SSRAM controller
- clock generator
- system bus connectors
- logic analyzer connectors for AHB and Trace port.

2.2.3 Integrator/LM-XCV600E+ Logic module

The main features of the logic module are as follows:

- Xilinx VirtexE XCV2000E FPGA that implements:
  - display interface, PrimeCell CLCD controller (PL110)
  - touchscreen interface, PrimeCell SSP (PL022)
  - *Smart Card Interface* (SCI), PrimeCell SCI (PL130)
  - serial interfaces, PrimeCell UART (PL011)
  - audio interface, PrimeCell AACI (PL041)
  - MMC interface, PrimeCell MMCI (PL181).
- configuration PLD and flash memory for storing FPGA configurations
- 1MB ZBT SSRAM
- clock generators and reset sources
- switches
- user LEDs
- prototyping grid
- JTAG, Trace, and logic analyzer connectors
- system bus connectors to a motherboard or other modules.

2.2.4 Integrator/IM-PD1 Interface module

The main features of the interface module are as follows:

- display support:
  - interface to 8.4 inch Sharp color full VGA LCD
  - video DAC to support the connection of a VGA or SVGA PC monitor.
- audio CODEC
- *MultiMedia Card* (MMC) interface
- smartcard socket
- two serial RS232 transceivers
- IrDA transceiver
- six push buttons
- buzzer.

