PXA3xx Processor Family

Design Guide

Doc. No. MV-S301368-00, Rev. A
April 6, 2009 Released

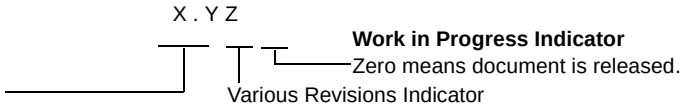
Document Conventions

	**Note:** Provides related information or information of special importance.
	**Caution:** Indicates potential damage to hardware or software, or loss of data.
	**Warning:** Indicates a risk of personal injury.

Document Status

Draft	For internal use. This document has not passed a complete technical review cycle and ECN signoff process.
Preliminary Tapeout (Advance)	This document contains design specifications for a product in its initial stage of design and development. A revision of this document or supplementary information may be published at a later date. Marvell may make changes to these specifications at any time without notice. Contact Marvell Field Application Engineers for more information.
Preliminary Information	This document contains preliminary specifications. A revision of this document or supplementary information may be published at a later date. Marvell may make changes to these specifications at any time without notice. . Contact Marvell Field Application Engineers for more information.
Complete Information	This document contains specifications for a product in its final qualification stages. Marvell may make changes to these specifications at any time without notice. Contact Marvell Field Application Engineers for more information.

Milestone Indicator:
Draft = 0.xx
Advance = 1.xx
Preliminary = 2.xx
Complete = 3.xx



Doc Status:	Technical Publication: 0.xx

For more information, visit our website at: www.marvell.com

Contents

Figures

Tables

1 Introduction

This manual outlines design guidelines, board schematics, and debug guidelines for the Marvell® PXA30x, PXA31x, and PXA32x processors (referred to throughout this document as the PXA3xx processor family). The design guidelines presented here should provide valuable information and flexibility for board designers and help reduce the risk of board-related problems. In addition to this design guide, sample schematics, software tools, operating system board support packages, and a large number of documents are available from Marvell to help system developers understand, improve, and speed up the PXA3xx processor family development process.

1.1 Document Organization and Overview

This document consists of multiple sections:

Sections 1 to 4 and Appendix A contain information that applies to the entire system design, and provides guidelines for all designs. Read and thoroughly understand these sections before attempting a new design with the PXA3xx processor family.

Sections 5 to 27 contain specific design considerations for each class of PXA3xx processor family on-chip peripherals. All sections are not applicable to a specific design because all units of the processor are not used in every design.

Sections 5 to 27 describe design recommendations and constraints related to specific on-chip peripherals of the PXA3xx processor family. These recommendations include information regarding signal connections, block diagrams, and notes related to system implementation.

The examples in this design guide represent one of many methods to connect and use the peripherals described. This does not imply that the method recommended in this document is the best or sole method of connecting and using the peripherals. Carefully review the recommendations to ensure they are appropriate for a particular design consideration.

Each peripheral is described in a separate chapter. Sections 5 to 27 are organized similarly to the sections in *Marvell® PXA30x, PXA31x, and PXA32x Processor Developers Manuals, Vols. I-IV* and use a similar format to make the information easier to locate. If multiple configurations of a peripheral exist, all possible configuration combinations are shown. However, not all configuration combinations are described.

The recommendations provided here are intended to guide and assist in the implementation of the peripherals, but they are not necessarily a “drop-in” solution for all designs. The use of this design guide and the information contained within must not replace careful consideration of system requirements and good design practices.

Table 1: How Design Guide Chapters Map to Developers Manual Volumes

Many of these PXA3xx Processor Family Design Guide Chapters Map to...	...these Volumes of the PXA3xx Processor Family Developers Manual
Clocks and Power, 1-Wire, Interrupt Controller, RTC, OST Test, JTAG (Debug)	Volume 1: System and Timer Configuration
EMPI, DFI, MMC	Volume 2: Memory Controller Configuration
LCD, Mini-LCD, Keypad, Graphic Controller, Quick Capture Interface Technology, Video	Volume 3: Graphics and Input Controller Configuration
USB 1.1 Client, USB 2.0 Client, USB 1.1 Host, SSP, AC97, UARTs, CIR, PWM, USIM, I^2C	Volume 4: Serial Controller Configuration

Additional documents that provide guidance in the design and implementation of any new system are shown in Table 2. All documents may not yet be available. Contact a Marvell representative for information regarding availability of these documents and how to access them. Regularly check to make sure you have the latest and most up-to-date versions of these documents.

Table 2: Related Documentation and Versions as of April 6, 2009*

Document Title
Marvell® PXA30x, PXA31x, and PXA32x Processors Vol I: System and Timer Configuration Developers Manual
Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual
Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. III: Graphics and Input Controller Configuration Developers Manual
Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. IV: Serial Controller Configuration Developers Manual
Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification
Marvell® PXA30x, PXA31x, and PXA32x Processor Sightings Report††
Book of iButton Standards
NOTE: † To be released only as yellow cover document to customers with an NDA. †† To be released as a yellow cover document before launch *Most documents listed here are in the process of being updated. Contact your Marvell representative to get the most recent versions.

1.2 Functional Overview

The PXA3xx processor family offers an integrated system-on-a-chip design based on the XScale®[1] Microarchitecture. The PXA3xx processor family integrates the XScale® Microarchitecture core with many on-chip peripherals allowing design of many different products for the handheld and cellular handset market segments.

See the *Marvell® PXA30x, PXA31x, and PXA32x Processor Developers Manuals, Vols. I-IV* for an overview of the PXA3xx processor family and detailed information about the PXA3xx processor family's architecture and registers. See the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for information about the AC and DC characteristics, package information, ball-map and pin list information, and power and reset timings.

1. Intel XScale® is a trademark or registered trademark of Intel Corporation and its subsidiaries in the United States and other countries.

2 PCB Design Guidelines

This section provides printed circuit-board (PCB) design guidelines for the PXA3xx processor family.

2.1 General PCB Characteristics

See Table 3 for the list of recommended PCB design characteristics using the PXA3xx processor family.

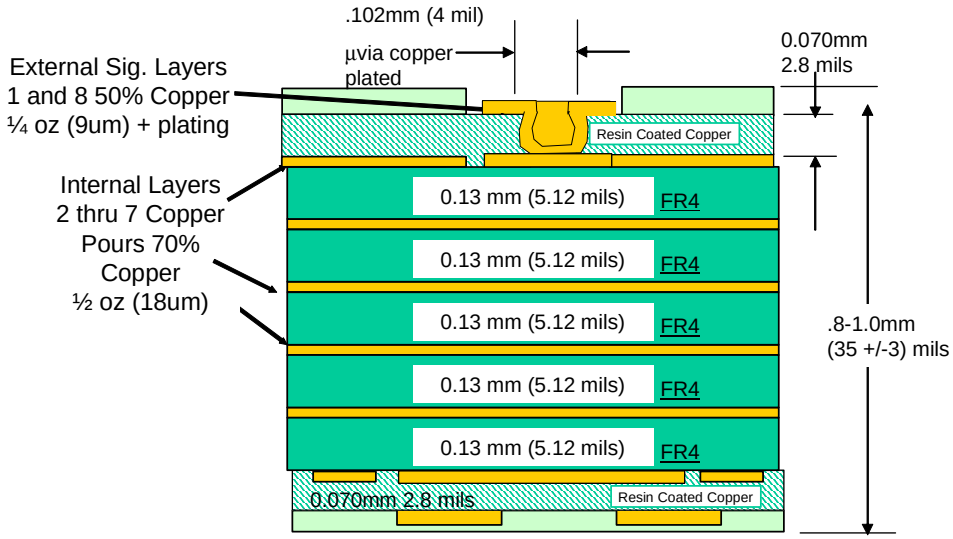
Table 3: Recommended PCB Design Guidelines

Feature	Dimensions [mm]	Dimensions [inches]
PCB Thickness	0.7874 to 1.5748 (typical)	0.0310 to 0.0620 (typical)
Top of Solder Stencil Aperture	0.2790	0.0110
Bottom of Solder Stencil Aperture	0.3000	0.0120
Solder Stencil Thickness	0.1270	0.005
PCB Layers	6 to 8 layers (typical)	
Land Pad Size	See Section 2.1.2 for package-specific specifications.	
Solder Mask Opening	See Section 2.1.2 for package-specific specifications.	
Typical trace width	See Section 2.1.2 for package-specific specifications.	
Reduced trace width between Land Pads	See Section 2.1.2 for package-specific specifications.	
Typical micro-via size	See Section 2.1.2 for package-specific specifications.	

2.1.1 PCB Layer Assignment (Stackup)

See Figure 1 for an illustration of the recommended PCB stackup dimensions and materials. The illustration shows the recommended PCB layer assignment for an eight-layer PCB using two power and two ground planes. This configuration provides a continuous VCC_APPS power plane and a divided I/O power plane for the memory and peripheral domains. See Figure 2 for an illustration of the recommended layout of the divided I/O power plane.

Figure 1: 1+6+1 Microvia PCB Stackup



For the microvia in pad, design the microvia using RCC (resin-coated copper) surface-mount capture pads.

Follow the recommendations for meshing:

- For internal layers: 70% cu = 50 mil (1.27mm) pitch with 14.6 mil (.37mm) trace width
- For external layers: 50% cu = 50 mil (1.27mm) pitch with 22.6 mil (.57mm) trace width

Follow the recommendations for surface finish and requirements for physical testing:

- Surface finish OSP
  - Use Organic Solder Preservatives (OSP) Entek 106A
  - Ensure that land pads are as flat as possible (no HASL)
  - Be aware that industry-wide problems with black pad on ENIG (electroless Ni, Immersion Au) render results useless
- Physical testing of PCBs
  - Moving point probe damages pads that affect mechanical testing results. Therefore, do not perform physical tests of PCBs

Figure 2 indicates one possible solution to a single power plane implementation of a power plane on a PCB. Exact implementation depends on system configuration, typical operation, and peripheral voltage choices. Ensure that all systems—regardless of how many power planes are implemented—account for maximum current requirements. It is recommended to use two power planes. One should be a continuous plane assigned to VCC_APPS. The second plane should be divided between memory and peripheral I/O.

Figure 2: Recommended Layer Assignment for an Eight-Layer PCB

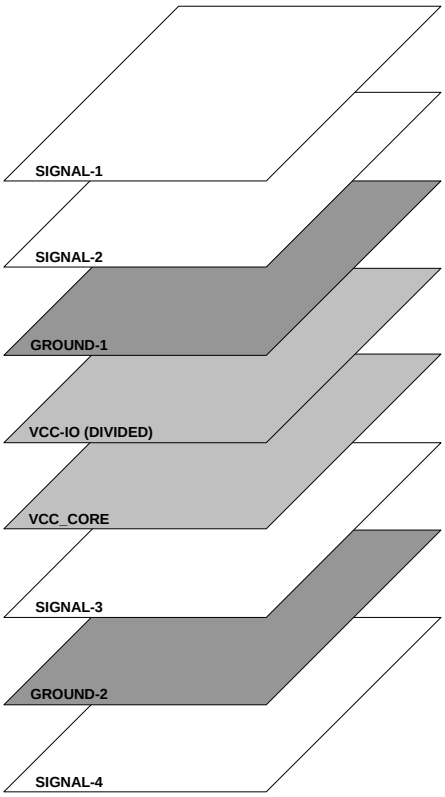
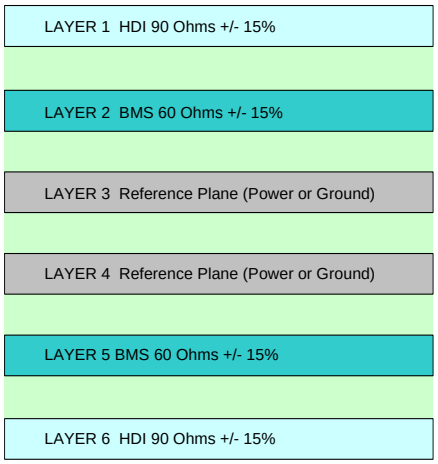


Figure 3 shows an example of 6 layers board. In this example, one full power plane is required for the DDR interface. The same power plane can be used to route signals for other interfaces as long as it acts as an unbroken reference plane for DDR signals. To avoid crosstalk, avoid parallel routing on the HDI (high-density interconnect) and BMS (buried microstrip) layers. Route the HDI layers perpendicular to the BMS layers.

Figure 3: Recommended Layer Assignment for a Six-Layer PCB



Marvell recommends allocating the following groups of power supplies as individual power planes:

- VCC_MVT, VCC_PLL, VCC_BG, VCC_OSC13M
- VCC_DF (other supplies using the same voltage may be grouped with VCC_DF)

The remaining domains that are used can be grouped together according to their operating voltage. Some power domains may not require actual power domains for proper operation, depending on the number of pins and their current requirements.

2.1.2 HD-CSP PCB Escape Routing

One important consideration when implementing HD-CSP (VF_BGA) packages on a PCB is the design of *escape routing*. Escape routing is the layout of the package signals from underneath the package to other components on the PCB. Escape routing requires high-density interconnect (HDI) PCB fabrication technology or microvias to route signals from the inner rows of balls. Table 4 shows recommended dimensions.

Table 4: PCB Dimensions for Copper-Defined Land Pads[1]

Feature	.5 mm BGA Finished[2]	.5 mm BGA Designed
A: Land Pad Size	.254 mm (.010 inches)	.279 mm (.011 inches)
B: Solder Mask Opening	.381 mm (.015 inches)	.381 mm (.015 inches)
C: Typical Trace Width	.1016 mm (.004 inches)	.127 mm (.005 inches)
D: Reduced Trace Width Between Land Pads	.0635 mm (.0025 inches)	.0889 mm (.0035 inches)
E: Typical Micro Via (Via-in-Pad) Drill Size	.102 mm (.004 inches)	.102 mm (.004 inches)

Table 4: PCB Dimensions for Copper-Defined Land Pads[1] (Continued)

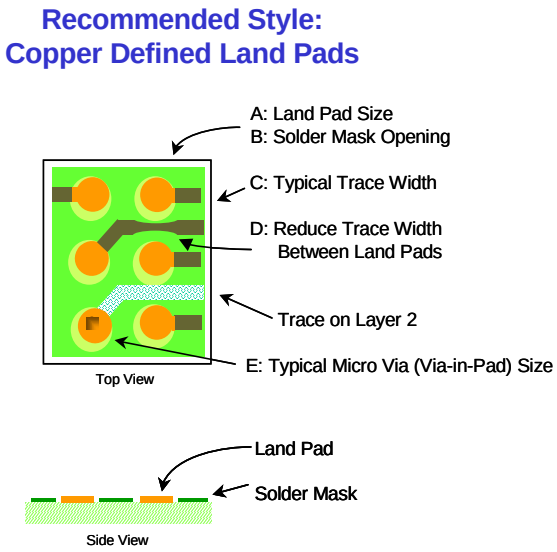
Feature	.5 mm BGA Finished[2]	.5 mm BGA Designed

NOTE:

1. PCB land pads fall into two categories:
   - Solder mask defined style (SMD)
   - Copper-defined/non-solder mask defined style (NSMD)
     Marvell recommends NSMD pads as opposed to SMD pads due to NSMDs tighter copper etching tolerance. Furthermore, copper-defined land pads provide a larger copper pad area than SMD, allowing the solder to anchor to the edges of the copper pads, which provides improved solder joint reliability
2. Finished sizes accounts for copper etch.

On the two inner rows of the HD-CSP, route the signals down from the top layer to the inner PCB layers for routing away from the package. See Figure 4 for illustration of the PCB escape routing using the HD-CSP method.

Figure 4: PCB Escape Routing for Copper-Defined Land Pads



2.1.3 PCB Keep-out Zones

Another key PCB design element is the keep-out zone, which is the distance on each side of the CSP component to the nearest adjacent component on the board. This keep-out zone varies depending on the application and is generally much tighter in handheld products that require many components in a very small PCB area. While system designers often design keep-out zones anywhere from 0.100 to 0.050 inches (2.54 mm to 1.27 mm) for embedded applications, many handheld applications are trending toward 0.025 inches (.635 mm) and smaller. The key factor to consider is how the component must be reworked if the component is replaced. Some Original Equipment Manufacturers (OEMs) require rework using a hot-air nozzle that isolates the rework area to the specific component that is being reworked. Allow adequate area for the hot-air nozzle to surround the CSP being reworked.

Another factor that impacts the PCB keep-out area requirements is the use of a socket. While sockets are only used during product development, the sockets require larger keep-out areas to accommodate mechanical mounting holes. The sockets often require backing plates that prevent the use of decoupling components under the IC package where the components are most effective.

2.2 Power Supply Decoupling Requirements

Each major power plane section or feeder trace requires a 4.7 µF or larger bulk decoupling capacitor near the processor in addition to the recommended high-speed decoupling capacitors shown in Table 5. If the voltage regulator for a given voltage domain is within 0.5” of the PXA3xx processor family, designers can use the bulk decoupling at the voltage regulator output as the bulk decoupling at the processor.

See Table 5 for decoupling capacitor recommendations for each voltage domain.

Note

- These interface guidelines are based on basic understandings, educated estimations, and simulation data.
- All decoupling capacitors should be placed within 0.5” inch of the PXA300, PXA310, and PXA320 processor. The decoupling capacitors can be placed on the board’s bottom side as long as the length restrictions are met. Each decoupling capacitor must be connected to the VCC and VSS planes by private vias – no shared vias.
- All VCC planes should be routed as planes. A thick trace (> 250 mils) is only allowed for very low frequency interfaces. The VSS plane can be a common plane.
- In actual cell phone or PDA designs, the length of VCC planes from the voltage regulator to the PXA3xx processor family should not exceed 3 inches in general. For very slow interfaces, this length can be extended up to 4 inches. DDR, USB and DFI have dedicated guidelines.
- If a signal jumps from power plane reference to ground plane reference, a stitch decap between that power and ground is recommended. If a signal jumps from one power plane reference to another power plan reference, stitch both power planes to ground. Stitching decaps should be close to where the signal changes reference. Stitching between power planes could cause noise coupling.
- Marvell recommends using 1.0 µF, 0402 size, X6S dielectric; ESL 500pH, ESR 11.2 MΩ high-frequency decoupling capacitors for the PXA3xx processor family power domains. One part that meets these specs is the TDK P/N C1005X5ROJ105. This part number is provided as a reference only. There are many other manufacturers who sell parts that meet these specifications (ESR and ESL targets).

Table 5: Decoupling Capacitors Per Voltage Domain Recommendations

Domain	PCB Decoupling	Voltage Level (TYP)	Group
VCC_BBATT	2 x 1.0 µF x 0402	3.0 V	BATT
VCC_APPS	5 x 1.0 µF x 0402	1.15 V	LVT
VCC_SRAM	2 x 1.0 µF x 0402	1.15 V	LVT
VCC_MVT	4 x 1.0 µF x 0402	1.8 V	MVT
VCC_BG	1 x 1.0 µF x 0402	1.8 V	MVT
VCC_PLL	2 x 4.7 µF x 0402	1.8 V	MVT

Table 5: Decoupling Capacitors Per Voltage Domain Recommendations (Continued)

Domain	PCB Decoupling	Voltage Level (TYP)	Group
VCC_OSC13M	—	1.8 V	MVT
VCC_IO1	1 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3 V	HVT
VCC_IO3	1 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3 V	HVT
VCC_IO4	2 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3 V **NOTE:** PXA32x processor only	HVT
VCC_IO6	2 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3 V **NOTE:** PXA32x processor only	HVT
VCC_USB	3 x 1.0 µF x 0402	3.3 V	HVT
VCC_DF	See Table 7	1.8 V, 3.0 V, 3.3V	HVT
VCC_CI	2 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3V	HVT
VCC_MEM	See Table 6	1.8 V	HVT
VCC_CARD1	2 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3 V	HVT
VCC_CARD2	2 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3 V	HVT
VCC_LCD	2 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3 V	HVT
VCC_MSL	2 x 1.0 µF x 0402	1.8 V, 3.0 V, 3.3 V	HVT
VCC_BIAS		3.3 V **NOTE:** PXA30x and PXA31x processor only	HVT
VCC_ULPI		1.8 V **NOTE:** PXA31x processor only	HVT
VCC_TSI	1 x 1.0 µF x 0402	3.3 V **NOTE:** PXA32x processor only	HVT

Table 6: VCC_MEM (DDR DRAM) Decoupling Recommendations

Device	Recommendation	Notes
PMIC Series Resistance and Inductance	R <= 50 mOhms; L ~ 10 µH	Effect of inductance is less, so a 20–30% variation might be OK
PMIC DC Set Point Error Tolerance	1.8 V +/- 3% **NOTE:** PXA32x processor only	Maximum tolerance
Tolerance at PXA3xx processor family balls	1.8 V nominal	As specified in the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification*
VCC_MVT Supply (I/O Core)	Must be separate supply from VCC_MEM (I/O output stage) supply	Common VSS at motherboard; separate VSS at package
PMIC Noise Filtration Capacitor	ESC >= 20 µF; ESL ~ 1 nH ESR <= 20 mOhms	Use 0805 or 1206 type capacitor

Table 6: VCC_MEM (DDR DRAM) Decoupling Recommendations (Continued)

Device	Recommendation	Notes
Motherboard Plane	Common VSS, but separate VCC	Thick VCC trace is OK, but inductance guidelines should be met
Motherboard Loop Inductance	Loop inductance <= 2nH Loop resistance <= 10 mΩ VCC_MEM plane is preferred, but thick trace is acceptable if: 2 inch maximum length; 250 mil minimum width with 4 mil separation between VCC and VSS planes	If stackup is different, use a simulator to determine length and width that meets 2 nH and 10 mOhm guidelines. Length is measured from PMIC/regulator pin to the capacitor's via.
Decoupling Capacitors	Use 5 capacitors: 0402, 1.0 µF, X6S dielectric; TDK P/N C1005X5ROJ105 (ESL 500pH, ESR 11.2 MΩ)	Can use equivalent capacitors that meet ESR and ESL target
Via Structure for Decoupling Capacitor	Stacked via, µvia, PTH: so that all are vertically aligned	Each capacitor should be directly connected to planes using two vias.
Distance between Decoupling Capacitor and package	Less than 1/2 inch between package edge and decoupling capacitor edge	It is assumed that VCC/VSS planes within 1/2 inch of DDR interface are very wide so that they can be considered to be infinite planes.
Via structure for ball-to-plane connection	Stacked via, µvia, PTH: so that all are vertically aligned	Each pair of VCC/VSS balls must be directly connected to corresponding planes using vias. Do not share vias between balls.
Signal Routing	See routing guidelines	Refer to Section 10.4, EMPI Layout Notes

Note VCC_MEM decoupling recommendations are based on simulation data.

Table 7: VCC_DF Decoupling Recommendations (DFI Bus Power)

Device	Recommendation	Notes
PMIC Series Resistance and Inductance	R <= 50 mΩ L ~ 10 µH	Effect of inductance is less, so a 20–30% variation might be OK
PMIC DC Set Point Error Tolerance	1.8 V +/- 5% or 3.0 V +/- 5% **NOTE:** PXA32x processor only	Maximum tolerance
Tolerance at PXA3xx processor family balls	Nominal voltage: 1.8 V, 3.0, or 3.3 V **NOTE:** PXA30x and PXA31x processor only 1.62 V - 1.98 V or 2.7 V - 3.3 V **NOTE:** PXA32x processor only	As specified in the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification*
PMIC Noise Filtration Capacitor	ESC >= 20 µF ESR <= 20 mΩ	Use 0805 or 1206 type capacitor

Table 7: VCC_DF Decoupling Recommendations (DFI Bus Power) (Continued)

Device	Recommendation	Notes
Motherboard Plane	Common VSS, but separate VCC	Thick VCC trace is OK, but inductance guidelines should be met
Motherboard Loop Inductance	Loop inductance <= 2nH Loop resistance <= 10 mΩ VCC_DF plane is preferred, but thick trace is acceptable if: 2 inch maximum length; 250 mil minimum width with 4 mil separation between VCC and VSS planes	If stackup is different, use a simulator to determine length and width that meets 2 nH and 10 mΩ guidelines. Length is measured from PMIC/regulator pin to the capacitor's via.
Decoupling Capacitors	Use 5 capacitors: 0402, 1.0 µF, X6S dielectric; TDK P/N C1005X5ROJ105 (ESL 500pH, ESR 11.2 MΩ)	Can use equivalent capacitors that meet ESR and ESL target
Via Structure for Decoupling Capacitor	Stacked via, µvia, PTH: so that all are vertically aligned	Each capacitor should be directly connected to planes using two vias.
Distance between Decoupling Capacitor and package	Less than 1/2 inch between package edge and decoupling capacitor edge	It is assumed that VCC/VSS planes within 1/2 inch of DDR interface are very wide so that they can be considered to be infinite planes.
Via structure for ball-to-plane connection	Stacked via, µvia, PTH: so that all are vertically aligned	Each pair of VCC/VSS balls must be directly connected to corresponding planes using vias. Do not share vias between balls.
Signal Timing	6" total length. Total load capacitance up to 100 pF. Driver strength 10x~6x for data; 8x~4x for control. 12x driver strength not allowed.	60 Ohm ± 15%. Follow SI guidelines.

Note VCC_DF decoupling recommendations are based on simulation data.

2.3 Unique VCC_MEM Power Supply Requirements

To reduce noise problems coupled from the DDR interface, VCC_MEM should be driven from a separate 1.8 V power supply:

- VCC_MEM is preferred be on its own 1.8 V +/- 0.1 V power supply. If not possible, other domains sharing the 1.8 V supply may require noise filtering measures, and the shared 1.8V supply that VCC_MEM is connected to must be 1.8 V +/- 0.1 V.
- VCC_MEM must not be connected to the 1.8 V supply powering VCC_BG/PLL/OSC_13M/MVT. VCC_BG/PLL/OSC_13M/MVT have their own (linear) regulator in the PMIC. VCC_MEM is typically generated by a switching regulator in the PMIC.

Note When designing shared 1.8 V power domains, Marvell recommends that customers contact their PMIC vendors to help design and characterize optimal noise filtering circuits.

2.4 Silicon Daisy Chain (SDC) Evaluation Units

Marvell offers SDC evaluation units that have been internally shorted together (to the silicon) in a daisy-chain pattern. This pattern ensures that the I/O path of the package is complete through the ball, substrate, lead beam or bond wire, silicon, and back down through a separate I/O path. These units are useful for setup and evaluation of manufacturing equipment.

2.5 RoHS (Lead Free Compliance)

Note All production Marvell® PXA30x, PXA31x, and PXA32x Processor parts are RoHS lead free

3 Minimizing Power Consumption

3.1 Overview

The PXA3xx processor family uses a complex power-management system that provides the best possible power use. Marvell recommends taking additional steps to minimize total power consumption. This section describes recommended methods for measuring power and provides detailed recommendations for minimizing overall power usage.

3.2 Achieve Minimum Power Usage During All Power Modes

The following are methods for achieving the lowest power consumption during all power modes:

- Connecting the crystal signals to an external oscillator achieves the lowest power consumption by the PXA3xx processor family. Yet, the power consumed by the oscillator is considered for total system power use. Use caution when selecting an external oscillator rather than an external crystal. Verify the external oscillator is equally or more efficient than the PXA3xx processor family using a crystal.
- Design the system so that all power domains use the lower voltage level supported within the specifications. Specifically, VCC_DF and other I/O VCC rails are at 1.8 V whenever possible.
- Ensure I/O voltage levels track power domain levels; that is, for VCC_MEM = 1.8 V, all MD pins must be driven at 1.8 V; otherwise negative current could occur.
- Ideally, the system design allows measurements of power across all domains individually, helping designers isolate high current problems down to a specific voltage domain.
- Enable all internal pull-up/pull-down resistors on GPIOs, reserved, and unused chip select pins, which is accomplished through the MFPR register settings:
  - In addition to the assigned multi-function options, each multi-function pin has an assigned register (MFPR) that allows the enabling of pull-up or pull-down functions. For example, to set PULLDOWN_EN on GPIO0, first set MFPR[PULL_SEL] (bit 15) at physical address 0x40E1_0124, then set MFPR[PULLDOWN_EN] (bit 13) at physical address 0x40E1_0124. See *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 4: “Pin Descriptions and Control”.
- Review all clock and memory controller settings. Default settings used in S0/D0/C0 and S0/D0/C1 might not achieve optimal power usage for low power modes.

3.3 Achieve Minimum Power Usage During S3/D4/C4

Methods for achieving lowest power consumption during S3/D4/C4 power mode:

- Ground VCC_APPS and VCC_SRAM to attain the lowest power consumption and validate the design.
- To verify that the PXA3xx processor family PMU is in the S3 state use JTAG to verify that PSR[SS3S] = 0b1. This step is optional.
- Clear PCFR[PUDH]. This disables the pull-up and pull-down registers for nGPIO_RESET, EXT_WAKEUP<1:0>, PWR_SCL and PWR_SDA.
- Configure OSCC[TENS3] appropriately. This setting is system dependent.
- Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for the proper configuration state for the pins during S3/D4/C4.

- Ensure TDI and TMS pins and EXT_WAKEUP<1:0> are pulled high or remain floating.
- Ensure the USB OTG differential inputs (USBOTG_P and USBOTG_N) are driven high or remain floating with no impact to OTG pins.

3.4 Achieve Minimum Power Usage During S2/D3/C4

The following are methods used for achieving lowest power consumption during S2/D3/C4 power mode:

- If none of the contents of internal SRAM are to be retained, ground VCC_APPS and VCC_SRAM to attain the lowest power consumption and validate the design. If contents within internal SRAM are to be retained, only ground VCC_APPS to attain the lowest power consumption.
- Set all other power domains to their minimum possible voltage allowable by the system configuration. This minimum possible voltage depends on the peripherals connected to the PXA3xx processor family. All voltages must comply with the voltage requirements of the PXA3xx processor family. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for more information.
- Ensure AD3R[AD3_L2] = 0b0. This verifies that the L2 cache is in the D4 state. This step is optional.
- To verify that the PXA3xx processor family PMU is in the S2 state use JTAG to check that PSR[SS2S] = 0b1. This step is optional.
- To verify that the applications subsystem is in the D3 state use JTAG to check that ASCR[D3S] = 0b1. This step is optional.
- Configure AD3R[AD3_Rx] appropriately to retain state in the internal SRAM. Write 0b1 for each bank of internal SRAM that will retain data during S2/D3/C4 and write 0b0 to all other banks.
- Clear PCFR[PUDH]. This disables the pull-up and pull-down registers for nGPIO_RESET, EXT_WAKEUP<1:0>, PWR_SCL, and PWR_SDA.
- Configure OSCC[TENS2] appropriately. This setting is system dependent. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for the proper configuration state for the pins during S2/D3/C4. Configure the multi-function pins of the PXA3xx processor family using the MFPRx registers. If a pin is configured as an input and not used as a wake-up source it should be driven low from an external source.
- Ensure TDI and TMS pins and EXT_WAKEUP<1:0> are pulled high or remain floating.
- Ensure the USB OTG differential inputs (USBOTG_P and USBOTG_N) are driven high or remain floating with no impact to OTG pins.

3.5 Achieve Minimum Power Usage During S0/D2/C2 (PXA30x and PXA31x only)

The following are methods used to achieve lowest power consumption during S0/D2/C2 power mode:

- Set VCC_APPS and VCC_CORE to the minimum allowable voltage for S0/D2/C2 as defined in the *Marvell® PXA30x and PXA31x Processor Electrical, Mechanical, and Thermal Specification*.
- Set all other power domains to their minimum possible voltage allowable by the system configuration. This is dependent upon the peripherals connected to the PXA30x and PXA31x processor. All voltages must comply with the voltage requirements of the PXA3xx processor family. Refer to the *Marvell® PXA30x and PXA31x Processor Electrical, Mechanical, and Thermal Specification* for more information.
- Clear PCFR[PUDH]. This disables the pull-up and pull-down registers for nGPIO_RESET, EXT_WAKEUP<1:0>, PWR_SCL and PWR_SDA.
- Configure AD2R[AD2_Rx] appropriately to retain state in the internal SRAM. Write 0b1 for each bank of internal SRAM that will retain data during S0/D2/C4 and write 0b0 to all other banks.

- Configure OSCC[TENS0] appropriately. This setting is system dependent.
- To verify that the applications subsystem is in the D2 state use JTAG to check that ASCR[D2S] = 0b1. This step is optional.
- Ensure AD2R[AD2_L2] = 0b0. This verifies that the L2 cache is in the D2 state. This step is optional.
- Ensure TDI and TMS pins and EXT_WAKEUP<1:0> are pulled high or remain floating.
- Ensure the USB OTG differential inputs (USBOTG_P and USBOTG_N) are driven high or remain floating with no impact to OTG pins.

3.6 Achieve Minimum Power Usage During S0/D1/C2 (PXA30x and PXA31x only)

The following methods are used to achieve lowest power consumption during S0/D1/C2 power mode:

- Set VCC_APPS and VCC_CORE to the minimum allowable voltage for S0/D1/C2 as defined in the *Marvell® PXA30x and PXA31x Processor Electrical, Mechanical, and Thermal Specification*.
- Set all other power domains to their minimum possible voltage allowable by the system configuration. This is dependent upon the peripherals connected to the PXA3xx processor family. All voltages must comply with the voltage requirements of the PXA30x and PXA31x processor. Refer to the *Marvell® PXA30x and PXA31x Processor Electrical, Mechanical, and Thermal Specification* for more information.
- Clear PCFR[PUDH]. This disables the pull-up and pull-down registers for nGPIO_RESET, EXT_WAKEUP<1:0>, PWR_SCL, and PWR_SDA.
- Configure AD2R[AD2_Rx] appropriately for the LCD buffer in the internal SRAM. Write 0b1 for each bank of internal SRAM that contains LCD frame buffer data and write 0b0 to all other banks.
- Configure OSCC[TENS0] appropriately. This setting is system dependent.
- To verify that the applications subsystem is in the D1 state use JTAG to verify that ASCR[D1S] = 0b1. This step is optional.
- Ensure AD1R[AD1_L2] = 0b0. This verifies that the L2 cache is in the D1 state. This step is optional.
- Ensure TDI and TMS pins and EXT_WAKEUP<1:0> are pulled high or remain floating.
- Ensure the USB OTG differential inputs (USBOTG_P and USBOTG_N) are driven high or remain floating with no impact to OTG pins.

3.7 Achieve Minimum Power Usage During S0/D0CS Idle

The following are methods used to achieve lowest power consumption during S0/D0CS Idle power mode:

- Set VCC_APPS and VCC_CORE to the minimum allowable voltage for S0/D0CS Idle as defined in the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification*.
- Set all other power domains to their minimum possible voltage allowable by the system configuration. This is dependent upon the peripherals connected to the PXA3xx processor family. All voltages must comply with the voltage requirements of the PXA3xx processor family. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for more information.
- Clear PCFR[PUDH]. This disables the pull-up and pull-down registers for nGPIO_RESET, EXT_WAKEUP<1:0>, PWR_SCL, and PWR_SDA.
- Configure OSCC[TENS0] appropriately. This setting is system dependent.

- Ensure TDI and TMS pins and EXT_WAKEUP<1:0> are pulled high or remain floating.
- Ensure the USB OTG differential inputs (USBOTG_P and USBOTG_N) are driven high or remain floating with no impact to OTG pins.

3.8 Achieve Minimum Power Usage During S0/D0CS Run and S0/D0/C0

The following are methods used to achieve the lowest power consumption during S0/D0CS Run and S0/D0/C0 power modes:

- S0/D0CS Run
  - Enable the L2 cache and both the instruction and data caches.
  - Use read allocate/write back (caching policy).
- S0/D0/C0
  - Ensure interrupts are disabled to prevent unexpected wake-ups.

Table 8: Power State Comparison

S0/D0/C0 similar to Marvell® PXA27x Run/Turbo	S0/D0/C1 similar to Marvell® PXA27x Core Idle	S0/D1/C2 similar to Marvell® PXA27x Standby with LCD Refresh
S0/D2/C2 similar to Marvell® PXA27x Standby	S2/D3/C4 similar to Marvell® PXA27x Sleep	S3/D4/C4 similar to Marvell® PXA27x Deep Sleep

4 Clocks and Power Interface

4.1 Overview

This section describes design recommendations and requirements for the external clock and power supply components connected to the PXA3xx processor family.

4.2 Signals

The following sections describe signals required for the clock and power interface.

4.2.1 Clock Interface Signals

See Table 9 for definition of the PXA3xx processor family clock signals as well as reset and wake-up source signals.

Table 9: Clock Manager Pin Definitions

Device I/O	Type	Definition
PXTAL_IN	Analog Input	The PXTAL_IN signal can be connected to an external 13 MHz crystal or to an external 13 MHz clock source and is the reference clock for internal PLLs.
PXTAL_OUT	Analog Output	The PXTAL_OUT signal can be connected to an external 13 MHz crystal or floated if an external 13 MHz clock source is used to drive PXTAL_IN.
TXTAL_IN	Analog Input	The TXTAL_IN signal can be connected to an external 32.768 kHz crystal or to an external clock source and is distributed to the Timekeeping control system and Power Management Unit (PMU).
TXTAL_OUT	Analog Output	The TXTAL_OUT signal can be connected to an external 32.768 kHz crystal or grounded when TXTAL_IN is driven by an external oscillator.
CLK_POUT	Output	Processor oscillator clock output for system use.
CLK_TOUT	Output	Timekeeping oscillator clock output for system use.
VCTCXO_EN	Output	VCTCXO_EN output signal is used to enable the processor oscillator clock source.

4.2.2 Power Manager Interface Control Signals

The PXA3xx processor family has an internal power manager unit (PMU) and a set of I/O signals for communicating with an external power management integrated circuit (PMIC). The I/O signals are active for initial power-up, certain reset events, device on/off events, and transitions between some operating modes. The PMIC must supply the fault signal, nBATT_FAULT, to communicate the onset of power supply problems to the PXA3xx processor family.

The PXA3xx processor family communicates with the external power controller using the signals defined in Table 10.

Table 10: Power Management Interface Signals

Signal	Definition	Active State	Signal Direction[1]
PWR_EN	Power Enable is an active-high output that enables the external low-voltage power supplies.	High	Output
SYS_EN	System Enable is an active-high output that enables the external high-voltage power supplies.	High	Output
nRESET	Forces an unconditional hardware reset	Low	Input
nRESET_OUT	Indicates to the PMIC and system that the PXA3xx processor family is in a reset state (configurable for S2, S3 and for GPIO reset). nRESET_OUT is optionally connected to an input pin on the PMIC.	Low	Output
nRESET_GPIO	Soft reset used to reset device while retaining memory state. The application subsystem are reset, but the memory controller is allowed to complete current transfers and the external memory is placed in self-refresh mode before nRESET_GPIO is propagated internally. nRESET_GPIO is a required system level input to the PXA3xx processor family; however, nRESET_GPIO connection to an output pin of the PMIC is optional.	Low	Input
EXT_WAKEUP<1:0>	Wake-up event sources from S2 and S3 state. EXT_WAKEUP<1:0> provides a wake-up event from S3 state even when S3 was entered due to an nBATT_FAULT assertion. At least one EXT_WAKEUP<1:0> is required as a system level input to the processor; however, EXT_WAKEUP<1:0> connection to output pins of the PMIC are optional. **NOTE:** PXA30x and PXA31x only use EXT_WAKEUP<0>	Low	Input
nBATT_FAULT	Indicates main battery removed or discharged	Low	Input
PWR_SCL	Dedicated Power I^2C bus clock	Clock	Bidirectional
PWR_SDA	Dedicated Power I^2C bus data	—	Bidirectional
PWR_CAP<1:0>	PWR_CAP<1:0> connect to an external capacitor which is used with the on-chip DC-DC converter circuitry to achieve very low power in S3 state. The PWR_CAP signals must be connected, because the DC-DC converter is used at all times.	Analog	N/A
PWR_OUT	PWR_OUT connects to an isolated external 0.1 µF capacitor. This connection is required and is the only allowed connection or load on the PWR_OUT pin.	Analog	N/A

NOTE:
1. Signal directions (input and output) are defined relative to the PXA3xx processor family.

The PXA3xx processor family has 20 power domains that must be sourced by external power supplies as shown in Table 11. Depending on the application, some power domains can be connected externally to be driven from a single power supply.

Table 11: Power Domains

Voltage Supply	Domain Name and Description
VCC_BBATT	BATTERY VOLTAGE Voltage-limited power from the main battery, or directly from a backup battery. VCC_BBATT must be supplied to start the power manager. VCC_BBATT powers the real-time clock and power management circuitry during initial power-on, S2/D3/C4 mode, and S3/D4/C4 mode when the main battery is installed. VCC_BBATT remains powered from the backup battery when the main power source has been discharged or removed.
VCC_MVT	Tightly regulated power supply for the ring oscillator and logic for communicating with the PMIC and their associated I/Os. This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. This supply must not be connected to any other PXA3xx processor family power domains with the exception that the PMIC can use a single supply to provide VCC_MVT, VCC_BG, VCC_OSC13M, and VCC_PLL.
VCC_BG	BANDGAP VOLTAGE Reference voltage for the bandgap generator. This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. This supply must not be connected to any other PXA3xx processor family power domains with the exception that the PMIC can use a single supply to provide VCC_MVT, VCC_BG, VCC_OSC13M, and VCC_PLL.
VCC_PLL	PLL VOLTAGE Power for the internal PLL. This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. This supply must not be connected to any other PXA3xx processor family power domains with the exception that the PMIC can use a single supply to provide VCC_MVT, VCC_BG, VCC_OSC13M, and VCC_PLL.
VCC_OSC13M	13 MHz OSCILLATOR VOLTAGE Power for the 13 MHz oscillator circuitry. This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. This supply must not be connected to any other PXA3xx processor family power domains with the exception that the PMIC can use a single supply to provide VCC_MVT, VCC_BG, VCC_OSC13M, and VCC_PLL.
VCC_APPS	APPLICATIONS PROCESSOR CORE VOLTAGE Dynamically variable core voltage. In a full-featured system, this supply is software controllable as described in Clock Controllers and Power Management Chapter in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* . In a simple system, this supply might be a fixed voltage chosen to meet the minimum voltage requirement for the highest frequency at which the PXA3xx processor family operates. VCC_APPS must be enabled when PWR_EN is asserted and disabled when PWR_EN is de-asserted.
VCC_SRAM	Variable voltage power for the internal SRAM during operation. VCC_SRAM must be enabled when PWR_EN is asserted and disabled when PWR_EN is de-asserted.
VCC_IO1 VCC_IO3	Fixed supplies for standard CMOS I/Os interfacing to external component I/Os, which are also powered from this same supply. The I/Os for external components connected to the corresponding signals on the PXA3xx processor family must be supplied from the same regulator. The VCC_IOx domains must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. Although separately named, these supplies may be connected together if their voltages are identical. For example, VCC_IO1 and VCC_IO3 both require 1.8 V if these supplies are connected together on the system PCB.

Table 11: Power Domains (Continued)

Voltage Supply	Domain Name and Description
VCC_IO4 VCC_IO6	**NOTE:** PXA32x only Fixed supplies for standard CMOS I/Os interfacing to external component I/Os, which are also powered from this same supply. The I/Os for external components connected to the corresponding signals on the PXA32x must be supplied from the same regulator. The VCC_IOx domains must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. Although separately named, these supplies may be connected together if their voltages are identical. For example, VCC_IO1 and VCC_IO6 both require 1.8 V if these supplies are connected together on the system PCB.
VCC_DF	Power to I/Os at Data Flash Interface (DFI). This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. **NOTE:** PXA30x and PXA31x processors using the MCP package must have VCC_DF connected to 1.8 V. NAND devices connected to the MCP versions are 1.8 V.
VCC_CI	Power to I/Os of Capture Interface. This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted.
VCC_LCD	Power for output drivers to LCD panel. Optionally, this domain may be strapped to another I/O supply at the required voltage if desired. This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted.
VCC_MEM	Tightly regulated power supply for DDR DRAM memory or companion chip bus I/O. Corresponding I/Os of the memory components or companion chips must be powered from the same regulator. This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. This supply must not be connected to any other PXA3xx processor family power domains.
VCC_MSL	Power for I/Os of a multipurpose fast serial port, which connects to external baseband module or other devices. Corresponding I/Os of the baseband device must be powered from the same regulator. Optionally, this domain may be strapped to another I/O supply at the required voltage if desired. This supply must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted.
VCC_CARD1 VCC_CARD2	Power for I/Os to an external removable storage or I/O card, including Universal Subscriber Identity Module (USIM) cards. These may have independent power supplies operating at different voltages. The VCC_CARDx voltage generated by the PMIC is software configurable at settings of 1.8 V, 3.0 V or 3.3 V or disabled. The software voltage control is implemented using I^2C commands or the PMIC decodes the PXA3xx processor family UVS0, nUVS1, and nUVS2 outputs. Refer to the Clock Controllers and Power Management Chapter in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* for more information. **NOTE:** 3.3 V is not a voltage supported by USIM cards. However, some alternate functions can be configured using the pins powered by VCC_CARD1 and VCC_CARD2. Therefore, these domains can be required to operate at 3.3 V.
VCC_BIAS	**NOTE:** PXA31x Processor only Internal SRAM back bias.
VCC_ULPI	**NOTE:** PXA31x Processor only Supply for USB for standard differential USB digital I/Os interfacing to external components.

Table 11: Power Domains (Continued)

Voltage Supply	Domain Name and Description
VCC_USB	**NOTE:** PXA30x Processor and PXA32x Processor only Power for USB for standard differential USB I/Os interfacing to external components, which are also supplied from a fixed 3.3V supply. **NOTE:** VCC_USB powers the I/O for the USB interfaces. The USB differential signals D+ and D- are out of compliance with the USB specification if VCC_USB falls below 2.8 V. The +5 V VBUS source from USB host controller, which must be available for bus-powered peripherals, must be supplied from an external source like the PMIC. It is not part of the PXA3xx processor family silicon.
NOTE: PXA31x Processors trusted boot designs must ensure that VCC_MVT, VCC_BG, VCC_PLL and VCC_OSC13M are connected to a power supply that is capable of 1.9 V and that no other power domains are connected to this power supply.	

Note The USB 2.0 Client UTMI connection to an external USB 2.0 UTMI PHY is 22 pins. These pins connect to multiple power domains inside the PXA3xx processor family. Since the UTMI interface must be run at 3.3 V, all of the domains the UTMI pins use must be run at 3.3 V.

4.2.2.1 Power Enable (PWR_EN)

PWR_EN is an active-high output from the PXA3xx processor family (input to the PMIC), which enables the low-voltage core and internal SRAM power supplies (VCC_APPS and VCC_SRAM). De-asserting PWR_EN informs the external regulator that the processor is going into S2/D3/C4 mode or S3/D4/C4 mode, and that the low-voltage core power supplies are to be shut down.

The PMIC turns on the low-voltage supplies in response to a PWR_EN assertion to resume normal operation. During S2/D3/C4 mode or S3/D4/C4 mode, the power controller must preserve the previous state of its regulators including the voltage for the core, so that on resumption of core power, the regulators return to their last known voltage levels.

Note The low-voltage power supplies (VCC_APPS and VCC_SRAM) can also be controlled via I^2C commands. All low-voltage power supplies must be removed before removing any high-voltage power supply.

4.2.2.2 System Power Enable (SYS_EN)

SYS_EN is an active-high output from the PXA3xx processor family (input to the PMIC), which enables the system (high-voltage) power supplies. De-asserting SYS_EN informs the power supply that the PXA3xx processor family is going into S3/D4/C4 mode, and that the high-voltage system power supplies (VCC_BG, VCC_PLL, VCC_MVT, VCC_OSC13M, VCC_IO1, VCC_IO3, VCC_LCD, VCC_MEM, VCC_CARDx, VCC_MSL, VCC_CI, VCC_DF, and VCC_BIAS, VCC_ULPI) are to be shut down. Assertion and de-assertion of SYS_EN occurs in the correct sequence with PWR_EN or I^2C commands to ensure the correct sequencing of power supplies when powering on and off the various voltage domains. To resume normal operation, the PMIC must first turn on the system I/O (high-voltage) supplies in response to SYS_EN assertion and then turn on the core (low-voltage) supplies in response to PWR_EN assertion or I^2C commands. The power controller must return all system I/O voltages to their pre-S3/D4/C4 mode levels.

Note All low-voltage power supplies must be removed before removing any high-voltage power supply. If the low-voltage power supplies (VCC_APPS and VCC_SRAM) are not disabled prior to SYS_EN being de-asserted, VCC_APPS and VCC_SRAM must be disabled if any of the high voltage supplies are disabled by the de-assertion of SYS_EN.

4.2.2.3 Hardware Reset (nRESET)

The nRESET signal is an active-low input which causes the PXA3xx processor family to enter the reset state. Assertion of nRESET cannot be gated and causes the PXA3xx processor family to enter a complete and unconditional reset state. The PXA3xx processor family does not recognize any external events while nRESET signal is asserted. nRESET typically is driven directly by an external PMIC.

nRESET is a hard reset that can cause the system to lose state or data when asserted. It is asserted for a cold start power-on event, or if for any reason a user pushes the system-reset button. The PMIC must assert nRESET for both events.

Note Because the power supplies are not on at the power-on reset, nRESET is unknown until the power supply driving nRESET is enabled. When the supply powering the circuit driving nRESET is enabled, nRESET must be asserted and then de-asserted.

The minimum assertion time for nRESET is described in the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification*.

While nRESET is asserted, the only activity inside the PXA3xx processor family is the stabilization of the timekeeper oscillator and application of the internal reset signals. All remaining internal clocks are stopped and the PXA3xx processor family is fully static. Additionally, all output signals assume their reset conditions, and the nBATT_FAULT signal from the PMIC is ignored. The nRESET_OUT signal from the PXA3xx processor family is asserted when nRESET is asserted.

4.2.2.4 Internal Reset (nRESET_OUT)

The nRESET_OUT signal is an active-low output that indicates to the PMIC and system that PXA3xx processor family is in the reset state. nRESET_OUT is asserted by the PXA3xx processor family during power-and reset sequences. If PCFR[SL_ROD] is cleared, nRESET_OUT is also asserted during S2/D3/C4 mode and S3/D4/C4 mode. If PCFR[GP_ROD] is cleared, nRESET_OUT is also asserted during GPIO reset.

4.2.2.5 GPIO Reset (nRESET_GPIO)

The nRESET_GPIO signal is an active-low input that is a wake-up event to the PXA3xx processor family and can be used to generate a GPIO reset. GPIO reset is used as a “soft” reset that allows a user to reset the PXA3xx processor family while maintaining data in external memory devices and the real-time clock. GPIO reset is invoked when nRESET_GPIO is asserted (or when PMCR[SWGR] is set). nRESET_GPIO contains an internal resistive pull-up that is enabled during power-on, hardware, watchdog, and GPIO resets. This pull-up is disabled when PCFR[PUDH] is cleared.

In many handheld systems, nRESET_GPIO is asserted when a user presses the reset button on the back of the device.

4.2.2.6 EXT_WAKEUP<1:0>

The EXT_WAKEUP<1:0> signals are used (PXA30x and PXA31x use only EXT_WAKEUP<0>) as a S2/D3/C4 mode or S3/D4/C4 mode wake-up source and function as a S3/D4/C4 mode wake-up even if S3/D4/C4 mode is entered due to the assertion of nBATT_FAULT. If nBATT_FAULT caused entry into S3/D4/C4 mode, then this pin is the only possible wake-up source. Therefore, if using nBATT_FAULT to indicate power supply health, this pin must be provided as a dedicated input.

The EXT_WAKEUP<1:0> signals are on the VCC_BBATT domain and thus this input has its Vil and Vih levels based upon VCC_BBATT voltage. See the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for details.

The EXT_WAKEUP<0> signal contains an internal resistive pull-down, and the EXT_WAKEUP<1> signal contains an internal resistive pull-up. The pull-down and pull-up are enabled during power-on, hardware, global watchdog, and GPIO resets, and disabled when PCFR[PUDH] is cleared. In addition, EXT_WAKEUP<1:0> can be used as general-purpose I/O pins and are controlled in "Power Manager EXT_WAKEUP[1:0] Control Register (PECR)" register. See *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* , "Section 8: Services Power Management" for more information on this register.

The PXA3xx processor family also detects additional wake-up inputs that cause exit from S2/D3/C4 mode. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* , "Section 8: Services Power Management" for more information on these wake-up sources.

Note PXA30x Processor and PXA31x Processor only use EXT_WAKEUP<0>.

Note The EXT_WAKEUP<1:0> signal is not detected as a wake-up-from-S2/D3/C4 mode until after S2/D3/C4 mode has been entered. S2/D3/C4 mode entry is complete when PWR_EN has been deasserted. Similarly, the EXT_WAKEUP<1:0> signal is not detected as a wake-up-from-S3/D4/C4 mode until after S3/D4/C4 mode has been entered. S3/D4/C4 mode entry is complete when SYS_EN is de-asserted.

4.2.2.7 Power Manager I^2C Clock (PWR_SCL)

The PWR_SCL signal is the power-manager I^2C clock to the external PMIC. The I^2C serial bus operates up to 400kbits/sec.

Note The PWR_SCL signal contains an internal resistive pull-up that is enabled during power-on, hardware, global watchdog, and GPIO resets and is disabled when PCFR[PUDH] is cleared. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* , "Section 8: Services Power Management" for more information.

4.2.2.8 Power Manager I^2C Data (PWR_SDA)

The PWR_SDA signal is the power-manager I^2C data signal to the external PMIC. PWR_SDA functions like an open-drain signal, so either the power-manager I2C controller or the PMIC can pull down PWR_SDA to a logic-low level.

Note The PWR_SDA signal contains an internal resistive pull-up that is enabled during power-on, hardware, global watchdog, and GPIO resets and is disabled when PCFR[PUDH] is cleared. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* , for more information.

4.2.2.9 nBATT_FAULT

The PXA3xx processor family monitors a digital input driven by the external PMIC that indicates power-supply regulator status and primary battery condition. This signal, nBATT_FAULT, permits a combination of hardware and software management for power fault conditions.

Assertion of nBATT_FAULT indicates that the main battery is low or has been removed from the system, giving the PXA3xx processor family an indication that power will shut down very soon. During this time, the processor can operate for a limited time from a lithium/lithium manganese coin-cell backup battery, or from a super cap that can supply the processor for a few cycles of full run power.

In the event of nBATT_FAULT assertion, the PXA3xx processor family enters an emergency S3/D4/C4 mode, where the only handshaking is with external SDRAM memory. This emergency S3/D4/C4 mode puts the SDRAM into self-refresh mode to preserve memory contents, if possible. The refresh current can eventually deplete the super cap or backup battery, but the backup supply would be exhausted much more quickly if the PXA3xx processor family remained in S0/D0/C0 mode. Supporting the S3/D4/C4 mode and power-saving features must be understood at both the board level design and by the PMIC.

Note The PXA3xx processor family does not recognize a wake-up event while nBATT_FAULT is asserted.

If the system is powered from an AC main source (90 VAC to 240 VAC or equivalent) while nBATT_FAULT is asserted, that fact may be used to gate off nBATT_FAULT and its normal effects on the system. This permits the system to continue operation even if the battery is removed or fully depleted. The PMIC can be designed to recognize this condition and indicate an active AC power source to the PXA3xx processor family via GPIO signals.

4.3 Layout Notes

4.3.1 Power Management Unit Capacitor Pins (PWR_CAP<1:0>)

The PXA3xx processor family integrates a low-power DC-to-DC converter for supplying internal voltages. PWR_CAP<1:0> must connect to the two terminals of an external 0.1 µF capacitor. If a polarized capacitor is used, the capacitor anode (+ plate) must be connected to PWR_CAP<1>.

4.3.2 Power Management Supply Output (PWR_OUT)

The services unit DC-to-DC converter requires an external 0.1 µF capacitor connected to the PWR_OUT pin. This connection is the only allowed connection or load on the PWR_OUT pin. This function is not optional and must be connected to a capacitor for correct operation. Connect the capacitor's other side to circuit GND. Use ceramic, non-polarized capacitors with a low equivalent series resistance (ESR).

4.4 Modes of Operations

This section provides detailed information on different modes of operations for both the clock and power interface.

4.4.1 Clock Interface

The PXA3xx processor family requires a 13 MHz timing reference that generates all core and most peripheral timing. Using a 32.768 kHz timing reference is required and dramatically reduces power consumption in the S0/D2/C2, S2/D3/C4, and S3/D4/C4 operating modes and provides accuracy of the real-time clock.

Both timing references are generated using a crystal with the on-chip oscillator or externally by clock oscillators. This flexibility eliminates the need for duplicate oscillators in systems that already use a 13.000 MHz or a 32.768 kHz oscillator. Additionally, when the PXA3xx processor family generates either clock using its on-chip oscillator, the CLK_POUT or CLK_TOUT pins can drive the clock inputs of other system components, such as a cellular baseband processor.

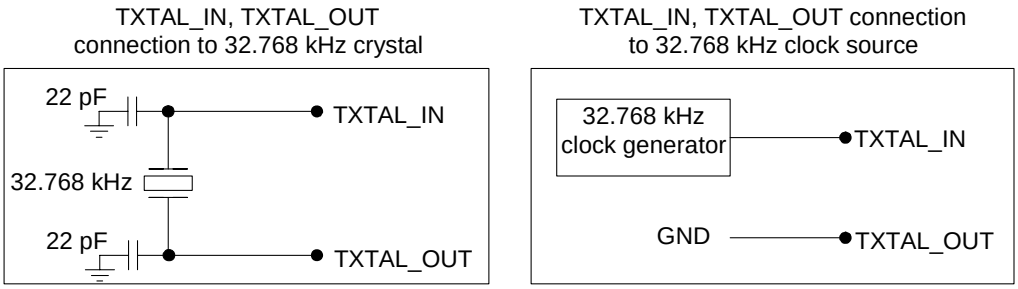
4.4.1.1 Using the On-Chip Oscillator with a 32.768 kHz Crystal

The *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* provides specifications for the 32.768 kHz crystal. To use the on-chip crystal oscillator, connect the 32.768 kHz crystal between the TXTAL_IN and TXTAL_OUT pins of the PXA3xx processor family. The on chip 32.768 kHz oscillator requires external load capacitance. Connect 22 pF external load capacitors from the TXTAL_IN and TXTAL_OUT pins of the PXA3xx processor family as shown in Figure 5. Place the crystal and capacitors as close as possible to the TXTAL_IN and TXTAL_OUT pins of the PXA3xx processor family to minimize PCB trace length and capacitance.

Route these traces parallel to each other on the same PCB layer. If the 32.768 kHz oscillator output is used by other components in the system, connect these inputs to the PXA3xx processor family CLK_TOUT output pin, and enable it. Do not attempt to connect an external load directly to the TXTAL_OUT pin.

Warning Do not route CLK_TOUT or other high-speed digital signals near the 32.768 kHz crystal, other crystal signals, or the 22 pF external load capacitors.

Figure 5: TXTAL_IN and TXTAL_OUT Connections to External Clock Sources



4.4.1.2 Using an External 32.768 kHz Clock

When using an external clock oscillator to supply the RTC timing, power the oscillator from the same source that is driving the VCC_BBATT supply input. Connect the external oscillator output to the TXTAL_IN pin. TXTAL_OUT must be grounded.

Note

Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for external 32.768 kHz oscillator-output drive levels.

4.4.1.3 Using the On-Chip Oscillator with a 13.000 MHz Crystal

The *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* provides specifications for the 13.000 MHz crystal. To use the on-chip crystal oscillator, connect the 13.000 MHz crystal between the PXTAL_IN and PXTAL_OUT pins of the PXA3xx processor family. The on-chip oscillator provides the required load capacitance. Therefore, do not connect external load capacitors to the 13.000 MHz crystal. Place the crystal as close as possible to the PXTAL_IN and PXTAL_OUT pins of the PXA3xx processor family to minimize PCB trace length and capacitance.

Route these traces parallel to each other on the same PCB layer. If the 13.000 MHz oscillator output is used by other components in the system, connect these inputs to the PXA3xx processor family CLK_POUT output pin, and enable it. Do not attempt to connect an external load directly to the PXTAL_OUT pin.

Warning

Do not route CLK_POUT or other high-speed digital signals near the 13.000 MHz crystal or other crystal signals.

Figure 6: PXTAL_IN and PXTAL_OUT Connections to External Clock Sources

4.4.1.4 Using an External 13.00 MHz Clock

When using an external clock oscillator to supply the 13 MHz timing, power the oscillator from the same source that is driving the VCC_MVT supply input. Connect the oscillator output to the PXTAL_IN pin. PXTAL_OUT is left floating.

Note

Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for external 13 MHz oscillator-output drive levels.

4.4.2 Power Interface

To enable low-power system design, the PXA3xx processor family has separate power-supply domains for the PXA3xx processor family core, memory, and peripherals. All functional units within a power domain connect to the same power supply and are powered up and down together. This architecture provides flexibility in system configuration (for example, selection of different I/O voltages for memory and peripherals) and efficient power management (for example, selecting which peripherals are powered at the same time).

In a complete system, there are other components besides the processor (such as DRAM, flash memory, audio CODECs, touchscreen controllers, and specialized companion chips) that have their own unique power requirements. In many designs, a highly integrated power controller supplies power for the processor and other components, particularly those that interface directly to the PXA3xx processor family. An advanced power controller contains circuitry for charging batteries and powering the display panel, and includes other analog functions as required by the system. The PXA3xx processor family provides several dedicated control signals as well as an Inter-Integrated Circuit (I^2C) interface (PWR_I2C) to communicate with an external power-management integrated circuit (PMIC).

.

5 One-Wire Interface

5.1 Overview

The 1-Wire interface controller is designed to receive and transmit 1-Wire bus data and provides complete control of the 1-Wire bus through eight-bit commands. The processor loads commands, reads and writes data, and sets interrupt control through five registers. All of the 1-Wire bus timing and control are generated within the 1-Wire interface controller after the host loads a command or data. When bus activity has generated a response that the CPU needs to receive, the 1-Wire master sets a status bit and, if enabled, generates an interrupt to the CPU.

The operation of the 1-Wire bus is described in detail in the *Book of iButton Standards*. Refer to this document for details on specific slave implementations.

5.2 Signals

See Table 12 for the description of the 1-Wire bus interface unit signal.

Table 12: 1-Wire Signal Description

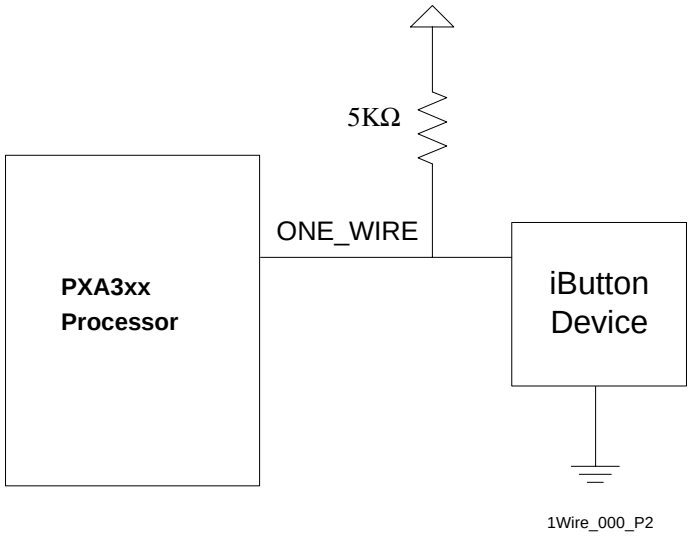
Signal Name	Input/Output	Description
ONE_WIRE	Bidirectional	This open-drain signal is the 1-Wire bidirectional data bus. 1-Wire slave devices are connected to this pin. This pin must be pulled high by an external resistor, nominally 5 K ohms.

The 1-Wire bus serial operation uses an open-drain, wired-AND bus structure that allows multiple devices to drive the bus lines and to communicate status on events such as arbitration, wait states, and error conditions.

5.2.1 1-Wire Connection

See Figure 7 for illustration of the schematic showing the connection of the 1-Wire interface to an iButton (a 1-Wire slave device).

Figure 7: 1-Wire Connection



5.2.2 Pull-ups

This signal must be pulled high by an external resistor (5 K ohms is recommended) to the same voltage of the power domain supplying power to the GPIO configured for the ONE_WIRE alternate function. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 4: Pin Descriptions and Control”. As all devices connected to the 1-Wire bus pull the ONE_WIRE signal high, power loss is minimal in this system. This line is pulled low only when data is being transmitted.

5.3 Layout Notes

The 1-Wire bus is designed to be rugged and stable. The switching frequency of the 1-Wire bus can be as low as 1 Hz; therefore, layout and routing considerations are relaxed. However, for best results, adhere to common layout recommendations. Separate the physical routing of the ONE_WIRE signal from other potential noise sources, such as switching regulators or signals with high switching frequencies.

6 Interrupt Controller Interface

This section describes the procedures for interfacing with the PXA3xx processor family interrupt controller.

6.1 Overview

The interrupt controller interfaces to both internal and external peripheral interrupt requests. A means of interfacing an external peripheral interrupt request is through the GPIO signals. All of the GPIO signals can be configured to generate an interrupt on a rising edge, falling edge, or both edges. Refer to the Interrupt and GPIO sections in the *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* for enabling interrupts through GPIO signals and for setting the edge of the appropriate GPIO interrupt.

6.2 Signals

See Table 13 for a description of the signals associated with the GPIO unit. All GPIO unit signals are programmable as interrupt signals and are used by the interrupt controller.

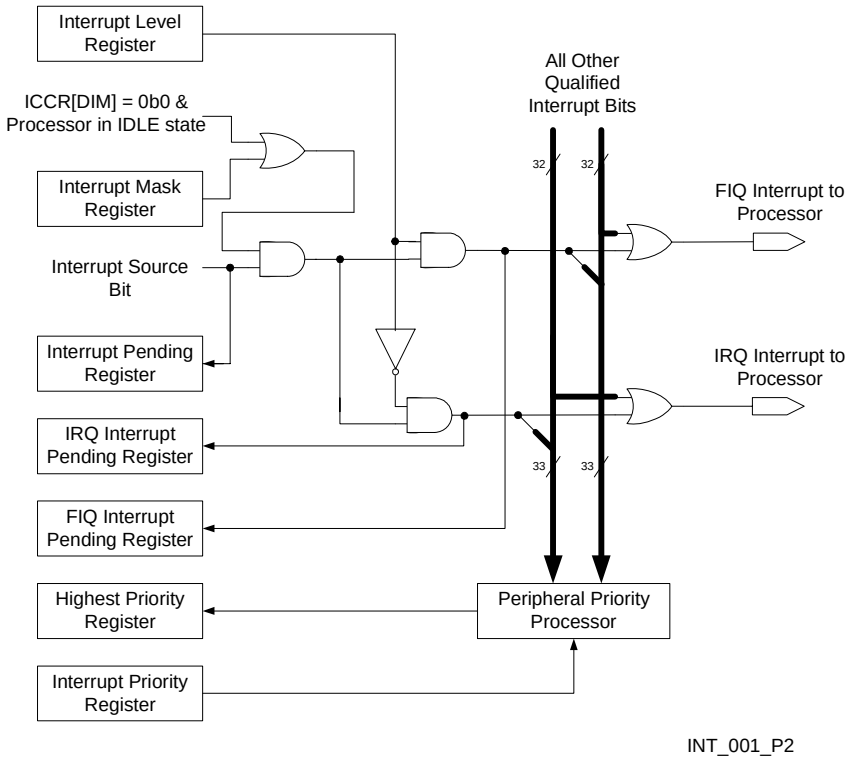
Table 13: GPIO Unit I/O Signal

Signal Name	Type	Description
GPIO<0>	Input/Output	Causes an independent first-level interrupt. Configured to generate an interrupt by setting ICPR8 bit field.
GPIO<1>	Input/Output	Causes an independent first-level interrupt. Configured to generate an interrupt by setting ICPR9 bit field.
GPIO<127:2>	Input/Output	Causes a second-level interrupt. These signals cause an interrupt if an edge is detected and ICPR10 bit field is set.

6.3 Block Diagram

See Figure 8 for an illustration of how the interrupt registers are implemented in the interrupt controller on a bit-by-bit basis. Each register in the illustration represents a bit related to a specific interrupt. This logic is copied multiple times for the additional supported possible interrupts represented. Figure 8 shows all other qualified interrupt bits.

Figure 8: Interrupt Controller Block Diagram



6.4 Layout Notes

Use caution to avoid crosstalk that is caused by routing signals too close to asynchronous signals, such as interrupt signals. Induced noise as a result of an adjacent signal causes a spurious interrupt if the amplitude of the adjacent signal is great enough and if it meets the setup and hold timing requirements on both edges. Try to keep as large a distance between interrupt signal traces and other traces either by physically separating them on the same layer, or routing them on different layers, preferably separated by a solid ground or power plane.

7 Real-Time Clock Interface

7.1 Overview

The real-time clock (RTC) interface of the PXA3xx processor family contains a single programmable signal that is configured to generate a 1.0 Hz output signal. The configuration of the RTC is accomplished through software and is described in detail in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 13: Real Time Clock (RTC)”.

The RTC HZ_CLK output signal is implemented through the GPIOs. Therefore, the hardware considerations that are necessary for the signal are the same as that of the GPIOs. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 13: Real Time Clock (RTC)” for information regarding the proper hardware implementation of the real-time clock signal.

7.2 Signals

The RTC HZ_CLK output signal is implemented through the GPIOs of the PXA3xx processor family. Refer to the GPIO alternate function table in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 4: Pin Descriptions and Control” for the GPIO assignments of the RTC HZ_CLK signal.

See Table 14 for the description of the RTC HZ_CLK output signal, which is controlled by the RTC controller of the PXA3xx processor family.

Table 14: RTC Interface Signal List

Signal Name	Type	Description
HZ_CLK	Output	1 Hz clock generated by the RTC trimmer module

8 OS Timer Interface

This section describes procedures for interfacing the OS timer controller to the PXA3xx processor family.

8.1 Overview

The operating system timers (OSTs) block provides a set of timer channels that allow software to generate timed interrupts (or wake-up events). In the PXA3xx processor family, these interrupts are generated by two sets of timer channels:

One set provides one counter and four match registers that are clocked from a 3.25 MHz clock. This block maintains the four Marvell® PXA25x processor-compatible timer channels.

- The other set (additional to the PXA25x processor) provides eight counters and eight match registers that are clocked from any of the following:
  - 32.768 kHz timer clock
  - 13 MHz clock
  - An externally supplied clock (CLK_EXT pin) that provides for a wide range of timer resolutions.

All references to registers are documented in the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* and *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 14: Operating System Timers” unless otherwise noted.

8.2 Signals

See Table 15 for the list of signals used to interface to the OSTs.

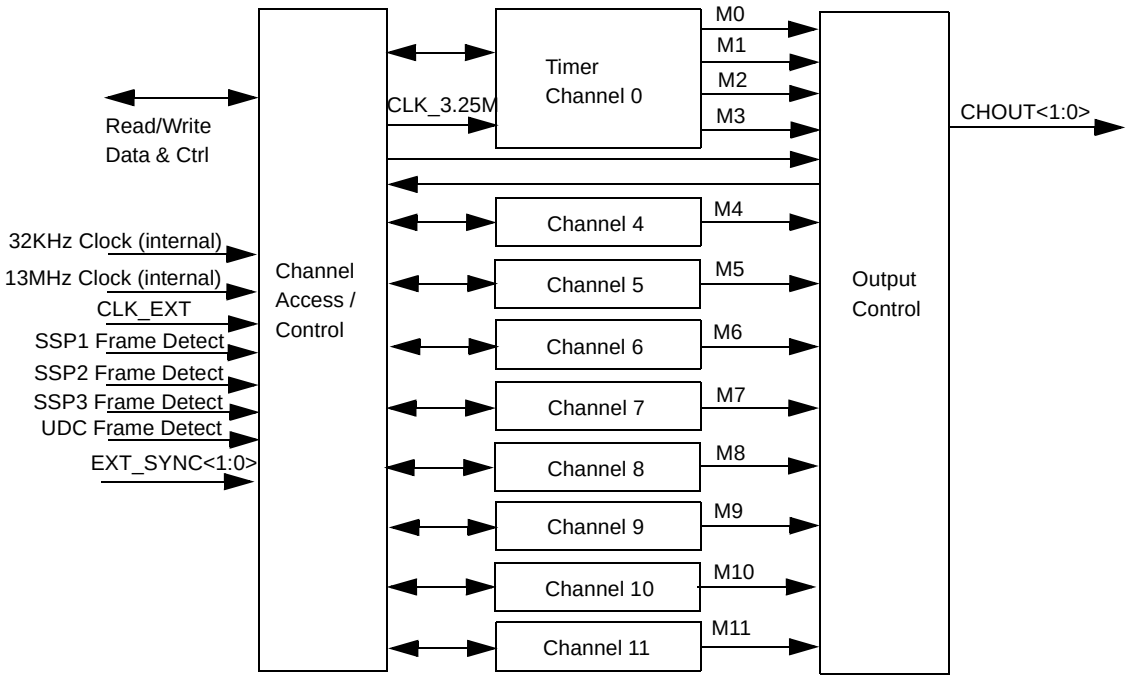
Table 15: Operating System Timers Interface Signals Summary

Name	Type	Polarity	Description
EXT_SYNC<1:0>	Input	Active High	External synchronization signals used to reset timers.
CHOUT<1:0>	Output	N/A	CHOUT<1> is a periodic output clock generated from channel 11. CHOUT<0> is a periodic output clock generated from channel 10. These active-high output signals are used to generate output clocks from timer channels 11 and 10, respectively, if these channels have been programmed to be periodic. The signal for an active channel that has been programmed to be periodic will change state each time a match occurs. The signal will be low for any non-periodic channel.

8.3 Block Diagram

See the block diagram of the OST controller in Figure 9.

Figure 9: Block Diagram of Operating System Timers Block



8.3.1 Channel Access / Control Block

This block controls reads and writes to registers within the OST controller. It is also responsible for maintaining the OS Match Control registers (OMCR4 - OMCR11) and generating the appropriate clocks and control signals for each timer channel.

8.3.2 Marvell® PXA25x Processor-Compatible Channels 0-3 Block

This block maintains the four Marvell® PXA25x processor-compatible timer channels and for generating the appropriate channel-match signals.

8.3.3 Channels 4 - 11 Blocks

Channels 4 through 11 are eight additional independent channels each with its own counter, match register, and Control register. Each independent counter is clocked with any of these software selectable clocks:

- 32.768 kHz clock for low power
- 13.0 MHz clock for high accuracy
- Externally supplied clock (CLK_EXT pin) for network synchronization
- The frame-detect signal from SSP1, SSPSFRM (available only on channels 8-11)

- The frame-detect signal from SSP2, SSPSFRM2 (available only on channels 8-11)
- The frame-detect signal from SSP3, SSPSFRM3 (available only on channels 8-11)
- The start-of-frame signal from USB 1.1 client controller (UDC) (available only on channels 8-11)

8.3.4 Output Control

This block collects the match signals from each timer channel and generates the following signals:

- Match signals to the interrupt controller
- Channel-output signals to the GPIO block
- Watchdog-reset signal

8.4 Layout Notes

The EXT_SYNCx input signals are clocked using a two-stage synchronizer. Therefore, depending on the setting of the OMCRx[CRES] bit field, an EXT_SYNCx signal must remain asserted for three clock periods of the source clock.

Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for all AC timing information.

9 JTAG Debug

This section describes the hardware interface for the PXA3xx processor family JTAG port. JTAG provides a way of driving and sampling the external pins of the device regardless of the core state. This method enables test of both the device electrical connections to the circuit board and (in conjunction with other devices on the circuit board having a similar interface) the integrity of the circuit board connections between devices. JTAG also provides a mechanism for device debug via the XScale® CPU debug features.

9.1 Overview

JTAG provides a way of driving and sampling the external pins of the device regardless of the core state, as well as a mechanism for device debug. JTAG logic includes a test-access port (TAP) controller, TAP pins, an instruction register, and Test Data registers (TDRs). The JTAG interface is controlled through five dedicated TAP pins that interface to the TAP controller: TDI, TMS, TCK, nTRST, and TDO. The IEEE 1149.1 standard effectively requires that TDI, TMS, and nTRST have internal pull-up resistors. The TDI and TMS pins can remain unconnected for normal operation and over driven to use the JTAG features. Marvell recommends that TCK be tied low for cases where JTAG is not used.

9.2 Features

The JTAG interface has the following features and conformities:

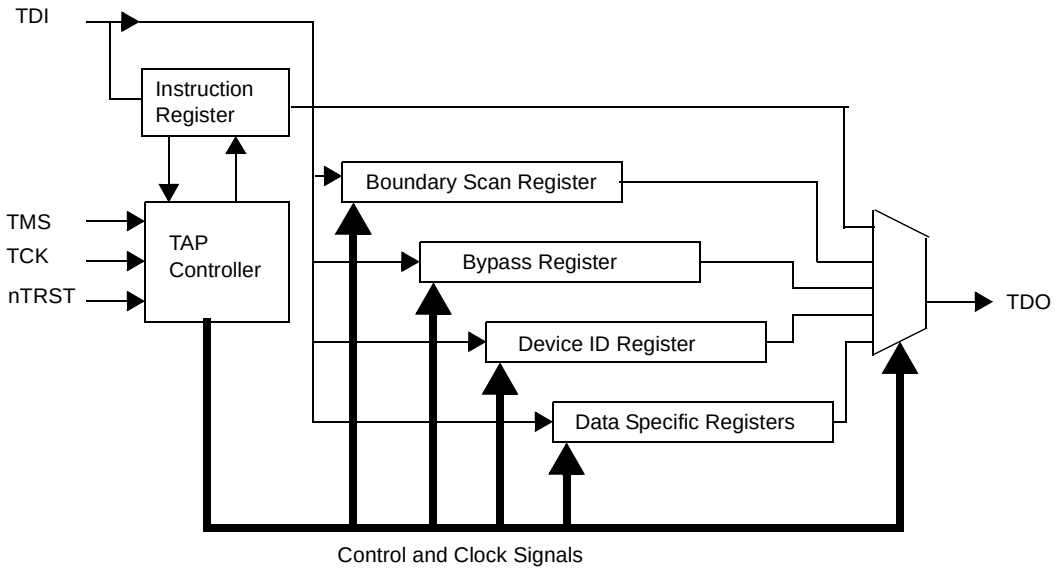
- JTAG interface that conforms to the IEEE Std. 1149.1 – 1990, IEEE Std. 1149.1a-1993, Standard Test Access Port and Boundary-Scan Architecture.
- Support for hardware/software debug

Refer to the IEEE Std. 1149.1 for an explanation of the terms used in this section and for a complete description of the TAP controller states.

9.3 Block Diagram

JTAG logic includes a test-access port (TAP) controller, TAP pins, an instruction register, and Test Data registers (TDRs). These TDRs include the Boundary Scan register (BSR - the register used to directly control the I/O pins), Bypass register, Device Identification (ID) register, and data-specific registers, as shown in Figure 10. Data are shifted into all registers most significant bit (MSB) first serially. The JTAG interface is controlled through five dedicated TAP pins that interface to the TAP controller: TDI, TMS, TCK, nTRST, and TDO.

Figure 10: JTAG Block Diagram



9.4 Signal Descriptions

The JTAG interface is controlled through five dedicated TAP pins: TDI, TMS, TCK, nTRST, and TDO. These pins are described in Table 16.

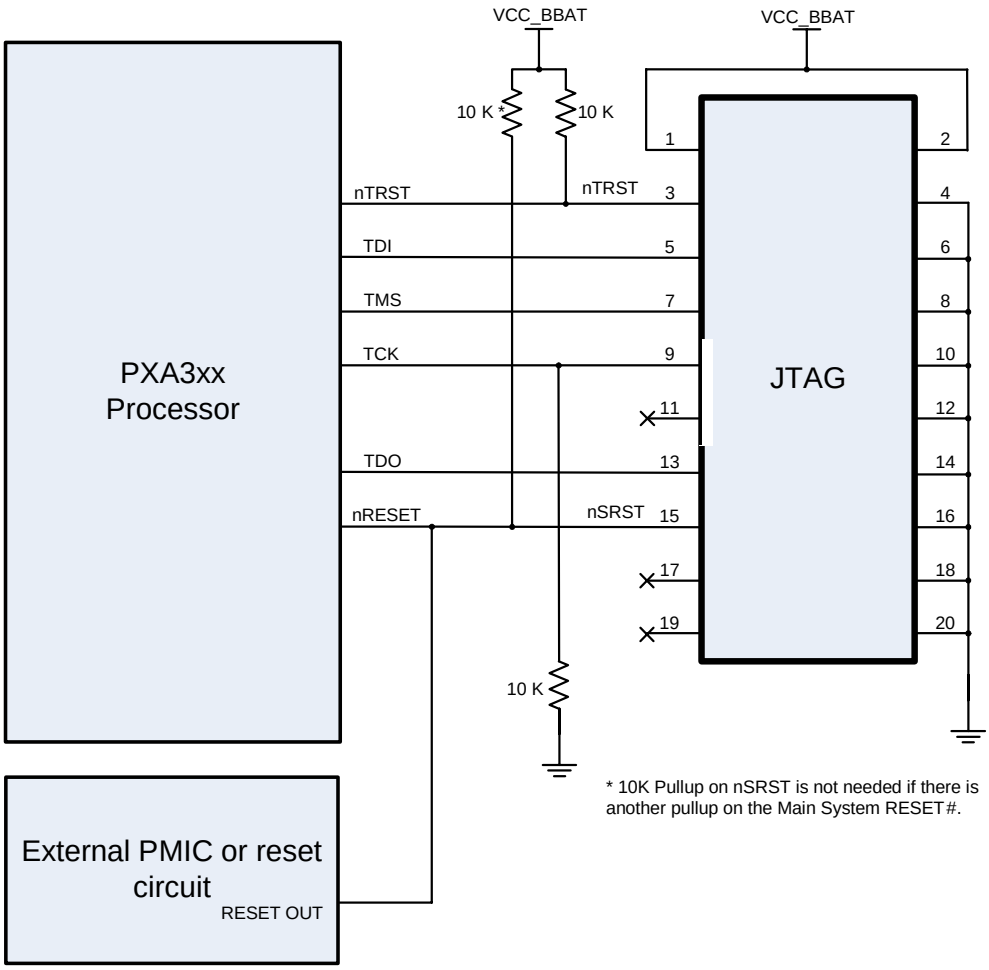
Table 16: TAP Controller Pin Definitions

Name	Type	Description
TCK	Input	Clock input for the TAP controller, Instruction register, and test data registers. Limited to 3.25 MHz maximum frequency for Boundary Scan Register, and 13 MHz for all other registers. There is no minimum.
TMS	Input	Controls operation of the TAP controller. TMS is sampled on the rising edge of TCK. The TMS input is pulled high when not being driven.
TDI	Input	Serial data input to the instruction and test data registers. Data at TDI is sampled on the rising edge of TCK. TDI is pulled high by an internal pull-up resistor when not being driven.
TDO	Output	Serial data output. Data at TDO is clocked out on the falling edge of TCK. It provides an inactive (high-Z) state during non-shift operations to support parallel connection of TDO outputs at the board or module level.
nTRST	Input	Provides asynchronous initialization of the JTAG test logic. Assertion of this pin puts the TAP controller in the TEST_Logic_Reset state. An external source must drive nTRST either before or at the same time as the hardware nRESET pin for correct TAP controller and device operation. The nTRST input is pulled high by an internal pull-up resistor when not being driven.

9.5 JTAG Board Connection

Figure 11 shows the necessary connections to a 20-pin, 100-mil JTAG-keyed connector.

Figure 11: PCB JTAG Connection



Note

For systems with a system-level reset (reset button):

1. The processor nTRST input should be connected to the JTAG nTRST ANDed with Main System RESET.

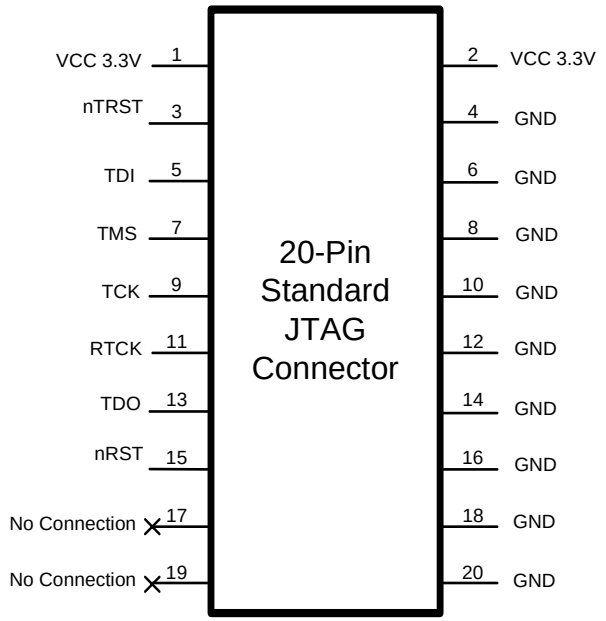
   PXA3xx processor family nTRST = JTAG nTRST AND nSYSTEM_RESET

2. The processor nRESET input should be connected to the JTAG nSRST ANDed with the Main System RESET

   PXA3xx processor family nRESET = JTAG nSRST AND nSYSTEM_RESET

These connections ensure the processor core and JTAG scan chain are both reset to known states when the reset button is used to reset the system.

Figure 12: Standard 20-pin JTAG Connector Pin Assignments



Note: Pin 11 is not connected in a typical PXA3xx processor family JTAG implementation.

9.5.1 Pull-up Resistors

The IEEE 1149.1 standard requires that TDI, TMS, and nTRST have internal pull-up resistors. The TDI and TMS pins can remain unconnected for normal operation and over driven to use the JTAG features. Marvell recommends that TCK be tied low for cases where JTAG is not used.

9.6 Layout Notes

The JTAG signals can run up to 13 MHz and are not particularly fast. However, use care to follow good design practices when routing the signals from the JTAG connector to the PXA3xx processor family.

9.7 Operation

For information on using and programming the PXA3xx processor family JTAG debug port, refer to the debug recommendations provided in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 17: JTAG”.

If a JTAG port is to be included in a design, ensure the JTAG debug port is correctly implemented. Consult the debug recommendations before completing the board design.

10 DDR Memory Interface (EMPI Bus)

This section describes guidelines for connecting the PXA3xx processor family External Memory Pin Interface (EMPI) to external DDR memory. Be sure to reference vendor-specific examples of schematics and timing diagrams for SDRAM devices.

All EMPI memory device types are discussed in this section, which includes design considerations for DDR devices as discussed in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual*, “Section 1: Dynamic Memory Controller”.

10.1 Overview

The EMPI for the PXA3xx processor family supports the following:

- DDR SDRAM Clock
  - Up to 30 MTps during ring oscillator (S0/D0CS/C0) mode
  - 130 MTps during run/turbo (S0/D0/C0) mode

Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual*, “Section 1: Dynamic Memory Controller” for the physical addresses used to configure and map the external memory.

The SDRAM must always reside on the EMPI bus and must always be controlled by the Dynamic Memory Controller (DMC). The maximum supported configurations on the EMPI bus is as follows:

- PXA30x Processor and PXA31x Processor DDR SDRAM maximum bus width is 16-bits (x16)
- PXA32x DDR SDRAM maximum bus width is 32-bits (x32)

Note The above configurations describe only the memory devices on the EMPI bus. Any allowable memory devices may be placed on the data-flash interface (DFI) without regard to the memory devices on the EMPI bus.

10.2 Signals

See Table 17 for the list of EMPI signals.

Table 17: PXA30x DDR SDRAM I/O Signals

Signal Name	Direction	Polarity	Description
MD<31:0>	Bidirectional	N/A	Bidirectional data for external static and dynamic memory. **NOTE:** PXA30x Processor and PXA31x Processor only use MD<15:0>
MA<15:11> MA<9:0> SDMA10	Output	N/A	Output address to all memory types **NOTE:** MA<15:14> are always connected to DDR BA<1:0> signals regardless of how many row/col addresses used.

Table 17: PXA30x DDR SDRAM I/O Signals (Continued)

Signal Name	Direction	Polarity	Description
DQM<3:0>	Output	Active High	Data byte mask control for external SDRAM. DQM<0> corresponds to MD<7:0> DQM<1> corresponds to MD<15:8> DQM<2> corresponds to MD<23:16> (PXA32x Processor Only) DQM<3> corresponds to MD<31:24> (PXA32x Processor Only) 0 = Do not mask out corresponding byte 1 = Mask out corresponding byte
DQS<3:0>	Bidirectional	N/A	DDR Strobe for DDR SDRAM. On reads from DRAM used by controller to latch data. On writes to DRAM used by DDR DRAM to latch data. Data latched on both rising and falling clock edges. DQM<0> corresponds to MD<7:0> DQM<1> corresponds to MD<15:8> DQM<2> corresponds to MD<23:16> (PXA32x Processor Only) DQM<3> corresponds to MD<31:24> (PXA32x Processor Only)
SDCLK<1:0>	Output	N/A	Differential output clock for DDR SDCLK<0> 0 degree phase clock for DDR SDCLK<1> 180 degree phase clock for DDR
nSDCS<1:0>	Output	Active Low	Chip selects for dynamic memory controller. All memory connected to a chip select must be either stacked memory or external memory. External/stacked memory on same chip select is not supported. nSDCS<1> DDR Chip Select One nSDCS<0> DDR Chip Select Two
SDCKE	Output	Active High	Output clock enable signals for external memory. SDCKE is for all dynamic memory chip selects. **NOTE:** The same clock enable is used for both Chip Selects.
nSDRAS	Output	Active Low	Row address strobe for DMC
nSDCAS	Output	Active Low	Column address strobe for dynamic memory controller
nSDWE	Output	Active Low	Write enable for dynamic memory controller
RCOMP_DDR	Output	—	Resistive compensation

10.3 Block Diagram

See Figure 13 and Figure 14 for illustrations of how the PXA3xx processor family memory controller signals are used to interface DDR SDRAM.

Figure 13: PXA30x and PXA31x EMPI Memory Configuration

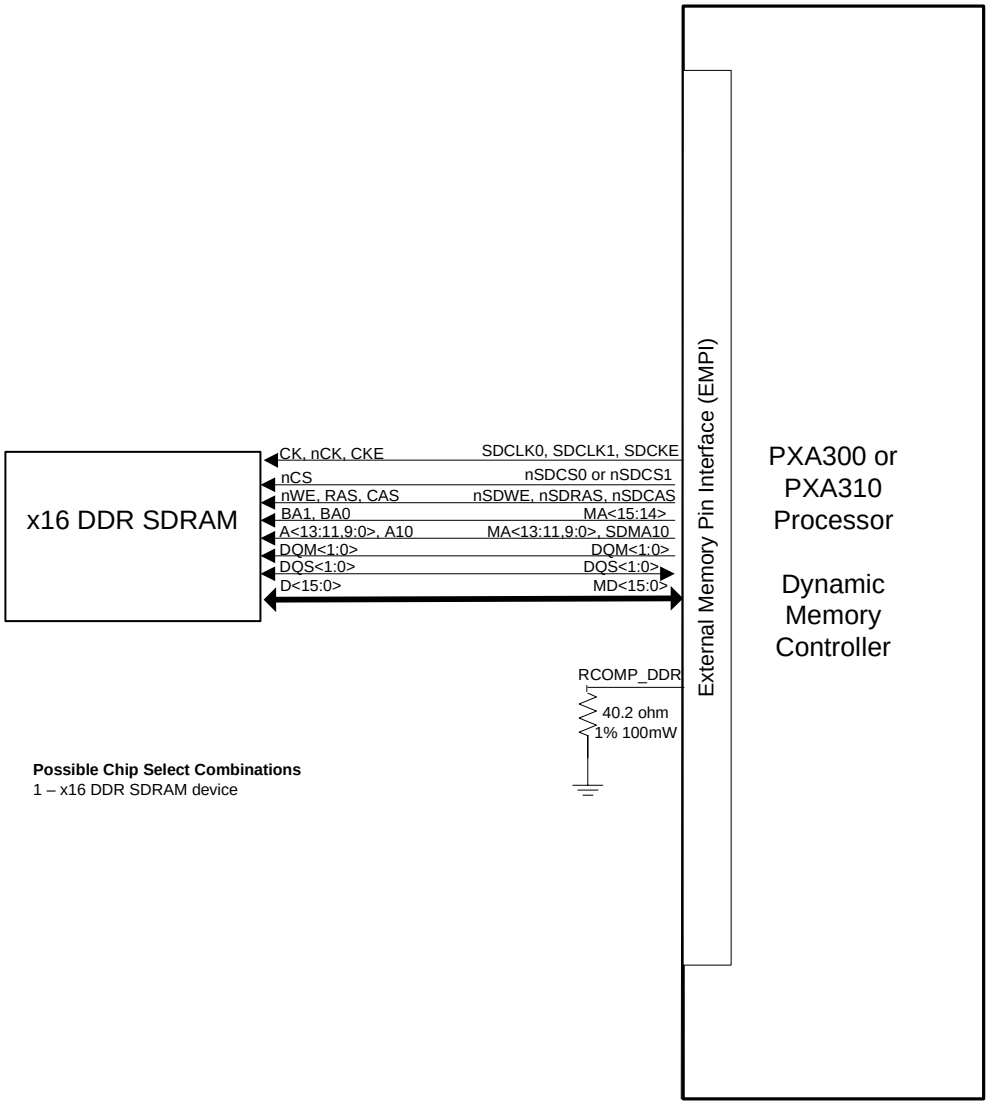
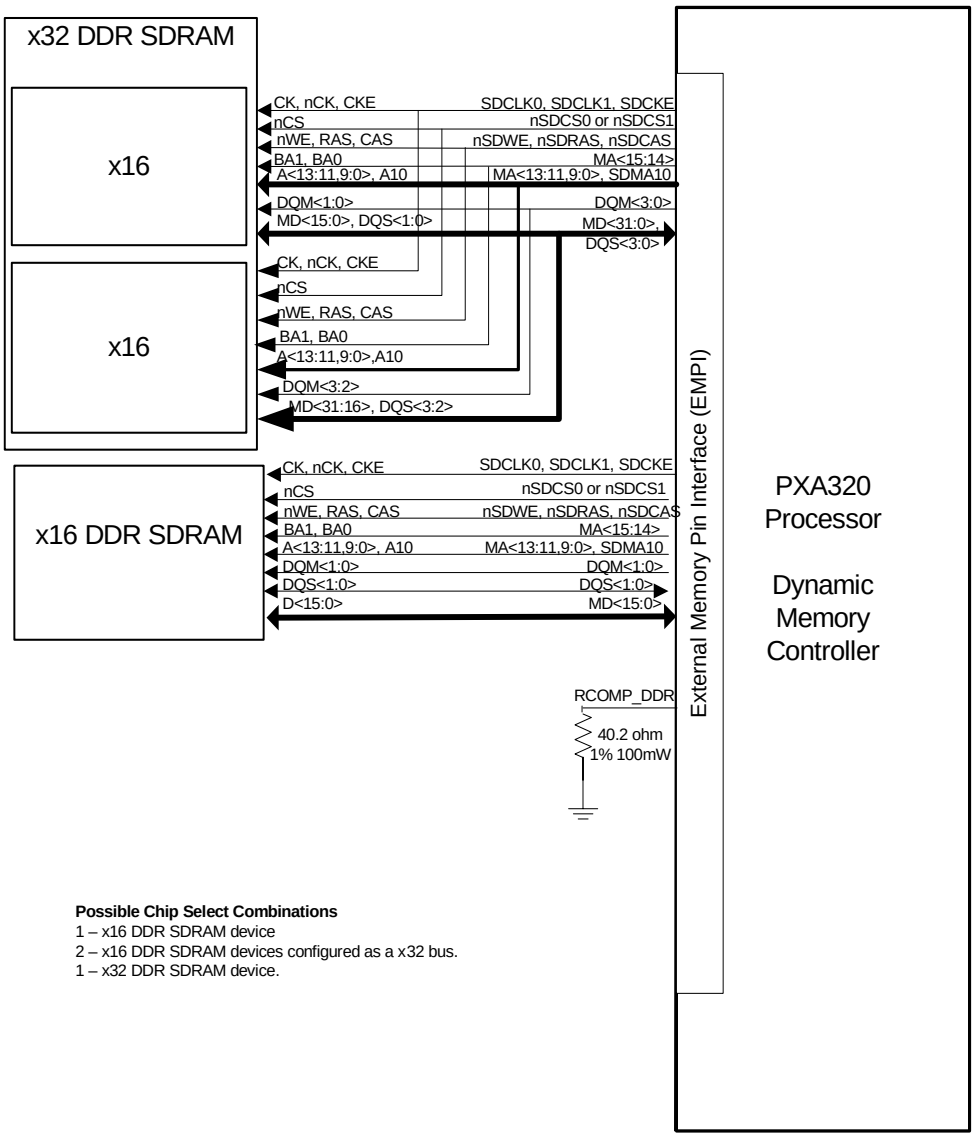


Figure 14: PXA32x EMPI Memory Configuration



10.4 EMPI Layout Notes

This section contains information for recommended trace lengths, size, and routing guidelines for the PXA3xx processor family EMPI bus.

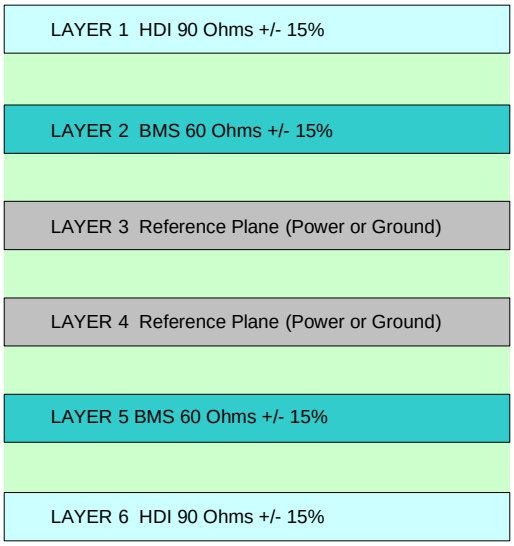
10.4.1 General DDR Routing Recommendations

These recommendations are based on the following assumptions:

1. There should not be stubs on any signal groups.
2. All high-density interconnect (HDI) traces target Impedance is 90 Ω +/- 15%.

3. All Buried Micro Strip (BMS) traces target Impedance is 60 Ω +/- 15%.
4. Spacing recommendations given are edge to edge from any other signal.
5. DDR requires 40.2 Ω 1%, 100 mW resistor connected to GND on the RCOMP_DDR pin. 40.2 Ω is a standard value for 1% tolerance resistors.
6. The RCOMP_DDR trace from package ball to component pad should be as short as possible, < 0.100 inches, and as wide as possible, >= 0.010 inches.
7. Follow DDR Power delivery guidelines as stated in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*.
8. In actual form factor design it is not advisable to have through-hole vias on PXA3xx processor family and DDR balls for direct debug. However, with some known timing impact this should be investigated for debug platforms. DDR inner balls are perfect candidates for this, since through-hole vias on those balls do not infringe breakout.
9. Route signals over an unbroken reference plane. If reference is broken (only allowed in case of address/cmd lines, although not recommended) stitching caps of 1 µf need to be placed within 0.050 inches of the vias where the reference jump occurs.
10. DDR memories used on the boards MUST follow LP-DDR JEDEC spec to guarantee timing compliance.
11. Board bring up code must upload DDR RCOMP slew rate table as provided in example Marvell BSP releases.
12. If attempting to maintain DDR contents while the PXA3xx processor family is in deep sleep mode:
    a) The DDR VCC voltage supply must be separate from the VCC_MEM and VCC_MVT supplies.This allows the DDR to self-refresh when the PXA32x processor is almost entirely powered off.
    b) SDCKE, SDCLK0 and SDCLK1 must have 100 K Ω pull-down resistors to GND to keep the DDR in self-refresh mode while the PXA32x processor is in deep sleep mode.
13. Verify SDMA10 is connected to A10 on all DDR SDRAMs. If this signal is not connected, the DDR auto-precharge and refresh will not work properly.
14. A minimum of one reference plane for every two signal layers is required. Figure 15 shows an example of a six-layer PCB. In this example, one full power plane is required for the DDR interface. The same power plane can be used to route signals for other interfaces as long as it acts as an unbroken reference plane for DDR signals. To avoid crosstalk, avoid parallel routing between the HDI (high-density interconnect) and BMS (buried microstrip) layers. Route the HDI layers perpendicular to the BMS layers.
15. Maximum loading is four loads (see Table 30)

Figure 15: Six Layer Board Stack Up Example



10.4.2 Routing Recommendations for x32 - 1 Device (PXA32x only)

Figure 16: Signal: MD<31:0>, DQM<3:0>

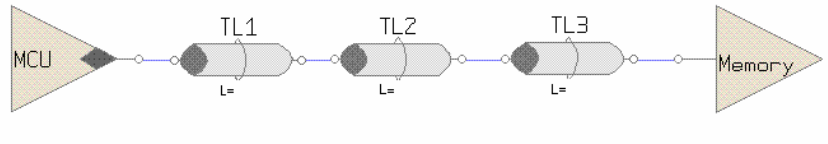


Table 18: MD<31:0> and DQM<3:0> Signals Routing Recommendations for x32 - 1 Device

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.3”	3 mils	3 mils	HDI
TL2	0.6”	4 mils	4 mils	Buried Micro Strip
TL3	0.3”	3 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 6

- Total length of (TL1+TL2+TL3) Max = 1.1”
- Refer to matching rules for length matching requirement between MD and DQS.

Figure 17: Signal: DQS<3:0>

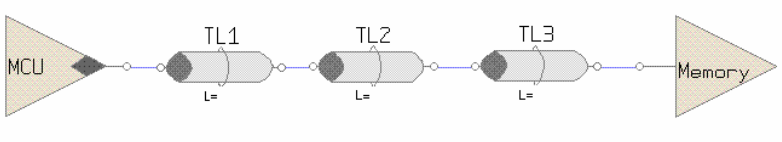


Table 19: DQS<3:0> Signals Routing Recommendations for x32 - 1 Device

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.3”	3 mils	4 mils	HDI
TL2	0.6”	4 mils	5 mils	Buried Micro Strip
TL3	0.3”	3 mils	4 mils	HDI

- Recommended maximum via count, including breakout via = 6
- Total length of (TL1+TL2+TL3) Max = 1.1”
- Refer to matching rules for length matching requirement between DQS and SDCLK.
- Refer to matching rules for length matching requirement between DQS and MD.

10.4.2.1 Length Matching between DQS and MD for x32 - 1 Device

There are four DQS byte lanes: DQS<0>, DQS<1>, DQS<2> and DQS<3>.

- Every data signal between MD<7:0> and DQM<0> should be length matched to DQS<0>.
- Every data signal between MD<15:8> and DQM<1> should be length matched to DQS<1>.
- Every data signal between MD<23:16> and DQM<2> should be length matched to DQS<2>.
- Every data signal between MD<31:24> and DQM<3> should be length matched to DQS<3>.

There is only length matching required within each byte lane. There is no signal length matching required outside the byte lane. For example, any signal within MD<7:0> need not be length matched to DQS<1>.

Total length of MD and DQM = Total Length of DQS +/- 0.100”

Total length of DQS is TL1+TL2+TL3

Total length of MD is TL1+TL2+TL3

Figure 18: Signal: SDCLK<1:0>

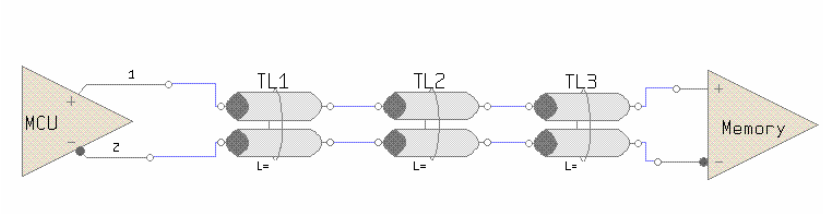


Table 20: SDCLK<1:0> Signals Routing Recommendations for x32 - 1 device

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Intra pair Spacing	Layer
TL1 (Breakout)	0.4”	3 mils	4 mils	3 mils	HDI
TL2	0.8”	4 mils	7 mils	4 mils	BMS
TL3	0.5”	3 mils	4 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 6
- Total length of (TL1+TL2+TL3) Max = 1.6"
- Length matching within the same signal:
  - Total length of SDCLK<0> = Total length of SDCLK<1> +/- 20 mils
- Refer to matching rules for length matching requirement between SDCLK and CMD; SDCLK and DQS.

10.4.2.2 Length Matching between SDCLK and DQS for x32 - 1 Device

All DQS signals (DQS<3:0>) should meet this requirement:

- (Total length of DQS) Max = Total Length of SDCLK<1:0>
- (Total length of DQS) Min = Total Length of SDCLK<1:0> - 1.0”

Total length equals trace from MCU Pin to Memory Pin, including HDI and Buried Micro Strip.

Figure 19: Signal: Command (including Address) and Control

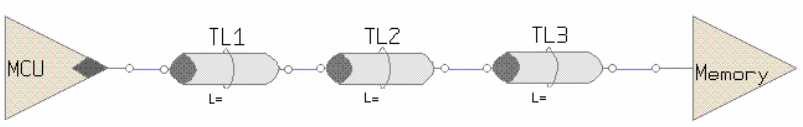


Table 21: Command and Control Signals Routing Recommendations for x32 - 1 Device

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.4”	3 mils	3 mils	HDI
TL2	0.9”	4 mils	4 mils	Buried Micro Strip
TL3	0.4”	3 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 8
- Total length of (TL1+TL2+TL3) Max = 1.7"
- Refer to matching rules for length matching requirement between CMD and SDCLK.

10.4.2.3 Length Matching between SDCLK and CMD for x32 - 1 Device

- Total length of CMD/Control = Total Length of SDCLK<1:0> +/- 0.12"
- Total length = TL1+TL2+TL3.

10.4.3 Routing Recommendations for x32 - 2 (x16) Devices (PXA32x only)

Figure 20: Signal: MD<31:0>, DQM<3:0>

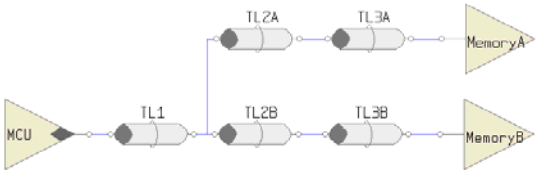


Table 22: MD<31:0> and DQM<3:0> Signals Routing Recommendations for x32 - 2 (x16) devices

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.3”	3 mils	3 mils	HDI
TL2A and TL2B	0.7”	4 mils	4 mils	Buried Micro Strip
TL3A and TL3B	0.2”	3 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 8
- Total length of (TL1+TL2A+TL3A) Max = 1.1" and
- Total length of (TL1+TL2B+TL3B) Max = 1.1"
- Length Matching within MD (applies for all 32 MD signals and 4 DQM signals)

Length of (TL2A+TL3A) = Length of (TL2B+TL3B) +/- 0.1"

- Refer to matching rules for length matching requirement between MD and DQS.

Figure 21: Signal: DQS<3:0>

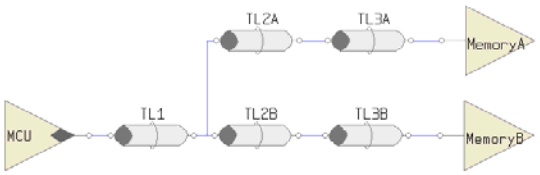


Table 23: DQS<3:0> Signals Routing Recommendations for x32 - 2 (x16) devices

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.3”	3 mils	4 mils	HDI
TL2A and TL2B	0.7”	4 mils	5 mils	Buried Micro Strip
TL3A and TL3B	0.2”	3 mils	4 mils	HDI

- Recommended maximum via count, including breakout via = 8
- Total length of (TL1+TL2A+TL3A) Max = 1.1" and
- Total length of (TL1+TL2B+TL3B) Max = 1.1"
- Length Matching within DQS (applies to all 4 DQS Signals):
  Length of (TL2A+TL3A) = Length of (TL2B+TL3B) +/- 0.1"
- Refer to matching rules for length matching requirement between DQS and SDCLK.
- Refer to matching rules for length matching requirement between DQS and MD/DQM.

10.4.3.1 Length Matching between DQS and MD/DQM for x32 - 2 (x16) Devices

There are four DQS byte lanes: DQS<0>, DQS<1>, DQS<2> and DQS<3>.

- Every data signal between MD<7:0> and DQM<0> should be length matched to DQS<0>.
- Every data signal between MD<15:8> and DQM<1> should be length matched to DQS<1>.
- Every data signal between MD<23:16> and DQM<2> should be length matched to DQS<2>.
- Every data signal between MD<31:24> and DQM<3> should be length matched to DQS<3>.

There is only length matching required within each byte lane. There is no signal length matching required outside the byte lane. For example, any signal within MD<7:0> need not be length matched to DQS<1>.

Total length of each branching in MD/DQM = Total Length of the corresponding DQS +/-0.100"

Total length of MD/DQM (TL2A+TL3A) = Total length of DQS (TL2A+TL3A) +/- 0.1"

Total length of MD/DQM (TL2B+TL3B) = Total length of DQS (TL2B+TL3B) +/- 0.1"

Figure 22: Signal: SDCLK<1:0>

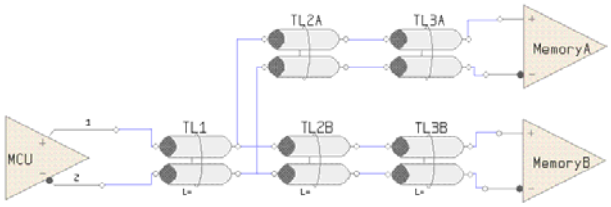


Table 24: SDCLK<1:0> Signals Routing Recommendations for x32 - 2 (x16) Devices

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Intra pair Spacing	Layer
TL1 (Breakout)	0.4”	3 mils	4 mils	3 mils	HDI
TL2	0.8”	4 mils	7 mils	4 mils	BMS
TL3	0.5”	3 mils	4 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 6
- Total length of (TL1+TL2A+TL3A) Max = 1.6"
- Length matching within the same signal:
  Total length of SDCLK<0> = Total length of SDCLK<1> +/- 20 mils
  - (TL1+TL2A+TL2B) of SDCLK0 = (TL1+TL2A+TL2B) of SDCLK1+/- 20 mils

  Total length of (TL2A+TL3A) = Total length of (TL2B+TL3B) +/- 25 mils
- Refer to matching rules for length matching requirement between SDCLK and DQS.
- Refer to matching rules for length matching requirement between SDCLK and CMD.

10.4.3.2 Length Matching between SDCLK and DQS for x32 - 2 (x16) Devices

- All DQS signals (DQS<3:0>) should meet this requirement
  (Total length of DQS) Max = Total Length of SDCLK<1:0>
  (Total length of DQS) Min = Total Length of SDCLK<1:0> -1.0"
- Total length equals trace from MCU Pin to Memory Pin, including HDI and Buried Micro Strip

Figure 23: Signal: Command (including Address) and Control

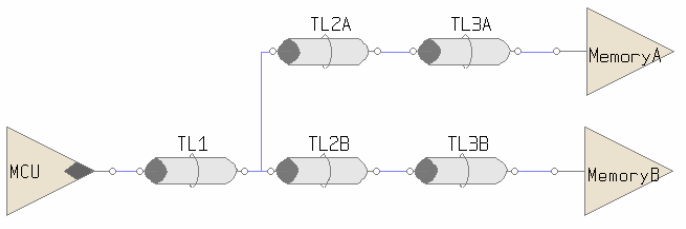


Table 25: Command and Control Signals Routing Recommendations for x32 - 2 (x16) Devices

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.4”	3 mils	3 mils	HDI
TL2A and TL2B	0.9”	4 mils	4 mils	Buried Micro Strip
TL3A and TL3B	0.4”	3 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 12
- Total length of (TL1+TL2A+TL3A) Max = 1.7"
- Refer to matching rules for length matching requirement between CMD and SDCLK.

10.4.3.3 Length Matching between SDCLK and CMD for x32 - 2 (x16) Devices

Total length of CMD/Control = Total Length of SDCLK<1:0> +/- 0.12"

10.4.4 Routing Recommendation for 4 - x16 Devices (PXA32x Only)

Figure 24: Signal: DQS<3:0>, DQ<31:0>, DQM<3:0, nSDCS<1:0>

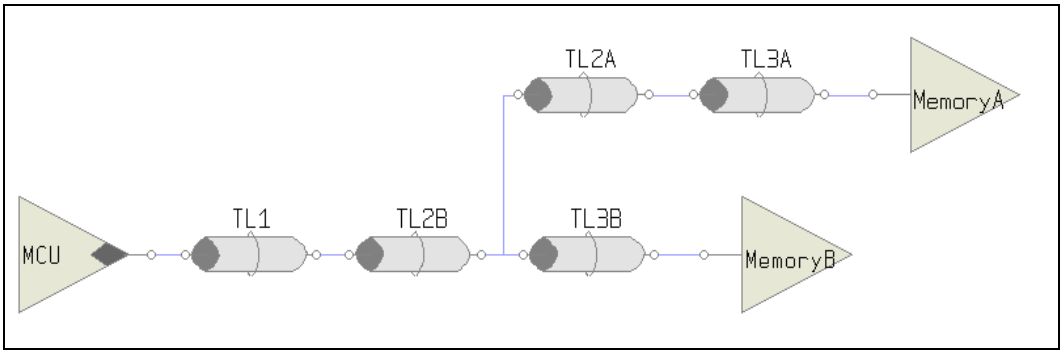


Table 26: MD<31:0>, DQS<3:0>, DQM<3:0> and nSDCS<1:0> Signals Routing Recommendations for 4 - x16 devices

Signal Group	Trace	Max Length	Trace Width (Min)	Spacing from any other signal (Min)	Layer
MD<31:0> DQM<3:0> nSDCS<1:0>	TL1	0.2"	3 mills	3 mills	HDI
	TL2A and TL2B	0.6"	4 mills	5 mills	BMS
	TL3A and TL3B	0.1"	3 mills	3 mills	HDI
DQS<3:0>	TL1	0.2"	3 mills	4 mills	HDI
	TL2A and TL2B	0.6"	4 mills	7 mills	BMS
	TL3A and TL3B	0.1"	3 mills	4 mills	HDI

Figure 25: MA<15:0>, SDMA10, nSDRAS, nSDCAS, nSDWE, SDCKE

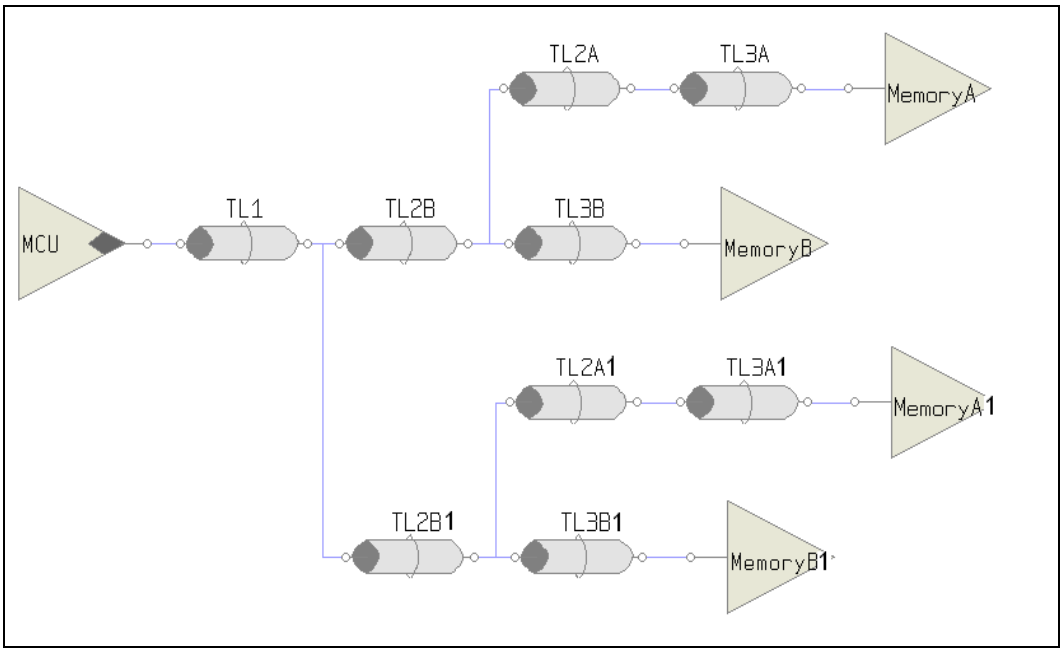


Table 27: MA<15:0>, SDMA10, nSDRAS, nSDCAS, nSDWE, SDCKE Signals Routing Recommendations for 4 - x16 devices

Signal Group	Trace	Max Length	Trace Width (Min)	Spacing from any other signal (Min)	Layer
Ma<15:0> nSDRAS nSDCAS nSDCKE	TL1	0.5"	3 mills	3 mills	HDI
	TL2B and TLB1	0.7"	4 mills	4 mills	BMS
	TL2A and TL32A1	0.6"	4 mills	4 mills	BMS
	TL3A,TL3B,TL3A1,TL3B1	0.1"	3 mills	3 mills	HDI

Figure 26: SDCLK<1:0>

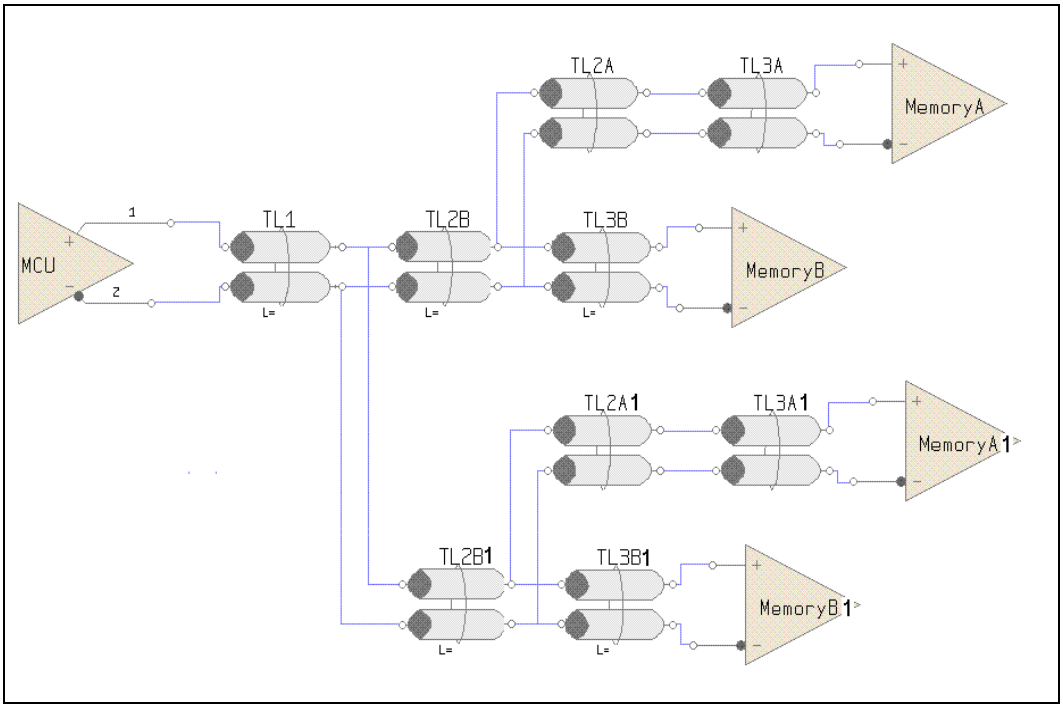


Table 28: SDCLK<1:0> Signal Routing Recommendations for 4 - x16 devices

Signal Group	Trace	Max Length	Trace Width (Min)	Spacing from any other signal (Min)	Layer
SDCLK<1:0>	TL1	0.1"	3 mills	3 mills	HDI
	TL2B and TLB1	0.7"	4 mills	4 mills	BMS
	TL2A and TL32A1	0.5"	3 mills	3 mills	BMS
	TL3A,TL3B,TL3A1,TL3B1	0.1"	3 mills	3 mills	HDI

10.4.5 Routing Recommendations for x16 - 2 Devices (PXA32x only)

Refer to Table 29 for x16 - 2 devices routing recommendations.

Table 29: Routing Recommendations for x16 - 2 devices

Signal Group	Where to refer for Recommendation
Address/Command Signal	Refer to x32 - 2 Device Address/Command Guideline
Data/Data Mask Signal	Refer to x32 - 1 Device Data Signal Guidelines
DQS Signal	Refer to x32 - 1 Device Data Signal Guidelines
Clock Signal	Refer to x32 - 2 Device Clock Signal Guidelines
Control Signal	Refer to x32 - 1 Device Control Signal Guidelines

10.4.6 Routing Recommendations for x16 - 1 Device

Table 30 shows the number of loads on each EMPI pin.

Table 30: Number of EMPI Pin Loads

Load on EMPI Bus/ Configuration	x32 - 2 Devices	x32 - 2 Devices	x16 - 1 Devices
Address/Command Signal	2 loads	1 load	2 loads
Data Signal	2 loads	1 load	1 load
Clock Signal	2 loads	1 load	2 loads
Control Signal	1 load	1 load	1 load

Figure 27: Signal: MD<15:0>, DQM<1:0>

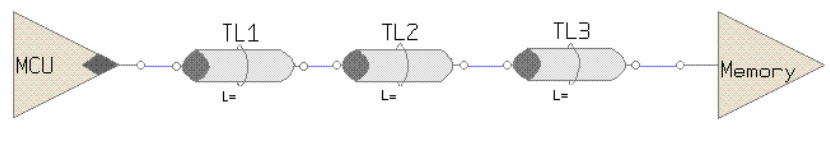


Table 31: MD<15:0> and DQM<1:0> Signals Routing Recommendations for x16 - 1 Device

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.3”	3 mils	3 mils	HDI
TL2	0.6”	4 mils	4 mils	Buried Micro Strip
TL3	0.3”	3 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 6
- Total length of (TL1+TL2+TL3) Max = 1.1”
- Refer to matching rules for length matching requirement between MD and DQS.

Figure 28: Signal: DQS<1:0>

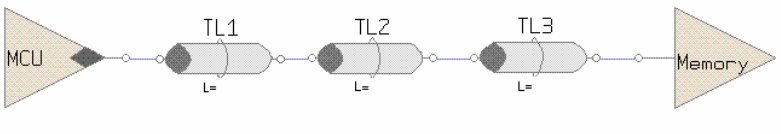


Table 32: DQS<3:0> Signals Routing Recommendations for x16 - 1 Device

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.3”	3 mils	4 mils	HDI
TL2	0.6”	4 mils	5 mils	Buried Micro Strip
TL3	0.3”	3 mils	4 mils	HDI

- Recommended maximum via count, including breakout via = 6
- Total length of (TL1+TL2+TL3) Max = 1.1”
- Refer to matching rules for length matching requirement between DQS and SDCLK.
- Refer to matching rules for length matching requirement between DQS and MD.

10.4.6.1 Length Matching between DQS and MD for x16 - 1 Device

There are two DQS byte lanes: DQS<0> and DQS<1>.

- Every data signal between MD<7:0> and DQM<0> should be length matched to DQS<0>.
- Every data signal between MD<15:8> and DQM<1> should be length matched to DQS<1>.

There is only length matching required within each byte lane. There is no signal length matching required outside the byte lane. For example, any signal within MD<7:0> need not be length matched to DQS<1>.

Total length of MD and DQM = Total Length of DQS +/- 0.100”

Total length of DQS is TL1+TL2+TL3

Total length of MD is TL1+TL2+TL3

Figure 29: Signal: SDCLK<1:0>

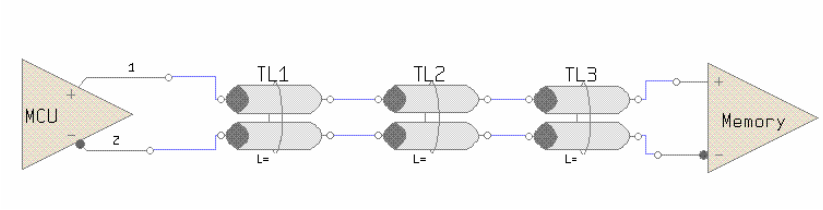


Table 33: SDCLK<1:0> Signals Routing Recommendations for x16 - 1 device

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Intra pair Spacing	Layer
TL1 (Breakout)	0.4”	3 mils	4 mils	3 mils	HDI
TL2	0.8”	4 mils	7 mils	4 mils	BMS
TL3	0.5”	3 mils	4 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 6
- Total length of (TL1+TL2+TL3) Max = 1.6"
- Length matching within the same signal:
  Total length of SDCLK<0> = Total length of SDCLK<1> +/- 20 mils
- Refer to matching rules for length matching requirement between SDCLK and CMD; SDCLK and DQS.

10.4.6.2 Length Matching between SDCLK and DQS for x16 - 1 Device

All DQS signals (DQS<1:0>) should meet this requirement:

- (Total length of DQS) Max = Total Length of SDCLK<1:0>
- (Total length of DQS) Min = Total Length of SDCLK<1:0> - 1.0”

Total length equals trace from MCU Pin to Memory Pin, including HDI and Buried Micro Strip.

Figure 30: Signal: Command (including Address) and Control

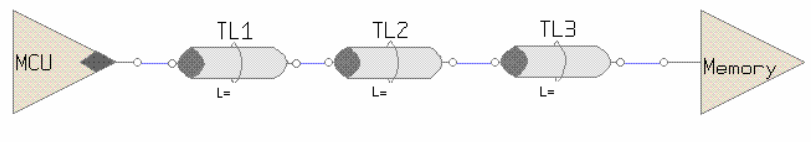


Table 34: Command and Control Signals Routing Recommendations for x16 - 1 Device

Trace	Max Length	Trace Width	Spacing between this Signal trace and any other Signal trace	Layer
TL1 (Breakout)	0.4”	3 mils	3 mils	HDI
TL2	0.9”	4 mils	4 mils	Buried Micro Strip
TL3	0.4”	3 mils	3 mils	HDI

- Recommended maximum via count, including breakout via = 8
- Total length of (TL1+TL2+TL3) Max = 1.7"
- Refer to matching rules for length matching requirement between CMD and SDCLK.

10.4.6.3 Length Matching between SDCLK and CMD for x16 - 1 Device

- Total length of CMD/Control = Total Length of SDCLK<1:0> +/- 0.12"
- Total length = TL1+TL2+TL3.

11 Data Flash Interface (DFI)

This section describes guidelines for connecting the PXA3xx processor family Data Flash Interface (DFI) to external memory. All DFI memory device types are discussed in this section, which includes design considerations for DFI devices covered in the following sections of the *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual*.

- Section 2 - Static Memory Controller (SMC)
- Section 3 - NAND Flash Controller (NFC)

11.1 Overview

The DFI bus for the PXA3xx processor family supports the following:

- NAND Flash
- NOR Flash (both Synchronous and Asynchronous AA/D muxed)
- SRAM, including variable latency I/O (VLIO) devices
- PC Card (PCMCIA) / Compact Flash (PXA32x only)
- Non-AA/D muxed devices are supported with the use of external latches to latch the address signals

Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual*, “Section 2: Static Memory Controller” for the physical addresses used to configure and map the external memory.

11.2 DFI Signals

See Table 35 for the complete list of DFI-specific signals from the PXA3xx processor family DFI controller. Table 35 details how various memory configurations (SRAM, VLIO, PC-Card, 8/16 NAND, 16-bit NOR) connect to the DFI signals.

Table 35: PXA3xx Processor Family DFI Signals

Signal Name	Direction	Polarity	Description
Arbitrated SMC and NFC Signals			
DF_IO<15:0>	Bidirectional	N/A	Data/Address bus. Carries data and multiplexed address information for NAND and Static Memory Controller accesses. The format of the data and address depends on the bus configuration and whether the access is to a flash or a static memory device.
DF_ALE_nWE1 DF_ALE_nWE2 (PXA32x Only)	Output	Active Low	NAND Flash Controller: Address Latch Enable signal (DF_ALE). Static Memory Controller: Write Enable signal (DF_nWE). **NOTE:** PXA32x has two signals, DF_ALE_nWE1 and DF_ALE_nWE2, which have the same DF_ALE and DF_nWE functionality.
DF_CLE_nOE	Output	Active Low	NAND Flash Controller: Command Latch Enable signal (DF_CLE). Static Memory Controller: Output Enable signal (DF_nOE).
Non-Arbitrated Static Memory Controller Signals			

Table 35: PXA3xx Processor Family DFI Signals (Continued)

Signal Name	Direction	Polarity	Description
DF_SCLK_E	Output	N/A	**Output Clock** This is generated by the combination of the SMCFS bit in the Application Subsystem Clocks Configuration Register (ACCR) and the DF_CLKDIV bit in the Clock Configuration Register (MEMCLKCFG).
nCS<3:0>	Output	Active Low	**Chip Selects** • nCS<1> and nCS<0> (PXA30x and PXA31x Only) • nCS<3> and nCS<2> (Configured on GPIO pins) •PXA32x - GPIO<4> and GPIO<3> •PXA31x and PXA30x - GPIO<2> and GPIO<1>
nLUA	Output	Active Low	**Latch Upper Address** This signal can be used to latch the high order address bits of the cycle from the address bus in an external device.
nLLA	Output	Active Low	**Latch Lower Address** This signal can be used to latch the low order address bits of the cycle from the address bus in an external device.
nXCVREN	Output	Active Low	**External Transceiver Enable** This signal can be used to enable an external transceiver for SMC accesses. **NOTE:** For PXA30x and PXA31x nXCVREN corresponds to GPIO<0>. For PXA32x nXCVREN corresponds to nXCVREN (Dedicated).
DF_ADDR <3:0>	Output	Active High	**Low-order address bits** This can be used as the lowest four address bits during a burst transfer instead of the values in the lower address latch, this allows higher performance data transfer by avoiding most of the LLA cycles. **NOTE:** These 4 address signals are present no matter which addressing mode is used for flash, SRAM and VLIO accesses. Not used for CF Card accesses (PXA32x Only)
nBE<1:0>	Output	Active Low	**Data byte enable.** nBE<0> corresponds to DF_IO<7:0> nBE<1> corresponds to DF_IO<15:8> 0 = Do not mask out corresponding byte 1 = Mask out corresponding byte
RDY	Input	Active Low	**Variable latency VLIO signal for inserting wait states.** 0 = Busy 1 = Ready **NOTE:** For PXA30x and PXA31x RDY corresponds to GPIO<0>. For PXA32x RDY corresponds to GPIO<2>
DREQ	Input	Active Low	**DMA Request (PX32x Only)** External DMA request pin used by a companion chip. Positive edge triggered signal that notifies the DMA Controller of a Static Memory Controller access requests.
Non-Arbitrated NAND Flash Controller Signals			

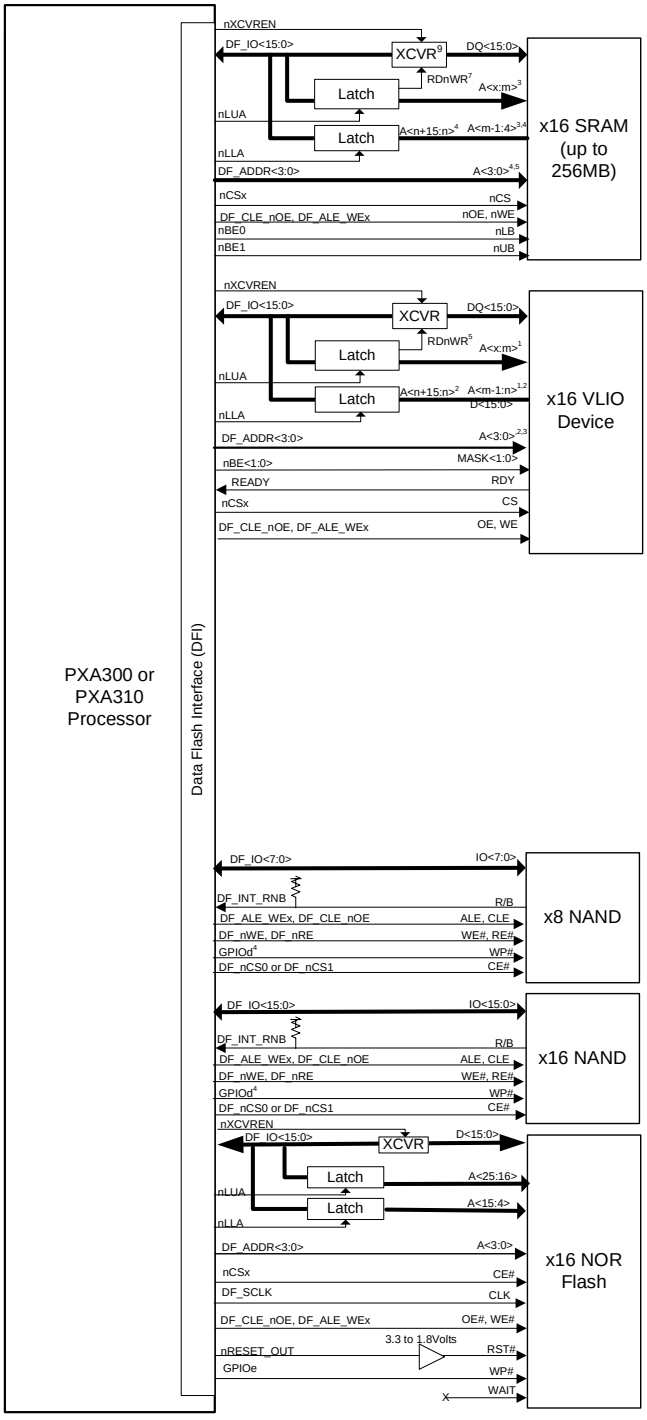
Table 35: PXA3xx Processor Family DFI Signals (Continued)

Signal Name	Direction	Polarity	Description
DF_nCS<1:0>	Output	Active Low	**Chip Selects** **NOTE:** For devices booting from NAND, nCS<0> must be used. For more information on supported boot configurations refer to the PXA3xx Boot ROM Reference Manual.
DF_nWE	Output	Active Low	**Write Enable**
DF_nRE	Output	Active Low	**Read Enable**
DF_INT_RnB	Input	Active Low	**Ready/Busy#** 0 = Busy 1 = Ready **NOTE:** Must be connected to the NAND device on nCS<0> when booting from a NAND device.

11.3 Block Diagram

See Figure 31 and Figure 32 for an illustration of how the PXA3xx processor family DFI memory controller signals are used to interface to NOR Flash, SRAM, VLIO, Compact Flash, and 8/16-bit NAND Flash.

Figure 31: PXA30x and PXA31x DFI Memory Configuration

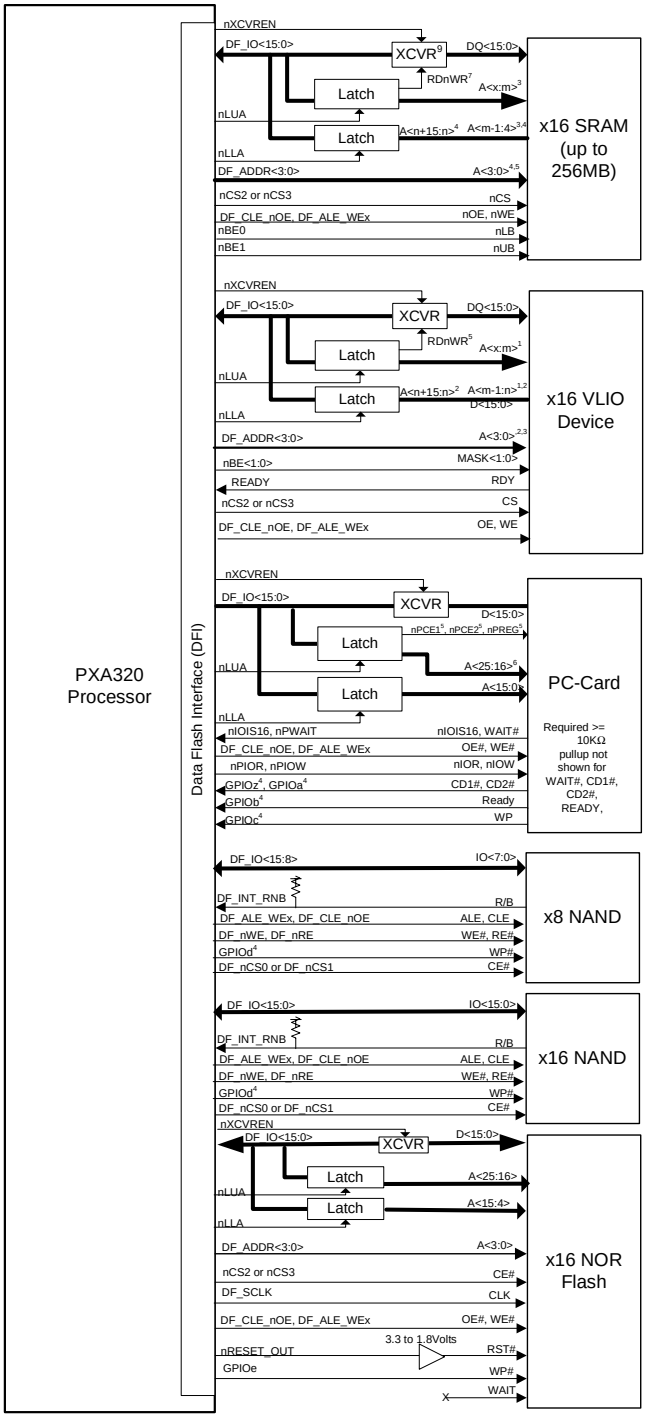


Possible Combination on DFI
nCS0, nCS1, nCS2, nCS3 can support any combination of NOR Flash, SRAM, and VLIO.

DF_nCS0, DF_nCS1 can support any combination of NAND

Notes:
1. 'm' variable is determined by CSADRCFGx[ADDRSPLIT] programmed value.
2. An and DF_ADDR<0> address signals will address either byte, half word, or word depending on CSADRCFGx[ADDRBASE] programmed value.
3. CSADRCFGx[ADDRCONFIG] must be set to 0b11 to use DF_ADDR<3:0> signals. This allows for fast addressing to lower address bits for SRAM and VLIO devices.
4. Transceiver is optional for NOR Flash, SRAM, and VLIO depending on bus loading.

Figure 32: PXA32x DFI Memory Configuration



Possible Combination on DFI
nCS2 and nCS3 can support an any combination of NOR Flash, SRAM, and VLIO with PC-Card and/or (16-bit NAND and/or 8-bit NAND)

Notes:
1. 'm' variable is determined by CSADRCFGx[ADDRSPLIT] programmed value.
2. An and DF_ADDR<0> address signals will address either byte, half word, or word depending on CSADRCFGx[ADDRBASE] programmed value.
3. CSADRCFGx[ADDRCONFIG] must be set to 0b11 to use DF_ADDR<3:0> signals. This allows for fast addressing to lower address bits for SRAM and VLIO devices.
4 GPIO signals must be configured within the GPIO controller and are not part of the DFI as shown.
5. CSARDCFG_P[ADDRCONFIG] must be set to 0b00 for PC-Card support.
   nPREG is latched during nLUA assertion as A26
   nPCE1 is latched during nLUA assertion as A29
   nPCE2 is latched during nLUA assertion as A30
   RDnWR is latched during nLUA assertion as A31
6. CSADRCFG_P[ADDRSPLIT] must be programmed to 0x1000.
7. Transceiver is optional for NOR Flash, SRAM, VLIO, and PC-Card depending on bus loading.
.

11.4 DFI Layout Notes

This section contains information for recommended trace lengths, size, and routing guidelines for the PXA3xx processor family DFI bus.

11.4.1 General DFI Routing Recommendations

Note

These recommendations are based on the following assumption:

- Maximum loading is four loads. Be careful to ensure even loading of data signals vs. address vs. clocks vs. control signals.

11.5 Modes of Operation Overview

Refer to the following subsections for description of the specific operations and signals including example schematics, timing diagrams, and layout notes for the PXA3xx processor family memory controller-supported memories:

- Section 11.5.1 — NOR Flash Operation
- Section — SRAM Operation
- Section 11.5.3 — Variable Latency Input/Output (VLIO) Operation
- Section 11.5.4 — PC Card (PCMCIA) Interface (PXA32x only)
- Section 11.5.5 — NAND Operation

Memory types are programmable through the memory-interface configuration registers. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual* for detailed information on the configuration registers.

11.5.1 NOR Flash Operation

NOR Flash must be attached to the DFI bus on the PXA3xx processor family. The static memory controller (SMC) is used to control NOR Flash. Latches are required for non-AA/D muxed NOR Flash devices to demultiplex the address/address/data bus. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for detailed information on the timing of signals used for NOR Flash accesses.

Note

Synchronous operation is only available when performing Flash accesses.

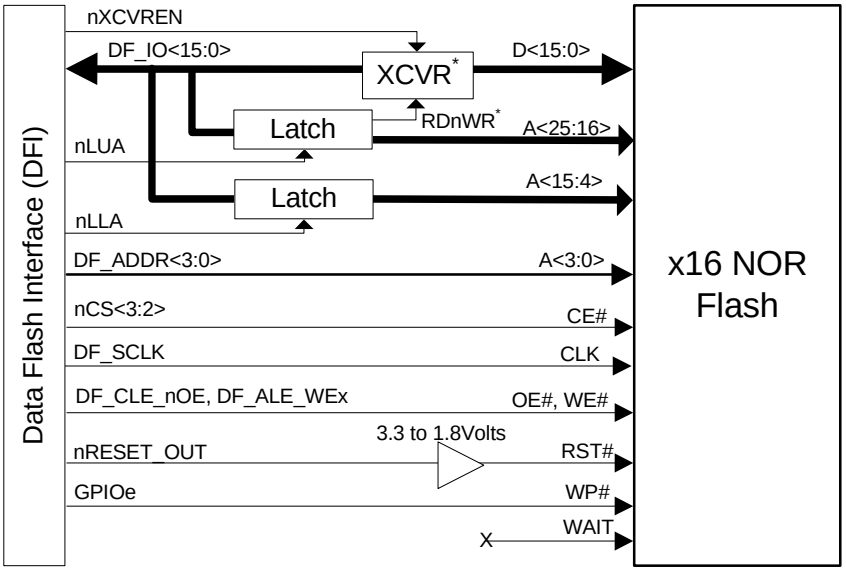
11.5.1.1 NOR Flash Block Diagram

Figure 33 is a block diagram depicting the connection of a 16 bit NOR Flash to the DFI. The particular configuration shown in Figure 33 connects to nCS2 or nCS3. All chip selects (nCS0, nCS1, nCS2, nCS3) can be used to connect SRAM to the PXA320 processor SMC. This example also shows the use of DF_ADDR<3:0>, which are used with burst operations. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual*, “Section 2: Static Memory Controller” for detailed information on the use of DF_ADDR<3:0>.

This example shows the address split between two latches, starting with address A<16>. However, this split point is programmable and is controlled by the CSADRCFGx registers. If the address split point is not configured to be A<16>, then the address lines from the SRAM must be connected to the

appropriate latch. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual*, “Section 2: Static Memory Controller” for detailed information.

Figure 33: NOR x16 Flash on Data Flash Interface



* Transceiver is optional for NOR Flash, SRAM, VLIO, and PC-Card depending on bus loading.

Note For Static Memory Conroller booting, use the nCS2 chip select.

11.5.2 SRAM Operation

SRAM must be attached to the DFI on the PXA3xx processor family. The static memory controller (SMC) is used to control SRAM. Latches are required for SRAM devices to demultiplex the address/address/data bus. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for detailed information on the timing of signals used to operate SRAM.

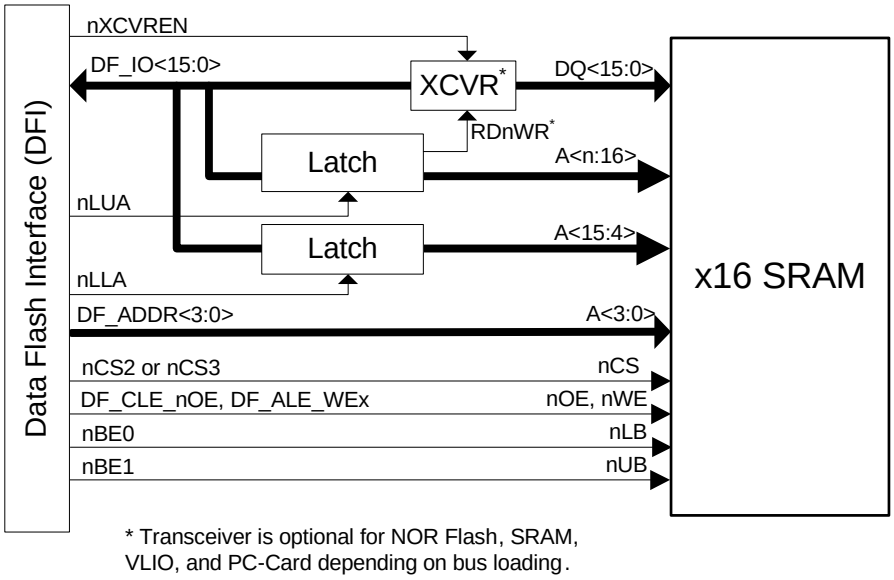
11.5.2.1 SRAM Block Diagram

See Figure 34 for illustration of the connection between a SRAM and the PXA3xx processor family memory controller. The particular configuration shown in Figure 34 connects to nCS2 or nCS3. All chip selects (nCS0, nCS1, nCS2, nCS3) can be used to connect SRAM to the PXA3xx processor family SMC. This example also shows the use of DF_ADDR<3:0>, which are used with burst operations. Refer to the to the *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory*

Controller Configuration Developers Manual, “Section 2: Static Memory Controller” for detailed information on the use of DF_ADDR<3:0>.

This example shows the address split between two latches, starting with address A<16>. However, this split point is programmable and is controlled by the CSADRCFGx registers. If the address split point is not configured to be A<16>, then the address lines from the SRAM must be connected to the appropriate latch. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual*, “Section 2: Static Memory Controller” for detailed information.

Figure 34: Block Diagram Connecting SRAM to nCS2 or nCS3



11.5.3 Variable Latency Input/Output (VLIO) Operation

When a companion chip is used as a variable-latency I/O device, its functionality is similar to that of an SRAM. The chip can insert a variable number of wait states through the use of the RDY pin.

VLIO devices must be attached to the PXA3xx processor family DFI bus. The static memory controller (SMC) control accesses to VLIO devices. Latches are required for non-AA/D muxed devices to demultiplex the address/address/data bus. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* for detailed information on the signal timing for VLIO accesses.

VLIO Interfaced to the Data Flash Interface (DFI)

See Figure 35 for an illustration of the signals and connections when connecting a VLIO companion chip to the PXA3xx processor family DFI bus. If the VLIO device does not support AA/D muxed operation (it has separate data and address pins), external latches are needed to latch the addresses for the VLIO device.

Note The RDY signal is only used when performing VLIO accesses.

Figure 35: VLIO Interface Block Diagram - AA/D Muxed Mode on DFI with External Latches

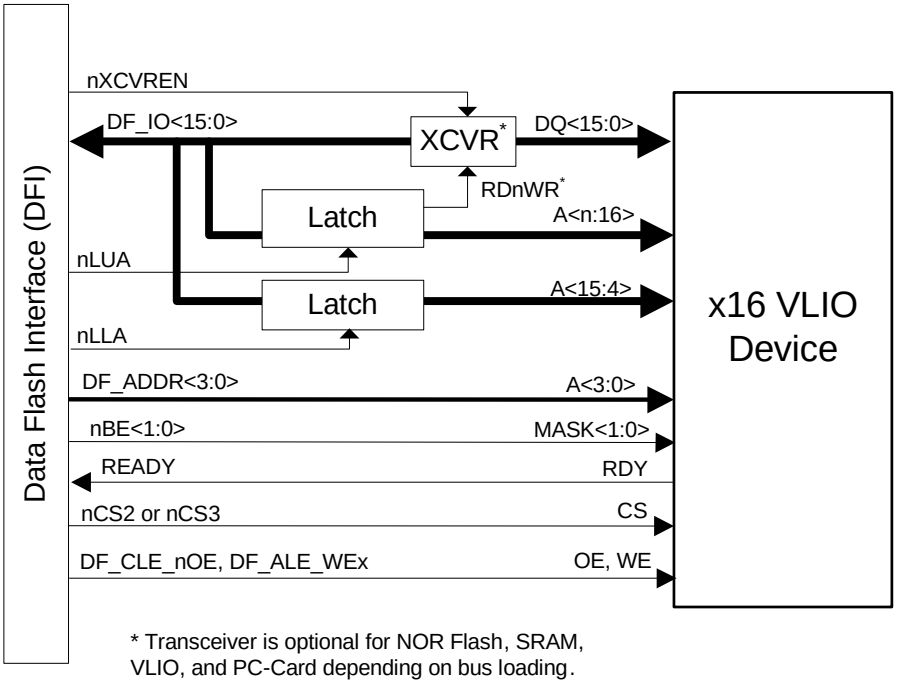
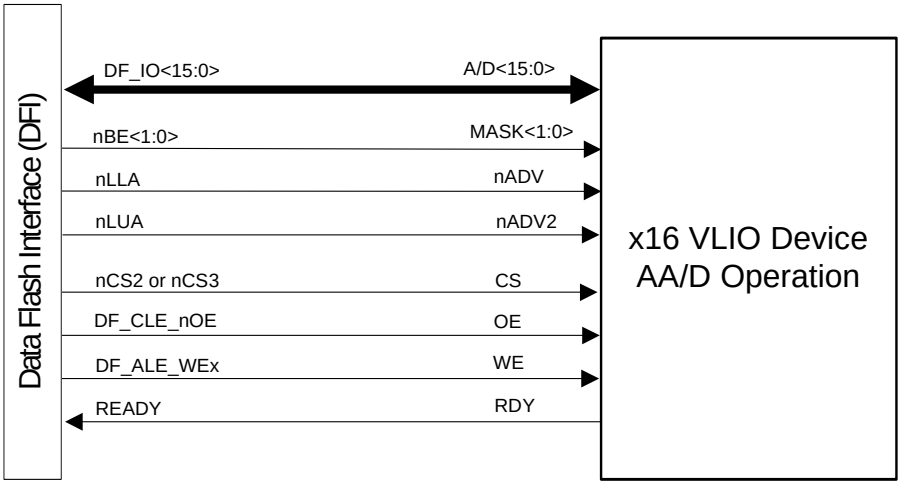


Figure 36: VLIO Interface Block Diagram - Operation on DFI with AA/D Compatible VLIO Device



11.5.4 PC Card (PCMCIA) Operation (PXA32x only)

The PXA32x process requires external glue logic to complete the 16-bit PC card socket interface. Only 1-socket solutions are possible.

Figure 37 shows a general solution for a PC-card configuration:

The pull-ups shown are included as specified in the *PC Card Standard, Volume 2, Electrical Specification, PCMCIA/JEITA*. Low-power systems must remove power from the pull-ups during sleep to avoid unnecessary power consumption.

GPIO or memory-mapped external registers control the reset of the 16-bit PC card interface, power selection (VCC and VPP), and drive enables. The INPACK# signal is not used.

Figure 37 shows the logical connections necessary to support hot-insertion capability:

For dual-voltage support, level-shifting buffers are required for all PXA32x input signals. Hot-insertion capability requires that each socket be electrically isolated from the remainder of the memory system. If hot-insertion capability is not required, then some of the logic shown in the following diagrams is eliminated.

Use software to set the MECR[NOS] and MECR[CIT] bits. MECR[NOS] indicates the number of sockets that the system supports, while MECR[CIT] is written when the card is in place. Input pins nPWAIT and nIOIS16 are three-stated until the card detect (CD) signal is asserted. To achieve this state, software programs the MECR[CIT] bit when a card is detected. If the MECR[CIT] is 0, the nPWAIT and nIOIS16 inputs are ignored.

11.5.4.1 PC Card Block Diagrams

This section describes how to interface a one-socket PC card adapter to the PXA32x memory controller.

Unknown behavior may result if connected to a PC card or CF card that is powered and directly attached to the memory bus at reset, depending on the states of the remaining card interface signals.

To prevent erroneous nPCE<2,1> assertions during reset, the board must have board-level 4.7 KΩ pull-ups.

Refer to the *Marvell® PXA32x Processor Electrical, Mechanical, and Thermal Specification* for additional information. The weak (50 KΩ nominal) on-chip pull-downs are released when the RDH bit is cleared. Subsequently, contentions through the two board-level pull-ups occur only during PC-card accesses.

External Logic for PC-Card Implementation Block Diagram

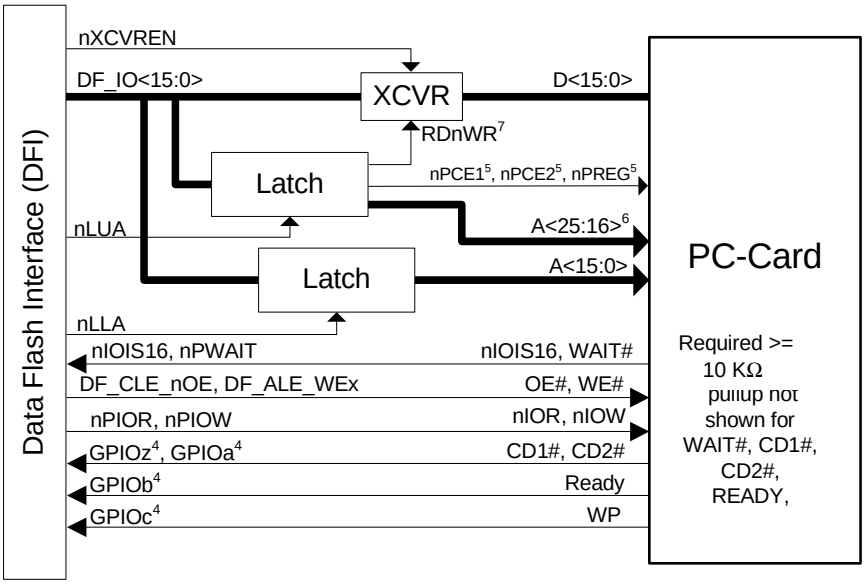
See Figure 37 for illustration of the minimal glue logic needed for a PC-card system. The illustration shows:

- Data transceivers
- Address buffers

Figure 37 does not show all level-shifting buffers that may be required to interface between the PC-Card and the DFI interface.

The transceivers are enabled by the PSKTSEL signal. The DIR pin of the transceiver is driven by the RDnWR pin. A GPIO is used for the three-state signal of the address and nPWE lines. These signals must be three-stated because they are used for memories other than the card interface. The Card Detect<1:0> signals are driven by the single device.

Figure 37: PC-Card Block Diagram - Operation in AA/D Muxed Mode



11.5.4.2 PC Card Layout Notes

Pull-up resisters shown in Figure 37 must be 10 KΩ or greater in value. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification* timing information.

11.5.5 NAND Flash Operation

NAND memories must be attached to the PXA3xx processor family DFI bus. The NAND Flash Controller (NFC) controls accesses to NAND memories. See and *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. II: Memory Controller Configuration Developers Manual*, “Section 3: NAND Flash Controller” for detailed information on the signals and registers used to operate NAND memories.

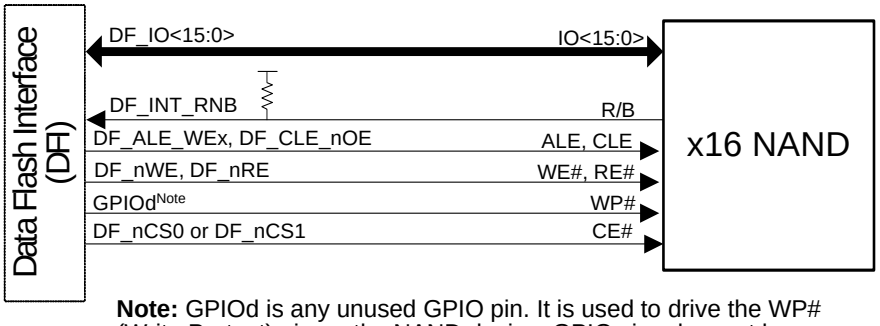
11.5.5.1 NAND Block Diagram

See Figure 38 and Figure 39 for illustration of the connection between NAND memory and the PXA3xx processor family memory controller.

Note PXA3xx processor family members using the MCP package must have VCC_DF connected to 1.8 V. NAND devices connected to the MCP versions are 1.8V.

Note The DF_INT_RNB pin must be connected to the READY/BUSY# pin on NAND device connected to DF_nCS0.

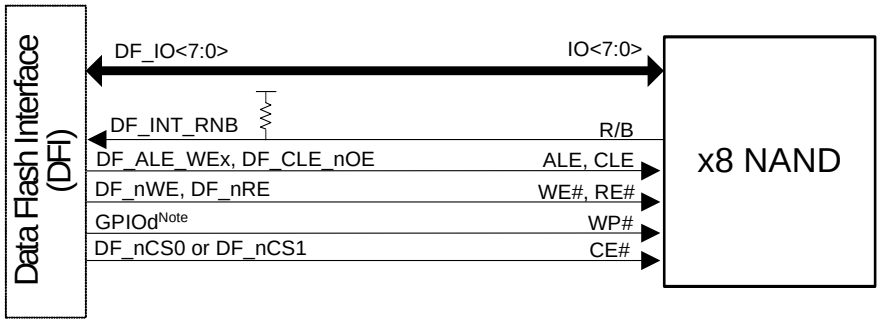
Figure 38: Block Diagram Connecting 16 bit NAND to Data Flash Interface



Note: GPIOd is any unused GPIO pin. It is used to drive the WP# (Write Protect) pin on the NAND device. GPIO signals must be configured within the GPIO controller and are not part of the DFI bus as shown. This signal needs an external 10K ohm pull-up resistor.

Note For NAND booting, use the nCS0 chip select.

Figure 39: Block Diagram Connecting 8-Bit NAND to Data Flash Interface



Note: GPIOd is any unused GPIO pin. It is used to drive the WP# (Write Protect) pin on the NAND device. GPIO signals must be configured within the GPIO controller and are not part of the DFI bus as shown. This signal needs an external 10K ohm pull-up resistor.

12 MultiMediaCard/SD/SDIO Card Controller

This section describes guidelines to interface the MultiMediaCard (MMC)/Secure Digital (SD)/Secure Digital IO (SDIO) controller of PXA3xx processor family to MMC/SD/SDIO sockets for use with MMC, SD, and SDIO card devices.

12.1 Overview

The PXA3xx processor family has two MMC/SD/SDIO controllers that perform the following tasks:

- Act as links between the software used to access the PXA3xx processor family and the MMC/SD/ SDIO devices
- Supports:
  - MultiMediaCard
  - Secure Digital
  - Secure Digital I/O
  - SPI communication protocols

The MMC/SD/SDIO controller supports:

- MMC, SD, and SDIO card systems
- Low-cost data storage and communications systems1

The PXA3xx processor family MMC/SD/SDIO controller is based on the standards outlined in these specifications:

- MultiMediaCard System Specification, Version 3.3.1
- SD Memory Card Specification, Version 1.1
- SDIO Card Specification, Version 1.0

The MMC/SD/SDIO controller supports the translation protocol from a standard MMC/SD/SDIO or serial peripheral interface (SPI) bus to the MMC/SD/SDIO card devices. The software used to access the PXA3xx processor family must indicate whether to use MMC, SD card, SDIO card, or SPI protocol to communicate with the MMC/SD/SDIO controller.

12.2 Signals

See Table 36 for the description of MMC/SD/SDIO card controller signal functions.

Table 36: Multimedia Card/SD/SDIO Card Controller Interface Signal Summary

Signal	Direction (MMC/SD/ SDIO Mode)	Direction (SPI Mode)	Description
MMC/SD/SDIO Card Controller #1			
MM1_CLK	Output	Output	**MultiMediaCard and SD Card Bus Clock**—MMC/SD/SDIO Controller #1

Table 36: Multimedia Card/SD/SDIO Card Controller Interface Signal Summary (Continued)

Signal	Direction (MMC/SD/ SDIO Mode)	Direction (SPI Mode)	Description
MM1_CMD	Bidirectional	Output	**MultiMediaCard Command**—MMC/SD/SDIO Controller #1 **MMC and SD:**bidirectional line for command and response tokens **SPI:**output for command and write data
MM1_DAT0	Bidirectional	Input	**MultiMediaCard Data 0**—MMC/SD/SDIO Controller #1 **MMC and SD:** Bidirectional line for read and write data. **SPI:** input for response token and read data
MM1_DAT1	Bidirectional	Input	**MultiMediaCard Data 1**—MMC/SD/SDIO Controller #1 **MMC and SD:** Bidirectional line for read and write data. **SDIO:**Signals an interrupt condition to the controller.
MM1_DAT2/ MM1_CS0	Bidirectional	Output	**MultiMediaCard Data 2**—MMC/SD/SDIO Controller #1 **MMC and SD:** Bidirectional line for read and write data. **MultiMediaCard Chip Select 0**—MMC/SD/SDIO Controller #1 **SPI:** Chip Select 0
MM1_DAT3/ MM1_CS1	Bidirectional	Output	**MultiMediaCard Data 3**—MMC/SD/SDIO Controller #1 MMC and SD: Bidirectional line for read and write data. **MultiMediaCard Chip Select 1**—MMC/SD/SDIO Controller #1 **SPI:**chip select 1
MMC/SD/SDIO Card Controller #2			
MM2_CLK	Output	Output	**MultiMediaCard and SD Card Bus Clock**—MMC/SD/SDIO Controller #2
MM2_CMD	Bidirectional	Output	**MultiMediaCard Command**—MMC/SD/SDIO Controller #2 **MMC and SD:**Bidirectional line for command and response tokens **SPI:** Output for command and write data
MM2_DAT0	Bidirectional	Input	**MultiMediaCard Data 0—**MMC/SD/SDIO Controller #2 MMC and SD: Bidirectional line for read and write data. **SPI:** Input for response token and read data
MM2_DAT1	Bidirectional	Input	**MultiMediaCard Data 1—**MMC/SD/SDIO Controller #2 **MMC and SD:** Bidirectional line for read and write data. **SDIO:**Signals an interrupt condition to the controller.
MM2_DAT2/ MM2_CS0	Bidirectional	Output	**MultiMediaCard Data 2**—MMC/SD/SDIO Controller #2 **MMC and SD:** Bidirectional line for read and write data. **MultiMediaCard Chip Select 0**—MMC/SD/SDIO Controller #2 **SPI:** Chip select 0
MM2_DAT3/ MM2_CS1	Bidirectional	Output	**MultiMediaCard Data 3**—MMC/SD/SDIO Controller #2 MMC and SD: Bidirectional line for read and write data. **MultiMediaCard Chip Select 1**—MMC/SD/SDIO Controller #2 **SPI:** Chip select 1
MultiMediaCard/SD/SDIO Controller #3 Signals **NOTE: *MMC/SD/SDIO Card Controller* #3** (PXA31x Processor Only)			

Table 36: Multimedia Card/SD/SDIO Card Controller Interface Signal Summary (Continued)

Signal	Direction (MMC/SD/ SDIO Mode)	Direction (SPI Mode)	Description
MM3_CLK	Bidirectional	Output	**MultiMediaCard and SD Card Bus Clock**—MMC/SD/SDIO Controller #3
MM3_CMD	Bidirectional	Output	**MultiMediaCard Command**—MMC/SD/SDIO Controller #3 **MMC and SD:** Bidirectional line for command and response tokens **SPI:** Ooutput for command and write data
MM3_DAT0	Bidirectional	Input	**MultiMediaCard Data 0—**MMC/SD/SDIO Controller #3 **MMC and SD:** Bidirectional line for read and write data. SPI: Input for response token and read data
MM3_DAT1	Bidirectional	Input	**MultiMediaCard Data 1—**MMC/SD/SDIO Controller #3 **MMC and SD:** Bidirectional line for read and write data. **SDIO:** Signals an interrupt condition to the controller.
MM3_DAT2/ MM3_CS0	Bidirectional	Output	**MultiMediaCard Data 2**—MMC/SD/SDIO Controller #3 **MMC and SD:** Bidirectional line for read and write data. **MultiMediaCard Chip Select 0**—MMC/SD/SDIO Controller #3 **SPI:** Chip select 0
MM3_DAT3/ MM3_CS1	Bidirectional	Output	**MultiMediaCard Data 3**—MMC/SD/SDIO Controller #3 MMC and SD: Bidirectional line for read and write data. **MultiMediaCard Chip Select 1**—MMC/SD/SDIO Controller #3 **SPI:** Chip select 1

12.3 Layout Notes

Each PXA3xx processor family MMC/SD/SDIO controller is connected to MMC, SD, and SDIO card devices, but there are limitations to which device type installs into which socket type. See Table 37 for information on sockets and devices supported by the MMC/SD/SDIO controller.

Table 37: MMC/SD/SDIO Controller Supported Sockets and Devices

Sockets	Devices Supported
SD/SDIO card socket	SD card device SDIO card device MMC device
MMC socket	MMC device

See Table 38 for the list of maximum number of memory devices supported for the different operating modes and communication protocols of the MMC/SD/SDIO controller of the PXA3xx processor family.

Table 38: MMC/SD/SDIO Controller Supported Device Configurations

Controller Mode	Comm Protocol	Maximum Devices Supported	Description
MMC/SD/SDIO	MMC	MMC device stack size determined by signal loading limitations	MMC protocol supports multiple cards.
	SD	One SD card	Each SD card must have its own data bus. Each PXA3xx processor family MMC/SD/SDIO controller supports one data bus.
	SDIO	One SDIO card	Each SDIO card must have its own data bus. Each PXA3xx processor family MMC/SD/SDIO controller supports one data bus.
SPI	SPI	Two SD cards, SDIO cards, two MMC devices, or a combination of two device types	Each MMC/SD/SDIO controller of the PXA3xx processor family supports two chip selects, MMCCS0 and MMCCS1, and thus can support two devices in SPI mode. The specified devices must be in SPI mode.

Each PXA3xx processor family MMC/SD/SDIO card controller communicates at a maximum data rate of 26 Mbps for MMC, 1-bit SD/SDIO, and SPI mode data transfers. The maximum data rate is 104 Mbps for 4-bit MMC/SD/SDIO data transfers (26 Mbps x 4-bits).

The 3.3 V regulator serves several purposes in the reference platform:

- Provides a clean 3.3 V supply to the MMC/SD/SDIO card socket.
- Protects the main voltage supply on the board from being shorted or damaged by a defective MMC/SD/SDIO card device; there is the possibility of the MMC/SD/SDIO voltage regulator being damaged instead.
- Provides ability to easily power down the MMC/SD/SDIO card socket without having to power down the main 3.3 V supply.

The system designer determines whether to implement a dedicated regulator or pass-transistor for supplying 3.3 V to the MMC/SD/SDIO socket. The decision is based on analysis of functionality versus component/board real-estate costs.

Special considerations must be taken into account when choosing alternate-function signals to be used for each instantiation of the MMC/SD/SDIO controller. Signals within each instantiation could cross different power domains. Either the power domains must receive the same voltage or a different voltage-level shifter is required to interface the controller signals to the MMC/SD/SDIO socket.

Signals for the second instantiation can be selected such that they all appear on one power domain. The two different groups of signals for the second instantiation can be selected to appear either only

on the VCC_CARD2 power domain or the VCC_MSL power domain. All other combinations of signals appear on two or more power domains. See Table 39 for detailed information.

Table 39: Consideration of Power Domains During GPIO Alternate Function Selection (PXA30x processor only)

GPIO	VCC_Domain	MMC/SD/SDIO 1	MMC/SD/SDIO 2
GPIO_3	VCC_CARD1	MM1_DATA0	—
GPIO_4	VCC_CARD1	MM1_DATA1	—
GPIO_5	VCC_CARD1	MM1_DATA2	—
GPIO_6	VCC_CARD1	MM1_DATA3	—
GPIO_7	VCC_CARD1	MM1_CLK	—
GPIO_8	VCC_CARD1	MM1_CMD	—
GPIO_9	VCC_CARD2	—	MM2_DATA0
GPIO_10	VCC_CARD2	—	MM2_DATA1
GPIO_11	VCC_CARD2	—	MM2_DATA2
GPIO_12	VCC_CARD2	—	MM2_DATA3
GPIO_13	VCC_CARD2	—	MM2_CLK
GPIO_14	VCC_CARD2	MM1_CMD	MM2_CMD
GPIO_15	VCC_CARD2	MM1_CMD	—
GPIO_77	VCC_MSL	—	MM2_DATA0
GPIO_78	VCC_MSL	—	MM2_DATA1
GPIO_79	VCC_MSL	—	MM2_DATA2
GPIO_80	VCC_MSL	—	MM2_DATA3
GPIO_81	VCC_MSL	—	MM2_CLK
GPIO_82	VCC_MSL	—	MM2_CMD

Table 40: Consideration of Power Domains During GPIO Alternate Function Selection (PXA31x processor only)

GPIO	VCC_Domain	MMC/SD/SDIO 1	MMC/SD/SDIO 2	MMC/SD/SDIO 3
GPIO_3	VCC_CARD1	MM1_DATA0	—	—
GPIO_4	VCC_CARD1	MM1_DATA1	—	—
GPIO_5	VCC_CARD1	MM1_DATA2	—	—
GPIO_6	VCC_CARD1	MM1_DATA3	—	—

Table 40: Consideration of Power Domains During GPIO Alternate Function Selection (PXA31x processor only) (Continued)

GPIO	VCC_Domain	MMC/SD/SDIO 1	MMC/SD/SDIO 2	MMC/SD/SDIO 3
GPIO_7	VCC_CARD1	MM1_CLK	—	—
GPIO_8	VCC_CARD1	MM1_CMD	—	—
GPIO_9	VCC_CARD2	—	MM2_DATA0	—
GPIO_10	VCC_CARD2	—	MM2_DATA1	—
GPIO_11	VCC_CARD2	—	MM2_DATA2	—
GPIO_12	VCC_CARD2	—	MM2_DATA3	—
GPIO_13	VCC_CARD2	—	MM2_CLK	—
GPIO_14	VCC_CARD2	MM1_CMD	MM2_CMD	—
GPIO_15	VCC_CARD2	MM1_CMD	—	—
GPIO_29	VCC_IO3	MM1_DATA0	—	—
GPIO_77	VCC_MSL	—	MM2_DATA0	—
GPIO_78	VCC_MSL	—	MM2_DATA1	—
GPIO_79	VCC_MSL	—	MM2_DATA2	—
GPIO_80	VCC_MSL	—	MM2_DATA3	—
GPIO_81	VCC_MSL	—	MM2_CLK	—
GPIO_82	VCC_MSL	—	MM2_CMD	—
GPIO_84	VCC_MSL	MM1_DATA1	—	—
GPIO_89	VCC_MSL	MM1_DATA2	—	—
GPIO_90	VCC_MSL	MM1_DATA3	—	—
GPIO_103	VCC_IO1	—	—	MMC3_CLK
GPIO_105	VCC_IO1	—	—	MMC3_CMD
GPIO_7_2	VCC_IO1	—	—	MMC3_DATA0
GPIO_8_2	VCC_IO1	—	—	MMC3_DATA1
GPIO_9_2	VCC_IO1	—	—	MMC3_DATA2
GPIO_10_2	VCC_IO1	—	—	MMC3_DATA3

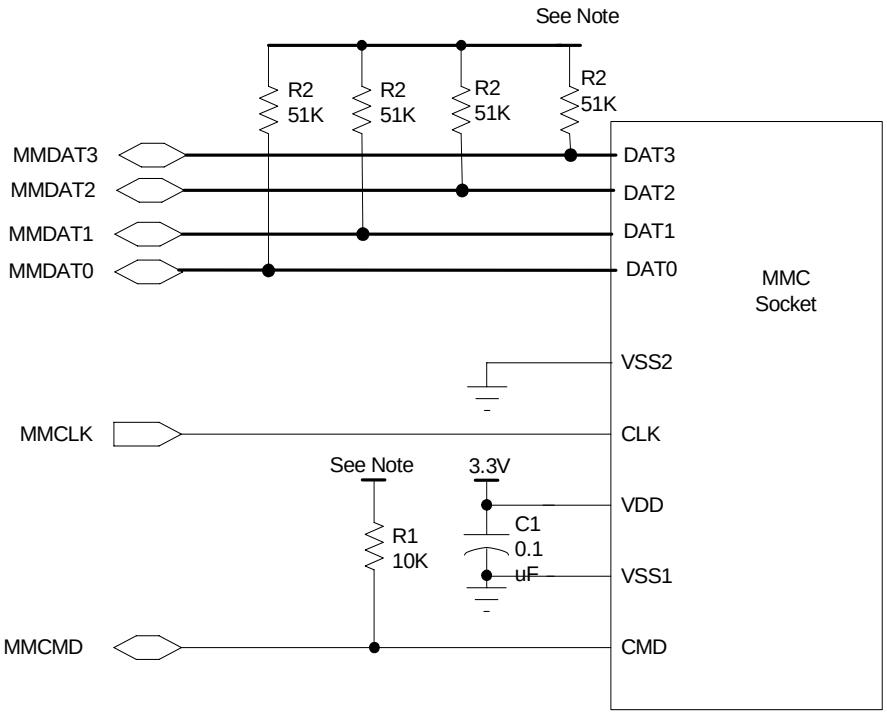
Table 41: Consideration of Power Domains During GPIO Alternate Function Selection (PXA32x processor only)

GPIO	VCC_Domain	MMC/SD/SDIO 1	MMC/SD/SDIO 2
GPIO5	VCC_DF	MM1_DATA0	MM2_DATA0
GPIO6	VCC_DF	MM1_DATA1	MM2_DATA1
GPIO7	VCC_DF	MM1_DATA2	MM2_DATA2
GPIO8	VCC_DF	MM1_DATA3	MM2_DATA3
GPIO18	VCC_CARD1	MM1_DATA0	—
GPIO19	VCC_CARD1	MM1_DATA1	—
GPIO20	VCC_CARD1	MM1_DATA2	—
GPIO21	VCC_CARD1	MM1_DATA3	—
GPIO22	VCC_CARD1	MM1_CLK	—
GPIO23	VCC_CARD1	MM1_CMD	
GPIO24	VCC_CARD2	—	MM2_DATA0
GPIO25	VCC_CARD2	—	MM2_DATA1
GPIO26	VCC_CARD2	—	MM2_DATA2
GPIO27	VCC_CARD2	—	MM2_DATA3
GPIO28	VCC_CARD2	—	MM2_CLK
GPIO29	VCC_CARD2	—	MM2_CMD
GPIO30	VCC_IO4	MM1_CLK	MM2_CLK
GPIO31	VCC_IO4	MM1_CMD	MM2_CMD
GPIO75	VCC_MSL	—	MM2_DATA0
GPIO76	VCC_MSL	—	MM2_DATA1
GPIO77	VCC_MSL	—	MM2_DATA2
GPIO78	VCC_MSL	—	MM2_DATA3
GPIO79	VCC_MSL	—	MM2_CLK
GPIO80	VCC_MSL	—	MM2_CMD

12.3.1 MMC Protocol Schematic Diagram

See the schematic diagram in Figure 40 of the interface between the MMC/SD/SDIO card controller and an MMC device socket using 4 bit mode.

Figure 40: MMC Protocol Interface Block Diagram



Note: Voltage and Resistors depend on the power domain associated with the connected GPIO pins.

12.3.2 MMC Protocol Layout Notes

See Table 42 for the list of pullup and pulldown resistors required for MMC devices according to their respective specifications.

Table 42: MMC Pullup and Pulldown Resistors

Signal	Pullup or Pulldown	Min	Max	Remark
MMCMD	pullup	4.7 KΩ	100 KΩ	Prevents bus floating
MMDAT0	pullup	10 KΩ	100 KΩ	Prevents bus floating

The 0.1 µF capacitor (C1) shown in Figure 40 must be located within 0.512 inches (13 mm) of the MMC socket pin 3 (VDD) and pin 4 (VSS1).

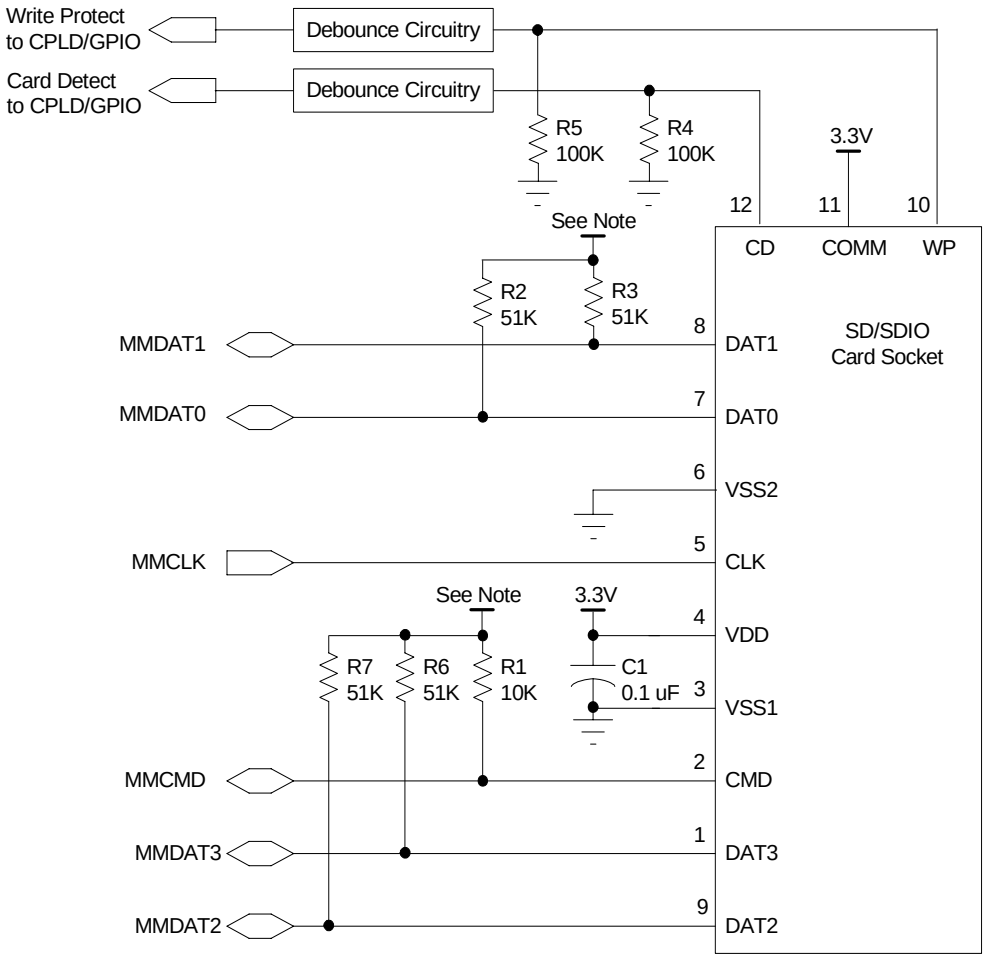
The MMC socket is not equipped with a mechanical card-detect switch; therefore, take other measures to produce a card detect. Use an SD/SDIO card socket if a card-detect and write-protection signal are preferred, even if only MMC devices are being used. The example circuit

shown in Figure 41 supports the use of an MMC device with an SD/SDIO card socket; MMDAT1 and MMDAT2 are not used as the MMC device does not have physical contacts to interface to these SD/SDIO card socket signals.

12.3.3 SD and SDIO Protocol Schematic Diagram

See Figure 41 for the detailed schematic diagram of the interface between the MMC/SD/SDIO card controller and an SD/SDIO card socket.

Figure 41: SD and SDIO Protocol Interface Block Diagram



Note: Voltage and resistors depend on the power domain associated with the connected GPIO pins.

MMC_007_P2

12.3.4 SD and SDIO Protocol Layout Notes

See Table 43 for the list of pullup and pulldown resistors required for SD card and SDIO card devices according to their respective specifications.

Table 43: SD/SDIO Card Pullup and Pulldown Resistors

Signal	Pull-up or Pull-down	Min	Max	Remark
MMCMD	pullup	10 KΩ	100 KΩ	Prevents bus floating
MMDAT<3:0>	pullup	10 KΩ	100 KΩ	Prevents bus floating
WP	pulldown	50 KΩ	100 KΩ	Any value sufficient to prevent bus floating

The 0.1 µF capacitor (C1) shown in Figure 41 must be located within 0.512 inches (13 mm) of the SD/SDIO card socket pin 3 (VDD) and pin 4 (VSS1).

COMM and the mechanical switches WP and CD signals are shown on the SD/SDIO card socket in Figure 41. COMM is optional (not required by the SD card and SDIO card specifications) and thus some SD/SDIO card sockets do not have this signal. When a device is inserted in the socket shown in Figure 41, CD is connected to COMM and WP is connected to COMM using mechanical switches inside the socket. Ensure the WP and CD signals have debounce circuitry.

SD cards have a write-protect tab that is optional for SDIO cards. Depending on the position of the tab, the WP signal is either connected or not connected to the COMM signal. In the example shown in Figure 41, COMM is tied to VCC and WP has a resistor. This scenario causes a rising edge to occur on the WP signal when the SD (or optional SDIO) card tab is in the write-protect position; the WP signal remains low when the tab is in the read/write position.

The mechanical switch CD signal functions in a similar manner as the switch in the socket closes to connect COMM to the socket CD signal when a device is inserted in the socket. In the example shown in Figure 41, the sockets CD signal has a pulldown resistor. This resistor causes a rising edge to occur on the socket CD signal when an SD/SDIO card device is inserted in the socket. Routing the CD of the socket signal to an appropriately configured GPIO input signal on the PXA3xx processor family is used as a technique for the processor to detect (using the socket's mechanical switch) that an SD/SDIO card device insertion occurred. Select a GPIO signal that can wake up the PXA3xx processor family from sleep mode using this technique. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, "Section 5: GPIO" for details on configuring GPIOs and GPIO sleep wake-up capabilities.

For sockets not having the COMM signal, connect the CD and WP signals to a CPLD or other device that can indicate to the driver software that the card is detected and write protected, respectively.

12.3.5 SPI Protocol Schematic Diagram

See the block diagram in Figure 41 showing the interface between the MMC/SD/SDIO card controller and the following devices:

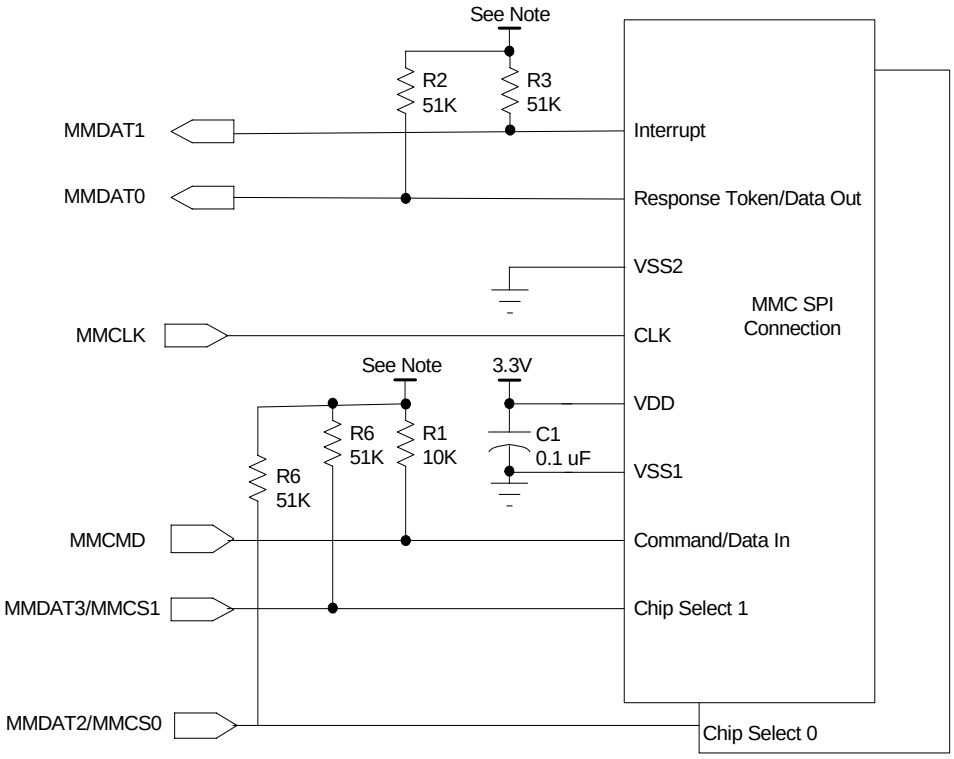
- Two MMC devices
- Two SD cards
- Two SDIO cards
- A maximum of two mixed type devices

The block diagram shows high-level signal use and connectivity when using the SPI communication protocol with the specified devices.

12.3.6 SPI Protocol Layout Notes

The PXA3xx processor family signals MMCLK, MMCMD, and MMDAT0 are connected in parallel to both device sockets for dual-device SPI-mode operation. MMCCS0 and MMCCS1 are used as chip selects only in SPI mode and connect separately to the respective socket chip select signals.

Figure 42: SPI Protocol Interface Block Diagram



Note: Voltage and resistors depend on the power domain associated with the connected GPIO pins.

13 LCD Interface

The PXA3xx processor family LCD panel controller is backward-compatible with the Marvell PXA25x Processor Family, the Marvell PXA26x Processor Family, and Marvell PXA27x Processor Family LCD controllers. Several additional features are also supported:

- Interrupt status and enable bits for real end-of-frame interrupts.
- Support for an additional interrupt when using multiple descriptors
- Ability to turn off base fetching and use a constant color for the base
- Programmable capability to delay L_BIAS one cycle clock for active panels.
- PXA31x Processor also supports chroma-key operations for 16-bit RGB base layers with overlays and for 16-bit RGB overlays in 19-bit mode.
- PXA31x Processor also supports 18-bit RGB data formatted as 24 bits (6 bits of red, two zeroes, 6 bits of green, two zeroes, 6 bits of blue, and two zeroes). This is supported for 18-bit packed and unpacked data as well as 19-bit unpacked data (18 bits of data with overlays enabled).

13.1 Signals

See Table 44 for the list of PXA3xx processor family LCD interface signals.

Table 44: LCD Interface Signal List

Signal Name	Type	Description
LDD<17:0>	Bidirectional	Data lines used to transmit 4-, 8-, 16-, or 18-data values at a time to the LCD display module. Not all panels require all data lines for operation. Consult the panel manufacturer's documentation. LDD<7:0> are used as the data bus to/from a smart panel. Used to: 1. Transfer data values to the panel's frame buffer or transfer a command to the panel 2. Transfer data values from the panel's frame buffer or transfer status from the panel
L_PCLK_WR	Output	Pixel clock used by the LCD display module to clock the pixel data into the Line Shift register. In passive mode, pixel clock toggles only when valid data is available on the data pins. In active mode, pixel clock toggles continuously and the AC bias pin is used as an output to signal when data is valid on the LCD data pins. Write signal for writing to LCD panels with internal frame buffer.
L_LCLK_A0	Output	Line clock used by the LCD display module to signal the end of a line of pixels that transfers the line data from the Shift register to the screen and increments the line pointers. Also, it is used by active (TFT) displays as the horizontal synchronization signal. For active displays, this is also referred to as the "horizontal sync signal" or "HSYNC". A control signal specifies command or data transactions when interfacing to an LCD panel with internal frame buffer.

Table 44: LCD Interface Signal List (Continued)

Signal Name	Type	Description
L_FCLK_RD	Output	Frame clock used by the LCD display module to signal the start of a new frame of pixels that resets the line pointers to the top of the screen. Also, it is used by active (TFT) display module as the vertical synchronization signal. For active displays, this is also referred to as the “vertical sync signal” or “VSYNC”. Read signal during reads from an LCD panel with internal frame buffers.
L_BIAS	Output	AC bias used to signal the LCD display module to switch the polarity of the power supplies to the row and column axis of the screen to counteract DC offset. In active (TFT) mode, it is used as the output-enable to signal when data should be latched from the data pins using the pixel clock.
L_CS	Output	No connect for active (TFT) and passive (STN) panels. For smart panels, L_CS is used as chip-select signal.
L_VSYNC	Input	Refresh sync signal from smart panels. No connect for active (TFT) and passive (STN) panels.

Note

Not all signals are required for all modes of operation. Refer to the LCD panel reference documentation specific to the manufacturer for information on:

- Specific signals required for correct LCD operation
- Correct names of the signals used by the LCD panel manufacturer

13.2 Modes of Operation Overview

13.2.1 Passive Color Mode

Passive color displays send dithered data to the panel. Each bit of color data requires three pins.

13.2.1.1 Signals

For passive color displays, see Table 45 for description of the pins required for connections between the PXA3xx processor family and the LCD panel.

Table 45: Passive Display Pins Required

PXA30x, PXA31x, PXA32x Processor Pin	LCD Panel Pin	Pin Type[1]	Definition
LDD<7:0>	D<7:0>	Output	Data lines used to transmit 2 2/3 data values at a time to the LCD display. Groupings of three pin values represent one pixel (red, green, and blue data values).
L_PCLK_WR	Pixel_Clock	Output	Pixel Clock – used by the LCD display to clock the pixel data into the line shift register.
L_LCLK_A0	Line_Clock	Output	Line Clock – used by the LCD display to signal the end of a line of pixels. Line clock transfers a line of pixels from the shift register to the screen and increment the line pointer.

Table 45: Passive Display Pins Required (Continued)

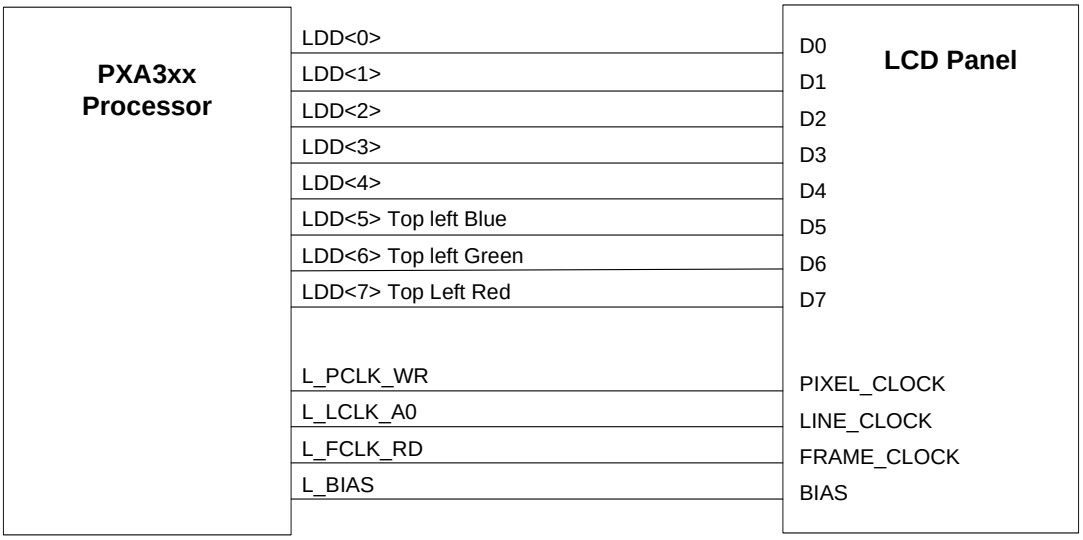
PXA30x, PXA31x, PXA32x Processor Pin	LCD Panel Pin	Pin Type[1]	Definition
L_FCLK_RD	Frame_Clock	Output	Frame Clock – used by the LCD displays to signal the start of a new frame of pixels. Frame clock resets the line pointer to the top of the screen.
L_BIAS	Bias	Output	Bias – used to signal the LCD display that it should switch the polarity of the power supplies to the row and column axis of the screen to counteract DC offset.
N/A	Vcon[2]	N/A	Contrast voltage – adjustable voltage input to LCD panel. External voltage circuitry is required (no pin is provided on the PXA3xx processor family).

NOTE:
1. "Pin Type" is in reference to the PXA3xx processor family. Therefore, outputs are pins that drive a signal from the processor to another device.
2. Vcon is a signal external to the PXA3xx processor family.
3. Names used for "LCD Panel Pin" are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on specific signals required for correct LCD operation and correct names of the signals used by the LCD panel manufacturer.

13.2.1.2 Schematics / Block Diagram

See Figure 43 for an illustration of typical connections for a single-scan color passive panel. The figure indicates the top, left pixel color bits.

Figure 43: Passive Color Display Typical Connection



13.2.2 Active Color

An active color display does not send dithered data to the panel. Rather, the driven data lines together hold the digital value of the pixel that is being transferred. A single pixel is transferred per clock cycle. The 16 or 18 bits of data describe the intensity level of the red, green, and blue colors for each pixel. Typically, the 16 bit data is formatted as five bits for red, six bits for green, and five bits for blue (5:6:5) and 18 bit data is 6:6:6. The format varies by display and is controlled by the software writing to the frame buffer. Refer to the panel's documentation to ensure the correct PXA3xx processor family LCD data lines are connected to the correct LCD panel data lines.

13.2.2.1 Connecting to 16-Bit, 18-Bit, and 24-Bit Active LCD Panels

The PXA3xx LCD controller has 18 data bits total. There are two main concerns when connecting a panel: the physical hardware connection, and the arrangement of the pixel data in the frame buffer.

Connecting 16-, 18-, or 24-bit panels is possible in the following way.

- These are the typical 1:1 type connections:
  - 16-bit data to 16-bits of data on the panel - shown in Figure 44.
  - 18-bit data to 18-bits of data on the panel - shown in Figure 45.

This type of interfacing and simple to connect. The data in the frame buffer is organized in 16 or 18 bits and the actual bus width that connects to the LCD panel are the same width.

- These are methods of connecting panels with different data line widths;
  - 16-bit data to 18-bits of data on the panel - shown in Figure 46.
  - 16-bit data to 24-bits of data on the panel - shown in Figure 47.
  - 18-bit data to 24-bits of data on the panel - shown in Figure 48.

For example, the Zylonite development board (connections not shown in this document) uses the 16-bit LCD bus to connect to an 18-bit panel. This is interfaced by tying the R0 and R1 of the panel to R0 of the PXA3xx LCD data bus. Likewise for Blue. The 6 Green data pins connect directly

between the PXA3xx device and the panel. Keeping the pixel data in the driver at 16 bits allows software complexity to be kept to a minimum. To the LCD driver and operating system, the data is only 16 bit. At the hardware level, it could be an 18-bit or 24-bit LCD panel.

This type of connection does work; however, the transition from black to white for each pixel will have a step function around the mid-point due to the LSB toggling more often.

Figure 46, Figure 47, and Figure 48 provide examples, which enable platforms with the PXA3xx processor to drive the higher BPP panels and provide a more linear transition between black and white.

Note

The PXA3xx processor family LCD controller includes the ability to use logical overlays for simplified software control of multiple planes (overlays) of image data. While this is essentially a software-related feature, the implementation of the overlays for 16-bpp operation impacts the way in which signals are connected between the PXA3xx processor family and LCD panel. Set LCCR0[LDDALT] to avoid any issues associated with the implementation of the overlays. Refer to the description of the LCD Controller Control Register 0 in *PXA30x Processor and PXA31x Processor Vol. III: Graphics and Input Controller Configuration Developers Manual*, “Section 1: LCD Controller” for more information.

13.2.2.2 PXA3xx LCD Signals

The signals in Table 46 implement an active color 16-bpp and 18-bpp display with the PXA3xx processor family.

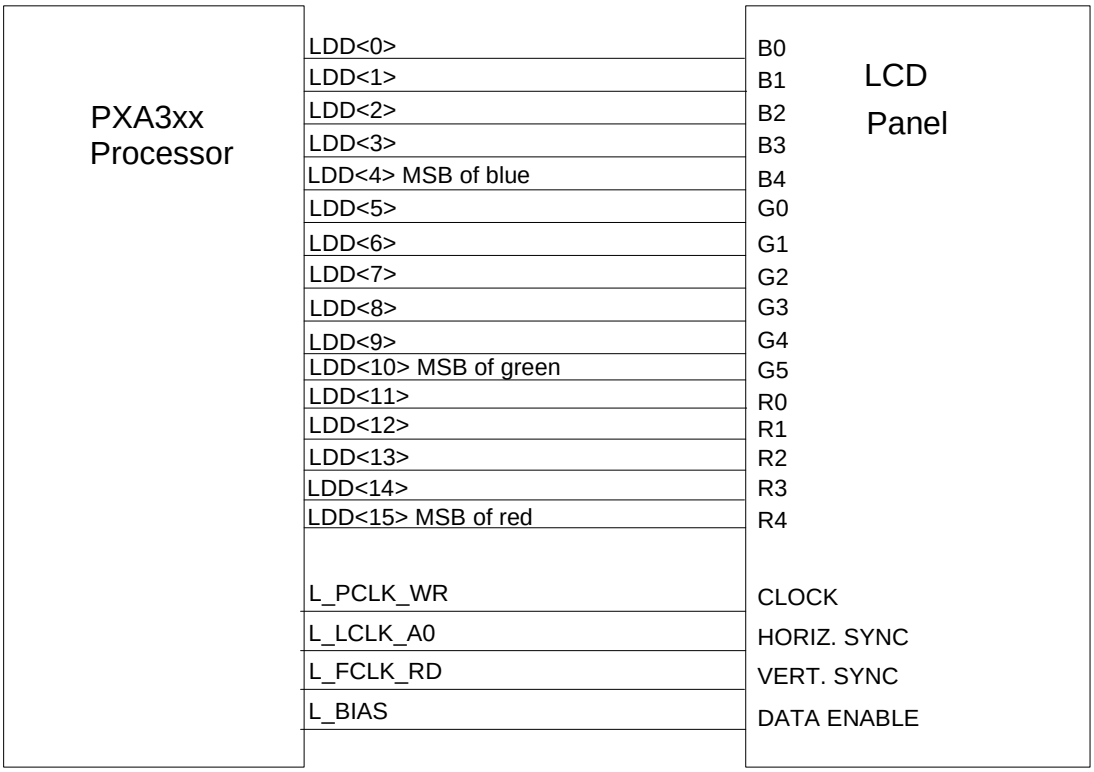
Table 46: Active Color (16bpp and 18bpp) Display Pins Required

PXA30x, PXA31x, PXA32x Processor Pin	LCD Panel Signal Name	Type	Definition
LDD<15:11> LDD<10:5> LDD<4:0>	R<4:0> G<5:0> B<4:0>	Bidirectional	Data Lines - used to transfer pixel values to the LCD panel in 16 bit mode (RGB 5:6:5)
LDD<17:12> LDD<11:6> LDD<5:0>	R<5:0> G<5:0> B<5:0>	Bidirectional	Data Lines – used to transfer pixel values to the LCD panel in 18 bit mode (RGB 6:6:6)
L_PCLK_WR	CLOCK	Output	Pixel Clock – used by the LCD display to clock the pixel data into the line shift register.
L_LCLK_A0	HORIZONTAL SYNC.	Output	Horizontal Sync. - used by an active (TFT) panel as its horizontal synchronization signal.
L_FCLK_RD	VERTICAL SYNC.	Output	Vertical Sync. - used by an active (TFT) panel as its vertical synchronization signal.
L_BIAS	DATA ENABLE	Output	Data Enable - used as an output-enable to signal the panel when data can be latched using the pixel clock.
NOTE: Names used for “LCD Panel Signal Name” are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on specific signals required for correct LCD operation and correct names of the signals used by the LCD panel manufacturer.			

13.2.2.3 16bpp Schematics / Block Diagram

See Figure 44 for an illustration of typical connections for a 16-bpp active panel display. The sample connections serve as a guide for designing systems that contain active LCD displays. The most significant bit (MSB) of each color is indicated.

Figure 44: Active Color 16bpp Display Typical Connection



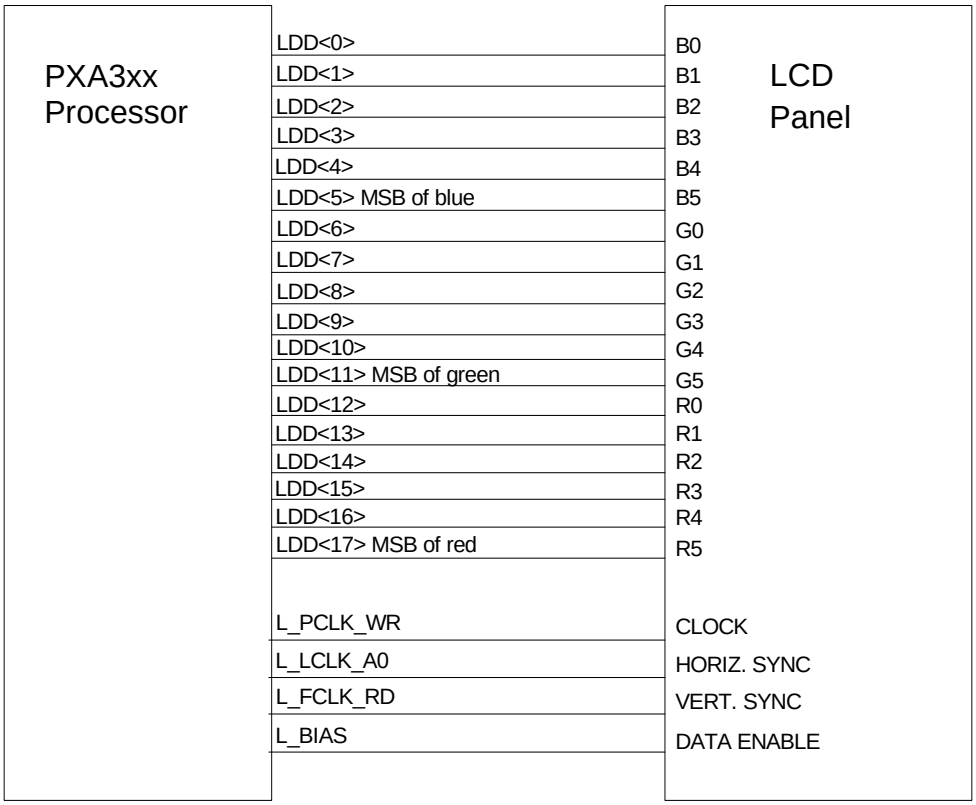
13.2.3 Active Color, 18bpp Mode

The bits of data describe the intensity level of the red, green, and blue colors for each pixel, in this case 18bpp - RGB 6:6:6.

13.2.3.1 Schematics / Block Diagram

See Figure 45 for an illustration of typical connections for an 18-bpp active panel display. The sample connections serve as a guide for designing systems that contain active LCD displays. The most significant bit (MSB) of each color is indicated.

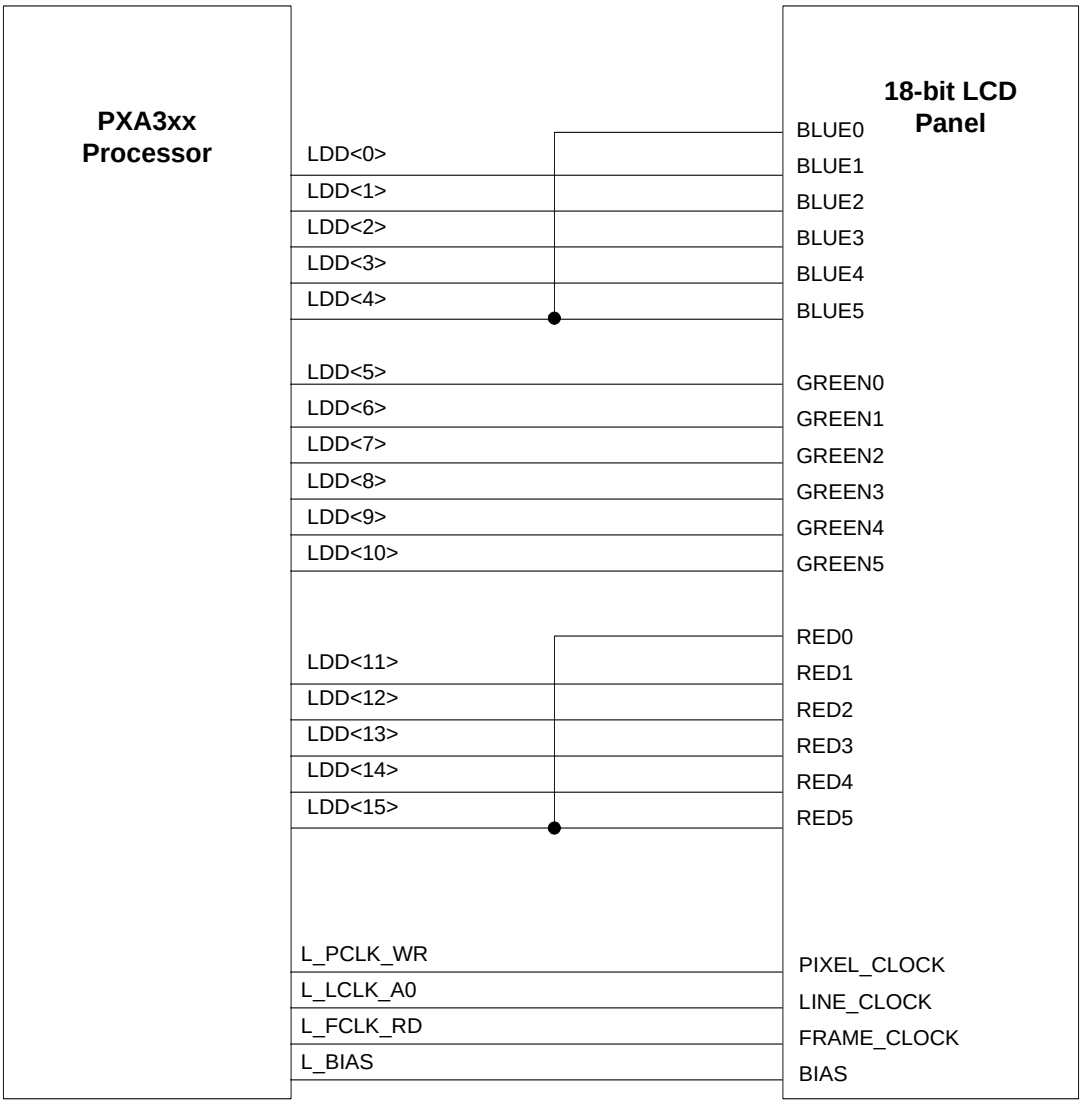
Figure 45: Active Color 18-bits-per-Pixel Display Typical Connection



13.2.3.2 16-Bit to 18-Bit Data on Panel

See Figure 46 for an illustration of typical connections for a 16-bpp connection to an 18-bit active panel display.

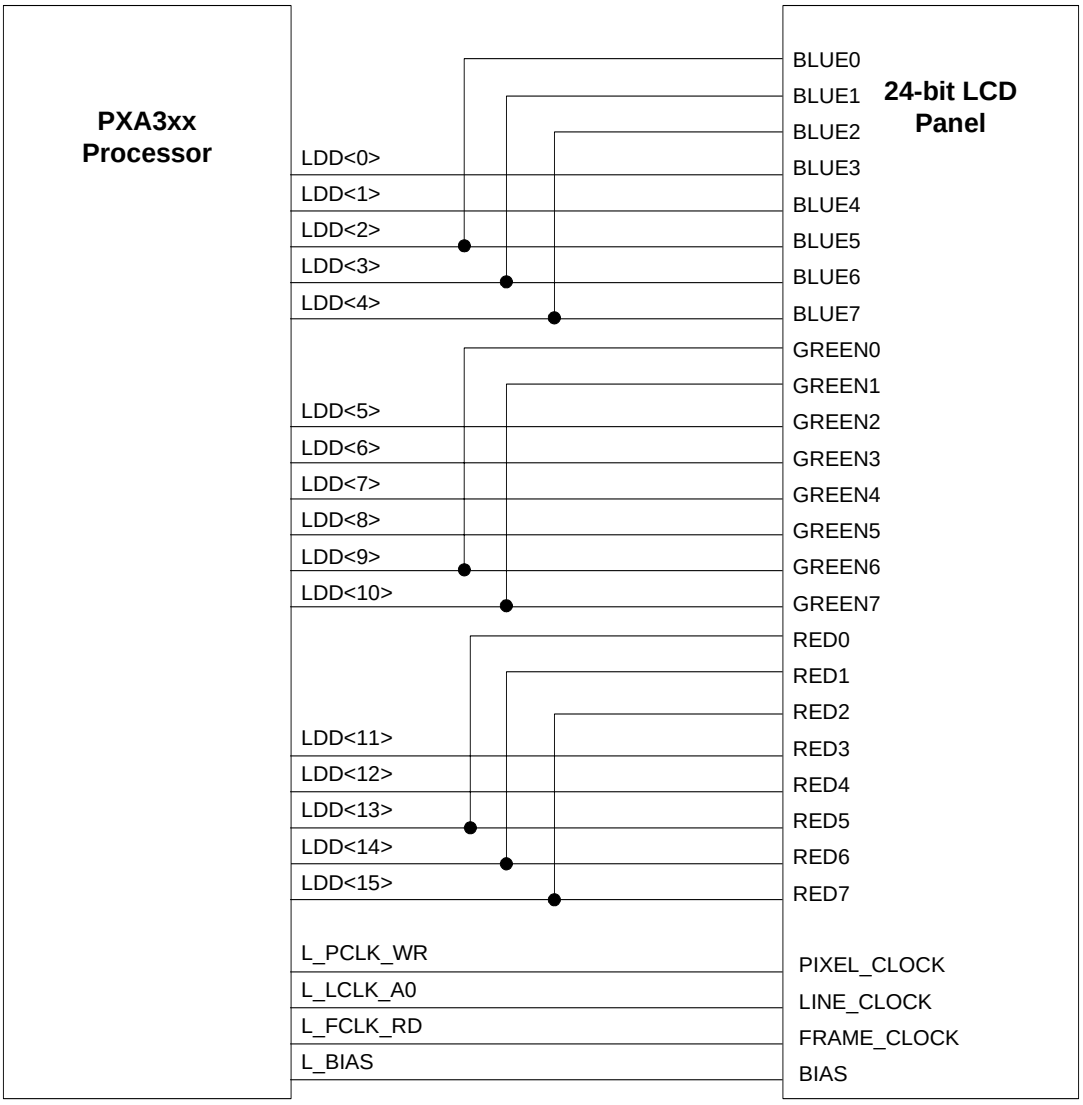
Figure 46: Active Color 16bpp to 18-bit Display Typical Connection



13.2.3.3 16-Bit to 24-Bit Data on Panel

See Figure 46 for an illustration of typical connections for a 16-bpp connection to a 24-bit active panel display.

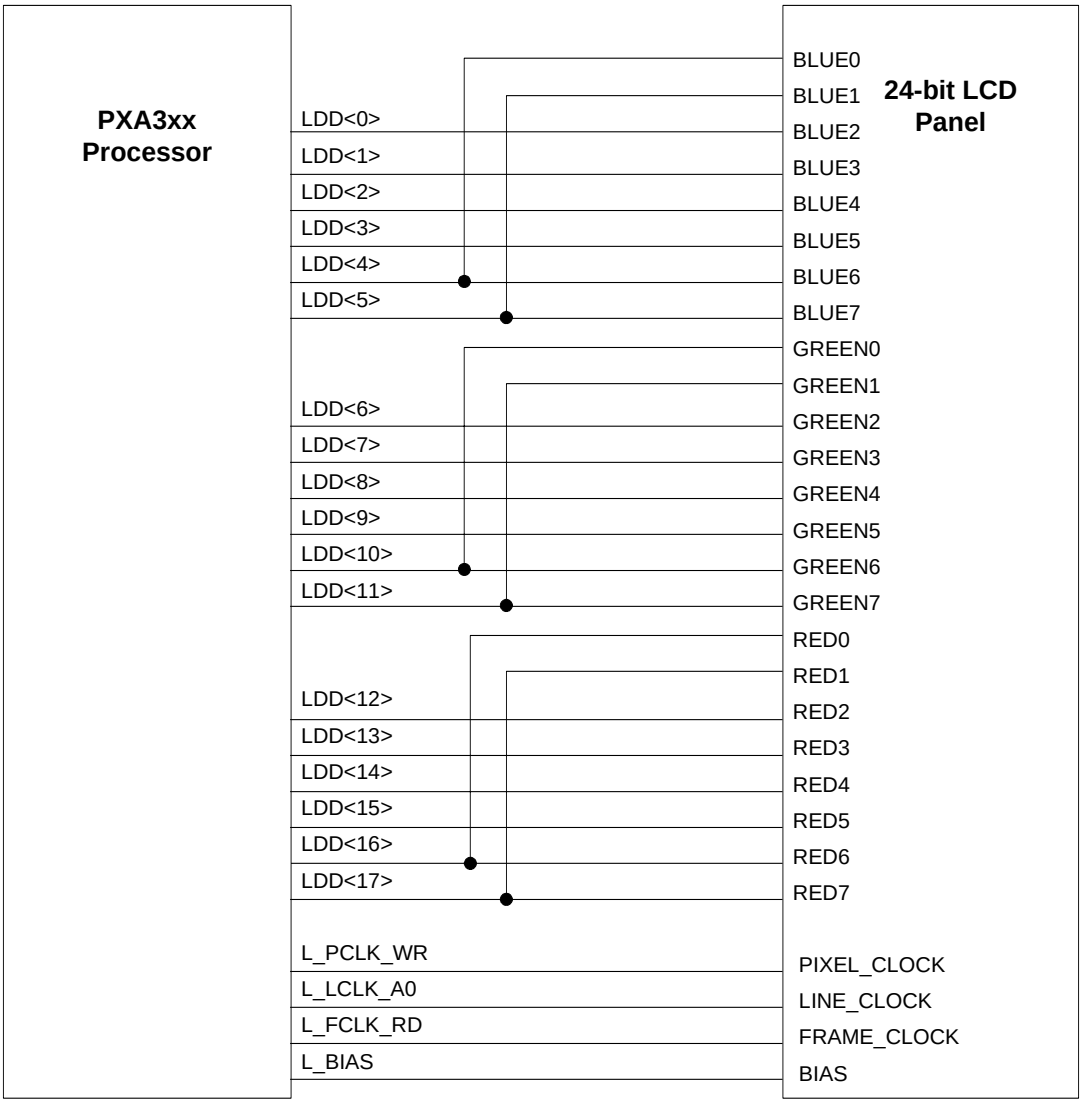
Figure 47: Active Color 16bpp to 24-bit Display Typical Connection



13.2.3.4 18-Bit to 24-Bit Data on Panel

See Figure 46 for an illustration of typical connections for a 18-bpp connection to an 24 bit active panel display.

Figure 48: Active Color 18-bits-per-Pixel to 24-bit Display Typical Connection



13.2.4 Layout Notes

Refer to Section 13.3, Layout Notes for layout notes and considerations.

13.2.5 Smart Panel

A smart panel is an LCD panel (which is typically active, but could be passive) that contains the required frame buffer memory. Having the frame buffer local to the smart panel requires the panel to contain more sophisticated logic than compared with a normal LCD panel. Smart panels are equivalent to simple microcontrollers in capabilities and interface characteristics.

Accordingly, the pin interface between the LCD controller on the PXA3xx processor family and the smart panel is similar to that of a microcontroller interface. The PXA3xx processor family uses the same physical pins as with a normal active or passive panel, but with two additions. See Table 47 for description of the pins used in the connection for smart panels and their functions.

Note

The interface only uses eight data pins, LDD<7:0>. Therefore, three successive clock cycles are used to transfer one pixel to the panel. Refer to the smart panel manufacturer's documentation for information on how data is transmitted to the display.

13.2.5.1 Signals

For active displays, connect the pins as described in Table 47 between the PXA3xx processor family and LCD panel.

Table 47: Smart Panel Active Display Pins Required

PXA30x, PXA31x, PXA32x Processor Pin	LCD Panel Pin	Pin Type[1]	Definition
LDD<7:0>	D<7:0>	Output	LDD<7:0> are used as the data bus to/from a smart panel. Used to: 1. Transfer data values to the panel's frame buffer or transfer a command to the panel 2. Transfer data values from the panel's frame buffer or transfer status from the panel
L_PCLK_WR	Write	Output	Write - write signal for a smart panel.
L_LCLK_A0	Command	Output	Command - control signal that specifies a command or data transaction to panels.
L_FCLK_RD	Read	Output	Read - read signal for a smart panel.
L_CS	Select	Output	Select - used as a chip-select signal for smart a panel.
L_VSYNC	Sync	Input	Refresh sync signal from the smart panel.
N/A	Vcon[2]	N/A	Contrast Voltage – adjustable voltage input to smart panel. External voltage circuitry is required (no pin is provided on the PXA3xx processor family).

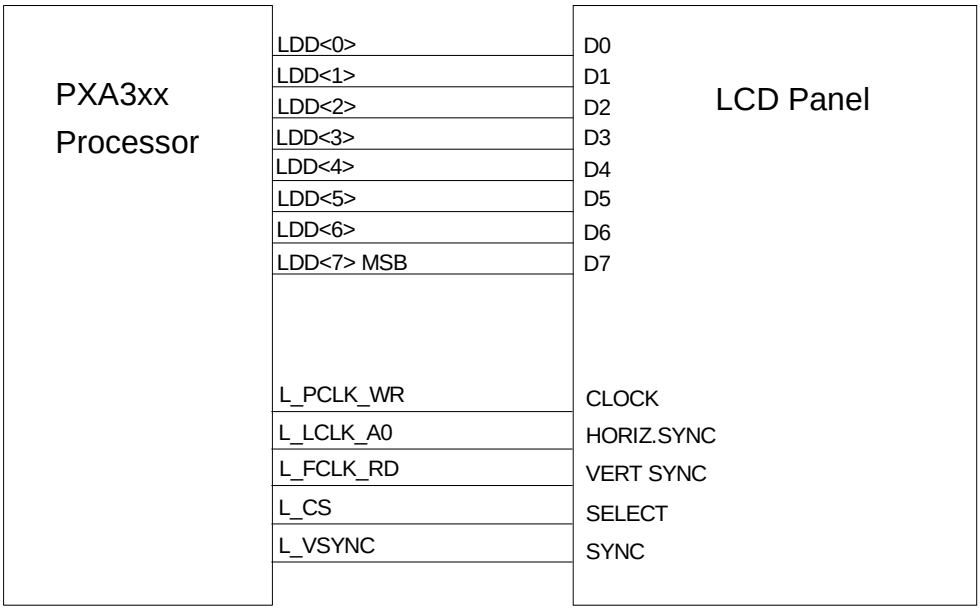
NOTE:
1. In reference to the PXA3xx processor family. Therefore, outputs are pins that drive a signal from the PXA3xx processor family to another device.
2. Vcon is a signal external to the PXA3xx processor family. Refer to Section 13.3.1 for detailed information.

NOTE: Names used for "LCD Panel Pin" are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on specific signals required for correct LCD operation and correct names of the signals used by the LCD panel manufacturer.

13.2.5.2 Schematics / Block Diagram

See Figure 49 for an illustration of typical connections for a smart panel. The sample connections serve as a guide for designing systems that contain LCD displays with an embedded frame-buffer memory. The figure indicates the most significant bit (MSB) of each byte that is transferred.

Figure 49: Smart Panel Active Color Display 8-bit Typical Connection



13.3 Layout Notes

Refer to Section 13.3, Layout Notes for layout notes and considerations.

13.3.1 General LCD Routing Recommendations

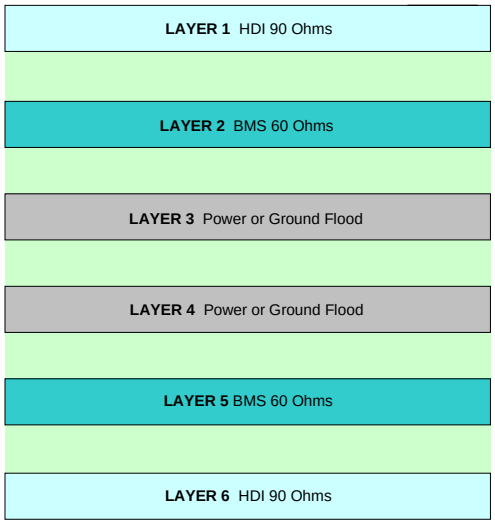
The following recommendations are intended for single-panel and dual-panel (flip-phone) LCD. These recommendations are valid for both 1.8V and 3.3V modes of operation.

- Minimum 6-layer board is recommended.
- HDI and BMS routes are recommended. Further simulations are required for stripline.
- HDI should be used for breakout and BMS for actual signal routing. Minimum length on HDI is determined by board technology limitations (no specific signal integrity restrictions).
- 90 ohm target impedance is recommended for HDI.
- 60 ohm target impedance is recommended for BMS.
- Impedance for ribbon cable (or flat top cable) is assumed to be 100 ohms.
- A solid VSS (non-dedicated) and a power flood (dedicated for return path) is recommended for this interface.
- Minimum 1:1 edge-to-edge (Trace spacing => Trace Width) spacing is recommended for all traces.
- 4x (GPIO register setting 011/Fast_4mA) drive strength is recommended for all signals to mitigate any potential EMI issues with ribbon cable.

- Maximum frequency of operation supported by this design guide (freq. of L_PCLK_WR) is 33 MHz.

13.3.2 Recommended Layer Stack Up

Figure 50: SIx Layer Board Stack Up Example



13.3.3 LCD Routing Recommendations

The Table 48 lists the signals for which these recommendations are provided.

Table 48: LCD Signals

Signals	Type
LDD[17:0]	Output
L_PCLK_WR	Output
L_LCLK_A0	Output
L_FCLK_RD	Output
LDD[7:0]	Input

13.3.3.1 Single Load Designs

Figure 51 shows a point -to-point topology for single load cases (no stubs). LDD, L_LCLK_A0, L_FCLK_RD and L_PCLK_WR needs to be matched within +- 1.0 inch

Figure 51: Signals: LDD[17:0], L_PCLK_WR, L_LCLK_A0, L_FCLK_RD

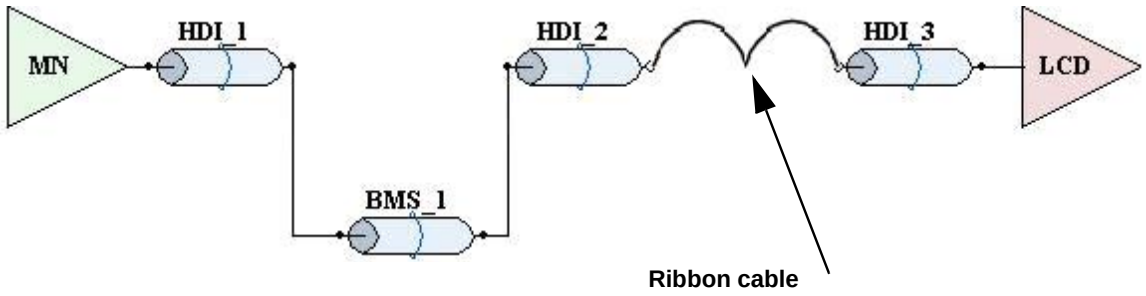


Table 49: Routing Length Recommendations

Mode	Section	Maximum Length*	Minimum Length
1.8V	HDI_1 + BMS_1 + HDI_2 + Ribbon Cable + HDI_3	9 inches	0.5 inches
3.3V	HDI_1 + BMS_1 + HDI_2 + Ribbon Cable + HDI_3	9 inches	0.5 inches
NOTE: HDI should be used mainly for breakout. Max length on HDI_1 or HDI_2 <=0.5 inches			

13.3.3.2 Dual Panel Display Designs

Figure 52 shows the topology for dual panel display (flip-phones) designs. LDD and L_PCLK_WR needs to be matched within + 1.0 inch

Figure 52: Signals: LDD[17:0], L_PCLK_WR, L_LCLK_A0, L_FCLK_RD

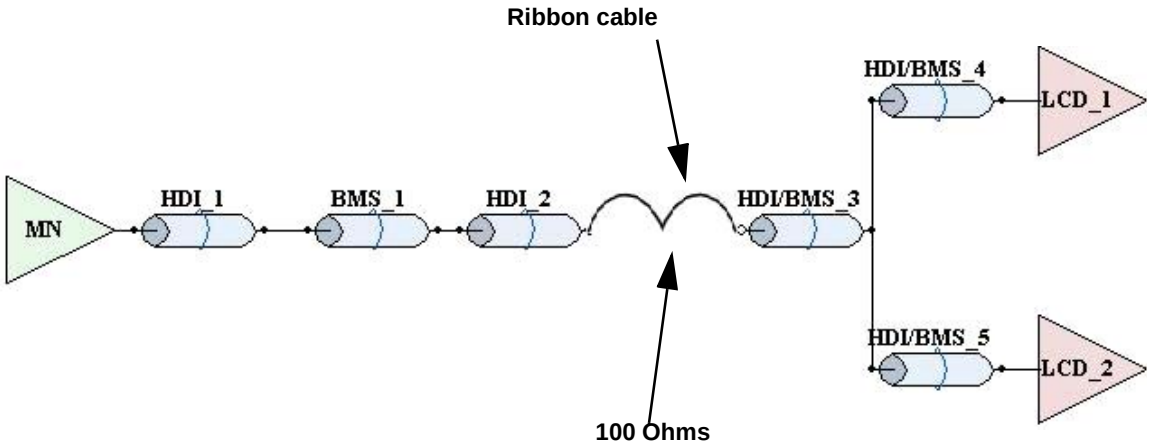


Table 50: Routing Length Recommendations

Mode	Section	Maximum Length*	Minimum Length
1.8V	HDI_1 + BMS_1 + HDI_2 + Ribbon Cable + HDI/BMS_3 + HDI/BMS_4	8.5 inches	0.5 inches
3.3V	HDI_1 + BMS_1 + HDI_2 + Ribbon Cable + HDI/BMS_3 + HDI/BMS_4	8.5 inches	0.5 inches
NOTE: * HDI should be used mainly for breakout. Max length on HDI_1 or HDI_2 <=0.5 inches			

14 Mini-LCD Controller

This section describes examples of hardware connections between the mini-LCD controller of the PXA30x processor and/or PXA31x processor and the LCD panel.

The mini-LCD controller is intended for use when the processor is in its S0/D1/C2 mode (standby with LCD refresh)—the lowest power mode with LCD refresh. The basic function of the mini-LCD controller is to support low-power operation refreshing an LCD panel using DMA and a frame buffer stored in the internal SRAM of the PXA30x processor and/or PXA31x processor.

The mini-LCD controller supports active (TFT) panels only. An example is provided showing connection of the PXA30x processor and/or PXA31x processor to an LCD panel.

Note

The Mini-LCD mode (S0/D1/C2) is NOT supported in the PXA32x processor. This chapter is applicable only to PXA30x and PXA31x processors.

14.1 Overview

The mini-LCD controller is used during the S0/D1/C2 low-power mode, standby with static image refresh (mini-LCD). S0/D1/C2 is used when the LCD panel needs to stay on, but the PXA30x processor and/or PXA31x processor has no work or only infrequent work to perform. The mini-LCD controller is used to allow refresh of the LCD panel while the other systems of the PXA30x processor and/or PXA31x processor remain inactive. The mini-LCD controller supports a small subset of the features supported by the main LCD controller. It does not support any overlays or the cursor. It does support the following features:

- Display mode
  - Active color mode with up to 32768 colors (15 bits) driven over 16 data lines
- RGB 1:5:5:5 pixel format only.
- Maximum frame refresh rate of 60 Hz
- Programmable wait-state insertion at beginning and end of each display line
- Programmable polarity for output enable, frame clock, and line clock
- Operation at 39 MHz frequency
- *No support* for overlays, hardware cursor and passive panels

See the *PXA30x Processor and PXA31x Processor Vol. III: Graphics and Input Controller Configuration Developers Manual*, “Section 2: Mini-LCD Controller” for details.

14.2 Signals

The mini-LCD controller signals are shown in Table 51.

The mini-LCD controller displays the data only when the processor is in S0/D1/C2 mode. During this mode, the main LCD controller is turned off. The mini-LCD controller gets the frame data from the internal SRAM memory through a private interface. The mini-LCD controller outputs are multiplexed on the same pins as the main LCD controller. All mini-LCD I/O and control signals are powered using the VCC_LCD power domain.

There are no additional connection notes or block diagrams required for enabling mini-LCD mode.

Table 51: Mini-LCD Interface Signal Descriptions

Mini-LCD Signal Name	GPIO Alternate Function Signal Name	Type	Definition
ML_DD<17:0>	L_LP_DD<15:0>	Output	Data Lines - used to transfer a 18-bit pixel to the mini-LCD panel. **NOTE:** These signals come out on the same GPIO pins as the regular LCD data lines.
ML_PCLK	L_LP_PCLK	Output	Pixel Clock - used by a mini-LCD panel to clock pixel data into its Line Shift register. Pixel clock transitions continuously, and the ML_EN_BIAS pin is used to signal the mini-LCD panel that data is valid on the data lines.
ML_LCLK	L_LP_LCLK_A0	Output	Line Clock - used by a mini-LCD panel to signal the end of a line of pixels. Line clock transfers a line of data from the panel's Shift register to the display and increments the line pointer. It is used by active (TFT) panels as the horizontal synchronization signal.
ML_FCLK	L_LP_FCLK_RD	Output	Frame Clock - used by a mini-LCD panel to signal the start of a new frame of pixels. Frame clock resets the line pointer to point to the top of the display. It is used by active (TFT) panels as the vertical synchronization signal.
ML_EN_BIAS	L_LP_BIAS	Output	Enable signal - used as an output-enable to signal the mini-LCD panel when valid data should be latched from the data lines by using the pixel clock.

15 Keypad Interface

This section describes the procedures for interfacing with the PXA3xx processor family keypad controller.

15.1 Overview

The keypad interface block provides an interface to two styles of keypads: direct key and matrix key, and supports both types of keypads simultaneously.

The keypad controller supports up to 8 x 8 matrix keys, up to eight direct keys, and one rotary encoder (which can implement items such as scroll keys, jog-dials, and thumb wheels).

The matrix keypad interface supports manual and automatic scans of the keypad array. Additional information concerning the different type of scan modes is documented *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. III: Graphics and Input Controller Configuration Developers Manual*, “Section 5: Keypad Controller”.

Included within the keypad controller is debounce logic with a programmable interval period. The internal debounce logic is disabled by setting the interval period to zero.

The PXA3xx processor family integrates internal pull-up and pull-down resistors that are individually configurable and can operate in all power modes except for S3/D4/C4. This eliminates the need to add external resistors to the system for keypad operation.

All references to registers are documented in the *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. III: Graphics and Input Controller Configuration Developers Manual*, “Section 5: Keypad Controller” unless otherwise noted. The following information provides examples for specific configurations. This section has not attempted to document all possible configurations, just the most commonly used.

15.2 Signals

This section describes the keypad interface signals. See Table 52 for summary of the signals.

Table 52: Interface Signals Summary

Name	Type	Description
KP_DKIN<7:0>	Input	Direct Key Inputs – These are the input signals from the direct keys and the rotary encoder sensors. Pins KP_DKIN<7:4> are dedicated input pins for direct keys 7 through 4. Pins KP_DKIN<3:2> are used as input pins for direct keys 3 and 2, or rotary-encoder sensor readings for rotary encoder 1 if it is enabled. Pins KP_DKIN<1:0> are used as input pins for direct keys 1 and 0 or rotary-encoder sensor readings for rotary encoder 0 if it is enabled.
KP_MKIN<7:0>	Input	Matrix Key Returns – These are input signals from the matrix keypad and are the matrix keypad row readings.

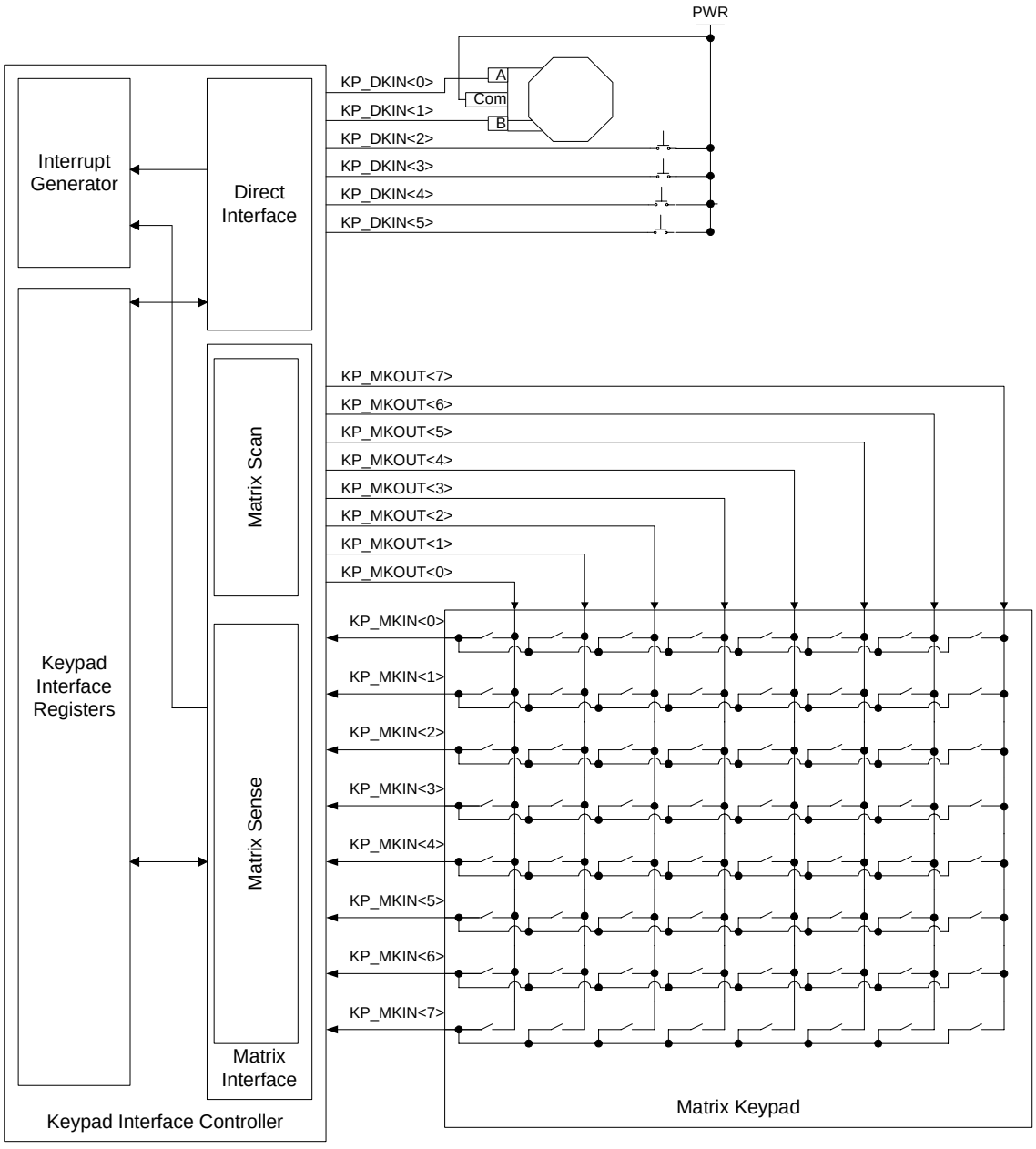
Table 52: Interface Signals Summary (Continued)

Name	Type	Description
KP_MKOUT<7:0>	Output	Matrix Key Outputs – The keypad interface sends scan signals on these outputs to the columns of the matrix keypad to detect any key(s) pressed. In the automatic-scan mode, the scan signals are sent when the automatic scan (AS) bit of the keypad interface control (KPC) register is set, or if there is stable keypad activity for one key debounce interval and the automatic scan on activity bit (ASACT) of the KPC is set. In the manual-scan mode, the scan signals are specified in the KPC register.

15.3 Block Diagram

See Figure 53 for an illustration of one of many possible configurations when interfacing to the PXA3xx processor family keypad controller. This specific configuration shows the keypad controller interfaced to the maximum size matrix keypad of 64 keys and six direct keys with a single rotary encoder.

Figure 53: Keypad Interface Diagram to 8x8 Matrix Keys, 4 Direct Keys, Rotary Encoder



15.4 Layout Notes

This section contains information about the internal pull-down resistors and the appropriate implementation during standby and sleep mode.

15.4.1 Internal Pulldown Resistors

The nominal value of the internal pulldown resistors internal to the PXA3xx processor family is 100 KΩ. The recommendation is not to have external pullup resistors because the value needed to overpower the internal pulldown resistance during S0/D2/C2 and S2/D3/C4 would consume excessive power during normal operation.

15.4.2 Using the Keypad Signals to Wake-up from S0/D2/C2 and S2/D3/C4 Mode

Using the appropriate Multi-Function Pin register (MFPRx), the keypad pins can be configured to minimize power usage and to detect a key press during low power mode operation. The KP_DKIN and KP_MKIN signals interfacing to the keypad controller must be configured with their pull-down resistors enabled and must be configured to detect a rising edge event. The KP_MKOUT signals used to interface to a matrix keypad must be configured with their pull-ups enabled. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. III: Graphics and Input Controller Configuration Developers Manual*, “Section 5: Keypad Controller”.

It is possible to use the keypad signals to wake up the PXA3xx processor family from S0/D2/C2 or S2/D3/C4 by configuring the “Application Subsystem Wake-Up From D2 To D0 state Enable register (AD2D0ER)” or “Application Subsystem Wake-Up From D3 Enable register (AD3ER)” registers. After wake-up, the “Application Subsystem Wake-Up From D2 to D0 Status register (AD2D0SR)” or the “Application Subsystem Wake-Up From D3 Status register (AD3SR)” to determine what unit caused the wake-up for low-power mode. The Power Manager Keyboard Edge-Detect Status register (PKSR) must be read to determine which GPIO corresponding to the keypad input signal detected an edge transition. After this event, software must follow proper procedures to re-enable the keypad registers (if required) and read the key that was pressed.

15.4.3 How to Enable Specific Combinations of Direct Keys

The enabling bit field KPC[DKN] implies that all KP_DKIN<x> signals must be enabled sequentially from KP_DKIN<0> to KP_DKIN<7>. This is true for this specific controller, but it is still possible not to map any of the direct keypad signals to a specific ball on the PXA3xx processor family package by using the PXA3xx processor family Multi-Function Pin register (MFPRx). By not mapping a specific direct input signal to the PXA3xx processor family package, a value of zero is always read in and the input is not left floating. A floating input causes excessive use of power.

For a given configuration of the PXA3xx processor family GPIO, some direct key inputs are not connected to the keypad interface if their corresponding GPIO pins are being used as inputs/outputs for other functional blocks. For example:

- GPIO pins corresponding to direct key inputs 2 and 3 are used as inputs/outputs for other blocks and hence unavailable for the keypad interface. In such a case, the KP_DKIN<3:2> input signals are guaranteed a logic 0 on them all the time, denoting no activity on direct keys 2 and 3.
- The rest of the direct keys, KP_DKIN<1:0> and KP_DKIN<7:4>, are utilized as inputs and are connected to rotary encoders or direct keys. By specifying the number of direct keys in the Keypad Control register as 6 (KPC[DKN] = 0b101), a logic 0 on direct key inputs 2 and 3 is guaranteed and no activity is detected on them.

For details about multi-function pin configuration, refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. III: Graphics and Input Controller Configuration Developers Manual*, “Section 5: Keypad Controller”.

15.4.4 Interfacing to a Matrix Keypad

When interfacing to a matrix keypad, the KP_MKIN<x> and KP_MKOUT<x> signals must be connected sequentially from KP_MKIN<0> to KP_MKIN<7> and from KP_MKOUT<0> to

KP_MKOUT<7>. The number of columns in the keypad signifies the highest order KP_MKOUT<x> signal minus one to be used. For example, KP_MKOUT<7> is the highest order KP_MKOUT<x> signal used with an eight-column keypad. The number of rows in the keypad signifies the highest order KP_MKIN<x> signal minus one to be used. For example, KP_MKIN<4> is the highest order KP_MKIN<x> signal used with a five-row keypad.

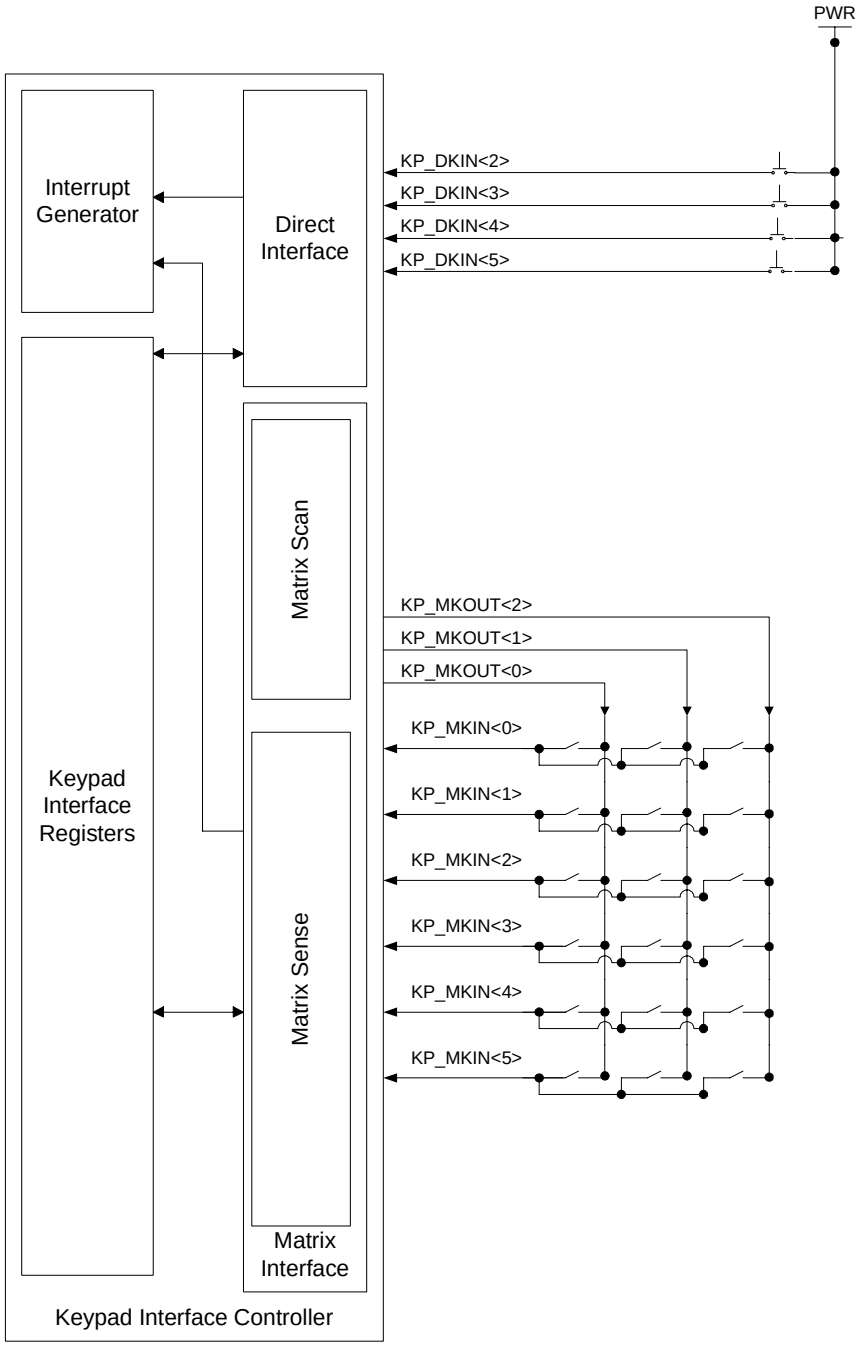
15.5 Modes of Operation Overview

The following subsection show examples of how to use the keypad controller with one, two, or no rotary encoders. Some example have different size matrix keypads.

15.5.1 Keypad Matrix and Direct Keys and No Rotary Encoder

This example shows one of many possible configurations when interfacing to the PXA3xx processor family keypad controller. This specific configuration in Figure 54 shows the keypad controller interfaced to a 3 x 6 matrix keypad of 18 keys, 4 direct keys, and no rotary encoder. The voltage reference must be at the same level as the internal voltage domain of the keypad signals. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification*.

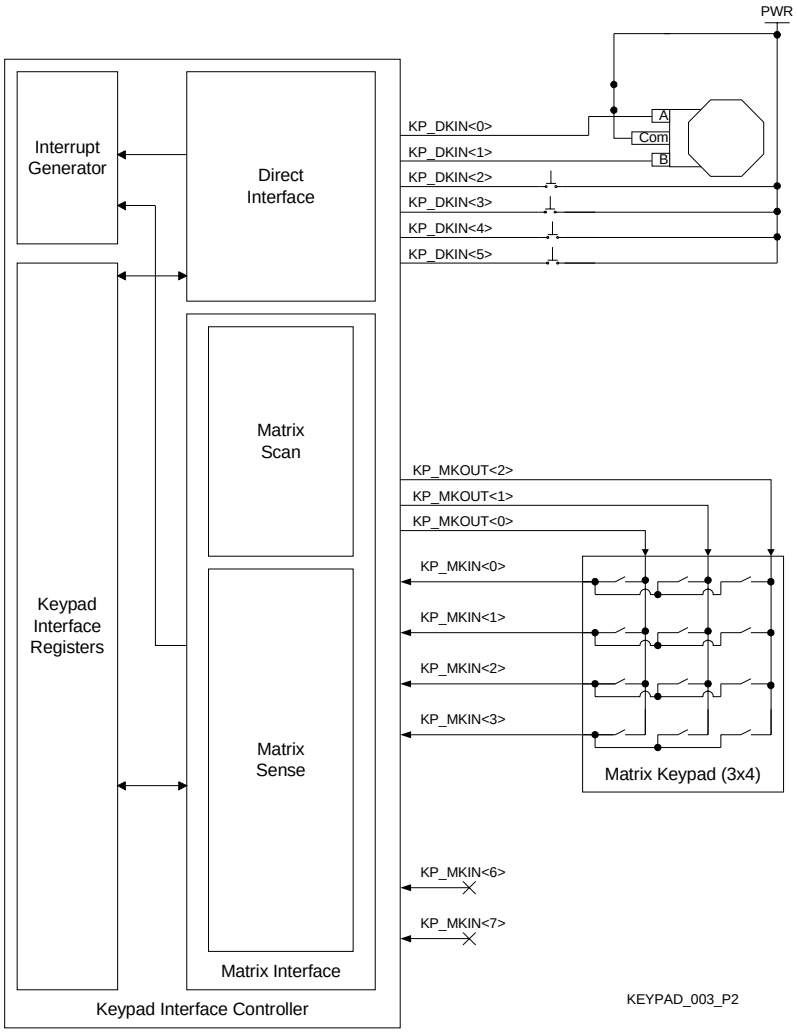
Figure 54: Keypad Interface Diagram to 6x3 Matrix Keys, 4Direct Keys, No Rotary Encoder



15.5.2 Keypad Matrix and Direct Keys with One Rotary Encoder

This example shows one of many possible configurations when interfacing to the PXA3xx processor family keypad controller. This specific configuration in Figure 55 shows the keypad controller interfaced to a 3 x 4 matrix keypad of 12 keys, 4 direct keys, and 1 rotary encoder. The voltage reference must be at the same level as the internal voltage domain of the keypad signals. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification*.

Figure 55: Keypad Matrix and Direct Keys Block Diagram (with One Rotary Encoder)



16 Quick Capture Interface

This section describes guidelines for connecting camera image sensors and sensor modules to the PXA3xx processor family Quick Capture Interface. The PXA3xx processor family supports a variety of operating modes, data widths, formats, and clocking schemes. Only a subset of these modes are described in this section. For further details, see *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. III: Graphics and Input Controller Configuration Developers Manual*, “Section 3: Quick Capture Interface”.

16.1 Overview

The Quick Capture Interface is intended for use in a PDA or mobile phone product that requires image-capture capability. Some common usage scenarios include:

- Capturing simple still images, sharing images using email, and sending images to a web photo finisher
- Using images for “pictures-as-information”
- Capturing video clips
- Providing a two-way video conference
- Performing “text imaging” (scanner/OCR)

The Quick Capture Interface modes typically include, but are not limited to:

- Image preview - full screen, with limited color, minimal lag
- Still image capture - 640x480 up to 5 megapixel resolution (PXA31x processor), up to 3 megapixel resolution (PXA30x processor)
- Video capture - 320x240 resolution
- Two-way video conference:
  - Displays own window at 1/4 of screen size
  - Displays other party’s window at full screen, 15 or 30 fps (frames per second)

16.2 Signals

See Table 53 for the list of signals used by the Quick Capture Interface.

Table 53: Signal Descriptions for Quick Capture Interface

Signal Name	Type	Description
C_DD[9:0]	Input	Data lines to transmit 8 or 10 bits at a time
C_MCLK	Output	Programmable output clock used by the camera capture sensor
C_PCLK	Input	Pixel clock used by the quick capture interface of the camera to clock the pixel data into the input FIFO
C_LV	I/O	Line start or alternate synchronization signal used by the sensor to signal line read-out or as an external horizontal synchronization
C_FV	I/O	Frame start or alternate synchronization signal used by the sensor to signal frame read-out or as an external vertical synchronization

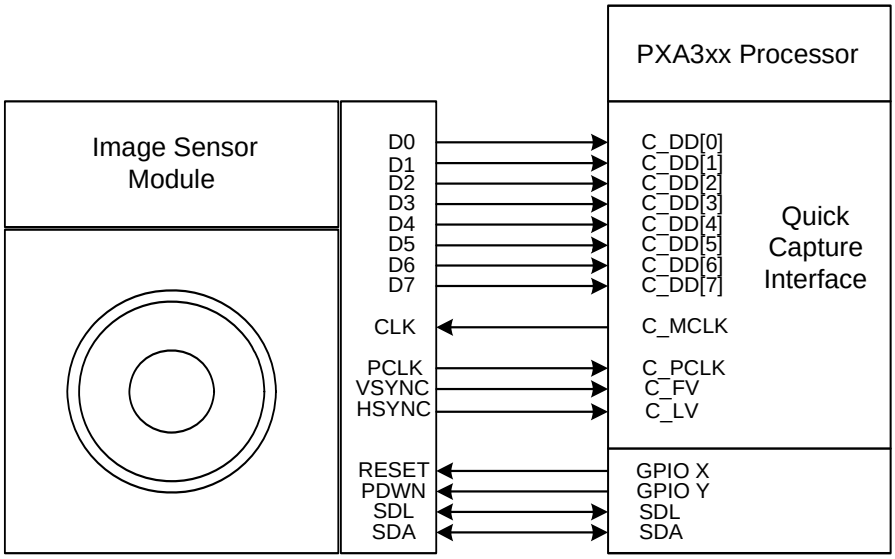
Any additional interface requirements typically are met through the use of standard GPIOs. A couple of common examples are “sensor reset” or “sensor power-down.” Signal direction is with respect to the PXA3xx processor family.

In addition to the data and data control signals, there is usually a separate interface for programming the image sensor. The most common interface used for programming and control is I^2C.

16.3 Block Diagram

See Figure 56 for illustration of a typical 8-bit master parallel connection between the PXA3xx processor family and an image-sensor module. Master mode refers to the case where the sensor module drives the line and frame-synchronization signals.

Figure 56: Block Diagram for 8-bit Master Parallel Interface



16.4 Layout Notes

The Quick Capture Interface supports frequencies up to 52 MHz for M-clock (C_MCLK) and up to 96 MHz for P-clock (C_PCLK). Take this into account when implementing the proper PCB layout for the Quick Capture Interface signals.

C_PCLK rate differs depending on which PXA3xx processor is used in the design.

17 USB 1.1 Full Speed OTG and Device Controller (PXA30x processor or PXA32x processor only)

This section describes guidelines for interfacing the PXA30x processor or PXA32x processor Universal Serial Bus (USB) full-speed device controller to a USB device controller (USB device controller = “client”) cable connector for attachment to a USB Host. Included in this chapter is a description of how to connect the PXA30x processor or PXA32x processor single-ended USB host and USB on-the-go (OTG) ports to external components.

17.1 Overview

A working knowledge of the *Universal Serial Bus Specification*, *Revision 2.0*[1] is necessary to fully understand the material contained in this section. The USB device controller is USB 1.1-compliant and supports all standard device requests issued by any USB host controller. Refer to the *Universal Serial Bus Specification*, *Revision 2.0* for a full description of the USB protocol and its operation. The USB device controller is a full-speed device that operates half-duplex at a baud rate of 12 Mbps as a device controller only, not a host or hub controller. The USB host controller referenced in this section refers to any *Universal Serial Bus Specification*, *Revision 1.1*-compliant USB host controller, including the internal USB host controller internal to the PXA30x processor or PXA32x processor.

The USB on-the-go operation is specified in the *On-The-Go Supplement to Universal Serial Bus Specification, Revision 2.0.*[2] Also, refer to *the Pull-up/Pull-down Resistors Engineering Change Notice to the USB 2.0 Specification*[3] for additional information.

17.2 Signals

The PXA30x processor or PXA32x processor have three USB1.1 ports. However, only Port 2 is available to use as a client or OTG interface. Port 2 supports differential or single-ended operation. See Table 54 for description of the USB 1.1 device controller interface signals available on the PXA30x processor or PXA32x processor. See Section 18, USB High-Speed Device Controller (U2D) (PXA30x or PXA32x processor only) for the separate USB high-speed device controller.

Table 54: USB Device Controller Interface Signals Summary

Signal Name†	Ball Name	Type	Description
USB Full Speed Transceiver Signals - Host Port 1			
USBH1_P	USBH1_P	Analog	**USB Full Speed Host Port 1 D+ Line —** this is the positive pin of the differential pair that connects to the USB full-speed host for Port 1.

1. Access the latest revision of the *Universal Serial Bus Specification, Revision 2.0* using the Internet site at: http://www.usb.org/
2. Access the latest revision of the *On-The-Go Supplement to Universal Serial Bus Specification, Revision 2.0* using the Internet site at: http://www.usb.org/
3. Access the latest revision of the *Pull-up/pull-down Resistors Engineering Change Notice to the Universal Serial Bus 2.0 Specification* using the Internet site at: http://www.usb.org/

Table 54: USB Device Controller Interface Signals Summary (Continued)

Signal Name†	Ball Name	Type	Description
USBH1_N	USBH1_N	Analog	**USB Full Speed Host Port 1 D- Line —** this is the negative pin of the differential pair that connects to the USB full-speed host for Port 1.
USBHPEN	GPIO0_2 (PXA30x processor or PXA32x processor)	Output	Controls power to the USB host Port 1.
USBHPWR	GPIO1_2 (PXA30x processor or PXA32x processor)	Input	Over-current indicator.
USB Full Speed Transceiver Signals - Port 2 Host/device controller and OTG			
USBOTG_P	USBOTG_P	Analog	**USB Full Speed Host/USB Device Controller/OTG Port 2 Positive Line** — this is the positive pin of the differential pair that connects to the USB full speed host, or the USB full speed device controller, or the USB OTG interface for port 2.
USBOTG_N	USBOTG_N	Analog	**USB Full Speed Host/USB Device Controller/OTG Port 2 Negative Line** — this is the negative pin of the differential pair that connects to the USB Full Speed Host, or the USB Full Speed device controller, or the USB OTG interface for port 2.
USBHPEN2	GPIO103 (PXA30x processor or PXA32x processor)	Output	**USB Full Speed Host Port 2 Power Control** — controls power to the USB Host port 2. This signal is muxed with USB_P2_8 but depends on the value of UP2OCR[SEOS]. Refer to *Marvell® PXA30x Processor and PXA31x Processor Vol. IV: Serial Controller Configuration Developers Manual*, “Section 3: Universal Serial Bus Host Controller” for more information.
USB Full Speed Single-Ended Signals - Port 2 Host/USB Device Controller and OTG			
USB_P2_1	GPIO101 (PXA30x processor or PXA32x processor)	Bidirectional	**USB Full Speed Host/USB Device Controller and OTG Port 2 RCV/INT/SRP** — This signal is the receive data from an external USB transceiver for port 2. When configured for an external OTG transceiver this signal is an interrupt input. When configured for an external OTG power controller and the internal transceiver, this signal is the SRP detect input.
USB_P2_2	GPIO99 or GPIO100 (PXA30x processor or PXA32x processor)	Bidirectional	**USB Full Speed Host/USB Device Controller and OTG Port 2 OE/Valid** — This signal connects to the OE signal of an external USB transceiver for USB Port 2. This signal connects to the session valid output of an external OTG power controller.

Table 54: USB Device Controller Interface Signals Summary (Continued)

Signal Name†	Ball Name	Type	Description
USB_P2_3	GPIO104 (PXA30x processor or PXA32x processor)	Bidirectional	**USB Full Speed Host/USB Device Controller and OTG Port 2 RXD-/SV —** This signal is the receive negative data line from an external USB transceiver for port 2. For an external OTG power controller, this signal is the session valid status input.
USB_P2_4	GPIO102 (PXA30x processor or PXA32x processor)	Bidirectional	**USB Full Speed Host/USB Device Controller and OTG Port 2 TXD-/SE0_VM/SRP** — This signal is the transmit negative data line to an external USB transceiver for port 2. For an external OTG transceiver, this signal is the SE0_VM bidirectional data line. For an external OTG power controller this signal is the SRP enable control output.
USB_P2_5	GPIO105 (PXA30x processor or PXA32x processor)	Bidirectional	**USB Full Speed Host/USB Device Controller and OTG Port 2 RXD+/DAT_VP/VALID40** — This signal is the receive positive data line from an external USB transceiver for port 2. For an external OTG transceiver, this signal is the DAT_VP bidirectional data line. For an external OTG power controller this signal is the 4.0V Vbus valid status input.
USB_P2_6	GPIO99 or GPIO100 (PXA30x processor or PXA32x processor)	Bidirectional	**USB Full Speed Host/USB Device Controller and OTG Port 2 TXD+ —** This signal is the positive transmit data line for an external USB transceiver for port 2.
USB_P2_7	GPIO106 (PXA30x processor or PXA32x processor)	Bidirectional	**USB Full Speed Host/USB Device Controller and OTG Port 2 Speed/OTGID** — Speed select signal for USB port 2 when configured for an external USB transceiver. Provides the OTG ID configuration using the internal transceiver.
USB_P2_8	GPIO103 (PXA30x processor or PXA32x processor)	Input	**USB Full Speed Host/USB Device Controller and OTG Port 2 Suspend** — Suspend enable for USB port 2 when configured for an external USB OTG transceiver.
USB Full Speed Single-Ended Signals - Port 3 Host			
USB_P3_1	GPIO77 (PXA30x processor or PXA32x processor)	Bidirectional	USB Full Speed Host Port 3 RCV — Receive data signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_2	GPIO78 (PXA30x processor or PXA32x processor)	Bidirectional	USB Full Speed Host Port 3 OE — Output enable signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).

Table 54: USB Device Controller Interface Signals Summary (Continued)

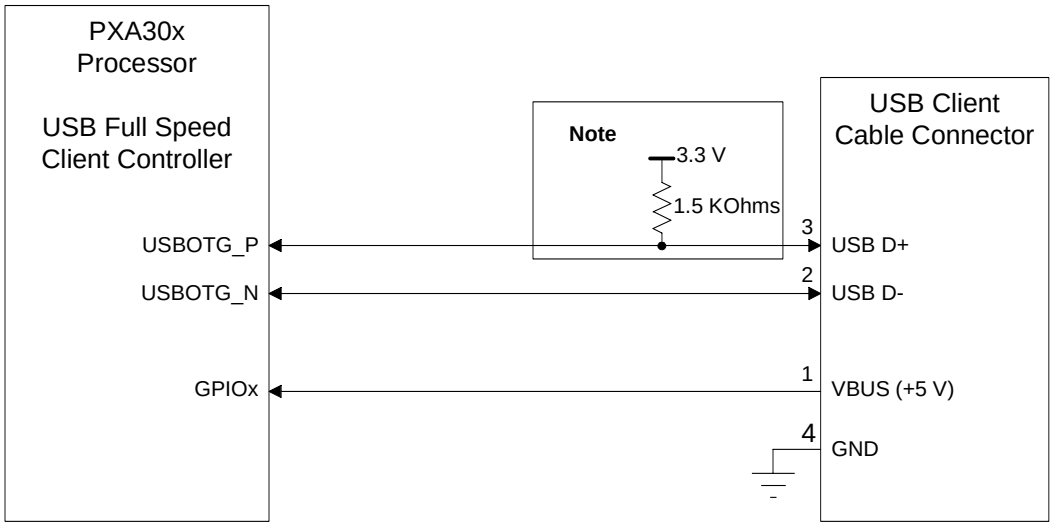
Signal Name†	Ball Name	Type	Description
USB_P3_3	GPIO79 (PXA30x processor or PXA32x processor)	Bidirectional	USB Full Speed Host Port 3 RXD– — Receive data (-) signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_4	GPIO80 (PXA30x processor or PXA32x processor)	Bidirectional	USB Full Speed Host Port 3 TXD– — Transmit data (-) signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_5	GPIO81 (PXA30x processor or PXA32x processor)	Bidirectional	USB Full Speed Host Port 3 RXD+ — Receive data (+) signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_6	GPIO82 (PXA30x processor or PXA32x processor)	Bidirectional	USB Full Speed Host Port 3 TXD+ — Transmit data (+) signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).

†USB Host-specific signals in Table 54 are greyed out. See Section 20, USB 1.1 Host Interface for a complete description of the USB 1.1 Host.

17.3 Block Diagram

See the USB device controller-interface block diagram in Figure 57.

Figure 57: USB Device Controller Interface Block Diagram



Note: Some designs use the internal pull-up resistor. The OTG transceiver contains one pull-up resistor and two pull-down resistors, on each D+ and D-, that can be enabled using the “USB Port 2 output control register (UP2OCR)” pull-up/pull-down enable bits (DPPUE and DPPDE, and DMPDE).

17.4 Layout Notes

USBOTG_P and USBOTG_N are differential pair signals; follow these layout guidelines:

- Route the signals close to each other as parallel traces on the PCB.
- Route the differential pair with trace width and spacing needed to achieve 90 Ohms differential impedance. Deviations will normally occur due to package breakout and routing to connector pins. Keep the amount and length of width/spacing deviations to a minimum.
- Do not route USB traces under crystals, oscillators, magnetic devices or IC’s that use and/or duplicate clocks.
- Avoid stubs on USB signals as the stubs will cause reflections and affect signal quality.
- Route USB signals over continuous planes (VCC or GND) with no interruptions. Avoid crossing plane splits if possible.
- Minimize the length of parallel running signals, especially clocks to avoid crosstalk. Minimum suggested spacing to clocks is 50 mils. Minimum suggested spacing to other signals is 20 mils.
- Match the trace lengths as closely as possible (within ±0.5 inches (12.7 mm)).

This section includes layout notes for self-powered USB device controllers and for bus-powered USB device controllers.

Power-supply considerations must be made when using GPIO alternate functions. Not all USB signals are powered from the same power source. See Table 55 for the power source required depending on signals being used. Rows shaded in grey are dedicated for USB host port 1 and host

port 3 controllers. See Section 20, USB 1.1 Host Interface for specific design guidelines for USB host ports 1 and 3.

Table 55: USB Power Supplies

Power Supply	Signal Group[1,2]
VCC_BIAS (Analog signals, PXA300 processor only) VCC_ULPI (Analog signals, PXA31x processor only) VCC_IO1 (Digital signals)	USB Full Speed Transceiver Differential Signals - Host Port 1
VCC_BIAS (PXA300 processor only) VCC_ULPI (PXA31x processor only)	USB Full Speed Transceiver Differential Signals - Port 2 Host/Client and OTG
VCC_IO2	USB Full Speed Single-Ended Signals - Port 2 Host/Client and OTG
VCC_MSL	USB Full Speed Single-Ended Signals - Port 3 Host

NOTE:
1. See Table 54 for signals within each groups.
2. USB Host signals in Table 55 are greyed out. See Section 20, USB 1.1 Host Interface for a description of USB Host signals.

17.4.1 Self-Powered Devices

Figure 58 and Figure 59 show two similar USB interface-connection diagrams for self-powered devices. USB D+ connects directly to the PXA30x processor or PXA32x processor USBOTG_P signal and USB D- connects directly to the USBOTG_N signal. The USBOTG_P and USBOTG_N signals should match the impedance of a USB cable, 90 Ω, without the use of external series resistors. The USBOTG_P/USBOTG_N differential pair should be designed to meet the 90 Ω differential impedance specification. The 0 Ω resistors are optional to compensate for minor differences between the USB cable and the board-trace impedances.

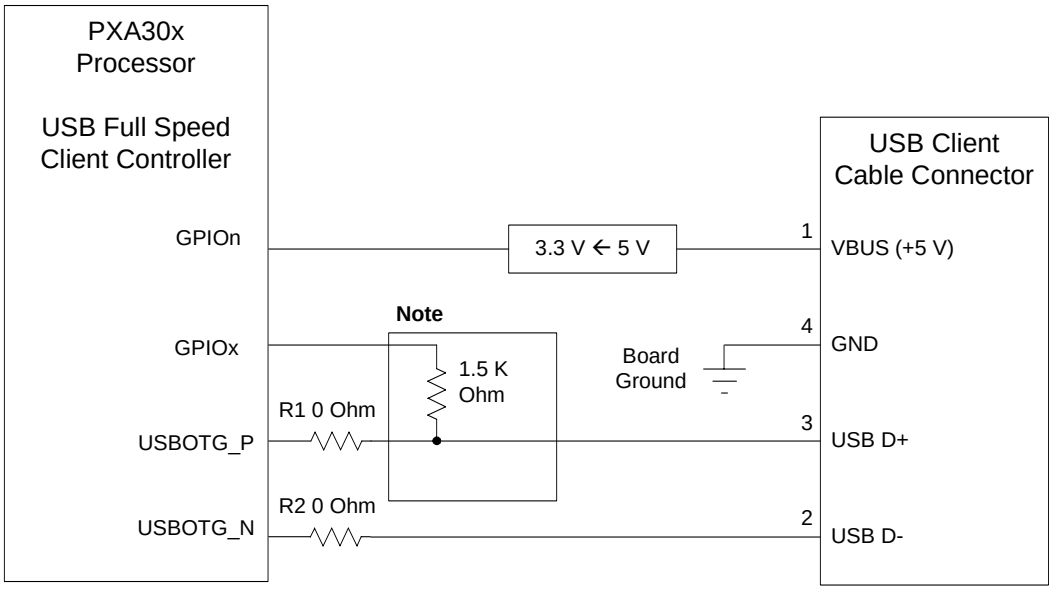
The 5.0 V-to-3.3 V voltage divider shown in Figure 58 and Figure 59 is required since the PXA30x processor or PXA32x processor GPIO pins cannot exceed 3.3 V. This voltage divider is implemented in a number of ways:

- One option is to use a MAX6348 (or equivalent) power-on-reset device. This solution produces a very clean signal edge and minimizes signal bounce.
- Another solution is to use a 3.3 V line buffer with 5.0 V-tolerant inputs. This solution does not reduce signal bounce; thus, software must compensate by reading the GPIO signal after it stabilizes.
- The last solution is to implement a signal-bounce minimization circuit that is 5.0 V-tolerant, but produces a 3.3 V signal to the GPIO signal.

17.4.1.1 Operation if GPIOn and GPIOx are Different Pins

See Figure 58 for an illustration of a self-powered USB client device using two different GPIO signals for USB cable detection.

Figure 58: Self-Powered Device when GPIOn and GPIOx are Different Pins



Note: Some designs use the internal pull-up resistor.

USBC_002_P2

Any GPIO signal is defined as GPIOn and GPIOx; however, GPIOn must be a GPIO that brings the PXA30x processor or PXA32x processor out of reduced-power modes. Out of reset, software configures GPIOx as an input to allow the USB D+ signal to float. GPIOn is configured as an input and programmed to cause an interrupt whenever a rising or falling edge is detected. Software must read the GPIOn pin when an interrupt occurs to determine if a USB cable is connected or not. GPIOn is driven high when a cable is connected and returns to a low state if the cable is disconnected. If a USB cable connect is detected, then software enables the USB client peripheral and drives the GPIOx signal high. This results in USB D+ being pulled high using the 1.5 KΩ resistor to indicate to the host controller that a fast USB client device is connected per the *Universal Serial Bus Specification*, *Revision 2.0*.

The configuration of using separate GPIOs for GPIOn and GPIOx lets the PXA30x processor or PXA32x processor enter S2/D3/C4 or S3/D4/C4 mode with a USB cable connected or disconnected to the USB client interface. Before entering S2/D3/C4 or S3/D4/C4 mode, the software must read the GPIOn pin to determine if a USB cable is connected.

If a USB cable is not detected, the software must configure the GPIOx pin as an input to allow the USB D+ line to float, and configure the GPIOn pin to detect a wake-up event. The PXA30x processor or PXA32x processor is then put into S2/D3/C4 or S3/D4/C4 mode.

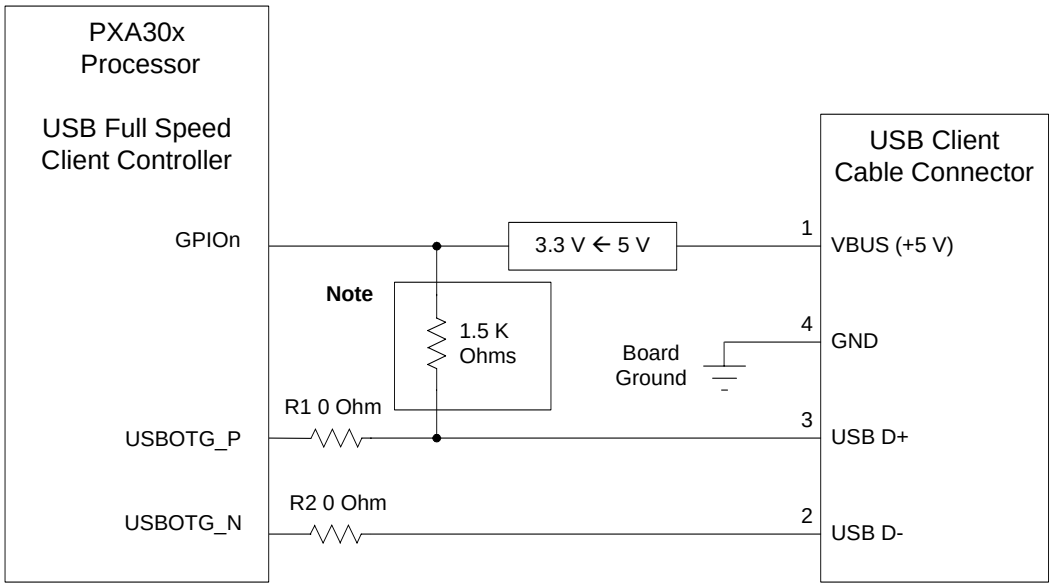
If a USB cable is detected, software must configure the GPIOx pin as an input to allow the USB D+ line to float. If the processor is initiating a low-power mode, this needs to be communicated to USB via a USBSuspend sequence. The PXA30x processor or PXA32x processor is then put into S2/D3/C4 or S3/D4/C4 mode.

When the PXA30x processor or PXA32x processor wakes up, software must drive the GPIOx signal high pulling USB D+ igh indicating to the host controller that a full-speed device has connected.

17.4.1.2 Operation if GPIOn and GPIOx are the Same Pin

See Figure 59 for illustration of a self-powered USB client device using one GPIO signal for detection.

Figure 59: Self-Powered Device when GPIOn and GPIOx are Same Pins



Note: Some designs use the internal pull-up resistor.

USBC_003_P2

Out of reset, GPIOn is configured as an input and programmed to cause an interrupt whenever a rising or falling edge is detected. Software must read the GPIOn pin when an interrupt occurs to determine if a USB cable is connected. GPIOn is driven high when a cable is connected and returns to a low state if the cable is disconnected. If a USB cable is connected, USB D+ is pulled high using the 1.5 KΩ resistor indicating to the host controller that a full-speed device has connected. Software must enable the USB client peripheral before the host sends the first USB command.

Caution

When the single GPIOn/x configuration is in use, never place the PXA30x processor or PXA32x processor into S2/D3/C4 or S3/D4/C4 mode while the USB cable is connected to the USB client interface for these reasons:

- During S2/D3/C4 or S3/D4/C4 modes, the USB controller is in reset and does not respond to the host.
- After S2/D3/C4 or S3/D4/C4 mode, the USB client does not respond to its host-assigned address because the previously assigned address is not retained during S2/D3/C4 or S3/D4/C4 modes.

Use the following sequence to enter S2/D3/C4 mode when GPIOn and GPIOx are the same signal. Software must:

1. Read the GPIOn pin to verify a USB cable is not connected.
2. Configure the GPIOn pin to detect a wake-up event.

3. Enter S2/D3/C4 mode.

When a USB cable is attached and detected, and the host controller notified as explained in the caution above, the host assigns a USB address for the USB client using the USB “reset” command.

17.4.2 Bus-Powered Device

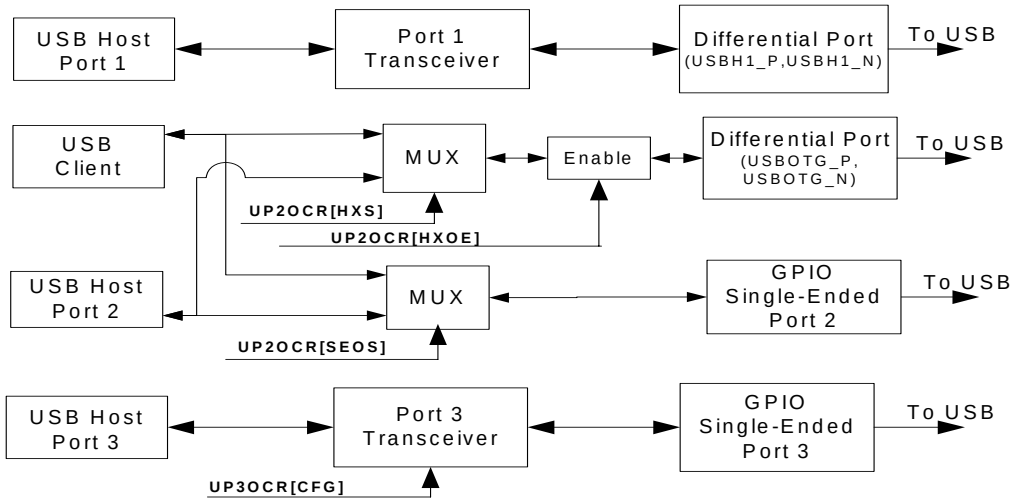
The PXA30x processor or PXA32x processor supports a bus-powered device model. Check the *Marvell® PXA30x Processor and PXA31x Processor Vol. IV: Serial Controller Configuration Developers Manual* for details. Use UDCCR[PWRMD] register bit to configure the USB device controller as self-powered or bus-powered. Refer to the *Universal Serial Bus Specification*, *Revision 2.0* for rules regarding bus-powered device current draw.

17.4.3 USB On-the-Go Transceiver Usage

See Figure 60 for an illustration of each of the configurations provided to support USB on-the-go (OTG) operation. Each of the configurations are described in detail.

Note The PXA30x processor or PXA32x processor does not provide direct connection to the USB Vbus and control of the USB Vbus.

Figure 60: USB OTG Configurations



The USB host port 2 transceiver is designed in accordance with the *Pullup/Pulldown Resistors Engineering Change Notice to the USB 2.0 Specification* to provide on-chip resistors and OTG compliant transceiver operation. The USB host controller port 2 multiplexor is a bidirectional I/O mux that connects to the USB host port 2 transceiver and the single-ended I/O through the GPIO. The port 2 multiplexor provides an interface that allows the USB device control (UDC) port or USB host control (UHC) port 2 to connect to the UHC port 2 transceiver for direct bi-directional connection to

the USB. The port 2 multiplexor also provides an interface that allows the UDC port, UHC port 2, and the UHC port 3 to connect to single-ended I/O through the GPIOs.

The PXA30x processor or PXA32x processor OTG transceiver consists of one pullup resistor and two pull-down resistors on each D+ and D-.

- The resistors are enabled using the “USB Port 2 Output Control Register (UP2OCR)” pull-up/pull-down enable bits *(DPPUBE, DMPUBE, DPPUE, DMPUE, DPPDE, DMPDE)*. See Figure 61 for illustration of the on-chip host port 2 transceiver pad with the pull-up and pull-down resistors.
- As shown in Figure 61, SW3 is enabled for both D+ and D- when host port 2 is being used for USB host-controller data.
- As shown in Figure 61, SW1 on the D+ pad is enabled and SW1 on the D- pad is disabled when host port 2 is being used for USB device-controller data.
- As shown in Figure 61, SW2 on the D+ and D- pads is disabled when host port 2 is being used for USB device-controller data, but SW2 is enabled and disabled by hardware when the UDC is idle and receiving data from an upstream device as specified in the *Pullup/Pulldown Resistors Engineering Change Notice to the USB 2.0 Specification*.

See Table 56 for the list of switch settings used for the USB host and USB device controller I/O.

Figure 61: Host Port 2 OTG Transceiver

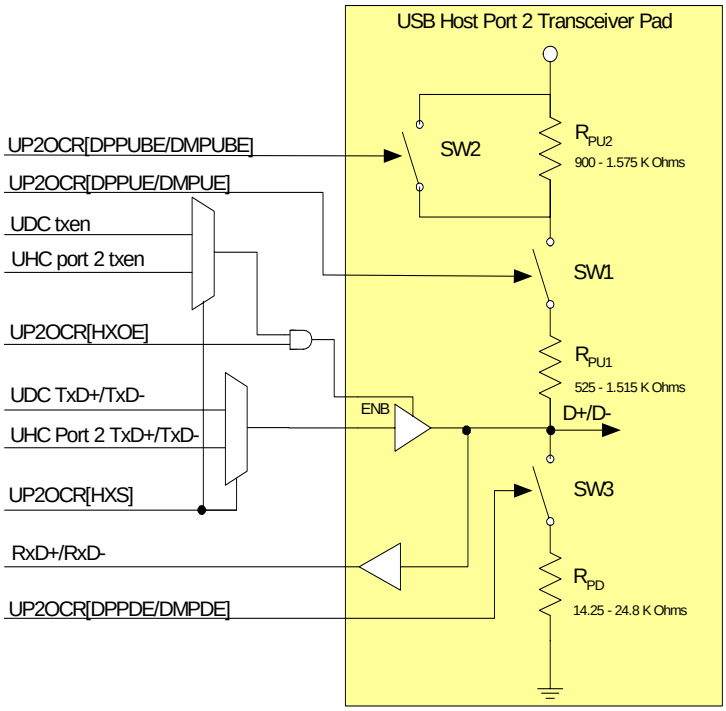


Table 56: Host Port 2 OTG Transceiver Switch Control Settings

Controller Selected	D+ Transceiver			D- Transceiver		
	SW1†	SW2†	SW3†	SW1†	SW2†	SW3†
USB Host	Disabled	Disabled	Enabled	Disabled	Disabled	Enabled
USB Device	Enabled	Hardware controlled	Disabled	Disabled	Disabled	Disabled
† SW1, SW2 and SW3 refer to the switches shown in Figure 61.						

Note

See the *Marvell® PXA30x Processor and PXA31x Processor Vol. IV: Serial Controller Configuration Developers Manual for* programming requirements for disabling and enabling the transceiver and pullup/pulldown resistors for USB Host Port 2 and for S2/D3/C4 and S0/D2/C2 mode operation.

17.4.4 Interface to External Transceiver (OTG)

If the internal OTG transceiver is not used, the USB device controller (UDC) contains control, status, and interrupt registers to provide seamless interface to external transceivers.

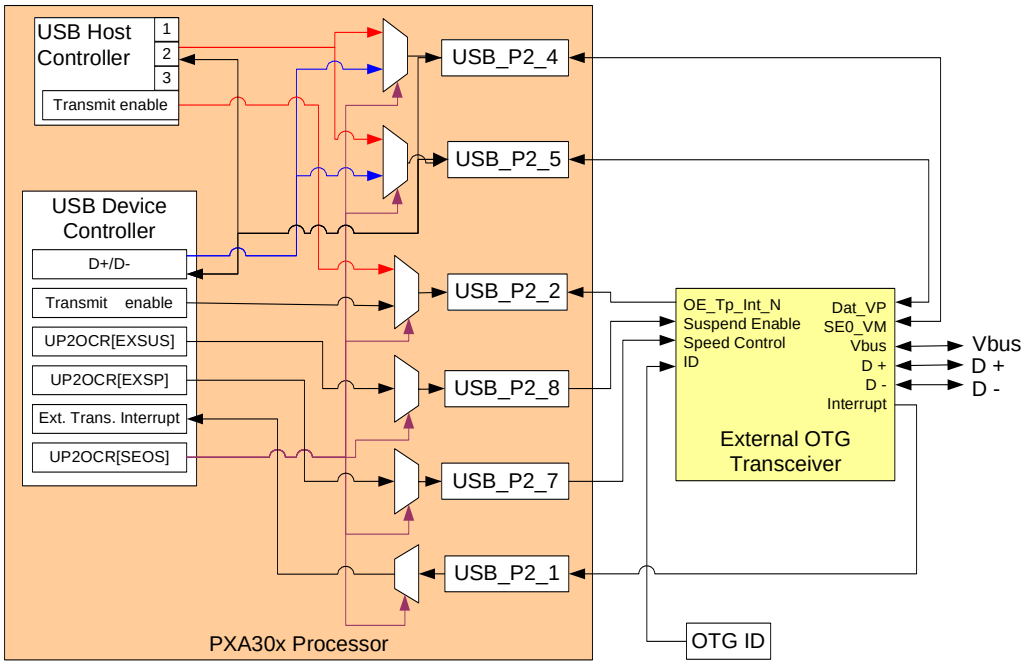
External transceivers provide D+, D- and Vbus driver to the USB cable. In this mode, the D+ and D- signals of the USB are output through GPIO pads with the “USB Port 2 Output Control Register (UP2OCR)” [SE0S] using the control multiplexors to select between UDC and D+, D-, and transmit-enable signals of USB host controller.

In addition, “USB Port 2 Output Control Register (UP2OCR)” provides the following:

- External transceiver suspend (EXSUS) control output bit
- External transceiver speed (EXSP) control output bit
- External transceiver interrupt input to interface to the external transceiver.

See Figure 62 for an illustration of the PXA30x processor or PXA32x processor OTG connections to an external transceiver.

Figure 62: Connection to External OTG Transceiver



17.4.5 Interface to External Charge Pump Device (OTG)

In addition to interface options to internal and external OTG transceivers, the UDC provides control outputs and interrupt inputs to drive and monitor an external charge pump device. In this mode, D+ and D- signals of the USB are output using the on-chip OTG transceiver and the Vbus interface is provided by an external charge pump device. In this mode, the HXS bit in “USB Port 2 Output Control Register (UP2OCR)” uses the control multiplexors to select between the UDC and D+, D-, and transmit-enable signals of the USB host controller. These signals are output through the USB host controller port 2 transceiver.

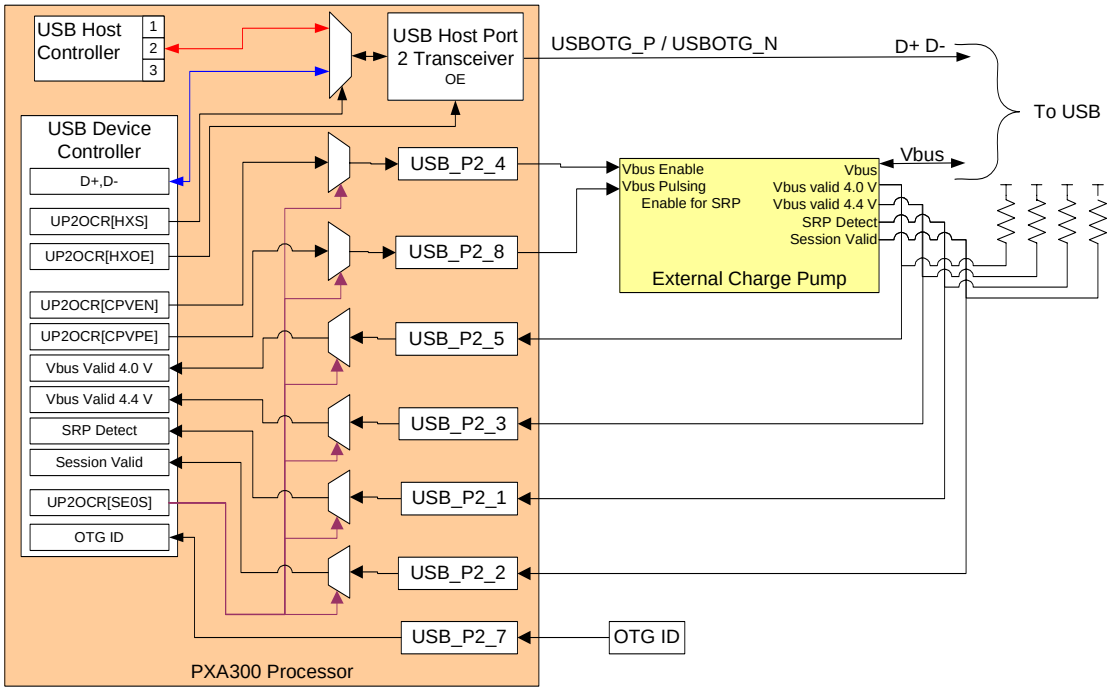
In addition, to enable the driving of Vbus and the driving of pulses on Vbus, the “USB Port 2 Output Control Register (UP2OCR)” provides (1) the charge pump Vbus-enable (CPVEN) control-output bit and (2) the charge pump Vbus pulse-enable (CPVPE) control-output bit.

Additionally, “USB Port 2 Output Control Register (UP2OCR)” provides the following inputs to interface to the external charge pump device:

- Vbus valid 4.0
- Vbus valid 4.4
- Session valid
- Session-request protocol (SRP) detected interrupt

See Figure 63 for illustration of the PXA30x processor or PXA32x processor OTG connections to an external charge pump device.

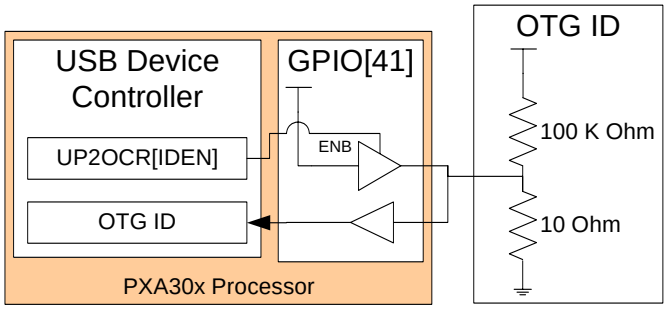
Figure 63: Connection to External OTG Charge Pump



17.4.6 OTG ID

The UDC provides OTG ID interface support through USB_P2_7 on GPIO[104]. The UDC provides "USB Port 2 Output Control Register (UP2OCR)" ID output enable (IDEN) to enable the output for OTG ID reading and the OTG ID interrupt input to detect changes in the OTG ID signal. When IDEN is set to 1, it enables the output of GPIO[104] and driven high with a weak output driver. Therefore, when the OTG ID pin on the USB connector is connected to a 100 KΩ resistor, the OTG ID input results in 1; when the OTG ID pin is connected to a 10 Ω resistor to ground, the GPIO[104] output driver is unable to drive OTG ID to a 1, resulting in the OTG ID input being a 0. See Figure 64 for illustration of the PXA30x processor or PXA32x processor interface to OTG ID.

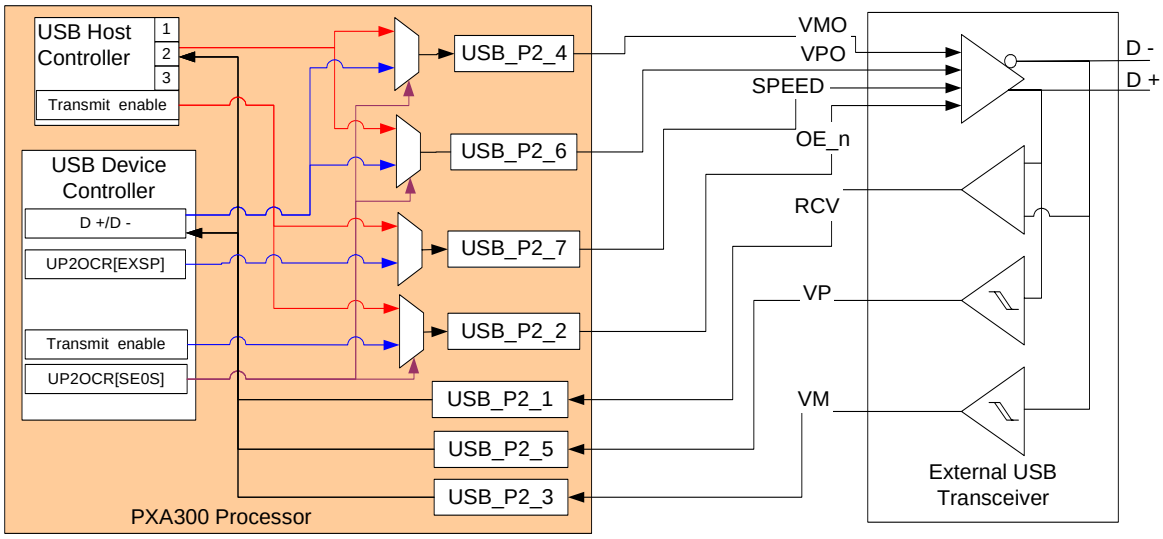
Figure 64: Connection to OTG ID



17.4.7 Interface to External USB Transceiver (non-OTG)

In addition to the OTG interfaces to external transceiver and charge pump devices, the UDC and USB host controller interfaces to a non-OTG external USB transceiver through the single-ended interface with the GPIOs. In this mode, the GPIOs provide unidirectional connections to an external transceiver and the external transceiver provides bidirectional connections for D+ and D- to the USB. This mode is selected when single-ended zero (SEO) in the “USB Port 2 Output Control Register (UP2OCR)” is set to 2 or 3. See Figure 65 for illustration of the PXA30x processor or PXA32x processor connection to an external USB transceiver.

Figure 65: PXA30x and PXA32x Processor Connection to External USB Transceiver



See Table 57 and Table 58 for definitions associated with the possible combinations of data while using the external USB transceiver mode.

Table 57: Output to External USB Transceiver

P2_6/P3_6	P2_4/P3_4	Result
0	0	SE0
0	1	Logic “0”
1	0	Logic “1”
1	1	“SE1” (not allowed)

Table 58: Input from External USB Transceiver

P2_5/P3_5	P2_3/P3_3	Result
0	0	“SE0”
0	1	Low Speed
1	0	Full Speed
1	1	“SE1” (not allowed)

18 USB High-Speed Device Controller (U2D) (PXA30x or PXA32x processor only)

This chapter describes guidelines for connecting the PXA30x or PXA32x processor Universal Serial Bus 2.0 (USB) interface to a USB 2.0 Transceiver Macrocell Interface (UTMI) transceiver interface. The UTMI transceiver then connects to a USB 2.0 device controller cable connector for attachment to a USB network. For further details, see *Marvell® PXA30x and PXA32x Processors Vol. IV: Serial Controller Configuration Developers Manual*.

18.1 Overview

A working knowledge of the *Universal Serial Bus Specification*, *Revision 2.0*[1] is necessary to fully understand the material contained in this chapter. The USB device controller is USB 2.0-compliant. Also, refer to *the Pull-up/Pull-down Resistors Engineering Change Notice to the USB 2.0 Specification*[2] for additional information.

18.2 Signals

See Table 59 for description of the UTMI interface signal descriptions.

Table 59: UTMI Interface Signals Summary

Signal Name	Type	Signal Descriptions
UTM_CLK	Input	**UTMI Clock—**Connect to the clock output of an external UTMI transceiver.
U2D_DATA<7:0>	Bidirectional	**UTMI Data Bus—**Connect to the data bus of an external UTMI transceiver.
U2D_RESET	Output	**UTMI Reset—**Connect to the reset input of an external UTMI transceiver.
U2D_XCVR_SELECT	Output	**UTMI Transceiver Select—**Connect to the external UTMI transceiver input that selects high-speed and full-speed operating modes. The full-speed transceiver should be enabled when this signal is high.
U2D_TERM_SELECT	Output	**UTMI Termination Select—**Connect to the external UTMI transceiver input that selects high speed and full speed termination modes. The full speed termination should be enabled when this signal is high.
U2D_SUSPENDM_X	Output	**UTMI Suspend**—Connects to the external UTMI transceiver input that goes low to place the transceiver in a mode that draws minimal power from supplies while retaining the capability for suspend/resume operation.

1. Access the latest revision of the *Universal Serial Bus Specification, Revision 2.0* using the Internet site at: http://www.usb.org/
2. Access the latest revision of the *Pull-up/pull-down Resistors Engineering Change Notice to the Universal Serial Bus 2.0 Specification* using the Internet site at: http://www.usb.org/

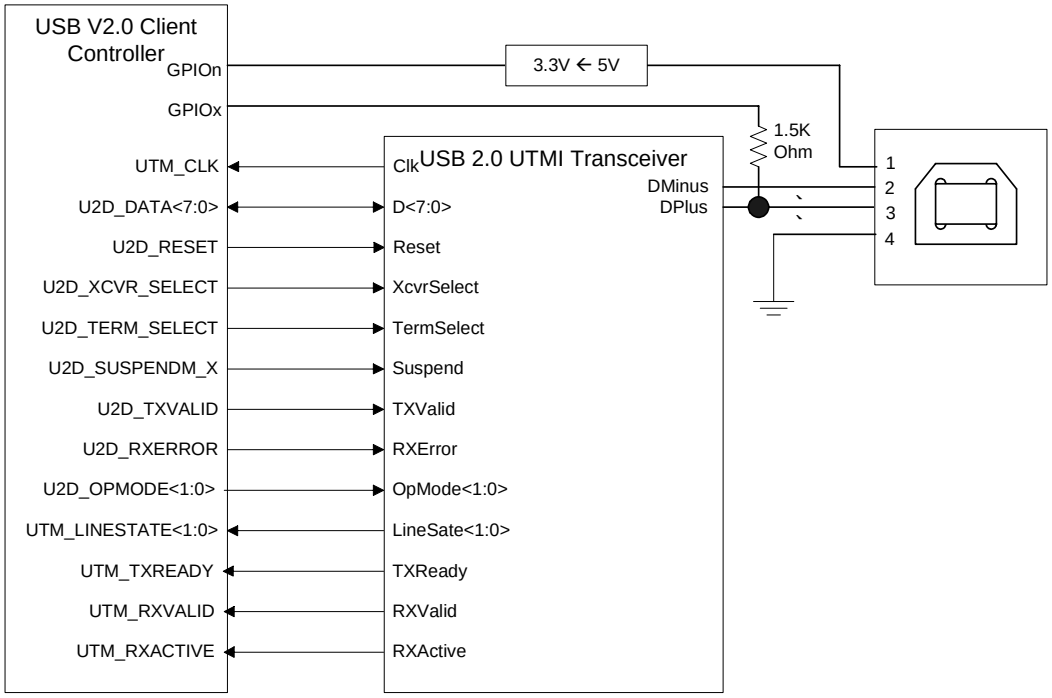
Table 59: UTMI Interface Signals Summary (Continued)

Signal Name	Type	Signal Descriptions
UTM_LINESTATE<1:0>	Input	**UTMI Line State**—Connects to the single-ended receiver status signals. They are asynchronous until a usable CLK is available, then they are synchronized to CLK. They directly reflect the current state of the DP (LineState[0]) and DM (LineState[1]) signals: D– D+ Description 0 0 SE0 0 1 “J” State 1 0 “K” State 1 1 SE1
U2D_TXVALID	Output	**UTMI Transmit Valid—**Indicates that the transmit output data is valid.
UTM_TXREADY	Input	**UTMI Transmit Data Ready**—If this signal is asserted, the controller will have data available for clocking in to the TX Holding register on the rising edge of UTM_CLK.
UTM_RXVALID	Input	**UTMI Receive Data Valid**—Indicates that the data bus has valid data.
UTM_RXACTIVE	Input	**UTMI Receive Active**—Indicates that the receive state machine has detected SYNC and is active.
U2D_RXERROR	Output	**UTMI Receive Error**—Connects to the transceiver error output. High indicates that a receive error has been detected.
U2D_OPMODE<1:0>	Output	**UTMI Operating Mode**—These signals configure the transceiver operating modes: OPMODE<1:0> Description 0 0 Normal Operation 0 1 Non-Driving 1 0 Disable Bit Stuffing and NRZI encoding 1 1 Reserved

18.3 Block Diagram

Figure 66 shows the U2D block diagram for the PXA30x or PXA32x processor using the UTMI interface.

Figure 66: USB 2.0 Device Controller Interface Block Diagram



Any GPIO signal can be defined as GPIOn or GPIOx; however, the 1.5 KΩ pull-up direct connect to VBUS may not be required depending on the type of transceiver being used. If either or both are required, then out of reset software must configure GPIOx as an input to allow the DPlus signal to float. GPIOn is configured as an input and programmed to cause an interrupt whenever a rising or falling edge is detected. Software must read the GPIOn pin when an interrupt occurs to determine whether a USB cable is connected. GPIOn is driven high when a cable is connected and returns to a low state if the cable is disconnected. If a USB cable connect is detected, then software enables the USB device controller peripheral and drives the GPIOx signal high. This results in DPlus being pulled high using the 1.5 KΩ resistor to indicate to the host controller that a fast USB device controller device is connected per the *Universal Serial Bus Specification*, *Revision 2.0*.

The 5.0 V-to-3.3 V voltage divider is required because the PXA30x or PXA32x processor GPIO pins cannot exceed 3.3 V. This voltage divider is implemented in a number of ways:

- The most robust and expensive solution is to use a MAX6348 (or equivalent) power-on-reset device. This solution produces a very clean signal edge and minimizes signal bounce.
- A less expensive solution is to use a 3.3 V line buffer with 5.0 V-tolerant inputs. This solution does not reduce signal bounce, thus software must compensate by reading the GPIO signal after it stabilizes.
- Another solution is to implement a signal-bounce minimization circuit that is 5.0 V-tolerant, but produces a 3.3 V signal to the GPIO signal.

18.4 Layout Notes

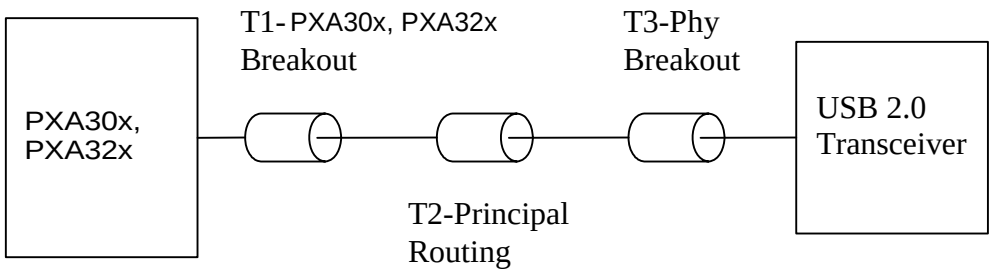
Refer to the PHY data sheet for additional guidelines and layout notes. Some of the PHY devices may have strict requirements which must be adhered too.

DPlus and DMinus are differential pair signals. Follow these layout guidelines:

- Route the signals close to each other as parallel traces on the PCB.
- Route the differential pair with trace width and spacing needed to achieve 90 ohms differential impedance. Deviations will normally occur due to package breakout and routing to connector pins. Keep the amount and length of width/spacing deviations to a minimum.
- Route high-speed USB signals using a minimum of vias and corners. When a corner is necessary, use two 45 degree turns or an arc instead of a single 90 degree turn. This reduces signal reflections and impedance discontinuities.
- Do not route USB traces under crystals, oscillators, magnetic devices or IC's that use and/or duplicate clocks.
- Avoid stubs on USB signals as the stubs will cause reflections and affect signal quality.
- Route USB signals over continuous planes (VCC or GND) with no interruptions. Avoid crossing plane splits if possible.
- Minimize the length of parallel running signals, especially clocks to avoid crosstalk. Minimum suggested spacing to clocks is 50 mils. Minimum suggested spacing to other signals is 20 mils.
- Match the trace lengths for USB high speed signals as closely as possible (within ±0.150 inches (3.81 mm)).

Refer to Figure 67 and the following sections for general routing guidelines.

Figure 67: General Signal Routing Diagram - Applicable to All Signals



Compatible USB UTMI (Universal Transceiver Macrocell Interface) transceivers are: SMSC USB3280, Cypress CY7C68000.

18.4.1 Signal Routing Guidelines

18.4.1.1 Data Signals (Data 0 through Data 7)*

T1- Microstrip construction, 50 Ω <= Z0 <= 120 Ω**, 0 mils <= length <= 500 mils, minimum spacing of 3 mils.***

T2- Either microstrip or stripline construction, 40 Ω <= Z0 <= 80 Ω**, 0 mils <= length <= 3000 mils, minimum spacing 6 mils.***

T3- Microstrip construction, 50 Ω <= Z0 <= 120 Ω**, 0 mils <= length <= 500 mils, minimum spacing 3 mils.***

Total Length: 0 mils <= T1+T2+T3 <= 3000 mils

18.4.1.2 Control Signals (TXREADY, TXVALID, RXACTIVE, RXVALID, RXERROR)*

T1- Microstrip construction, 50 Ω <= Z0 <= 120 Ω**, 0 mils <= length <= 500 mils, minimum spacing 3 mils.***

T2- Either microstrip or stripline construction, 40 Ω <= Z0 <= 80 Ω**, 0 mils <= length <= 2000 mils, minimum spacing 6 mils.***

T3- Microstrip construction, 50 Ω <= Z0 <= 120 Ω**, 0 mils <= length <= 500 mils, minimum spacing 3 mils.***

Total Length: 0 mils <= T1+T2+T3 <= 2000 mils

18.4.1.3 Clock*

T1- Microstrip construction, 50 Ω <= Z0 <= 120 Ω**, 0 mils <= length <= 500 mils, minimum spacing 3 mils.***

T2- Either microstrip or stripline construction, 40 Ω <= Z0 <= 80 Ω**, 0 mils <= length <= 1400 mils, minimum spacing 6 mils.***

T3- Microstrip construction, 50 Ω <= Z0 <= 120 Ω**, 0 mils <= length <= 500 mils, minimum spacing 3 mils.***

Total Length: T1+T2+T3 = 1400 mils +/- 100 mils

Note

*Trace routing layer changes should only occur when transitioning from one trace segment to the next (T1, T2, T3), but are not required.

**Controlled impedance boards should have upper and lower impedance limits that fit within the impedance bound specified. For example, 50 Ω +/- 10%= 45 Ω to 55 Ω, which fits within the range of 40 Ω <= Z0 <= 60 Ω.

***Dielectric thickness (h) is defined as the height above/below the nearest reference plane.

18.4.2 Power Distribution Routing Guidelines

The USB 2.0 and core grounds can be tied to a common ground plane.

- Keep USB 2.0 supply separate from other IO supplies.
- A plane is recommended for USB 2.0 Supply. If plane is not possible to implement, the USB 2.0 power supply trace needs to follow the requirements:
  - Loop inductance <= 2 nH
  - Loop resistance <= 10 mΩ
  - Max Length 2”
  - Trace width >= 250 mils
  - Minimum 4 mil separation between power trace/plane and GND plane.
- Route the power trace/plane with direct reference to the ground plane. The referred portion of ground plane should have no discontinuities. For example, slot on the ground below the power supply routing is not allowed.

- Do not leave any PXA30x or PXA32x processor power supply or ground balls floating.
- Minimize the IR drop between PMIC and BGA. Ensure that VCC level at the PXA30x or PXA32x processor balls is within the VCC tolerance specified in the PXA30x or PXA32x processor specification.
- At least 1 via for each power/ground ball is required.

18.4.3 Power Decoupling Guidelines

- 1 x 20 µF Bulk Decoupling Capacitor. Recommend 0805 or 1206 size capacitor.
- 3 x 1.0 µF, 0402 edge decoupling capacitors are recommended.
- F in total. Recommend X6S dielectric, ESL 500 pH, ESR 11.2 mΩ, TDK part # C1005X5ROJ105 or equivalent.
- Edge decoupling capacitors should be placed within 250 mils from the edge of PXA30x or PXA32x processor.
- Capacitors should be connected to power and ground using shortest possible trace and vertically stacked vias.
- Use separate trace and via for each decoupling capacitor. I.e., never route multiple capacitors through the same via.

19 USB High-Speed Controller (U2D) with OTG Functionality (PXA31x processor only)

This section describes the connection guidelines for the PXA31x processor U2D interface and discusses the option of combining the U2D with the USB Host Controller (UHC) for an OTG solution. The U2D can operate in either high-speed or full-speed mode. The PXA31x processor utilizes the UTMI+ Low Pin Count Interface (ULPI). For more information on the various PHY interface standards, refer to the ULPI and UTMI industry specifications.

Table 60: U2D ULPI Interface Signals Summary

Name	Direction	Description
ULPI_CLK	Input	Clock. This input from the PHY is used for clocking receive and transmit parallel data. 60 MHz HS/FS, with 8-bit interface
ULPI_DIR	Input	Controls the direction of the data bus. When the PHY has data to transfer to the Link, it drives ulpi_dir high to take ownership of the bus. When the PHY has no data to transfer it drives ulpi_dir low and monitors the bus for commands from the Link. The PHY will pull ulpi_dir high whenever the interface cannot accept data from the Link, such as during PLL startup.
ULPI_NXT	Input	The PHY asserts ulpi_nxt to throttle the data. When the Link is sending data to the PHY, ULPI_NXT indicates when the current byte has been accepted by the PHY. The Link places the next byte on the data bus in the following clock cycle.
ULPI_DATA<7:0>	Bidirectional	8-bit data bus to the PHY
ULPI_STP	Output	The Link asserts ULPI_STP for one clock cycle to stop the data stream currently on the bus. If the Link is sending data to the PHY, ULPI_STP indicates the last byte of data was on the bus in the previous cycle.

19.1 OTG Usage

The PXA31x processor can be programmed to act as a USB OTG controller by combining the UHC and U2D functionality. Internal multiplexing capabilities allow the two separate controllers to drive one set of USB ports for communicating with other OTG devices. In this mode, the PXA31x processor provides full & low-speed host, and high & full-speed device capabilities. UHC Port 2 is used to achieve the host functionality. To take advantage of this feature, the ULPI transceiver must support the 6-pin or 3-pin serial modes.

Figure 68: OTG Block Diagram

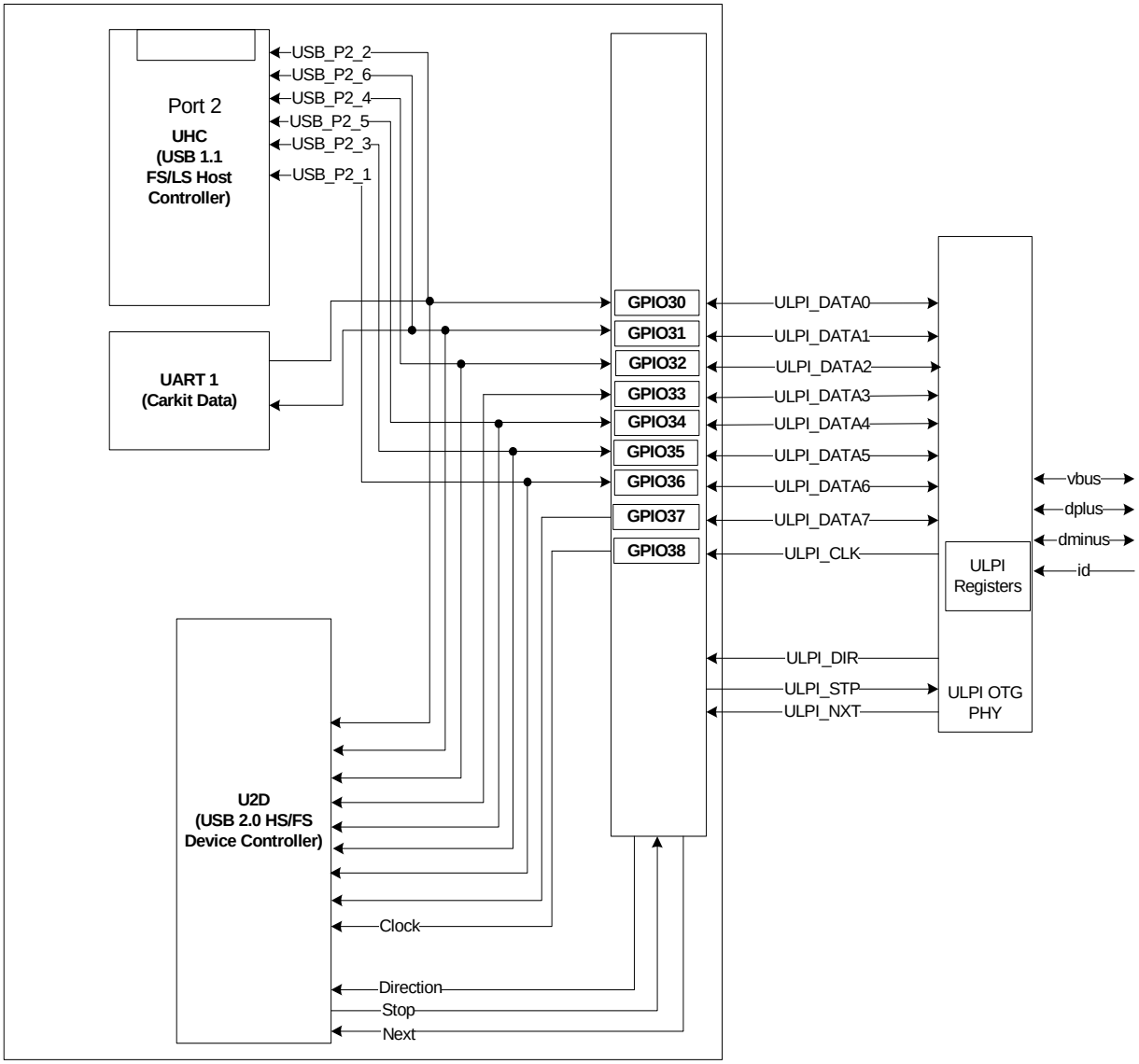


Table 61: ULPI 3/6-pin Serial Mode Signals

Signal	ULPI	GPIO	PHY DIR	Description
6-pin Serial Pinout				
TX_ENABLE	Data[0]	GPIO30	Input	Active high Tx enable

Table 61: ULPI 3/6-pin Serial Mode Signals (Continued)

Signal	ULPI	GPIO	PHY DIR	Description
TX_DAT	Data[1]	GPIO31	Input	Tx differential data on D+/D-
TX_SE0	Data[2]	GPIO32	Input	Tx single-ended zero on D+/D-
INTERRUPT	Data[3]	GPIO33	Output	Active high interrupt
RX_DP	Data[4]	GPIO34	Output	Single-ended Rx data from D+
RX_DM	Data[5]	GPIO35	Output	Single-ended Rx data from D-
RX_RCV	Data[6]	GPIO36	Output	Differential Rx data
RESERVED	Data[7]	GPIO37	Output	Reserved; PHY required to drive this pin low during this mode
3-pin Serial Pinout				
TX_ENABLE	Data[0]	GPIO30	Input	Active high Tx enable
DAT	Data[1]	GPIO31	I/O	Tx differential data on D+/D- when TX_ENABLE is high Rx differential data on D+/D- when TX_ENABLE is low
SE0	Data[2]	GPIO32	I/O	Tx SE0 on D+/D- when TX_ENABLE is high Rx SE0 on D+/D- when TX_ENABLE is low
INTERRUPT	Data[3]	GPIO33	Output	Active high interrupt

19.2 ULPI Low-Power Mode

The U2D supports a low-power mode, or suspend mode, whereby the controller places the PHY into a power savings mode. The PHY will draw minimal suspend current during this time. Below are the signal mappings during low-power mode.

Table 62: Low-Power Mode Signals

Signal	ULPI	GPIO	PHY DIR	Description
LINESTATE<0>	Data[0]	GPIO30	Output	Combinatorial LINESTATE<0> driven directly by FS analog receiver
LINESTATE<1>	Data[1]	GPIO31	Output	Combinatorial LINESTATE<1> driven directly by FS analog receiver
RESERVED	Data[2]	GPIO32	Output	Reserved; PHY must drive this pin low
INTERRUPT	Data[3]	GPIO33	Output	Active high interrupt

19.3 ULPI Carkit Mode

The PXA31x processor supports ULPI carkit mode, which is used to implement an automotive speakerphone, either through a self-contained carkit speakerphone adapter or through a carkit-enabled car stereo. The processor supports the carkit mode of operation using either a stand-alone external carkit IC, or using a carkit-enabled ULPI PHY.

Carkit mode support requires the ability to multiplex a UART peripheral interface and an audio CODEC over the USB D+ and D- signal pins. The processor UART peripheral interface can be used for this function by placing it in Transparent mode. The UART transmit (TxD) and receive (RxD) data pins are multiplexed over the USB D- and D+ pins during carkit UART signaling mode, respectively. The audio codec is an external component to the processor, and its SPKR_L and SPKR_R/MIC analog signals are multiplexed over the USB D- and D+ pins during the carkit audio modes. (Both mono and stereo audio modes are supported by the carkit standard). Refer to the CEA-936A standard for more information on this option.

UART requirements are:

- UART data rate of 9600 kbps
- 8-bit
- No parity
- One stop bit
- Assertion high polarity

Table 63: Carkit Signal Mapping

Signal	ULPI	GPIO	PHY DIR	Description
TxD	Data[0]	GPIO30	Input	UART TxD routed to D- pin
RxD	Data[1]	GPIO31	Output	UART RxD routed to D+ pin
RESERVED	Data[2]	GPIO32	Output	Reserved
INTERRUPT	Data[3]	GPIO33	Output	Active high interrupt

19.4 Layout Notes

Section 19.4.1, “General Routing Recommendations” describes general routing recommendations for the PXA31x processor ULPI interface. Refer to the PHY data sheet and layout notes for additional guidelines. Some of the PHY devices may have strict requirements which must be followed.

19.4.1 General Routing Recommendations

The following recommendations are intended for the PXA31x ULPI interface.

- Minimum 6-layer board is recommended.
- HDI and BMS routes are recommended.
- Minimum 1:1 edge-to-edge spacing is recommended for all traces.

Figure 69: Recommended Layer Stack up

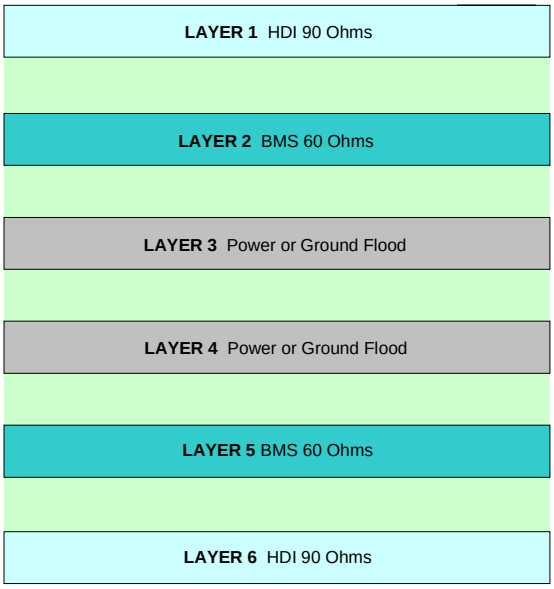


Table 64: Data<7:0>

Trace	Minimum Length	Maximum Length	Min Isolation Spacing	Z0 (Ohms)
TL1 (HDI Breakout)	0.1"	0.5"	3 mills	90±15%
TL2(BMS)	0.25"	2.5"	4 mills	60±15%
TL3 (HDI Breakout)	0.1"	0.5"	3 mills	90±15%
NOTE: Max Total Length: TL1 + TL2 + TL3 <= 3.0"				

Table 65: ULPI_DIR/ULPI_STP/ULPI_NXT

Trace	Minimum Length	Maximum Length	Min Isolation Spacing	Z0 (Ohms)
TL1 (HDI Breakout)	0.1"	0.5"	3 mills	90±15%
TL2(BMS)	0.25"	2.5"	4 mills	60±15%

Table 65: ULPI_DIR/ULPI_STP/ULPI_NXT (Continued)

Trace	Minimum Length	Maximum Length	Min Isolation Spacing	Z0 (Ohms)
TL3 (HDI Breakout)	0.1"	0.5"	3 mills	90±15%
NOTE: Max Total Length: TL1 + TL2 + TL3 <= 3.0"				

Table 66: ULPI_CLK

Trace	Minimum Length	Maximum Length	Min Isolation Spacing	Z0 (Ohms)
TL1 (HDI Breakout)	0.1"	0.5"	3 mills	90±15%
TL2(BMS)	0.25"	2.5"	4 mills	60±15%
TL3 (HDI Breakout)	0.1"	0.5"	3 mills	90±15%
NOTE: Max Total Length: TL1 + TL2 + TL3 <= 3.0"				

20 USB 1.1 Host Interface

This section describes guidelines for:

- Interfacing the universal serial bus (USB) host controller of the PXA3xx processor family to a USB host cable connector
- Attaching USB client devices to the USB host controller of the PXA3xx processor family.

20.1 Overview

The USB cable supports serial-data exchange between a host device and a variety of simultaneously accessible clients. The attached clients share USB bandwidth through a host-scheduled, token-based protocol. Clients are attached, configured, used, and detached, while the host and other clients continue operation. Familiarity with the *Universal Serial Bus Specification*, *Revision 1.1*[1] and the OHCI specification[2] are necessary to fully understand the material contained in this section.

20.2 Signals

See Table 67 for description of the USB host controller signals.

Table 67: USB Host Controller Interface Signals Summary

Signal Name	Ball Name	Type	Description
USB Full Speed Transceiver Signals - Host Port 1			
USBH1_P	USBH1_P	Analog	**USB Full Speed Host Port 1 Positive Line** — this is the positive pin of the differential pair that connects to the USB full speed host for Port 1.
USBH1_N	USBH1_N	Analog	**USB Full Speed Host Port 1 Negative Line** — this is the negative pin of the differential pair that connects to the USB full speed host for Port 1.
USBHPEN	GPIO0_2 (PXA3xx processor family)	Output	Controls power to the USB host Ports.
USBHPWR	GPIO1_2 (PXA3xx processor family)	Input	Over-current indicator.
USB Full Speed Transceiver Signals - Port 2 Host/Client and OTG			

1. To access the latest revision of the *Universal Serial Bus Specification Revision 1.1*, use the Internet site at: http://www.usb.org/
2. Refer to *Open Host Controller Interface Specification for USB, Release 1.0a,loc cit.*

Table 67: USB Host Controller Interface Signals Summary (Continued)

Signal Name	Ball Name	Type	Description
USBOTG_P	USBOTG_P	Analog	**USB Full Speed Host/Client/OTG Port 2 Positive Line** — this is the positive pin of the differential pair that connects to the USB full speed host, or the USB full speed client, or the USB OTG interface for Port 2.
USBOTG_N	USBOTG_N	Analog	**USB Full Speed Host/Client/OTG Port 2 Negative Line** — this is the negative pin of the differential pair that connects to the USB Full Speed Host, or the USB Full Speed Client, or the USB OTG interface for Port 2.
USBHPEN2	GPIO103 (PXA30x processor)	Output	**USB Full Speed Host Port 2 Power Control** — controls power to the USB Host Port 2. This signal is muxed with USB_P2_8 but depends on the value of UP2OCR[SEOS]. Refer to The *Universal Serial Bus Client Controller* chapter of the *Marvell® PXA30x, PXA31x, and PXA32x Processor Serial Controller Configuration Developers Manual, Vol. IV Developers Manual*: *Serial Controller Configuration* for more information.
USB Full Speed Single-Ended Signals - Port 2 Host/Client and OTG			
USB_P2_1	GPIO101 (PXA30x processor) GPIO36 (PXA31x processor)	Bidirectional	**USB Full Speed Host/Client and OTG Port 2 RCV/INT/SRP** — This signal is the receive data from an external USB transceiver for Port 2. When configured for an external OTG transceiver this signal is an interrupt input. When configured for an external OTG power controller and the internal transceiver, this signal is the SRP detect input.
USB_P2_2	GPIO99 or GPIO100 (PXA30x processor) GPIO30 (PXA31x processor)	Bidirectional	**USB Full Speed Host/Client and OTG Port 2 OE/Valid**—This signal connects to the OE signal of an external USB transceiver for USB Port 2. This signal connects to the session valid output of an external OTG power controller.
USB_P2_3	GPIO104 (PXA30x processor) GPIO35 (PXA31x processor)	Bidirectional	**USB Full Speed Host/Client and OTG Port 2 RXD-/SV—**This signal is the receive negative data line from an external USB transceiver for Port 2. For an external OTG power controller, this signal is the session valid status input.
USB_P2_4	GPIO102 (PXA30x processor) GPIO32 (PXA31x processor)	Bidirectional	**USB Full Speed Host/Client and OTG Port 2 TXD-/SE0_VM/SRP**—This signal is the transmit negative data line to an external USB transceiver for Port 2. For an external OTG transceiver, this signal is the SE0_VM bidirectional data line. For an external OTG power controller this signal is the SRP enable control output.
USB_P2_5	GPIO105 (PXA30x processor) GPIO34 (PXA31x processor)	Bidirectional	**USB Full Speed Host/Client and OTG Port 2 RXD+/DAT_VP/VALID40**—This signal is the receive positive data line from an external USB transceiver for Port 2. For an external OTG transceiver, this signal is the DAT_VP bidirectional data line. For an external OTG power controller this signal is the 4.0 V Vbus valid status input.

Table 67: USB Host Controller Interface Signals Summary (Continued)

Signal Name	Ball Name	Type	Description
USB_P2_6	GPIO99 or GPIO100 (PXA30x processor) GPIO31 (PXA31x processor)	Bidirectional	**USB Full Speed Host/Client and OTG Port 2 TXD+ —**This signal is the positive transmit data line for an external USB transceiver for Port 2P
USB_P2_7	GPIO106 (PXA30x processor)	Bidirectional	**USB Full Speed Host/Client and OTG Port 2 Speed/OTGID**—Speed select signal for USB Port 2 when configured for an external USB transceiver. Provides the OTG ID configuration using the internal transceiver.
USB_P2_8	GPIO103 (PXA30x processor)	Input	**USB Full Speed Host/Client and OTG Port 2 Suspend**—Suspend enable for USB Port 2 when configured for an external USB OTG transceiver.
USB Full Speed Single-Ended Signals - Port 3 Host			
USB_P3_1	GPIO77 (PXA3xx processor family)	Bidirectional	**USB Full Speed Host Port 3 RCV—**receive data signal that connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_2	GPIO78 (PXA3xx processor family)	Bidirectional	**USB Full Speed Host Port 3 OE**—output enable signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_3	GPIO79 (PXA3xx processor family)	Bidirectional	USB Full Speed Host Port 3 RXD**-** —receive data (-) signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_4	GPIO80 (PXA3xx processor family)	Bidirectional	**USB Full Speed Host Port 3 TXD-** —transmit data (-) signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_5	GPIO81 (PXA3xx processor family)	Bidirectional	**USB Full Speed Host Port 3 RXD+** —receive data (+) signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).
USB_P3_6	GPIO82 (PXA3xx processor family)	Bidirectional	**USB Full Speed Host Port 3 TXD+** —transmit data (+) signal which connects to an external transceiver or the transceiver interface of a USB device controller as defined by the CFG bits in the USB Port 3 Output Control register (UP3OCR).

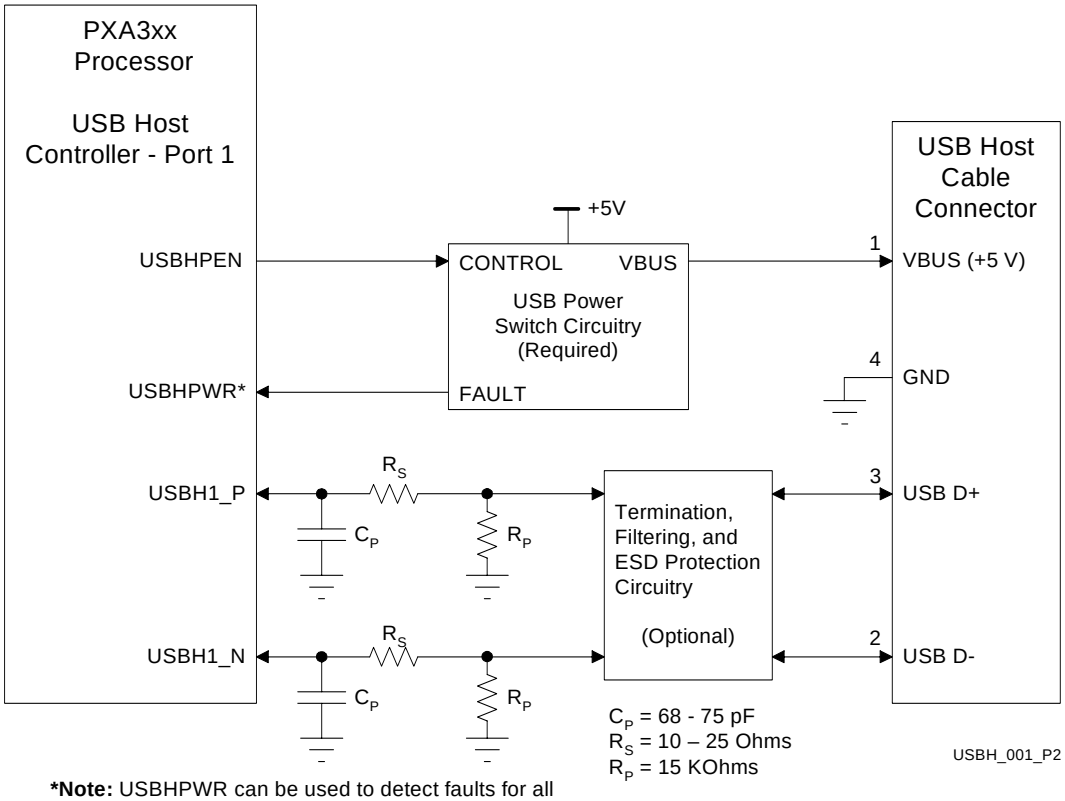
20.3 Block Diagrams

Host controller Port 1 is used only with a differential connections. Host controller Port 2 is used with either a differential connection or a single-ended connection. Host controller Port 3 is used only with a single-ended connection.

20.3.1 USB Host Differential Connection (Port 1)

See Figure 70 for USB host Port 1 differential connections of the PXA3xx processor family.

Figure 70: USB Host (Port 1) Differential Connections Block Diagram



***Note:** USBHPWR can be used to detect faults for all three host ports (Port 1, 2, 3)

20.3.2 USB Host Connections (Port 2)

Host controller Port 2 is used with either single-ended connections or a differential connections, but not in both at the same time.

See Figure 71 for a USB host Port 2 single-ended connections. See Figure 72 for USB host Port 2 differential connections of the PXA3xx processor family.

Figure 71: PXA3xx Processor Family Host 2 Single-Ended Connections to External Transceiver

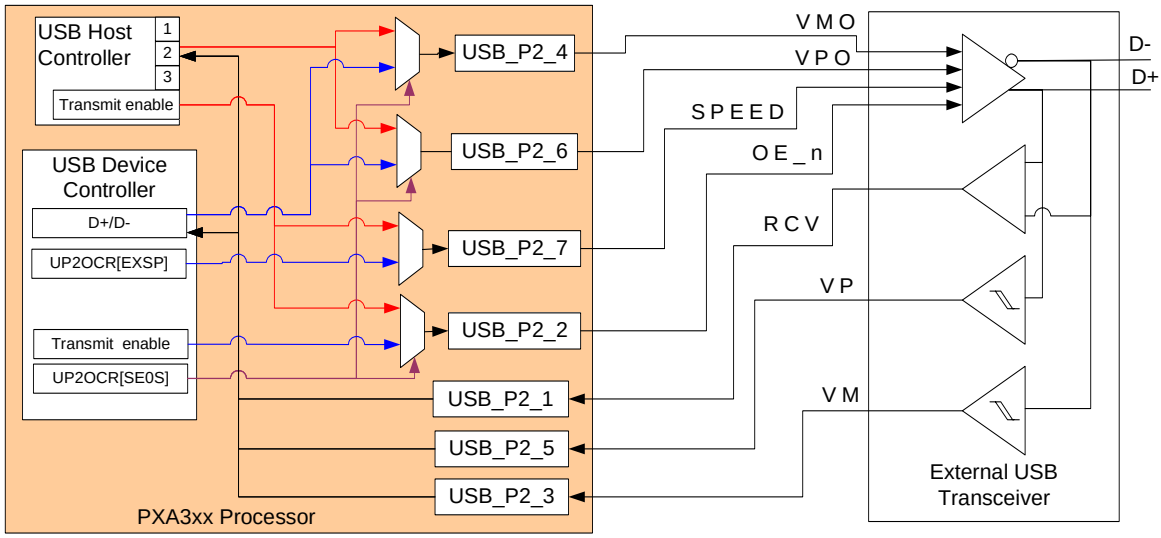
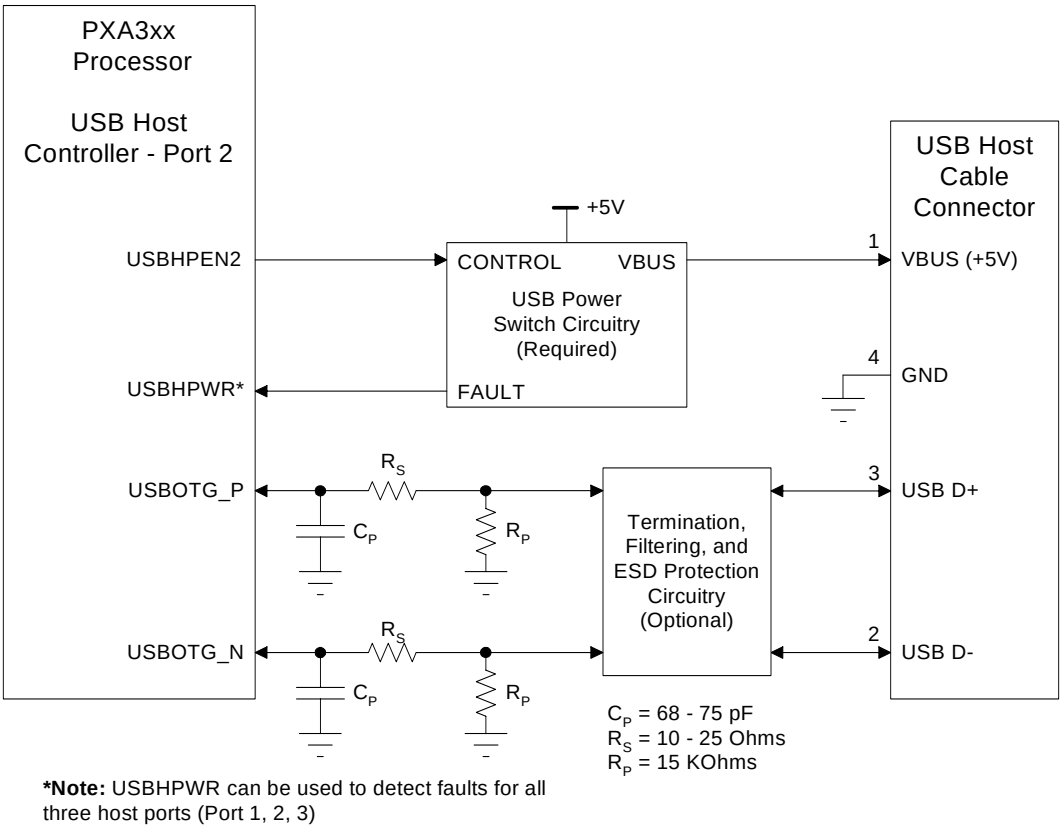


Figure 72: USB Host (Port 2) Differential Connections Block Diagram

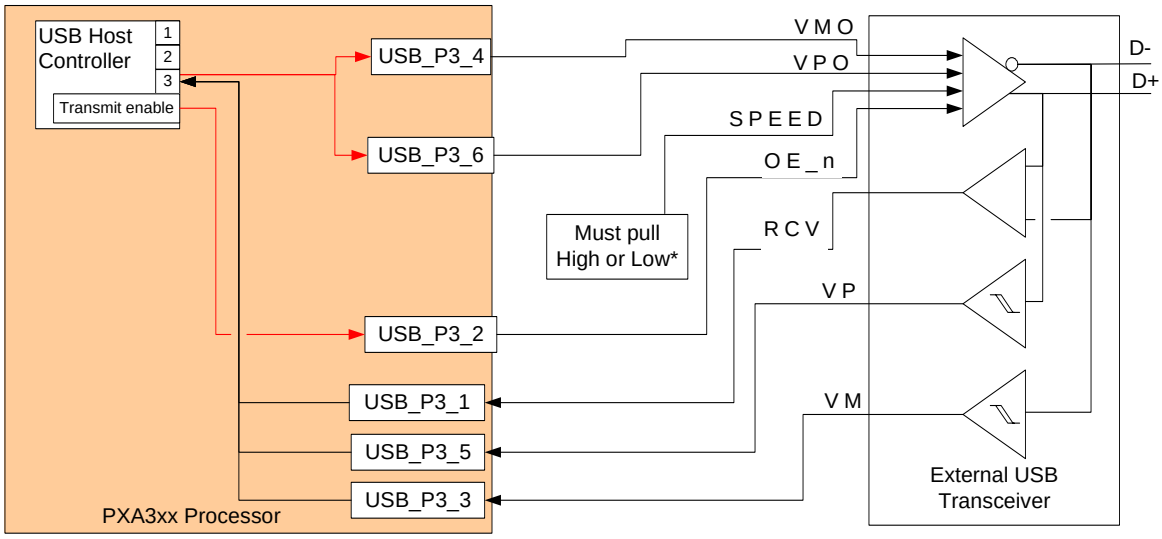


***Note:** USBHPWR can be used to detect faults for all three host ports (Port 1, 2, 3)

20.3.3 USB Host Single-Ended Connections (Port 3)

Host controller Port 3 is used only with single-ended connections as shown in Figure 73.

Figure 73: PXA3xx Processor Family Host 3 Connections to External USB Transceiver



Note The "SPEED" signal does not exist for PXA3xx processor family host Port 3. The "SPEED" signal of the external device must be tied high or low. Program PXA3xx processor family host 3 to match the external device speed.

20.4 Layout Notes

The USB power supply must be +5.0 V (per the USB specification). However, the PXA3xx processor family does not have 5.0 V-tolerant inputs. Provide an external device to interface the USBHPENx and USBHPWR pins to the power supply and over current detection circuits.

Note There is no USBHPWR2 input signal/function on the PXA3xx processor family. Use an external circuit to OR the overcurrent for port 2 back into USBPWR1

The C_P and R_S components are required by the USB host controller for USBH1_P, USBOTG_P, USBH1_N, and USBOTG_N signal compliance with the USB specification and must be placed as close as possible to the Monahans USBH1_P, USBOTG_P, USBH1_N, and USBOTG_N balls.

The R_P pulldown resistors shown on the USBH1_P, USBOTG_P, USBH1_N, and USBOTG_N signals are required per USB specification.

Terminations and filtering for signal integrity and electrostatic discharge (ESD) protection circuitry are recommended, but varies for different systems depending upon layout and end application environment.

USBH1_P and USBH1_N along with USBOTG_P and USBOTG_N are differential pair signals, respectively. Use these recommended layout guidelines:

- Route the signals close to each other as parallel traces on the PCB.
- Route the differential pair with trace width and spacing needed to achieve 90 ohms differential impedance. Deviations will normally occur due to package breakout and routing to connector pins. Keep the amount and length of width/spacing deviations to a minimum.
- Do not route USB traces under crystals, oscillators, magnetic devices or IC's that use and/or duplicate clocks.
- Avoid stubs on USB signals as the stubs will cause reflections and affect signal quality.
- Route USB signals over continuous planes (VCC or GND) with no interruptions. Avoid crossing plane splits if possible.
- Minimize the length of parallel running signals, especially clocks to avoid crosstalk. Minimum suggested spacing to clocks is 50 mils. Minimum suggested spacing to other signals is 20 mils.
- Match the trace lengths as closely as possible (within ±0.5 inches (12.7 mm)).

Power supply considerations must be made when using GPIO alternate functions. Not all USB signals are powered from the same power source. See Table 68 for the power source required depending on signals being used.

Table 68: USB Power Supplies

Power Supply	Signal Group*
VCC_USB (Analog signals, PXA300 processor only) VCC_ULPI (Analog signals, PXA31x processor only) VCC_IO1 (Digital signals)	USB Full Speed Transceiver Differential Signals - Host Port 1
VCC_BIAS (PXA300 processor only) VCC_ULPI (PXA31x processor only)	USB Full Speed Transceiver Differential Signals - Port 2 Host/Client and OTG
VCC_IO2	USB Full Speed Single-Ended Signals - Port 2 Host/Client and OTG
VCC_MSL	USB Full Speed Single-Ended Signals - Port 3 Host
NOTE: *See Table 67 for signals within each groups.	

21 SSP Port Interface

21.1 Overview

The PXA3xx processor family contains four programmable synchronous serial ports (SSP) that connect to various devices, but not limited to:

- External analog-to-digital (A/D) converters
- Audio and telecommunication CODECs
- Devices that use serial protocols for data transfer

All four SSPs are identical except that SSP3 and SSP4 have no external clock and no external clock enable. SSP1 and SSP2 are configurable in any of the modes described in this section, while SSP3 and SSP4 are configurable for any mode except external clocking.

Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. IV: Serial Controller Configuration Developers Manual*, “Section 5: SSP Serial Ports” for information on programming and configuring the SSPs. This section describes only the physical connections for the SSPs.

The differences between the three possible operational modes (National* Semiconductor Microwire Protocol, Motorola* SPI Protocol, and Programmable Serial Protocol) are software differences and do not affect the design and implementation of the SSPs in an embedded system. The TI* Synchronous Serial Protocol can be emulated by appropriately programming the SSP in PSP mode. The different operational modes are not discussed further in this document. Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. IV: Serial Controller Configuration Developers Manual*, “Section 5: SSP Serial Ports” for detailed information on operation.

21.2 Signals

See Table 69 for a description of the signals associated with each of the four SSPs. Each of the signals for the four SSPs is described separately due to slight differences in some of the signals. These signals are available on the GPIO pins. The GPIO must be enabled correctly to have access to the functionality of the SSPs described in Table 69. Refer to the GPIO alternate function table in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 5: General-Purpose I/O Unit” for GPIO assignments of the SSP signals.

Table 69: SSP Serial Port I/O Signals

Name	Direction	Description
SSPSCLK	Inout	Synchronous Serial Protocol Serial Clock – SSPSCLK is the serial bit clock used to control the timing of a transfer. SSPSCLK is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_1[SCLKDIR].
SSPSYSCLK	Output	Synchronous Serial Protocol System Clock – SSPSYSCLK is one, four, or eight times the SSPSCLK1 value, depending on SSACD_1[SCDX8] and SSACD_1[SCDB].

Table 69: SSP Serial Port I/O Signals (Continued)

Name	Direction	Description
SSPSFRM	Inout	Synchronous Serial Protocol Serial Frame Indicator – SSPSFRM, the serial frame indicator, determines the beginning and the end of a serialized data word. SSPSFRM is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_1[SFRMDIR].
SSPTXD	Output	Synchronous Serial Protocol Transmit Data – SSPTXD is the transmit data (serial data out) serialized data line.
SSPRXD	Input	Synchronous Serial Protocol Receive Data – SSPRXD is the receive data (serial data in) serialized data line.
SSPEXTCLK	Input	Synchronous Serial Protocol External Clock – SSPEXTCLK is an external clock that is selected to replace the internal 13 MHz clock. SSPEXTCLK is multiplexed with the SSPSCLKEN alternate function.
SSPSCLKEN	Input	Synchronous Serial Protocol Serial Clock Enable – SSPSCLKEN is an asynchronous external enable for SSPSCLK. SSPSCLKEN is multiplexed with the SSPEXTCLK alternate function.
SSPSCLK2	Inout	Synchronous Serial Protocol Serial Clock 2 – SSPSCLK2 is the serial bit clock used to control the timing of a transfer. SSPSCLK2 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_2[SCLKDIR].
SSPSYSCLK2	Output	Synchronous Serial Protocol System Clock 2 – SSPSYSCLK2 is one, four, or eight times the SSPSCLK2 value, depending on SSACD_2[SCDX8] and SSACD_2[SCDB].
SSPSFRM2	Inout	Synchronous Serial Protocol Serial Frame Indicator 2 – SSPSFRM2, the serial frame indicator, determines the beginning and the end of a serialized data word. SSPSFRM2 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_2[SFRMDIR].
SSPTXD2	Output	Synchronous Serial Protocol Transmit Data 2 – SSPTXD2 is the transmit data (serial data out) serialized data line.
SSPRXD2	Input	Synchronous Serial Protocol Receive Data 2 – SSPRXD2 is the receive data (serial data in) serialized data line.
SSPEXTCLK2	Input	Synchronous Serial Protocol External Clock 2 – SSPEXTCLK2 is an external clock that is selected to replace the internal 13 MHz clock. SSPEXTCLK2 is multiplexed with the SSPSCLK2EN alternate function.
SSPSCLK2EN	Input	Synchronous Serial Protocol Serial Clock Enable 2 – SSPSCLK2EN is an asynchronous external enable for SSPSCLK2. SSPSCLK2EN is multiplexed with the SSPEXTCLK2 alternate function.
SSPSCLK3	Inout	Synchronous Serial Protocol Serial Clock 3 – SSPSCLK3 is the serial bit clock used to control the timing of a transfer. SSPSCLK3 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_3[SCLKDIR].

Table 69: SSP Serial Port I/O Signals (Continued)

Name	Direction	Description
SSPSYSCLK3	Output	Synchronous Serial Protocol System Clock 3 – SSPSYSCLK3 is one, four, or eight times the SSPSCLK3 value, depending on SSACD_3[SCDX8] and SSACD_3[SCDB].
SSPSFRM3	Inout	Synchronous Serial Protocol Serial Frame Indicator 3 – SSPSFRM3, the serial frame indicator, determines the beginning and the end of a serialized data word. SSPSFRM3 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_3[SFRMDIR].
SSPTXD3	Output	Synchronous Serial Protocol Transmit Data 3 – SSPTXD3 is the transmit data (serial data out) serialized data line.
SSPRXD3	Input	Synchronous Serial Protocol Receive Data 3 – SSPRXD3 is the receive data (serial data in) serialized data line.
SSPSCLK4	Inout	Synchronous Serial Protocol Serial Clock 4 – SSPSCLK4 is the serial bit clock used to control the timing of a transfer. SSPSCLK4 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_4[SCLKDIR].
SSPSYSCLK4	Output	Synchronous Serial Protocol System Clock 4 – SSPSYSCLK4 is one, four, or eight times the SSPSCLK4 value, depending on SSACD_4[SCDX8] and SSACD_4[SCDB].
SSPSFRM4	Inout	Synchronous Serial Protocol Serial Frame Indicator 4 – SSPSFRM4, the serial frame indicator, determines the beginning and the end of a serialized data word. SSPSFRM4 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_4[SFRMDIR].
SSPTXD4	Output	Synchronous Serial Protocol Transmit Data 4 – SSPTXD4 is the transmit data (serial data out) serialized data line.
SSPRXD4	Input	Synchronous Serial Protocol Receive Data 4 – SSPRXD4 is the receive data (serial data in) serialized data line.
CLK_EXT	Input	External Network Clock – CLK_EXT replaces the internal 13 MHz clock. Use CLK_EXT when SSCR0_x[NCS] is set and SSCR1_x[SCLKDIR] is cleared. CLK_EXT is used by multiple SSPs.

21.3 Block Diagrams

This section discusses alternate SSP configuration schemas (topologies) and includes a block diagram for each schema.

21.3.1 Standard SSP Configuration Schema

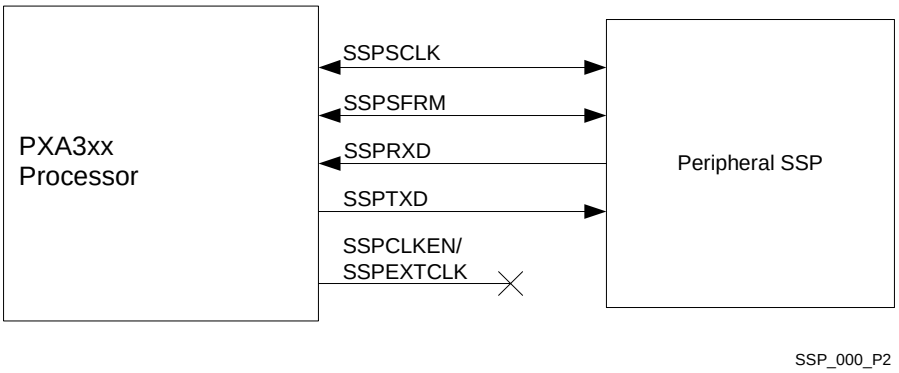
The standard SSP configuration schema is the most common method of configuration and allows one SSP to interface directly to another SSP on another device.

This configuration allows for a number of possible configurations. The SSPSCLK can be programmed to be either a master or a slave to the peripheral. Likewise, the SSPSFRM can be programmed to be a master or a slave to the peripheral, which allows for one physical connection to result in one of four configurations with only software changes:

- Master to SSPSCLK / master to SSPSFRM
- Master to SSPSCLK / slave to SSPSFRM
- Slave to SSPSCLK / master to SSPSFRM
- Slave to SSPSCLK and slave to SSPSFRM

See Figure 74 for illustration of the physical connections of the above configurations.

Figure 74: Standard SSP Configuration Schema Block Diagram



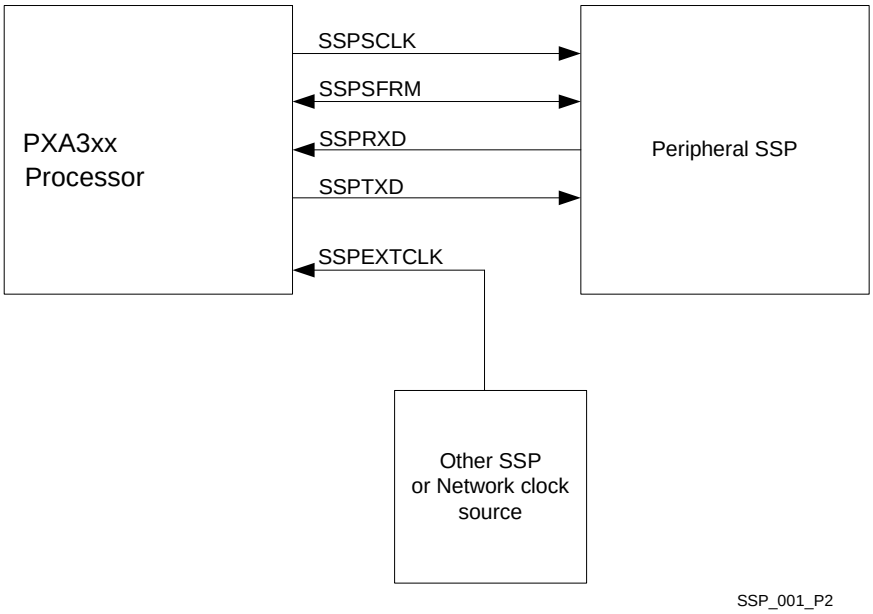
21.3.2 External Clock Source Configuration Schema

The external clock source configuration allows for an external clock source to be the SSPCLK generation source. Using an external clock source is different than the SSPSCLK operating as a slave. When using an external clock source, the SSPSCLK is still a master; however, the external clock replaces the internal 13 MHz clock for PXA30x processor or 26 MHz internal clock for PXA31x processor (with SSCR0_x[52MM] set to one) as the source clock for SSPSCLK generation.

The external clock source configuration allows SSPSCLK frequencies with bases other than 13 MHz, or for using a network clock to serve as a clock source for an SSP, but the SSP is still the master of SSPSCLK.

See Figure 75 for illustration of the physical connection of the external clock source configuration.

Figure 75: External Clock Source Configuration Schema Block Diagram

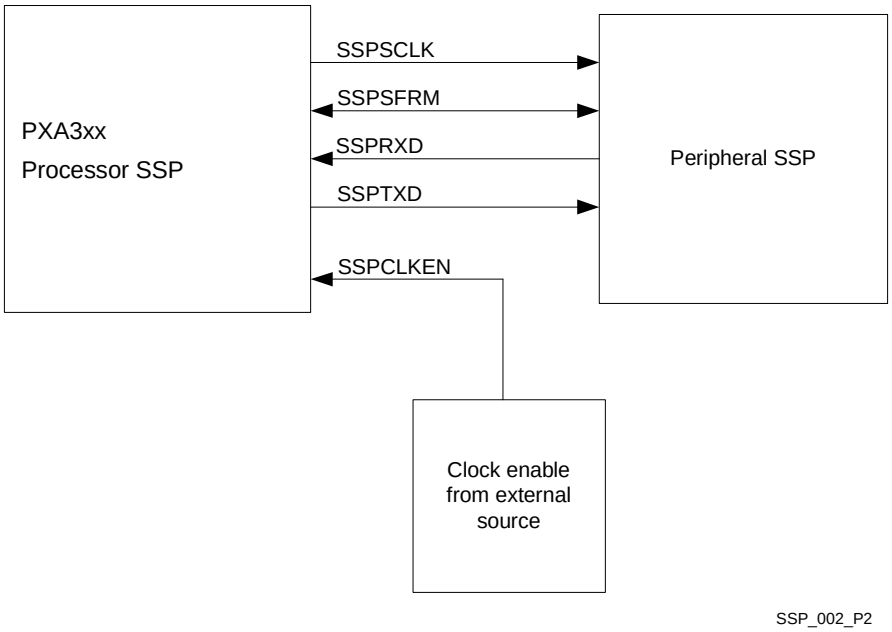


21.3.3 External Clock-Enable Configuration Schema

The external clock-enable configuration allows an external device to control when the SSP is enabled. Configuring the SSP for external clock enable is performed by the use of the SSPCLKEN signal. In this mode, the SSP must be the master of SSPSCLK and does not use an external or network clock as the base clock.

See Figure 76 for illustration of the physical connection of the external clock-enable configuration.

Figure 76: External Clock Enable Configuration Schema Block Diagram

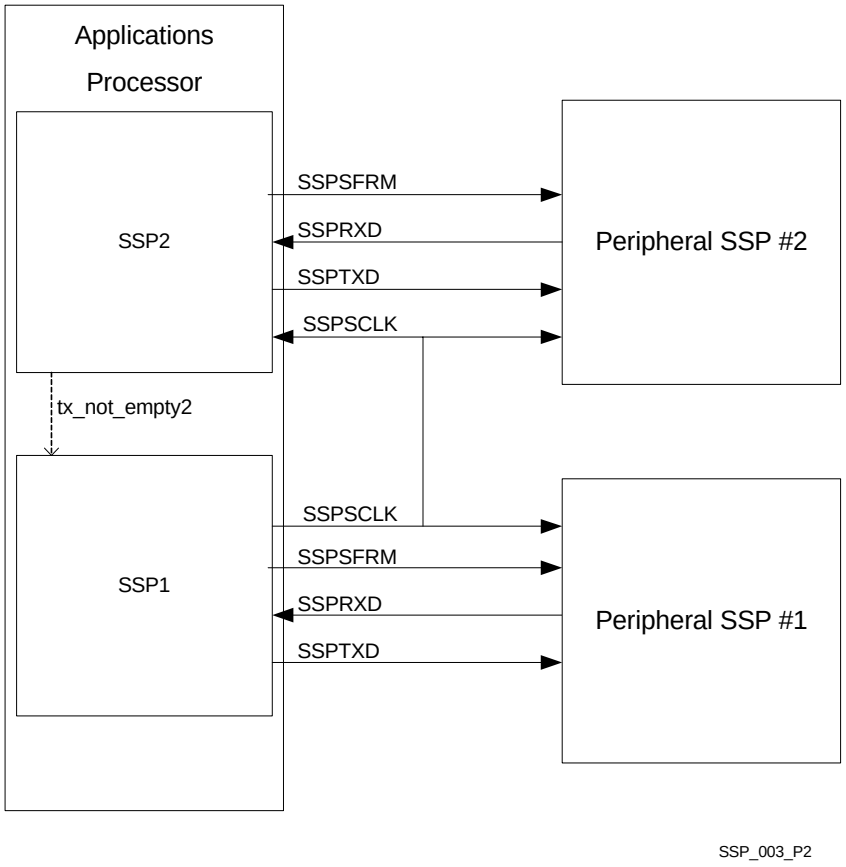


21.3.4 Internal (to PXA3xx Processor Family) Clock Enable Configuration Design

The physical connections for the internal clock-enable configuration design using internal clock enable are identical to those for the standard SSP configuration design; although certain configurations cause a different SSP within the processor to generate SSPSCLK regardless of whether the master of the SSPSCLK has data to send. When the SSP is internally enabled, the SSP must be the master of SSPSCLK in this mode.

See Figure 77 for an example of the physical connections in the internal clock-enable configuration design. In this example, SSP1 is the master of SSPSCLK and SSP2 is internally forcing SSP1 to generate SSPSCLK using the internal signal tx_not_empty2. Doing so allows SSP2 of the PXA3xx processor family to send data to Peripheral SSP #2. While Peripheral SSP #1 would also receive the SSPSCLK, it would not receive a frame signal (SSPSFRM) from SSP1 of the PXA3xx processor family (since SSP1 does not have any data to send to Peripheral SSP #1) and therefore, would not receive any data. In this example, the only function of SSP1 of the PXA3xx processor family is to generate the SSPSCLK signal.

Figure 77: Internal Clock Enable Configuration Scheme Block Diagram



21.4 Layout Notes

The tested maximum switching speed of the SSPs are 13 MHz for PXA30x processor or 26 MHz for PXA31x processor (with SSCR0_x[52MM] set to one). Board layout and design constraints at this speed are not required to prevent signal degradation. However, SSP configurations normally have one clock source switching at the master clock frequency (a second source is also possible), plus the two data lines switching at half that frequency, increasing the possibility of crosstalk between signals.

Follow these recommendations when routing signals to ensure the most reliable design possible:

- Keep all signal traces as short as possible.
- Separate the signals as far as possible when routing.
- Minimize running clock and data signals in parallel to each other.
- Route clocks with as much of the trace on an inner signal layer as possible.

SSPSFRM and SSPCLKEN have slower switching frequencies and are routed with fewer restrictions. However, for optimal system operation, follow all recommendations for the clock and data signals of the SSP.

22 AC ’97 Interface

This section describes guidelines for interfacing the PXA3xx processor family AC ’97 controller to an external CODEC device.

22.1 Overview

The AC ’97 controller interface connects external audio integrated circuit devices and CODECs to the PXA3xx processor family. The AC ’97 controller supports the AC ’97 (Audio CODEC ‘97) specification revision 2.3 features and AC-link, a synchronous, fixed-rate serial bus interface to the digital AC ’97 controller for transferring digital audio, modem, microphone input (MIC-in), controlling the CODEC register, and providing status information.

The AC ’97 port is a bidirectional, serial, pulse-code-modulated (PCM), digital-stream interface. For recording activities, the AC ’97 CODEC sends the digitized audio samples to the AC ’97 controller, which stores them in memory. For playback or synthesized audio production, the PXA3xx processor family retrieves stored audio samples and sends them to the CODEC through the AC link. The external digital-to-analog converter (DAC) inside the CODEC then converts the audio samples to an analog audio waveform.

22.2 Signals

AC ’97 interface signals form the AC link, which is a point-to-point synchronous serial interconnect that supports full-duplex data transfers. All digital audio streams, modem line CODEC streams, and command/status information are communicated over the AC link. The AC link uses multi-function pins (MFPs). Software must reconfigure the MFPs to use them as the AC link. See Table 70 for the list and description of AC link pins.

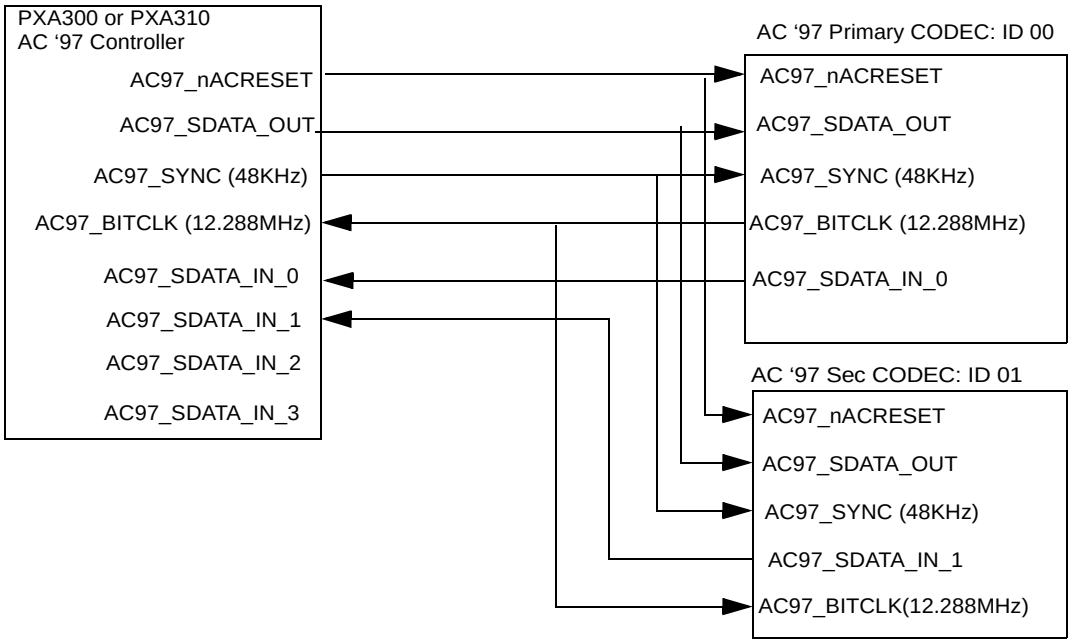
Table 70: External Interface to CODECs

Signal Name	Direction	Description Summary
AC97_nACRESET	Output	Active-low CODEC reset The CODEC registers are reset when AC97_nACRESET is asserted
AC97_BITCLK	Input	12.288 MHz bit-rate clock
AC97_SYNC	Output	48 kHz frame indicator and synchronizer
AC97_SDATA_OUT	Output	Serial audio output data to CODEC for digital-to-analog conversion
AC97_SDATA_IN_0	Input	Serial audio input data from primary CODEC
AC97_SDATA_IN_1	Input	Serial audio input data from secondary CODEC 01
AC97_SDATA_IN_2	Input	Serial audio input data from secondary CODEC 10
AC97_SDATA_IN_3	Input	Serial audio input data from secondary CODEC 11
AC97_SYSCLK	Output	Optional 24.5 MHz clock output

22.3 Block Diagram

See Figure 78 for a block diagram showing AC '97 connections for three CODECs. It is possible to connect up to four AC '97 CODECs (not shown). The primary CODEC supplies the 12.288 MHz clock to the AC '97 interface. This clock is then driven into the AC '97 controller on the PXA3xx processor family and the AC '97 secondary CODECs.

Figure 78: AC '97 Controller to CODEC Block Diagram



22.4 AC97_nACRESET Signal

The AC Link must be reset whenever the PXA3xx processor family enters a low power state since the AC '97 controller is unable to operate while in these states. The AC97_nACRESET signal must be held in the appropriate state while the PXA3xx processor family is in the low-power state. For power consumption considerations, holding the CODEC devices in reset may not be the device's minimal power state. PXA3xx processor family low-power states are S0/D1/C2 (standby with LCD refresh), S0/D2/C2 (standby) and S2/D3/C4 (sleep). During these states it is possible to configure PXA3xx processor family to hold AC97_nACRESET in the desired state by configuring the multi-function pins (MFPs) correctly. It is not possible to drive AC97_RESET_n using the MFP capabilities of the PXA3xx processor family in S3/D4/C4 (deep-sleep). Instead, a pull-up or pull-down resistor must be employed.

If AC97_RESET_n is held low during low power modes the CODEC devices with wake-up interrupt-generating capability, such as touch-screen controllers, cannot use this wake-up functionality because the CODEC devices are being held in reset.

To drive AC97_RESET_n during low power mode the multi-function pins (MFP) must be correctly configured before entering the low-power mode. Use the MFPR registers to configure the MFPs. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer*

Configuration Developers Manual, “Section 4: Pin Descriptions and Control” for detailed information on the MFPR registers and the proper steps for configuring the MFPs for low-power mode.

22.5 Layout Notes

Because of the analog/digital nature of the CODEC, it is important to follow proper mixed-signal layout procedures. Refer to the layout recommendations provided in the CODEC data sheet. Some general recommendations are as follows:

- A 6 layer (minimum) board is recommended.
- High Density Interconnect (HDI) and Buried Microstrip Interface (BMS) layers are recommended. HDI should be primarily used for breakout. BMS should be used for actual signal routing. Minimum length on HDI is determined by board technology limitations.
- HDI Target Impedance: 90 Ohms
- BMS Target Impedance: 60 Ohms
- A solid VSS (non-dedicated) and a power flood (dedicated for return path) is recommended for this interface.
- Minimum 1:1 edge-to-edge spacing (Trace Spacing => Trace Width) is recommended for all traces.
- 8X (GPIO register setting 101/Fast_6mA) drive strength is recommended for all signals in 1.8 V mode. 4X (GPIO register setting 011/Fast_4mA) is recommended for all signals in 3.3V mode.
- Use a separate power supply for the analog audio portion of the design.
- Do not route digital signals through the analog section of the AC97 interface. Route digital signals around the analog portion.
- For signals that go off of the board, as when a CODEC is on a daughter card, use buffers between the connector and CODEC.
- Locate the decoupling capacitors for the analog portion as close to the CODEC as possible.

The optional AC97_SYSCLK output is used for clocking the AC97 XTL_IN input instead of using an external oscillator or an external crystal. The AC97_SYSCLK output option offers system designers opportunities for reducing part count, cost, and PCB space.

Figure 79: Recommended Layer Stack Up

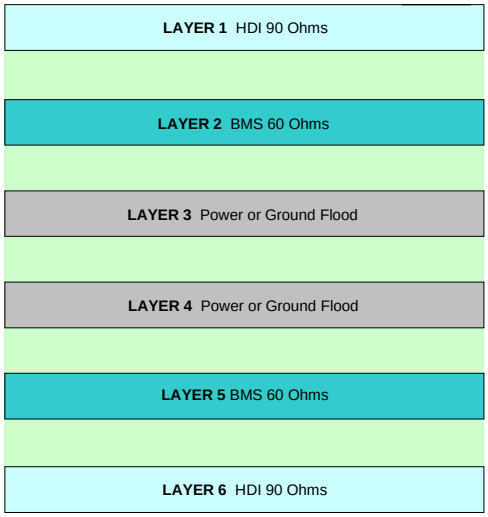


Table 71: AC '97 Signals and Possible Loading Configurations

Signals	Type	1 Load	2 Loads
AC97_SDATA_OUT	Output	Yes	Yes
AC97_SYNC	Output	Yes	Yes
AC97_nACRESET	Output	Yes	Yes
AC97_SDATA_IN[3:0]	Input	Yes	Yes
AC97_BITCLK	Input	Yes	Yes

22.6 Routing Recommendations

22.6.1 Topology # 1: SDATA_OUT / SYNC with a Single Load

Figure 80: Topology # 1 - PXA3xx Processor Family SDATA_OUT and SYNC to a Single AC '97 Codec

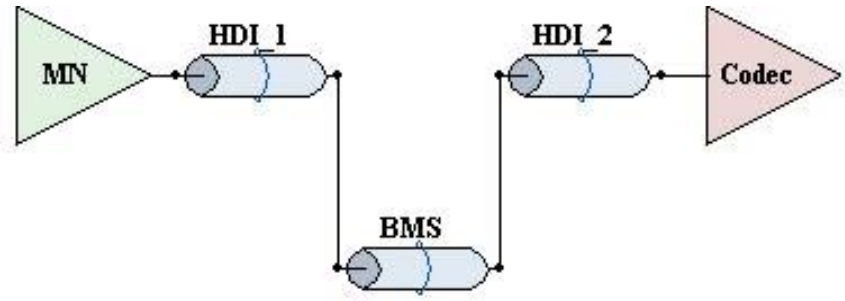


Table 72: Signal: AC97 SDATA_OUT and SYNC (Single Load)

Mode	Section	Maximum Length *	Minimum Length
1.8 V	HDI_1 + BMS + HDI_2	5 inches	0.5 inches
3.3 V	HDI_1 + BMS + HDI_2	7 inches	0.5 inches

- HDI should be used mainly for breakout. Maximum length on HDI_1 or HDI2 is 0.5 inches.
- This topology is point-to-point with no stubs.

22.6.2 Topology # 2: SDATA_OUT with Two Loads

Figure 81: Topology # 2-PXA3xx Processor Family SDATA_OUT to Two AC '97 Codecs

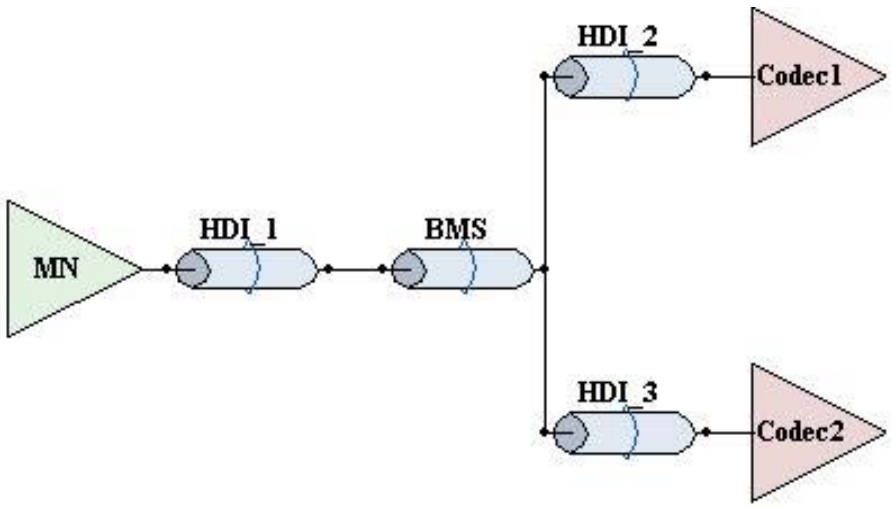


Table 73: Topology # 2-PXA3xx Processor Family SDATA_OUT to Two AC '97 Codecs

Mode	Section	Maximum Length*	Minimum Length
1.8 V	HDI_1 + BMS + HDI_2	2.5 inches	0.5 inches
1.8 V	HDI_1 + BMS + HD_I3	2.5 inches	0.5 inches
3.3 V	HDI_1 + BMS + HDI_2	7.0 inches	0.5 inches
3.3 V	HDI_1 + BMS + HDI_3	7.0 inches	0.5 inches

- HDI should be used mainly for breakout. Maximum length on HDI_1, HDI_2 or HDI_3 is 0.5 inches.

22.6.3 Topology # 3: SDATA_IN with a Single Load

Figure 82: Topology # 3 - Primary Codec to PXA3xx Processor Family - SDATA_OUT (Single Load)

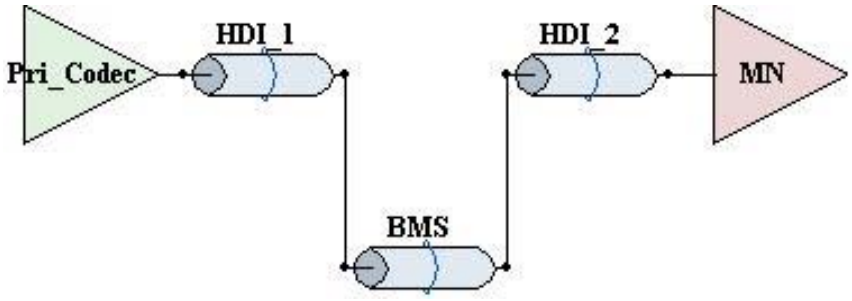


Table 74: Topology # 3: AC97_SDATA_IN[1:0] (Single Load)

Mode	Section	Maximum Length*	Minimum Length
1.8 V	HDI_1 + BMS + HDI_2	7.0 inches	0.5 inches
3.3 V	HDI_1 + BMS + HDI_2	7.0 inches	0.5 inches

- HDI should be used mainly for breakout. Maximum length on HDI_1 or HDI2 is 0.5 inches.
- This topology is point-to-point with no stubs.
- AC97_SDATA should be length matched to AC97_BITCLK to within ± 1.0 inch.

Table 75: Topology # 3: AC97_BITCLK (Single Load)

Mode	Section	Maximum Length*	Minimum Length
1.8 V	HDI_1 + BMS + HDI_2	5.0 inches	0.5 inches
3.3 V	HDI_1 + BMS + HDI_2	7.0 inches	0.5 inches

- HDI should be used mainly for breakout. Maximum length on HDI_1 or HDI2 is 0.5 inches.
- This topology is point-to-point with no stubs.

22.6.4 Topology # 4: AC97_BITCLK with Two Loads

Figure 83: Topology # 4: Primary Codec Driving BITCLK to PXA3xx Processor Family and Secondary Codec

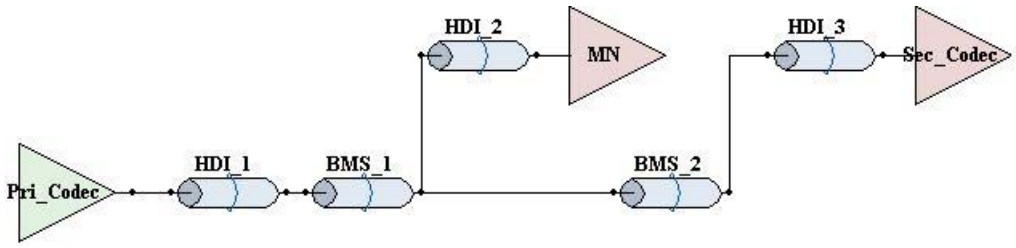


Table 76: Topology # 4: Primary Codec driving BITCLK to Two Loads

Mode	Section	Maximum Length*	Minimum Length
1.8 V	HDI_1 + BMS_1 + HDI_2	2.5 inches	0.5 inches
1.8 V	HDI_1 + BMS_1 + BMS_2 + HDI_3	2.5 inches	0.5 inches
3.3 V	HDI_1 + BMS_1 + HDI_2	7.0 inches	0.5 inches
3.3 V	HDI_1 + BMS_1 + BMS_2 + HDI_3	7.0 inches	0.5 inches

- HDI should be used mainly for breakout. Maximum length on HDI_1 or HDI_2 or HDI_3 is 0.5 inches.

23 UART Interfaces

This section describes interface guidelines for the PXA3xx processor family universal asynchronous receiver/transmitter (UART) serial ports.

23.1 Overview

The PXA3xx processor family has three UARTs that all use the same programming model:

- One full function UART (UART1)
- Two Bluetooth* UARTs (UART2 and UART3)

Each serial port contains a UART and a slow serial infrared transmit encoder and receive decoder that conform to the IrDA serial infrared specification.[1]

Each UART includes a programmable baud-rate generator that internally divides a 14.7456 Mhz input clock to produce a 16X baud rate clock that can be used to drive the internal transmit and receive logic. The supported baud rates are 9600, 19.2 K, 38.4 K, 57.6 K, 115.2 K, 230 K, 460 K, 921 K, 1.8 M, and 3.6 M.

Receive and transmit FIFO fill and drain operations can be done using programmed IO or DMA transfers. To minimize CPU overhead for UART communications, device driver software can setup interrupts and DMA for data transfers to/from memory.

All three UARTs support the 16550A and 16750[2] functions, but are slightly different in the features supported.

23.2 Signals

See Table 77 for the list and description of the external signals connected to the UART modules. The signals are connected to the PXA3xx processor family through GPIOs. The signal names used in Table 77 are not specific to the UART with which they are used, but are a combination of the non-specific signal name and the abbreviation for the associated UART. For example, the specific name of the RXD signal of the full function UART is UART1_RXD, a concatenation of “UART1_” (denoting full function) and “RXD” (the non-specific signal name). Refer to Table 78 and Table 79 for the specific signals associated with each UART.

Table 77: UART Signal Descriptions

Name	Direction	Description
RXD	Input	SERIAL INPUT – Serial data input to the Receive Shift register. In infrared mode, it is connected to the infrared RXD signal. This signal is present on all three UARTs.
TXD	Output	SERIAL OUTPUT – Serial data output to the communications link-peripheral, modem, or data set. TXD is set to the logic 1 state upon a reset operation. It is connected to the TXD signal of the infrared transmitter in infrared mode. This signal is present in all three UARTs.

1. Infrared Data Association, *Serial Infrared Physical Layer Link Specification*, October 17, 1995, Version 1.1
2. The 16550A was originally produced by National Semiconductor Inc. The 16750 is produced as the TL16C750 by Texas Instruments.

Table 77: UART Signal Descriptions (Continued)

Name	Direction	Description
nCTS	Input	CLEAR TO SEND – When low, indicates the modem or data set is ready to exchange data. nCTS has no effect on the transmitter. This signal is present on the full function UART and bluetooth UART.
nDSR	Input	DATA SET READY – When low, indicates the modem or data set is ready to establish a communications link with a UART. This signal is present only on the full function UART.
nDCD	Input	DATA CARRIER DETECT – When low, indicates the data carrier has been detected by the modem or data set. nDCD has no effect on the receiver. This signal is present only on the full function UART.
nRI	Input	RING INDICATOR – When low, indicates the modem or data set has received a telephone ringing signal. This signal is present only on the full function UART.
nDTR	Output	DATA TERMINAL READY – When low, signals the modem or the data set that the UART is ready to establish a communications link. This signal is present only on the full function UART.
nRTS	Output	REQUEST TO SEND – When low, signals the modem or the data set that the UART is ready to exchange data. This signal is used by the full function UART and bluetooth UART.

23.3 Types of UARTs

The PXA3xx processor family has three separate UARTs that use same programming model. The UARTs are described in this section.

23.3.1 Full Function UART

Full function UART (UART1) supports modem control capability and supports baud rates of 9600, 19.2 K, 38.4 K, 57.6 K, 115.2 K, 230 K, 460 K, 921 K, 1.8 M, and 3.6 M.

23.3.1.1 Full Function UART Signals

See Table 78 for the list and description of the full-function UART interface signals.

Table 78: Full Function UART Interface Signals

Signal Name	Direction	Description
UART1_RXD	Input	Full function UART Receive, serial data input to the UART (RXD)
UART1_TXD	Output	Full function UART Transmit, serial data output from the UART (TXD)
UART1_CTS	Input	Full function UART Clear to Send (nCTS)
UART1_DSR	Input	Full function UART Data Set Ready (nDSR)
UART1_DCD	Input	Full function UART Data Carrier Detect (nDCD)
UART1_RI	Input	Full function UART Ring Indicator (nRI)

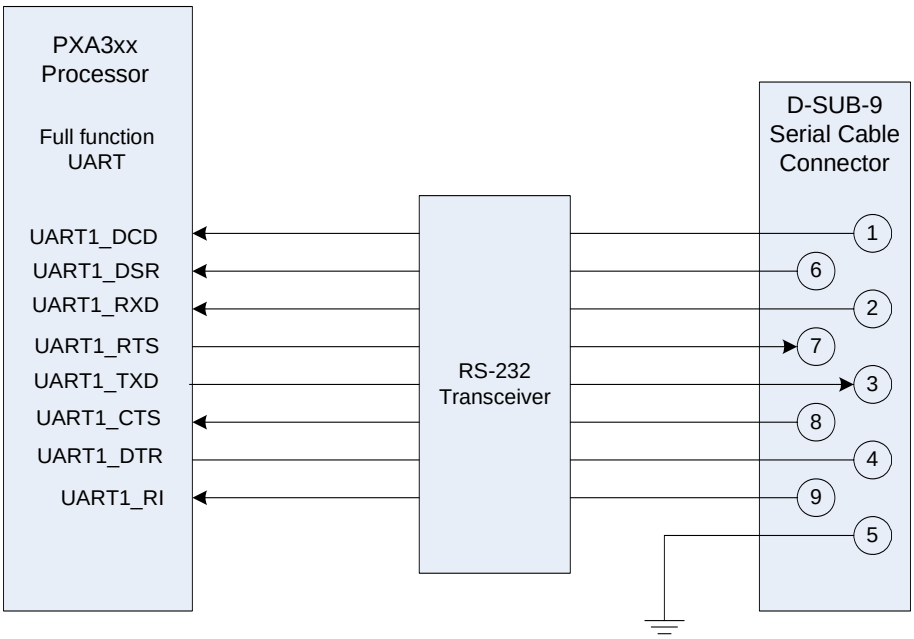
Table 78: Full Function UART Interface Signals (Continued)

Signal Name	Direction	Description
UART1_DTR	Output	Full function UART Data Terminal Ready (nDTR)
UART1_RTS	Output	Full function UART Request to Send (nRTS)

23.3.1.2 Full Function UART Block Diagram

See Figure 84 for a full function UART block diagram that shows the interface between the full function UART and a standard 9-pin D-SUB-9 connector. The block diagram demonstrates high-level signal use and connectivity when using the full function UART.

Figure 84: Full Function UART Interface Block Diagram



23.3.1.3 Full Function UART Layout Notes

The RS-232 transceiver device converts CMOS logic voltage levels to RS-232 standard line voltage levels. Locate the RS-232 transceiver close to the 9-pin D-SUB-9 connector.

23.3.2 Bluetooth UART

Bluetooth UARTs (UART2 and UART3) are high-speed UARTs and support baud rates of 9600, 19.2 K, 38.4 K, 57.6 K, 115.2 K, 230 K, 460 K, 921 K, 1.8 M, and 3.6 M. UART2 and UART3 can be connected to Bluetooth modules. The bluetooth UART only supports two modem control signals (nCTS, nRTS).

23.3.2.1 Bluetooth UART Signals

See Table 79 for the list and description of the bluetooth UART interface signals.

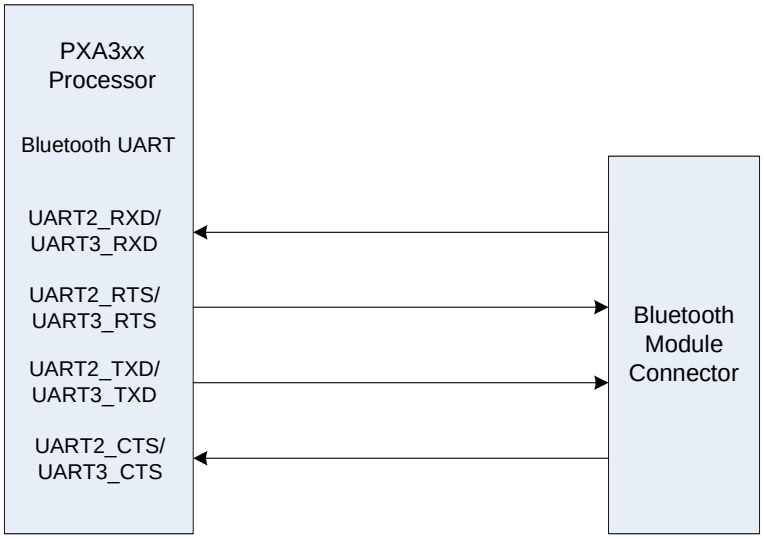
Table 79: Bluetooth UART Interface Signals

Signal Name	Direction	Description
UART2_RXD/UART3_RXD	Input	Bluetooth UART Receive, serial data input to the UART (RXD)
UART2_TXD/UART3_TXD	Output	Bluetooth UART Transmit, serial data output from the UART (TXD)
UART2_CTS/UART3_CTS	Input	Bluetooth UART Clear to Send (nCTS)
UART2_RTS/UART3_RTS	Output	Bluetooth UART Request to Send (nRTS)

23.3.2.2 Bluetooth UART Block Diagram

See Figure 85 for a block diagram that shows the interface between the bluetooth UART and a Bluetooth module connector. The block diagram demonstrates high-level signal usage and connectivity when using the Bluetooth UART.

Figure 85: Bluetooth UART Interface Block Diagram



24 Consumer Infrared Interface

This section describes guidelines for interfacing the external LED transceivers to the PXA3xx processor family consumer infrared (CIR) controller.

24.1 Overview

The consumer infrared unit (CIR) enables the PXA3xx processor family to remotely control consumer devices such as televisions and VCRs. Since there are several existing CIR standards in the market, such as RC-5, RC-5 extended, RC-6, and others, the module design is generic and flexible, so that the design is compatible with existing standards.

24.2 Signals

See Table 80 for a description of the function of the CIR signal. The CIR unit is capable of connecting directly to an infrared diode, although the current draw of the diode is likely to exceed the drive strength of the PXA3xx processor family multi-function pin. Use a drive transistor to overcome excessive diode current draw.

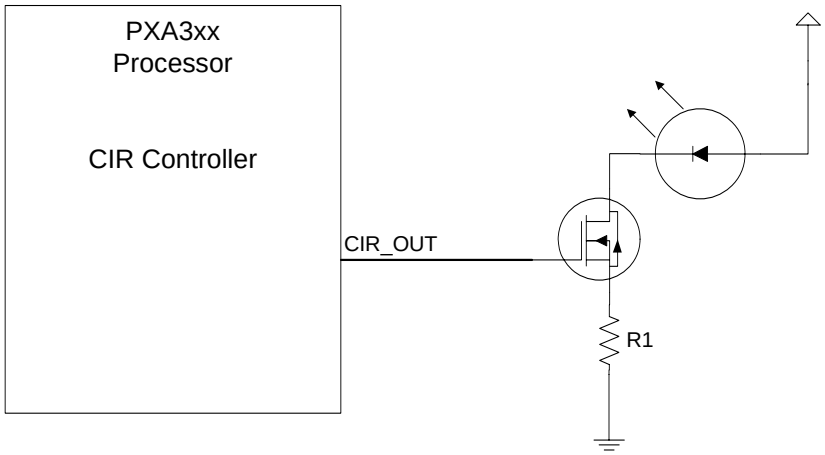
Table 80: CIR Signal Description

Signal Name	Input/Output	Description
CIR_OUT	Output	CIR unit output. Connects to IR diode.

24.3 Block Diagram

See Figure 86 for the block diagram showing the CIR output of the PXA3xx processor family interfaced to an RC5 LED transmitter. This example includes a drive transistor, which is recommended to limit the current sourced by PXA3xx processor family on the CIR_OUT multi-function pin. The value of R1 is dependant upon the infrared diode chosen. Refer to the diode manufacturer’s datasheets to choose an appropriate value.

Figure 86: Consumer Infrared Unit Connection Example



24.4 Layout Notes

The maximum specified switching speed of the CIR is 7.43 MHz, but applications use significantly lower carrier (switching) rates. For example, RC5 uses a 36 kHz carrier frequency. At this speed, degradation of signal integrity is not likely to occur. But, the use of the best possible design and routing practices is still encouraged.

Follow these recommendations when routing signals to ensure the most reliable design possible:

- Keep the signal trace as short as possible.
- Route the signal with as much of the trace on an inner signal layer as possible.
- The traces supplying power to the IR LED/drive transistor/current limit resistor should be sized to accommodate the current driven through the IR LED.

25 Pulse-Width Modulator Interface

25.1 Overview

The PXA3xx processor family pulse-width modulator (PWM) interface contains four pins that can be configured to generate periodic output signals. Configuration of the PWMs is accomplished through software and is described in detail in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. IV: Serial Controller Configuration Developers Manual*, “Section 9: Pulse-Width Modulator Controller”.

PWM signals are implemented through GPIOs. Therefore, the hardware considerations necessary for the PWM signals are the same as that of the GPIOs.

25.2 Signals

The PWM signals are implemented through the GPIOs of the PXA3xx processor family. Refer to the GPIO alternate-function table in *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* , “Section 5: General-Purpose I/O Unit” for the GPIO assignments of the PWM signals.

See Table 81 for the list of signals controlled by the PWM controller of the PXA3xx processor family.

Table 81: PWM Interface Signal List

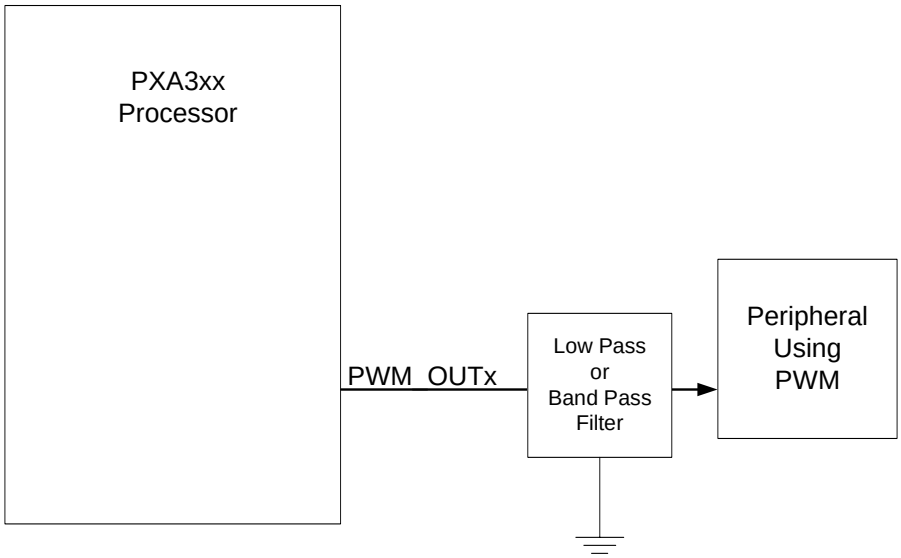
Signal Name	Type	Description
PWM_OUT0	Output	Pulse-width modulated output signal for PWM0
PWM_OUT1	Output	Pulse-width modulated output signal for PWM1
PWM_OUT2	Output	Pulse-width modulated output signal for PWM2
PWM_OUT3	Output	Pulse-width modulated output signal for PWM3

25.3 Block Diagram

The connection of the PWM to the peripheral device using the PWM is typically direct. However, if the PWM is used as a software-controlled voltage source, a low-pass filter must be placed in line. An example of a software controlled voltage source is for the contrast control on an LCD panel. If the PWM output is used for generating audio, an appropriate filter is also required. In this case, the filter is a band-pass filter.

The specification and design of the filter depends on the exact purpose of the application. See the block diagram for a design requiring a low-pass or band-pass filter in Figure 87.

Figure 87: PWM Block Diagram For Applications Requiring a Filter



25.4 Layout Notes

The maximum switching frequency of the PWM signals is approximately 1.6 MHz. For best results, adhere to all GPIO layout recommendations.

A typical use for a PWM output is the contrast control for LCD panels. In this situation, the panel backlight inverter is a large source of noise. To ensure that the noise is not induced into the contrast control or the processor, follow all recommendations for shielding the inverter and separating the inverter from the contrast control line (PWM output).

26 USIM Controller Interface

This section describes interface guidelines for the universal subscriber identity module (USIM) of the PXA3xx processor family.

26.1 Overview

The USIM controller is an interface for a GSM mobile handset. The USIM interface supports communication with SmartCards as specified in *ISO standard 7816-3* and technical specification *3G TS 31.101* of the 3rd Generation Partnership Project.

SmartCards are used in many applications and the GSM network USIM card is only one of those applications. SmartCards usually consist of a CPU, flash memory, and a serial communication interface device similar to the one described in this section. More sophisticated cards contain a PLL for frequency enhancement. Encryption accelerators also exist in a SmartCard as many applications are security-oriented. In all SmartCard applications, the physical layer and data link layer are identical, so the USIM module can serve both layers as well.

The USIM interface is composed of eight major components:

- Registers
- Card interface
- Receive FIFO (RXFIFO)
- Transmit FIFO (TXFIFO)
- Receiver
- Transmitter
- Clock generator
- Baud-rate generator

The transmitter converts data transferred to its TXFIFO to a serial output as specified by the standard. The receiver performs the reverse action and stores data taken from its RXFIFO. Each FIFO holds up to 16 bytes. The RXFIFO also holds a parity-error indicator for each byte.

Transforming byte convention (inverse to direct and vice-versa) is performed when the byte exits the FIFOs; that is, the FIFO output is either inverted or left as is, according to the session convention. In this manner, the program does not have to perform a format inversion before character receipt. Receiver, transmitter, and baud-rate generator blocks are clocked by an internal clock generated in the clock generator. This unit also generates the card clock. Both receiver and transmitter baud rates are controlled by the baud-rate generator. Software controls the session with the card by updating the USIM registers. All signals going to the USIM card are processed in the card interface block as shown in Figure 88.

26.2 Signals

This section describes the PXA3xx processor family USIM interface I/O signals. See Figure 88 for an illustration showing connectivity between the PXA3xx processor family and an external USIM card.

26.2.1 PXA3xx Processor Family USIM Interface Signals

See Table 82 for the list of all of the PXA3xx processor family USIM interface signals.

Table 82: USIM Interface Signals

Signal Name[3]	Direction	Description
UIO[2]	Bidirectional	USIM I/O Data - Receive and transmit data signal. The bi-directional signal should be connected directly to the USIM SmartCard. When asserted, the UIO signal is forced to Vlow. When de-asserted, the UIO signal is pulled up by a 20 KΩ external resister. The USIM SmartCard can act as a pull-down on the UIO signal. **NOTE:** When the USIM SmartCard and the processor operate at different voltage levels, the I/O voltage level of the SmartCard must be a voltage level that the processor supports, that is, 1.8 V or 3.0 V.
UVS0[1]	Output	Used as the select signal to a power transistor that controls the voltage level that is supplied to the SmartCard: 0 = OFF 1 = Selects zero-volts to the USIM SmartCard power-supply pin (VCC)
nUVS1[1]	Output	Used as the select signal to a power transistor that controls the voltage level that is supplied to the SmartCard: 0 = Selects 1.8 V to the USIM SmartCard power-supply pin (VCC) 1 = OFF
nUVS2[1]	Output	Used as the select signal to a power transistor that controls the voltage level that is supplied to the SmartCard: 0 = Selects 3.0 V to the USIM SmartCard power-supply pin (VCC) 1 = OFF
UCLK[2]	Output	SmartCard clock. UCLK should be connected directly to the SmartCard clock pin. The SmartCard cannot use any other clock source.
nURST[2]	Output	SmartCard reset. nURST should be directly connected to the SmartCard reset pin: 0 = Resets the SmartCard 1 = OFF

NOTE:

1. Use caution when placing the processor into S3/D4/C4 power mode (Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 8: Services Power Management Unit” for a detailed description of S3/D4/C4). The voltage control pins are powered from the VCC_IO power domain, but in S3/D4/C4 mode this is turned off. Powering off the voltage select pins (UVS0, nUVS1, nUVS2) can cause shorting of the power supplies that are switched to provide power to the SmartCard. To prevent these electrical problems:
   - The external power supply must be turned off before (or at the same time as) the PXA3xx processor family enters S3/D4/C4 power mode using USIM[USCCR] bit field.
   - GPIOs must be correctly configured prior to entering S3/D4/C4 mode via software writes to USIM[USCCR].
   - The control and status of this signal is defined using the PUCR in the PXA3xx processor family PX PMU module.
2. Voltage level for this signal pins is set off chip via the VCC_CARDx power domain pins. The signal pins work at either 1.8 V or 3.0 V.
3. There are two SmartCard USIM controllers in the PXA3xx processor family, both are identical in function. For one of the controllers, the signal names are preceded by a “U_”. The signal names for the other SmartCard controller are the same except preceded by an “SC_”. For example, the UIO signal maps to the U_IO pin for one of the SmartCard controllers, and the UIO signal maps to the SC_IO signal for the other SmartCard controller. See *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual*, “Section 4: Pin Descriptions and Control” for alternate function assignments.

26.2.2 USIM Card Interface Signals

See Table 83 for a list of the five USIM card signals that connect to the PXA3xx processor family USIM interface.

Table 83: USIM Card Signals

Card Signal	Function	Signal Direction	PXA3xx Processor Family Signal
I/O	Data input/output (bidirectional) pad with a pull-up resistor Refer to *ISO Standard 7816-3* for pad specifications.	USIM card <-> PXA3xx processor family	UIO
CARD_RST	USIM card reset	PXA3xx processor family –> USIM card	nURST
CLK_CARD	Clock input Frequencies are between 1.0 MHz and 5.0 MHz Refer to *ISO Standard 7816-3* for pad specifications. Clock stops on low or high phase are supported. **NOTE:** Cards manufactured before April 2000 have a frequency limitation of 4.0 MHz.	PXA3xx processor family –> USIM card	UCLK
VCC	Card power supply Supplies 0 V, 1.8 V, 3.0 V with maximum currents of 0 mA, 30 mA, 50 mA, respectively according to card class (B, C) Refer to *ISO Standard 7816-3* for pad specifications.	Power supply –> Card	VCC_CARDx
GND	Mutual USIM card, PXA3xx processor family USIM interface, and VCC ground reference voltage	Power supply –> Card	VSS_CARD

26.3 Block Diagram

See Figure 88 and Figure 89 for an illustration of the connectivity scenario between the PXA3xx processor family and an external USIM card. Figure 88 and Figure 89 shows configurations supporting both 1.8 V and 3.0 V USIM cards.

Refer to *Marvell® PXA30x, PXA31x, and PXA32x Processors Vol. I: System and Timer Configuration Developers Manual* for information on control/status programmability issues relating to an external USIM card.

Figure 88: Interfacing the First USIM Controller to a USIM SmartCard

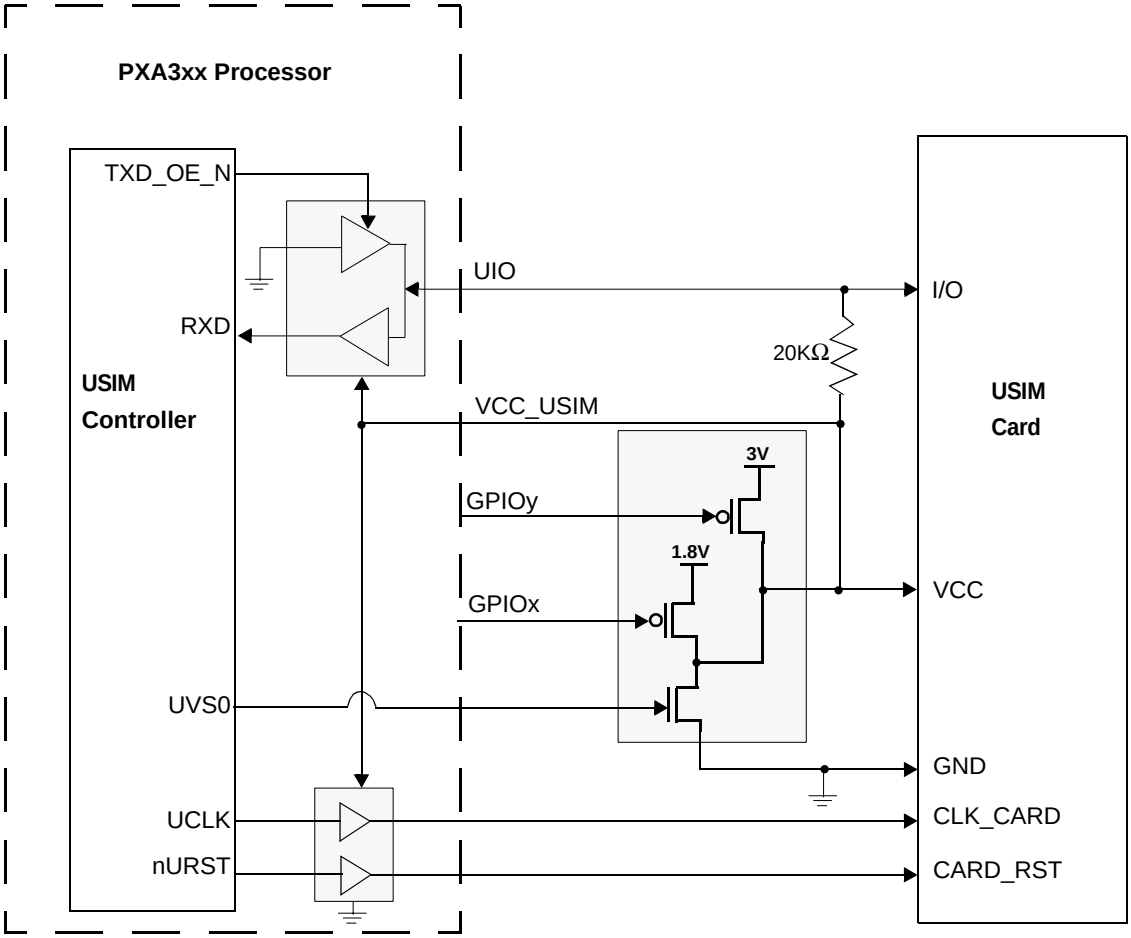
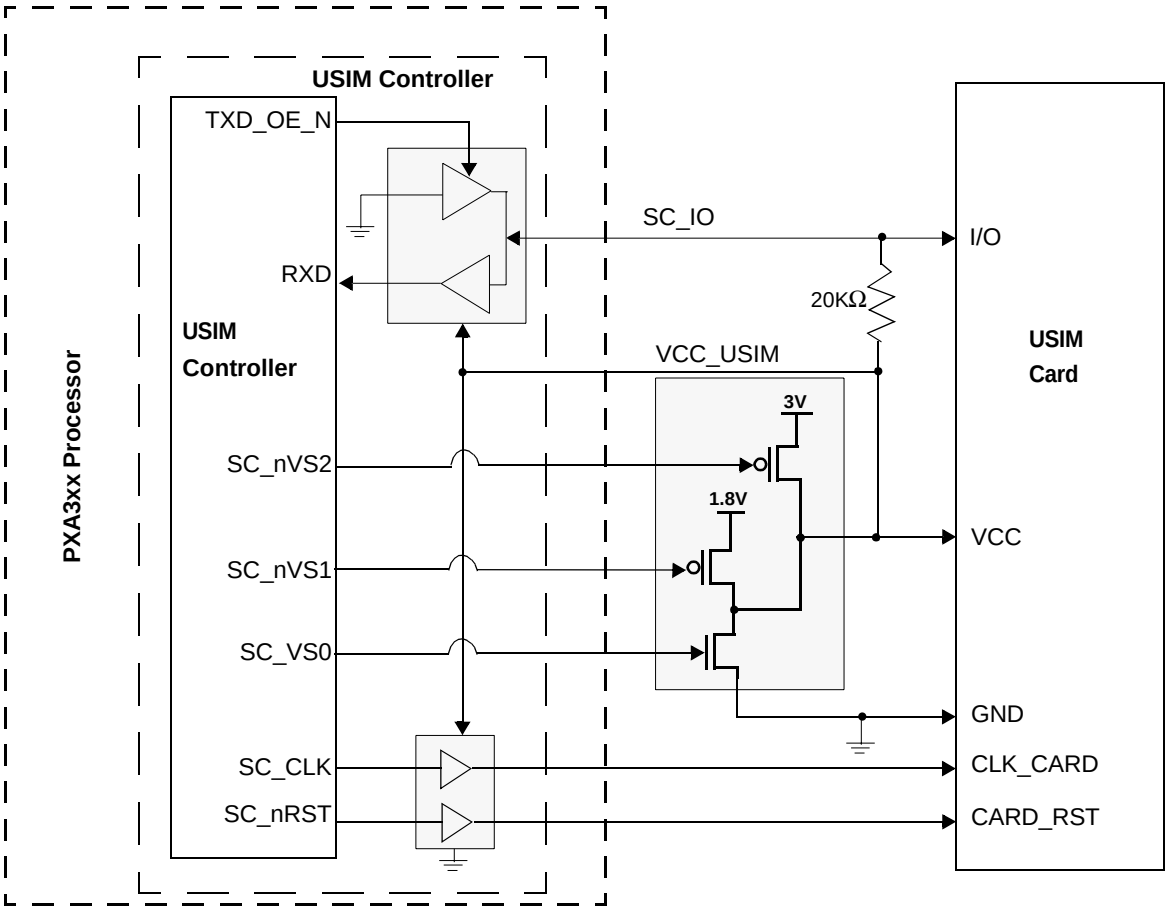


Figure 89: Interfacing the Second USIM Controller to a USIM SmartCard



26.4 Layout Notes

Voltage levels for the PXA3xx processor family USIM pads are set off-chip, that is, external to the PXA3xx processor family. The pads work at either 1.8 V or 3.0 V. When the USIM card and the PXA3xx processor family operate at different voltage levels, the I/O voltage level of the USIM card must be at a voltage level supported by the PXA3xx processor family.

Warning

Use caution when placing the PXA3xx processor family into S3/D4/C4 mode. The voltage control signals are powered from the VCC_IO power domain. In S3/D4/C4 mode, this power is turned off. Powering off the voltage-select signals (UVS0, nUVS1, and nUVS2) causes shorting of the power supplies that are switched to provide power to the card. See notes from Table 82 for information on how to prevent these electrical problems.

27 Inter-Integrated Circuit (I^2C)

27.1 Overview

The PXA3xx processor family has two I^2C peripherals: the standard I^2C interface and the Power I^2C interface. The I^2C bus is a serial bus based on a standard developed by the Philips Corporation consisting of a two-pin interface. It requires minimal hardware to set up an economical communication system for relaying status and reliability information to an external device. A complete list of features and capabilities are found in the *I^2C Bus Specification*.

The standard I^2C bus serves as the PXA3xx processor family interface to other I^2C peripherals and microcontrollers, as well as a method of managing system functions. The standard I^2C bus allows the PXA3xx processor family to serve as a master and slave device residing on the I^2C bus. Control and status information is relayed through a set of memory-mapped registers.

The Power I^2C interface contains a subset of the standard I^2C interface and is dedicated for connection to an external voltage regulator for hard coded power management communication. This interface allows the transmission of pre-determined I^2C command sequences to the PMIC. These sequences are fixed and cannot be altered. If PWR_EN or SYS_EN change state (for example during a power state change), the appropriate I^2C commands are transmitted to the PMIC as well. No configuration or software interaction is required. The Power I^2C interface can not be used as a general- purpose I^2C interface.

Both the standard I^2C and the Power I^2C interfaces have a standard bus speed of 100 kbps and a fast-mode operation of 400 kbps. Refer to the *Marvell® PXA30x, PXA31x, and PXA32x Processor Developers Manual, Vols. I-IV* for information about the Power I^2C interface.

If the Power I^2C interface is not used to control a PMIC, the standard I^2C interface can be connected to the PMIC instead. However, if the standard I^2C controller is used, the discrete power signals PWR_EN and SYS_EN must be connected to the PMIC.

27.2 Signals

See Table 84 for the description of the I^2C bus interface unit signals.

Table 84: I^2C Signal Description

Signal Name	Input/Output	Description
SDA	Bidirectional	Serial data for the standard I^2C controller. **NOTE:** For 3.3 V or less on I^2C controller, 1.2 KΩ pull-up resistor must be used.
SCL	Bidirectional	Serial clock for the standard I^2C controller. **NOTE:** For 3.3 V or less on I^2C controller, 1.2 KΩ pull-up resistor must be used.
PWR_SDA	Bidirectional	Serial data for Power I^2C controller. **NOTE:** For 3.3 V or less on I^2C controller, 1.2 KΩ pull-up resistor must be used.

Table 84: I²C Signal Description (Continued)

Signal Name	Input/Output	Description
PWR_SCL	Bidirectional	Serial clock for Power I²C controller. **NOTE:** For 3.3 V or less on I²C controller, 1.2 KΩ pull-up resistor must be used.

The I²C bus serial operation uses an open-drain, wired-AND bus structure that allows multiple devices to drive the bus lines and to communicate status on events such as arbitration, wait states, and error conditions. For example, when a master drives the SCL during a data transfer, it transfers a bit every time the clock is high. When the slave is unable to accept or drive data at the rate that the master is requesting, the slave holds SCL low between the high states to insert a wait interval. The master clock is only altered by a slow slave peripheral keeping the clock line low or by another master during arbitration.

The I²C bus allows design of a multi-master system, meaning more than one device can initiate data transfers at the same time. To support this feature, the I²C bus arbitration relies on the wired-AND connection of all I²C interfaces to the I²C bus. Two masters can drive the bus simultaneously, provided they are driving identical data. The first master to drive SDA high while another master drives SDA low loses the arbitration. The SCL consists of a synchronized combination of clocks generated by the masters using the wired-AND connection to the SCL.

27.3 Layout Notes

The maximum switching frequency of the I²C signals is 400 kHz. Therefore, layout and routing considerations are not stringent. However, for best results, adhere to common layout recommendations.

Separate the physical routing of the data and clock signals and ensure that lines are not routed near other potential noise sources, such as switching regulators or signals with high switching frequencies.

Note

Be sure and connect all I²C signals to the correct voltage levels using 1.2 KΩ pull-up resistors.

A Design Checklist

This Appendix is a design-review checklist for systems using the PXA3xx processor family prior to layout, during layout, during bring-up, and software optimization. It is a good idea to consult this section if any problems occur while bringing up the PXA3xx processor family system for the first time. Each table within this section has a column reserved for reviewing designers to verify the line item has been checked.

A.1 Schematic Checklist

Table 85 briefly describes requirements to add in schematics reviews. Refer to the various sections within this Design Guide for more information about these requirements.

Table 85: PXA3xx Processor Family Schematic Checklist

✓	Signal Name	Category	Details
	Power Domains		
	All VCC power domains	Verify proper decoupling	Check all VCC power domains for proper decoupling – both Bulk and Edge decoupling capacitors. Recommend using 0402, 1.0 µF, X5R, low inductance, low ESR capacitors. See Section 2.2, Power Supply Decoupling Requirements
	All power domain connections	Verify connections	Power Distribution: • Verify each power domain gets proper power/voltage (including power-up sequencing) • Verify each chip (DDR, SRAM, FLASH, companion chips, etc.) has proper power/voltage • Analog sections may need their power filtered if using common regulator • Check for maximum required current and total available current for each voltage rail
	nBATT_FAULT	Weak pull-up	Add weak pull-up (50 KΩ) to nBATT_FAULT
	Test and Reset Signals		
	TEST	Tie to GND	Required on all systems during normal operation. Reserved for factory testing.
	TESTCLK	Tie to GND	Required on all systems during normal operation. Reserved for factory testing.
	nRESET_OUT	Level shifter may be required	Driven high at VCC_BBATT voltage level. Need to level-shift down to 1.8 V when used as reset on 1.8 V components.
	Clocks and Power		
	PXTAL_IN, PXTAL_OUT	13 MHz crystal only	No other discrete components required or recommended. Do not connect load capacitors from either signal to ground.

Table 85: PXA3xx Processor Family Schematic Checklist (Continued)

✓	Signal Name	Category	Details
	PXTAL_IN, PXTAL_OUT	External clock source	Connect 13 MHz clock to PXTAL_IN and leave PXTAL_OUT floating.
	TXTAL_IN, TXTAL_OUT	32.768 kHz crystal only	Both TXTAL_IN and TXTAL_OUT require 22 pF capacitor connection to ground. Capacitor requires: 20.9 pF (min.), 23.1 pF (max), Temperature coefficient equals +/- 30 ppm/C
	TXTAL_IN, TXTAL_OUT	External clock source	Connect 32.768 KHz clock to TXTAL_IN and TXTAL_OUT to ground.
	CLK_TOUT and all high speed signals/clocks	High speed digital signals	Do not run high-speed digital signals or clocks (especially CLK_TOUT) directly underneath the crystals and traces to the processor.
	PWR_OUT	0.1 µF cap to GND	Required on all systems
	PWR_CAP0 and PWR_CAP1	0.1 µF cap	Required connections between PWR_CAP0 and PWR_CAP1 signals for DC-DC convertor operation.
	One-Wire Interface		
	ONE_WIRE	5 KΩ pull-up to VCC_IOx	This signal must be pulled high by a resistor 5 KΩ, nominally. The required connection for VCC_IOx depends on the GPIO used.
	All Memory Interfaces		
	Address busses	Verify connections	Verify memory device address signals are connected properly within the design.
	Chip selects and control signals	Verify connections	Verify correct chip selects and control signals go to the correct memory devices: • Do not mix EMPI and DFI signals • DF_nCS0 and DF_nCS1 are for the NAND Flash Controller (NFC) • nCS0 and nCS1 are for the Static Memory Controller (SMC) (PXA30x Processor and PXA31x Processor only) • nCS2 and nCS3 are for the Static Memory Controller (SMC) • nSDCS0 and nSDCS1 are chip selects for Dynamic Memory Controller (DMC)
	DDR SDRAM		
	RCOMP_DDR	40.2 Ω 1%-100 mW resistor to GND	RCOMP_DDR signal requires 40.2 Ω 1%-100 mW resistor connected to GND. Keep the RCOMP breakout as small as physically possible to reduce effective R. Specifically, keep trace as short as possible (<0.1”) and as wide as possible (>0.010”).
	SDMA10	Connect to A10 on DDR SDRAM devices	Verify SDMA10 is connected to A10 on all DDR SDRAMs. If this signal is not connected, the DDR auto-precharge and refresh will not work properly.
	DDR Bus	Maximum loading	Maximum loading is 4 loads
	DFI Bus		

Table 85: PXA3xx Processor Family Schematic Checklist (Continued)

✓	Signal Name	Category	Details
	NAND Controller	Proper support of NAND devices	Verify NAND devices are supported by the PXA3xx processor family NAND controller.
	DFI Bus	Data transceivers	The data transceiver is optional for NOR Flash, SRAM and VLIO depending on Bus loading.
	DFI Bus	Maximum loading	Maximum loading is 4 loads. Care must be taken to ensure even loading of data signals vs. address vs. clocks vs. control signals.
	Power I^2C and Standard I^2C		
	SDA and SCL	1.2-KΩ pull-up resistor	For 3.3 V or less on I^2C controller, 1.2 KΩ pull-up resistor must be used.
	PWR_SDA and PWR_SCL	1.2 KΩ pull-up resistor	For 3.3 V or less on I^2C controller, 1.2 KΩ pull-up resistor must be used. **NOTE:** Must connect to a PXA3xx processor family-compatible PMIC that responds to the hard coded commands. Do not use Arrava PMIC that is on the Zylonite board. Instead, use a PXA3xx processor family-compatible PMIC
	All I^2C signals	Pull-up voltage	Verify all I^2C signals are pulled up to the proper voltage. Inside the PXA3xx processor family the I^2C signals are on the following voltage domains: **PXA32x:** SDA/SCL are on VCC_IO4, PWR_SDA/PWR_SCL are on VCC_IO1. **PXA30x and PXA31x:** SDA/SCL are on VCC_IO3, PWR_SDA/PWR_SCL are on VCC_IO1
	LCD Controller		
	LDDx	LCD Data Signals	Verify LCD data signals are connected correctly for active mode per Table 47, Smart Panel Active Display Pins Required.
	MMC/SD/SDIO Controller		
	MMDAT1	Connect to GPIO wake-up source externally.	To enable SDIO cards to wake up the PXA3xx processor family the MMDAT1 signals must be connected externally to a GPIO capable of waking up the PXA3xx processor family. See the Power Manager Wake- Up Enable register (PWER) in the *Marvell® PXA30x, PXA31x, and PXA32x Processor System and Timer Configuration Developers Manual, Vol. I* for GPIO wake-up sources. External logic is required when using GPIO wake-up signal that is only able to detect a rising edge.
	MMDATx	50 K – 100-KΩ pull-up resistor to VCC_IO	Covers requirements for SD card and MMC socket. SD card socket allows down to 10 KΩ.
	MMCMDx	10 K – 100-KΩ pull-up resistor to VCC_IO	Covers requirements for SD Card and MMC socket. MMC socket allows down to 4.7 KΩ.
	MMC and SD/SDIO Card Socket Pins VDD, VSS1	0.1 µF capacitor decoupling cap required	The 0.1 µF capacitor (C1) shown in Figure 40 and Figure 40 must be located within 0.512 inches (13 mm) of the MMC socket and SD/SDIO Card socket pin 3 (VDD) and pin 4 (VSS1).

Table 85: PXA3xx Processor Family Schematic Checklist (Continued)

✓	Signal Name	Category	Details
	Keypad Interface		
	KP_MKINx and KP_DKINx	No connection	No external pull-down resistor is required. The PXA3xx processor family includes the required internal pull-down resistor to ground.
	Universal Subscriber ID (USIM) Interface		
	SC_IO	20 KΩ pull-up resistor to VCC_CARD1.	Required only if using the USIM controller. See Section 26, USIM Controller Interface for additional information. Leave the SC_IO signal disconnected if the USIM controller is never used.
	Universal Serial Bus Device Controller (UDC)		
	USBOTG_P	1.5 KΩ pull-up resistor using GPIO signal.	Some designs may be able to use the internal pull-up resistor. See Section 17, USB 1.1 Full Speed OTG and Device Controller (PXA30x processor or PXA32x processor only) for additional information. Signal must remain floating or pull-up (1.5 KΩ) to minimize power use during standby, sleep, and deep sleep modes.
	USBOTG_N	Floating or pull-up	Signal must remain floating or pull-up (10 K - 100 KΩ) to minimize power use during standby, sleep, and deep sleep modes.
	All Signals	—	See Section 17, USB 1.1 Full Speed OTG and Device Controller (PXA30x processor or PXA32x processor only) for different possible configurations.
	Universal Serial Bus Host Controller (UHC)		
	All Signals	—	See Section 20, USB 1.1 Host Interface for different possible configurations.
	Host Ports 1 and 2	Termination	See Section 20.3.1, USB Host Differential Connection (Port 1) , and Section 20.3.2, USB Host Connections (Port 2) which shows a block diagrams detailing host port termination and connection requirements.
	General Purpose IO (GPIO) Controller		
	GPIOx	Pull-down unused GPIOx signals.	Unused GPIO signals can be either pull-down or driven as outputs to minimize power use. These states are software configurable.
	nGPIO_RESET	Weak pull-up	nGPIO_RESET should have a weak pull-up (50 KΩ). The internal pull-up can be disabled via software.
	JTAG Debug		

Table 85: PXA3xx Processor Family Schematic Checklist (Continued)

✓	Signal Name	Category	Details
	nTRST	For JTAG operation, must be driven from both the system reset and the JTAG tools independently without contention.	During power-up, nTRST must be driven from low to high either before or at the same time as nRESET. During power-up, 10 µs must elapse after nTRST is de-asserted before proceeding with any JTAG operation. See Section 9, JTAG Debug for additional information. Pullup to VCC_BBATT using a 10K resistor.
	TCK	For JTAG operation, use 10K ohm pulldown resistor.	See Section 9, JTAG Debug for additional information.
	TMS	Floating or pullup.	Signal must remain floating or pullup (10-100 KΩ) to minimize power use during standby, sleep, and deep sleep modes.
	TDI	Floating or pullup	Signal must remain floating or pull-up (10-100 KΩ) to minimize power use during standby, sleep, and deep sleep modes.
	TDO	No pullups or pulldowns required	Connect from JTAG connector. No pull-ups or pull-downs required.
	nSRST	Bidirectional	nSRST is not a signal on the PXA3xx processor family. The ARM standard JTAG connector specifies the nSRST should be bidirectional and allow JTAG to both sense and drive the CPU reset (nRESET). A JTAG module connected to an XScale® processor must be able to control the nRESET signal in order to enable proper JTAG debugger communication. See Section 9, JTAG Debug for additional information.
	Typical Voltage Rail Connections		
	PMIC OUTPUT 1	Voltage Supply	Regulated Main battery voltage, nominally 3.0V (not to exceed max 4.0V) to power VCC_BBATT
	PMIC OUTPUT 2	Voltage Supply	VCC_MEM @ 1.8V
	PMIC OUTPUT 3	Voltage Supply	VCC_BG, VCC_MVT, VCC_PLL, VCC_OSC13M connected together @ 1.8V
	PMIC OUTPUT 4	Voltage Supply	VCC_LCD, VCC_MSL, VCC_CI, VCC_DF connected together at 1.8V
	PMIC OUTPUT 5	Voltage Supply	VCC_APPS dynamically adjustable
	PMIC OUTPUT 6	Voltage Supply	VCC_SRAM dynamically adjustable
	PMIC OUTPUT 7	Voltage Supply	VCC_IO1, VCC_IO3, VCC_IO4 and VCC_IO6 connectected together at 1.8V, 3.0V or 3.3V (depends on voltage required for external components) **NOTE:** VCC_IO4 and VCCIO6 only apply to the PXA32x Processor
	PMIC OUTPUT 8	Voltage Supply	VCC_CARD1 and VCC_CARD2 connected together at 1.8V or 3.0V (+/- 10%) to USIM or Removable storage card. This can be supplied by another regulator passing through an analog power switch. This supply can be connected to another supply of the appropriate voltage is USIM functions are not used.

Table 85: PXA3xx Processor Family Schematic Checklist (Continued)

✓	Signal Name	Category	Details
	PMIC OUTPUT 9A	Voltage Supply	VCC_USB at 3.3V. This supply can be connected to another supply of appropriate voltage is USB functions are not used.
	PMIC OUTPUT 9B	Voltage Supply	VCC_TSI (Touch Screen Interface): can be passively filtered output of the USB regulator. This can be connected to another supply of the appropriate voltage if TSI functions are not used. **NOTE:** PXA32x Only
NOTE: Connections to external Power supplies depend on overall system requirements and the PMIC being used. The above recommendations may vary depending on board and processor requirements.			
	Miscellaneous Schematic Checklist Items		
	Verify PXA3xx processor family symbol pin-out matches *Marvell® PXA30x, PXA31x, and PXA32x Processor Electrical, Mechanical, and Thermal Specification*		
	Verify net names are consistent across schematic. If using bussed nets, verify signal numbers match on all symbols where used.		
	PXA3xx processor family nRESET input should be connected to PMIC's nRESET out or main System Watchdog/Reset controller.		
	nRESET_OUT connected to devices not needed at Boot. Do not connect to M-Systems M-DOC or Samsung OneNand.		
	If possible, provide for options to margin voltage supplies for product reliability testing (4 corners testing)		
	Pull-up/pull-downs on active low/high signals and USB signals; ensure that no memory devices can be selected in low-power modes or power down (because of floating signals)		
	Verify LED Polarity		
	Put test points on critical signals to aid debugging boards (if possible)		
	Put test points on unused GPIO signals if board space permits		
	Add debug features, especially on non-form-factor boards (full test points, logic analyzer pads – Tektronix DMAX connectorless test connections)		

Note

Marvell's schematic review checklist does not replace a customer's in-house design review or substitute for training in design or Marvell architecture basics. Although Marvell makes a good faith effort to find potential design problems, the customer remains responsible for the success of their design. Marvell makes no claims or guarantees that the Marvell checklist will uncover all defects, or that the design will work. Neither does Marvell accept responsibility for any impact to the customer's project schedules.

A.2 Bring-up Checklist

This section consists of two tables. Use Table 85 for a bring-up overview checklist and Table 87 for a complete list of bring-up signals to monitor prior to and including the first instruction being fetched from flash (nCS<0>).

Table 86: Bring-up Overview Checklist

✓	Items
	Before assembly, test continuity between each power domain and GND.
	After board is assembled and before power is applied, test continuity between each power domain and GND.
	Verify all components were populated and the correct values were used. Verify component orientation is correct (especially BGA and QFP components.)
	Verify all IC's and polarized caps are properly oriented (IC pin 1 to PCB footprint pin 1)
	Have schematics and layout files ready to help locate components and routing
	Before turning on power, verify power rails are not shorted to GND
	Verify all power supply voltage levels are within tolerances.
	To locate problem areas, check for overheating components and check current measurements.
	Observe the following signals to check if the PXA3xx processor family is functioning: (See the complete list of signals to bring the PXA3xx processor family out of reset.) • VCC_BBATT • PWR_EN and SYS_EN should be asserted (high) • nRESET should be asserted low for at least 20 msec before being de-asserted • nRESET_OUT should be de-asserted "a specified time" after nRESET is de-asserted • Crystal signals (PXTAL_IN, PXTAL_OUT, TXTAL_IN, TXTAL_OUT)[1] • VCTCXO_EN is high if used to enable the PXA3xx processor family oscillator • Check boot device chip select activity: nCS2 (NOR Flash) or DF_nCS0 (NAND Flash) should be toggling, indicating reads from FLASH. FLASH reads may only last a short time if the processor is reading a bootloader into SRAM or DDR
	Use simple code to initialize the processor and toggle LEDs or a GPIO signal to indicate activity
	Use JTAG for testing systems and stepping through first boot code.
	Number your new PCB boards to keep track of issues (system1, system2, etc...).
	Write down issues and required board changes whenever you encounter them.
	Keep a non-populated board around for continuity tests and for help finding component locations.
	For data integrity problems, first try changing EMPI (DDR) buffer impedance (30 or 32 ohms) or try changing DFI drive strength (1x – 12x)
	If test points or vias are available, use a high-speed oscilloscope to check signal integrity of all memory signals
	Tips for high speed oscilloscope probing: • When probing high-speed signals, it is important to use VERY short ground wires to the scope probe. Example: solder a 1" – 1.5" 24 AWG wire to a GND on the PCB and connect it directly to the scope probe's GND connection. • Probe high-speed signals with an oscilloscope with an analog bandwidth 5-10 times higher than the signal of interest (Use a 1 GHz scope to probe 100 MHz signals). • Scope probes for high-speed signals must not load down the signal being probed, so use probes with a bandwidth 10 times higher than the signal of interest; use low-capacitance FET probes or high-bandwidth resistive probes
	Verify GPIO pins are configured for the correct use; for example, the FFUART pins may also be configured as USB pins or MMC pins

Table 86: Bring-up Overview Checklist (Continued)

✓	Items
	If a silicon bug exists that affects a product, document the problem and its implications. Provide: • Logic analyzer or oscilloscope traces • Source code • Methods for easy replication of the problem to the Marvell support team
	NOTE: 1. probing a crystal's pin with an oscilloscope may cause it to change its frequency or stop running

Most of the PXA3xx processor family signals are multiplexed so the signals can be configured for one of seven available alternate functions using the GPIO alternate-function select registers. Some signals can be configured to appear on one of several different pins.

For pins that have more than one alternate function, use the check list in Table 87 only for the signal name whose function is configured for a pin with alternate functions.

Table 87: Signals to Monitor During Bring-up of the PXA3xx Processor Family

✓	Signal Name	Pin Connection	Details
	PWR_EN	PMIC input signal	Power Enable (Output) – Should be used to enable these supplies: • VCC_APPS • VCC_SRAM
	SYS_EN	PMIC input signal	System Enable (Output) – Should be used to enable these supplies: • VCC_MVT • VCC_BG • VCC_OSC13M • VCC_PLL • VCC_MEM • VCC_IO1 • VCC_IO3 • VCC_IO4 (PXA32x only) • VCC_IO6 (PXA32x only) • VCC_USB (PXA30x and PXA32x only) • VCC_MSL • VCC_DF • VCC_LCD • VCC_CARD1 • VCC_CARD2 • VCC_CI • VCC_BIAS (PXA31x only) • VCC_ULPI (PXA31x only)
	PWR_OUT	0.1 µF cap to GND	Power Out (Power Input Signal)
	PWR_CAP<0> PWR_CAP<1>	0.1 µF cap between the pins	The PWR_CAP signals connect to external capacitors used to achieve very low power in sleep and deep sleep mode.
	PWR_CAP<2> PWR_CAP<3>	0.1 µF cap between the pins	The PWR_CAP signals connect to external capacitors used to achieve very low power in sleep and deep sleep mode.
	nBATT_FAULT	PXA3xx processor family input	The nBATT_FAULT signal is an active-low input that signals the PXA3xx processor family that the main battery is low or has been removed from the system.

Table 87: Signals to Monitor During Bring-up of the PXA3xx Processor Family (Continued)

✓	Signal Name	Pin Connection	Details
	Clock Signals		
	PXTAL_IN	13.000 MHz crystal or to an external clock source	Processor crystal input
	PXTAL_OUT	13.000 MHz crystal	Processor crystal output
	TXTAL_IN	32.768 kHz crystal or external clock source	Distributes to the timekeeping control system
	TXTAL_OUT	32.768 kHz crystal	Distributes to the timekeeping control system
	Reset Signals		
	nRESET	Must be debounced to cause clean assertion of nRESET for a minimum of 20 msec.	Reset Active low, level sensitive input that starts the PXA3xx processor family from the reset vector at address 0. Assertion causes the current instruction to terminate abnormally and a reset occurs. When nRESET is driven high, the processor starts execution from address 0. nRESET must remain low until the power supply is stable and the internal 13.000 MHz oscillator has stabilized.
	nRESET_OUT	Output	Reset Out Asserted when nRESET is asserted and de-asserts after nRESET is deasserted, but before the first instruction fetch. nRESET_OUT is also asserted for “soft” reset events: sleep, watchdog reset, or GPIO reset.
	JTAG and Test Signals		
	nTRST	Must be driven before or at the same time as the hardware nRESET pin for correct TAP controller and device operation.	JTAG test reset. IEEE 1194.1 test reset.
	TDI	Not Connected when JTAG interface is not used	JTAG test data input (TDI) The JTAG controller sends data to the PXA3xx processor family using the test signal. The pin has an internal pull-up resistor, eliminating the need for a connection if the JTAG interface is not used.
	TMS	Not Connected when JTAG interface is not used	JTAG test mode select (TMS) Selects the test mode required from the JTAG controller. The test signal has an internal pull-up resistor, eliminating the need for a connection if the JTAG interface is not used.
	TCK	Tied Low when JTAG interface is not used	JTAG test clock Clock for all transfers over the JTAG test interface.
	TEST	Tied Low for normal operation	Test Mode Reserved for manufacturing test. Must be grounded for normal operation.

Table 87: Signals to Monitor During Bring-up of the PXA3xx Processor Family (Continued)

✓	Signal Name	Pin Connection	Details
	TESTCLK	Tied Low for normal operation	Test Clock Reserved for manufacturing test. Must be grounded for normal operation.
	Memory Controller Signals		
	nCS<3:0>	Chip select signals for memory devices	Chip select 0, 1, 2, 3
	DQM<3:0>	Corresponds DQM0 –> MD<7:0> DQM1 –> MD<15:8> **PXA32x only:** DQM3 –> MD<31:24> DQM2 –> MD<23:16>	• SDRAM DQM data byte mask control for data bytes 1 and -0 **PXA32x only:** • SDRAM DQM data byte mask control for data bytes 3 and 2
	nSDRAS	Output	SDRAM RAS Connect to the row address strobe (RAS) pins for all banks of SDRAM.
	nSDCAS	CAS for SDRAM	SDRAM CAS Connect to the column address strobe (CAS) pins for all banks of SDRAM.
	SDCKE	Output	SDRAM clock enable The signal is de-asserted during sleep. SDCKE1 is always de-asserted upon reset. The memory controller provides control register bits for deassertion.
	DF_SCLK_E	Synchronous Flash Only	Synchronous static memory clocks
	SDCLK0, SDCLK1	SDRAM partition pair 0/1	DDR SDRAM memory clocks
	RD/nWR	Transceivers Directional Signal	Special care must be taken when the RD/nWR signal is used with external transceivers during any reset mode and the power modes (idle, standby, sleep, and deep sleep). A problem occurs when the last transaction is a read (RnW indicates a read), but the PXA3xx processor family drives the data bus to keep the data from floating.

A.3 PCB Layout Checklist

The information in Table 88 is intended for use by those responsible for starting or reviewing a PCB layout for the PXA3xx processor family.

Table 88: Layout Checklist

✓	Items
	Route DDR memory signals as described in Section 10.4, EMPI Layout Notes .
	Design for power measurements, if possible.
	If there is any chance the signal might need to be accessed, bring the signal out to a 0 ohm resistor or a test point.

Table 88: Layout Checklist (Continued)

✓	Items
	Minimize layer transitions of high speed signals.
	Double check ball out and GPIO alternate function mapping.
	For best results, use the Marvell-recommended pad size for mounting the PXA3xx processor family to the PCB.
	Consider part size and assembly issues for parts placement.
	Consider the space needed to interface cables to the platform.
	Keep high speed signals short and minimize stubs.
	Separate analog and digital signals to minimize cross talk.
	Route like signals together (for example, SDRAM data, address, and control can all be grouped separately.) Routing of like signals helps in reducing the effects of cross talk.
	Using the PXA3xx processor family Development Platform, consider testing interface of the PXA3xx processor family to any desired devices, such as a camera module, LCD, and digital basebands.
	USBOTG_P, USBOTG_N, USBH1_P, and USBH1_N are differential pair signals. Route the signals close to each other as parallel traces on the PCB. Match the trace lengths as closely as possible (within ±0.5 inches (12.7 mm)).

A.4 Low Power Considerations Checklist

The information in Table 89 is intended for use by individuals responsible for developing a low power product that uses the PXA3xx processor family.

Table 89: Low Power Considerations Checklist

✓	Items
	Use the Marvell-provided power managment software early in the design process. Optimal power management is a significant task that takes considerable time to implement.
	Use 1.8 V memory bus interface when possible.
	Configure all GPIOs to the right state in sleep mode (do not drive pull-downs high).
	Don’t leave inputs floating.
	Measure power early and often.
	Power down board devices and peripherals when not in use. Use device low power modes when inactive.
	Keep track of all peripheral power consumption. (How much power does the LCD consume? What about your radio?)
	If possible, take advantage of quadrant refresh using extended mode registers within the LVSDRAM (most full featured OSs do not currently support partial refresh).
	Take advantage of the temperature compensated self refresh using extended mode within the LVSDRAM.
	Program the SDRAM refresh interval (MDREFR[DRI]) to an optimal value to avoid excessive refreshes.

Marvell Semiconductor, Inc.
5488 Marvell Lane
Santa Clara, CA 95054, USA

Tel: 1.408.222.2500
Fax: 1.408.752.9028

www.marvell.com

Marvell. Moving Forward Faster