

Versatile/PB926EJ-S FAQ v1.0

Table of contents

1. General Advice and Latest News	4
1.1 Upgrade to Versatile/PB926EJ-S v1.1 now!	4
1.2 Upgrades, bug fixes and frequently asked questions	4
1.3 Help to design AHB masters and slaves in logic tiles	4
1.4 Help to use the DMA controller and the AACI audio controller.....	4
2. Introduction to Versatile/PB926EJ-S	6
2.1 What is Multi-layer AHB? What is a bus matrix?	6
2.2 What is the ARM926EJ-S PrimeXSys Platform (PXP)? What is the ARM926EJ-S development chip?.....	6
2.3 What are the contents of the Versatile/PB926EJ-S installation CD?	6
2.4 What software and tools do I need in order to use Versatile/PB926EJ-S?	7
2.5 How much does Versatile/PB926EJ-S cost?.....	8
2.6 Where can I find the Versatile/PB926EJ-S schematics and user guide?.....	8
3. Versatile/PB926EJ-S Architecture	9
3.1 What peripherals are available in Versatile/PB926EJ-S?	9
3.2 Can I extend the Versatile/PB926EJ-S peripherals with an interface board?	9
3.3 How much memory is available in Versatile/PB926EJ-S? What memory devices are fitted on Versatile/PB926EJ-S?	9
3.4 Versatile/PB926EJ-S memory map.....	9
3.5 Can I change the Versatile/PB926EJ-S memory map?	11
3.6 Where in the memory map can I access the 128MB of SDRAM and 256MB of expansion SDRAM?	11
3.7 Can I access the VIC from the reset vector?	12
3.8 Can the configuration Flash be accessed by the ARM926EJ-S?	12
3.9 Can I use the MPMC to access static memory?.....	12
3.10 Versatile/PB926EJ-S seems too slow	13
3.11 Where can I find information about AHB Monitor? What can I use the AHB Monitor for?..	13
3.12 How is PCI implemented in the Versatile/PB926EJ-S FPGA? Why does it disappear when I rebuild the RTL?	13
3.13 Can I develop ASB peripherals with Versatile/PB926EJ-S?	14
3.14 Can I develop a multi-processor system with Versatile/PB926EJ-S? Can I connect a core module to Versatile/PB926EJ-S?	14
3.15 Can I connect a DSP to Versatile/PB926EJ-S?	14
3.16 Can I use Versatile/PB926EJ-S in big endian mode?	14

3.17	Can I connect Versatile/PB926EJ-S to a PCI motherboard? Can I connect Versatile/PB926EJ-S to a PC as a PCI card?	15
4.	Versatile/PB926EJ-S Hardware	16
4.1	What is the size of the ARM926EJ-S TCMs and caches?	16
4.2	What FPGA is fitted on Versatile/PB926EJ-S?.....	16
4.3	Can I change the design in the Versatile/PB926EJ-S FPGA?	16
4.4	Can I drive GLOBAL_CLK from the logic tile?.....	16
4.5	How do I need to program the clocks to access a logic tile?.....	17
4.6	The on-board oscillators do not generate the frequency I programmed	17
4.7	What is the maximum clock speed for the Versatile/PB926EJ-S clock domains?	18
4.8	What is the minimum clock speed for the Versatile/PB926EJ-S clock domains?	18
4.9	What are LTHBUSREQ, LTHGRANT, LTHLOCK and HMASTLOCKS?.....	18
4.10	The schematics and user guide show incorrect names on signals connected to logic tile stacking headers	19
4.11	Can I set the AHB bridges inside the development chip in pass-through mode?	19
4.12	How can I measure the signals on the AHB M1, M2 and S buses?	19
4.13	Can the PL180 access an SD Card via SPI?.....	20
4.14	How can I set up the input clock for the Dual Timer Module?.....	20
4.15	At what frequency is the watchdog timer clocked?.....	20
4.16	How many logic tiles can I stack on top of Versatile/PB926EJ-S?	20
4.17	The memory configuration script makes AXD or RVD hang.....	20
5.	Versatile/PB926EJ-S Firmware and Software.....	22
5.1	What operating systems have been ported to Versatile/PB926EJ-S? What operating systems are provided with Versatile/PB926EJ-S? Where can I get the operating systems and board support packages (BSPs) for Versatile/PB926EJ-S?	22
5.2	What peripherals are supported by the Versatile/PB926EJ-S BSPs?	22
5.3	What do I need to do to run the pre-built operating systems?.....	22
5.4	Where can I get drivers for the PrimeCells included in Versatile/PB926EJ-S?	23
5.5	Where can I get drivers for the Ethernet and USB peripherals?	23
5.6	What do I need to do to configure the PCI interface on Versatile/PB926Ej-S? Where can I get example code?	23
5.7	Where can I get drivers or example code for the MOVE coprocessor and the MBX graphics accelerator?.....	23
5.8	What is the Versatile/PB926EJ-S boot monitor?	23
5.9	How can I rebuild the Versatile/PB926EJ-S boot monitor?	24
5.10	Boot monitor reports an error when trying to write an image to Flash.....	24
5.11	Boot monitor semihosting problems when using the USB debugger	25
5.12	Boot monitor reports that the command line is too long	25

5.13	Boot monitor reports that the argument is invalid / Boot monitor cannot handle spaces inside arguments	25
5.14	Can I program disk-on-chip NAND flash with the Boot Monitor? What is the difference between NOR and NAND flash?	25
5.15	Boot monitor: binaries of size not multiple of 4 bytes are not written correctly to Flash / Linux is not launched correctly by Boot monitor v1.1	26
5.16	Can I load images into Flash via Ethernet?	26
5.17	Where can I find initialization code for the ARM926EJ-S MMU, caches and TCMs?	26
5.18	What does my boot code need to do to initialize the Versatile/PB926EJ-S?	27
5.19	Why do I see Flash instead of SDRAM at address 0? I cannot load my image in the target memory	27
5.20	I cannot see anything on the terminal window when I power up the board	28
6.	Versatile/PB926EJ-S Debug	29
6.1	What tools can I use to connect to the ARM926EJ-S processor?	29
6.2	What tools do I need to download new images into the Versatile/PB926EJ-S and logic tiles' FPGAs and PLDs?	29
6.3	What tools do I need in order to use the Versatile/PB926EJ-S USB debugger?	29
6.4	How can I see the Versatile/PB926EJ-S registers and memory map in RealView Debugger?	29
6.5	Can I use Angel with Versatile/PB926EJ-S?	30
6.6	I cannot get a trace from the ARM926EJ-S	30
6.7	Problems with the Versatile/PB926EJ-S USB debugger	30
6.8	Progcards_USB cannot program the logic tile PLD	31
6.9	RVD reports "Unable to connect to remote emulator" when connecting to the ARM926EJ-S using the USB debugger	31
6.10	Can I use the USB debugger to do multi-core debugging?	31
6.11	Can I use the USB debugger to get a trace of the program execution?	31
6.12	Reset exceptions are not caught by the USB debugger	31
6.13	What is the order of the TAP controllers in the system?	32
6.14	Multi-ICE cannot autoconfigure the devices in the scanchain in configuration mode.	32
6.15	Multi-ICE and the USB debugger cannot recognize the XC2V8000 device on the logic tile	33
6.16	What do LOCAL DONE and GLOBAL DONE indicate?	33
6.17	How does the board disable the USB debugger when an ICE is connected to the ICE connector?	33
6.18	What is the "USB Debug PROG" header (J38) used for?	34
6.19	How can I generate a DEVCHIP RECONFIG reset without the debugger losing connection with the ARM926EJ-S?	34

1. General Advice and Latest News

1.1 Upgrade to Versatile/PB926EJ-S v1.1 now!

The Versatile/PB926EJ-S v1.1 installation CD has already been released. This CD has the latest versions of the FPGA and PLD images of the baseboard. It also includes the latest version of the software and fixes for several problems found in the first version of the CD.

We recommend that you reprogram the Versatile/PB926EJ-S FPGA with build 50 or later. You should also reprogram the NOR and/or Disk-on-chip NAND Flash with the boot monitor image provided in v1.1 of the CD.

If you bought your Versatile/PB926EJ-S from ARM, please send an email to orders@arm.com, quoting your board serial number and contact details.

If you bought your Versatile/PB926EJ-S from a distributor, you should get support and maintenance contact from them instead of directly from ARM.

The support and maintenance policy for ARM tools is described in <http://www.arm.com/support/obtaining.html>.

1.2 Upgrades, bug fixes and frequently asked questions

The ARM website contains important information about the Versatile/PB926EJ-S. We suggest you check periodically the following pages:

- Versatile FAQs - <http://www.arm.com/support/versatile.html> - contains frequently asked questions and miscellaneous information about the board
- Versatile downloads - <http://www.arm.com/support/downloads/versatile.html> - contains schematics, netlist and bill of materials for all the Versatile boards. It also contains the latest FPGA and PLD configuration images for Versatile boards and patches for the example firmware
- Documentation - <http://www.arm.com/documentation/> - contains the user guides of all the Versatile boards, and application notes with examples showing the typical use of the boards

1.3 Help to design AHB masters and slaves in logic tiles

Application note 119 contains detailed information about how to implement AHB masters and slaves in logic tiles, and how to interface them to a Versatile/PB926EJ-S.

The application note also contains example RTL and Xilinx FPGA bit-files, which you can use as a starting point for your project and also to verify that your boards are not faulty.

You can download application note 119 from http://www.arm.com/documentation/Application_Notes/index.html.

1.4 Help to use the DMA controller and the AACI audio controller

We are currently in the process of writing a new application note (application note 115), which explains how to configure and use the AACI and DMA controllers in the Versatile/PB926EJ-S. The

application note will also include example software, which you can use as a starting point for your application.

Once released, application note 115 will be available for download from http://www.arm.com/documentation/Application_Notes/index.html.

In the meantime you can ask for a draft from ARM technical support: support-cards@arm.com

2. Introduction to Versatile/PB926EJ-S

2.1 *What is Multi-layer AHB? What is a bus matrix?*

Multi-layer AHB is an interconnection scheme, based on the AMBA AHB protocol, which enables parallel access paths between multiple masters and slaves in a system. This is achieved by using a bus interconnection matrix with several master and slave ports.

Multi-layer AHB gives the benefit of increased overall bus bandwidth and performance, and more flexible system architecture. Another key advantage of multi-layer AHB is that standard AHB master and slave modules can be used without the need for modifications.

In a typical multi-layer AHB system, such as the one in Versatile/PB926EJ-S, each AHB layer only has one master, so no arbitration or master-to-slave multiplexing is required. These layers can use the AHB-Lite protocol, meaning that they do not have to support retry or split transactions.

2.2 *What is the ARM926EJ-S PrimeXsys Platform (PXP)? What is the ARM926EJ-S development chip?*

The ARM926EJ-S PrimeXsys Platform is a high performance, extendable system solution based on the ARM926EJ-S processor. This product can be licensed from ARM and used to provide the infrastructure of a system on chip. It can then be extended with the vendor's own peripherals and logic blocks.

The ARM926EJ-S PXP is built around a multi-layer AHB bus matrix, and contains several high speed AHB masters and slaves such as PrimeCell Colour LCD, DMA and memory controllers.

The ARM PrimeXsys Community is a network of the third parties, which supply PrimeXsys compatible application software, operating systems, hardware IP blocks and EDA tools. Operating system support for the ARM926EJ-S PXP includes Linux, Symbian and WinCE.

For more information about the ARM926EJ-S PXP please contact your nearest ARM regional sales office: http://www.arm.com/contact_us/offices.html.

The ARM926EJ-S development chip is an ASIC designed by ARM and based around an ARM926EJ-S PXP. The development chip contains some extra components such as the MBX graphics accelerator and AHB bridges, which extend the functionality of the PrimeXsys platform.

The ARM926EJ-S development chip is the main component of the Versatile/PB926EJ-S board. Detailed information about the blocks inside the development chip can be found in the Versatile/PB926EJ-S user guide.

2.3 *What are the contents of the Versatile/PB926EJ-S installation CD?*

The Versatile/PB926EJ-S installation CD contains has the following contents:

- Bit-files for programming the FPGA, display PLD and configuration PLD; boardfiles and utilities Progcards and Progcards_USB
- Documentation of the board, the development chip and the PrimeCell blocks used in the board
- RTL and build scripts for the FPGA and the display PLD
- Board schematics

The software deliverables included in the CD are:

- Pre-built operating systems supplied as binary images:
 - Symbian
 - Standalone Linux
 - Linux with DHCP support

Binary images for two Linux bootloaders, Milo and U-Boot, are also provided.

WinCE is not provided as a pre-built binary image, although it has already been ported to Versatile/PB926EJ-S.

- Self-test image:

The Versatile/PB926EJ-S includes self-test source code and a pre-built image. This code checks most of the on-board resources and can be used as a starting point to develop your own applications.

The self-test code supports the following peripherals:

- Audio codec (AACI)
- Character LCD
- Colour LCD / VGA and touch screen
- GPIO, LEDs and switches
- Vectored Interrupt Controller (VIC)
- Mouse and keyboard interface
- Multimedia card (MMCI)
- SmartCard interface (SCI)
- Serial peripheral interface (SSP)
- UART

The self-test code also includes basic loop-back tests for the USB and Ethernet, but in order to use those devices you need a proper software stack, which usually comes with an operating system. You can also get drivers for the USB and Ethernet chips from the manufacturer's website.

If you plan to connect the Versatile/PB926EJ-S to a PCI card, you need to get the PCI card drivers from the card manufacturer.

- Boot Monitor:

The Boot Monitor is pre-loaded in Flash and provided as source code. It includes support code for:

- Loading and running images
- Programming and erasing NOR Flash
- Implementing a file system in disk-on-chip NAND Flash
- Accessing the system clock
- Setting the ARM926EJ-S MMU and caches

2.4 What software and tools do I need in order to use Versatile/PB926EJ-S?

The answer depends on what you want to do with the board.

Versatile/PB926EJ-S includes an on-board USB debugger. You can use the USB debugger to:

- Load new configurations to the board's FPGA and PLDs. You only need to connect your computer to the board with a USB cable and launch the utility Progcards_USB, which is provided in the Versatile/PB926EJ-S installation CD

- Connect to the ARM926EJ-S processor and debug software. At the moment the only debugger (software) that supports the USB debugger is RealView Debugger (RVD), which is part of the RealView Developer Suite (RVDS). Support for the USB debugger is included in RVDS v2.1 and later

For faster debugging you should use an external ICE box such as RealView ICE (RVI), which is compatible with RealView Debugger.

The Versatile/PB926EJ-S also has a boot monitor and the Linux utility U-Boot pre-programmed in Flash. You can use these two utilities to download new images in Flash and execute them without a debugger.

If you want to develop software for the Versatile/PB926EJ-S you need to buy some software development tools. For example you can use the RealView Compilation Tools (RVCT), which is also part of RVDS.

In order to develop hardware and synthesize it in the baseboard FPGA or in logic tiles, you need hardware development tools. All the FPGAs and PLDs in the system are from Xilinx, so you need the Xilinx tools to do the place & route. For the RTL synthesis you can use tools from Xilinx or third-party companies.

The FPGA and PLD images are synthesized by ARM with Synplify Pro and placed & routed with Xilinx ISE. Script files for these tools are provided in the installation CD.

2.5 How much does Versatile/PB926EJ-S cost?

Tools pricing information is available on the ARM website:
<http://www.arm.com/products/DevTools/pricing-devtools.html>.

For detailed pricing information please contact your local distributor or nearest ARM regional sales office: http://www.arm.com/contact_us/offices.html.

2.6 Where can I find the Versatile/PB926EJ-S schematics and user guide?

The Versatile/PB926EJ-S schematics and user guide are included in the installation CD. You can also download the latest versions from the ARM website:

- Schematics: <http://www.arm.com/support/downloads/versatile.html>
- User guide: http://www.arm.com/documentation/Boards_and_Firmware/index.html

3. Versatile/PB926EJ-S Architecture

3.1 *What peripherals are available in Versatile/PB926EJ-S?*

The following peripherals can be connected to the board:

- VGA monitor
- Colour LCD display and touch-screen (using CLCD expansion board)
- Character LCD (included with the board)
- Ethernet
- USB on-the-go (OTG) and 2 USB hosts
- 4 RS232 Serial ports
- Synchronous Serial Port
- General Purpose Input-Output (GPIO)
- 2 SmartCard sockets
- 2 Multimedia Card (MMC) or Secure Digital Card (SD) sockets
- Keyboard and mouse
- Stereo Audio Codec
- PCI interface – 3.3V edge connector

3.2 *Can I extend the Versatile/PB926EJ-S peripherals with an interface board?*

You can buy from ARM a CLCD adaptor board to connect an LCD and touch-screen to the Versatile/PB926EJ-S.

You can also get from ARM a PCI mini-backplane, so that you can connect the Versatile/PB926EJ-S to other PCI cards.

Some customers implement their own peripherals in logic tiles, and they want another board to provide a prototyping grid and connectors for these peripherals. ARM is currently designing a new board of these characteristics, which will be called Versatile/IT1. You can also design your own interface board and stack it directly on top of the logic tile.

3.3 *How much memory is available in Versatile/PB926EJ-S? What memory devices are fitted on Versatile/PB926EJ-S?*

The memory devices on the board are:

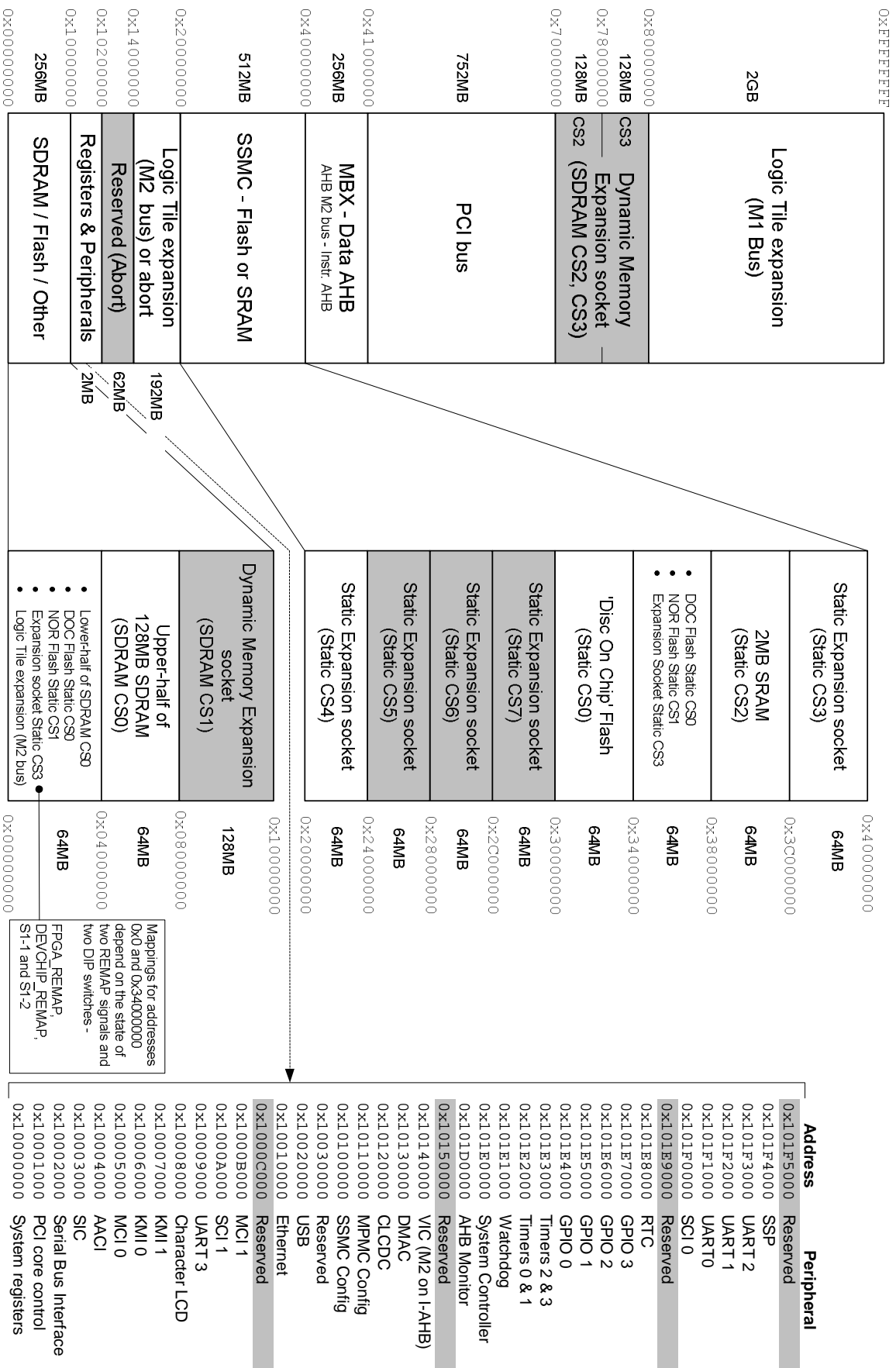
- 64MByte 32bit-wide NOR Flash, implemented with two Intel 28F256L30B90
- 64MByte Disk-on-chip NAND Flash, implemented with one M-systems MD3831-D64-V3-T
- 128MByte 32bit-wide SDRAM, implemented with two Micron MT48LC32M16A2TG-7E
- 2MByte 32bit-wide SSRAM, implemented with two Samsung K6F8016U6A-F55

The total amount of memory can be extended with optional static or dynamic memory expansion cards.

Information about the connections to the memory devices can be found in the board schematics (see section 2.6 - *Where can I find the Versatile/PB926EJ-S schematics and user guide?* of this FAQ)

3.4 *Versatile/PB926EJ-S memory map*

V/PB926EJ-S System memory map - as seen by the ARM926EJ-S data AHB interface



The figure “V/PB926EJ-S System memory map – as seen by the ARM926EJ-S data AHB interface” shows the different regions in the Versatile/PB926EJ-S memory map.

The main points are:

- Different AHB masters in the system have access to different regions of the memory map. This figure shows all the accessible regions. For information about what masters can access what slaves, see the ARM926EJ-S Development Chip Reference Manual.
- Some banks of the memory controllers are associated with chip select signals that are only routed to expansion memory connectors: J13 for dynamic memory (SDRAM) and J19 for static memory (SSRAM or Flash). The areas of memory assigned to these memory banks are shown in the figure as “static and dynamic expansion sockets”. Depending on what chip select signal is routed to a memory chip in the memory expansion card, the memory appears at a different address in memory.
- The areas of memory at address 0x0 and 0x34000000 can be remapped. The type of memory mapped to a particular area depends on the state of the DIP switches S1-1 and S1-2, and the configuration signals FPGA_REMAP and DEVCHIP_REMAP.

3.5 Can I change the Versatile/PB926EJ-S memory map?

Part of the memory map is fixed inside the development chip, part of it is hard coded in the FPGA and part of it is configurable by the customer.

The parts of the memory map fixed inside the development chip are:

- Memory at addresses 0x0 – 0x0FFFFFFF, 0x20000000 – 0x3FFFFFFF and 0x70000000 – 0x7FFFFFFF. Some of these areas can be remapped to different memory devices
- Development chip peripherals at addresses 0x10100000 – 0x10F4FFFF
- MBX at addresses 0x40000000 – 0x4FFFFFFF
- Abort at addresses 0x101F5000 – 0x13FFFFFF

The parts of the memory map hard coded in the FPGA are listed below. You can change the devices mapped at these addresses by modifying the FPGA RTL and rebuilding the FPGA image.

- Registers at addresses 0x10000000 – 0x100FFFFFF
- PCI windows at addresses 0x41000000 – 0x6FFFFFFF

The area of the memory map between addresses 0x80000000 and 0xFFFFFFFF is assigned to slaves in logic tiles, so it is totally configurable by the user.

3.6 Where in the memory map can I access the 128MB of SDRAM and 256MB of expansion SDRAM?

MPMCDYCS0 - dynamic memory chip select 0 - is the chip select signal for the Versatile/PB926EJ-S SDRAM. Chip selects 1 to 3 (MPMCDYCS1-MPMCDYCS3) are routed to the dynamic memory (SDRAM) expansion modules.

- Accesses to addresses 0x0 - 0x03FFFFFF are configurable. When the signal DEVCHIP_REMAP is high (default after reset), non-volatile memory is mapped to those addresses. When DEVCHIP_REMAP is low, MPMCDYCS0 is activated and the bottom 64MB of SDRAM appear at those addresses.
- Accesses to addresses 0x04000000 - 0x07FFFFFF always activate MPMCDYCS0 and the top 64MB of SDRAM are accessed.

Therefore, after reset, only half of the SDRAM can be accessed. In order to access the full 128MB of SDRAM you need to undo DEVCHIP_REMAP.

- Accesses to addresses 0x08000000 - 0x0FFFFFFF always activate MPMCDYCS1. If a dynamic memory expansion module is connected, the 128MB are mapped to that module.
- Accesses to addresses 0x70000000 - 0x77FFFFFF always activate MPMCDYCS2. If a dynamic memory expansion module is connected, the 128MB are mapped to that module.
- Accesses to addresses 0x78000000 - 0x7FFFFFFF always activate MPMCDYCS3. If a dynamic memory expansion module is connected, the 128MB are mapped to that module.

Note that the size of the memory devices in different memory expansion modules can be different. Please refer to the schematics or specification of your memory expansion module for more information.

3.7 Can I access the VIC from the reset vector?

The Vectored Interrupt Controller (VIC) technical reference manual recommends that the VIC is mapped within the upper 4K of memory.

This way the vector address register can be accessed directly from the IRQ vector with a load instruction from a PC-relative address in memory:

```
LDR PC, Address_Vector_Address_Register
```

If the VIC is placed anywhere else, the ARM processor needs to branch to the “main” IRQ handler first and then use these instructions to branch to the address of the handler for the source that caused the interrupt:

```
LDR Rn, =Address_Vector_Address_Register  
LDR PC, [Rn]
```

This solution is slower than the previous one and increases the interrupt latency.

On Versatile/PB926EJ-S the VIC is placed at address 0x10140000. It wasn't be placed at the top of the memory map because it would have complicated the address decoding inside the development chip and reduced its maximum operating frequency.

3.8 Can the configuration Flash be accessed by the ARM926EJ-S?

No. The configuration Flash, which contains the FPGA image, cannot be accessed by software.

The configuration Flash is not connected to the memory controllers inside the development chip, but only to the configuration port of the FPGA. The standard FPGA image does not give access to configuration Flash in order to protect it from broken software.

3.9 Can I use the MPMC to access static memory?

CFGMPMCnSMC is bit 3 of SYS_CFGDATA2. This configuration bit selects if static memory (SSRAM and Flash) is accessed by:

- Static Memory Controller (SMC) – default state, when CFGMPMCnSMC is 0
- MultiPort Memory Controller (MPMC) – if CFGMPMCnSMC is 1

The first mode is normally preferred, since using the SMC to access static memory frees the MPMC to access dynamic memory, which in general gives better system performance. The second mode is not recommended and not supported.

The Versatile/PB926EJ-S boot monitor configures the system to access static memory with the SMC.

If you want to emulate an ASIC that does not have an SMC, you can set CFGMPMCnSMC low. However you will need to configure the MPMC to access static memory. ARM cannot provide support for this configuration.

3.10 Versatile/PB926EJ-S seems too slow

Versatile/PB926EJ-S is a powerful system with fast performance. If you think that the board is not running your software as fast as it should, please check that the board is correctly configured:

- Ensure the clocks are programmed with the default frequencies: 210MHz CPU, 70MHz internal bus and 35MHz external bus
- The MMU, I-cache and D-cache should be correctly configured and enabled. You can do this with the boot monitor command “enable caches”. For example code, see the FAQ entry 5.17 - *Where can I find initialization code for the ARM926EJ-S MMU, caches and TCMs?*
- The settings of the memory controllers in the first versions of the boot monitor (v1.0 and v1.0.1) were very conservative, so the board was far from its optimum performance. The boot monitor provided in v1.1 of the installation CD configures the memory system with much faster settings.
- In some cases you can speed up the whole system by using the DMA controller to move data between memory and peripherals.

3.11 Where can I find information about AHB Monitor? What can I use the AHB Monitor for?

The ARM926EJ-S Development Chip Reference Manual contains a detailed description of the AHB Monitor.

The AHB Monitor can be used to debug complex problems or to do accurate benchmarking. It provides information about what is happening in the bus matrix inside the development chip.

- The internal bus clock (HCLK) and 32 bits of data are connected to a MICTOR connector, so that you can plug-in a logic analyzer. The 32 bits of data are divided in fields, each one showing information about the transfers taking place in a particular layer of the bus matrix.
- AHB Monitor provides memory mapped registers that contain statistical information about the accesses generated inside the development chip.

3.12 How is PCI implemented in the Versatile/PB926EJ-S FPGA? Why does it disappear when I rebuild the RTL?

The Versatile/PB926EJ-S system FPGA includes a Xilinx PCI-64 LogiCORE, which is used to interface the PCI bus. Information about this product can be found in the Xilinx website:

www.xilinx.com.

Since the PCI-64 LogiCORE is not an ARM product, it is not available in the Versatile/PB926EJ-S installation CD as RTL or even as a pre-synthesized netlist. In the FPGA RTL the PCI core has been replaced with an empty PCI module.

The result of this is that the PCI is not accessible when you re-build the FPGA RTL.

The PCI core is however implemented in the FPGA bit-files provided in the installation CD and the Support - Downloads section of the ARM website.

3.13 Can I develop ASB peripherals with Versatile/PB926EJ-S?

The Versatile/PB926EJ-S board is based around a multi-layer AHB bus matrix, so it is targeted for the development of AHB peripherals. These peripherals can be synthesized in the baseboard FPGA or in logic tiles, and can be connected directly to the three AHB buses coming off the development chip: AHB M1, M2 and S.

In order to develop ASB peripherals with Versatile/PB926EJ-S you need to include an AHB-ASB bridge in the FPGAs to connect your ASB peripherals to the AHB system buses. The AHB-ASB will reduce the overall performance of the system.

3.14 Can I develop a multi-processor system with Versatile/PB926EJ-S? Can I connect a core module to Versatile/PB926EJ-S?

It is possible to develop a multi-processor system with Versatile/PB926EJ-S, although the board is mainly oriented for single core applications.

It is not possible to connect an Integrator core module on top of Versatile/PB926EJ-S since their stacking connectors are different and there is no interface board for this type of system.

ARM is currently developing a new product called Core Tile, which contains an ARM test-chip and has logic tile form factor and header connectors. In the future it will be possible to connect a core tile on top of a Versatile/PB926EJ-S board and a logic tile. The logic tile will provide the necessary logic to interface the core tile and the Versatile/PB926EJ-S buses.

This solution will have asymmetric speed, with the ARM926EJ-S on the baseboard running faster than the processor in the core tile. The core tile will have access to most of the peripherals on the baseboard via the AHB S bus, but some peripherals can only be accessed by the ARM926EJ-S data interface.

3.15 Can I connect a DSP to Versatile/PB926EJ-S?

The Versatile/PB926EJ-S architecture lets you connect any AHB master to the baseboard via the AHB S bus, which is routed to the logic tile headers.

In order to connect a DSP to Versatile/PB926EJ-S you need to design a board with the same stacking headers as a logic tile, and which includes the DSP and an AHB wrapper or a bridge to give access to the AHB S bus.

3.16 Can I use Versatile/PB926EJ-S in big endian mode?

Yes, Versatile/PB926EJ-S supports both big and little endian modes. Note that some peripherals such as Ethernet are big or little endian only.

For more information about endianness in ARM system please see the ARM Architecture Reference Manual and Application Note 04: Programmer's Model for big-endian ARM (http://www.arm.com/documentation/Application_Notes/index.html)

3.17 Can I connect Versatile/PB926EJ-S to a PCI motherboard? Can I connect Versatile/PB926EJ-S to a PC as a PCI card?

Yes, the Versatile/PB926EJ-S behaves as a PCI card and can work in either Host or Client modes. The board can be connected only to a 3.3V PCI socket: it will not physically plug into a 5V socket, which prevents damaging the card.

The Versatile/PB926EJ-S can be connected to 32-bit and 64-bit wide PCI buses clocked at up to 66 MHz, although it only supports 8, 16 and 32 bit PCI transfers.

If you want the resources on the Versatile/PB926EJ-S to be accessed from the host processor, then the ARM926EJ-S boot code must program the PCI_SMAPx registers. The PCI core is configured so that the PCI I/O window is 1KB in size and the PCI memory windows are 1MB. This setting cannot be changed by software.

4. Versatile/PB926EJ-S Hardware

4.1 *What is the size of the ARM926EJ-S TCMs and caches?*

The ARM926EJ-S inside the development chip has:

- 32KB instruction cache
- 32KB data cache
- 32KB instruction TCM
- 32KB data TCM

Instruction and data TCMs are accessed by the core with one wait state, that is in two CPU clock cycles.

4.2 *What FPGA is fitted on Versatile/PB926EJ-S?*

The Versatile/PB926EJ-S FPGA is a Xilinx Virtex-II XC2V2000.

Information about the components fitted on the board is available in the schematics and bill of materials, which are included in the Versatile/PB926EJ-S installation CD and can also be downloaded from the Technical Support – Downloads area of the ARM website:

<http://www.arm.com/support/downloads/versatile.html>.

4.3 *Can I change the design in the Versatile/PB926EJ-S FPGA?*

Yes. The RTL for the Versatile/PB926EJ-S FPGA is provided in the installation CD. You can modify the RTL, re-build it and reprogram the configuration Flash with the bit-file generated. We only recommend this for expert users.

The XC2V2000 FPGA on the baseboard is quite small and it is nearly full with its current design. In order to implement complex designs you should use a logic tile. If you want to use the baseboard FPGA you need to remove some of the existing peripherals to make some room for your own RTL.

Some FPGA peripherals such as the system control registers must not be removed, since they are needed for the correct operation of the whole system. It is better to remove modules with a simpler interface such as the PrimeCell blocks.

4.4 *Can I drive GLOBAL_CLK from the logic tile?*

Yes, but first you need to upgrade the Versatile/PB926EJ-S FPGA to build 50 or later. The build 50 bit-file is supplied in v1.1 of the Versatile/PB926EJ-S installation CD.

The signal nGLOBALCLKEN (also called F2LSPARE[4] and connected to Z50 on HDRZ) is used by the Versatile/PB926EJ-S FPGA to drive the HCLKCTRL[0] bit of SYS_CFGDATA1.

On the baseboard HCLKCTRL[0] drives the active-low enable pin of the tri-state driver for the GLOBAL_CLK signal.

- If the logic tile does not drive GLOBAL_CLK, it should tie nGLOBALCLKEN low or leave it in high impedance
- If the logic tile drives GLOBAL_CLK, it should tie nGLOBALCLKEN high, so that the motherboard does not drive GLOBAL_CLK

Detailed information about nGLOBALCLKEN is included in the clock architecture section of the latest revision of the Versatile/PB926EJ-S user guide.

4.5 How do I need to program the clocks to access a logic tile?

First of all, if you are implementing your own peripherals in logic tiles you should upgrade to v1.1 of the Versatile/PB926EJ-S installation CD and program the Versatile/PB926EJ-S FPGA with build 50 or later.

Early versions of the FPGA such as the one supplied with the board configure the baseboard clock multiplexers so that the clock sent to the logic tile is different from the clock used by the AHB bridges. This behaviour was improved in build 50 of the FPGA, so that OSC0 is connected by default to all the clock signals in the system. This can be changed by software by writing to the HCLKCTRL bits of SYS_CFGDATA1.

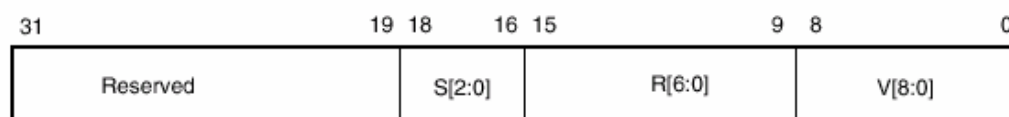
You can clock the design in the FPGA with HCLKM1F2L (for AHB M1), HCLKM2F2L (for AHB M2) and HCLKSF2L (for AHB S). Alternatively you can clock the whole design with GLOBAL_CLK.

The HCLKCTRL bits of SYS_CFGDATA1 and the CFGAHBxASYNC bits of SYS_CFGDATA2 must be set up so that the clock signals used by the logic tile are the same as the clock signals used by the AHB bridges.

Detailed information about HCLKCTRL and the clock multiplexers can be found in the clock architecture section of the latest version of the Versatile/PB926EJ-S user guide.

4.6 The on-board oscillators do not generate the frequency I programmed

The format of the SYS_OSC registers is the following:



The formula to calculate the frequency of the clock generated is explained in the clock architecture section of the Versatile/PB926EJ-S user guide. The most common reasons why the ICS307 oscillator does not generate the correct frequency are:

- 1- The user is inverting the order of V and R in the register

This is a common mistake, because the order in which these bits are streamed into the ICS307 devices is S – V – R instead of S – R – V.

When you program the SYS_OSC register you need to follow the format of the SYS_OSC register itself, not the format specified in the ICS307 datasheet

- 2- The user writes the value of R starting from bit 8 of the register instead of from bit 9

It is confusing that R does not start on a 4-bit boundary, so the value of R in hexadecimal cannot be copied directly to SYS_OSC: it needs to be shifted left by one bit first.

- 3- The user's particular combination of values for S, R and V is not supported by the ICS307.

This device has an internal PLL with a working range of frequencies. Some values of S, R and V may result in an incorrect PLL frequency, although according to the formula in the user guide the output clock is in the supported range of frequencies.

In case of doubt please check the ICS307 datasheet and use the ICS307 frequency calculator available from the Integrated Circuit Systems website: www.icst.com.

4.7 What is the maximum clock speed for the Versatile/PB926EJ-S clock domains?

This information is available in section B.2 of the Versatile/PB926EJ-S user guide (DUI 0224B)

Note that the frequencies in Table B-5 of the user guide are the maximum ones for each clock domain. In practice, the clock frequencies are interrelated, since the CPUCLK and HCLK frequencies are generated by multiplying the HCLKEXT frequency by an integer number.

The default frequencies of 35/70/210MHz for HCLKEXT/HCLK/CPUCLK are close to the maximum that work reliably with this board.

4.8 What is the minimum clock speed for the Versatile/PB926EJ-S clock domains?

The minimum frequency that can be generated by the ICS307 oscillators is 6MHz. Unless you change the design in the FPGA, this is the minimum clock frequency for HCLKEXT, HCLK and CPUCLK.

The RTL for the baseboard FPGA can be modified so that OSC0 is divided and then assigned to XTALCLKDRV.

It is important to note that if the HCLK (internal bus clock) frequency falls below 1MHz then the SDRAM controller generates refresh cycles all the time and the ARM926EJ-S is not able to access SDRAM. In this case the SDRAM should be configured in self-refresh mode, and the ARM926EJ-S should use a different memory device such as SSRAM.

The development chip's AHB bridges can be configured in synchronous or asynchronous mode. If the bridges are in asynchronous mode there is no lower limit to the frequency of the external buses.

4.9 What are LTHBUSREQ, LTHGRANT, LTHLOCK and HMASTLOCKS?

These are the AHB S bus arbitration signals between the baseboard FPGA and the logic tile stack. If there are one or more logic tiles on top of the Versatile/PB926EJ-S, then an AHB arbiter for the AHB S bus must be implemented on the bottom logic tile.

A detailed description of these signals and how they are used can be found in application note 119. A basic description is:

- LTHBUSREQ is the HBUSREQ output of the PCI master
- LTHLOCK is the HLOCK output of the PCI master
- LTHGRANT is the HGRANT input of the PCI master
- HMASTLOCKS is the HMASTLOCK output from the arbiter

The AHB arbitration signals are described in the AMBA rev2 specification, which you can download from the ARM website: <http://www.arm.com/products/solutions/AMBAHomePage.html>.

4.10 The schematics and user guide show incorrect names on signals connected to logic tile stacking headers

The Versatile/PB926EJ-S schematics and user guide assign confusing names to signals connected to the logic tile connectors.

- The documentation does not show which signals of the logic tile on top they connect to.
- Instead they show what would be the equivalent signal names if the Versatile/PB926EJ-S was a logic tile.

In summary, the documentation shows names of logic tile signals normally connected to the top headers, not to bottom headers.

For example, the signal AHBMONITOR29 is connected to HDRX[2] on the Versatile/PB926EJ-S HDRX header. In a logic tile this signal is called XU89, so that is the name showed in the VPB926EJ-S schematics and user guide. However, this signal is connected to XL89 on a logic tile stacked on top of the VPB926EJ-S.

4.11 Can I set the AHB bridges inside the development chip in pass-through mode?

By default the AHB bridges register the AHB signals from the development chip. This reduces the length and load of these signals, which allows faster clock frequencies for the internal bus clock HCLK. The downside is that the AHB bridges insert wait states when slaves outside the development chip are accessed.

In some cases customers may want to configure the bridges in pass-through (transparent) mode. In this mode the bridges behave like wires, so the transfers seen on the external buses are the same as inside the development chip.

The bridges are configured in pass-through mode by setting bit 25 (CFGAHBPASST) of SYS_CFGDATA2 and generating a DEV CHIP RECONFIG reset.

If you configure the bridges in pass-through mode, you must set CFGHCLKEXTDIVSEL = b000, so that the clock frequencies on the internal and external sides of the AHB bridges are the same. You also need to set HCLK at a low frequency (e.g. 20MHz), since the delays on the AHB signals are much longer now. When the system works you can increase the HCLK frequency until the system begins to fail and then step back a little to get the maximum working frequency.

4.12 How can I measure the signals on the AHB M1, M2 and S buses?

The best way to measure the signals on the Versatile/PB926EJ-S AHB buses is to connect a Versatile/AT1 analyzer tile on top of the baseboard. If there are any logic tiles in the system you can stack them on top of the AT1.

The AT1 provides access to all the signals on the logic tile header connectors HDRX, HDRY and HDRZ, via high-speed MICTOR connectors. These connectors are compatible with Agilent and Tektronix logic analyzers.

More information about the Versatile/AT1 can be found in the ARM website:
<http://www.arm.com/products/DevTools/AnalyzerTile1.html>.

4.13 Can the PL180 access an SD Card via SPI?

Secure Digital Cards (SD Cards) can be accessed in two modes: SD/MMC mode and SPI mode.

The Multimedia Card Interface PrimeCell PL180 can access MMC cards and SD cards in SD mode. It does not support access to SD cards in SPI mode.

In order to access an SD card via SPI you can use the PL022 SSP controller inside the development chip. This PrimeCell does support the SPI protocol.

In order to connect the SSP PrimeCell PL022 to the SD Card socket you need to modify the VPB926EJ-S FPGA RTL to:

- Remove the SD Card interface PrimeCell PL180
- Wire the signals from the SSP PrimeCell PL022 to the FPGA pins that connect to the SD Card socket
- Configure the FPGA's unused pins as inputs (or drive them with high impedance)

4.14 How can I set up the input clock for the Dual Timer Module?

The Dual Timers can only be clocked with REFCLK (32KHz) or TIMCLK (1MHz).

The clock used for each timer is selected with the TimerEnxSel bits of register SCCTRL in the System Controller. In Versatile/PB926EJ-S this register is mapped at address 0x101E0000.

The dual timer clock selection logic and the bit definition of SCCTRL are explained in the PrimeXsys System Controller (SP810) Technical Reference Manual (ARM DDI 0254B), provided on the Versatile/PB926EJ-S installation CD.

4.15 At what frequency is the watchdog timer clocked?

The watchdog timer is clocked inside the development chip by the REFCLK signal. On the Versatile/PB926EJ-S, REFCLK is driven with a 32.768KHz clock.

4.16 How many logic tiles can I stack on top of Versatile/PB926EJ-S?

There is no architectural limit to the number of logic tiles that you can stack on top of a Versatile/PB926EJ-S baseboard. The operational limit depends on the size and complexity of the design implemented in the tiles and:

- Speed of the system: stacking several tiles together increases the capacitance on the AHB M1 bus, which reduces the maximum clock speed of this bus.
- Current drawn by the logic tiles: depending on the design in the logic tiles, it is possible that the stack requires more power than can be provided by the standard power supply unit.

In this case the two possible solutions are to either use the PCI backplane to provide additional power from an ATX PSU, or connect power to the screw terminals on the baseboard.

4.17 The memory configuration script makes AXD or RVD hang

In the Versatile/PB926EJ-S installation CD there are two initialization scripts: VPB926EJS_SDRAM_Init_axd.li and VPB926EJS_SDRAM_Init_rvd.li. These scripts initialize the memory controller and remap SDRAM at address 0.

The scripts provided in v1.1 of the CD may not work in some circumstances. We endeavour to provide a fix in a future version of the CD. In the meantime, the following modifications to the script provide a temporary solution:

VPB926EJS_SDRAM_Init_axd.li – delete or comment the following lines.

```
mem 0x04020000,+4  
mem 0x08020000,+4  
mem 0x70020000,+4  
mem 0x78020000,+4
```

VPB926EJS_SDRAM_Init_axd.li – delete or comment the following lines:

```
dump /w 0x04020000  
dump /w 0x08020000  
dump /w 0x70020000  
dump /w 0x78020000
```

5. Versatile/PB926EJ-S Firmware and Software

5.1 *What operating systems have been ported to Versatile/PB926EJ-S? What operating systems are provided with Versatile/PB926EJ-S? Where can I get the operating systems and board support packages (BSPs) for Versatile/PB926EJ-S?*

Several operating systems have been ported to Versatile/PB926EJ-S, including Symbian, Linux and WinCE.

When you buy the board you get pre-built images of Symbian OS 7S and Embedded Linux 2.4.19, as binary files on the installation CD and also preloaded in Flash:

WinCE has also been ported to Versatile/PB926EJ-S but we don't provide a pre-built image with the board.

It is important to note that with the board you only get pre-built images. You do not get the OS source code or the BSP for the board. The pre-built images simply show that the operating system has been ported and give you an idea of how fast it can run.

If you want to develop applications for an operating system, you need to contact the OS vendor to get the BSP for Versatile/PB926EJ-S and OS support.

Embedded Linux sources are placed in the ARM Linux kernel source tree at www.kernel.org. You can also find related information at www.arm.com/linux and www.arm.linux.org.uk. We recommend that you use the Linux 2.6.6 release branch or later.

5.2 *What peripherals are supported by the Versatile/PB926EJ-S BSPs?*

The peripherals supported by Linux are listed at www.arm.com/linux/prebuilt_download.html.

For information about other operating systems you should contact the appropriate OS vendor.

5.3 *What do I need to do to run the pre-built operating systems?*

Instructions to program in flash and run the pre-built software images shipped with Versatile/PB926EJ-S, including the Symbian and Linux operating systems, can be found in the file C:\Program Files\ARM\Platforms\VPB926EJS\software\binaries\flash_images\readme.txt

Note that to run an operating system you need a minimum set of peripherals connected to the board.

- In order to run Symbian you need to connect a screen and a keyboard to the board.
- You can connect to Linux with a serial link to a terminal window with the same settings as boot monitor. If you launch Linux from the boot monitor, you can run a session from the boot monitor terminal window.

If you plug a screen and a keyboard to the baseboard, you can run two sessions in parallel: one from the PC terminal window and another one from the screen/keyboard.

When you run Linux for the first time you can log in as "root" with a blank password.

5.4 Where can I get drivers for the PrimeCells included in Versatile/PB926EJ-S?

We do not provide drivers for the PrimeCells included in Versatile/PB926EJ-S. These are only available to PrimeCell licensees.

However, on the installation CD we provide self-test routines that use the basic features of all the PrimeCells on the board. The self-test routines are installed into directory
C:\Program Files\ARM\Platforms\VPB926EJS\software\projects\selftest.

If you want to use all the features of a PrimeCell you can find the necessary information in the PrimeCell reference manuals provided in
C:\Program Files\ARM\Platforms\VPB926EJS\docs\ARM_TRMs.

5.5 Where can I get drivers for the Ethernet and USB peripherals?

We do not provide any driver sources or binaries for the Transdimension OTG243 USB and the SMSC LAN91C111 Ethernet chipsets, since these are not products designed by ARM.

At the moment the only way to use USB is to get drivers directly from Transdimension (www.transdimension.com) or to develop the drivers yourself from the chipset datasheet.

You can download drivers for the LAN91C111 from www.smisc.com.

Normally, in order to use Ethernet and USB you need a TCP/IP or USB stack. We recommend getting one with an operating system, which has the benefit of being already integrated.

5.6 What do I need to do to configure the PCI interface on Versatile/PB926Ej-S? Where can I get example code?

The steps required to configure the PCI interface are explained in section 4.17.2 of the Versatile/PB926EJ-S user guide – DUI 0224B.

On the Versatile/PB926EJ-S CD v1.1 there is example code that sets up the PCI interface and detects what cards are connected to the PCI bus. This software is installed at
C:\Program Files\ARM\Platforms\VPB926EJS\software\projects\examples\SimplePCIScan.

5.7 Where can I get drivers or example code for the MOVE coprocessor and the MBX graphics accelerator?

Details about the MOVE coprocessor and MBX graphics accelerator are only available to licensees of these products. We don't supply with Versatile/PB926EJ-S any detailed information or example code for these products.

5.8 What is the Versatile/PB926EJ-S boot monitor?

The boot monitor is a piece of software provided with Versatile/PB926EJ-S, which sets up and helps develop applications on this board. It is delivered as source code in the installation CD and also pre-programmed in Flash.

Boot monitor configures the board clocks and memory controllers at power-up, then displays a menu on your host computer's terminal window or debugger's console window. You can send commands to the boot monitor to perform certain operations and access the supported devices.

You can integrate the boot monitor platform library in your application in order to retarget the C library I/O functions to use the board's resources.

Boot monitor supports the following devices:

- NOR flash, used to store software images
- Disk-on-chip NAND Flash, used to store files and data
- UARTs
- Real-time (time of the year) clock
- Character LCD

Detailed information about boot monitor can be found in the Getting Started section of the Versatile/PB926EJ-S user guide – DUI 0224B.

5.9 How can I rebuild the Versatile/PB926EJ-S boot monitor?

This is explained in detail in the Getting Started section of the Versatile/PB926EJ-S user guide – DUI 0224B.

In order to rebuild the boot monitor you need RealView Developer Suite (RVDS) v2.1 or later, and the free utility gnumake.

Instructions to download the GNU utilities for a Linux environment can be found at www.gnu.org.

If you are working on a Windows platform, you need to install Cygwin in order to use gnumake. Cygwin can be downloaded from www.cygwin.com.

5.10 Boot monitor reports an error when trying to write an image to Flash

Boot monitor uses a feature called semihosting to access files in the host computer, which only works when a debugger is connected to the processor. Boot monitor detects at start-up if semihosting is available or not.

If you call the boot monitor command “write image” when no debugger is attached to the processor, then boot monitor reports that it is unable to open the file. The particular error message generated depends on the version of the boot monitor.

The best way to write images to Flash is to follow these steps:

- Power-up the board with an ICE attached to it
- Boot from NOR Flash, so that boot monitor configures the memory controllers and maps SDRAM at address 0
- Open the debugger and connect to the ARM926EJ-S
- Load the image “Boot_Monitor.axf” provided in the Versatile/PB926EJ-S installation CD
- Run the image

When the boot monitor runs this second time, it detects that semihosting is present.

- If S6-3 is OFF, the boot monitor uses the debugger's console window.
- If S6-3 is ON, the boot monitor uses the UART

You can now execute the “write image” command and boot monitor will be able to access the files in your host computer.

5.11 Boot monitor semihosting problems when using the USB debugger

Boot monitor does not detect correctly that the USB debugger is attached to the ARM926EJ-S processor. Therefore, when you connect to the board with the USB debugger and run boot monitor, it tries to communicate with the user via the serial port instead of the debugger's console window.

This limitation makes it impossible to use the boot monitor to write images to Flash. In order to do this you need an ICE box such as Multi-ICE or RealView ICE.

This is a known problem and we expect to fix it in a future release of the boot monitor or in a patch for the USB debugger.

A workaround for this problem is to configure the RealView Debugger Connection Properties to use the vpb926ej-s.bcd file. This file contains a description of the memory map and flash method files for this board. If you use this .bcd file, you can load software images directly to NOR Flash from the debugger.

5.12 Boot monitor reports that the command line is too long

Old versions of the boot monitor (v1.0 and v1.0.1) have a very low limit for the length of the command line. This is a problem when writing a file into Flash, since the command line must include the whole path for the file.

The maximum length of the command line has been extended in the version of the boot monitor provided in the v1.1 Versatile/PB926EJ-S installation disk. If you are hitting this problem, please order a CD and program the new boot monitor in Flash.

Note that there is also a known bug with the interaction of the boot monitor, RVI and RVD's console window, which limits the command line length to 64 characters. To workaround this, use boot monitor v1.1 or later and set the DIP switch S6-3 ON so that the boot monitor uses the UART instead of the debugger's console window.

5.13 Boot monitor reports that the argument is invalid / Boot monitor cannot handle spaces inside arguments

Boot monitor does not correctly handle spaces inside arguments. This can be a problem when accessing files from a computer, since in Windows OS file and directory names can contain spaces.

If you get this error, please move your files to a folder that does not contain spaces. You can replace them, for example, with underscores "_".

5.14 Can I program disk-on-chip NAND flash with the Boot Monitor? What is the difference between NOR and NAND flash?

NOR flash and NAND flash are very different devices.

NOR flash is a read-only random access memory device. It has enough address pins to map its entire contents, allowing for easy read access to each and every one of its bytes. Erase/write requires a special algorithm whose performance is extremely low.

NAND flash devices are interfaced serially via a rather complex I/O interface, which may vary from one device to another or from vendor to vendor. The advantages of NAND flash are its very high density and erase/write performance.

Disk-on-chip Flash uses NAND technology, but it has a simpler interface and is much more reliable. It also includes an "eXecute In Place (XIP)" boot block that the processor can run code from. Data outside the XIP block only support sequential accesses, so the core cannot run code from them.

All this makes NOR good for running code, while NAND is best used as a data storage device (hard drive/block device replacement) disk-on-chip flash can be seen as the next generation of NAND flash.

The Versatile/PB926EJ-S boot monitor lets you program software images into NOR Flash, but not into disk-on-chip Flash. Old versions of the boot monitor (v1.0 and v1.0.1) do not support disk-on-chip Flash at all.

Boot monitor v1.1 does include support for disk-on-chip Flash. It can be loaded in the XIP block and can retarget the file I/O library to disk-on-chip Flash.

5.15 *Boot monitor: binaries of size not multiple of 4 bytes are not written correctly to Flash / Linux is not launched correctly by Boot monitor v1.1*

Boot monitor v1.1 has an issue with the WRITE BINARY command. If the size of the binary file is not a multiple of 4 bytes, the boot monitor truncates the end of the file before writing it to Flash. This prevents the boot monitor from launching the Linux image provided in the CD.

This bug has been fixed in the boot monitor patch v1.1.1. This file can be downloaded from the ARM website: <http://www.arm.com/support/downloads/versatile.html>.

5.16 *Can I load images into Flash via Ethernet?*

Yes. The v1.1 Versatile/PB926EJ-S installation CD includes the Linux utility U-Boot as a pre-built binary. This file is also pre-programmed in NOR Flash.

You can use the boot monitor to run U-Boot and then use this utility to load images into Flash. This way you can program images into Flash, quickly, and without a debugger.

U-Boot source and documentation is available at <http://sourceforge.net/projects/u-boot>.

5.17 *Where can I find initialization code for the ARM926EJ-S MMU, caches and TCMs?*

Initialization code for the ARM926EJ-S MMU, caches and TCMs is available in:

- Versatile/PB926EJ-S boot monitor, provided in the board's installation CD
- ADS v1.2 Cached Dhrystone benchmark for the ARM926EJ-S, available for download from the ARM website: <http://www.arm.com/support/downloads/info/2317.html>
- RVDS v2.1 Cached Dhrystone benchmark for the ARM926EJ-S, available in the examples folder of the RVDS installation directory

5.18 What does my boot code need to do to initialize the Versatile/PB926EJ-S?

The easiest way to initialize the Versatile/PB926EJ-S is to use the boot monitor. You can configure the boot monitor to run your application straight after configuring the system. This is done with a boot script.

If you really want the ARM926EJ-S to boot directly from your application after a reset, then you simply need to program your boot image into flash at address 0x34000000. You can do this with boot monitor or with RVD and the vpb926ej-s.bcd board-chip definition file.

The boot software must do some configuration steps to set up the whole system. You can reuse the boot monitor initialisation code in your application:

C:\Program Files\ARM\Platforms\VPB926EJS\software\firmware\Platform\Source\sys_boot.s

The boot monitor initialization code (boot.s) performs the following functions:

- Disable interrupts
- Initialize endianness of the processor
- Configure SMC and MPMC memory controllers
- Perform 'REMAP' to put SDRAM at address 0x0
- Copy code image to execution address in RAM
- Create stack space
- Initialize peripherals, e.g. UART
- Enable TCMs if required
- Copy exception vector table to RAM
- Jump to main body of application code

The source code for any functions called in boot.s is included on the installation CD.

You may want to add some code in your boot image that configures and enable the MMU and sets the caches. For more information see the FAQ entry 5.17 - *Where can I find initialization code for the ARM926EJ-S MMU, caches and TCMs?*

5.19 Why do I see Flash instead of SDRAM at address 0? I cannot load my image in the target memory

At power-up a Flash device is mapped at address 0. When the boot monitor runs, it changes the memory map (it clears the DEVCHIP_REMAP and FPGA_REMAP bits), so that SDRAM appears at address 0.

If you connect with a debugger to the ARM926EJ-S and you see Flash at address 0 that means that the boot monitor has not run. This can happen for several reasons, the most common being:

- The boot selection DIP switches are not configured correctly, so the ARM926EJ-S does not boot from NOR Flash or disk-on-chip Flash
- The user DIP switches are configured so that boot monitor runs a boot script instead of displaying a menu
- The boot monitor has been accidentally erased from Flash

The correct settings for the DIP switches and how to reprogram boot monitor into Flash are explained in the Getting Started section of the Versatile/PB926EJ-S user guide.

When using ARM's software development tools, images are linked by default at address 0x8000. If Flash is mapped at address 0, the images may not be loaded correctly by the debugger.

- AXD does not check if the image is correctly programmed or not. The execution will simply fail when you run the image
- RVD does check if the image is correctly programmed or not. If there is Flash at address 0, RVD reports an error when trying to load the image

5.20 I cannot see anything on the terminal window when I power up the board

This can happen for a number of reasons:

- One of the FPGAs or PLDs has lost its configuration image. In this case the GLOBAL DONE LED on the baseboard is OFF. The solution is to use the utility Progcards or Progcards_USB to reprogram the FPGA and PLD configurations
- The terminal application has not been configured correctly. The correct settings are 38400 baud, 8 data bits, no parity and 1 stop bit
- If one or more logic tiles are stacked on top of the baseboard, they may be resetting the system or keeping the ARM926EJ-S in wait state, which prevents it from running any code
- The boot selection DIP switches are not configured correctly, so the ARM926EJ-S does not boot from NOR Flash or disk-on-chip Flash
- The user DIP switches are configured so that boot monitor runs a boot script instead of displaying a menu
- The boot monitor has been erased accidentally from Flash

The correct settings for the DIP switches and how to reprogram boot monitor into Flash are explained in the Getting Started section of the Versatile/PB926EJ-S user guide.

A logic tile example that works on Versatile/PB926EJ-S is provided in application note 119: http://www.arm.com/documentation/Application_Notes/index.html.

6. Versatile/PB926EJ-S Debug

6.1 *What tools can I use to connect to the ARM926EJ-S processor?*

ARM software tools are bundled together as the RealView Developer Suite (RVDS). RVDS includes compilation tools (RVCT), two debuggers (RealView Debugger and AXD) and an instruction set simulator (RVISS, also called ARMulator).

The RealView Debugger (RVD) included in RVDS v2.1 and later supports the Versatile/PB926EJ-S USB debugger, so you only need a USB cable to connect to the ARM926EJ-S processor and debug software. If you want faster debug speed or advanced debug features, you can use a RealView ICE (RVI) to connect RVD to the ARM926EJ-S.

If you are using the older ARM Developer Suite (ADS), AXD or a third party debugger, you need an ICE box such as Multi-ICE to connect to the ARM926EJ-S.

Information about all these products can be found on the ARM website www.arm.com.

6.2 *What tools do I need to download new images into the Versatile/PB926EJ-S and logic tiles' FPGAs and PLDs?*

You can load new images into the FPGAs and PLDs with either:

- Multi-ICE and the utility Progcards
- The USB debugger and the utility Progcards_USB

The utilities Progcards and Progcards_USB are included in the Versatile/PB926EJ-S installation CD. They are also part of the Versatile/PB926EJ-S zip file with the latest FPGA and PLD configuration images: <http://www.arm.com/support/downloads/versatile.html>.

6.3 *What tools do I need in order to use the Versatile/PB926EJ-S USB debugger?*

As explained in the user guide, the Versatile/PB926EJ-S has an on-board ICE emulator which connects to a host computer with a USB cable. This hardware can be used to:

- Reprogram the boards' FPGAs and PLDs with the utility Progcards_USB.
- Connect to the ARM926EJ-S with a debugger. The Versatile/PB926EJ-S USB debugger is only supported by RVD in RVDS v2.1 and later.

6.4 *How can I see the Versatile/PB926EJ-S registers and memory map in RealView Debugger?*

RealView Debugger v1.7 (included in RVDS v2.1) and later include a board-chip definition file for the Versatile/PB926EJ-S. This file is called vpb926ej-s.bcd.

If you configure the RVD connection properties to use this .bcd file, you can see the board's registers in the RVD register window. You can also see the board's memory map and load images directly into NOR Flash.

For more information about bcd files please see the RealView Debugger user guide:
http://www.arm.com/documentation/Trace_Debug/index.html.

6.5 Can I use Angel with Versatile/PB926EJ-S?

Angel has not been ported to Versatile/PB926EJ-S and there are no plans to do so.

If you want to use Angel with this board, you can port it yourself. The Angel source code is provided in the ARM Firmware Suite (AFS), which is included with Integrator boards and can also be purchased separately.

A better alternative to Angel is to use RealView Debugger (RVD) with the Versatile/PB926EJ-S USB debugger.

6.6 I cannot get a trace from the ARM926EJ-S

Due to the timing of the trace signals inside the ARM926EJ-S development chip, code cannot be traced at more than 140MHz. This happens in both normal and half-rate tracing modes.

By default, the ARM926EJ-S core clock is set to 210MHz. If you want to trace your code, you need to reduce the core clock frequency down to 140MHz or less.

This can be done by unlocking the system registers (writing 0xA05F to SYS_LOCK) and modifying the value of SYS_OSC0. By default, the core clock is programmed to be 6 times faster than the OSC0 clock, so a core clock running at 140MHz can be obtained by setting SYS_OSC0 = 0x00002C6C.

6.7 Problems with the Versatile/PB926EJ-S USB debugger

Some customers have reported problems when using the Versatile/PB926EJ-S debugger. The following steps are the first ones to try if your USB debugger is not working reliably. If this does not help, please contact your supplier for support.

- If possible, upgrade to the latest version of the driver, which is provided in RVDS v2.1

If you do not have RVDS and you just want to use Progcards_USB, you should use the USB driver provided in the Versatile/PB926EJ-S v1.1 installation CD
- After the USB driver is installed for the first time, disconnect and reconnect the USB cable to the board. We have found that if this is not done, Progcards_USB reports an error the first time it runs
- Before running Progcards_USB, close any other applications running on your computer. Progcards_USB sometimes times out if the host computer is loaded.
- If there are any logic tiles in the system, ensure that they are stacked correctly on top of the baseboard. If necessary, press on the logic tile to guarantee good electrical contact.

If the electrical contact is not good, the JTAG signals are degraded and JTAG errors are likely to occur.

6.8 Progcards_USB cannot program the logic tile PLD

Progcards_USB does not understand the format the the .svf files generated by the latest versions of the Xilinx tools. This includes the file *ltxc2v4000_102c_xc9572xl_bytestreamer_build3.svf* provided in some versions of the logic tile installation CD.

If you try to use Progcards_USB to reprogram the logic tile PLD with this file, the application ends reporting a failure and the PLD is incorrectly programmed, preventing the FPGA from loading from configuration Flash.

In order to reprogram the logic tile PLD with Progcards_USB, you need to use the file *ltxc2v4000_102c_xc9572xl_bytestreamer_build3_usb.svf*. This file was generated with an old version of the Xilinx tools to provide a workaround for this problem.

This file can be downloaded from the Versatile downloads section of the ARM website:
<http://www.arm.com/support/downloads/versatile.html>.

6.9 RVD reports “Unable to connect to remote emulator” when connecting to the ARM926EJ-S using the USB debugger

The on-board USB debugger of some of the first Versatile/PB926EJ-S boards was programmed incorrectly, which resulted in RVD reporting this error message.

This problem can be fixed locally with the Versatile/PB926EJ-S update for USB debugger, which is provided in the technical support - downloads section of the ARM website:
<http://www.arm.com/support/downloads/versatile.html>.

6.10 Can I use the USB debugger to do multi-core debugging?

No. The USB debugger is a low-end entry-level product that only supports single core debugging.

In order to debug several cores at the same time you need Multi-ICE, RealView ICE (RVI) or a third party ICE box.

6.11 Can I use the USB debugger to get a trace of the program execution?

No, the USB debugger does not work with any trace port analyzer. In order to do tracing you need one of the following combinations of tools:

- AXD + TDT + Multi-ICE + Multi-Trace
- RVD + Multi-ICE + Multi-Trace
- RVD + RealView ICE + RealView Trace

There are also third party development tools that can get a trace from the ARM926EJ-S.

6.12 Reset exceptions are not caught by the USB debugger

The USB debugger provided in RVDS v2.1 has an issue with the reset vector catch. Therefore when there is a reset while the processor is running, the execution does not stop but the processor keeps on running code from the reset vector.

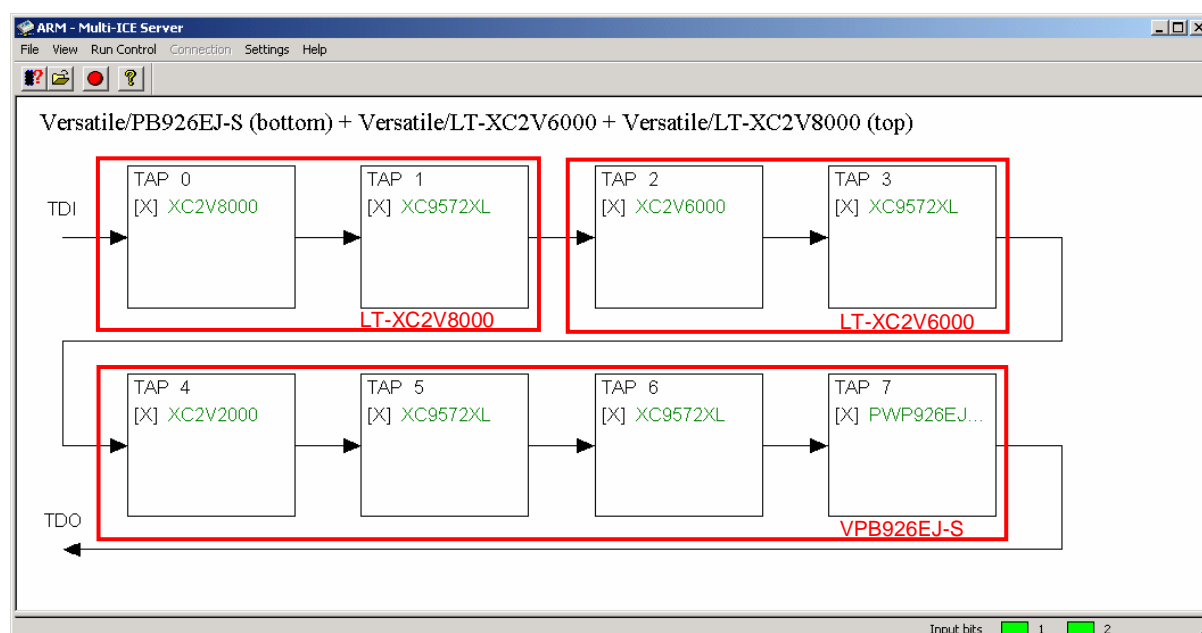
This behaviour will be fixed in RVDS v2.2.

6.13 What is the order of the TAP controllers in the system?

This information is available in the Versatile/PB926EJ-S and Versatile/LT-XC2V4000+ user guides.

The JTAG signals are routed up to the top logic tile. Then they go through the components in the stack of tiles down to the baseboard. Finally the JTAG signals are routed through the devices on the baseboard.

The figure shows the devices in the JTAG chain when a system with a Versatile/PB926EJ-S and two logic tiles is in configuration mode (CONFIG link fitted).



6.14 Multi-ICE cannot autoconfigure the devices in the scanchain in configuration mode.

Multi-ICE v2.2.5 or earlier cannot auto-configure the devices on the Versatile/PB926EJ-S. This happens because the auto-configuration code is not programmed to recognize the on-board XC2V2000 FPGA.

Currently, this can be overcome by loading a manual configuration file in Multi-ICE Server. Future versions or patches of Multi-ICE will include auto-configuration code for the Versatile/PB926EJ-S.

The Multi-ICE manual configuration files provided in the Versatile/PB926EJ-S installation CD v1.0 are incorrect. In order to reprogram the board you need to use the files provided in v1.1 of the CD or the update files from <http://www.arm.com/support/downloads/versatile.html>.

If you are using Multi-ICE v2.2.5 or earlier you need to manually add the following lines to irlength.arm in the Multi-ICE installation folder C:\Program Files\ARM\Multi-ICE:

XC2V2000=6

PWP926EJ-S_BS2=2

and/or

ARM926PXPDEV_BS2=2

The PWP926EJ-S_BS2 is the test TAP controller for the ARM926EJ-S development chip. The XC2V2000 is a Xilinx Virtex-II FPGA.

6.15 Multi-ICE and the USB debugger cannot recognize the XC2V8000 device on the logic tile

Multi-ICE v2.2.5 or earlier cannot auto-configure the XC2V8000 FPGA. Currently, this can be overcome by loading a manual configuration file in Multi-ICE Server. Future versions or patches of Multi-ICE will include auto-configuration code for this FPGA.

For Multi-ICE support for the XC2V8000, you need to add the following line to the file irlength.arm in the Multi-ICE installation folder.

```
XC2V8000=6
```

For USB support you need to add the following line to the file tapid.arm, located in the same directory as Progcards_USB.exe:

```
XC2V8000 0x31070093 0x0ffffff
```

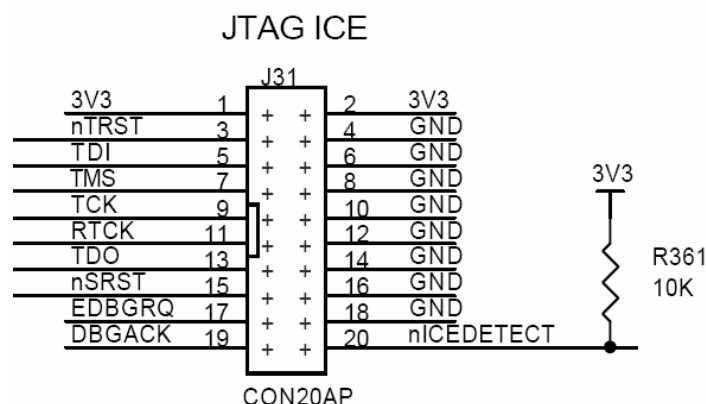
6.16 What do LOCAL DONE and GLOBAL DONE indicate?

The LOCAL DONE LED on the Versatile/PB926EJ-S is ON when the baseboard FPGA has correctly loaded its configuration image. This signal is directly connected to the DONE output of the FPGA.

The GLOBAL DONE LED is ON when all the FPGAs in the system (including the FPGAs in the logic tiles) have loaded correctly their configuration images. This signal is the logical AND of the DONE outputs of all the FPGAs.

6.17 How does the board disable the USB debugger when an ICE is connected to the ICE connector?

The signal nICEDETECT is used by the USB debugger to detect if an ICE box is connected to the board or not. When this signal is low, the USB debugger is disabled.



Pin 20 of the 20-pin JTAG connector is normally connected to ground. The Versatile/PB926EJ-S assumes that this pin is grounded inside the ICE box. Therefore, when an ICE is connected, nICEDETECT is low. When the ICE is disconnected, the pull-up resistor on nICEDETECT pulls the signal high.

The ICE detection logic works with ARM Multi-ICE and RealView ICE. We cannot guarantee that third-party ICEs will also work with this circuit.

6.18 What is the “USB Debug PROG” header (J38) used for?

The 2-pin header (J38) is associated with the USB debugger. This is labelled 'USB Debug PROG' on the PCB silk screen. This header is not described in the user documentation, since there is no need for the end user to configure this jumper. It is only used during board manufacture, for programming the XCR3064XL PLD (U68), which contains the USB debug controller logic.

ARM does not provide any utility or PLD image for reprogramming this device. If there is a problem with the USB debug port on your board, you should contact your supplier for technical support.

J38 is open-circuit (not fitted) by default, and it should be left so at all times, otherwise problems will be encountered when the USB debug port is used.

6.19 How can I generate a DEVCHIP RECONFIG reset without the debugger losing connection with the ARM926EJ-S?

In order to configure the Versatile/PB926EJ-S board it may be necessary to generate a DEVCHIP RECONFIG reset. This is needed, for example, in order to set the clock dividers inside the development chip.

It is normal to configure the board from a debugger by stopping the core and poking values into the configuration registers. The problem is that most debuggers, including AXD and RealView Debugger (RVD) lose connection with the processor when it is reset while stopped (in debug state).

In order to generate a reset safely you can follow the following steps:

- Connect with the debugger. This configures vector catch on the reset vector.
- Change the value of the SYS_CFGDATAx registers
- Put the processor in run state
- Press DEVCHIP_RECONFIG

At this moment the processor stops at the reset vector and you can now continue debugging your system.

In some cases, there isn't an application loaded in memory, so the processor generates an exception and stops as soon as you put it in run state. In this case you can use the following steps:

- Connect with the debugger. This configures vector catch on the reset vector.
- Change the value of the registers
- Put the processor in run state
- Disconnect the debugger and, when prompted, leave the target running
- Press DEVCHIP_RECONFIG

At this moment the processor stops at the reset vector. You can now re-connect with the debugger to the processor and continue debugging your system.