Intel® PXA27x Processor Family

Design Guide

May 2005

Order No. 280001-002

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® PXA27x Processor Family may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation.

This document and the software described in it are furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Contact the local Intel sales office or the distributor to obtain the latest specifications and before placing the product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Copyright © Intel Corporation, 2005

AlertVIEW, i960, AnyPoint, AppChoice, BoardWatch, BunnyPeople, CablePort, Celeron, Chips, Commerce Cart, CT Connect, CT Media, Dialogic, DM3, EtherExpress, ETOX, FlashFile, GatherRound, i386, i486, iCat, iCOMP, Insight960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel ChatPad, Intel Create&Share, Intel Dot.Station, Intel GigaBlade, Intel InBusiness, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetStructure, Intel Play, Intel Play logo, Intel Pocket Concert, Intel Quick Capture Technology, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel TeamStation, Intel WebOutfitter, Intel Xeon, Intel XScale, Itanium, JobAnalyst, LANDesk, LanRover, MCS, MMX, MMX logo, NetPort, NetportExpress, Optimizer logo, OverDrive, Paragon, PC Dads, PC Parents, Pentium, Pentium II Xeon, Pentium III Xeon, Performance at Your Command, ProShare, Quick Capture Technology, RemoteExpress, Screamline, Shiva, SmartDie, Solutions960, Sound Mark, StorageExpress, The Computer Inside, The Journey Inside, This Way In, TokenExpress, Trillium, Vivonic, VTune, and Wireless Intel SpeedStep Technology are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Contents

Part I

1 Introduction to Part I ..... I: 1-1

1.1 Document Organization and Overview ..... I: 1-1
1.2 Functional Overview ..... I: 1-2
1.3 Package Introduction ..... I: 1-3
1.4 Signal Pin Descriptions ..... I: 1-4

2 PCB Design Guidelines ..... I: 2-1

2.1 Intel® Flash Memory Design Guidelines ..... I: 2-1
2.2 General PCB Characteristics ..... I: 2-1
2.2.1 PCB Layer Assignment (Stackup) ..... I: 2-2
2.2.2 PCB Component Placement ..... I: 2-4
2.2.3 PCB Escape Routing ..... I: 2-7
2.2.3.1 VF-BGA Escape Routing ..... I: 2-8
2.2.3.2 FS-CSP Escape Routing ..... I: 2-9
2.2.3.3 PBGA Escape Routing ..... I: 2-10
2.2.4 PCB Keep-out Zones ..... I: 2-11
2.2.5 Recommended Mobile Handset Dimensions ..... I: 2-11
2.3 Power Supply Decoupling Requirements ..... I: 2-12
2.4 Thermal Considerations ..... I: 2-12
2.5 Package to Board Assembly Process ..... I: 2-12
2.6 Silicon Daisy Chain (SDC) Evaluation Units ..... I: 2-12
2.7 Handling: Shipping Media ..... I: 2-12
2.8 Preconditioning and Moisture Sensitivity ..... I: 2-13
2.9 Tray Specifications ..... I: 2-13

3 Design Check List ..... I: 3-1

4 Mixed Voltage Design Considerations ..... I: 4-1

4.1 Overview ..... I: 4-1
4.2 Required Power Supplies ..... I: 4-1
4.3 Example Power Supply Utilizing Minimal Regulators ..... I: 4-2
4.4 Cautions ..... I: 4-4

5 Power Measurements ..... I: 5-1

5.1 Overview ..... I: 5-1
5.2 Measurement Guidelines ..... I: 5-1
5.2.1 Measure Voltage Across a Series Resistor ..... I: 5-1
5.2.2 Measure Current Directly with a Current Meter in Series ..... I: 5-1
5.3 Achieve Minimum Power Usage During All Power Modes ..... I: 5-2
5.4 Achieve Minimum Power Usage During Deep Sleep ..... I: 5-2
5.5 Achieve Minimum Power Usage During Sleep ..... I: 5-3
5.6 Achieve Minimum Power Usage During Standby ..... I: 5-3
5.7 Achieve Minimum Power Usage During Idle/13M/Run/Turbo ..... I: 5-3

Part II

1 Introduction to Part II ..... II: 1-1

2 Package and Pins ..... II: 2-1

3 Clocks and Power Interface ..... II: 3-1

3.1 Overview ..... II: 3-1
3.2 Signals ..... II: 3-1
3.2.1 Clock Interface Signals ..... II: 3-1
3.2.2 Power Manager Interface Control Signals ..... II: 3-2
3.2.3 Power Enable (PWR_EN) ..... II: 3-3
3.2.3.1 System Power Enable (SYS_EN) ..... II: 3-3
3.2.3.2 Power Manager I2C Clock (PWR_SCL) ..... II: 3-3
3.2.3.3 Power Manager I2C Data (PWR_SDA) ..... II: 3-3
3.2.3.4 nVDD_FAULT ..... II: 3-3
3.2.3.5 nBATT_FAULT ..... II: 3-4
3.3 Block Diagram ..... II: 3-4
3.4 Layout Notes ..... II: 3-6
3.5 Modes of Operations ..... II: 3-6
3.5.1 Clock Interface ..... II: 3-6
3.5.1.1 Using the On-Chip Oscillator with a 32.768-KHz Crystal ..... II: 3-7
3.5.1.2 Using an External 32.768-KHz Clock ..... II: 3-7
3.5.1.3 Using the On-Chip Oscillator with a 13.000-MHz Crystal ..... II: 3-7
3.5.1.4 Using an External 13.00-MHz Clock ..... II: 3-8
3.5.2 Power Interface ..... II: 3-8
3.5.2.1 Power Supplies ..... II: 3-8

4 Internal SRAM ..... II: 4-1

4.1 Overview ..... II: 4-1
4.2 Signals ..... II: 4-1
4.3 Block Diagram ..... II: 4-1
4.4 Layout Notes ..... II: 4-2

5 DMA Controller Interface ..... II: 5-1

5.1 Overview ..... II: 5-1
5.2 Signals ..... II: 5-1
5.3 Block Diagram ..... II: 5-2
5.4 Layout Notes ..... II: 5-2
5.5 Modes of Operation ..... II: 5-3
5.5.1 Fly-By DMA Transfers ..... II: 5-3
5.5.1.1 Signals ..... II: 5-3
5.5.1.2 Block Diagram ..... II: 5-4
5.5.1.3 Layout Notes ..... II: 5-4
5.5.2 Flow-Through DMA Transfers ..... II: 5-5
5.5.2.1 Signals ..... II: 5-5
5.5.2.2 Block Diagram ..... II: 5-5
5.5.2.3 Layout Notes ..... II: 5-6

6 System Memory Interface ..... II: 6-1

6.1 Overview ..... II: 6-1
6.2 Signals ..... II: 6-3

6.3 Block Diagram ..... II: 6-5
6.4 Memory Controller Layout Notes ..... II: 6-6
6.4.1 Memory Controller Routing Guidelines for 0.5mm and 0.65 mm Ball Pitch ..... II: 6-6
6.4.1.1 System Bus Recommended Signal Routing Guidelines (Excluding SDCLK<x> and SDCAS) ..... II: 6-6
6.4.1.2 SDCLK and SDCAS Recommended Signal Routing Guidelines ..... II: 6-7
6.4.1.3 Minimum Board Stack-up Configuration used for Signal Integrity ..... II: 6-8
6.5 Modes of Operation Overview ..... II: 6-9
6.5.1 SDRAM Interface ..... II: 6-9
6.5.1.1 SDRAM Signals ..... II: 6-9
6.5.1.2 SDRAM Memory Block Diagram ..... II: 6-11
6.5.1.3 SDRAM Layout Notes ..... II: 6-12
6.5.2 Flash Memory Interface (Asynchronous/Synchronous) ..... II: 6-15
6.5.2.1 Flash Memory Signals ..... II: 6-15
6.5.2.2 Flash Block Diagram ..... II: 6-16
6.5.2.3 Flash Layout Note ..... II: 6-16
6.5.3 ROM Interface ..... II: 6-17
6.5.3.1 ROM Signals ..... II: 6-17
6.5.3.2 ROM Block Diagram ..... II: 6-18
6.5.3.3 ROM Layout Notes ..... II: 6-18
6.5.4 SRAM Interface ..... II: 6-18
6.5.4.1 SRAM Signals ..... II: 6-19
6.5.4.2 SRAM Block Diagram ..... II: 6-20
6.5.4.3 SRAM Layout Notes ..... II: 6-20
6.5.5 Variable Latency Input/Output (VLIO) Interface ..... II: 6-21
6.5.5.1 VLIO Memory Signals ..... II: 6-22
6.5.5.2 VLIO Block Diagram ..... II: 6-23
6.5.5.3 VLIO Memory Layout Notes ..... II: 6-23
6.5.6 PC Card (PCMCIA) Interface ..... II: 6-23
6.5.6.1 PC Card Signals ..... II: 6-25
6.5.6.2 PC-Card Block Diagrams ..... II: 6-26
6.5.6.3 PC Card Layout Notes ..... II: 6-29
6.5.7 Alternate Bus Master Interface ..... II: 6-29
6.5.7.1 Alternate Bus Master Signals ..... II: 6-31
6.5.7.2 Alternate Bus Master Block Diagram ..... II: 6-32
6.5.7.3 Alternate Bus Master Layout Notes ..... II: 6-32

7 LCD Interface ..... II: 7-1

7.1 Overview ..... II: 7-1
7.2 Signals ..... II: 7-2
7.3 Schematics/Block Diagram ..... II: 7-3
7.4 Layout Notes ..... II: 7-3
7.4.1 Contrast Voltage ..... II: 7-3
7.4.2 Backlight Inverter ..... II: 7-4
7.4.3 Signal Routing and Buffering ..... II: 7-4
7.4.4 Panel Connector ..... II: 7-5
7.5 Modes of Operation Overview ..... II: 7-6
7.5.1 Passive Monochrome Single-Scan Mode ..... II: 7-6
7.5.1.1 Signals ..... II: 7-6
7.5.1.2 Schematics/Block Diagram ..... II: 7-7
7.5.1.3 Layout Notes ..... II: 7-7
7.5.2 Passive Monochrome Single-Scan Double-Pixel Mode ..... II: 7-8

7.5.2.1 Signals ..... II: 7-8
7.5.2.2 Schematics / Block Diagram ..... II: 7-9
7.5.2.3 Layout Notes ..... II: 7-9
7.5.3 Passive Monochrome Dual-Scan Mode ..... II: 7-9
7.5.3.1 Signals ..... II: 7-9
7.5.3.2 Schematics / Block Diagram ..... II: 7-10
7.5.3.3 Layout Notes ..... II: 7-11
7.5.4 Passive Color Single-Scan Mode ..... II: 7-11
7.5.4.1 Signals ..... II: 7-11
7.5.4.2 Schematics/Block Diagram ..... II: 7-12
7.5.4.3 Layout Notes ..... II: 7-12
7.5.5 Passive Color Dual-Scan Mode ..... II: 7-12
7.5.5.1 Signals ..... II: 7-12
7.5.5.2 Schematics / Block Diagram ..... II: 7-14
7.5.5.3 Layout Notes ..... II: 7-14
7.5.6 Active Color 12-bit per pixel Mode ..... II: 7-14
7.5.6.1 Signals ..... II: 7-14
7.5.6.2 Schematics / Block Diagram ..... II: 7-15
7.5.6.3 Layout Notes ..... II: 7-16
7.5.7 Active Color, 16-bit per pixel Mode ..... II: 7-16
7.5.7.1 Signals ..... II: 7-17
7.5.7.2 Schematics / Block Diagram ..... II: 7-17
7.5.7.3 Layout Notes ..... II: 7-18
7.5.8 Active Color, 18-bit per pixel Mode ..... II: 7-18
7.5.8.1 Signals ..... II: 7-19
7.5.8.2 Schematics / Block Diagram ..... II: 7-19
7.5.8.3 Layout Notes ..... II: 7-20
7.5.9 Smart Panel ..... II: 7-20
7.5.9.1 Signals ..... II: 7-21
7.5.9.2 Schematics / Block Diagram ..... II: 7-21
7.5.9.3 Layout Notes ..... II: 7-22

8 SSP Port Interface ..... II: 8-1

8.1 Overview ..... II: 8-1
8.2 Signals ..... II: 8-2
8.3 Block Diagram ..... II: 8-3
8.3.1 Standard SSP Configuration Scheme ..... II: 8-3
8.3.2 External Clock Source Configuration Scheme ..... II: 8-4
8.3.3 External Clock Enable Configuration Scheme ..... II: 8-5
8.3.4 Internal (to PXA27x Processor) Clock Enable Configuration Design ..... II: 8-5
8.4 Layout Notes ..... II: 8-6

9 Inter-Integrated Circuit (I2C) ..... II: 9-1

9.1 Overview ..... II: 9-1
9.2 Signals ..... II: 9-1
9.3 Schematic/Block Diagram ..... II: 9-2
9.3.1 Digital-to-Analog Converter (DAC) ..... II: 9-2
9.3.2 Other Uses of I2C ..... II: 9-3
9.3.3 Pull-Ups and Pull-Downs ..... II: 9-4
9.4 Layout Notes ..... II: 9-4

10 UART Interfaces ..... II: 10-1

10.1 Overview ..... II: 10-1
10.2 Signals ..... II: 10-2
10.3 Types of UARTs ..... II: 10-3
10.3.1 Full Function UART ..... II: 10-3
10.3.1.1 Full Function UART Signals ..... II: 10-3
10.3.1.2 FFUART Block Diagram ..... II: 10-4
10.3.1.3 FFUART Layout Notes ..... II: 10-4
10.3.2 Bluetooth UART ..... II: 10-5
10.3.2.1 Bluetooth UART Signals ..... II: 10-5
10.3.2.2 Bluetooth UART Block Diagram ..... II: 10-5
10.3.3 Standard UART ..... II: 10-6
10.3.3.1 Standard UART Signals ..... II: 10-6
10.3.3.2 Standard UART Block Diagram ..... II: 10-6

11 Fast Infrared Interface ..... II: 11-1

11.1 Overview ..... II: 11-1
11.2 Signals ..... II: 11-1
11.3 Block Diagram ..... II: 11-2

12 ‘USB Client Controller ..... II: 12-1

12.1 Overview ..... II: 12-1
12.2 Signals ..... II: 12-1
12.3 Block Diagram ..... II: 12-2
12.4 Layout Notes ..... II: 12-2
12.4.1 Self-Powered Devices ..... II: 12-2
12.4.1.1 Operation if GPIOn and GPIOx are Different Pins ..... II: 12-3
12.4.1.2 Operation if GPIOn and GPIOx are the Same Pin ..... II: 12-4
12.4.2 Bus-Powered Device ..... II: 12-5
12.4.3 USB On-The-GO Transceiver Usage ..... II: 12-6
12.4.4 Interface to External Transceiver (OTG) ..... II: 12-8
12.4.5 Interface to External Charge Pump Device (OTG) ..... II: 12-9
12.4.6 OTG ID ..... II: 12-11
12.4.7 Interface to External USB Transceiver (non-OTG) ..... II: 12-12

13 AC ’97 ..... II: 13-1

13.1 Overview ..... II: 13-1
13.2 Signals ..... II: 13-1
13.3 Block Diagram ..... II: 13-2
13.4 Layout Notes ..... II: 13-3

14 I2S Interface ..... II: 14-1

14.1 Overview ..... II: 14-1
14.2 Signals ..... II: 14-2
14.3 Block Diagram ..... II: 14-3
14.4 Layout Notes ..... II: 14-3
14.5 Modes of Operation Overview ..... II: 14-4
14.5.1 PXA27x Processor Provides BITCLK signal to CODEC ..... II: 14-4
14.5.1.1 Signals ..... II: 14-4
14.5.1.2 Block Diagram ..... II: 14-5
14.5.2 CODEC Provides BITCLK Signal to PXA27x Processor ..... II: 14-6

14.5.2.1 Signals ..... II: 14-6
14.5.2.2 Block Diagram ..... II: 14-6

15 MultiMediaCard/SD/SDIO Card Controller ..... II: 15-1

15.1 Overview ..... II: 15-1
15.2 Signals ..... II: 15-2
15.3 Layout Notes ..... II: 15-2
15.4 Modes of Operation Overview ..... II: 15-4
15.4.1 MMC/SD/SDIO Mode Using MMC Protocol ..... II: 15-4
15.4.1.1 MMC Protocol Signals ..... II: 15-4
15.4.1.2 MMC Protocol Block and Schematic Diagrams ..... II: 15-5
15.4.1.3 MMC Protocol Layout Notes ..... II: 15-6
15.4.2 MMC/SD/SDIO Mode Using SD or SDIO Protocols ..... II: 15-7
15.4.2.1 SD and SDIO Protocol Signals ..... II: 15-7
15.4.2.2 SD and SDIO Protocol Block and Schematic Diagrams ..... II: 15-8
15.4.2.3 SD and SDIO Protocol Layout Notes ..... II: 15-10
15.4.3 SPI Mode with MMC, SD Card, and SDIO Card Devices ..... II: 15-11
15.4.3.1 SPI Mode Signals ..... II: 15-11
15.4.3.2 SPI Protocol Block and Schematic Diagrams ..... II: 15-12
15.4.3.3 SPI Protocol Layout Notes ..... II: 15-12

16 Baseband Interface ..... II: 16-1

16.1 Overview ..... II: 16-1

17 Memory Stick Host Interface ..... II: 17-1

17.1 Overview ..... II: 17-1
17.2 Signals ..... II: 17-1
17.3 Schematic/Block Diagram ..... II: 17-1
17.4 Layout Notes ..... II: 17-2

18 Keypad Interface ..... II: 18-1

18.1 Overview ..... II: 18-1
18.2 Signals ..... II: 18-2
18.3 Block Diagram ..... II: 18-3
18.4 Layout Notes ..... II: 18-4
18.4.1 Recommended Pull-down Resistors ..... II: 18-4
18.4.2 Alternate Function During Standby and Sleep Mode ..... II: 18-4
18.4.3 Reduce Power During Standby and Sleep Mode ..... II: 18-4
18.4.4 Using the Keypad Signals to Wake-up from Standby and Sleep Mode ..... II: 18-4
18.4.5 How to Enable Specific Combinations of Direct Keys ..... II: 18-4
18.4.6 Interfacing to a Matrix Keypad ..... II: 18-5
18.5 Modes of Operation Overview ..... II: 18-6
18.5.1 Keypad Matrix and Direct Keys and No Rotary Encoder ..... II: 18-6
18.5.1.1 Signals ..... II: 18-6
18.5.1.2 Block Diagram ..... II: 18-7
18.5.2 Keypad Matrix and Direct Keys with One Rotary Encoder ..... II: 18-8
18.5.2.1 Signals ..... II: 18-8
18.5.2.2 Block Diagram ..... II: 18-9
18.5.3 Keypad Matrix and Direct Keys with Two Rotary Encoders ..... II: 18-10
18.5.3.1 Signals ..... II: 18-10
18.5.3.2 Block Diagram ..... II: 18-11

19 USIM Controller Interface ..... II: 19-1

19.1 Overview ..... II: 19-1
19.2 Signals ..... II: 19-2
19.2.1 PXA27x Processor USIM Interface Signals ..... II: 19-2
19.2.2 USIM Card Interface Signals ..... II: 19-3
19.3 Block Diagram ..... II: 19-4
19.4 Layout Notes ..... II: 19-5

20 Universal Serial Bus Host Interface ..... II: 20-1

20.1 Overview ..... II: 20-1
20.2 Signals ..... II: 20-1
20.3 Block Diagrams ..... II: 20-2
20.3.1 Block Diagram for USB Host Differential Connection (Port 1 or Port 2) ..... II: 20-2
20.3.2 Block Diagrams for USB Host Port 2 (Differential or Single-Ended) ..... II: 20-3
20.3.3 Block Diagram for USB Host Single-Ended Connection (Port 3) ..... II: 20-4
20.4 Layout Notes ..... II: 20-5

21 Real Time Clock Interface ..... II: 21-1

21.1 Overview ..... II: 21-1
21.2 Signals ..... II: 21-1
21.3 Block Diagram ..... II: 21-2
21.4 Layout Notes ..... II: 21-2

22 OS Timer Interface ..... II: 22-1

22.1 Overview ..... II: 22-1
22.2 Signals ..... II: 22-1
22.3 Block Diagram ..... II: 22-2
22.3.1 Channel Access/Control Block ..... II: 22-2
22.3.2 PXA25x Compatibility Channels 0-3 Block ..... II: 22-2
22.3.3 Channels 4 - 11 Blocks ..... II: 22-3
22.3.4 Output Control ..... II: 22-3
22.4 Layout Notes ..... II: 22-3

23 Pulse-Width Modulator Interface ..... II: 23-1

23.1 Overview ..... II: 23-1
23.2 Signals ..... II: 23-1
23.3 Block Diagram ..... II: 23-2
23.4 Layout Notes ..... II: 23-2

24 General Purpose Input/Output Interfaces ..... II: 24-1

24.1 Overview ..... II: 24-1
24.2 Signals ..... II: 24-1
24.3 Block Diagram/Schematic ..... II: 24-3
24.4 Layout Notes ..... II: 24-3

25 Interrupt Interface ..... II: 25-1

25.1 Overview ..... II: 25-1
25.2 Signals ..... II: 25-2
25.3 Block Diagram ..... II: 25-3
25.4 Layout Notes ..... II: 25-4

26 JTAG Debug ..... II: 26-1

26.1 Overview ..... II: 26-1
26.2 Features ..... II: 26-2
26.3 Signal Descriptions ..... II: 26-3
26.4 Operation ..... II: 26-3
26.4.1 TAP Controller Reset ..... II: 26-3
26.4.2 Pull-Up Resistors ..... II: 26-4
26.4.3 JTAG Instruction Register and Instruction Set ..... II: 26-5
26.4.4 Test Data Registers ..... II: 26-7
26.4.4.1 Bypass Register ..... II: 26-7
26.4.4.2 Boundary-Scan Register ..... II: 26-8
26.4.4.3 Data-Specific Registers ..... II: 26-9
26.4.4.4 Flash Data Register ..... II: 26-9
26.4.4.5 Intel XScale® Data Registers ..... II: 26-9
26.4.5 Test Access Port (TAP) Controller ..... II: 26-10
26.4.5.1 Test-Logic-Reset State ..... II: 26-11
26.4.5.2 Run-Test/Idle State ..... II: 26-11
26.4.5.3 Select-DR-Scan State ..... II: 26-11
26.4.5.4 Capture-DR State ..... II: 26-11
26.4.5.5 Shift-DR State ..... II: 26-11
26.4.5.6 Exit1-DR State ..... II: 26-12
26.4.5.7 Pause-DR State ..... II: 26-12
26.4.5.8 Exit2-DR State ..... II: 26-12
26.4.5.9 Update-DR State ..... II: 26-12
26.4.5.10 Select-IR-Scan State ..... II: 26-13
26.4.5.11 Capture-IR State ..... II: 26-13
26.4.5.12 Shift-IR State ..... II: 26-13
26.4.5.13 Exit1-IR State ..... II: 26-13
26.4.5.14 Pause-IR State ..... II: 26-13
26.4.5.15 Exit2-IR State ..... II: 26-14
26.4.5.16 Update-IR State ..... II: 26-14
26.5 Register Descriptions ..... II: 26-15
26.5.1 JTAG Device Identification (ID) Register ..... II: 26-15
26.5.2 JTAG Test Data Registers ..... II: 26-16
26.5.3 Debug Registers ..... II: 26-16
26.6 Test Register Summary ..... II: 26-16

27 Intel® Quick Capture Technology ..... II: 27-1

27.1 Overview ..... II: 27-1
27.2 Feature List ..... II: 27-2
27.3 Signals ..... II: 27-2
27.4 Block Diagram ..... II: 27-3

Appendix

A **PXA27x DVK Block Diagram** ..... II: A-1

B **PXA27x Processor Developer's Kit (DVK)** ..... II: B-1

C **PXA27x DVK Bill-of-Materials** ..... II: C-1

D **Intel® PXA27x Processor and Intel® PXA25x Processor Differences** ..... II: D-1

E **Companion Components for PXA27x Processor** ..... II: E-1

Glossary Glossary-1

Index IX-1

Figures

Part I

1-1 PXA27x Processor Block Diagram ..... I: 1-2
2-1 1+6+1 uvia PCB Stackup ..... I: 2-2
2-2 Recommended PCB Layer Assignment for an Eight-Layer PCB ..... I: 2-3
2-3 Recommended I/O Power Plane Layout ..... I: 2-4
2-4 VF-BGA 13mm x 13mm Component Layout Placement Guide (Top View) ..... I: 2-5
2-5 FS-CSP 14mm x 14mm Component Layout Placement Guide (Top View) ..... I: 2-6
2-6 PBGA 23mm x 23mm Component Layout Placement Guide (Top View) ..... I: 2-7
2-7 PCB Escape Routing for Copper-Defined Land Pads ..... I: 2-8
2-8 PCB Escape Routing for Copper Defined Land Pads ..... I: 2-9
2-9 Recommended Mobile Handset Dimensions Diagram ..... I: 2-11
2-10 FS-CSP (14x14) Tray Specification ..... I: 2-13
2-11 PBGA (23x23) Tray Specification ..... I: 2-14
4-1 Minimal Voltage Regulator Power System Design Example ..... I: 4-3

Part II

3-1 Typical Battery and External Regulator Configuration ..... II: 3-5
4-1 Internal SRAM Block Diagram ..... II: 4-2
5-1 DMA controller Block Diagram ..... II: 5-2
5-2 Companion Chip Using Fly-by DMA Transfer Interface ..... II: 5-4
5-3 Companion Chip Requesting Flow-Through DMA Transfers ..... II: 5-5
6-1 General Memory Interface Configuration ..... II: 6-5
6-2 PXA27x Processor Memory System Bus Routing Topologies (ExcludingSDCLK<x> and SDCAS) ..... II: 6-7
6-3 PXA27x Processor Memory Clock and SDCAS Routing Topology ..... II: 6-8
6-4 Minimum Board Stack-up Configuration used for Signal Integrity ..... II: 6-8
6-5 SDRAM Memory System Example ..... II: 6-11
6-6 Block Diagram Connecting Synchronous Flash to nCS<1:0> ..... II: 6-16
6-7 Block Diagram Connecting ROM to nCS<0> ..... II: 6-18
6-8 Block Diagram Connecting SRAM to nCS<2> ..... II: 6-20
6-9 Variable Latency Interface Block Diagram ..... II: 6-23
6-10 External Logic for a One-Socket Configuration Expansion PC Card ..... II: 6-27
6-11 External Logic for a Two-Socket Configuration Expansion PC Card ..... II: 6-28
6-12 Alternate Bus Master Mode ..... II: 6-32
7-1 Passive Monochrome Single-Scan Display Typical Connection ..... II: 7-7
7-2 Passive Monochrome Single-Scan Double-Pixel Data Display Typical Connection ..... II: 7-9
7-3 Passive Monochrome Dual-Scan Display Typical Connection ..... II: 7-10
7-4 Passive Color Single-Scan Display Typical Connection ..... II: 7-12
7-5 Passive Color Dual-Scan Display Typical Connection ..... II: 7-14
7-6 Active Color 12-bit per pixel Display Typical Connection ..... II: 7-16
7-7 Active Color 16-bit-per-pixel Display Typical Connection ..... II: 7-18
7-8 Active Color 18-bit-per pixel Display Typical Connection ..... II: 7-20
7-9 Active Color Display 24-bit Typical Connection ..... II: 7-22

8-1 Standard SSP Configuration Scheme Block Diagram ..... II: 8-4
8-2 External Clock Source Configuration Scheme Block Diagram ..... II: 8-4
8-3 External Clock Enable Configuration Scheme Block Diagram ..... II: 8-5
8-4 Internal Clock Enable Configuration Scheme Block Diagram ..... II: 8-6
9-1 Linear Technology DAC with I2C Interface ..... II: 9-2
9-2 Using an Analog Switch to Allow a Second CF Card ..... II: 9-3
9-3 I^2C Pull-Ups and Pull-Downs ..... II: 9-4
10-1 FFUART Interface Block Diagram ..... II: 10-4
10-2 BTUART Interface Block Diagram ..... II: 10-5
10-3 STUART Interface Block Diagram ..... II: 10-6
11-1 Fast Infrared Controller Port Interface Block Diagram ..... II: 11-2
12-1 USB Client Interface Block Diagram ..... II: 12-2
12-2 Self Powered Device when GPIOn and GPIOx are Different Pins ..... II: 12-3
12-3 Self-Powered Device when GPIOn and GPIOx are Same Pins ..... II: 12-4
12-4 USB OTG Configurations ..... II: 12-6
12-5 Host Port 2 OTG Transceiver ..... II: 12-7
12-6 Connection to External OTG Transceiver ..... II: 12-8
12-7 Connection to External OTG Charge Pump ..... II: 12-10
12-8 Connection to OTG ID ..... II: 12-11
12-9 PXA27x Processor Connection to External USB Transceiver ..... II: 12-12
13-1 AC ‘97 Controller to CODEC Block Diagram ..... II: 13-2
14-1 I2S Controller Interface Block Diagram ..... II: 14-3
14-2 PXA27x Processor Provides BITCLK ..... II: 14-5
14-3 PXA27x Processor Receives BITCLK ..... II: 14-6
15-1 MMC Protocol Interface Block Diagram ..... II: 15-5
15-2 MMC Protocol Interface Schematic Diagram ..... II: 15-6
15-3 SD and SDIO Protocol Interface Block Diagram ..... II: 15-8
15-4 SD and SDIO Protocol Interface Schematic Diagram ..... II: 15-9
15-5 SPI Protocol Interface Block Diagram ..... II: 15-12
17-1 Memory Stick Implementation Block Diagram ..... II: 17-2
18-1 Keypad Interface Block Diagram ..... II: 18-3
18-2 Keypad Matrix and Direct Keys Block Diagram (with No Rotary Encoder) ..... II: 18-7
18-3 Keypad Matrix and Direct Keys Block Diagram (with One Rotary Encoder) ..... II: 18-9
18-4 Keypad Matrix and Direct Keys Block Diagram (with Two Rotary Encoders) ..... II: 18-11
19-1 Connectivity USIM Card and PXA27x Processor USIM Interface using UVSx signals ..... II: 19-4
19-2 Connectivity USIM Card and PXA27x Processor USIM Interface Using nUEN ..... II: 19-5
20-1 USB Host (Port 1 or Port 2) Differential Connections Block Diagram ..... II: 20-2
20-2 PXA27x Processor Host 2 Single-Ended Connection to External Transceiver ..... II: 20-3
20-3 PXA27x Processor Host 3 Connection to External USB Transceiver ..... II: 20-4
21-1 Example HZ_CLK Block Diagram ..... II: 21-2
22-1 OS Timer Block Diagram ..... II: 22-2
23-1 PWM Block Diagram For Applications Requiring a Filter ..... II: 23-2
25-1 Interrupt Controller Block Diagram ..... II: 25-3
26-1 Test Access Port (TAP) Block Diagram ..... II: 26-2
26-2 PXA27x Scan Chain Arrangement ..... II: 26-5
26-3 TAP Controller State Diagram ..... II: 26-10
27-1 Block Diagram for 8-bit Master Parallel Interface ..... II: 27-3
27-2 Interface Options Summary ..... II: 27-4

Appendix

A-1 System Overview Block Diagram ..... II: A-2
A-2 Main Board Block Diagram ..... II: A-3
A-3 Daughter Card Block Diagram ..... II: A-4
A-4 Liquid Crystal Display Block Diagram ..... II: A-5
A-5 Audio Module Block Diagram ..... II: A-6
A-6 Keyboard Block Diagram ..... II: A-7
A-7 JTAG Block Diagram ..... II: A-8

Tables

Part I

1-1 Related Documentation ..... I: 1-1
2-1 Recommended PCB Design Guidelines ..... I: 2-1
2-2 PCB Dimensions for Copper-Defined Land Pads ..... I: 2-8
2-3 PCB Dimensions for Copper Defined Land Pads ..... I: 2-9
2-4 PCB Dimensions for Copper-Defined Land Pads ..... I: 2-10
4-1 External Power Supply Descriptions ..... I: 4-2

Part II

3-1 Clock Interface Signals ..... II: 3-1
3-2 Power Controller Interface Signals ..... II: 3-2
5-1 DMA Interface Signals ..... II: 5-1
5-2 Fly-By DMA Transfer Signals ..... II: 5-3
5-3 Flow-Through DMA Transfer Signals ..... II: 5-5
6-1 Memory Address Map ..... II: 6-1
6-2 PXA27x Processor Memory Controller I/O Signals ..... II: 6-3
6-3 Minimum and Maximum Trace Lengths for the SDRAM Signals (Excluding SDCLK<x> and SDCAS) ..... II: 6-6
6-4 Minimum and Maximum Trace Lengths for the SDCLK<x> and SDCAS signals ..... II: 6-7
6-5 SDRAM I/O Signals ..... II: 6-9
6-6 SDRAM Memory Types Supported by PXA27x Processor ..... II: 6-12
6-7 Normal and Alternate Mode Memory Address Signal Mapping ..... II: 6-13
6-8 SA-1110 Address Compatibility Mode Memory Address Signal Mapping ..... II: 6-14
6-9 Flash Interface Signals ..... II: 6-15
6-10 ROM Interface Signals ..... II: 6-17
6-11 SRAM Interface Signals ..... II: 6-19
6-12 VLIO Memory Interface Signals ..... II: 6-22
6-13 PC Card Interface Signals ..... II: 6-25
6-14 Alternate Bus Master Interface Signals ..... II: 6-31
7-1 LCD Interface Signal List ..... II: 7-2
7-2 LCD Controller Data Pin Utilization ..... II: 7-3
7-3 Passive Display Pins Required ..... II: 7-6
7-4 Passive Display Pins Required ..... II: 7-8
7-5 Passive Display Pins Required ..... II: 7-9
7-6 Passive Display Pins Required ..... II: 7-11
7-7 Passive Display Pins Required ..... II: 7-13
7-8 LCD Interface Signal List ..... II: 7-15
7-9 LCD Interface Signal List ..... II: 7-17
7-10 LCD Interface Signal List ..... II: 7-19
7-11 Active Display Pins Required ..... II: 7-21
8-1 SSP Serial Port I/O Signals ..... II: 8-2
9-1 I2C Signal Description ..... II: 9-1
10-1 UART Signal Descriptions ..... II: 10-2
10-2 FFUART Interface Signals ..... II: 10-3

10-3 BTUART Interface Signals ..... II: 10-5
10-4 STUART Interface Signals ..... II: 10-6
11-1 FICP Signal Description ..... II: 11-1
12-1 USB Client Controller Interface Signals Summary ..... II: 12-1
12-2 Host Port 2 OTG Transceiver Switch Control Settings ..... II: 12-7
12-3 Output to External USB Transceiver ..... II: 12-13
12-4 Input from External USB Transceiver ..... II: 12-13
13-1 External Interface to CODECs ..... II: 13-1
14-1 I2S Controller Interface to CODEC ..... II: 14-2
14-2 I2S Controller Interface to CODEC (PXA27x Processor Providing BITCLK to CODEC) ..... II: 14-4
14-3 I2S Controller Interface to CODEC (CODEC Providing BITCLK to the PXA27x Processor) ..... II: 14-6
15-1 Multimedia Card/SD/SDIO Card Controller Interface Signal Summary ..... II: 15-2
15-2 MMC/SD/SDIO Controller Supported Sockets and Devices ..... II: 15-2
15-3 MMC/SD/SDIO Controller Supported Device Configurations ..... II: 15-3
15-4 Multimedia Card Protocol Interface Signals ..... II: 15-4
15-5 MMC Pull-up and Pull-down Resistors ..... II: 15-6
15-6 SD Card and SDIO Card Protocol Interface Signals ..... II: 15-7
15-7 SD/SDIO Card Pull-Up and Pull-Down Resistors ..... II: 15-10
15-8 SPI Protocol Interface Signals ..... II: 15-11
17-1 Memory Stick Host Interface Signal List ..... II: 17-1
18-1 Interface Signals Summary ..... II: 18-2
19-1 PXA27x Processor Interface Signals Summary ..... II: 19-2
19-2 USIM Card Signals ..... II: 19-3
20-1 USB Host Controller Interface Signals Summary ..... II: 20-1
21-1 RTC Interface Signal List ..... II: 21-1
22-1 OS Timer Interface Signals ..... II: 22-1
23-1 PWM Interface Signal List ..... II: 23-1
24-1 GPIO Interface Signal List ..... II: 24-1
25-1 GPIO Unit I/O Signal ..... II: 25-2
26-1 TAP Controller Pin Definitions ..... II: 26-3
26-2 IEEE 1149.1 Boundary-Scan Instruction Set ..... II: 26-6
26-3 IEEE 1149.1 Boundary-Scan Instruction Descriptions ..... II: 26-6
26-4 I/O Pins Excluded from Boundary-Scan Register ..... II: 26-8
26-5 JTAG Device Identification (ID) Register ..... II: 26-15
26-6 Test Register Summary ..... II: 26-16
27-1 Signal Descriptions for Quick Capture Technology ..... II: 27-2

Appendix

B-1 Processor Developer's Kit (formerly NBMMNS3BVS DVK) ..... II: B-1
B-2 Processor Developer's Kit (formerly NBMMNS2BVS DVK) ..... II: B-1
B-3 Processor Developer's Kit ..... II: B-2
D-1 PXA27x Processor Operating Modes not Supported by the PXA25x Processor ..... II: D-2
E-1 Crystal Devices ..... II: E-1
E-2 Manufacturers of PMIC Devices ..... II: E-2
E-3 USB OTG Transceivers ..... II: E-4

Revision History

Date	Revision	Description
May, 2005	002	Updated Chapter 2 “PCB Design Guidelines” on page I:2-1. Updated Section 6.5.5, “Variable Latency Input/Output (VLIO) Interface,” on page II: 6-21 Updated Connection to OTG ID, Figure 12-7, “Connection to External OTG Charge Pump” on page II:12-10.
April 2004	001	Initial release

§§

Introduction to Part I 1

This document outlines the design recommendations, board schematics, and debug recommendations for Intel® PXA27x Processor Family (PXA27x processor). The guidelines presented in this document provide maximum flexibility for board designers, while reducing the risk of board-related problems. Intel® PXA27x Processor Family consists three devices:

- Intel® PXA270 Processor – discrete processor
- Intel® PXA271 Processor – 32 Mbytes of Intel StrataFlash® Memory and 32 Mbytes of Low Power SDRAM
- Intel® PXA272 Processor – 64 Mbytes of Intel StrataFlash® Memory

The schematics in Appendix B , "PXA27x Processor Developer's Kit (DVK)" are provided as a reference. While these schematics describe one specific design, many aspects of the schematics remain the same for most PXA27x processor-based platforms.

Refer to the debug recommendations provided in Part II Section 26, "JTAG Debug," when debugging a processor-based system. To ensure the correct implementation of the debug port if included in your design, consult the debug recommendations before completing your board design. Refer to Part II Section 26, "JTAG Debug," for more information.

1.1 Document Organization and Overview

This document consists of two parts with multiple chapters in each part:

Part 1 Contains information that applies to the entire system design and provides guidelines for all designs. Read and thoroughly understand Part I before attempting a new design with the PXA27x processor.

Part 2 Contains specific design considerations for each PXA27x processor on-chip peripheral. All sections are not applicable to all designs, as all units of the processor are not utilized in every designs. For easy reference, sections in Part II of the Design Guide correspond to sections in the *Intel® PXA27x Processor Family Developers Manual*.

There are additional documents that provide guidance in the design and implementation of any new system. See Table 1-1for a list of related documents. Contact your Intel representative for the latest revision of these Intel documents.

Table 1-1. Related Documentation (Sheet 1 of 2)

Document Title	Order Number
Intel® PXA27x Processor Family Developers Manual	280000
Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification	280002
Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification	280003
Intel® PXA27x Processor Family Optimization Guide	280004

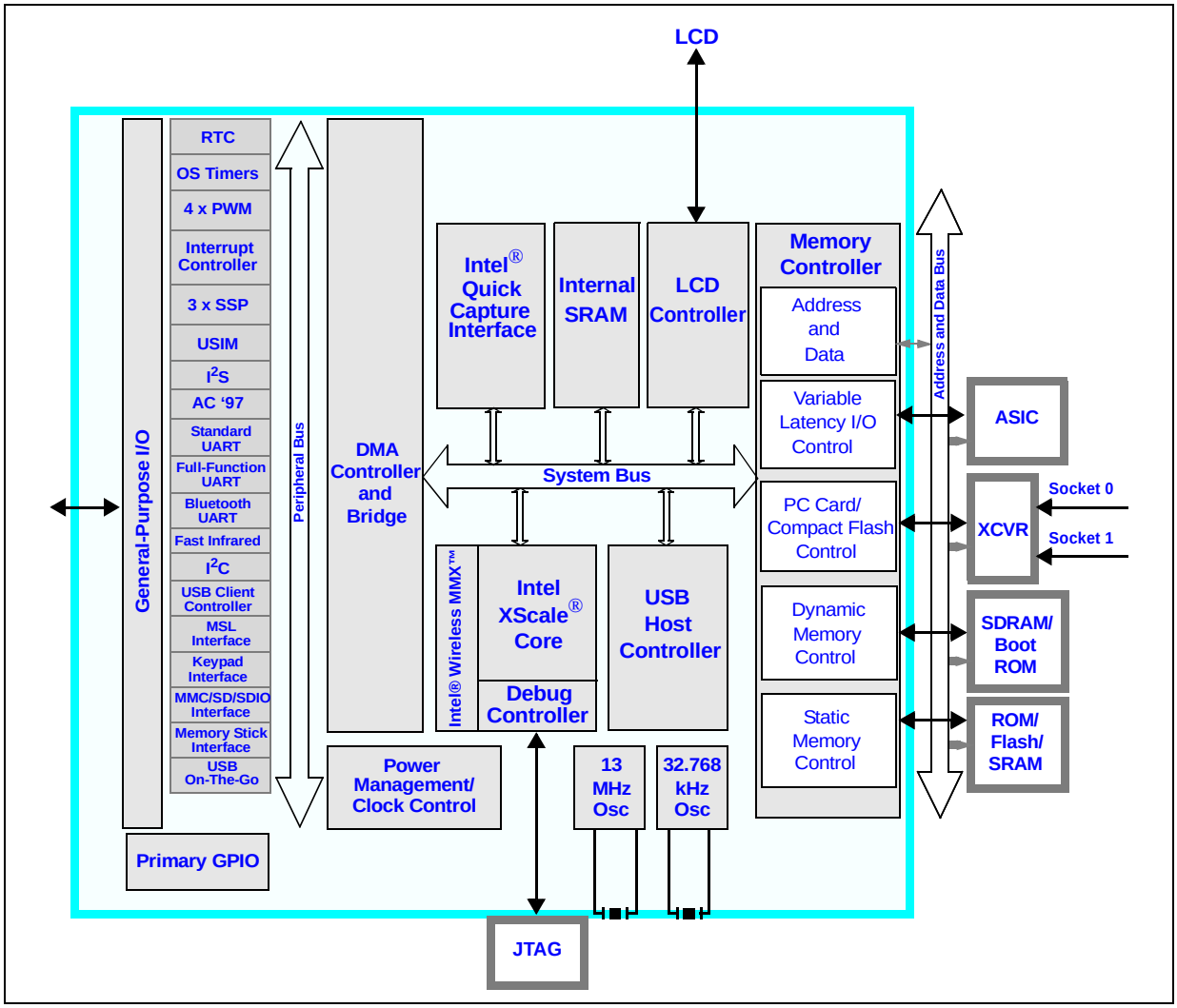
Table 1-1. Related Documentation (Sheet 2 of 2)

Document Title	Order Number
Intel® PXA27x Processor Family Power Requirements Application Note	280005
Intel® PXA27x Processor Family Design Check List Application Note	280013
Intel ® Flash Memory Design for a Stacked Chip Scale Package (SCSP) Application Note	252802-002

1.2 Functional Overview

The PXA27x processor offers an integrated system-on-a-chip design based on the Intel XScale® Microarchitecture. The PXA27x processor integrates the Intel XScale® Microarchitecture core with many on-chip peripherals that allows design of many different products for the handheld and cellular handset markets. Figure 1-1 is the block diagram for the PXA27x processor.

Figure 1-1. PXA27x Processor Block Diagram



The Intel® PXA270 Processor is available in a 13 mm x 13 mm (0.512 x 0.512 inches), 356-pin 0.50 mm (0.0197 inches) VF-BGA molded matrix array package with 32-bit functionality. The Intel® PXA271 Processor and Intel® PXA272 Processor are available in a 14 mm x 14 mm (0.551 x 0.551 inches), 336-pin 0.65 mm (0.0256 inches) FS-CSP with 32-bit functionality.

Refer to Part I Section 1.3, “Package Introduction,” for description of the supported features.

1.3 Package Introduction

The PXA27x processor features are:

- Maximum core frequencies of 624 MHz
- Variable core voltage from 0.85 V to 1.55 V
- System memory interface
  - — Up to 100-MHz SDRAM @ 1.8 V, 2.5 V, 3.0 V or 3.3 V
  - — Support for 16-, 64-, 128-, 256-, and 512-Mbit SDRAM technologies
  - — Four Banks of on-chip SRAM, each independently configurable and supporting 64 Kbytes of memory
  - — Clock enable (provided with 1 CKE pin to put the entire SDRAM interface into self refresh)
  - — Supports as many as six static memory devices (SRAM, flash, or VLIO)
- PCMCIA/Compact Flash card control pins
- LCD controller pins
- Full function UART pins
- Bluetooth* UART pins
- MMC controller pins
- SSP pins
- USB client pins
- USB host pins
- AC'97 controller pins
- Standard UART pins
- I^2C controller pins
- PWM pins
- Memory stick host controller pins
- Baseband interface pins
- Keypad interface pins
- Universal subscriber identity module interface pins
- Integrated JTAG support
- General-purpose I/O pins

1.4 Signal Pin Descriptions

Refer to Chapter 2, “System Architecture” of the *Intel® PXA27x Processor Family Developers Manual* for description of the signal descriptions for the PXA27x processor. Refer to this section for information regarding specific pin assignments and allocation.

§§

PCB Design Guidelines 2

This chapter provides printed-circuit board (PCB) design guidelines for the Intel® PXA27x Processor Family (PXA27x processor). The PXA27x processor family dimensions and package types are:

- PXA270 processor – 13 mm x 13 mm (0.512 x 0.512 inches) high density chip scale package (VF-BGA) package.
- PXA270 processor -- 23 mm x 23 mm (0.906 x 0.906 inches) plastic ball grid array package (PBGA).
- PXA271 processor, PXA272 processor, and PXA273 processor – 14 mm x 14 mm (0.551 x 0.551 inches) folded stacked chip scale package (FS-CSP).

The 0.5 mm (0.0197 inches) and 0.64 mm (0.0256 inches) ball pitch of the VF-BGA and FS-CSP packages provide the high density required in wireless handset applications and portable digital assistants. (PDAs). The 1.0 mm (0.0394 inches) ball pitch of the PBGA package allows for lower cost PCB technology in less space-constrained systems.

2.1 Intel® Flash Memory Design Guidelines

Skip to Section 2.2 if only designing the BF-VGA configurations.

Refer to the *Intel ® Flash Memory Design for a Stacked Chip Scale Package (SCSP) Application Note* for design guidelines with respect to the top package used within the FS-CSP package. The application note has information including a design checklist, recommended bypass capacitance, flash core voltage considerations, and additional information. See Table 1-1 for ordering information.

2.2 General PCB Characteristics

See Table 2-1 for the list of recommended PCB design characteristics using the PXA27x processor.

Table 2-1. Recommended PCB Design Guidelines (Sheet 1 of 2)

Feature	VF-BGA	PBGA
PCB Layers	6 to 8 layers (typical)	4 to 6 layers (typical)
PCB Thickness	0.7874 to 1.5748 mm (typical) / 0.0310 to 0.0620 inches (typical)	
Land Pad Size	See Section 2.2.3.1, Section 2.2.3.2, Section 2.2.3.3 for package specific specifications.	
Solder Mask Opening		
Typical Trace Width		
Reduced Trace Width between Land Pads		
Typical Micro-via Size		

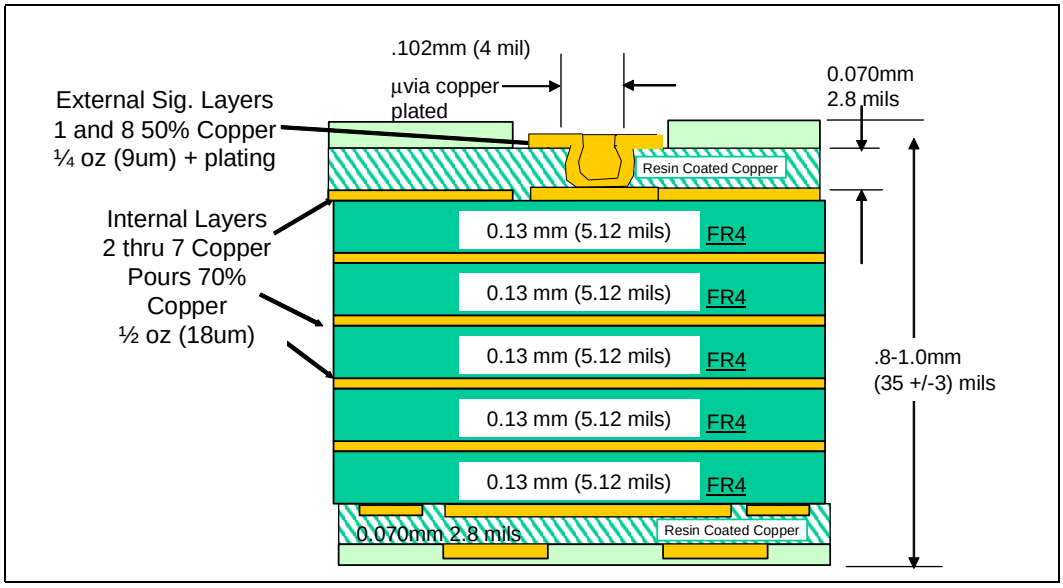
Table 2-1. Recommended PCB Design Guidelines (Sheet 2 of 2)

Feature	VF-BGA	PBGA
Top of Solder Stencil Aperture	0.2790 mm/0.0110 inches	0.330 mm/0.013 inches
Bottom of Solder Stencil Aperture	0.3000 mm/0.0120 inches	0.356 mm/0.014 inches
Solder Stencil Thickness	0.1270 mm/0.0050 inches	0.127 mm/.005 inches

2.2.1 PCB Layer Assignment (Stackup)

See Figure 2-1 for 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_CORE power plane and a divided I/O power plane for the memory and peripheral domains. See Figure 2-2 for recommended PCB layer assignment for an eight-layer PCB. See Figure 2-3 for illustration of the recommended layout of the divided I/O power plane.

Figure 2-1. 1+6+1 uvia PCB Stackup



For the uvia in pad, design the uvia 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 2-2. Recommended PCB Layer Assignment for an Eight-Layer PCB

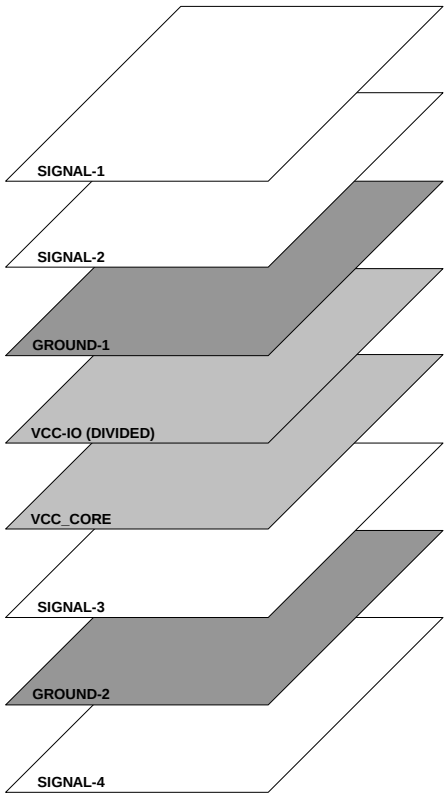
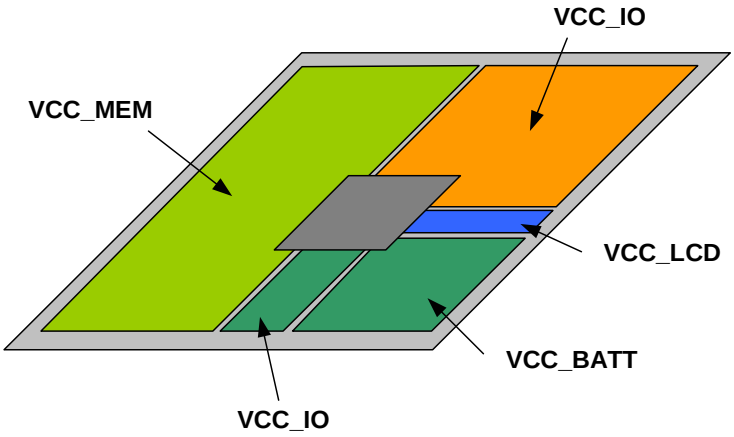


Figure 2-3. Recommended I/O Power Plane Layout



2.2.2 PCB Component Placement

PCB component placement requires careful planning to consider how signal and power traces map to the ten power supply domains on the PXA27x processor. See Figure 2-4, Figure 2-5, and Figure 2-6 for illustrations showing how the processor balls are grouped on the package for each supply domain. Place the external circuits as close as possible to the output pins of the PXA27x processor.

Recommendations for component placement for the PXA270 processor (VF-BGA, 13 mm x 13 mm) are:

- Place memory components on the left side.
- Place peripherals on the top or bottom side.
- Place the LCD panel in the middle of the right side.
- Place the USIM card interface on the upper right side.
- Place the crystals and power controller signals on the lower right side.

See Figure 2-4, Figure 2-5, and Figure 2-6 for illustrations showing how the VCC_CORE and VSS_CORE balls are present on all sides of the package. If the VSS references for all domains are connected to a single common ground plane, connections between all the VSS balls can be made without difficulty. However, use two power planes to facilitate connection of all VCC supply balls for each domain. When using two power planes, one continuous plane is assigned to the VCC_CORE power domain. The second plane is divided for memory and peripheral I/O power planes (see Figure 2-3).

Place the clock crystals and external load capacitors near the lower right corner of the package as close as possible to their package balls. If possible, install these components (clock crystals and external load capacitors) on the bottom of the board under the package to minimize trace capacitance and noise coupling.

In general, reserve the space on the bottom layer of the PCB under the package for the high-frequency decoupling caps and clock crystals. Install the high-frequency caps and clock crystals on the bottom layer before installing bulk decoupling caps or other components.

Figure 2-4. VF-BGA 13mm x 13mm Component Layout Placement Guide (Top View)

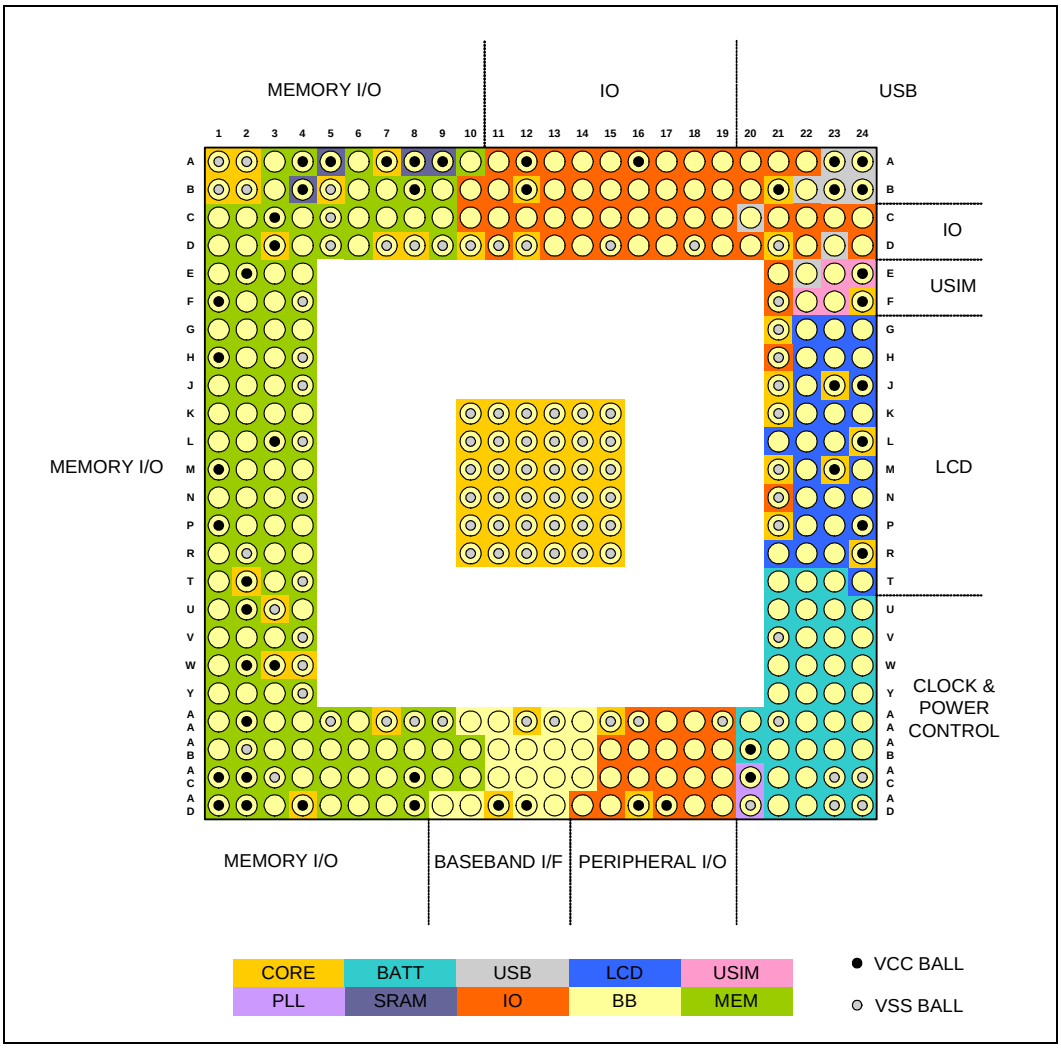


Figure 2-5. FS-CSP 14mm x 14mm Component Layout Placement Guide (Top View)

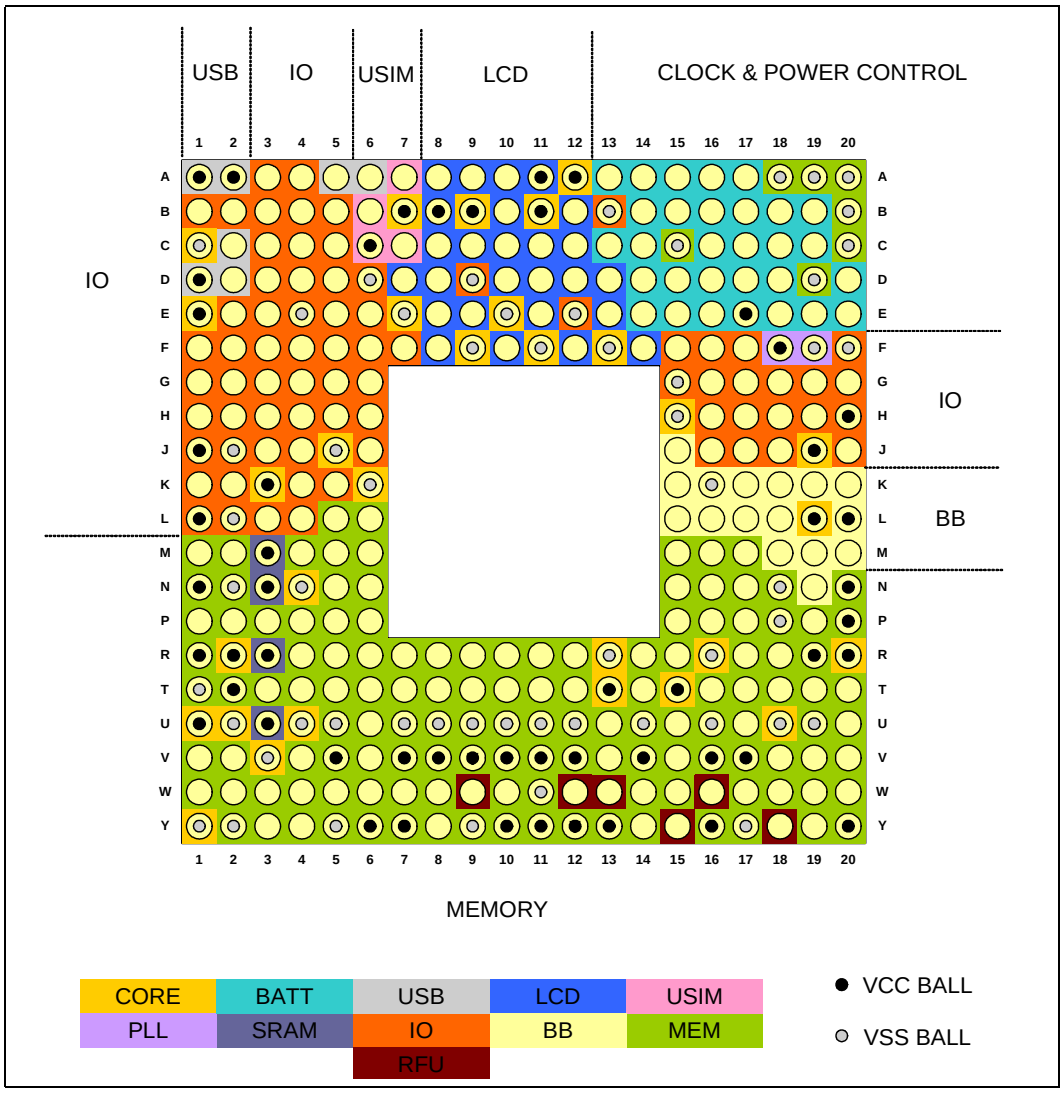
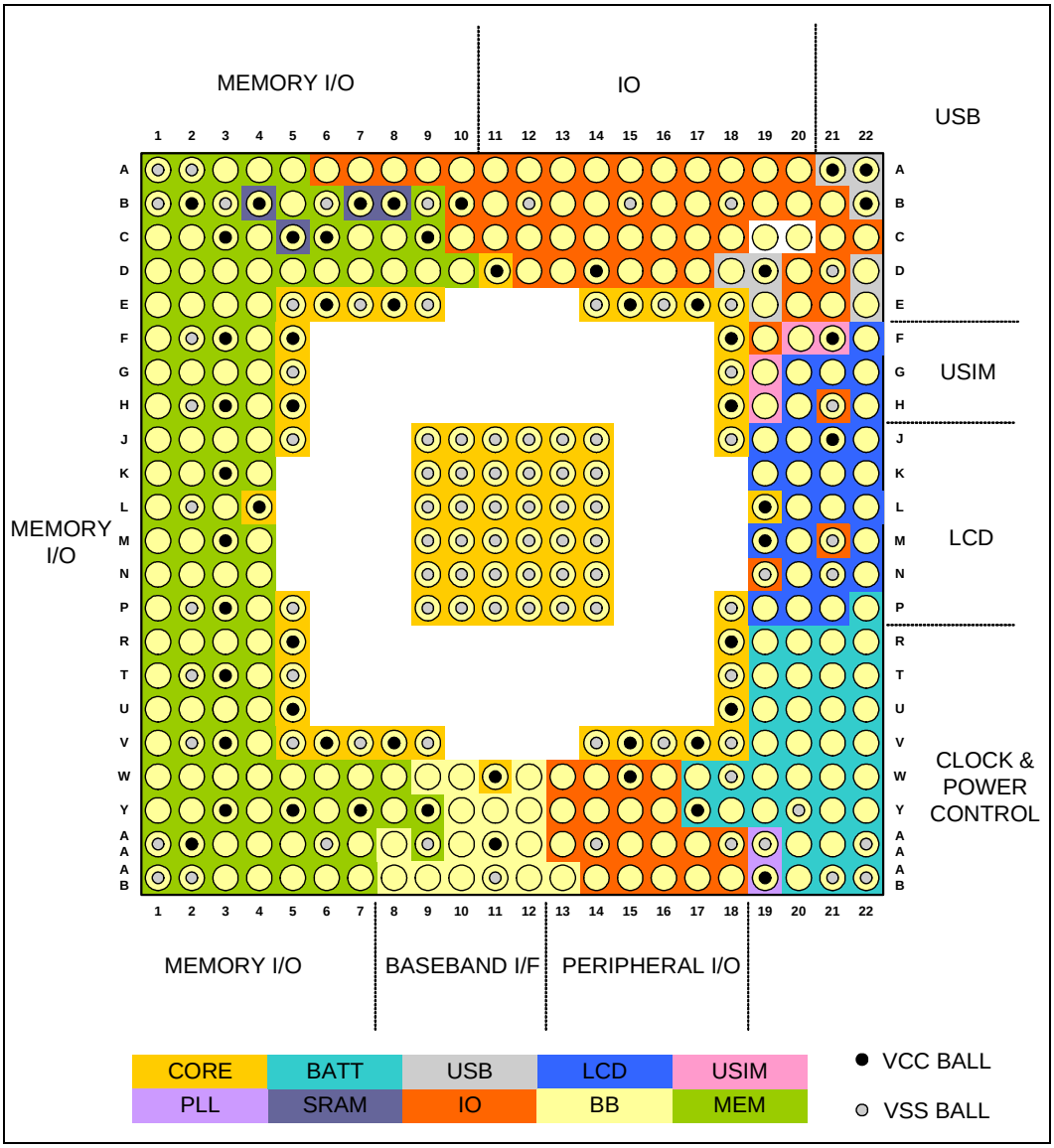


Figure 2-6. PBGA 23mm x 23mm Component Layout Placement Guide (Top View)



2.2.3 PCB Escape Routing

One important consideration when implementing chip scale packages (CSP) 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 micro-vias to route signals from the inner rows of balls on these packages:

- 0.5 mm (0.0197 inches) ball pitch packages (for example, VF-BGA)
- 0.65 mm (0.0256 inches) ball pitch packages (for example, FS-CSP)

2.2.3.1 VF-BGA Escape Routing

This section documents the method of VF-BGA routing along with the recommended dimensions.

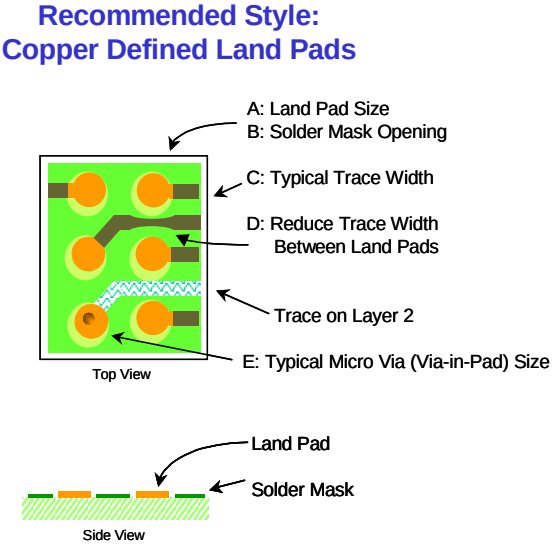
Table 2-2. PCB Dimensions for Copper-Defined Land Pads

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

NOTES:
1. All dimensions are in mm (inches).
2. Finished sizes accounts for copper etch.

On the two inner rows of the VF-BGA, route down the signals from the top layer to the inner PCB layers for routing away from the package. See Figure 2-7 for illustration of the PCB escape routing using the VF-BGA method.

Figure 2-7. PCB Escape Routing for Copper-Defined Land Pads



2.2.3.2 FS-CSP Escape Routing

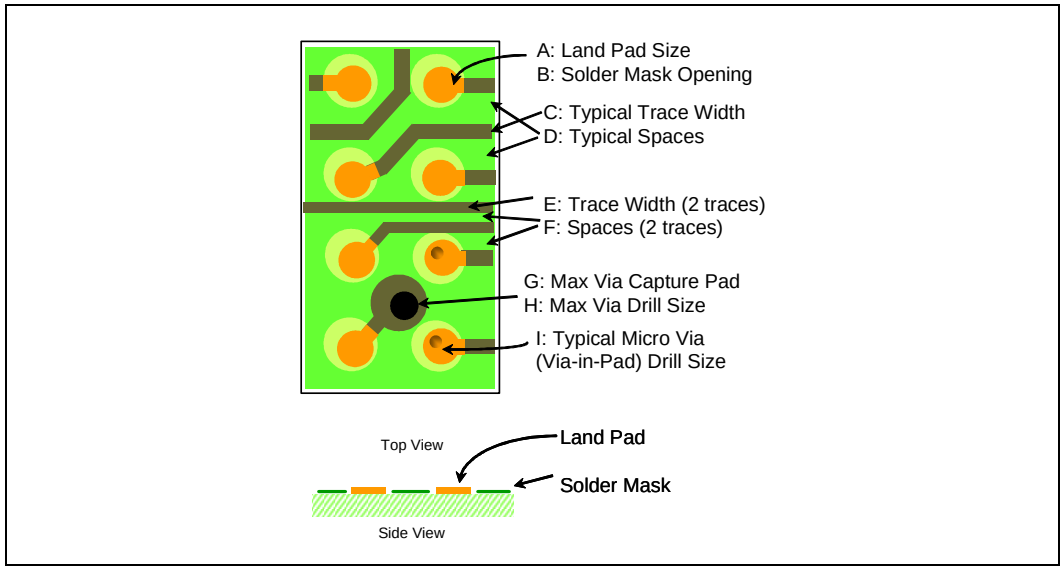
This section documents the method of FS-CSP routing along with the recommended dimensions.

Table 2-3. PCB Dimensions for Copper Defined Land Pads

Feature	.65 mm BGA
A: Land Pad Size	.30 (.012)
B: Solder Mask Opening	.432 (.017)
C: Typical Trace Width	.100 (.004)
D: Typical Spaces	.127 (.005)
E: Trace Width (2 Traces)	.070 (.00275)
F: Spaces (2 Traces)	.070 (.00275)
G: Max PTH Via Pad	.457 (.018)
H: Max PTH Via Drill Size	.25 (.008)
I: Typical Micro Via (Via-in-Pad) Drill Size	.127 (.005)
NOTE: All dimensions are in mm (inches).	

On the four inner rows of the FS-CSP, route down the signals from the top layer to the inner PCB layers for routing away from the package. See Figure 2-7 for illustration of the PCB escape routing using the FS-CSP method.

Figure 2-8. PCB Escape Routing for Copper Defined Land Pads



2.2.3.3 PBGA Escape Routing

This section documents the method of PBGA routing along with the recommended dimensions.

Table 2-4. PCB Dimensions for Copper-Defined Land Pads

Feature	1.0 mm PBGA
A: Land Pad Size	0.406 (0.016)
B: Solder Mask Opening	0.610 (0.024)
C: Typical Trace Width	0.127 (0.005)
D: Reduced Trace Width Between Land Pads	0.0889 (0.0035)
E: Trace Width (2 Traces)	0.090 (0.0035)
F: Spaces (2 Traces)	0.090 (0.0035)
G: Typical Micro Via (Via-in-Pad) Drill Size	N/A
Notes: 1. All dimensions are in mm (inches).	

2.2.4 PCB Keep-out Zones

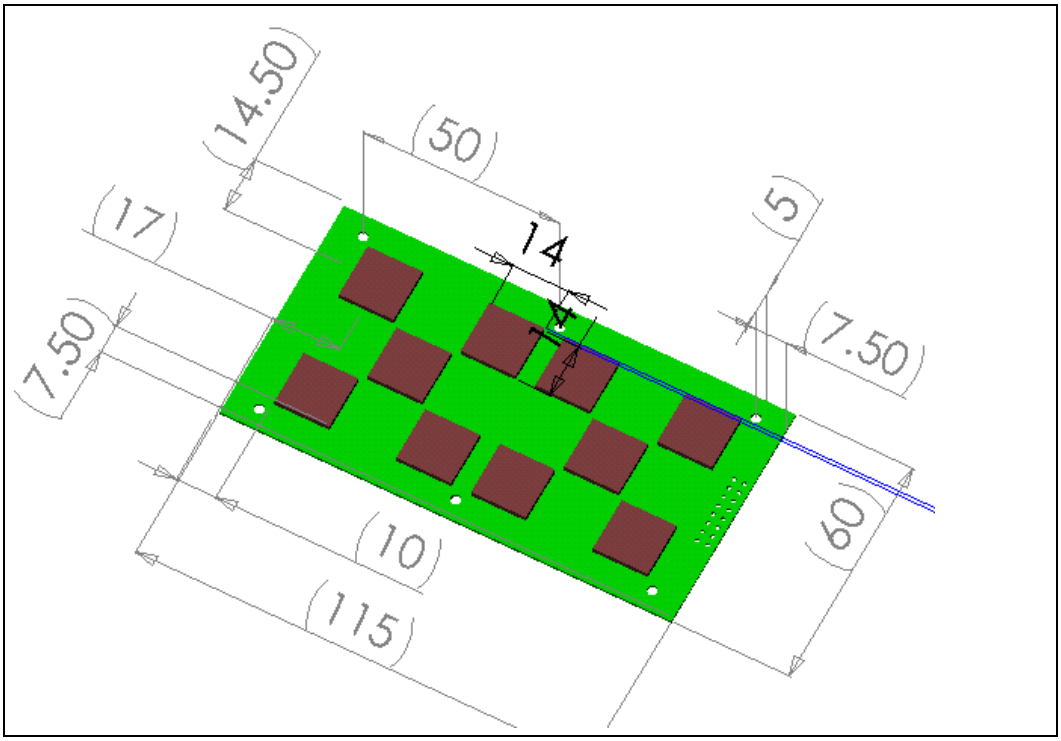
Another key PCB design element is the keep-out zone. The keep-out zone is the distance on each side of the CSP component to the nearest adjacent component on the board. This keep-out zone varies depending on the application and is generally much tighter in handheld applications that require many components 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 needs to 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. Pay special considerations for allowing adequate area for the hot air nozzle to surround the CSP being reworked. Currently, there is no rework procedure for the FS-CSP product.

Another factor that impacts the PCB keepout 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.

2.2.5 Recommended Mobile Handset Dimensions

For VF-BGA and FS-CSP packages with bodies >12xY mm and <=14x14 mm, the distance from board edge to center of package is 14.5 mm (package edge is always 2.5mm from support point). See Figure 2-9 for illustration of the PCB recommended dimensions.

Figure 2-9. Recommended Mobile Handset Dimensions Diagram



2.3 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. The processor also requires a 0.1 µF high-frequency decoupling capacitor on the bottom PCB layer under the package for each group of three to four power supply pins.

2.4 Thermal Considerations

This subsection describes requirements to ensure the PCB provides adequate thermal dissipation to ensure compliance with the device operating temperature limitations and long-term reliability.

In battery-powered handset and PDA applications, large heat sinks and forced air cooling are obviously not practical. For the PXA270 processor, the package heat dissipation is accomplished primarily using conduction from the 36 center ground balls to the PCB ground plane. For this reason, the 36 center balls of the VF-BGA package are connected to a solid ground plane under the package using at least one for every four balls to ensure adequate thermal dissipation.

When the PXA27x processor IC package has a mechanical connection to the product case, this path also helps dissipate package thermal energy. Refer to Chapter 4 in the *Intel® Packaging Data Book* at http://www.intel.com/design/PACKTECH/packbook.htm for more information on package thermal requirements.

2.5 Package to Board Assembly Process

Refer to Chapter 14 in the *Intel® Packaging Data Book* at http://www.intel.com/design/PACKTECH/packbook.htm for more information on package to board assembly considerations.

2.6 Silicon Daisy Chain (SDC) Evaluation Units

Intel also offers evaluation units that have been internally shorted together (to the silicon) in a daisy chain pattern. This 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 set-up/evaluation of manufacturing equipment.

2.7 Handling: Shipping Media

Intel® VF-BGA and FS-CSPs are shipped in either tape and reel or in mid-temperature thin matrix trays that comply with JEDEC standards. All JEDEC standard trays have the same 'x' and 'y' dimensions and are easily stacked for storage and manufacturing.

2.8 Preconditioning and Moisture Sensitivity

With most surface mount components, if the units are allowed to absorb moisture beyond a certain point, package damage occurs during the reflow process. Refer to Chapter 8 in the *Intel® Packaging Data Book* at http://www.intel.com/design/PACKTECH/packbook.htm for package preconditioning and moisture sensitivity requirements. Specific moisture classification levels are defined on the box label for the product.

2.9 Tray Specifications

This subsection describes the JEDEC tray specifications for configuring pick and place units. The JEDEC tray specifications for the VF-BGA are provided in a later revision.

Figure 2-10. FS-CSP (14x14) Tray Specification

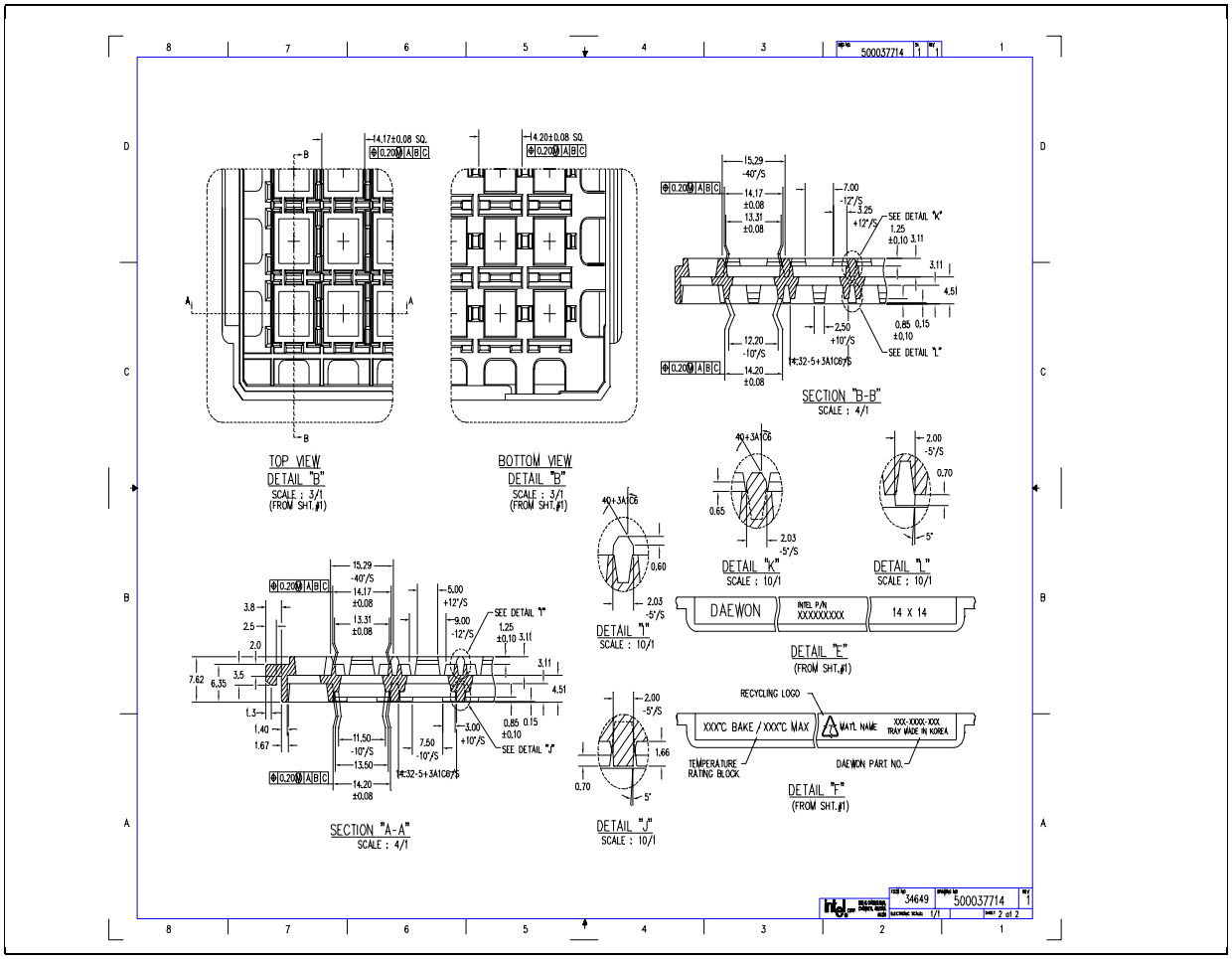
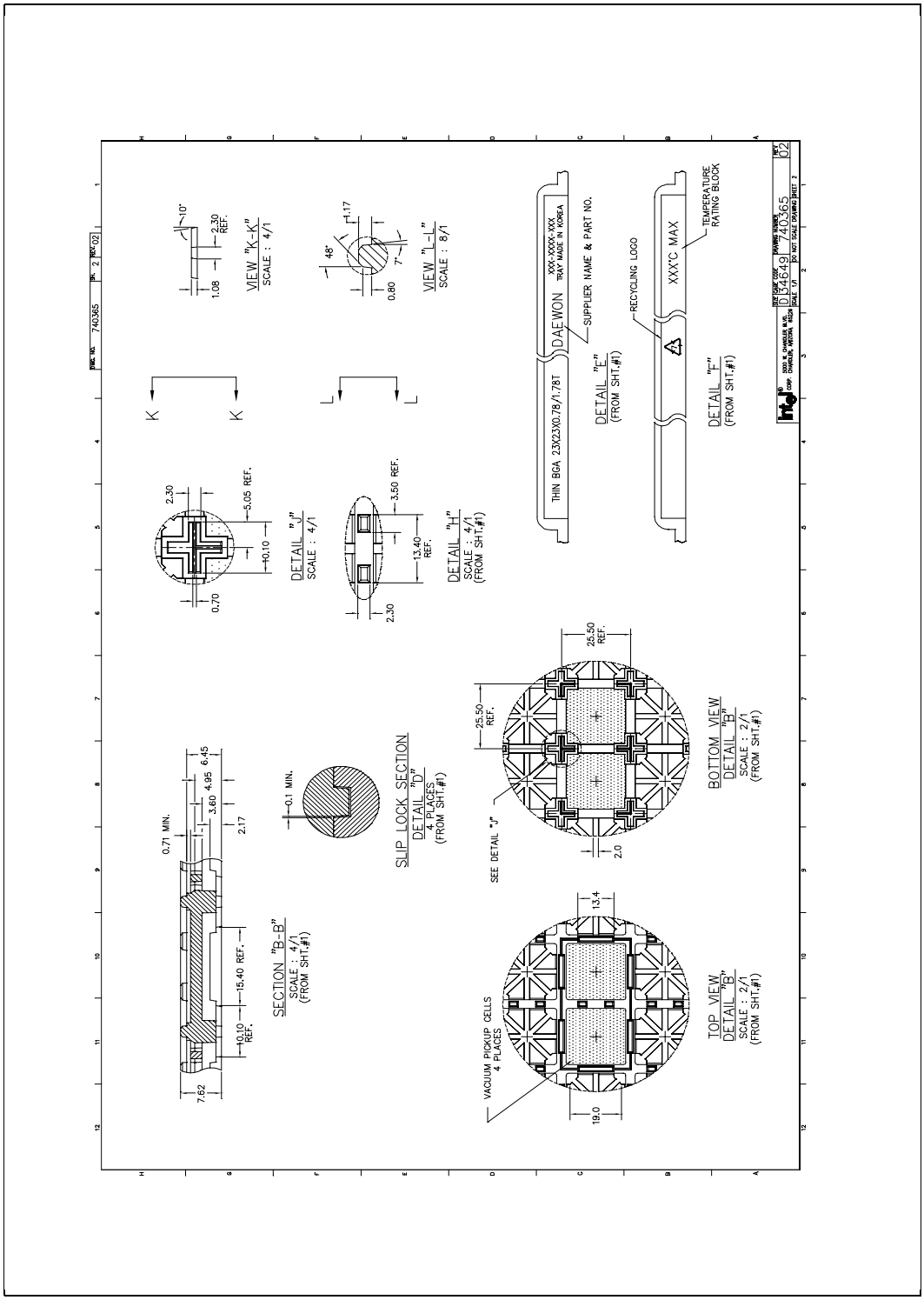


Figure 2-11. PBGA (23x23) Tray Specification



§§

Design Check List 3

For design check list information, refer to the *Intel® PXA27x Processor Family Design Check List Application Note*. See Table 1-1 for ordering information.

§§

Mixed Voltage Design Considerations 4

4.1 Overview

The Intel® PXA27x Processor Family (PXA27x processor) uses a complex power management system that provides the best possible power utilization. The power management system requires design of several voltage supplies into your system. Use a sophisticated power management integrated circuit (PMIC) in your system in order to utilize all the power savings possible with the PXA27x processor.

However, it is not required to incorporate a PMIC into a PXA27x processor-based system. By using four separate power regulators (optionally five), developing a power management system is possible for the PXA27x processor. However, the trade-off for such simplicity is increased power consumption and loss of flexibility in peripheral voltage selection and added requirements for battery backup.

4.2 Required Power Supplies

See Table 4-1 for the lists of the required power supplies and their associated values for the PXA27x processor. By careful selection, the three voltage supplies can supply the nine voltages. Using a 1.1 V, a 1.3 V, and a 3.3 V supply (all regulated at ±10%), satisfy all requirements for the nine voltage domains.

A fourth voltage regulator supplies the voltage for VCC_BATT. While the supplied voltage is the same voltage as a regulator already included in the design, a separate regulator must provide the supplied voltage in order to keep the voltage draw to a minimum. The current required to supply VCC_BATT is very small and so use a regulator with very small current source capability and correspondingly small current draw for the voltage supply. The other 3.3 V regulator powers the peripherals and must have much larger current sourcing capabilities. In a handheld system, the power draw for the voltage supply is not acceptable.

The trade-off for simplifying these nine supplies to three is that the peripheral supply voltages must all be 3.3 V. If 3.3 V is unacceptable to the system design, adding a fifth regulator of 1.8 V allows greater flexibility in the design without dramatically increasing the system cost or complexity. The addition of a fifth regulator also allows for a mixture of 3.3 V and 1.8 V peripherals for use in the system design.

Table 4-1. External Power Supply Descriptions

Name	Enable	Units	Default Voltage (V)	Allowed Levels (V)	Tolerance (%)
VCC_BATT	None	Sleep-active units, oscillators	3.0	3.0	±25
VCC_IO	SYS_EN	Peripheral I/O	3.3	3.0, 3.3	±10
VCC_LCD	SYS_EN	LCD I/O	3.3	1.8, 2.5, 3.0, 3.3	±10
VCC_USB	SYS_EN	USB I/O	3.3	3.0, 3.3	±10
VCC_MEM	SYS_EN	Memory Controller Interface	3.3	1.8, 2.5, 3.0, 3.3	±10
VCC_BB	SYS_EN	Baseband Interface	3.3	1.8, 2.5, 3.0, 3.3	±10
VCC_USIM[1]	SYS_EN	USIM Interface	3.3	1.8, 3.0, 3.3	±10
VCC_PLL	PWR_EN	Phase Locked Loops	1.3	1.3	±10
VCC_SRAM	PWR_EN	Internal SRAM units	1.1	1.1	±10
VCC_CORE	PWR_EN	All other internal units	any[2]	variable 0.85-1.55	±10

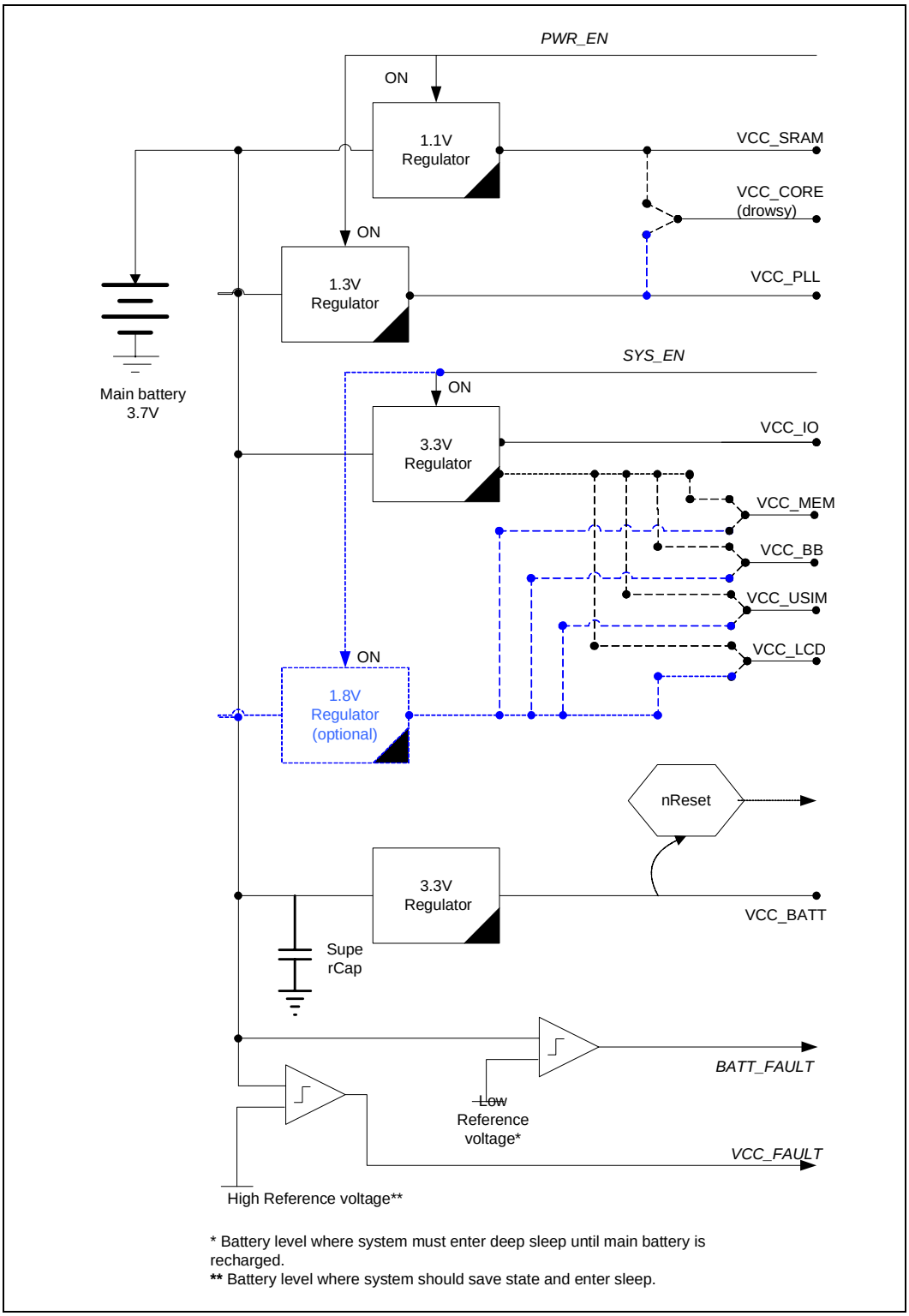
NOTES:
1. VCC_USIM must be held to the VCC_IO specification when using alternate function other than USIM signals.
2. Any legal voltage between 0.85 V and 1.55 V is valid. There is no default.

4.3 Example Power Supply Utilizing Minimal Regulators

See Figure 4-1 for an example of a power supply system for the PXA27x processor that uses the minimum number of voltage supplies.

The 1.8 V regulators are shown in dashed lines to indicate where the regulators are used. Use either the dashed black lines or the dashed lines if that peripheral is used in the system. However, ensure both voltages (if used) are not connected together.

Figure 4-1. Minimal Voltage Regulator Power System Design Example



4.4 Cautions

Observe the following cautions for proper operation and reliability of the PXA27x processor:

- All power domains are equal to or lower than VCC_IO (excluding VCC_USB).
- The application of VCC_BATT to the PXA27x processor from a state where there was no voltage applied to VCC_BATT is referred to as start-of-date. If only VCC_BATT is applied at the start-of-date, the current internal state of the PXA27x processor causes the current draw to be higher than when the PXA27x processor is in deep sleep mode. Such a situation is undesirable for systems that are production units, therefore, do not use these units immediately. After the backup battery is initially installed, if the devices are not brought into at least deep sleep mode through software, the backup battery experiences drainage. To avoid drainage of backup battery and to allow the backup battery to remain connected after production, boot the device and then place the device in deep sleep mode.

§§

Power Measurements 5

5.1 Overview

The Intel® PXA27x Processor Family (PXA27x processor) employs a complex power management system that provides the best possible power utilization. Take additional steps to minimize total power consumption. This chapter describes recommended methods for measuring power and provides the recommended steps in detail to minimize overall power usage.

5.2 Measurement Guidelines

There are two methods for measuring power on each of the power domains of the PXA27x processor:

- Measuring voltage across a series resistor
- Measure current directly with a current meter in series

The development design must allow for each power domain of the PXA27x processor to receive power independently through a dedicated shunt jumper followed by a shunt resistor for each power domain to use the two recommended methods.

No matter what method is used, measure the voltage at the PXA27x processor and monitor the voltage across all power domains during each measurement.

5.2.1 Measure Voltage Across a Series Resistor

This method measures voltage across a series resistor and calculates the total power consumed. This method involves using equipment to measure a voltage across a known series resistance and calculating the current works for calculating power during normal mode operations. This method is preferred for high current power supplies such as VCC_CORE. However, this method is not ideal for measuring low power due to the limitations of the equipment being used to measure power. The voltage drop when measuring low power could be too small for detection by the digital volt meter (DVM).

5.2.2 Measure Current Directly with a Current Meter in Series

This method measures current directly with a current meter in series, then calculates total power consumed. This methods works for all power modes, but could be too invasive and negatively affect the functioning of the PXA27x processor in some cases.

Use the following procedure to measure current directly with a current meter is series:

1. Check the internal resistance for the current meter.
2. Calculate the effective voltage drop across the internal resistance based of the maximum expected current.
3. Verify that the voltage drop minus the voltage supplying the voltage domain being tested does not fall below the required V_{in} for that domain.

Some DVMs have an internal resistance of 1.0 to 2.0 Ω that could cause the V_{in} to drop below the minimum required supply voltage depending on the power mode being used. If V_{in} drops below the minimum required supply voltage, the external power supply would be required to compensate.

The method of measuring current directly with a current meter in series is preferred for low current. Use the lowest possible range supported by the DVM. Avoid changing ranges during the operation of PXA27x processor.

5.3 Achieve Minimum Power Usage During All Power Modes

Methods for achieving the lowest power usage during all power modes:

- Connecting the crystal signals to an external differential oscillator achieves the least amount of power consumed by the PXA27x processor. Yet, the power consumed by the differential oscillator are considered for total system power usage. Exercise special care when selecting an external oscillator rather than an external crystal. Verify that the external oscillator is equally or more efficient than the PXA27x processor using a crystal.
- Design the system so that all power domains uses the lower voltage level supported within the specifications, that is, VCC_MEM – 1.8 V.
- Ensure I/O voltage levels track power domain levels, that is, for VCC_MEM – 1.8 V, all MD pins must drive at 1.8 V, otherwise it could cause negative current.
- Ideally, the system design allows measurements of power across all domains individually, helping the designer in isolating high current problems down to a specific voltage domain.
- Disable all internal pull-ups/pull-downs on GPIO, which is accomplished through PTCR register settings.
- Clearly define clock and memory controller settings.

5.4 Achieve Minimum Power Usage During Deep Sleep

Methods for achieving lowest power usage during deep sleep power modes:

- Ground all non VCC_BATT power domains to enable the designer find the lowest power consumption and validate the design.
- Monitor and register temperature sensitivity.
- Ensure TDI and TMS pins are pulled high or left floating.
- Ensure the USBC differential inputs (USBCP and USBCN) are pulled high or left floating with no impact to OTG pins.
- Enable DC-DC internal power supply.
- Ensure DC converter caps are connected properly.

5.5 Achieve Minimum Power Usage During Sleep

Methods used for achieving lowest power usage during sleep power modes:

- Ground SRAM/PLL/Core power domains to enable the designer find the lowest power consumption and validate the design.
- Configure GPIO signals of the PXA27x processor as input and drive a low from external source. Programmers can use PGSR/FS (GPIOs in O/P mode).
- Ensure TDI and TMS pins are pulled high or left floating.
- Ensure the USBC differential inputs (USBCP and USBCN) are pulled high or left floating with no impact to OTG pins.
- Enable DC-DC internal power supply.

5.6 Achieve Minimum Power Usage During Standby

Methods used to achieve lowest power usage during standby power modes:

- Ensure VCC_CORE and VCC_SRAM voltage is at 1.1 V.
- Configure GPIO signals of the PXA27x processor as input and drive a low from external source. Programmers can use PGSR/FS (GPIOs in O/P mode).
- Ensure TDI and TMS pins are pulled high or left floating.
- Ensure the USBC differential inputs (USBCP and USBCN) are driven high or left floating with no impact to OTG pins.

5.7 Achieve Minimum Power Usage During Idle/13M/ Run/Turbo

Methods used to achieve lowest power usage during idle, 13M, run and turbo power modes:

- Run/Turbo
  - Set auto power down (APD) bit.
  - Enable both instruction and data caches.
  - Use read allocate/write back (caching policy).
- 13M
  - Disable PLLs.
- Idle
  - Ensure interrupts are disabled to prevent unexpected walk-ups.

§§

Introduction to Part II 1

The chapters in Part II of the Design Guide describe the design recommendations and constraints related to specific on-chip peripherals of Intel® PXA27x Processor Family (PXA27x processor). These recommendations include information regarding signal connections, block diagrams, and notes related to system implementation.

The examples and schematics in this document represent one of the 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 consider the recommendations to ensure they are appropriate for your particular design.

Each peripheral is described in a separate chapter. The chapters in Part II are organized similarly to the chapters in the *Intel® PXA27x Processor Family Developers Manual* 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 manual and the information contained within must not replace careful consideration of system requirements and good design practices.

§§

Package and Pins 2

Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for complete information on processor signals, signal-to-package ball mapping, and package mechanical specifications for Intel® PXA27x Family Processor (PXA27x processor).

§§

Clocks and Power Interface 3

3.1 Overview

This chapter describes design recommendations and requirements for the external clock and power supply components connected to Intel® PXA27x Processor Family (PXA27x processor).

3.2 Signals

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

3.2.1 Clock Interface Signals

See Table 3-1 for definition of the PXA27x processor clocks signals.

Table 3-1. Clock Interface Signals (Sheet 1 of 2)

Name	Type	Definition
nRESET	Input	Active-low input Indicates to the processor to enter hardware-reset state.
nRESET_OUT	Output	Active-low output Indicates to the system that the processor is in a reset state (configurable for sleep and deep-sleep exit and GPIO reset).
GPIO<n>	Bidirectional	The GPIO<n> pins are used as standby and sleep wake-up sources. For possible values for *n*, refer to the GPIO chapter in the *Intel® PXA27x Processor Family Developers Manual*.
GPIO<3>	Bidirectional	The GPIO<3> pin is used as standby, sleep, and deep-sleep wake-up sources.
GPIO<1:0>	Bidirectional	The GPIO<1:0> pins are used as: • Standby and sleep/deep-sleep wake-up sources • Deep-sleep wake-up sources, after nBATT_FAULT or nVDD_FAULT is asserted
PXTAL_IN	Input	Connect to an external 13-MHz crystal or to an external clock source. For more information, refer to Section 3.5.1, “Clock Interface.”
PXTAL_OUT	Analog	Connect to an external 13-MHz crystal or to an external clock source that is complementary to PXTAL_IN or floated.
CLK_PIO	Bidirectional	Drives a buffered version of the PXTAL_IN oscillator input or used as a clock-input alternative to PXTAL_IN.
TXTAL_IN	Input	Connect to an external 32.768-KHz crystal or to an external clock source that is distributed to the timekeeping control system and power-management unit. Refer to Section 3.5.1, “Clock Interface,” for more information.
TXTAL_OUT	Analog	Connect to an external 32.768-KHz crystal or to an external clock source that is complementary to TXTAL_IN or floated.

Table 3-1. Clock Interface Signals (Sheet 2 of 2)

CLK_TOUT	Output	Drives a buffered and inverted version of the TXTAL_IN oscillator input.
CLK_REQ	Bidirectional	Input during power-on or hardware reset that indicates whether the processor oscillator clock input comes from PXTAL_IN (CLK_REQ low) or CLK_PIO (CLK_REQ floating). If CLK_PIO is the processor oscillator input, CLK_REQ becomes an output indicating when the processor oscillator is required. For more information, refer to Section 3.5.1, “Clock Interface.”
CLK_EXT	Input	Used by the Mobile Scalable Link (MSL), SSP or operating system (OS) timer module as a clock input (optional).

3.2.2 Power Manager Interface Control Signals

The PXA27x processor 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. There are also two Fault signals (nBATT_FAULT and nVDD_FAULT) required from the PMIC to communicate the onset of power supply problems to the processor.

The PXA27x processor communicates to the power controller using the signals defined in Table 3-2.

Table 3-2. Power Controller Interface Signals

Signal	Definition	Active State	Signal Direction†
PWR_EN	Power Enable	High	Output
SYS_EN	System Enable	High	Output
PWR_SCL	I^2C bus clock	Clock	Output
PWR_SDA	I^2C bus data	—	Bidirectional
nRESET	Forces an unconditional hardware reset	Low	Input
nBATT_FAULT	Indicates main battery removed or discharged	Low	Input
nVDD_FAULT	Indicates one or more supplies are out of regulation	Low	Input
PWR_CAP<3:0>	The PWR_CAP pins connect to external capacitors that are used with on-chip DC-DC converter circuits to achieve very low power in sleep mode.	—	Analog
PWR_OUT	Connects to an external isolated capacitor.	—	Analog
48_MHz	48 MHz output clock Used to generate peripheral timing from the 312-MHz peripheral clock	Clock	Output

† Input and Output refers to signal direction to or from the PXA27x processor.

3.2.3 Power Enable (PWR_EN)

PWR_EN is an active-high output from the PXA27x processor (input to the PMIC) that enables the external core power supplies (VCC_CORE, VCC_SRAM, VCC_PLL.) De-asserting PWR_EN indicates to the external regulator that the processor is going into sleep mode and that the low-voltage core power supplies are removed.

When PWR_EN is asserted, normal operation resumes and the PMIC turns on the core (low-voltage) supplies. The power controller must preserve, during sleep or deep sleep, the previous state of its regulators including the voltage for the core. Then, on resumption of core power, the regulators return to their last known voltage levels.

3.2.3.1 System Power Enable (SYS_EN)

SYS_EN is an active-high output from the PXA27x processor (input to the PMIC) that enables the external system power supplies. De-asserting SYS_EN indicates to the power supply that the processor is going into deep sleep mode, allowing the removal of high-voltage system power supplies (VCC_IO, VCC_LCD, VCC_MEM, VCC_USIM, VCC_BB, and VCC_USB). When powering on and off the various voltage domains, assertion and de-assertion of SYS_EN must occur in the correct sequence with PWR_EN to ensure the correct sequencing of power supplies.

When SYS_EN is asserted, normal operation resumes and the PMIC turns on the system I/O (high-voltage) supplies. Then, when PWR_EN is asserted, the PMIC turns on the core (low-voltage) supplies. The power controller must return all system I/O voltages to their pre-deep sleep mode levels.

3.2.3.2 Power Manager I^2C Clock (PWR_SCL)

The PWR_SCL signal is the power manager I^2C clock output to the external PMIC. The I^2C serial bus must operate at a minimum 40 KHz and optionally be capable of operating at 160 KHz clock rate.

3.2.3.3 Power Manager I^2C Data (PWR_SDA)

The PWR_SDA signal is the power manager I^2C data pin to the external PMIC. It functions like an open-drain signal so that either component pulls it down to a logic-low level.

3.2.3.4 nVDD_FAULT

nVDD_FAULT signals the PXA27x processor that one or more of its currently enabled supplies are below the minimum regulation limit (supplies that are not enabled, do not cause nVDD_FAULT assertion.) Functionally, nVDD_FAULT indicates to the processor when it is safe to exit sleep or when it must enter sleep (using the mechanism selected by the PMCR[xIDEA] bits) until the SYS_DEL timer expires. The PXA27x processor also has a configuration bit that allows nVDD_FAULT to be ignored in sleep mode. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for SYS_DEL and PWR_DEL timing specifications.

3.2.3.5 nBATT_FAULT

nBATT_FAULT indicates that the main battery is low or is being removed from the device, conveying to the PXA27x processor that power will be depleted shortly. During this time, the processor operates for a limited time from a Lithium/Lithium manganese “coin-cell” backup battery, or from a “super cap” that supplies the processor only for a few cycles of full-run power.

In the event of nBATT_FAULT assertion, the PXA27x processor enters an emergency form of sleep. In emergency sleep, the only handshaking is with external SDRAM memory (putting it into self-refresh mode.) This communication between the PXA27x processor and the external SDRAM memory ensures that memory contents are preserved, if possible. Obviously, the refresh currents eventually depletes the cap or backup battery, but not as fast as the PXA27x processor in run mode. Supporting these features must be understood at both the board level design and by the power controller/regulator.

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

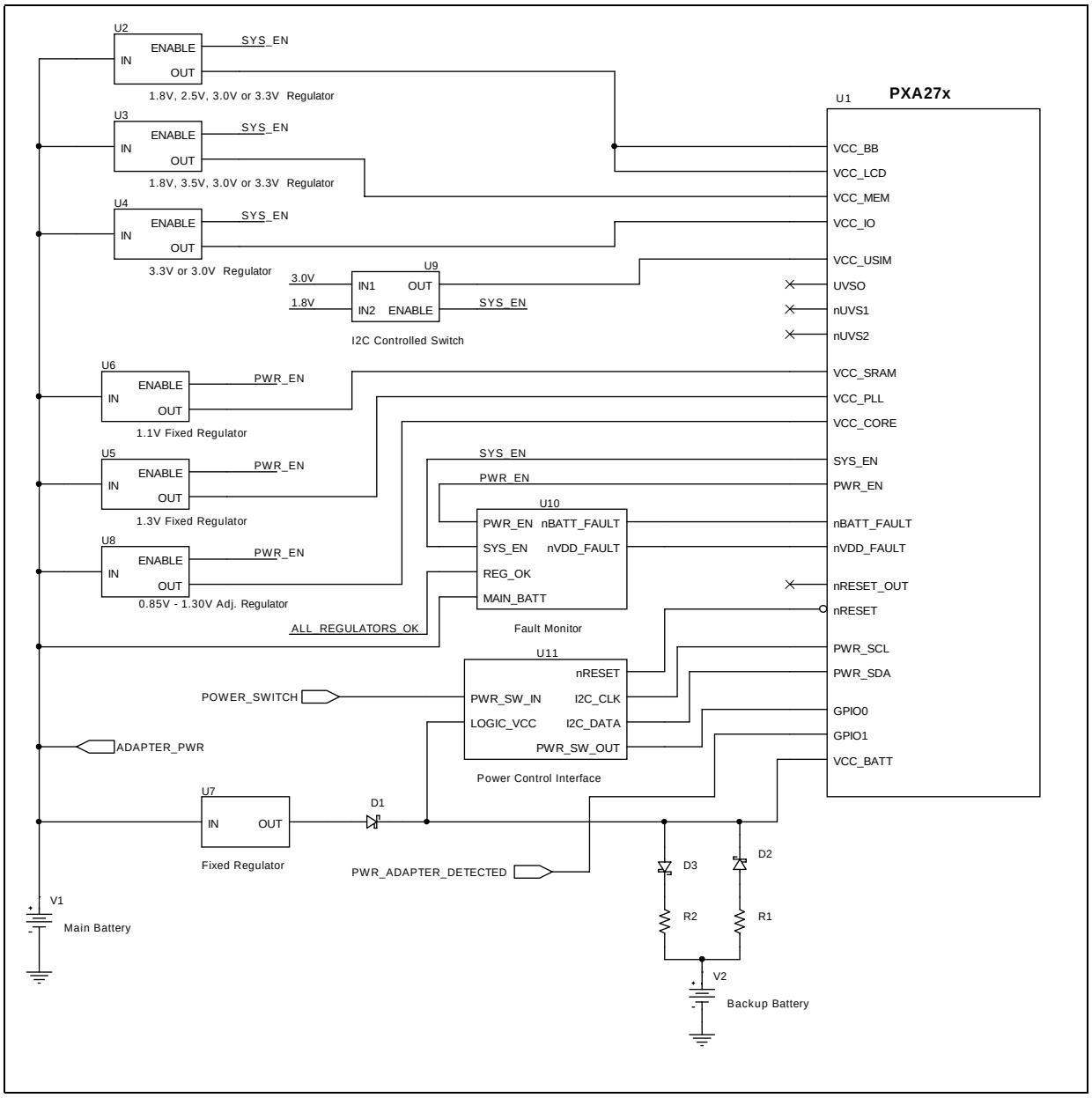
3.3 Block Diagram

See the system schematic in Figure 3-1 for illustration of one recommended configuration for connecting the PXA27x processor directly to the backup battery. In such a configuration, the regulated main battery powers VCC_BATT through regulator U7 and the backup battery powers VCC_BATT when the main battery discharges. Regulator U7 also charges the backup battery. The output voltage of U7 must ensure VCC_BATT remains:

- Between 2.25 V and 3.75 V when VCC_IO is disabled
- Within 200 mV of VCC_IO when VCC_IO is enabled

D1 protects regulator U7 from the back current when the backup battery drives VCC_BATT to a higher potential than the output of U7. D3 and R2 limit over-charging current to the backup battery. While the processor is powering up the VCC_REG domain from VCC_BATT, D2 and R1 prevent the PXA27x processor from over-charging the backup battery. See Figure 3-1. This preventive action from D2 and R1 occurs if an input signal on the VCC_REG domain is driven above the backup battery voltage.

Figure 3-1. Typical Battery and External Regulator Configuration



3.4 Layout Notes

Two external power capacitors must be connected to the PXA27x processor PWR_CAPx signals. The capacitors, C2 and C3, must be populated to achieve power efficiency using the DC-DC converter.

The sleep mode DC-DC converter requires three external components:

- C1: 0.1 µF capacitor connected to the PWR_OUT pin and ground (not optional)
- C2: 0.1 µF capacitor connected between the PWR_CAP0 and PWR_CAP1 pins
- C3: 0.1 µF capacitor connected between the PWR_CAP2 and PWR_CAP3 pins

The sleep/deep-sleep mode DC-DC converter is enabled by setting the DC_EN bit field in the Power Manager General Configuration register.

The DC-DC converter is used when all of these conditions apply:

Note: The sequence of setting these conditions is unimportant.

- The PXA27x processor is in sleep or deep-sleep mode.
- The PXA27x processor oscillator (13.000 MHz) is disabled.
- None of the internal SRAM banks, CPU, power island, or peripheral units are in a state-retaining mode.
- The power manager I^2C, JTAG, and timer are inactive.

The capacitors, C2 and C3, must be low equivalent series resistance (ESR), ceramic, unpolarized capacitors. No other connections are allowed on the PWR_OUT and PWR_CAP<3:0> pins.

3.5 Modes of Operations

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

3.5.1 Clock Interface

The PXA27x processor requires a 13-MHz timing reference that generates all core and most peripheral timing. While the real-time clock is operated from the 13-MHz reference to save cost and space in systems that do require high timekeeping accuracy, most systems use a 32.768-KHz timing reference. Using a 32.768-KHz timing reference dramatically reduces power consumption in the standby, sleep, and deep sleep operating modes and provides better accuracy with 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 PXA27x processor generates either clock using its oscillator, this clock drives the clock inputs of other system components such as a cellular baseband processor.

3.5.1.1 Using the On-Chip Oscillator with a 32.768-KHz Crystal

The *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* provide 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 processor. The on-chip oscillator provides the required load capacitance, so do not connect external load capacitors to the crystal. Place the crystal as close as possible to the TXTAL_IN and TXTAL_OUT pins of the processor 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 are used by other components in the system, connect these inputs to the processor CLK_TOUT output and enable it. Do not attempt to connect an external load directly to the TXTAL_OUT pin.

3.5.1.2 Using an External 32.768-KHz Clock

When using an external clock oscillator to supply the RTC timing, power up the oscillator from the same source that is driving the VCC_BATT supply input. Connect the oscillator output to the TXTAL_IN pin.

Note: The oscillator-output-high-drive level must be between 80% of VCC_BATT (0.8 x VCC_BATT) and VCC_BATT. Similarly, the oscillator-output-high-drive level must be between 80% of VCC_BATT (0.8 x VCC_BATT) and VCC_BATT. TXTAL_OUT is left floating or driving complimentary to TXTAL_IN.

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

The *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* provide 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 processor. To select the internal oscillator, ground the CLK_REQ input. The on-chip oscillator provides the required load capacitance, so do not connect an external load capacitors to the crystal. Place the crystal as close as possible to the PXTAL_IN and PXTAL_OUT pins of the processor 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 processor CLK_PIO output and enable it. Do not attempt to connect an external load directly to the PXTAL_OUT pin.

3.5.1.4 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 driving the VCC_BATT supply input. Connect the oscillator output to the CLK_PIO pin. Leave the CLK_REQ pin floating or connect it to the external oscillator's output enable input. During reset, sample the CLK_REQ pin to determine the oscillator configuration. During reset, when not driven externally, CLK_REQ pulls high internally. When CLK_REQ is pulled high internally, it allows CLK_PIO to be configured as an input and makes CLK_REQ an active high output enable to control the external oscillator. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for CLK_REQ timing specification.

Notes:

- The oscillator output high drive level must be between 80% of VCC_BATT (0.8 x VCC_BATT) and VCC_BATT. Similarly, the oscillator output low drive level must be between VSS and 20% of VCC_BATT (0.2 x VCC_BATT).
- CLK_REQ must not be pulled high or driven high externally.

3.5.2 Power Interface

To enable low power system design, the PXA27x processor has nine separate power supply domains for the processor 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, flexibility in selecting which peripherals are powered at the same time).

In a complete system, there are other components besides the processor such as DRAM and flash memory, audio CODECs, touchscreen controllers, and specialized companion chips with 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 PXA27x processor. An advanced power controller contains circuitry for charging batteries, for powering the display panel, and includes other analog functions as required by the system. The PXA27x processor provides several dedicated control signals as well as an Inter-Integrated Circuit (I^2C) interface to communicate with an external power management integrated circuit (PMIC).

3.5.2.1 Power Supplies

A summary of the voltage and tolerance requirements for each external supply input is provided in the *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification*. In systems which dynamically adjust the processor power supply and clock frequency to minimize power, there are additional requirements for the VCC_CORE power supply. Refer to the *Intel® PXA27x Processor Family Power Requirements Application Note* document for more information.

3.5.2.1.1 Power Supply Design Requirements

VCC_BATT powers the real time clock (RTC) and power management circuitry during initial power-on, sleep, deep sleep and sleep wake-up, so the RTC must remain powered from the backup battery when the main power source is discharged or removed. When the system main battery is installed, VCC_BATT must be driven by a regulator whose output matches the output of VCC_IO regulator. This ensures that VCC_IO and VCC_BATT remain within 200 mV of each other when the VCC_IO regulator is enabled. Power the external output drivers for the logic signals nRESET, nVDD_FAULT, nBATT_FAULT, PWR_SDA, GPIO0 and GPIO1 from the VCC_BATT supply. This also applies to all other digital outputs such as the JTAG signals driving PXA27x processor inputs on the VCC_REG domain. Refer to Figure 3-1. Any device that has a digital input driven by a PXA27x processor digital output and is powered from the VCC_REG domain, must tolerate output high drive levels between 2.25 V and 3.75 V.

VCC_PLL must driven by a regulator which cannot be shared with any other devices.

VCC_IO is the fixed supply for standard CMOS I/O interfacing to external components. VCC_IO must be the highest potential in the system (excluding VCC_BATT and VCC_USB) and must be turned on at the same time or before the other supplies enabled by SYS_EN. VCC_IO are connected to any of the VCC_USB, VCC_LCD, VCC_MEM, VCC_BB, or VCC_USIM supplies as long as none of these supplies are driven at a voltage higher than VCC_IO.

Caution: When VCC_IO is connected to a VCC_USIM supply with voltage higher than 3.0V, it does not damage the silicon, but exceeds the USIM card specification, which in turn can cause damage to USIM card.

VCC_IO must be enabled when SYS_EN is asserted and disabled when SYS_EN is de-asserted. The I/Os for external components connected to the corresponding I/O pins on the PXA27x processor must be supplied from the same regulator generating VCC_IO. VCC_USB powers the differential I/O signals for the USB client and host 1 interface. Be aware that the USB differential signals D+ and D- are out of compliance with the USB specification if VCC_USB is below 2.8 V. The +5 V V_{BUS} source from USB host controller, which is available for bus-powered peripherals, must be supplied from an external source.

§§

Internal SRAM 4

There is no existing hardware requirements related to the internal SRAM since the internal SRAM is not accessible using an external signal. This chapter provides information on the affects of the power capacitors on the internal SRAM.

4.1 Overview

The internal memory block provides 256 Kbytes of memory-mapped SRAM. The internal memory is divided into four banks, each is a 64 Kbytes bank. The memory has special power-saving features, including individual power management for each memory bank.

4.2 Signals

I/O signals are not associated with the internal memory block.

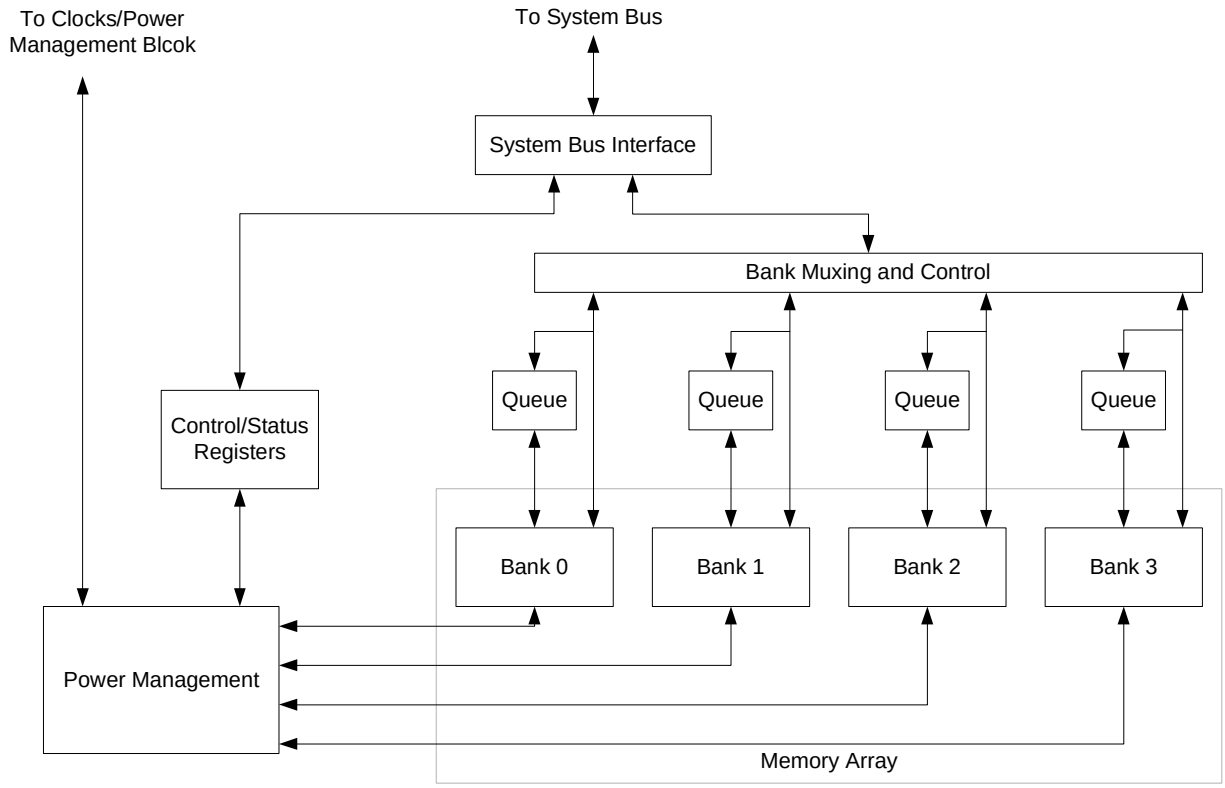
4.3 Block Diagram

The internal memory block has six major modules:

- System Bus Interface
- Control/Status Registers
- Power Management
- Memory Bank Muxing and Control
- Queues
- Four Memory Banks

See the internal memory block diagram in Figure 4-1.

Figure 4-1. Internal SRAM Block Diagram



ISRAM_001_P2

4.4 Layout Notes

The power caps allows the internal SRAM to be powered up during sleep. Refer to Section 3.4 for detailed information.

§§

DMA Controller Interface 5

This chapter describes the procedures for interfacing the DMA controller of the Intel® *PXA27x Processor Family* (PXA27x processor) with companion chips using the fly-by and flow-through DMA transfer.

5.1 Overview

The PXA27x processor contains a DMA controller that transfers data to and from the memory system in response to requests generated by peripherals or companion chips. These peripheral devices and companion chips do not directly supply addresses and commands to the memory system. Instead, the addresses and commands are maintained in 32 DMA channels within the DMA controller. Every DMA request from the peripheral device and companion chip generates a memory bus cycle. The DMA controller of the PXA27x processor supports both flow-through and fly-by transfers.

5.2 Signals

See Table 5-1 for the list of signals used to interface to the DMA controller.

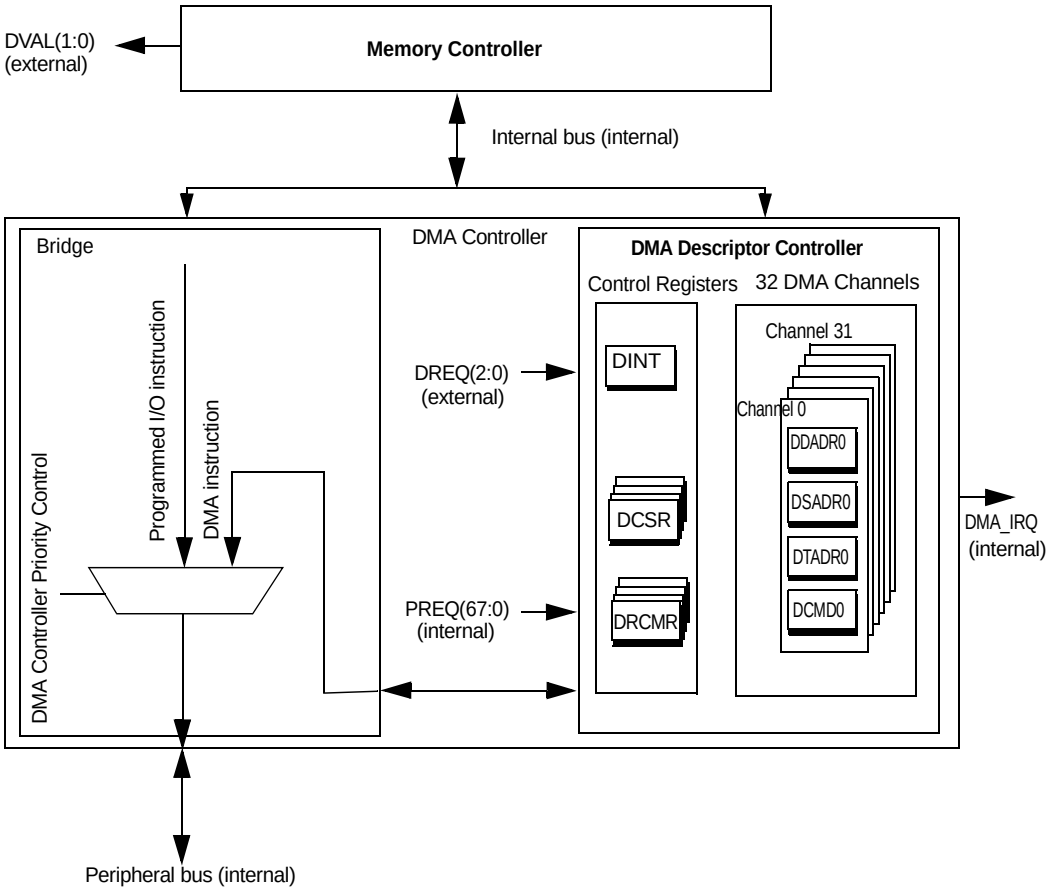
Table 5-1. DMA Interface Signals

Signal Name	Type	Description
DREQ<2:0>	Input	External Companion Chip Request The DMA controller detects the positive edge of the DREQ signal to log a request. The external companion chip asserts the DREQ signal when a DMA transfer request is required. The signal must remain asserted for four MEM_CLK cycles to allow the DMA controller to recognize the low-to-high transition. When the signal is de-asserted, the signal must remain de-asserted for at least four MEM_CLK cycles. The DMA controller registers the transition from low to high to identify a new request. The external companion chip need not wait until the completion of the data transfer before asserting the next request. This companion chip has up to 31 outstanding requests on each of the DREQ<2:0> pins. The number of pending requests are logged in special status registers, DRQSRx. Requests on pins DREQ<1:0> are used for data transfers in either fly-by or flow-through modes. Requests on pins DREQ<2> are used for data transfers in flow-through mode only.
DVAL<1:0>	Output	External Companion Chip Valid The memory controller asserts DVAL to notify the companion chip. Either data must be driven or is valid.

5.3 Block Diagram

See the block diagram for the DMA controller in Figure 5-1.

Figure 5-1. DMA controller Block Diagram



5.4 Layout Notes

The DREQ<2:0> signals must remain asserted for four CLK_MEM cycles for the DMA to recognize the low to high transition. When de-asserted, the DREQ<2:0> signals must remain de-asserted for at least four CLK_MEM cycles.

Refer to the *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for all AC timing information.

5.5 Modes of Operation

The following subsections describe the procedures for interfacing with the PXA27x processor DMA controller and memory controller when using either fly-by or flow-through DMA transfers.

5.5.1 Fly-By DMA Transfers

Fly-by transfers must occur only between any SDRAM partition and external peripherals or companion chips.

5.5.1.1 Signals

See Table 5-2 for the list of signals required for interfacing with the PXA27x processor using fly-by DMA transfer capabilities.

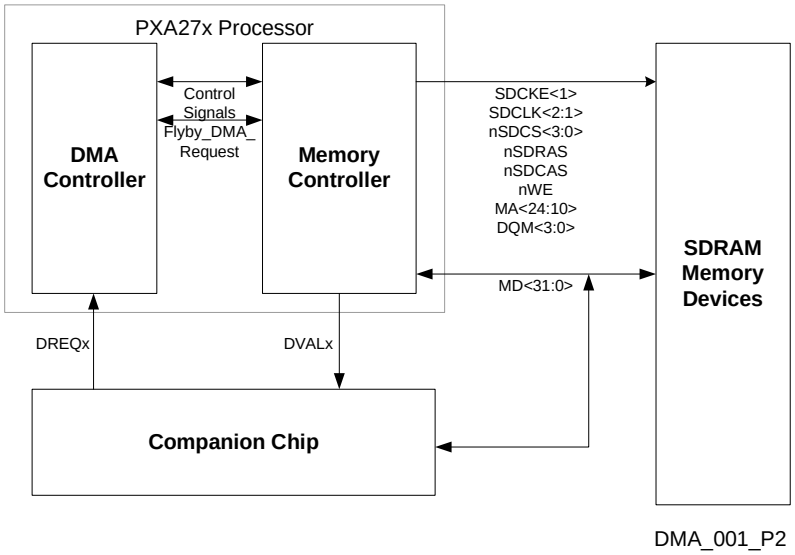
Table 5-2. Fly-By DMA Transfer Signals

Signal Name	Type	Description
DREQ<1:0>	Input	External Companion Chip Request The DMA controller detects the positive edge of DREQ signal to log a request. The external companion chip asserts the DREQ signal when a DMA transfer request is required. Requests on pins DREQ<1:0> are used for data transfers in fly-by mode.
DVAL<1:0>	Output	External Companion Chip Valid The memory controller asserts DVAL to notify the companion chip that data must be driven or is valid.

5.5.1.2 Block Diagram

See Figure 5-2 for illustration showing how to interface the PXA27x processor to a companion chip using fly-by DMA transfers to SDRAM memory devices.

Figure 5-2. Companion Chip Using Fly-by DMA Transfer Interface



5.5.1.3 Layout Notes

Using fly-by DMA transfers, high-performance companion chips are directly connected to the data bus of the SDRAM devices. All companion chips are restricted to transfers whose alignment and length match those of the SDRAM devices.

5.5.2 Flow-Through DMA Transfers

Unlike fly-by DMA transfers, flow-through DMA transfers have no restrictions on transferring data to either SDRAM partition, external peripherals or companion chips.

5.5.2.1 Signals

See Table 5-3 for the list of signals required when requesting flow-through DMA transfers from external logic.

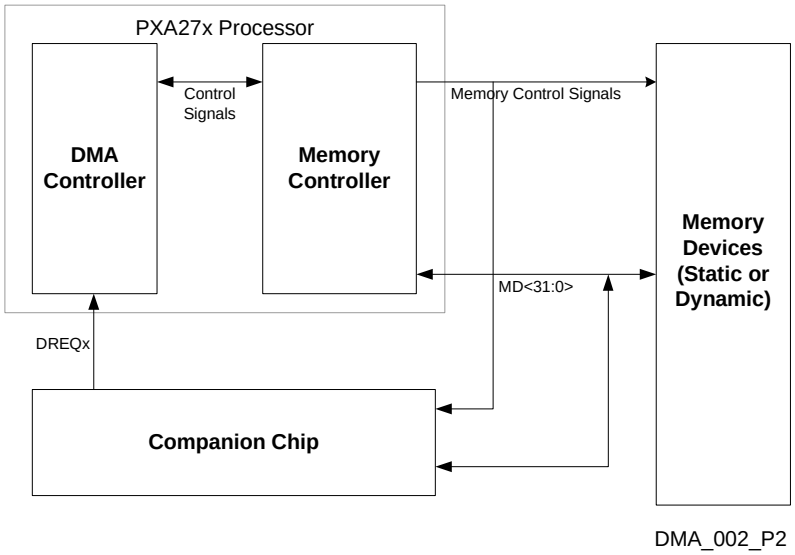
Table 5-3. Flow-Through DMA Transfer Signals

Signal Name	Type	Description
DREQ<2:0>	Input	External Companion Chip Request The DMA controller detects the positive edge of the DREQ signal to log a request. The external companion chip asserts the DREQ signal when a DMA transfer request is required. Requests on pins DREQ<2:0> are used for data transfers in flow-through mode.

5.5.2.2 Block Diagram

See Figure 5-3 for illustration showing how to interface the PXA27x processor to a companion chip using DMA flow-through transfers. In flow-through mode, a companion chip is addressed as if the companion chip is a memory device, allowing the companion chip to be the source or the target of flow-through DMA transfers.

Figure 5-3. Companion Chip Requesting Flow-Through DMA Transfers



5.5.2.3 Layout Notes

Refer to Section 5.4 for layout notes.

§§

System Memory Interface 6

This chapter describes guidelines for interfacing with the memory controller of Intel® PXA27x Processor Family (PXA27x processor) to external memory. See examples of schematics and timing diagrams for SDRAM, SRAM, flash, and PC Card devices.

6.1 Overview

The external memory bus interface for the PXA27x processor supports:

- SDRAM (100 MHz at 3.3 V, 3.0 V, 2.5 V, and 1.8 V)
- Flash (synchronous and asynchronous burst mode and page mode)
- ROM (page mode)
- SRAM (including variable latency I/O (VLIO) devices)
- PC Card (PCMCIA)/Compact Flash

In addition to supporting the memory types listed above, the memory controller of the PXA27x processor lets an alternate bus master request the bus and gain access to the SDRAM in partition 0.

See Table 6-1 for the physical addresses used to configure and map the external memory. Shaded cells in Table 6-1 indicate no alternate address mapping available. SDRAM and static memory are independently configured. Refer to the memory chapter in the *Intel® PXA27x Processor Family Developers Manual* on how to configure the SDRAM and static memory alternate memory map locations.

Table 6-1. Memory Address Map (Sheet 1 of 2)

Address	Mapped Function	Alternate Address	Alternate Mapped Locations
0x0000_0000	Static Chip Select 0 (64 Mbytes max)	0x0000_0000	Static Chip Select 0 (128 Mbytes max)
0x0400_0000	Static Chip Select 1 (64 Mbytes max)		
0x0800_0000	Static Chip Select 2 (64 Mbytes max)	0x0800_0000	Static Chip Select 1 (128 Mbytes max)
0x0C00_0000	Static Chip Select 3 (64 Mbytes max)		
0x1000_0000	Static Chip Select 4 (64 Mbytes max)	0x1000_0000	Static Chip Select 4 (64 Mbytes max)
0x1400_0000	Static Chip Select 5 (64 Mbytes max)	0x1400_0000	Static Chip Select 5 (64 Mbytes max)
0x2000_0000	PCMCIA/CF Slot 0 (256 Mbytes)	0x2000_0000	PCMCIA/CF Slot 0 (256 Mbytes)
0x3000_0000	PCMCIA/CF Slot 1 (256 Mbytes)	0x3000_0000	PCMCIA/CF Slot 1 (256 Mbytes)

Table 6-1. Memory Address Map (Sheet 2 of 2)

Address	Mapped Function	Alternate Address	Alternate Mapped Locations
0x4800_0000	Memory Mapped Registers (Memory Ctl)	0x4800_0000	Memory Mapped Registers (Memory Ctl)
0xA000_0000	SDRAM Bank 0 (64 Mbytes max)	0x8000_0000	SDRAM Bank 0 (256 Mbytes max)
0xA400_0000	SDRAM Bank 1 (64 Mbytes max)	0x9000_0000	SDRAM Bank 1 (256 Mbytes max)
0xA800_0000	SDRAM Bank 2 (64 Mbytes max)	0xA000_0000	SDRAM Bank 2 (256 Mbytes max)
0xAC00_0000	SDRAM Bank 3 (64 Mbytes max)	0xB000_0000	SDRAM Bank 3 (256 Mbytes max)

6.2 Signals

See Table 6-2 for the list of interface signals from the entire external memory controller.

Table 6-2. PXA27x Processor Memory Controller I/O Signals (Sheet 1 of 2)

Signal Name	Direction	Polarity	Description
Shared PXA27x Processor Memory Controller I/O Signals			
MD<31:0>	Bidirectional	NA	Bidirectional data for all memory types During reads from 16-bit memory devices, the upper 16 data bits are internally pulled low.
MA<25:0>	Output	NA	Output address to all memory types
DQM<3:0>	Output	Active High	Data byte mask control DQM<0> corresponds to MD<7:0> DQM<1> corresponds to MD<15:8> DQM<2> corresponds to MD<23:16> DQM<3> corresponds to MD<31:24> 0 = Do not mask out corresponding byte 1 = Mask out corresponding byte
SDRAM and Static Memory I/O Signals			
SDCLK<2:0>	Output	Active High	Output clocks to clock external memory SDCLK0 is for synchronous flash memory SDCLK1 is for SDRAM partitions 0 and 1 SDCLK2 is for SDRAM partitions 2 and 3
SDCKE	Output	Active High	Output clock enable signals for external memory SDCKE is for all SDRAM memory partitions
nSDRAS	Output	Active Low	Row address for SDRAM
nSDCAS	Output	Active Low	Column strobe for SDRAM Also, nADV (address strobe) for synchronous flash
nSDCS<3:0>[1]	Output	Active Low	Chips selects for SDRAM
nCS<5:0>[1]	Output	Active Low	Chip selects for static memory
nWE	Output	Active Low	Write enable for SDRAM and static memory
nOE	Output	Active Low	Output enable for static memory
Miscellaneous I/O Signals			
RDnWR	Output	Active High	Data direction signal to be used by output transceivers 0 = MD<31:0> is driven by the PXA27x processor 1 = MD<31:0> is not driven by the PXA27x processor
RDY[1]	Input	Active High	Variable Latency I/O signal for inserting wait states 0 = Wait 1 = VLIO is ready
BOOT_SEL<0>	Input	Tied at board level	Boot Select signals – indicates the type of boot memory the system has 0 = 32-bit ROM/flash 1 = 16-bit ROM/flash

Table 6-2. PXA27x Processor Memory Controller I/O Signals (Sheet 2 of 2)

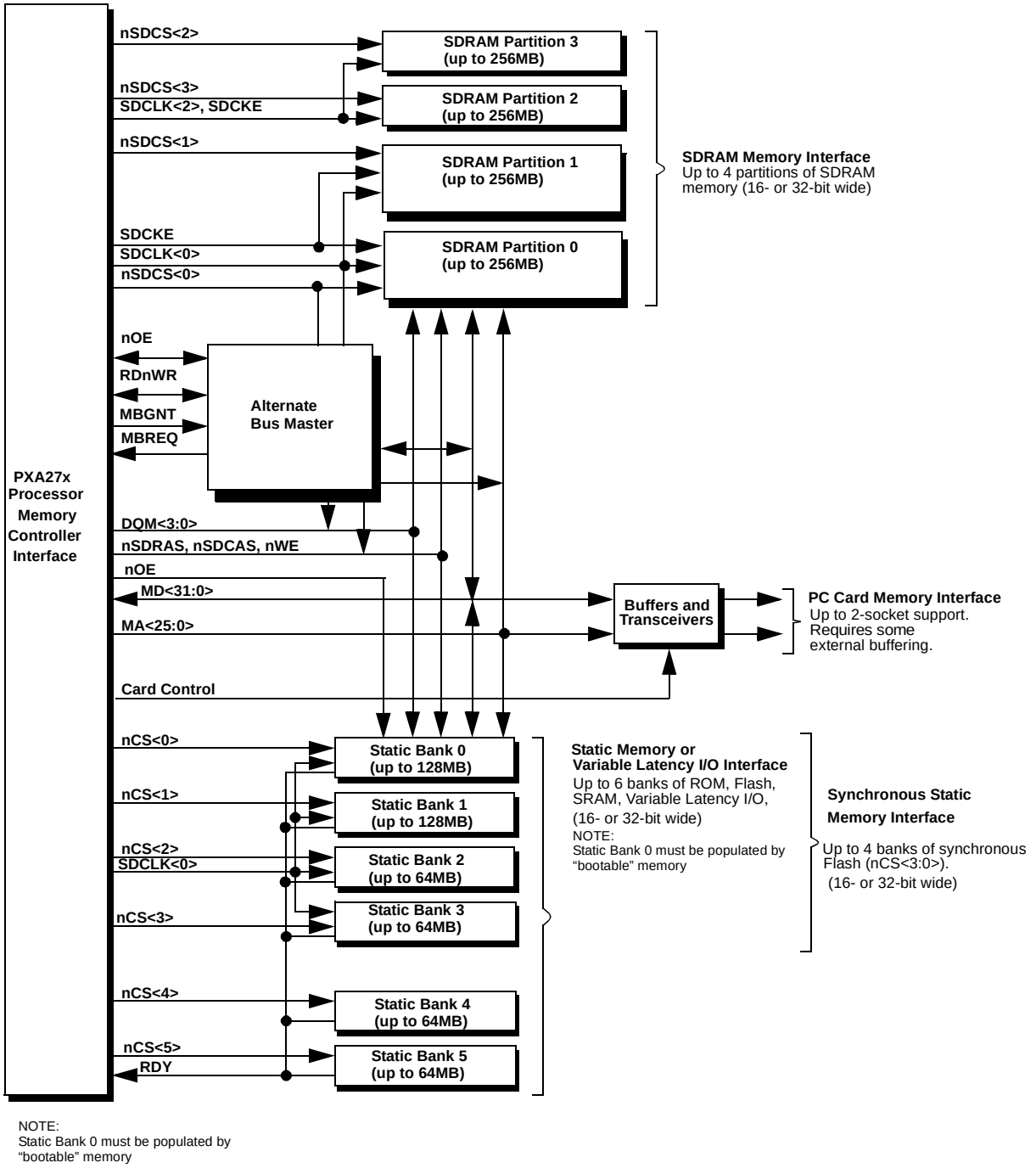
Signal Name	Direction	Polarity	Description
Alternate Bus Master Mode I/O Signals			
MBREQ[1]	Input	Active High	Alternate bus master request
MBGNT[1]	Output	Active High	Alternate bus master grant
Card Interface I/O Signals			
nPCE<2:1>[1]	Output	Active Low	Byte lane enables for the card interface. nPCE1 enables byte MD<7:0>; nPCE2 enables byte MD<15:8>
nPREG[1]	Output	NA	Serves as the card interface address bit <26> and selects register space (I/O or attribute) versus memory space
nPIOR[1]	Output	Active Low	Card interface I/O space output enable
nPIOW[1]	Output	Active Low	Card interface I/O space write enable
nPWE[1]	Output	Active Low	Card interface attribute and common memory space Write enable Also, write enable for variable latency I/O memory
nPOE[1]	Output	Active Low	Card interface attribute and common memory space output enable
nIOIS16[1]	Input	Active Low	Card interface input from I/O space telling size of data bus 0 = 16-bit I/O space 1 = 8-bit I/O space
nPWAIT[1]	Input	Active Low	Card interface input for inserting wait states 0 – Wait 1 – Card is ready
PSKTSEL[1]	Output	NA	In a single socket solution, this is the active low output enable used as the nOE for the data transceivers. In a dual socket solution, the socket select 0 – Socket 0 1 – Socket 1

NOTE:
1. The alternate function of the signal must be programmed to be accessed external of the PXA27x processor. Refer to the *Intel® PXA27x Processor Family Developers Manual* on how to configure the alternate function.

6.3 Block Diagram

See the block diagram in Figure 6-1 for illustration of how the memory controller signals of the PXA27x processor are used to interface to flash, static memory, SDRAM, PC Card/Compact Flash, and the use of an alternate bus master.

Figure 6-1. General Memory Interface Configuration



6.4 Memory Controller Layout Notes

This section contains information for recommended trace lengths, size, and routing guidelines for both the HD-CSP and FS-CSP configurations. The recommended targeted PCB trace impedance is 60 Ω ±15%.

6.4.1 Memory Controller Routing Guidelines for 0.5mm and 0.65 mm Ball Pitch

The following sections describes routing recommendations for HD-CSP using 0.5 mm and 0.65 ball pitch. The guidelines are recommendations and do not guarantee good signal integrity, but are a good starting point for a working solution.

The subsections describe how to minimize the read cycle hold violation and position clock with respect to data signal such that it meets both setup and hold requirements and eventually improve product yield. The recommended topologies are made to improve signal quality.

All guidelines within the section are based on a generic SDRAM driver with a driver impedance range 25-50 Ω and a rise/fall time range of 1-3 ns.

6.4.1.1 System Bus Recommended Signal Routing Guidelines (Excluding SDCLK<x> and SDCAS)

The goal is to achieve good signal quality at both driver and receiver as needed. Both data and clock trace lengths are very sensitive to timing. Lengths beyond the suggested region increases the risk of having setup or hold violation. The challenge is to deliver a non-monotonic clock to receivers at SDRAM and return-SDCLK for the PXA27x processor.

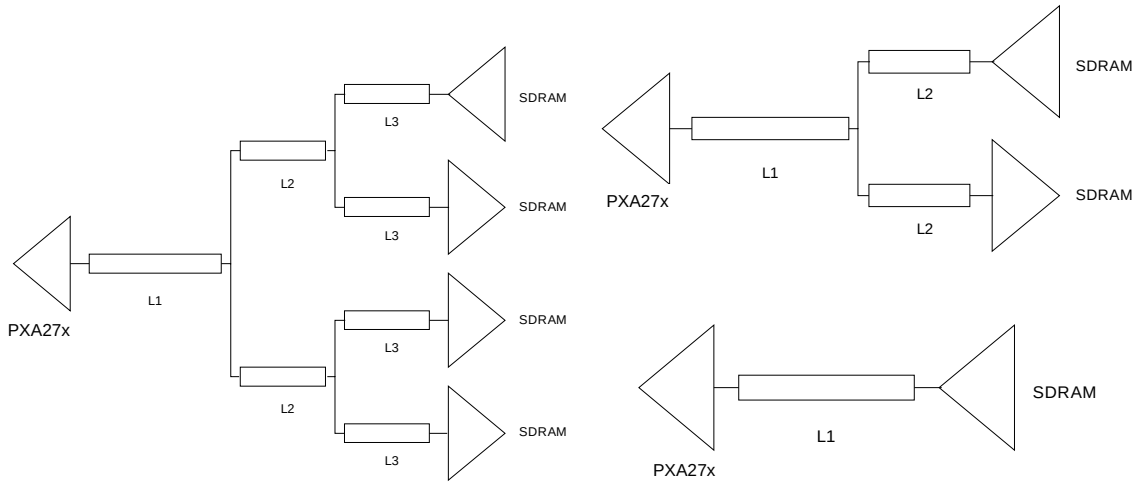
Use a balanced-T structure for routing of signals with more than one load. Routing with balanced-T structure requires more space than daisy chain. This limitation forces the design to add extra layers in the PCB overall design.

See Table 6-3 for details of the minimum and maximum trace lengths for all memory signals connected to the PXA27x processor memory controller, except the clock signals (SDCLKx) and SDCAS. The trace lengths are based on specific topologies from the memory controller. See Figure 6-2 for illustration of each topology.

Table 6-3. Minimum and Maximum Trace Lengths for the SDRAM Signals (Excluding SDCLK<x> and SDCAS)

Topology	Load	L1	L2	L3	PXA27x Strength	SDR Driver Impedance	SDR Driver rise/fall times	Motherboard Stack-Up
Point-to-point	1	3 - 4”	NA	NA	Code: 4-6	25 - 50 ohms	1 - 3 ns	Micro/strip with 60 Ω ±15% tolerance. 062 board
Balanced-T	2	2 - 3”	1 - 1.5”	NA	Code: 4-6			
Balanced-T	4	1 - 1.5”	0.5 - 0.75”	0.5 - 0.75”	Code: 4-6			
Note: Refer to Section 6.4.1.3 for information on board stack-up.								

Figure 6-2. PXA27x Processor Memory System Bus Routing Topologies (ExcludingSDCLK<x> and SDCAS)



6.4.1.2 SDCLK and SDCAS Recommended Signal Routing Guidelines

The longer clock trace lengths are needed to overcome hold violations while maintaining good setup margins. The SDCLK driver signal quality is very important for read cycles. The recommendations helps maintain non-monotonic clock edge for the return clock with the help of Schmitt's trigger of PXA27x SDCLK input path buffer.

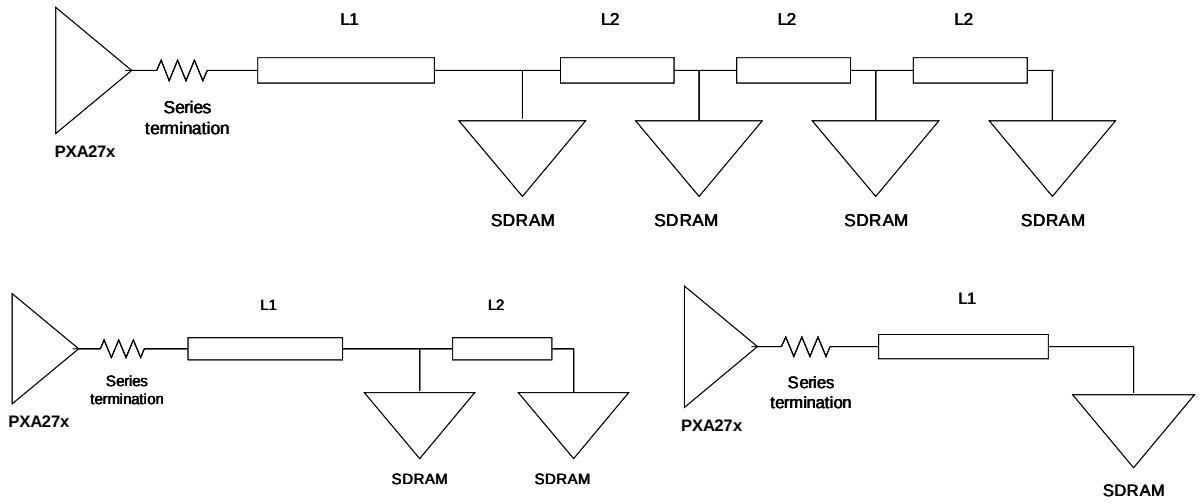
See Table 6-4 for details of the minimum and maximum trace lengths for all SDCLK<x> signals and SDCAS connected to the PXA27x processor memory controller. The trace lengths are based on a signal topology from the memory controller. See Figure 6-3 for illustration of the topology.

Table 6-4. Minimum and Maximum Trace Lengths for the SDCLK<x> and SDCAS signals

Topology	Load	L1	L2	Series Termination	PXA27x Strength	Motherboard Stack-Up
Point-to-point	1	5 - 6"	NA	20 ohms	Code: 4-B	Stripline with 60 Ω ±15% tolerance. 062 board.
Daisy Chain	2	5 - 6"	0.5"			
Daisy Chain	4	5 - 6"	0.5"			
Note: Refer to Section 6.4.1.3 for information on board stack-up.						

See Figure 6-3 for the recommended value for the 20 Ω ± 5% series termination.

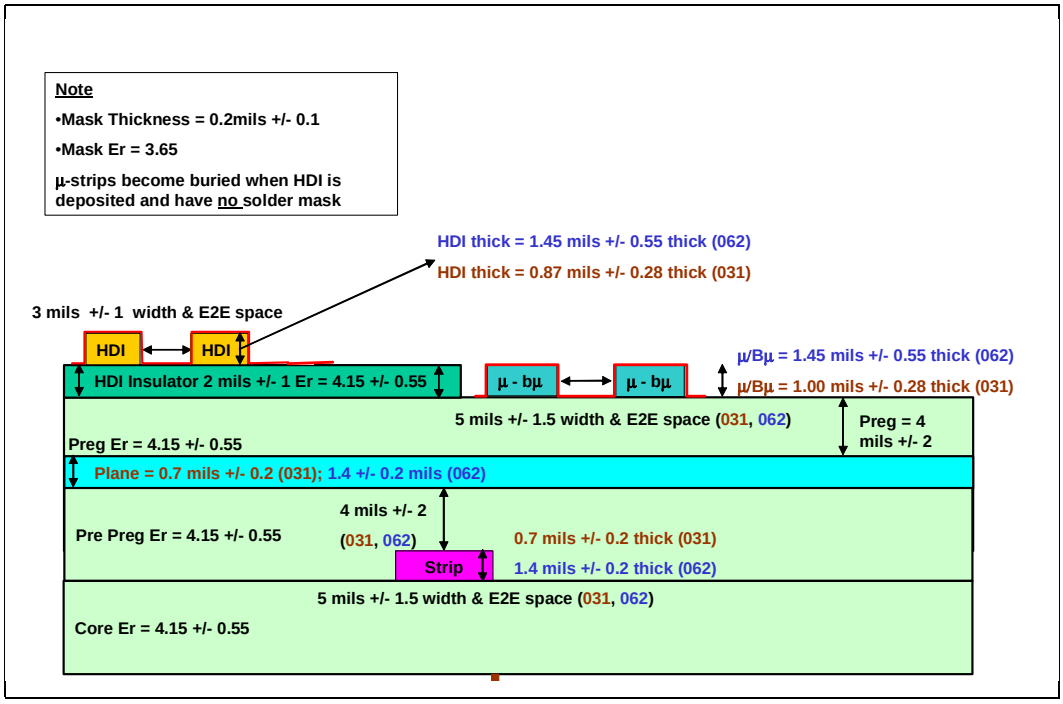
Figure 6-3. PXA27x Processor Memory Clock and SDCAS Routing Topology



6.4.1.3 Minimum Board Stack-up Configuration used for Signal Integrity

Information in Figure 6-4 indicates the board stack-up configuration used for signal integrity analysis. Refer to Part I: Section 2.2.1, "PCB Layer Assignment (Stackup)," for recommendation on board stack-up.

Figure 6-4. Minimum Board Stack-up Configuration used for Signal Integrity



6.5 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 PXA27x processor memory controller supported memories:

- Part II: Section 6.5.1, “SDRAM Interface”
- Part II: Section 6.5.2, “Flash Memory Interface (Asynchronous/Synchronous)”
- Part II: Section 6.5.3, “ROM Interface”
- Part II: Section 6.5.5, “Variable Latency Input/Output (VLIO) Interface”
- Part II: Section 6.5.6, “PC Card (PCMCIA) Interface”
- Part II: Section 6.5.7, “Alternate Bus Master Interface”

6.5.1 SDRAM Interface

The PXA27x processor supports an SDRAM interface at a maximum frequency of 100 MHz. The SDRAM interface supports four 16-bit or 32-bit wide partitions of SDRAM. Each partition is allocated 64 Mbytes of the internal memory map. However, the actual size of each partition is dependent on the particular SDRAM configuration used. The four partitions are divided into two partition pairs: 0/1 pair and the 2/3 pair. Both partitions within a pair must be identical in size and configuration. However, the two pairs can be different.

Example: Partition 0 and Partition 1. The 0/1 pair is 100-MHz SDRAM on a 32-bit data bus whereas the 2/3 pair is 66-MHz SDRAM on a 16-bit data bus.

6.5.1.1 SDRAM Signals

See Table 6-5 for the list of signals required to interface to SDRAM memory devices.

Table 6-5. SDRAM I/O Signals (Sheet 1 of 2)

Signal Name	Direction	Polarity	Description
SDCLK<2:1>	Output	Active High	Output clocks to clock external memory SDCLK1 is for SDRAM partitions 0 and 1 SDCLK2 is for SDRAM partitions 2 and 3
SDCKE	Output	Active High	Output clock enable signals for external memory SDCKE is for all SDRAM memory partitions
nSDCS<3:0>	Output	Active Low	Chips selects for SDRAM
MA<24:10>	Output	NA	Output address to all memory types
MD<31:0>	Bidirectional	NA	Bidirectional data for all memory types
DQM<3:0>	Output	Active High	Data byte mask control DQM<0> corresponds to MD<7:0> DQM<1> corresponds to MD<15:8> DQM<2> corresponds to MD<23:16> DQM<3> corresponds to MD<31:24> 0 = Do not mask out corresponding byte 1 = Mask out corresponding byte

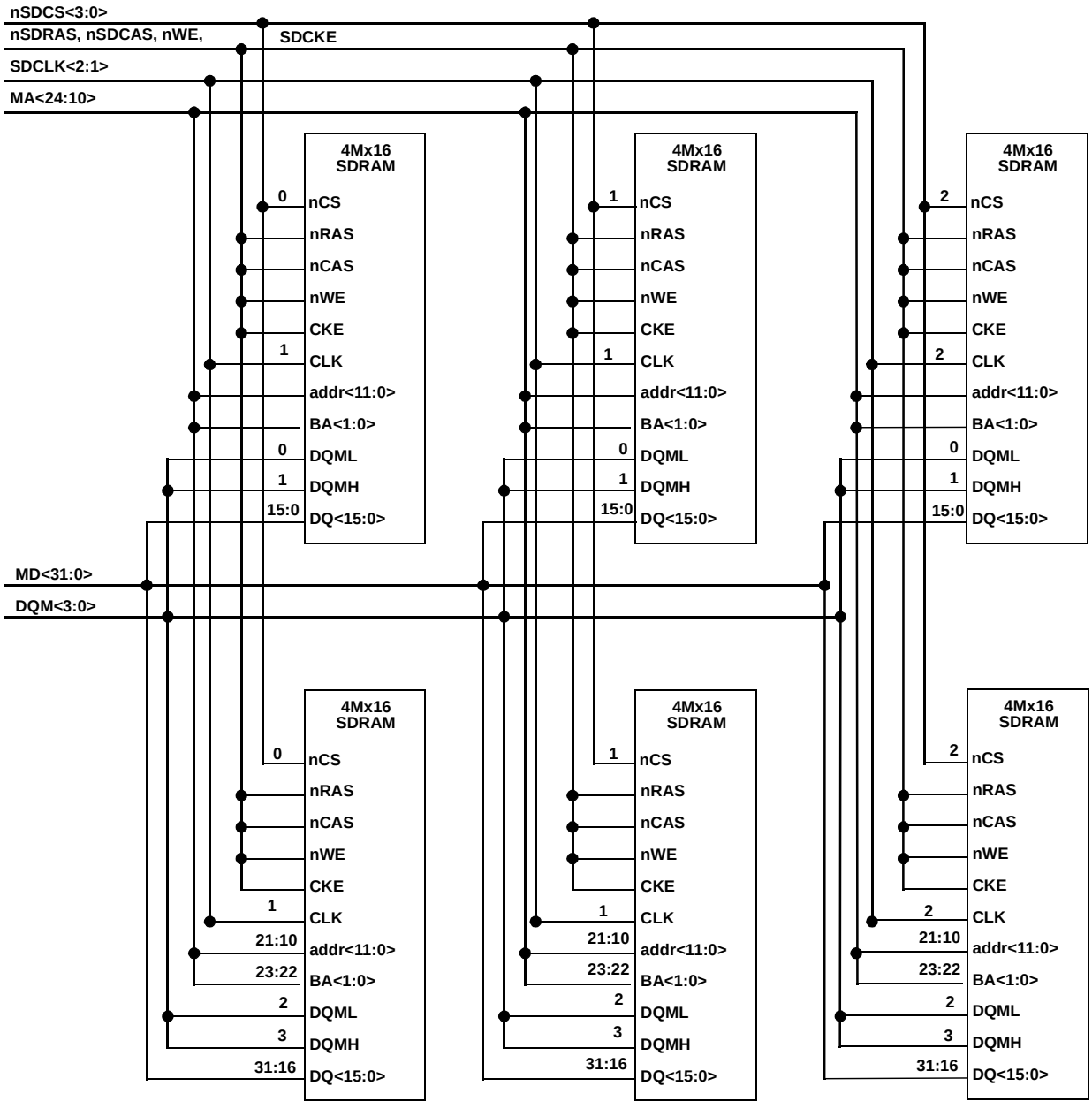
Table 6-5. SDRAM I/O Signals (Sheet 2 of 2)

Signal Name	Direction	Polarity	Description
nSDRAS	Output	Active Low	Row address for SDRAM
nSDCAS	Output	Active Low	Column strobe for SDRAM
nWE	Output	Active Low	Write enable for SDRAM and static memory
Miscellaneous I/O Signals			
RDnWR	Output	Active High	Data direction signal to be used by output transceivers 0 = MD<31:0> is driven by the PXA27x processor 1 = MD<31:0> is not driven by the PXA27x processor

6.5.1.2 SDRAM Memory Block Diagram

See Figure 6-5 for example of a wiring diagram showing a system using 1 Mword x 16-bit x 4-bank SDRAM devices for a total of 48 Mbytes. Refer to Section 6.5.1.3.2, "SDRAM Address Signal Mapping," on page II: 6-13, to determine the individual SDRAM component address.

Figure 6-5. SDRAM Memory System Example



6.5.1.3 SDRAM Layout Notes

For recommendations on trace lengths, size, and routing guidelines, refer to:

- Part II: Section 6.4, “Memory Controller Layout Notes”

For AC timing information, refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification*

6.5.1.3.1 SDRAM Memory Types Support

See Table 6-6 for the list of SDRAM memory types and densities that are supported by the PXA27x processor.

Table 6-6. SDRAM Memory Types Supported by PXA27x Processor

Chip Size	SDRAM Configu-ration (Words x Bits)	Bank Bits x Row bits x Column Bits	Partition Size (Mbyte/Partition)		Number Chips/ Partition		Total Number of Chips		Maximum Memory (4 Partitions)	
			16-bit Bus	32-bit Bus	16-bit Bus	32-bit bus	16-bit Bus	32-bit Bus	16-bit Bus	32-bit Bus
16 Mbit	1 M x 16	1 x 11 x 8	2 Mbyte	4 Mbyte	1	2	4	8	8 Mbyte	16 Mbyte
16 Mbit	2 M x 8	1 x 11 x 9	4 Mbyte	8 Mbyte	2	4	8	16	16 Mbyte	32 Mbyte
16 Mbit	4 M x 4	1 x 11 x 10	8 Mbyte	16 Mbyte	4	8	16	32	32 Mbyte	64 Mbyte
64 Mbit	2 M x 32	2 x 11 x 8	N/A	8 Mbyte	N/A	1	N/A	4	N/A	32 Mbyte
64 Mbit	4 M x 16	1 x 13 x 8 2 x 12 x 8	8 Mbyte	16 Mbyte	1	2	4	8	32 Mbyte	64 Mbyte
64 Mbit	8 M x 8	1 x 13 x 9 2 x 12 x 9	16 Mbyte	32 Mbyte	2	4	8	16	64 Mbyte	128 Mbyte
64 Mbit	16 M x 4	1 x 13 x 10 2 x 12 x 10	32 Mbyte	64 Mbyte	4	8	16	32	128 Mbyte	256 Mbyte
128 Mbit	8 M x 16	2 x 12 x 9	16 Mbyte	32 Mbyte	1	2	4	8	64 Mbyte	128 Mbyte
128 Mbit	16 M x 8	2 x 12 x 10	32 Mbyte	64 Mbyte	2	4	8	16	128 Mbyte	256 Mbyte
128 Mbit	32 M x 4	2 x 12 x 11	64 Mbyte	N/A	4	8	16	32	256 Mbyte	N/A
256 Mbit	16 M x 16	2 x 13 x 9	32 Mbyte	64 Mbyte	1	2	4	8	128 Mbyte	256 Mbyte
256 Mbit	32 M x 8	2 x 13 x 10	64 Mbyte	N/A	2	4	8	16	256 Mbyte	N/A
512 Mbit	32 M x 16	2 x 13 x 10	64 Mbyte	N/A	1	2	4	8	256 Mbyte	N/A

6.5.1.3.2 SDRAM Address Signal Mapping

See Table 6-7 and Table 6-8 for SDRAM address mapping. The bank select signals are listed in bold format to make it easier to locate signals.

Table 6-7. Normal and Alternate Mode Memory Address Signal Mapping

SDRAM		# Bits Bank x Row x Col	The processor pin mapping to SDRAM devices (The address lines at the top of the columns are the processor address lines)														
Device	Techno-logy		A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
1 M x 16	16 Mbit	1 x 11 x 8				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2M x 8	16 Mbit	1 x 11 x 9				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
4 M x 4	16 Mbit	1 x 11 x 10				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 12 x 8			BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 12 x 9			BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 12 x 10			BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 12 x 11			BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 13 x 8		BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 13 x 9		BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 13 x 10		BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 13 x 11		BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2 M x 32	64 Mbit	2 x 11 x 8			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		2 x 11 x 9			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		2 x 11 x 10			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
4 M x 16/ 4 M x 32	64 Mbit/ 128 Mbit	2 x 12 x 8		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8 M x 8/ 8 M x 16	64 Mbit/ 128 Mbit	2 x 12 x 9		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16 M x 4/ 16 M x 8	64 Mbit/ 128 Mbit	2 x 12 x 10		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
32 M x 4	128 Mbit	2 x 12 x 11		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8 M x 32	256 Mbit	2 x 13 x 8	BS1	BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16 M x 16	256 Mbit	2 x 13 x 9	BS1	BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
32 M x 16	512 Mbit	2 x 13 x 10	BS1	BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

See Table 6-8 for the list of physical signal connections to the SDRAM devices if the option to use SA-1110 address compatibility mode is chosen.

Table 6-8. SA-1110 Address Compatibility Mode Memory Address Signal Mapping

SDRAM		# Bits Bank x Row x Col	The processor pin mapping to SDRAM devices. (The address lines at the top of the columns are the processor address lines)															
Device	Techno-logy		MA 24	MA 23	MA 22	MA 21	MA 20	MA 19	MA 18	MA 17	MA 16	MA 15	MA 14	MA 13	MA 12	MA 11	MA 10
1 M x 16	16 Mbit	1 x 11 x 8				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2 M x 8	16 Mbit	1 x 11 x 9				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
4 M x 4	16 Mbit	1 x 11 x 10				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 12 x 8			A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 1 2x 9			A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 12 x 10			A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 12 x 11			A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 13 x 8		A12	A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x13 x 9		A12	A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 1 3 x 10		A12	A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1 x 13 x 11		A12	A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2 M x 32	64 Mbit	2 x 11 x 8			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		2 x 11 x 9			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		2 x 11 x 10			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
4 M x 16/ 4 M x 32	64 Mbit/ 128 Mbit	2 x 12 x 8		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8 M x 8/ 8 M x 16	64 Mbit/ 128 Mbit	2 x 12 x 9		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16 M x 4/ 16 M x 8	64 Mbit/ 128 Mbit	2 x 12 x 10		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
32 M x 4	128 Mbit	2 x 12 x 11		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8 M x 32	256 Mbit	2 x 13 x 8	A12	BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16 M x 16	256 Mbit	2 x 13 x 9	A12	BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

6.5.2 Flash Memory Interface (Asynchronous/Synchronous)

Memory types are programmable through the memory interface configuration registers. Refer to the *Intel® PXA27x Processor Family Developers Manual* for detail information on the configuration registers.

Six chip selects control the static memory interface, nCS<5:0>. All the chip selects are configurable for non-burst ROM or flash memory, burst ROM or flash, SRAM, or SRAM-like variable latency I/O devices. The variable latency I/O interface differs from SRAM in regard to its ability to allow the data ready input signal (RDY) to insert a variable number of memory-cycle-wait states. Program the data bus width for each chip select region to 16-bit (D<15:0>) or 32-bit (D<31:0>). nCS<3:0> signals are also configurable for synchronous static memory.

MA2 connects to the LSB of the static memory when the memory devices used are connected as 32-bit wide data bus interface. MA1 connects to the LSB of the static memory when the memory devices used are connected as 16-bit wide data bus interface.

6.5.2.1 Flash Memory Signals

See Table 6-9 for the list of signals required to interface to flash memory devices.

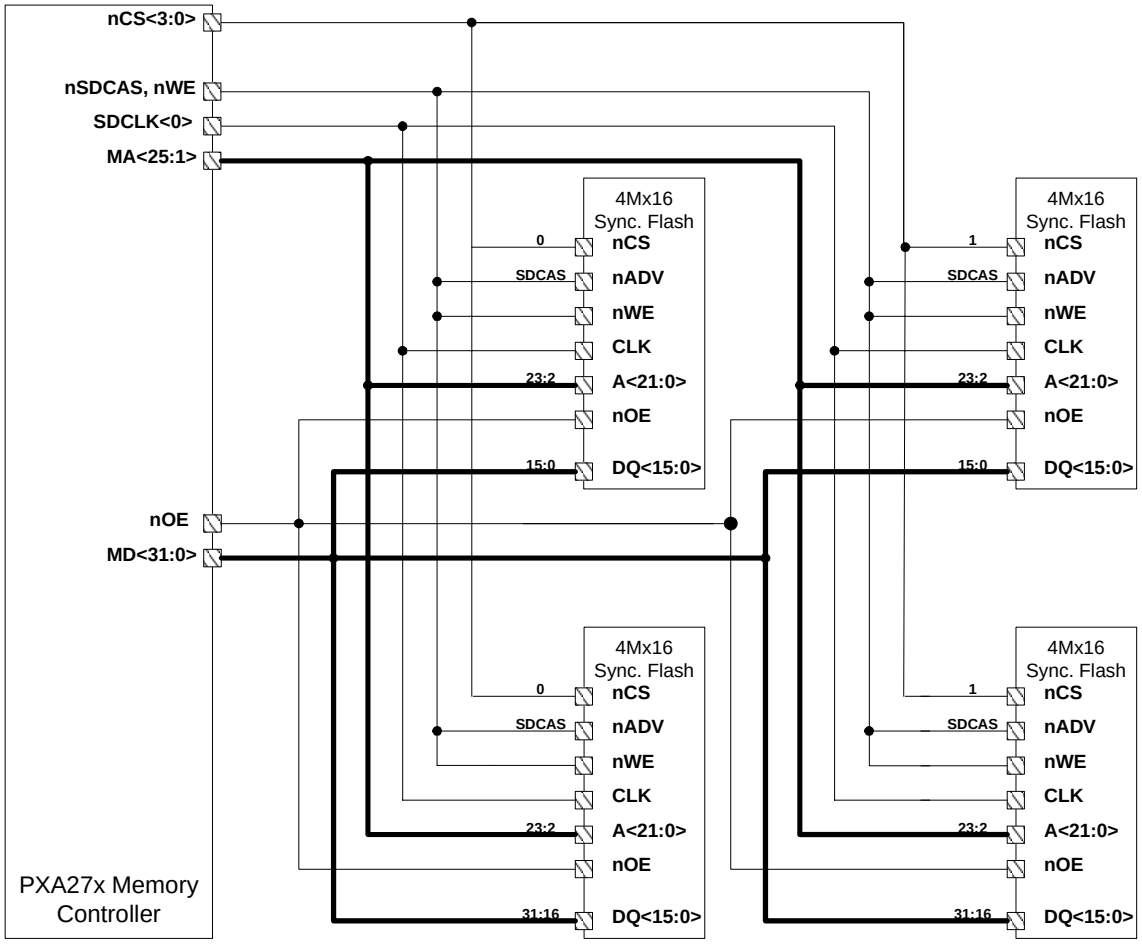
Table 6-9. Flash Interface Signals

Signal Name	Direction	Polarity	Description
Flash Interface Signals			
nCS<5:0>	Output	Active Low	Chip selects for static memory Only nCS<3:0> is configurable for synchronous flash memory
MA<25:0>	Output	NA	Output address to all memory types **NOTE:** Do not use MA0 for byte addressing because all flash devices must have a minimum bus width of 16 bits when interfacing to the PXA27x processor memory controller. Use MA0 to address the upper 64 MBytes of memory within a 128 MBytes partition.
nWE	Output	Active Low	Write enable for SDRAM and static memory
nOE	Output	Active Low	Output enable for static memory
MD<31:0>	Bidirectional	NA	Bidirectional data for all memory types
Additional I/O Signals Required to support Synchronous Flash Memory			
SDCLK0	Output	Active High	SDCLK<0> is for synchronous flash memory
nSDCAS	Output	Active Low	nADV (address strobe) for synchronous flash
Miscellaneous I/O Signals			
RDnWR	Output	Active High	Data direction signal to be used by output transceivers 0 = MD<31:0> is driven by the PXA27x processor 1 = MD<31:0> is not driven by the PXA27x processor
BOOT_SEL0	Input	Tied at board level	Boot Select signals allows two possible configuration for booting – indicates the type of boot memory possessed by the system 0 = 32-bit ROM/flash 1 = 16-bit ROM/flash

6.5.2.2 Flash Block Diagram

See Figure 6-6 for illustration of the connection between the synchronous flash memory and the PXA27x processor memory controller. This particular configuration shown in Figure 6-6 uses two partitions (chip select 0 and chip select 1). It is not required that both partitions be populated with synchronous flash memory.

Figure 6-6. Block Diagram Connecting Synchronous Flash to nCS<1:0>



6.5.2.3 Flash Layout Note

Refer to Section 6.4 for recommendations on trace lengths, size, and routing guidelines. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for AC timing information.

6.5.3 ROM Interface

6.5.3.1 ROM Signals

See Table 6-10 for the list of signals required to interface to ROM devices.

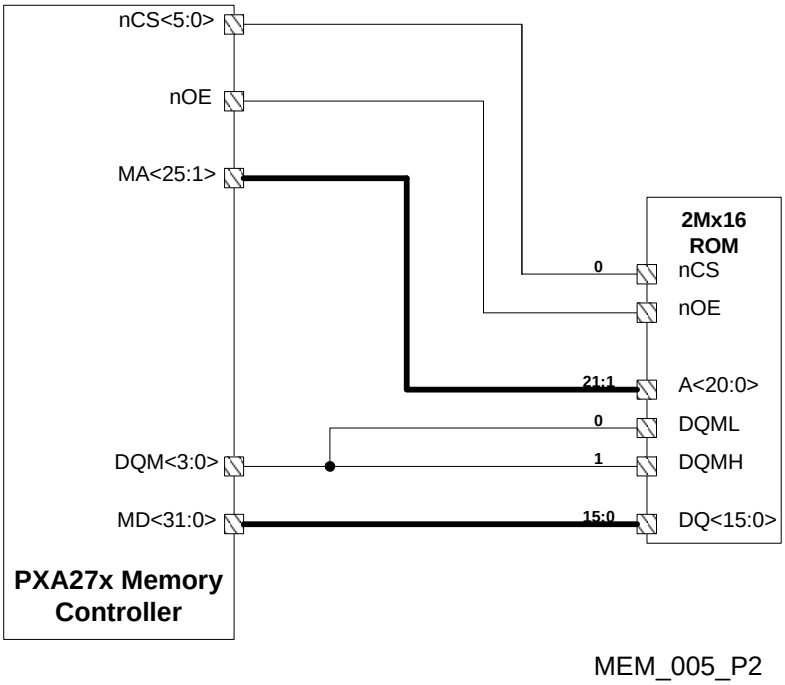
Table 6-10. ROM Interface Signals

Signal Name	Direction	Polarity	Description
ROM Interface Signals			
nCS<5:0>	Output	Active Low	Chip selects for static memory
MA<25:0>	Output	NA	Output address to all memory types **NOTE:** Do not use MA0 for byte addressing because all ROM devices must have a minimum bus width of 16 bits when interfacing to the PXA27x processor memory controller. Use MA0 to address the upper 64 MBytes of memory within a 128 MBytes partition.
MD<31:0>	Bidirectional	NA	Bidirectional data for all memory types
DQM<3:0>	Output	Active High	Data byte enable control DQM<0> corresponds to MD<7:0> DQM<1> corresponds to MD<15:8> DQM<2> corresponds to MD<23:16> DQM<3> corresponds to MD<31:24> 0 = Do not enable corresponding byte 1 = Enable corresponding byte
nOE	Output	Active Low	Output enable for static memory
Miscellaneous I/O Signals			
RDnWR	Output	Active High	Data direction signal to be used by output transceivers 0 = MD<31:0> is driven by the PXA27x processor 1 = MD<31:0> is not driven by the PXA27x processor

6.5.3.2 ROM Block Diagram

See Figure 6-7 for illustration of the connection between a 16-bit ROM and the PXA27x processor memory controller on chip select 0. Refer to this diagram when connecting SRAM and VLIO memories using a 16-bit interface and when using the procedure for connecting byte enable signals and data signals.

Figure 6-7. Block Diagram Connecting ROM to nCS<0>



6.5.3.3 ROM Layout Notes

Refer to Section 6.4 for recommendations on trace lengths, size, and routing guidelines. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for AC timing information.

6.5.4 SRAM Interface

For SRAM, DQM<3:0> signals are used for the write byte enables, where DQM<3> corresponds to the MSB in little endian mode. The processor supplies 26-bits of byte address for access of up to 128 Mbytes per chip select.

6.5.4.1 SRAM Signals

See Table 6-11 for the list of signals required to interface to SRAM devices.

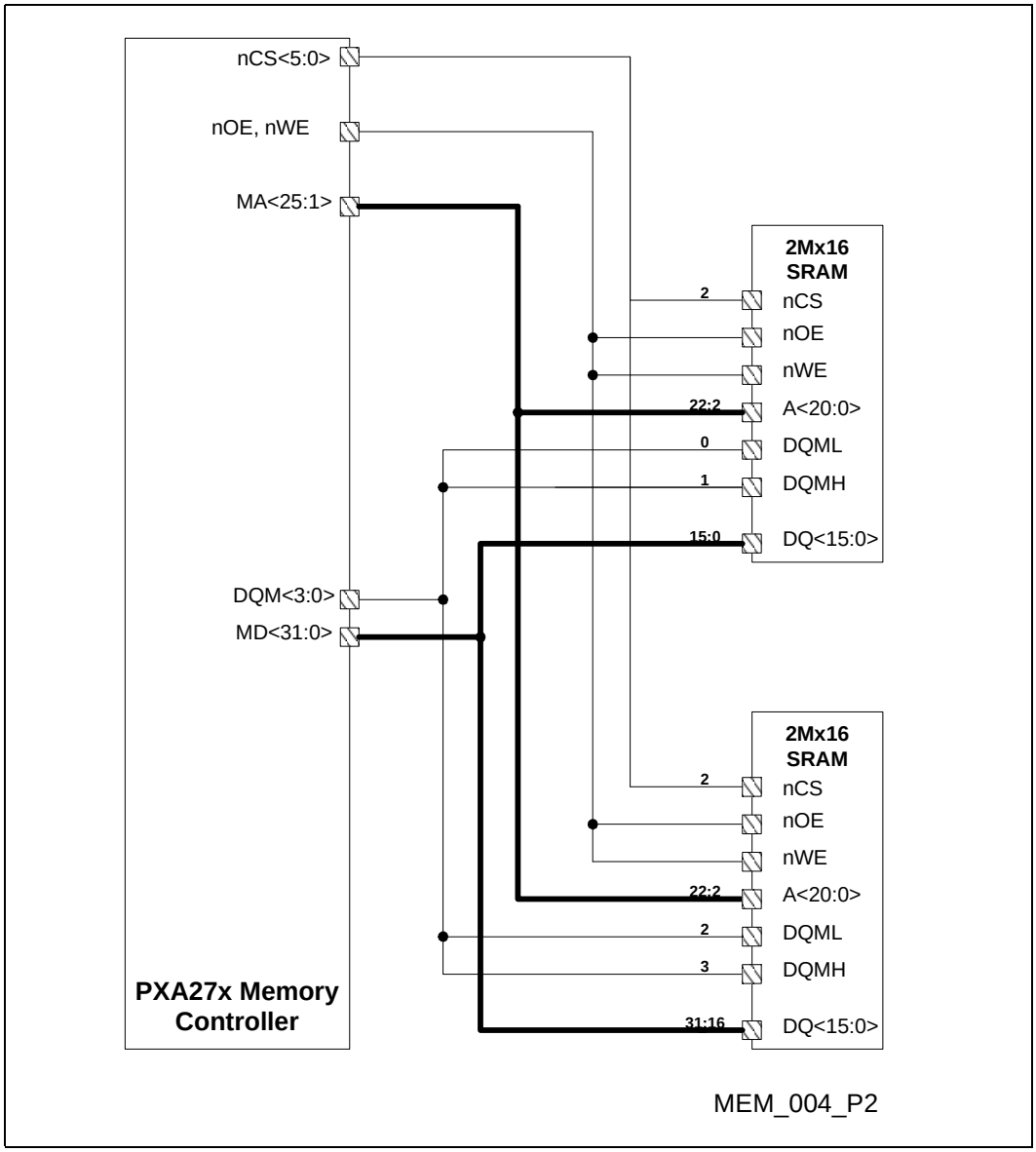
Table 6-11. SRAM Interface Signals

Signal Name	Direction	Polarity	Description
SRAM Interface Signals			
nCS<5:0>	Output	Active Low	Chip selects for static memory
MA<25:0>	Output	NA	Output address to all memory types Do no use MA0 for byte addressing because all SRAM devices must have a minimum bus width of 16 bits when interfacing to the PXA27x processor memory controller. Use MA0 to address the upper 64 Mbytes of memory within a 128 Mbytes partition.
MD<31:0>	Bidirectional	NA	Bidirectional data for all memory types
DQM<3:0>	Output	Active High	Data byte enable control DQM<0> corresponds to MD<7:0> DQM<1> corresponds to MD<15:8> DQM<2> corresponds to MD<23:16> DQM<3> corresponds to MD<31:24> 0 = Do not enable corresponding byte 1 = Enable corresponding byte
nWE	Output	Active Low	Write enable for SRAM memory
nOE	Output	Active Low	Output enable for static memory
Miscellaneous I/O Signals			
RDnWR	Output	Active High	Data direction signal to be used by output transceivers 0 = MD<31:0> is driven by the PXA27x processor 1 = MD<31:0> is not driven by the PXA27x processor

6.5.4.2 SRAM Block Diagram

See Figure 6-8 for illustration of the connection between a SRAM and the PXA27x processor memory controller. The particular configuration shown in Figure 6-8 connects to chip select 2, but it is possible to connect SRAM to any of the nCS signals on the PXA27x processor memory controller.

Figure 6-8. Block Diagram Connecting SRAM to nCS<2>



6.5.4.3 SRAM Layout Notes

Refer to Section 6.4 for recommendations on trace lengths, size, and routing guidelines. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for AC timing information.

6.5.5 Variable Latency Input/Output (VLIO) Interface

When a companion chip is used as a variable latency I/O, its functionality is similar to that of an SRAM. The chip is capable of inserting a variable number of wait states through the use of the RDY pin. Use variable latency I/O in the memory space for any of the six static memory locations (nCS<5:0>).

For variable latency I/O implementations, DQM<3:0> signals are used for the write byte enables, where DQM<3> corresponds to the MSB in little endian mode. The PXA27x processor supplies 26 bits of byte address for access of up to 128 MBytes per chip select.

Variable latency I/O read accesses differ from SRAM read accesses in that the nOE toggles for each beat of a burst. The first nOE assertion occurs two CLK_MEM cycles after the assertion of the chip select, nCS<x>. For VLIO writes, nPWE is used instead of nWE so SDRAM refreshes is executed while performing the VLIO transfers.

VLIO reads and writes differ from SRAM reads and writes. When the VLIO reads or writes, the PXA27x processor starts sampling the data-ready input (RDY) on the rising edge of CLK_MEM in two memory cycles. This is prior to the end of minimum nOE or nPWE assertion (MSCx[RDF]+1 memory cycles). The RDY signal is synchronized on input using a two-stage synchronizer, so when the synchronized signal is high, the signal indicates that the I/O device is ready for data transfer. RDY is tied high to cause a zero-wait-state I/O access. Read data is latched one memory cycle after the third successful sample (on the rising edge). nOE or nPWE is de-asserted on the next rising edge of CLK_MEM and the address changes on the subsequent rising edge of CLK_MEM. Prior to a subsequent data beat, nOE or nPWE remains de-asserted for RDN+1 memory cycles. The chip select and byte selects (DQM<3:0>) remain asserted for one memory cycle after the burst's final nOE or nPWE de-assertion.

For both reads and writes to and from VLIO, a special DMA mode exists that causes the address to not be increment to the VLIO. The special DMA mode allows port-type VLIO chips to interface to the PXA27x processor. This is only valid VLIO memory. For more information, refer to the DMA chapter in the *Intel® PXA27x Processor Family Developers Manual*.

For writes to VLIO, if all byte enables are turned off (masking out the data DQM = 0b1111), then the write enable is suppressed (nPWE = 1) for this write beat to VLIO. Turning off all byte enables causes a period when nCS is asserted, but neither nOE nor nPWE are asserted. This occurs when there is a write of 1 beat to VLIO and all byte enables are turned off.

The memory controller indefinitely waits for assertion of the RDY signal. This hangs the system if the external VLIO is not responding. System designers may want to consider a pull-up resistor on the RDY signal to ensure this signal is high under all conditions except when the companion chip drives this signal low to indicate the memory cycle needs extending.

6.5.5.1 VLIO Memory Signals

See Table 6-12 for the list of signals required to interface to VLIO memory devices.

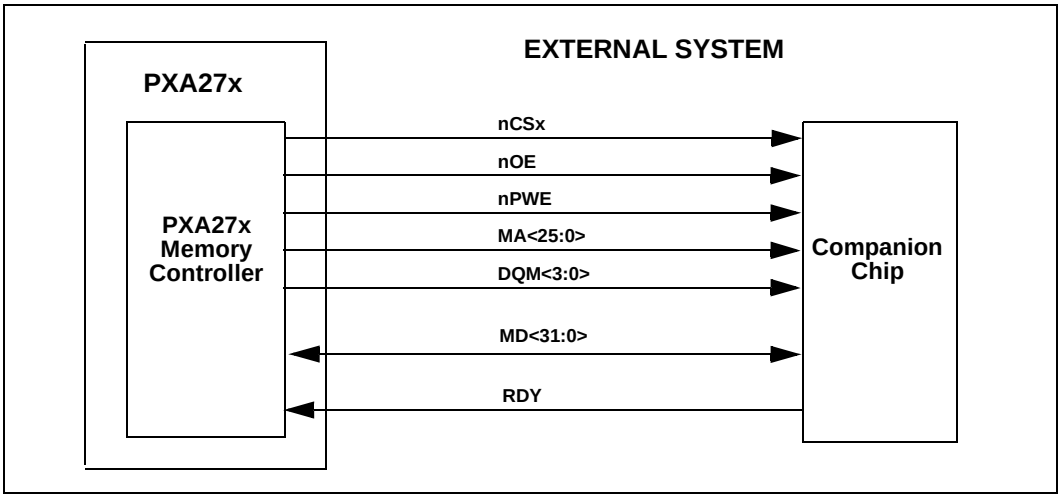
Table 6-12. VLIO Memory Interface Signals

Signal Name	Direction	Polarity	Description
VLIO Memory Interface Signals			
nCS<5:0>	Output	Active Low	Chip selects for static memory
MA<25:0>	Output	NA	Output address to all memory types **NOTE:** Do not use MA0 for byte addressing because all VLIO devices must have a minimum bus width of 16 bits when interfacing to the PXA27x processor memory controller. Use MA0 to address the upper 64 Mbytes of memory within a 128 Mbytes partition.
MD<31:0>	Bidirectional	NA	Bidirectional data for all memory types
DQM<3:0>	Output	Active High	Data byte mask control DQM<0> corresponds to MD<7:0> DQM<1> corresponds to MD<15:8> DQM<2> corresponds to MD<23:16> DQM<3> corresponds to MD<31:24> 0 = Do not mask out corresponding byte 1 = Mask out corresponding byte
nWE	Output	Active Low	Write enable for VLIO memory
nOE	Output	Active Low	Output enable for Static Memory
RDY	Input	Active High	Variable Latency I/O signal for inserting wait states 0 = Wait 1 = VLIO is ready
Miscellaneous I/O Signals			
RDnWR	Output	Active High	Data direction signal to be used by output transceivers 0 = MD<31:0> is driven by the PXA27x processor 1 = MD<31:0> is not driven by the PXA27x processor

6.5.5.2 VLIO Block Diagram

See Figure 6-9 for illustration of the signals when connecting a companion chip to the PXA27x processor using the VLIO memory interface.

Figure 6-9. Variable Latency Interface Block Diagram



6.5.5.3 VLIO Memory Layout Notes

Refer to Section 6.4 for recommendation on trace lengths, size, and routing guidelines. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for AC timing information.

6.5.6 PC Card (PCMCIA) Interface

The PXA27x processor requires external glue logic to complete the 16-bit PC Card socket interface that allows either 1-socket or 2-socket solutions.

The following illustrations show general solutions for a one- and two-socket configurations:

- Figure 6-10, “External Logic for a One-Socket Configuration Expansion PC Card,” on page II: 6-27
- Figure 6-11, “External Logic for a Two-Socket Configuration Expansion PC Card,” on page II: 6-28

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 controls the reset of the 16-bit PC Card interface, power selection (VCC and VPP), and drive enables. The INPACK# signal is not used.

The following illustrations show the logical connections necessary to support hot insertion capability:

- Figure 6-10, “External Logic for a One-Socket Configuration Expansion PC Card,” on page II: 6-27
- Figure 6-11, “External Logic for a Two-Socket Configuration Expansion PC Card,” on page II: 6-28

For dual-voltage support, level shifting buffers are required for all PXA27x processor input signals. Hot insertion capability requires that each socket be electrically isolated from the other and 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 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.

Note: If the system design incorporates PCMCIA interface, LCD and MSL (Baseband Interface), refer to Part II: Section 16.1, “Overview,” for important information on using these interfaces simultaneously.

6.5.6.1 PC Card Signals

See Table 6-13 for the list of signals required to interface to PC Card sockets.

Table 6-13. PC Card Interface Signals

Signal Name	Direction	Polarity	Description
PC Card Interface Signals			
nPCE<2:1>	Output	Active Low	Byte lane enables for the card interface nPCE1 enables byte MD<7:0>; nPCE2 enables byte MD<15:8>
nPREG	Output	NA	Serves as the card interface address bit 26 and selects register space (I/O or attribute) versus memory space
nPIOR	Output	Active Low	Card interface I/O space output enable
nPIOW	Output	Active Low	Card interface I/O space write enable
nPWE	Output	Active Low	Card interface attribute and common memory space write enable Also, write enable for variable latency I/O memory
nPOE	Output	Active Low	Card interface attribute and common memory space output enable
nIOIS16	Input	Active Low	Card interface input from I/O space telling size of data bus 0 = 16-bit I/O space 1 = 8-bit I/O space
nPWAIT	Input	Active Low	Card interface input for inserting wait states 0 – Wait 1 – Card is ready
PSKTSEL	Output	NA	In a single socket solution, this is the active low output enable which is used as the nOE for the data transceivers In a dual socket solution, the socket select 0 – Socket 0 1 – Socket 1
MA<25:0>	Output	NA	Output address to all memory types
MD<15:0>	Bidirectional	NA	Bidirectional data for all memory types
Miscellaneous I/O Signals			
RDnWR	Output	Active High	Data direction signal used by output transceivers 0 = MD<31:0> is driven by the PXA27x processor 1 = MD<31:0> is not driven by the PXA27x processor

6.5.6.2 PC-Card Block Diagrams

This section describes how to interface either one-socket PC Card adapter or a two-socket PC Card adapter to the PXA27x processor memory controller.

GPIO<102,86,15> has nPCE<1> as an alternate function and GPIO<105,87,78,54> has nPCE<2> as an alternate function. These pins are pulled low coming out of hardware reset and while the RDH bit has GPIOs in input/disabled mode. If these signals are used in transceiver control or multi-socket glue logic, it appears as though a PC Card or CF card is accessed and causes bus contention.

This causes unknown behavior 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.

In order to prevent erroneous nPCE<2,1> assertions during reset, the signals on GPIO<105, 102, 87, 86, 78, 54, 15> configured at nPCE must have board-level 4.7 KΩ pull-ups:

- Pull-up to VCC_MEM for nPCE configured on GPIO<78,15>
- Pull-up to VCC_BB for nPCE configured on GPIO<87,86,85,54>
- Pull-up to VCC_IO for nPCE configured on GPIO<105,102>

Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family 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.

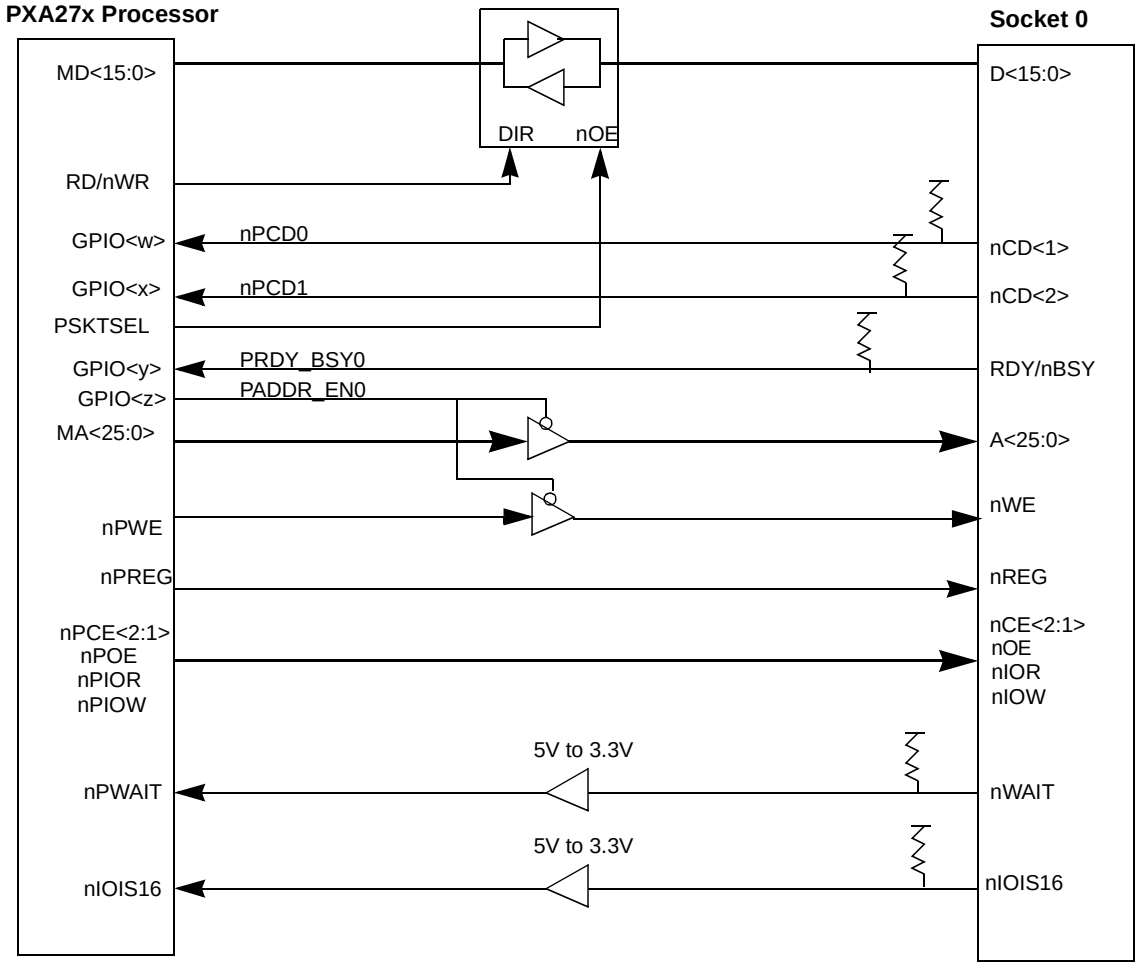
6.5.6.2.1 External Logic for One-Socket Card Implementation Block Diagram

See Figure 6-10 for illustration of the minimal glue logic needed for a 1-socket system. The illustration shows:

- Data transceivers
- Address buffers
- Level shifting buffers

The transceivers are enabled by the PSKTSEL signal. The DIR pin of the transceiver is driven by the RD/nWR 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.

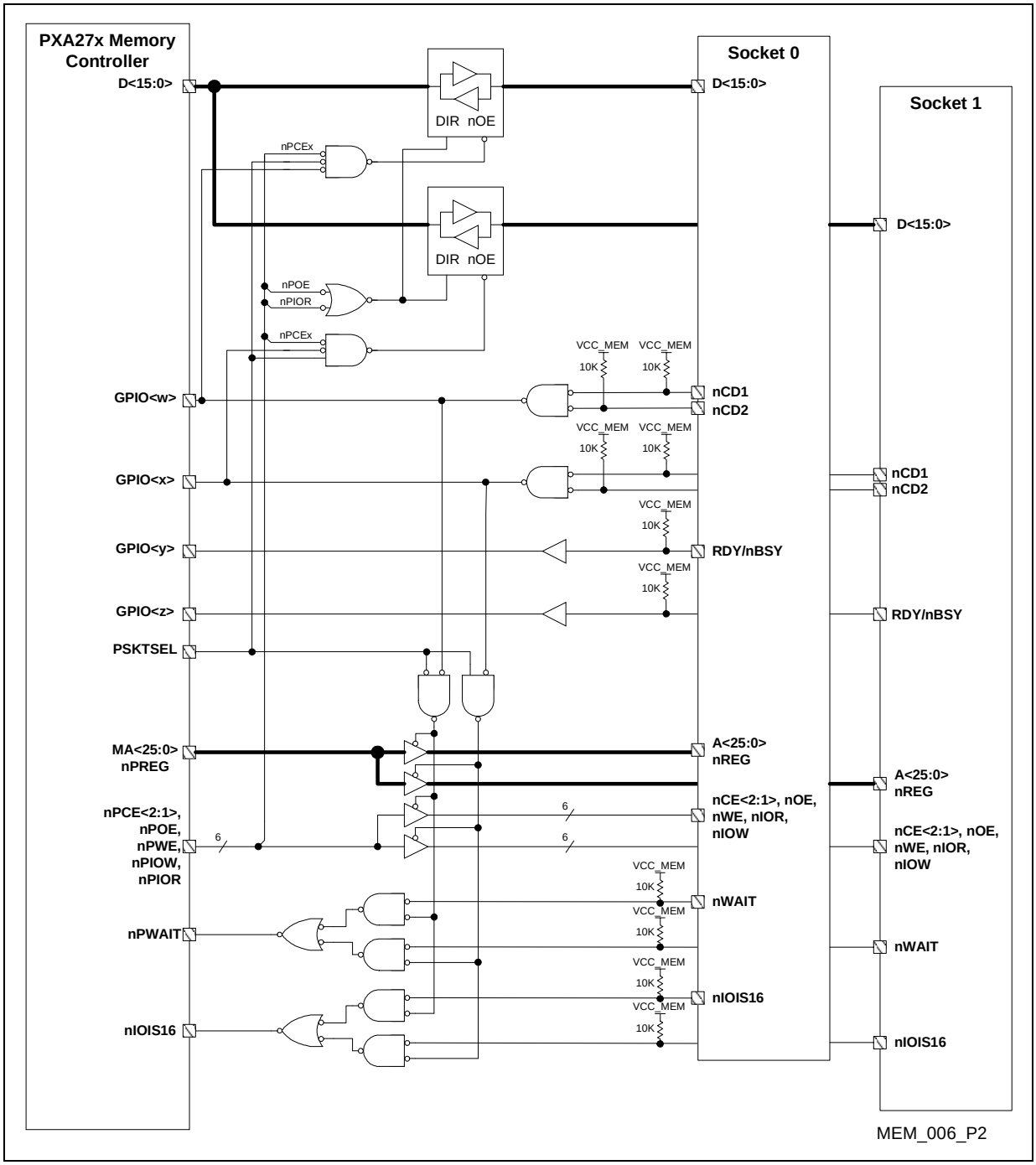
Figure 6-10. External Logic for a One-Socket Configuration Expansion PC Card



6.5.6.2.2 External Logic for Two-Socket Card Implementation Block Diagram

See Figure 6-11 for illustration of the glue logic need for a 2-socket system. RDY nBSY signals are routed through a buffer to two separate GPIO pins. In the data bus transceiver control logic, nPCE1 controls the enable for the low byte lane and nPCE2 controls the enable for the high byte lane.

Figure 6-11. External Logic for a Two-Socket Configuration Expansion PC Card



6.5.6.3 PC Card Layout Notes

Pull-up resisters shown in Figure 6-10 and Figure 6-11 must be 10 KΩ or greater in value. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for the A/C timings information.

Verify all signals are within the domain of their intended use. For example, some PC Card signals are on VCC_BB and some are on VCC_MEM. One solution is to tie the domains together that are used in common to a single interface. Another solution is to level shift one domain to equal that of the other domain being used.

6.5.7 Alternate Bus Master Interface

The PXA27x processor allows an alternate bus master to take control of the external memory bus and read/write data from the SDRAM in partition 0 (nSDCS<0>). The alternate master is given control of the bus using a hardware handshake. This handshake is performed through MBREQ and MBGNT that are invoked through the alternate functions on GPIO pins.

When the alternate master has to take control of the memory bus, it asserts MBREQ. The PXA27x processor completes any in-progress memory operation and any outstanding SDRAM refresh cycle. If the PXA27x processor starts a swap operation, the processor does not begin the alternate bus master mode grant sequence until the operation is complete.

The processor then de-asserts SDCKE and three-states all memory bus pins used with SDRAM bank 0 (nSDCS0, MA<25:0>, nOE, nWE, nSDRAS, nSDCAS, SDCLK1, MD<31:0>, DQM<3:0>). All other memory and PC Card pins remain driven. The RDnWR is also three-stated allowing the alternate bus master to control any transceiver logic that might exist in the design. The RDnWR pin must still be driven by the alternate bus master:

- If no transceiver logic exist between the processor and the SDRAM
- If the RDnWR is used by any devices within the design to prevent any inputs from floating

For the SA-1110 address compatibility mode, the nOE signal is driven after control is passed to the alternate bus master. To prevent inputs that receive the nOE signal from floating or possible contention on the data bus, the nOE signal must be driven high by the alternate bus master while possessing ownership of the bus.

After that, the PXA27x processor asserts MBGNT, the alternate master must start driving all pins (including SDCLK<1>), and the PXA27x processor must re-assert SDCKE. The grant sequence and timing are (the Tmem unit of time is the memory clock period):

1. Alternate master asserts MBREQ.
2. The PXA27x processor memory controller performs an SDRAM refresh if SDRAM clocks and clock enable are turned on.
3. The PXA27x processor memory controller sends an MRS command to the SDRAMs if the *MDCNFG[SA1110_x]* bit is turned on to change the SDRAM burst length to 1 instead of 4. The burst length is changed to 1 for the SA-1110 address compatibility mode.
4. The PXA27x processor de-asserts SDCKE at time (t).
5. The PXA27x processor three-states SDRAM outputs at time (t + 1 x Tmem).
6. The PXA27x processor asserts MBGNT at time (t + 2 x Tmem).
7. Alternate master drives SDRAM signals prior to time (t + 3 x Tmem).

8. The PXA27x processor asserts SDCKE at time (t + 4 x Tmem).

During the three-state period, both MBREQ and MBGNT remain high and an external device must assume control of the three-stated pins. The external device must drive all the three-stated pins even if some are not actually used. Otherwise, floating inputs causes excessive crossover current or erroneous SDRAM commands.

Note that during the three-state period, the PXA27x processor cannot perform SDRAM refresh cycles. The SDRAM memory controller of the PXA27x processor issues a CBR (auto refresh) command prior to releasing control of the bus. If the system is populated with SRAM in partition 0, the alternate master must assume the responsibility for SDRAM integrity during this period. Otherwise, design the system such that the period of alternate mastership is limited to much less than the refresh period, or that the alternate master implement a refresh counter enabling it to perform refreshes at the proper intervals. In other words, the alternate bus master must release control from the bus before the next CBR command is due. This is 7.8 µs for SDRAM with 13 row address lines, 15.6 µs for SDRAM with 12 row address lines, 31.0 µs for SDRAM with 12 row address lines.

To give up ownership of the bus, perform the procedure according to the release sequence and timing:

1. Alternate master de-asserts MBREQ.
2. The PXA27x processor de-asserts SDCKE at time (t).
3. The PXA27x processor de-asserts MBGNT at time (t + 1 x Tmem).
4. Alternate master three-states SDRAM outputs prior to time (t + 2 x Tmem).
5. The PXA27x processor drives SDRAM outputs at time (t + 3 x Tmem).
6. The PXA27x processor asserts SDCKE at time (t + 4 x Tmem).
7. The PXA27x processor memory controller performs an SDRAM refresh if SDRAM clocks and clock enable are turned on.
8. The PXA27x processor memory controller sends an MRS command to the SDRAMs if the *MDCNFG[SA1110_x]* bit is turned on. This is done to change the SDRAM burst length back to 4 instead of 1.

Alternate bus master mode is set up by writing these registers:

- Write the GPIO Pin Direction register (GPDR_x) to set the bit corresponding to MBGNT as an output and clear the bit corresponding to MBREQ an input.
- Write the GPIO Alternate Function register (GAFR0_x) to set the bits that map the alternate functions on the specified GPIO pins to the Alternate Bus Master mode operation.

6.5.7.1 Alternate Bus Master Signals

See Table 6-14 for the list of signals required to interface to an alternate bus master device.

Table 6-14. Alternate Bus Master Interface Signals

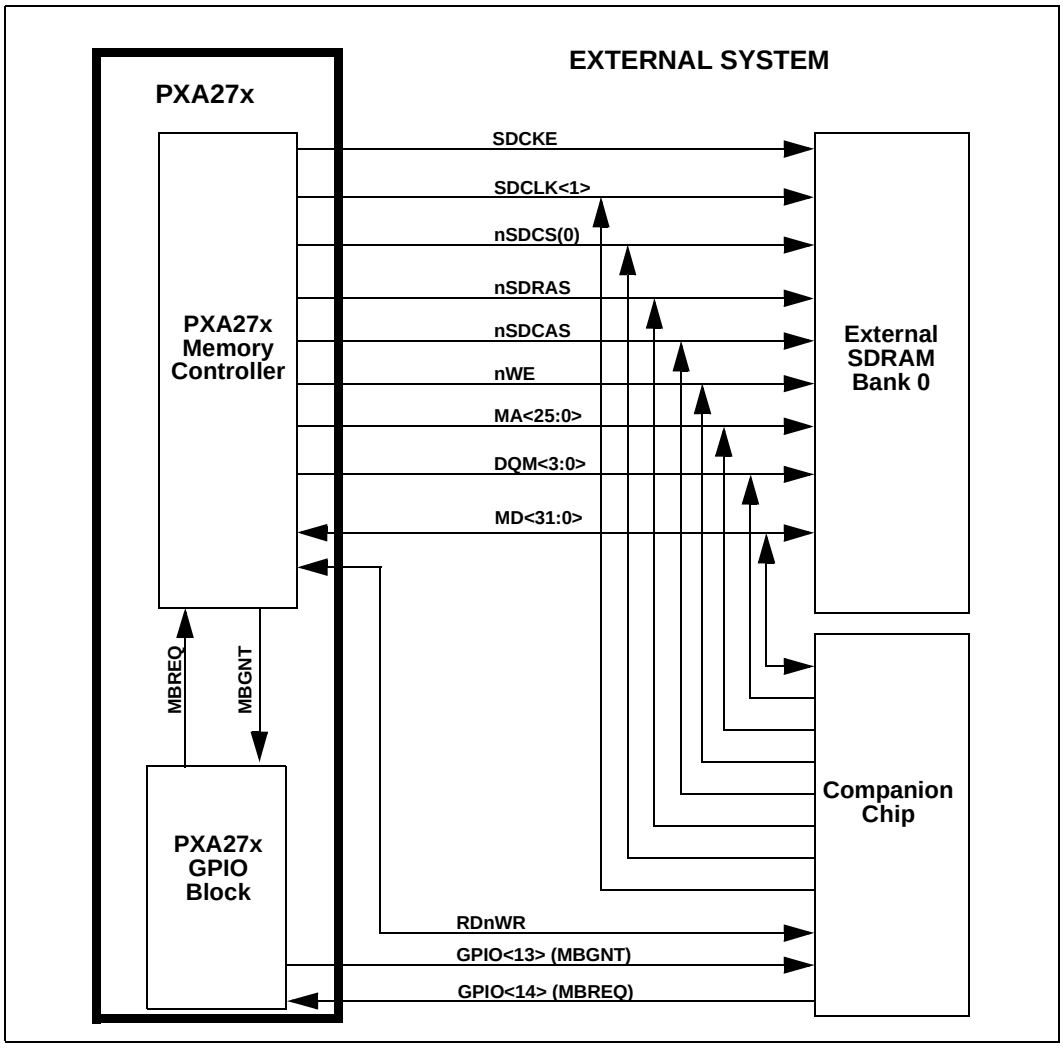
Signal Name	Direction	Polarity	Description
Alternate Bus Master Interface Signals			
MBREQ	Input	Active High	Alternate bus master request
MBGNT	Output	Active High	Alternate bus master grant
SDCKE	Output	Active High	Output clock enable signals for external memory SDCKE is for all SDRAM memory partitions
Signals Three-Stated During Alternate Bus Master Ownership			
SDCLK1	Output	High-Z	SDCLK1 is for SDRAM partitions 0 and 1
nSDCS0	Output	HIgh-Z	Chips select for SDRAM partition 0
MA<25:1>	Output	High-Z	Output address to all memory types **NOTE:** Do not use MA0 because all alternate bus master devices must have a minimum bus width of 16 bits when interfacing to the PXA27x processor memory controller.
MD<31:0>	Bidirectional	High-Z	Bidirectional data for all memory types
DQM<3:0>	Output	High-Z	Data byte mask control DQM<0> corresponds to MD<7:0> DQM<1> corresponds to MD<15:8> DQM<2> corresponds to MD<23:16> DQM<3> corresponds to MD<31:24> 0 = Do not mask out corresponding byte 1 = Mask out corresponding byte
nSDRAS	Output	High-Z	Row address for SDRAM
nSDCAS	Output	High-Z	Column Strobe for SDRAM Also, nADV (address strobe) for synchronous flash
nSDCS<3:0>	Output	High-Z	Chips selects for SDRAM
nCS<5:0>	Output	High-Z	Chip selects for static memory
nWE	Output	High-Z	Write enable for SDRAM and static memory
nOE	Output	High-Z	Output enable for static memory **NOTE:** This signal is three-stated to be compatible with the Intel SA-1110 and must be driven by the alternate bus master during alternate bus master ownership
RDnWR	Output	High-Z	Data direction signal to be used by output transceivers 0 = MD<31:0> is driven by the PXA27x processor 1 = MD<31:0> is not driven by the PXA27x processor

Note: All other memory and PC Card signals remain driven during alternate bus master mode.

6.5.7.2 Alternate Bus Master Block Diagram

See Figure 6-12 for illustration of the connections.

Figure 6-12. Alternate Bus Master Mode



6.5.7.3 Alternate Bus Master Layout Notes

Refer to Part II: Section 6.5.1.3, "SDRAM Layout Notes," for recommendation on trace lengths, size, and routing guidelines. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* documents for AC timing information.

§§

LCD Interface 7

This chapter describes examples of hardware connections between Intel® PXA27x Processor Family (PXA27x processor) and various types of Liquid Crystal Display (LCD) controllers. Active LCD displays, such as the Thin Film Transistor (TFT) display and passive LCD displays such as the Supertwist Nematic (STN) display, are discussed as well as single and dual-scan displays. Smart panels are relatively new devices in the LCD panel market and are yet to have standardized connections. An example connection for a smart panel is provided; however, the example requires verification of the connectivity of the LCD panel with the panel vendor.

The diagrams in this chapter serve as guidelines in connecting LCD panel to the PXA27x processor. These diagrams are intended to assist in system hardware design.

7.1 Overview

The PXA27x processor supports both active and passive LCD displays. Active displays generally produce better looking images, but at a higher cost. Passive displays are generally less expensive, but their displays are inferior to active displays. However, recent advances in dithering technology are closing the quality gap between passive and active displays.

Several different types of passive monochrome displays are available. These passive displays are either single or dual-scan displays. Additionally, some monochrome displays use double-pixel data mode (twice the number of pixels as a normal monochrome display). With the exception of the number of data pins required, all of these choices affect the software configuration and support, not the system hardware design. In fact, most passive displays use a single interconnection scheme. For information on the software changes and performance considerations of the various display options, refer to the *Intel® PXA27x Processor Family Developers Manual*.

Passive displays drive dithered data to the LCD panel, which means that for each pixel clock cycle, a single data line drives an ON/OFF signal for one color of a single pixel.

Active panels have fixed data pins assignments. For instance, LDD<0> is always the least significant byte (LSB) of the blue data, except for 24 bits per pixel (bpp) panels. The color depth determines the number of pins used. All 16 bpp active displays require 16 data lines. All 18 bpp active displays require 18 data lines. Only with 24 bpp does the PXA27x processor use a different number of pins than the color depth; the processor outputs the data on 8 lines.

Active displays do not dither the data to the LCD panel. The data driven is the direct intensity level for the pixel being transferred.

Smart panels offer greater flexibility in their use and also offer much greater power savings. To obtain the power savings requires additional communications between the LCD controller and the panel. Communications are attained by use of either additional signals or expanded use of existing signals on the processor. The particular implementation of smart panels is not yet standardized and is considered as customized implementation of each panel. While an example connection for this type of display is provided in this chapter, verify the proper connection of the LCD panel per recommendation of the panel vendor.

The PXA27x processor supports these types of panels:

- Single- or dual-scan display modules
- Up to 256 gray-scale levels (8 bits) in passive monochrome mode
- Up to 16777216 colors (24 bits) in active color mode
- A total of 16777216 colors (24 bits) in passive color mode
- LCD panels with an internal frame buffer (smart panels)
- Up to 8-bit (each) passive dual-scan color displays
- Up to 18-bit per pixel for active single-panel color displays
- Up to 24-bit per pixel single-panel color smart panels

7.2 Signals

See Table 7-1 for the list of LCD interface signals of the PXA27x processor.

Table 7-1. 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 to as input data bus during reads to the smart panels.
L_PCLK_WR	Output	Pixel clock used by the LCD display module to clock the pixel data into the panel. Write signal for writing to smart panels.
L_LCLK_A0	Output	For passive (STN) displays, the line clock used by the LCD display module signals the end of a line of pixels. Used by active (TFT) displays as the horizontal synchronization signal. This control signal specifies command or data transactions when interfacing with smart panels.
L_FCLK_RD	Output	For passive (STN) displays, the frame clock used by the LCD display module signals the start of a new frame of pixels. Used by active (TFT) display module as the vertical synchronization signal. Read signal during reads to the smart panels.
L_BIAS	Output	For passive (STN) displays, the AC bias signals 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, the AC bias is used as the output-enable to signal when data is latched from the data pins using the pixel clock.
L_CS	Output	For smart panels, this pin is used as chip-select signal. No connect for active (TFT) and passive (STN) panels.
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

The selection of the appropriate data pins for passive displays is complicated due to the multiple driving schemes possible. Active displays do not have such characteristics as dual-scan or double-pixel mode and, therefore, have much more straight forward selection of the data pins. See Table 7-2 for description of the number of LDD pins required for the various types of passive displays and the use of LCD data pins for specific panels, upper or lower.

Table 7-2. LCD Controller Data Pin Utilization

Color/ Monochrome Panel	Single/ Dual-Scan	Double-Pixel Mode	Screen Portion	Pins
Monochrome	Single	No	Whole	LDD<3:0>
Monochrome	Single	Yes	Whole	LDD<7:0>†
Monochrome	Dual	No	Top	LDD<3:0>
			Bottom	LDD<7:4>
Color	Single	N/A	Whole	LDD<7:0>
Color	Dual	N/A	Top	LDD<7:0>
			Bottom	LDD<15:8>

† Double pixel data mode (LCCR0[DPD]=1)

7.3 Schematics/Block Diagram

Refer to Section 7.5 for the appropriate block diagram for the system under consideration.

7.4 Layout Notes

The following design considerations are common to all modes of operation and must be properly accounted for in any system design in order to have optimal system performance.

7.4.1 Contrast Voltage

Many displays, both active and passive, include a pin for adjusting the display contrast voltage. This is a variable analog voltage that is supplied to the panel using a voltage source on the system board. The contrast voltage is adjusted using a variable resistor on the circuit board.

The required voltage range and current capabilities vary among panel manufacturers. Consult the data sheet for the LCD panel to determine the variable voltage circuit design. Ensure the contrast voltage is stable; unstable voltage causes visual artifacts. Possible contrast-voltage circuits are often suggested by panel manufacturers.

7.4.2 Backlight Inverter

One potential source of noise for the LCD panel is the backlight inverter. Since the backlight inverter is a high voltage device with frequent voltage inversions, it has the potential to induce spurious noise in the LCD panel lines. To minimize noise:

- Use a shielded backlight inverter.
- Physically place the inverter as far away as possible from the LCD data lines and system board that are usually located near the LCD panel.

If power consumption is an issue, choose a backlight inverter that disables through software. This saves power by automatically disabling the backlight if no activity occurs within a preset period of time.

7.4.3 Signal Routing and Buffering

Signal transmission rates between the LCD controller and the LCD panel are moderate, which helps to simplify the design of the LCD system. The minimum pixel clock divider (PCD) value results in a pixel clock rate of one half of the LCLK (this is not the L_LCLK of the LCD controller.) The maximum LCLK for the PXA27x processor is 104 MHz, resulting in a maximum pixel clock rate of 52 MHz. Therefore, use of 100 MHz design considerations are sufficient to ensure LCD panel signal integrity.

However, typical transfer rates are considerably less than 52 MHz. For example, a 800x600 color active display running at 75 Hz requires a transfer rate of approximately 34 MHz. To determine the transfer rate, calculate the number of pixels (800 x 600 = 480,000) and multiply by the screen refresh rate (75 Hz). Since active panels less than 24 bpp in color depth replace 1 pixel of data with every clock cycle, this determines the final transfer rate. Active displays normally do not require refresh rates as high as 75 Hz, so use a lower refresh rate to reduce transmission rates even more.

Panels with a color depth of 24 bpp only transfer one third of a pixel per clock cycle. These result in an LCD clock rate triple that of active panels of the same size.

Passive displays often require refresh rates greater than 75 Hz, but these displays transfer more pixels for each clock cycle. For instance, a color passive display with eight data lines transfers 2 2/3 pixels worth of data each clock cycle. This divides the transmission rate by 2 2/3. Further reductions in the transfer rate come by using dual-scan displays that use twice as many data lines to transfer data – halving the rate again.

Generally, this gives lower transfer rates to even larger displays and thus simpler design considerations and fewer layout constraints.

When laying out your design, minimize trace length of the LCD panel signals and allow sufficient spacing between signals to avoid crosstalk. Crosstalk decreases the signal integrity, especially the data line signals.

LCD system design is not considered critical as infrequent or single bit errors are, typically, not noticed by the user. Also, the errors are transitory, as the old data is constantly being replaced with new data. Slower panel refresh rates increase the likelihood that a single error is noticed by the user. However, there is a counteracting effect in that slower refresh rates relax LCD timing and, therefore, result in fewer screen transmission errors. This is not the case for panels with internal frame buffers as the data is transmitted to and then stored by the panel. This results in single bit errors remaining until the screen changes and the incorrect bit being replaced. There are other factors related to choosing a refresh rate for an LCD system, most significant is the impact on system bandwidth. Refer to the section on “Bandwidth Calculations” in the *Intel® PXA27x Processor Family Developers Manual* for more information.

If excessively long or poorly routed signals are used, one possible solution is to add buffers between the PXA27x processor and the LCD panel. This helps strengthen the LCD panel signal levels and synchronizes signal timing. However, this is usually not required as the LCD panel timings are fairly relaxed. Since the LCD display essentially operates asynchronously from the processor, the propagation delay of the buffers is not a significant concern.

7.4.4 Panel Connector

Most LCD panels are connected to the system board using a connector, instead of directly mounted on the system board. Connecting the LCD panel to the system board using a connector increases flexibility and makes it easier for the manufacturer. Typically, the manufacturer of the panel recommends a particular connector for the panel. Follow the recommendations of the manufacturer.

7.5 Modes of Operation Overview

7.5.1 Passive Monochrome Single-Scan Mode

Passive monochrome single-scan displays are the simplest displays to set up. A maximum gray-scale range of 256 shades (8 bits) is supported.

7.5.1.1 Signals

For passive monochrome single-scan displays, the pins described in Table 7-3 are required for connections between the PXA27x processor and LCD panel.

Table 7-3. Passive Display Pins Required

PXA27x processor Pin	LCD Panel Pin	PIn Type[1]	Definition
LDD<3:0>	D<3:0>	Output	Data lines used to transmit four data values at a time to the LCD display. Each pin value represents a single pixel.
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 that transfers the line data from the shift register to the screen and increment the line pointers.
L_FCLK_RD	Frame_Clock	Output	Frame Clock – used by the LCD displays to signal the start of a new frame of pixels that resets the line pointers to the top of the screen.
L_BIAS	Bias	Output	AC bias used to signal the LCD display to 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 available on PXA27x processor).

NOTES:
1. “Pin Type” is in reference to the PXA27x processor. Therefore, outputs are pins that drive a signal from the processor to another device.
2. Vcon is a signal external to the PXA27x processor. Refer to Section 7.4.1, “Contrast Voltage,” for detailed information.

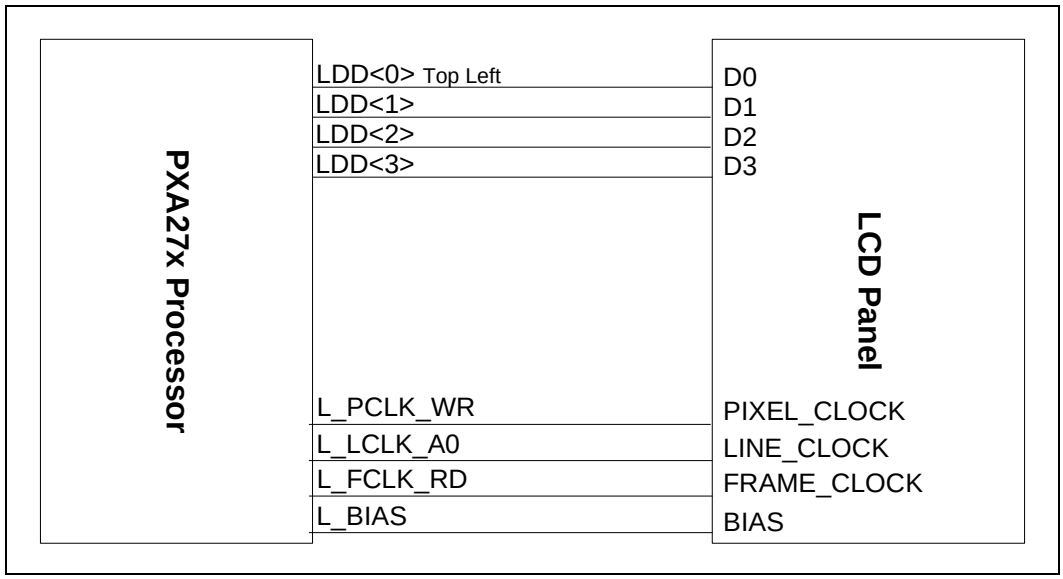
Note: Names used for “LCD Panel Pin” are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on:

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

7.5.1.2 Schematics/Block Diagram

See Figure 7-1 for illustration of typical connections for a passive monochrome single-scan display. The sample connections serve as a guide for designing systems with such passive displays. Panels differ on which is the panel's least significant bit (refer to the LCD panel reference documentation for the least significant bit.) The figure indicates the top-left pixel (1,1) bit. While dual-scan panels indicates the top-left pixel (1,n/2) of the upper and lower panels and color passive panels show the top-left-pixel color bits.

Figure 7-1. Passive Monochrome Single-Scan Display Typical Connection



7.5.1.3 Layout Notes

Refer to Section 7.4, "Layout Notes," for layout notes and considerations.

7.5.2 Passive Monochrome Single-Scan Double-Pixel Mode

Passive monochrome, single-scan, double-pixel mode displays are identical to passive monochrome, single-scan displays except that the displays transmit twice the number of pixels per clock. This is accomplished through the use of twice as many data lines. All other signals are identical.

7.5.2.1 Signals

For passive monochrome single-scan, double-pixel mode displays, see Table 7-4 for description of the pins required for connections between the PXA27x processor and LCD panel.

Table 7-4. Passive Display Pins Required

PXA27x processor Pin	LCD Panel Pin	PIn Type[1]	Definition
LDD<7:0>	D<7:0>x	Output	Data lines used to transmit eight data values at a time to the LCD display. Each pin value represents a single pixel. The panel is driven as if it is two separate panels, each with half the number of lines.
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 that transfers the line data from the shift register to the screen and increment the line pointers.
L_FCLK_RD	Frame_Clock	Output	Frame Clock – used by the LCD displays to signal the start of a new frame of pixels that resets the line pointers to the top of the screen.
L_BIAS	Bias	Output	AC bias used to signal the LCD display to 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 available on PXA27x processor).

NOTES:
1. “Pin Type” is in reference to the PXA27x processor. Therefore, outputs are pins that drive a signal from the processor to another device.
2. Vcon is a signal external to the PXA27x processor. Refer to Section 7.4.1, “Contrast Voltage,” for detailed information.

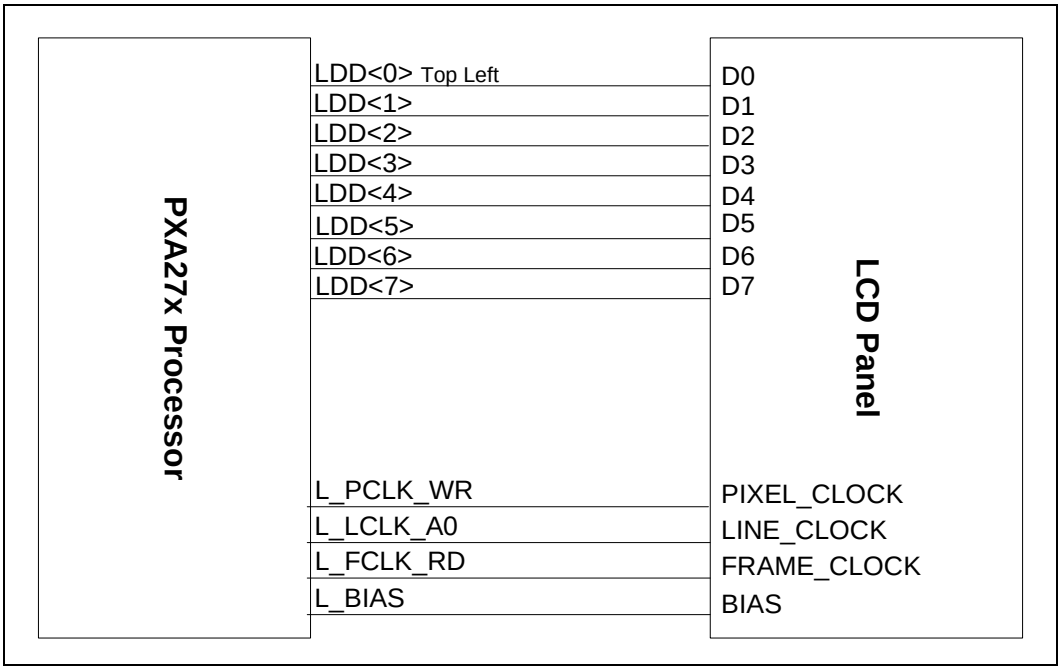
Note: Names used for “LCD Panel Pin” are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on:

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

7.5.2.2 Schematics / Block Diagram

See Figure 7-2 for illustration of typical connections for a single-scan, monochrome passive display using double-pixel data mode.

Figure 7-2. Passive Monochrome Single-Scan Double-Pixel Data Display Typical Connection



7.5.2.3 Layout Notes

Refer to Section 7.4, “Layout Notes,” for layout notes and considerations.

7.5.3 Passive Monochrome Dual-Scan Mode

Passive monochrome, dual-scan panels drive eight data values per clock cycle. These displays are connected and programmed as if they are two separate panels, each with half the numbers of lines.

7.5.3.1 Signals

For passive monochrome, dual-scan displays, see Table 7-5 for description of the pins required for connections between the PXA27x processor and LCD panel.

Table 7-5. Passive Display Pins Required (Sheet 1 of 2)

PXA27x processor Pin	LCD Panel Pin	PIn Type[1]	Definition
LDD<7:0>	DL<3:0>, DU<3:0>	Output	Data lines used to transmit eight data values at a time to the LCD display. Each pin value represents a single pixel. The panel is driven as if it is two separate panels, each panel with half the number of lines.
L_PCLK_WR	Pixel_Clock	Output	Pixel Clock – used by the LCD display to clock the pixel data into the line shift register.

Table 7-5. Passive Display Pins Required (Sheet 2 of 2)

L_LCLK_A0	Line_Clock	Output	Line Clock – used by the LCD display to signal the end of a line of pixels that transfers the line data from the shift register to the screen and increment the line pointers.
L_FCLK_RD	Frame_Clock	Output	Frame Clock – used by the LCD displays to signal the start of a new frame of pixels that resets the line pointers to the top of the screen.
L_BIAS	Bias	Output	AC bias used to signal the LCD display to 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 available on PXA27x processor).

NOTES:
1. "Pin Type" is in reference to the PXA27x processor. Therefore, outputs are pins that drive a signal from the processor to another device.
2. Vcon is a signal external to the PXA27x processor. Please refer to Section 7.4.1, "Contrast Voltage," for detailed information.

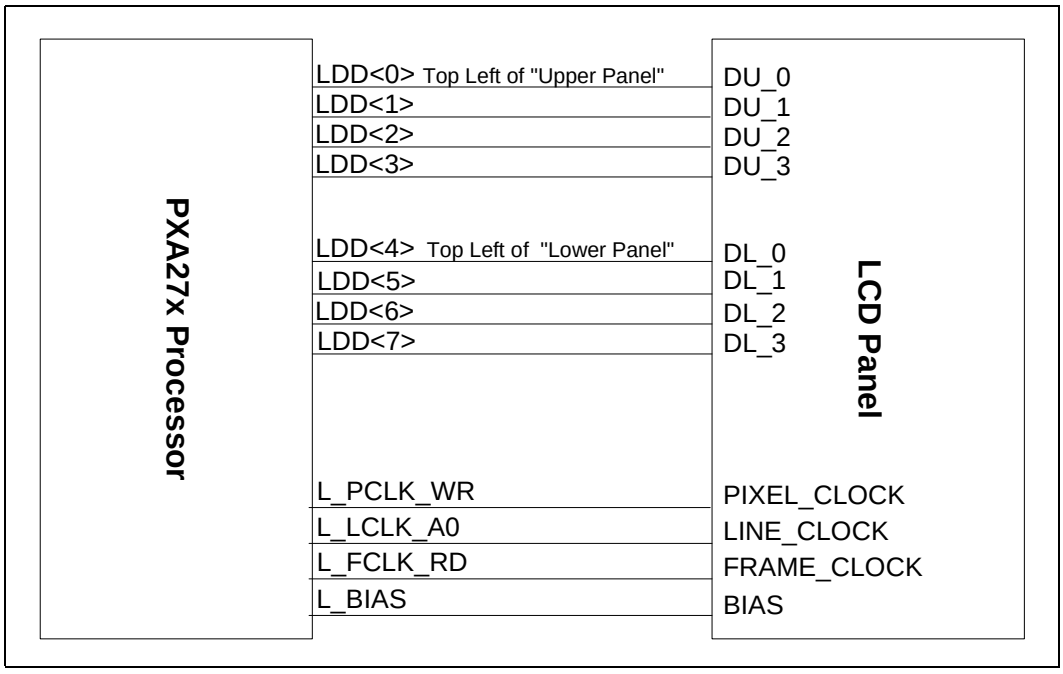
Note: Names used for "LCD Panel Pin" are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on:

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

7.5.3.2 Schematics / Block Diagram

See Figure 7-3 for illustration of typical connections for a dual-scan monochrome, passive display.

Figure 7-3. Passive Monochrome Dual-Scan Display Typical Connection



7.5.3.3 Layout Notes

Refer to Section 7.4, “Layout Notes,” for layout notes and considerations.

7.5.4 Passive Color Single-Scan Mode

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

7.5.4.1 Signals

For passive color displays, see Table 7-3 for description of the pins required for connections between the PXA27x processor and the LCD panel.

Table 7-6. Passive Display Pins Required

PXA27x processor Pin	LCD Panel Pin	PIn Type[1]	Definition
LDD<7:0>	D<7:0>	Output	Data lines used to transmit 2 2/3data 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 that transfers the line data from the shift register to the screen and increment the line pointers.
L_FCLK_RD	Frame_Clock	Output	Frame Clock – used by the LCD displays to signal the start of a new frame of pixels that resets the line pointers to the top of the screen.
L_BIAS	Bias	Output	AC bias used to signal the LCD display to 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 available on PXA27x processor).

NOTES:
1. “Pin Type” is in reference to the PXA27x processor. Therefore, outputs are pins that drive a signal from the processor to another device.
2. Vcon is a signal external to the PXA27x processor. Refer to Section 7.4.1, “Contrast Voltage,” for detailed information.

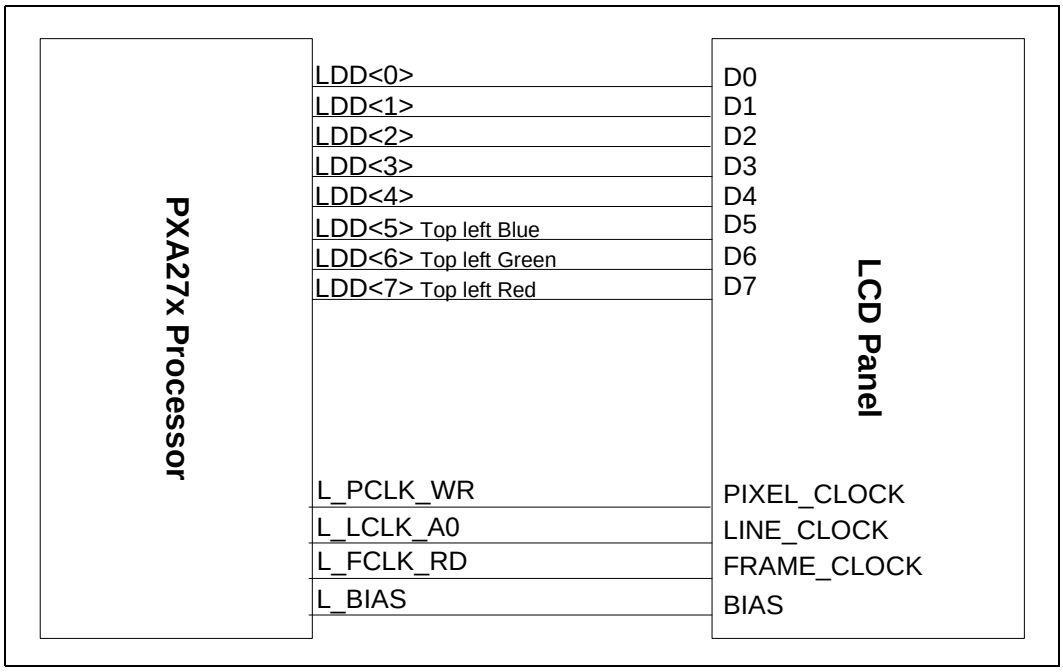
Note: Names used for “LCD Panel Pin” are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on:

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

7.5.4.2 Schematics/Block Diagram

See Figure 7-4 for illustration of typical connections for a single-scan color passive display.

Figure 7-4. Passive Color Single-Scan Display Typical Connection



7.5.4.3 Layout Notes

Refer to Section 7.4, “Layout Notes,” for layout notes and considerations.

7.5.5 Passive Color Dual-Scan Mode

Passive color dual-scan panels drive 5 1/3 data values per clock cycle. These panels are connected and programmed as if they are two separate panels, each with half the numbers of lines.

7.5.5.1 Signals

For passive color dual-scan displays, see Table 7-7 for description of the pins required for connections between the PXA27x processor and LCD panel.

Table 7-7. Passive Display Pins Required

PXA27x processor Pin	LCD Panel Pin	PIn Type[1]	Definition
LDD<15:0>	DU<7:0>, DL<7:0>	Output	Data lines used to transmit 5 1/3data values at a time to the LCD display. Groupings of three pin values represent one pixel (red, green, and blue data values). The panel is driven as if it is two separate panels, each panel with half the number of lines.
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 that transfers the line data from the shift register to the screen and increment the line pointers.
L_FCLK_RD	Frame_Clock	Output	Frame Clock – used by the LCD displays to signal the start of a new frame of pixels that resets the line pointers to the top of the screen.
L_BIAS	Bias	Output	AC bias used to signal the LCD display to 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 available on PXA27x processor).

NOTES:
1. “Pin Type” is in reference to the PXA27x processor. Therefore, outputs are pins that drive a signal from the processor to another device.
2. Vcon is a signal external to the PXA27x processor. Refer to Section 7.4.1, “Contrast Voltage,” for detailed information.

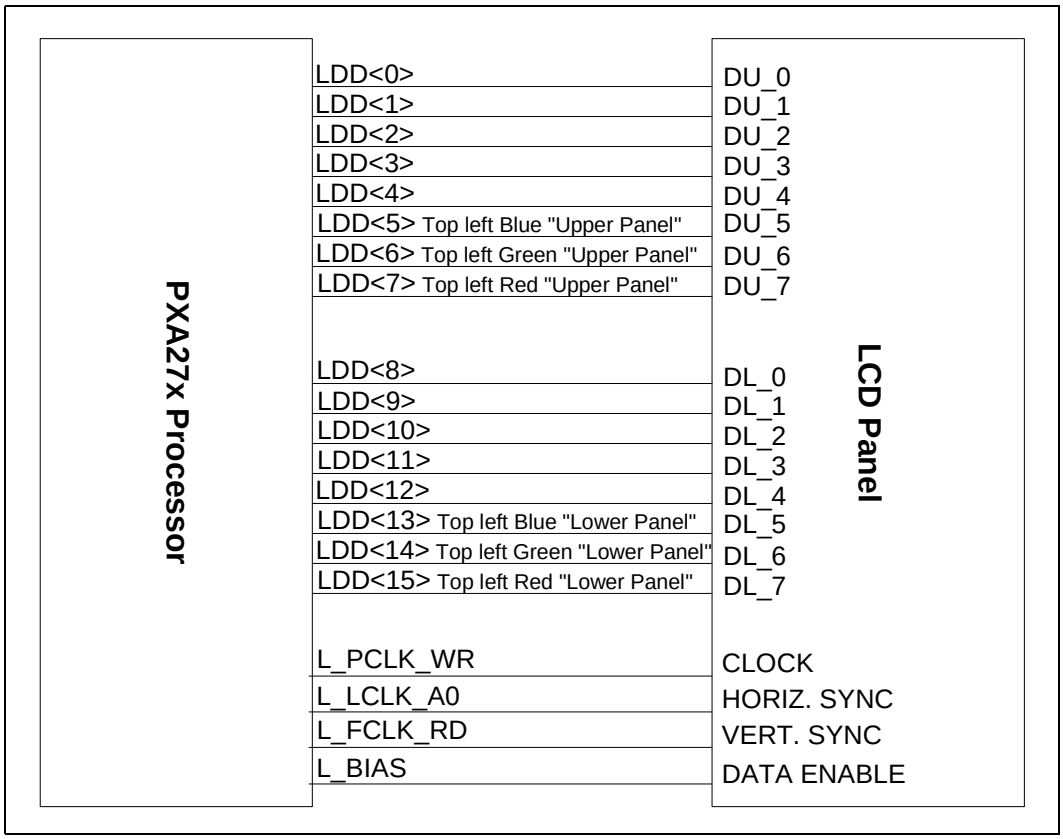
Note: Names used for “LCD Panel Pin” are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on:

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

7.5.5.2 Schematics / Block Diagram

See Figure 7-5 for illustration of typical connections for a dual-scan color passive display.

Figure 7-5. Passive Color Dual-Scan Display Typical Connection



7.5.5.3 Layout Notes

Refer to Section 7.4, “Layout Notes,” for layout notes and considerations.

7.5.6 Active Color 12-bit per pixel Mode

An active color display does not send dithered data to the panel. The lines driven represent the digital value of the pixel being transmitted. A single pixel is transmitted per clock cycle. The bits of data describe the intensity level of the red, green and blue for each pixel. 12 bpp mode is a mode that is not directly supported by the PXA27x processor. However, 12 bpp panels are supported by programming and operating the PXA27x processor in 16 bpp, but only connecting a subset of the data lines.

7.5.6.1 Signals

The signals described in Table 7-8 implement an active color 12-bit per pixel display with the PXA27x processor.

Note: The software must be enabled for an active 16-bit per pixel panel.

Table 7-8. LCD Interface Signal List

PXA27x Processor Signal Name	LCD Panel Signal Name	Type	Description
LDD<15:12>, LDD<10:7>, LDD<4:1>	R<3:0>,G<3:0>, B<3:0>	Bidirectional	Data lines used to transmit data values to the LCD display module.
L_PCLK_WR	CLOCK	Output	Pixel clock used by the LCD display module to clock the pixel data into the panel.
L_LCLK_A0	HORIZONTAL SYNC.	Output	Used by active (TFT) display module as the horizontal synchronization signal.
L_FCLK_RD	VERTICAL SYNC.	Output	Used by active (TFT) display module as the vertical synchronization signal.
L_BIAS	DATA ENABLE	Output	Used as the output-enable to signal when data is latched from the data pins using the pixel clock.

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:

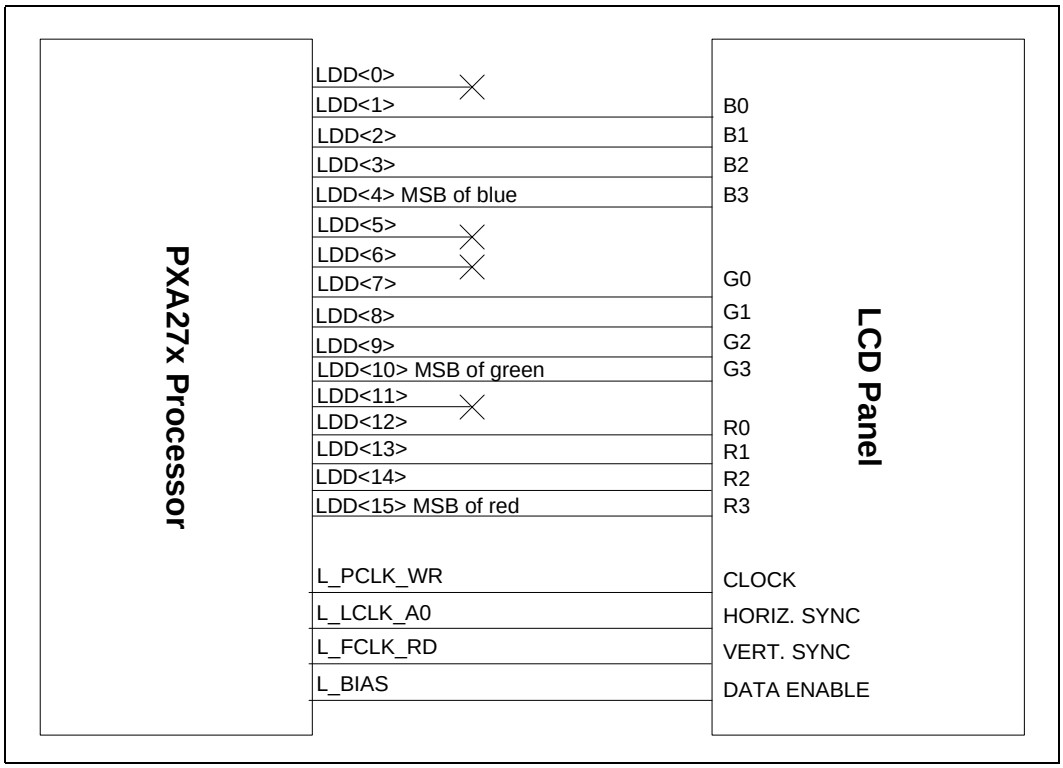
- Specific signals required for correct LCD operation
- Correct names of the signals used by the LCD panel manufacturer

7.5.6.2 Schematics / Block Diagram

See Figure 7-6 for illustration of typical connections for a 12 bpp active panel display. The sample connections serve as a guide for designing systems that contain active LCD displays. The most significant byte (MSB) of each color is indicated.

The sample below shows four red, four green, and four blue bits on the LCD panel. However, different active display panels might have more or different data lines. Consult the LCD panel manufacturer’s documentation for the actual data lines.

Figure 7-6. Active Color 12-bit per pixel Display Typical Connection



7.5.6.3 Layout Notes

Refer to Section 7.4, “Layout Notes,” for layout notes and considerations.

7.5.7 Active Color, 16-bit per pixel Mode

An active color display does not send dithered data to the panel. The lines driven represent the digital value of the pixel being transmitted. A single pixel is transmitted per clock cycle. The 16 bits of data describe the intensity level of red, green, and blue for each pixel. Typically, the 16-bit data is formatted as five bits for red, six bits for green, and five bits for blue, but the format varies by display and is controlled by the software writing to the frame buffer. Refer to the display documentation to ensure the correct PXA27x processor LCD data lines are connected to the correct LCD panel data lines.

Note: The LCD controller of the PXA27x processor 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-bit per pixel operation impacts the way in which signals are connected between the PXA27x processor 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 Section 7.5, “LCD Control Registers,” of the *Intel® PXA27x Processor Family Developers Manual* for more information.

7.5.7.1 Signals

The signals in Table 7-9 implement an active color 16-bit-per-pixel display with the PXA27x processor.

Table 7-9. LCD Interface Signal List

PXA27x Processor Signal Name	LCD Panel Signal Name	Type	Description
LDD<15:0>	R<4:0>,G<5:0>, B<4:0>	Bidirectional	Data lines used to transmit data values to the LCD display module.
L_PCLK_WR	CLOCK	Output	Pixel clock used by the LCD display module to clock the pixel data into the panel.
L_LCLK_A0	HORIZONTAL SYNC.	Output	Used by active (TFT) display module as the horizontal synchronization signal.
L_FCLK_RD	VERTICAL SYNC.	Output	Used by active (TFT) display module as the vertical synchronization signal.
L_BIAS	DATA ENABLE	Output	Used as the output-enable to signal when data is latched from the data pins using the pixel clock.

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:

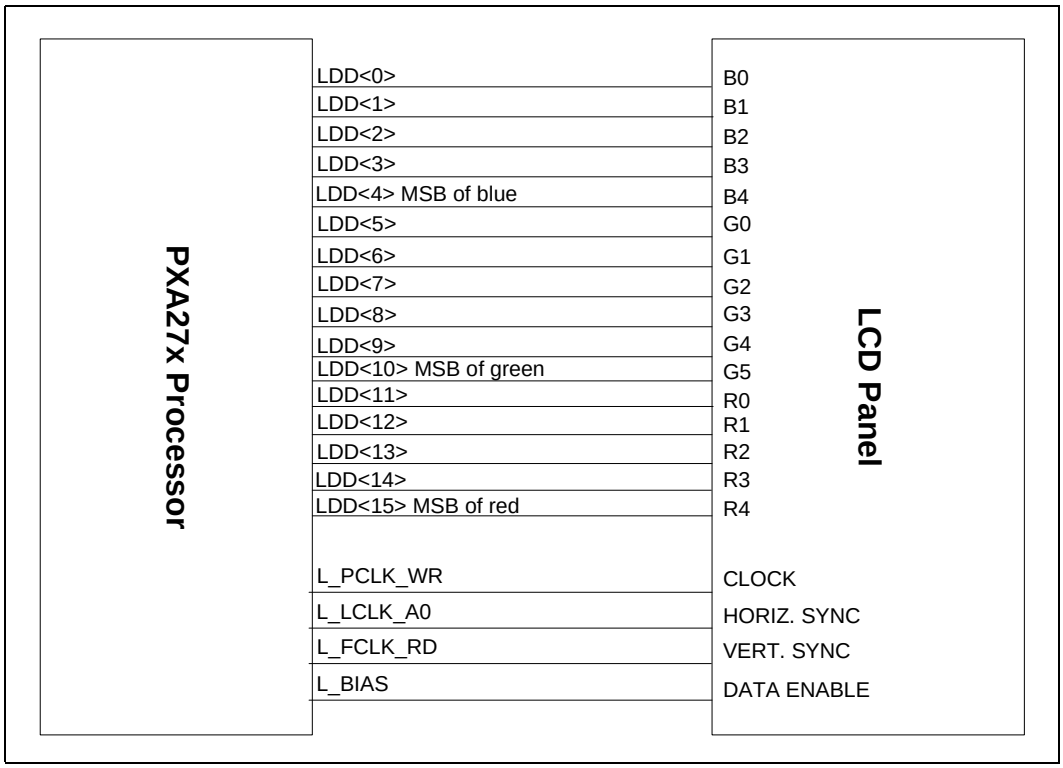
- Specific signals required for correct LCD operation
- Correct names of the signals used by the LCD panel manufacturer

7.5.7.2 Schematics / Block Diagram

See Figure 7-7 for 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 MSB of each color is indicated.

The example below shows five red, six green, and five blue bits on the LCD panel. However, different active display panels might have more or different data lines. Consult the LCD panel manufacturer’s documentation for the actual data lines.

Figure 7-7. Active Color 16-bit-per-pixel Display Typical Connection



7.5.7.3 Layout Notes

Refer to Section 7.4, “Layout Notes,” for layout notes and considerations.

7.5.8 Active Color, 18-bit per pixel Mode

An active color display does not send dithered data to the panel. The lines driven represent the digital value of the pixel being transmitted. A single pixel is transmitted per clock cycle. The bits of data describe the intensity level of the red, green, and blue for each pixel.

Note: If the system design incorporates PCMCIA interface, LCD and MSL (Baseband Interface), refer to Part II, Section 16.1, “Overview,” for important information on using these interfaces simultaneously.

7.5.8.1 Signals

The signals in Table 7-10 implement an active color 18-bit per pixel display with the PXA27x application.

Table 7-10. LCD Interface Signal List

PXA27x Processor Signal Name	LCD Panel Signal Name	Type	Description
LDD<17:0>	R<5:0>,G<5:0>, B<5:0>	Bidirectional	Data lines used to transmit data values to the LCD display module.
L_PCLK_WR	CLOCK	Output	Pixel clock used by the LCD display module to clock the pixel data into the panel.
L_LCLK_A0	HORIZONTAL SYNC.	Output	Used by active (TFT) display module as the horizontal synchronization signal.
L_FCLK_RD	VERTICAL SYNC.	Output	Used by active (TFT) display module as the vertical synchronization signal.
L_BIAS	DATA ENABLE	Output	Used as the output-enable to signal when data is latched from the data pins using the pixel clock.

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:

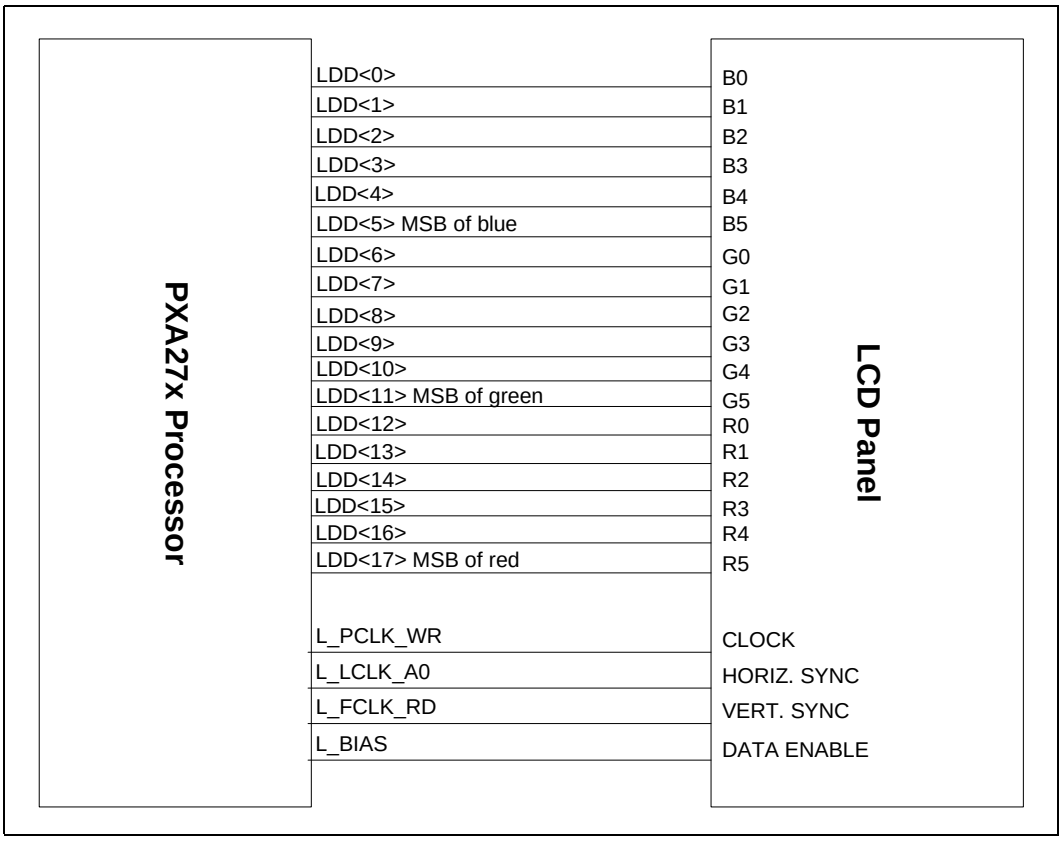
- Specific signals required for correct LCD operation
- Correct names of the signals used by the LCD panel manufacturer

7.5.8.2 Schematics / Block Diagram

See Figure 7-8 for 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 MSB of each color is indicated.

The example below shows six red, six green, and six blue bits on the LCD panel. However, different active display panels might have more or different data lines. Consult the LCD panel manufacturer’s documentation for the actual data lines.

Figure 7-8. Active Color 18-bit-per pixel Display Typical Connection



7.5.8.3 Layout Notes

Refer to Section 7.4, “Layout Notes,” for layout notes and considerations.

7.5.9 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 panel requires that the panel contain much more sophisticated logic than normal LCD panels. These smart panels are equivalent to simple microcontrollers in capabilities and interface.

Accordingly, the pin interface between the LCD controller on the PXA27x processor and the smart panel is similar to that of a microcontroller interface. The PXA27x processor uses the same physical pins as with a normal active or passive panel but with two additions. See Table 7-11 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 clock cycles transmit one pixel of information to the panel. Refer to the display manufacturer’s documentation for information on how data is transmitted to the display.

7.5.9.1 Signals

For active displays, connect the pins as described in Table 7-11 between the PXA27x processor and LCD panel.

Table 7-11. Active Display Pins Required

PXA27x processor Pin	LCD Panel Pin	PIn Type[1]	Definition
LDD<7:0>	D<7:0>	Output	Data lines used to transmit the data values to the LCD display.
L_PCLK_WR	Write	Output	Write signal for embedded frame-buffer LCD panels.
L_LCLK_A0	Command	Output	This control signal specifies command or data transactions for embedded frame-buffer LCD panels.
L_FCLK_RD	Read	Output	Read signal for smart panels.
L_CS	Select	Output	This pin is used as a chip-select signal.
L_VSYNC	Sync	Input	Refresh sync signal from the LCD panel.
N/A	Vcon[2]	N/A	Contrast voltage – adjustable voltage input to LCD panel – external voltage circuitry is required (no pin available on the PXA27x processor).

NOTES:
1. In reference to the PXA27x processor. Therefore, outputs are pins that drive a signal from the PXA27x processor to another device.
2. Vcon is a signal external to the PXA27x processor. Refer to Section 7.4.1, “Contrast Voltage,” for detailed information.

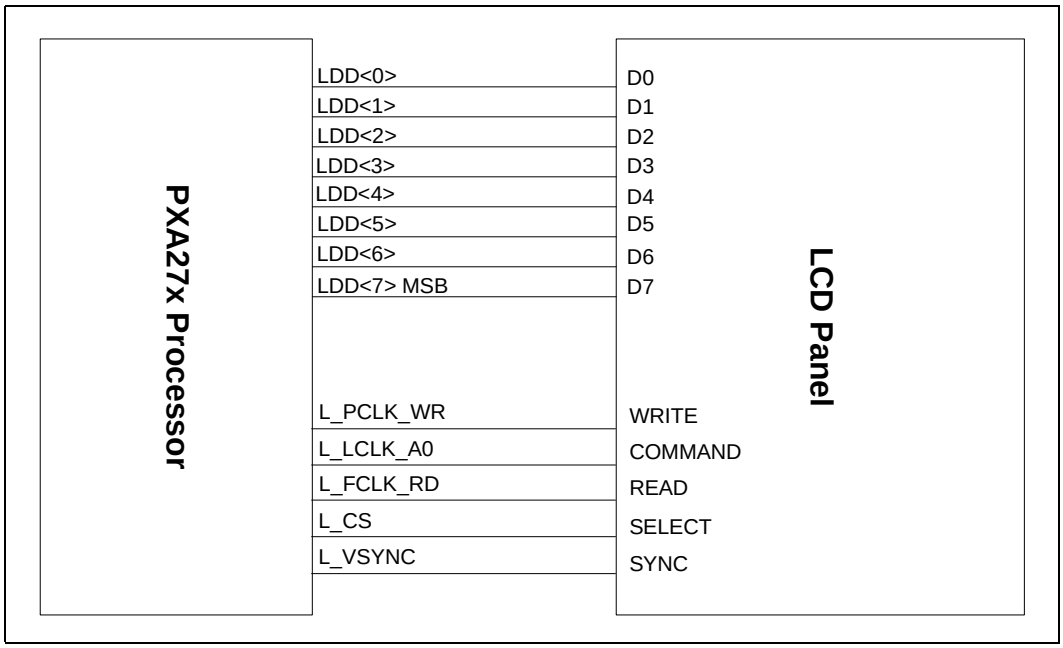
Note: Names used for “LCD Panel Pin” are representative names and do not match those on all LCD panels. Similarly, not all signals are required for all modes of operation. Refer to the LCD panel reference documentation for information on:

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

7.5.9.2 Schematics / Block Diagram

See Figure 7-9 for illustration of typical connections for a smart panel. The sample connections serve as a guide for designing systems that contain LCD displays with embedded frame-buffer memory.

Figure 7-9. Active Color Display 24-bit Typical Connection



7.5.9.3 Layout Notes

Refer to Section 7.4, “Layout Notes,” for layout notes and considerations.

§§

SSP Port Interface 8

8.1 Overview

The Intel® PXA27x Processor (PXA27x processor) contains three highly 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 three SSPs are identical except that SSP3 has no external clock and no external clock enable. Two of the SSPs are configurable in any of the modes described in this chapter; the third is configurable for any mode except external clocking.

Refer to chapter 8 of the *Intel® PXA27x Processor Family Developers Manual* for information on programming and configuring the SSPs. This chapter only describes the physical connections for the SSPs.

The differences between the four possible operational modes (TI* Synchronous Serial Protocol, 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 different operational modes are not discussed any further in this document. Refer to the *Intel® PXA27x Processor Family Developers Manual* for detailed information on the operation.

8.2 Signals

See Table 8-1 for description of the signals associated with each of the three SSPs. Each of the signals for the three SSPs is separately described due to slight differences in some of the signals. These signals are available on the GPIO pins. The GPIO must be correctly enabled in order to have access to the functionality of the SSPs described in Table 8-1. Refer to the GPIO Alternate Function table in the GPIO chapter of the *Intel® PXA27x Processor Family Developers Manual* for the GPIO assignments of the SSP signals.

Table 8-1. SSP Serial Port I/O Signals (Sheet 1 of 2)

Name	Direction	Description
SSPSCLK	Inout	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	SSPSYSCLK is four times the SSPSCLK1 value when using Audio Clock PLL Select (SSACD_1[ACPS]) and Audio Clock Divider (SSACD_1[ACDS]).
SSPSFRM	Inout	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 SSPCR1_1[SFRMDIR].
SSPTXD	Output	SSPTXD is the transmit data (serial data out) serialized data line.
SSPRXD	Input	SSPRXD is the receive data (serial data in) serialized data line.
SSPEXTCLK	Input	SSPEXTCLK is an external clock that is selected to replace the internal 13-MHz clock. SSPEXTCLK is multiplexed with the SSPSCLKEN alternate function (refer to Chapter 24, “General-Purpose I/O Controller”).
SSPSCLKEN	Input	SSPSCLKEN is an asynchronous external enable for SSPSCLK. SSPSCLKEN is multiplexed with the SSPEXTCLK alternate function (refer to Chapter 24, “General-Purpose I/O Controller”).
SSPSCLK2	Inout	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_1[SCLKDIR].
SSPSYSCLK2	Output	SSPSYSCLK2 is four times the SSPSCLK2 value when using Audio Clock PLL Select (SSACD_2[ACPS]) and Audio Clock Divider (SSACD_2[ACDS]).
SSPSFRM2	Inout	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 SSPCR1_2[SFRMDIR].
SSPTXD2	Output	SSPTXD2 is the transmit data (serial data out) serialized data line.
SSPRXD2	Input	SSPRXD2 is the receive data (serial data in) serialized data line.
SSPEXTCLK2	Input	SSPEXTCLK2 is an external clock that is selected to replace the internal 13-MHz clock. SSPEXTCLK2 is multiplexed with the SSPSCLK2EN alternate function (refer to Chapter 24, “General-Purpose I/O Controller”).

Table 8-1. SSP Serial Port I/O Signals (Sheet 2 of 2)

Name	Direction	Description
SSPSCLK2EN	Input	SSPSCLK2EN is an asynchronous external enable for SSPSCLK2. This function is multiplexed with other alternate functions. Refer to Chapter 24, "General Purpose I/O Controller," of the *Intel® PXA27x Processor Family Developers Manual*. SSPSCLK2EN is multiplexed with the SSPEXTCLK2 alternate function (refer to Chapter 24, "General-Purpose I/O Controller").
SSPSCLK3	Inout	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 SSPCR1_3[SCLKDIR].
SSPSYSCLK3	Output	SSPSYSCLK3 is four times the SSPSCLK3 value when using Audio Clock PLL Select (SSACD_2[ACPS]) and Audio Clock Divider (SSACD_2[ACDS]).
SSPSFRM3	Inout	SSPSFRM3 is the serial frame indicator that indicates 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 SSPCR1_3[SFRMDIR].
SSPTXD3	Output	SSPTXD3 is the transmit data (serial data out) serialized data line.
SSPRXD3	Input	SSPRXD3 is the receive data (serial data in) serialized data line.
CLK_EXT	Input	CLK_EXT is an external Network clock that replaces the internal 13 MHz clock. Use CLK_EXT when SSCR0_x[NSC] is set and SSCR1_x[SCLKDIR] is cleared. CLK_EXT is used by multiple SSPs.

8.3 Block Diagram

8.3.1 Standard SSP Configuration Scheme

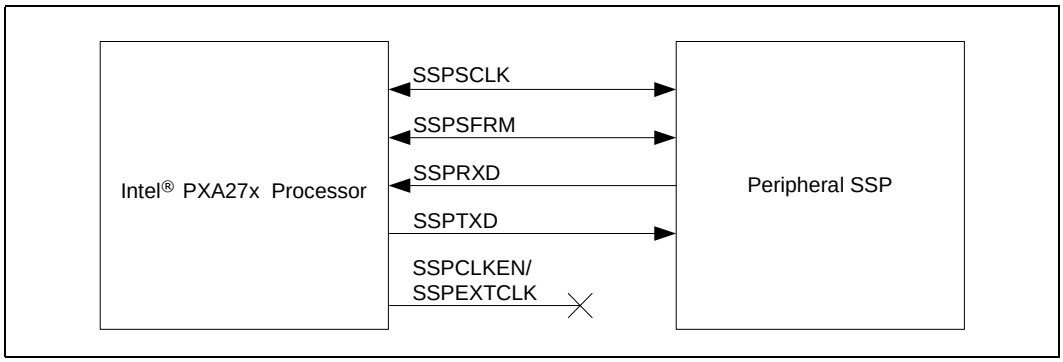
The Standard SSP configuration scheme is the most usual method of configuration and allows one SSP to interface directly to another SSP on another device.

This configuration allows for a number of possible configurations. The SSPSCLK is either a master or a slave to the peripheral. Likewise, the SSPSFRM is also programmed to be a master or a slave to the peripheral. This allows for one physical connection to result in either 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 8-1 for illustration of the physical connections of the above configurations.

Figure 8-1. Standard SSP Configuration Scheme Block Diagram



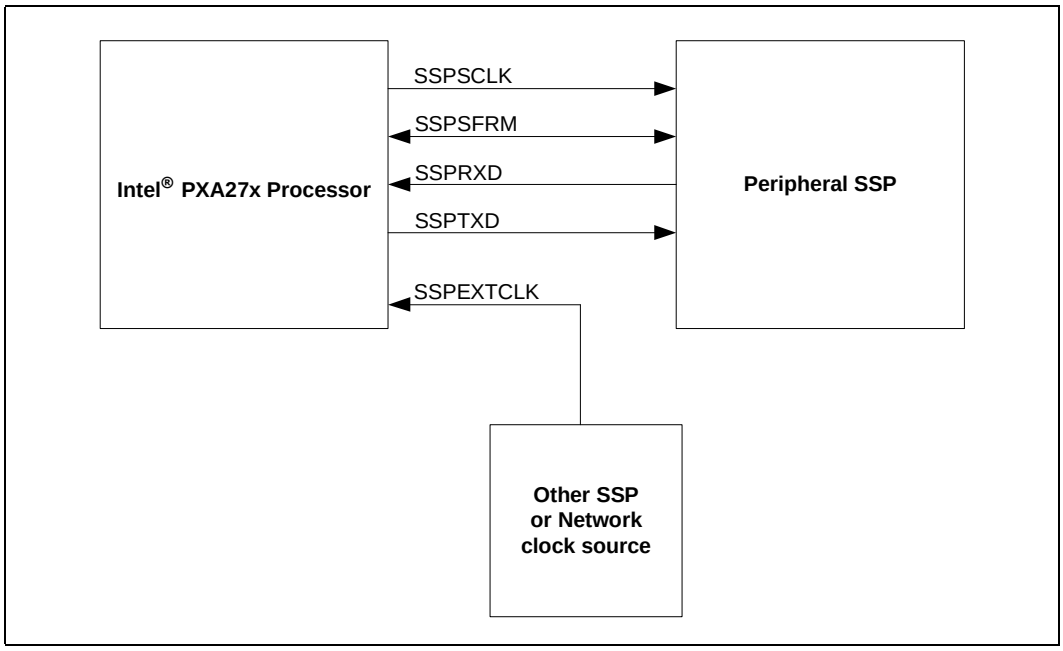
8.3.2 External Clock Source Configuration Scheme

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 as the source for the 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 8-2 for illustration of the physical connection of the external clock source configuration.

Figure 8-2. External Clock Source Configuration Scheme Block Diagram

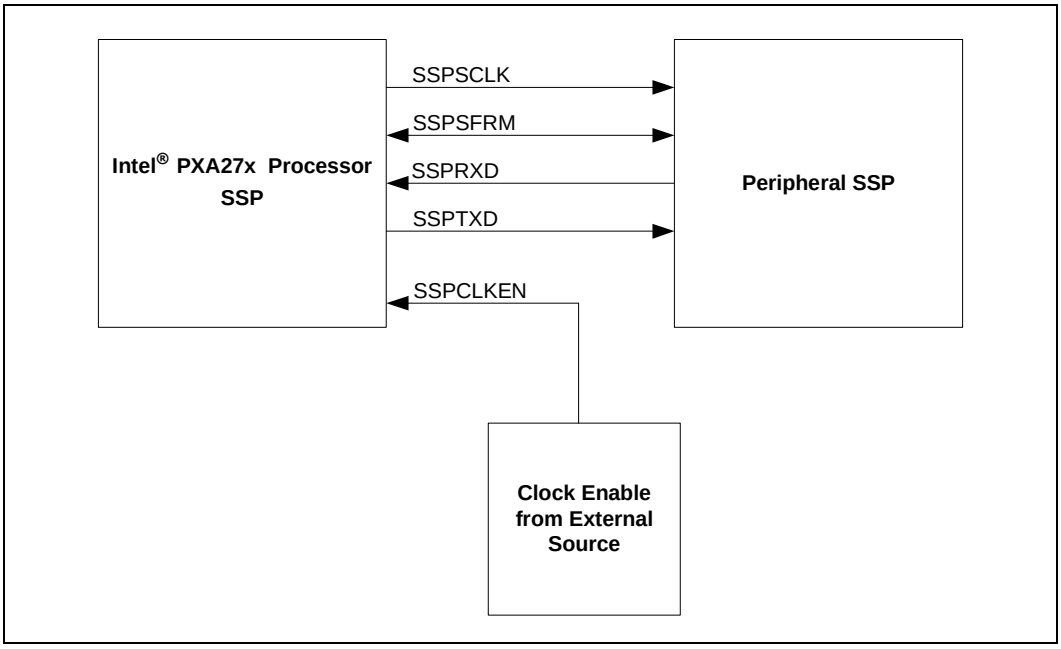


8.3.3 External Clock Enable Configuration Scheme

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 do not take an external or network clock as the base clock.

See Figure 8-3 for illustration of the physical connection of the external clock enable configuration.

Figure 8-3. External Clock Enable Configuration Scheme Block Diagram

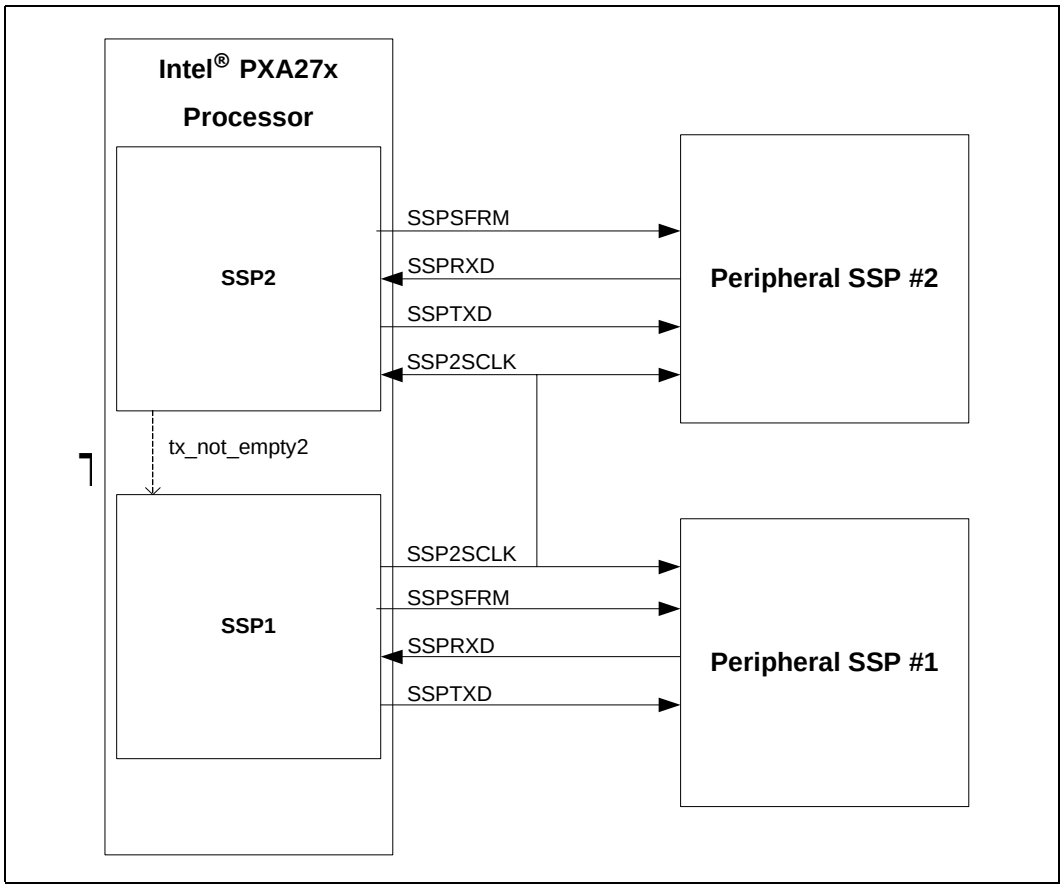


8.3.4 Internal (to PXA27x Processor) Clock Enable Configuration Design

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. In this mode, the SSP, which is internally being enabled, must be the master of SSPSCLK.

See Figure 8-4 for 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 PXA27x processor 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 PXA27x processor, and therefore, would not receive any data. In this example, the only function of SSP1 of the PXA27x processor is to generate the SSPSCLK signal.

Figure 8-4. Internal Clock Enable Configuration Scheme Block Diagram



8.4 Layout Notes

The tested maximum switching speed of the SSPs is 13 MHz. Design constraints at this speed are generally not very strict. 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.

To ensure the most reliable design possible follow these recommendations when routing signals:

- 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 inner signal layer as possible.

SSPFRM 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.

§§

Inter-Integrated Circuit (I^2C) 9

9.1 Overview

The Inter-Integrated Circuit (I^2C) bus interface unit allows Intel® PXA27x Processor Family (PXA27x processor) to serve as a master and slave device residing on the I^2C bus. The I^2C bus is a serial bus developed by the Philips Corporation consisting of a two-pin interface. SDA is the serial data line and SCL is the serial clock line. A complete list of features and capabilities are found in the *I^2C Bus Specification*.

Using the I^2C bus lets the PXA27x processor interface to other I^2C peripherals and microcontrollers for system management functions. The serial bus requires a minimum of hardware for an economical system to relay status and reliability information to an external device.

The I^2C bus interface unit is a peripheral device that resides on the PXA27x processor internal bus. Data is transmitted to and received from the I^2C bus using a buffered interface. Control and status information is relayed through a set of memory-mapped registers. Refer to the *I^2C Bus Specification* for complete details on I^2C bus operation.

9.2 Signals

See Table 9-1 for the description of the I^2C bus interface unit signals.

Table 9-1. I^2C Signal Description

Signal Name	Input/Output	Description
SDA	Bidirectional	Serial data
SCL	Bidirectional	Serial clock

The I^2C 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's clock is only altered by a slow slave peripheral keeping the clock line low or by another master during arbitration.

The I^2C 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 I^2C bus arbitration relies on the wired-AND connection of all I^2C interfaces to the I^2C 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.

9.3 Schematic/Block Diagram

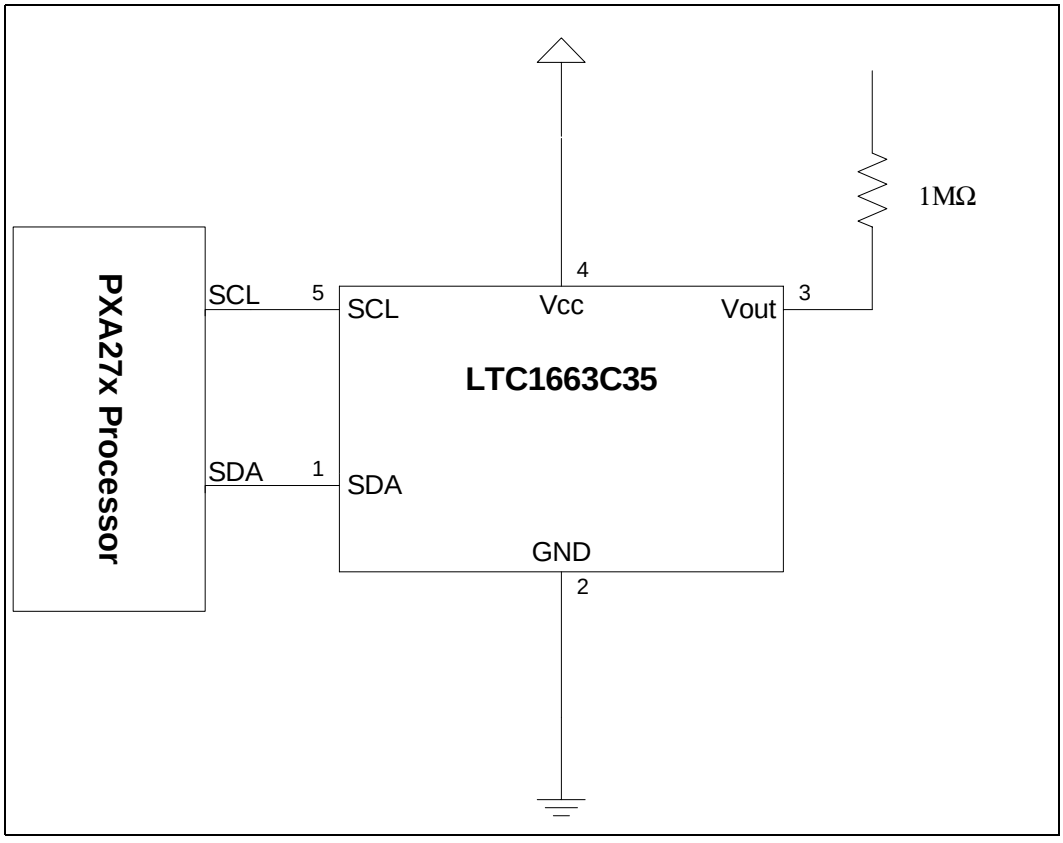
The I^2C bus is used by many different applications. This reference guide presents two possible methods for using the I^2C bus interface:

- The first method controls a digital-to-analog converter (DAC) to vary the DC voltage to the processor core.
- The second method expands the capabilities of an existing compact flash socket.

9.3.1 Digital-to-Analog Converter (DAC)

See Figure 9-1 for illustration of the schematic showing the connection of the I^2C interface to a Linear Technology micropower DAC. The DAC output is connected to the buck converter feedback path and is controlled by the I^2C bus interface unit. The DAC modifies the voltage of the feedback path that affects the processor core voltage.

Figure 9-1. Linear Technology DAC with I^2C Interface

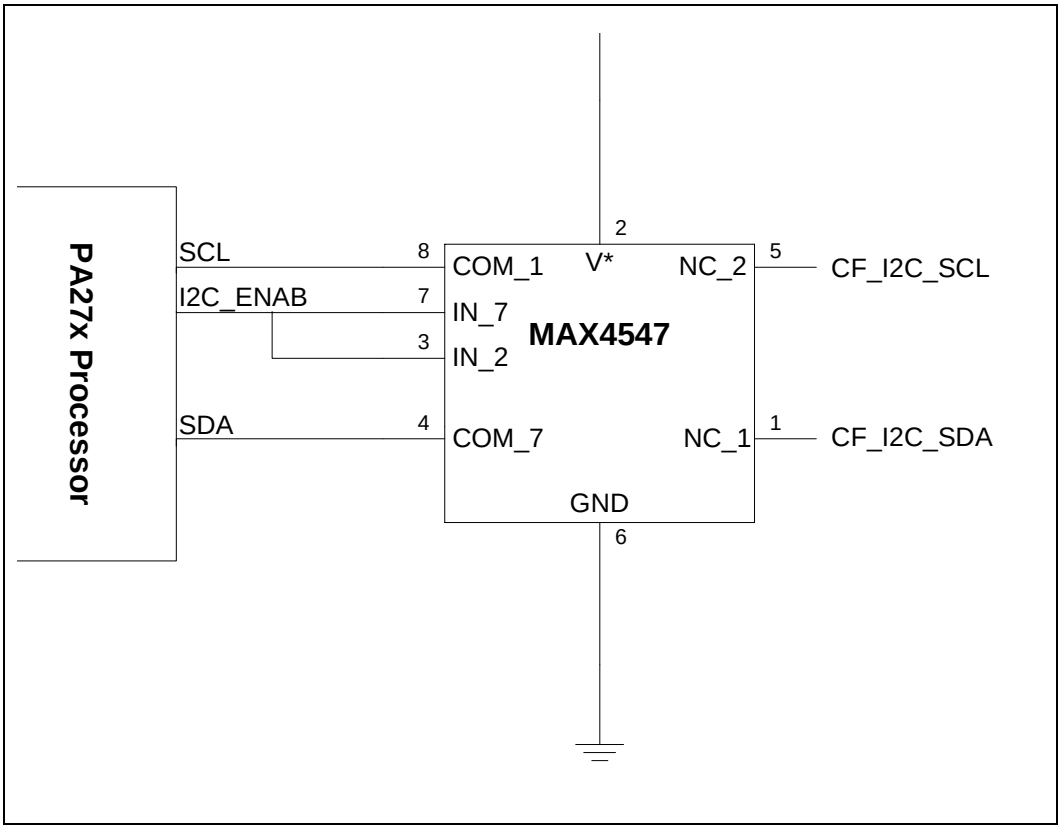


9.3.2 Other Uses of I^2C

See Figure 9-2 for illustration of the I^2C signals passing through an analog switch to a compact flash (CF) socket. Since the CF socket has all of the signals to support two CF cards, and this design only uses one CF card, the signals meant for a second card are being used for alternate functions. If a CF card is not to be used, a different application using a CF card socket could be designed to utilize the I^2C bus interface unit. If this alternate function is used, enable the I^2C bus to the CF socket by asserting the signal SA_I2C_ENAB, shown in Figure 9-2. If using a CF Card, negate the SA_I2C_ENAB signal so the I^2C bus traffic does not interfere with the CF card.

Note: The CF card socket is disabled if a device is inserted in the expansion bus.

Figure 9-2. Using an Analog Switch to Allow a Second CF Card



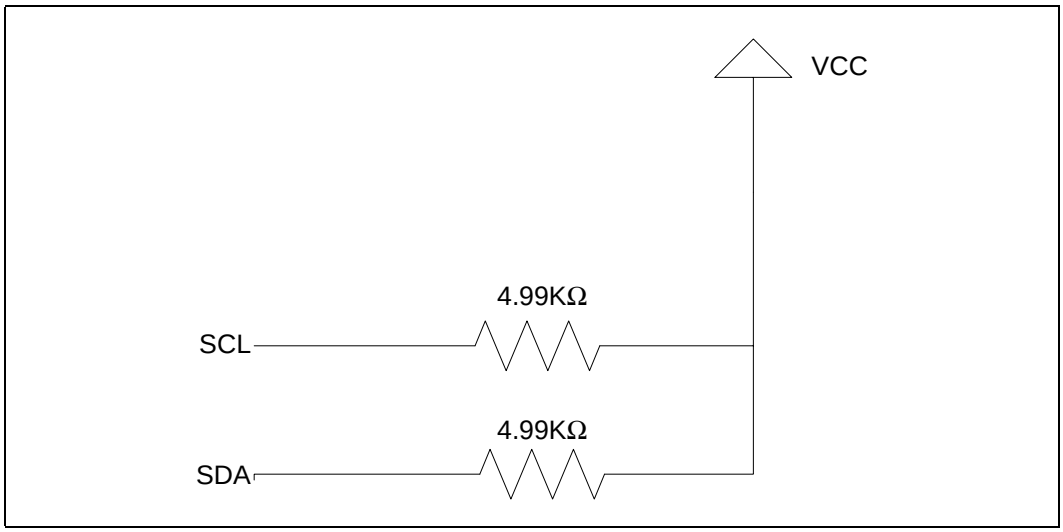
9.3.3 Pull-Ups and Pull-Downs

The *I²C Bus Specification*, available from Philips Corporation, states:

> "The external pull-up devices connected to the bus lines must be adapted to accommodate the shorter maximum permissible rise time for the Fast-mode I²C-bus. For bus loads up to 200 pF, the pull-up device for each bus line can be a resistor; for bus loads between 200 pF and 400 pF, the pull-up device can be a current source (3 mA max.) or a switched resistor circuit."

The design presented in this guide is not intended for loads larger than 200 pF, so the pull-up device is a resistor as shown in Figure 9-3.

Figure 9-3. I²C Pull-Ups and Pull-Downs



The actual value of the pull-up is system dependent and a guide is presented in the *I²C Bus Specification* on determining the maximum and minimum resistors to use when the system is intended for standard or fast-mode I²C bus devices.

9.4 Layout Notes

The maximum switching frequency of the PWM signals is 400 KHz. Therefore, layout and routing considerations are not stringent and somewhat relaxed. 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.

§§

UART Interfaces 10

This chapter describes guidelines to interface to the Universal Asynchronous Receiver/Transmitter (UART) serial ports of Intel® PXA27x Processor Family (PXA27x processor).

10.1 Overview

The PXA27x processor has three UARTs:

- A Full Function UART (FFUART)
- A Bluetooth[1] UART (BTUART)
- A standard UART (STUART)

All the UARTs use the same programming model.

The serial ports are controlled using direct memory access (DMA) or programmed I/O. Each serial port contains a UART and a slow infrared transmit encoder and receive decoder that conforms to the IrDA Serial Infrared (SIR) Physical Layer Link Specification.

Each UART performs serial-to-parallel conversion on data characters received from a peripheral device or a modem and parallel-to-serial conversion on data characters received from the processor. The processor reads the complete status of a UART during functional operation. Status information includes the type and condition of transfer operations and error conditions (parity, overrun, framing, or break interrupt) associated with the UART.

Each serial port operates in FIFO or non-FIFO mode. In FIFO mode, a 64-byte transmit FIFO holds data from the processor until it is transmitted on the serial link and a 64-byte receive FIFO buffers data from the serial link until it is read by the processor. In non-FIFO mode, the transmit and receive FIFOs are bypassed.

Each UART includes a programmable baud rate generator that divides the input clock by 1 to (2^{16}–1). This produces a 16X clock that drives the internal transmitter and receiver logic. Software programs interrupts to meet its requirements. This minimizes the number of computations required to handle the communications link. Each UART operates in an environment which is controlled by software and is polled or interrupt driven.

All three UARTs support the 16550A[2] and 16750[3] functions, but are slightly different in regard to features supported by each of the three UARTs.

1. Bluetooth is trademarked by the Bluetooth SIG Inc. U.S.A. See the Bluetooth SIG Internet site at http://www.bluetooth.com.
2. The 16550A was originally produced by National Semiconductor Inc.
3. The 16750 is produced as the TL16C750 by Texas Instruments.

10.2 Signals

See Table 10-1 for the list and description of the external signals connected to the UART modules. The signals are connected to the PXA27x processor through GPIOs.

Table 10-1. UART Signal Descriptions (Sheet 1 of 2)

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. The TXD signal 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.
nCTS	Input	CLEAR TO SEND – When low, indicates the modem or data set is ready to exchange data. The nCTS signal is a modem status input and its condition is tested by reading bit 4 (CTS) of the Modem Status Register (MSR). Bit 4 is the complement of the nCTS signal. Bit 0 (DCTS) of the MSR indicates whether the nCTS input has changed state since the last time the MSR was read. nCTS has no effect on the transmitter. This signal is present on the FFUART and BTUART. When the CTS bit of the MSR changes state and the Modem Status interrupt is enabled, an interrupt is generated. Non-Autoflow Mode: When not in Autoflow mode, bit 4 (CTS) of the MSR indicates the state of nCTS. Bit 4 is the complement of the nCTS signal. Bit 0 (DCTS) of the MSR indicates whether the nCTS input has changed state since the previous reading of the MSR. nCTS has no effect on the transmitter. The user programs the UART to interrupt the processor when DCTS changes state. The programmer then stalls the outgoing data stream by starving the transmit FIFO, or disabling the UART with the IER register. **NOTE:** If UART transmission is stalled by disabling the UART, the user does not receive an MSR interrupt when nCTS reasserts. This is because disabling the UART also disables interrupts. To get around this, either use Auto CTS in Autoflow Mode or program the nCTS GPIO pin to interrupt. Autoflow Mode: In Autoflow mode, the UART Transmit circuity checks the state of nCTS before transmitting each byte. If nCTS is high, no data is transmitted.
nDSR	Input	DATA SET READY – When low, indicates the modem or data set is ready to establish a communications link with a UART. The nDSR signal is a Modem Status input and its condition is tested by reading Bit 5 (DSR) of the MSR. Bit 5 is the complement of the nDSR signal. Bit 1 (DDSR) of the MSR indicates whether the nDSR input has changed state since the MSR was last read. This signal is present only on the FFUART. When the DSR bit of the MSR changes state, an interrupt is generated if the Modem Status interrupt is enabled.
nDCD	Input	DATA CARRIER DETECT – When low, indicates the data carrier has been detected by the modem or data set. The nDCD signal is a modem status input and its condition is tested by reading Bit 7 (DCD) of the MSR. Bit 7 is the complement of the nDCD signal. Bit 3 (DDCD) of the MSR indicates whether the nDCD input has changed state since the previous reading of the MSR. nDCD has no effect on the receiver. This signal is present only on the FFUART. When the DCD bit changes state and the Modem Status interrupt is enabled, an interrupt is generated.

Table 10-1. UART Signal Descriptions (Sheet 2 of 2)

Name	Direction	Description
nRI	Input	RING INDICATOR – When low, indicates the modem or data set has received a telephone ringing signal. The nRI signal is a Modem Status input whose condition is tested by reading Bit 6 (RI) of the MSR. Bit 6 is the complement of the nRI signal. Bit 2, the trailing edge of ring indicator (TERI), of the MSR indicates whether the nRI input signal has changed from low to high since the MSR was last read. This signal is present only on the FFUART. When the RI bit of the MSR changes from a high to low state and the Modem Status interrupt is enabled, an interrupt is generated.
nDTR	Output	DATA TERMINAL READY – When low, signals the modem or the data set that the UART is ready to establish a communications link. The nDTR output signal is set to an active low by programming Bit 0 (DTR) of the MSR to a 1. A Reset operation sets this signal to its inactive state. LOOP mode operation holds this signal in its inactive state. This signal is present only on the FFUART.
nRTS	Output	REQUEST TO SEND – When low, signals the modem or the data set that the UART is ready to exchange data. The nRTS output signal is set to an active low by programming Bit 1 (RTS) of the Modem Control Register to a 1. A Reset operation sets this signal to its inactive (high) state. LOOP mode operation holds this signal in its inactive state. This signal is used by the FFUART and BTUART. Non-Autoflow Mode: The nRTS output signal is asserted by setting bit 1 (RTS) of the Modem Control register to a 1. The RTS bit is the complement of the nRTS signal. Autoflow Mode: nRTS is automatically asserted by the autoflow circuitry when the Receive buffer exceeds its programmed threshold. It is de-asserted when enough bytes are removed from the buffer to lower the data level back to the threshold.

10.3 Types of UARTs

The PXA27x processor has three separate UARTs; however, each UART is different with respect to signals and features supported. The UARTs are described in this section.

10.3.1 Full Function UART

The FFUART supports modem control capability. The maximum baud rate is 921,600 bps.

10.3.1.1 Full Function UART Signals

See Table 10-2 for the list and description of the Full Function UART signals.

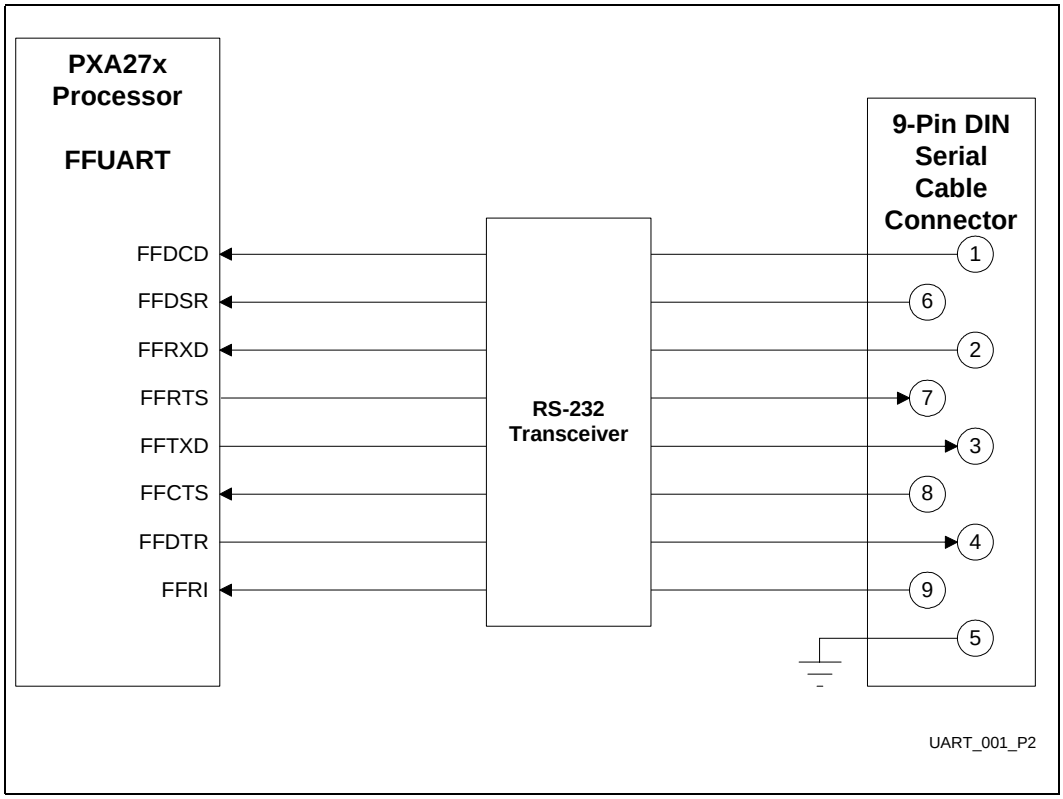
Table 10-2. FFUART Interface Signals

Signal Name	Direction	Description
FFRXD	Input	Full Function UART Receive, serial data input to the UART
FFTXD	Output	Full Function UART Transmit, serial data output from the UART
FFCTS	Input	Full Function UART Clear to Send
FFDSR	Input	Full Function UART Data Set Ready
FFDCD	Input	Full Function UART Data Carrier Detect
FFRI	Input	Full Function UART Ring Indicator
FFDTR	Output	Full Function UART Data Terminal Ready
FFRTS	Output	Full Function UART Request to Send

10.3.1.2 FFUART Block Diagram

See Figure 10-1 for illustration of the block diagram showing the interface between the FFUART and a standard 9-pin DIN connector. The block diagram demonstrates high-level signal usage and connectivity when using the FFUART.

Figure 10-1. FFUART Interface Block Diagram



10.3.1.3 FFUART 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 DIN connector.

10.3.2 Bluetooth UART

The BTUART is connected to a Bluetooth module. The maximum baud rate is 921,600 bps. The BTUART only supports two modem control pins (nCTS, nRTS).

10.3.2.1 Bluetooth UART Signals

See Table 10-3 for the list and description of the Bluetooth UART signals.

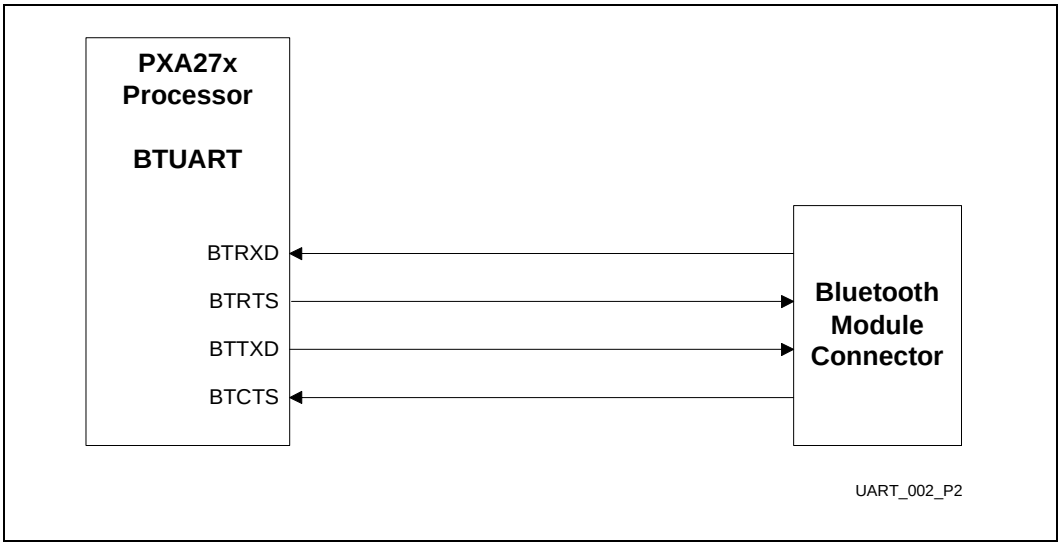
Table 10-3. BTUART Interface Signals

Signal Name	Direction	Description
BTRXD	Input	Bluetooth UART Receive, serial data input to the UART
BTTXD	Output	Bluetooth UART Transmit, serial data output from the UART
BTCTS	Input	Bluetooth UART Clear to Send
BTRTS	Output	Bluetooth UART Request to Send

10.3.2.2 Bluetooth UART Block Diagram

See Figure 10-2 for illustration of a block diagram showing the interface between the BTUART and a Bluetooth module connector. The block diagram demonstrates high level signal usage and connectivity when using the BTUART.

Figure 10-2. BTUART Interface Block Diagram



10.3.3 Standard UART

The STUART does not support modem control capability and is typically used for the (slow) IrDA transceiver interface. The maximum baud rate is 921,600 bps.

10.3.3.1 Standard UART Signals

See Table 10-4 for the list and description of the Standard UART signals.

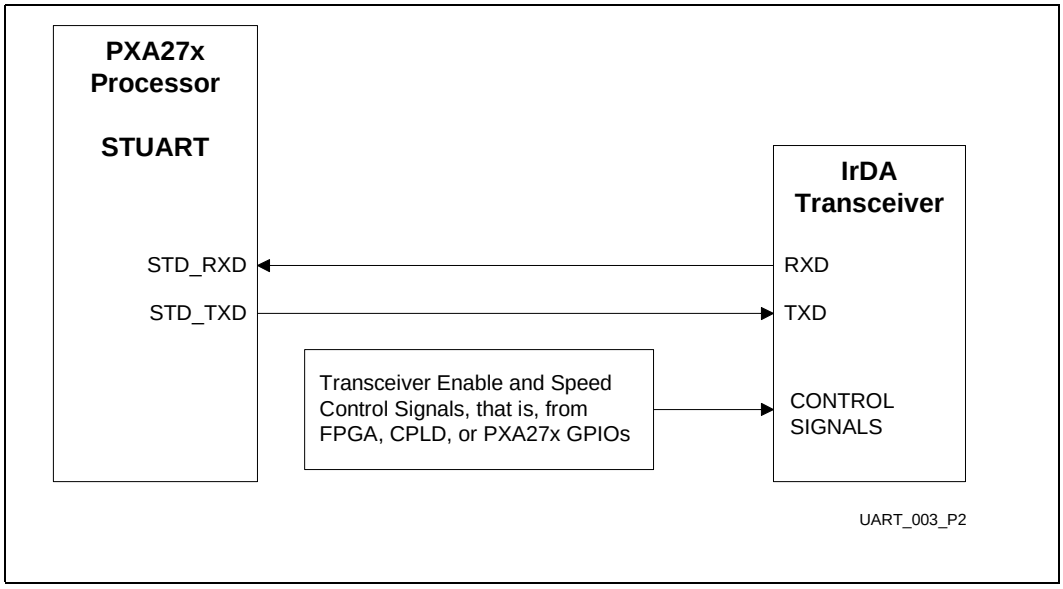
Table 10-4. STUART Interface Signals

Signal Name	Direction	Description
STD_RXD	Input	Standard UART Receive, serial data input to the UART
STD_TXD	Output	Standard UART Transmit, serial data output from the UART

10.3.3.2 Standard UART Block Diagram

See Figure 10-3 for illustration of a block diagram showing the interface between the STUART and an IrDA transceiver. The block diagram demonstrates high level signal usage and connectivity when using the STUART.

Figure 10-3. STUART Interface Block Diagram



§§

Fast Infrared Interface 11

This chapter describes guidelines for interfacing with the external LED transceivers to fast infrared (FIR) controller of Intel® PXA27x Processor Family (PXA27x processor).

11.1 Overview

The Fast Infrared Communications Port (FICP) operates at half-duplex and provides direct connection to commercially available Infrared Data Association (IrDA) compliant LED transceivers. The FICP is based on the 4 Mbps IrDA standard[1] and uses four-position pulse modulation (4 PPM) and a specialized serial packet protocol developed for IrDA transmission. To support the standard, the FICP is equipped with:

- A bit encoder/decoder
- A serial-to-parallel data engine
- A transmit First In/First Out (FIFO) 64 entries deep and 8-bits wide
- A receive FIFO 64 entries deep and 11-bits wide

The FICP shares GPIO pins for transmit and receive data with the standard UART and the Bluetooth UART. Only two of these three ports, which are FICP, STUART, and BTUART, are used at any one time. To support a variety of IrDA transceivers, both the transmit and receive data pins are individually configured to communicate using normal or active low data.

11.2 Signals

See Table 11-1 for description of the function of each signal. Most IrDA transceivers also have enable and speed pins. Use GPIOs or control signals from external logic to enable the transceiver and select the speed

Table 11-1. FICP Signal Description

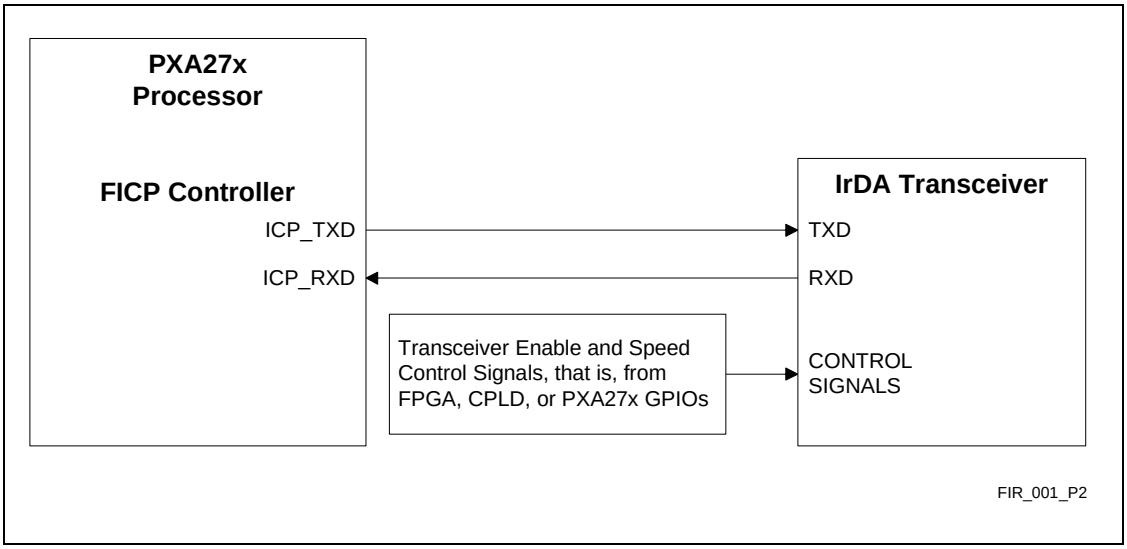
Signal Name	Input/Output	Description
ICP_RXD	Input	Receive pin for FICP
ICP_TXD	Output	Transmit pin for FICP

1. Refer to Infrared *Data Association Serial Infrared Physical Layer Specification Version 1.3*, October 15, 1998 available from www.irda.org

11.3 Block Diagram

See Figure 11-1 for the block diagram showing the FICP of the PXA27x processor interfaced to an IrDA LED transceiver.

Figure 11-1. Fast Infrared Controller Port Interface Block Diagram



§§

'USB Client Controller 12

This chapter describes guidelines to interface the Universal Serial Bus (USB) Client Controller of Intel® PXA27x Processor Family (PXA27x processor) to a USB client cable connector for attachment to a USB network. Included in this chapter is a description of how to connect the PXA27x processor single-ended USB Host and USB On-The-Go (OTG) ports to external components.

12.1 Overview

A working knowledge of the *Universal Serial Bus Specification, Revision 1.1*[1] is necessary to fully understand the material contained in this chapter. The USB Client 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 1.1* for a full description of the USB protocol and its operation. The USB Client is a full-speed device that operates half-duplex at a baud rate of 12 Mbps as a slave only, not a Host or Hub controller. The USB Host controller referenced in this chapter refers to any *Universal Serial Bus Specification, Revision 1.1*-compliant USB Host controller, including the internal USB host controller of the PXA27x 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.

12.2 Signals

See Table 12-1 for description of the two USB Client signals.

Table 12-1. USB Client Controller Interface Signals Summary

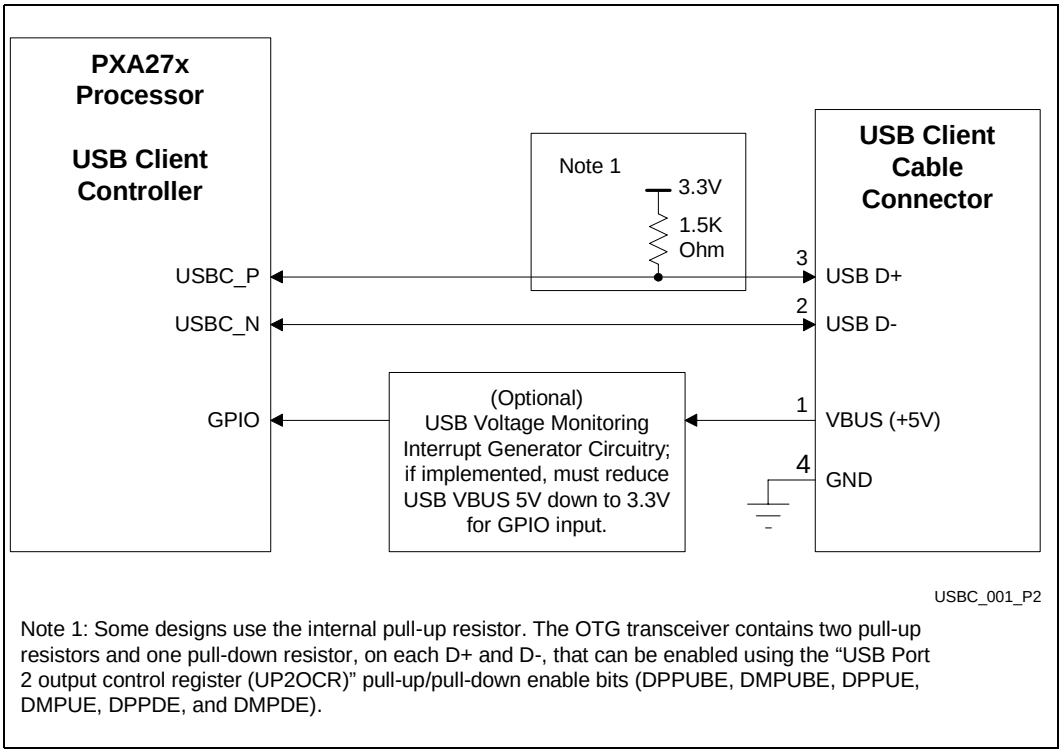
Signal Name	Direction	Description
USBC_P	Bidirectional	USB Client port positive pin of differential pair
USBC_N	Bidirectional	USB Client port negative pin of differential pair

1. Access the latest revision of the *Universal Serial Bus Specification, Revision 1.1* 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/

12.3 Block Diagram

See the USB Client interface block diagram in Figure 12-1.

Figure 12-1. USB Client Interface Block Diagram



Note 1: Some designs use the internal pull-up resistor. The OTG transceiver contains two pull-up resistors and one pull-down resistor, on each D+ and D-, that can be enabled using the "USB Port 2 output control register (UP2OCR)" pull-up/pull-down enable bits (DPPUBE, DMPUBE, DPPUE, DMPUE, DPPDE, and DMPDE).

12.4 Layout Notes

USBC_P and USBC_N are differential pair signals; follow these layout guidelines:

- 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)).

This section includes layout notes for self-powered USB Client devices and for bus-powered USB Client devices.

12.4.1 Self-Powered Devices

Figure 12-2 and Figure 12-3 show two similar USB interface connection diagrams for self-powered devices. USB D+ connects directly to the PXA27x processor USBC_P signal and USB D- connects directly to the USBC_N signal. The USBC_P and USBC_N signals match the impedance of a USB cable, 90 Ω, without the use of external series resistors. The 0 Ω resistors are optional and are installed if series resistors are necessary to compensate for minor differences between the USB cable and the board trace impedances.

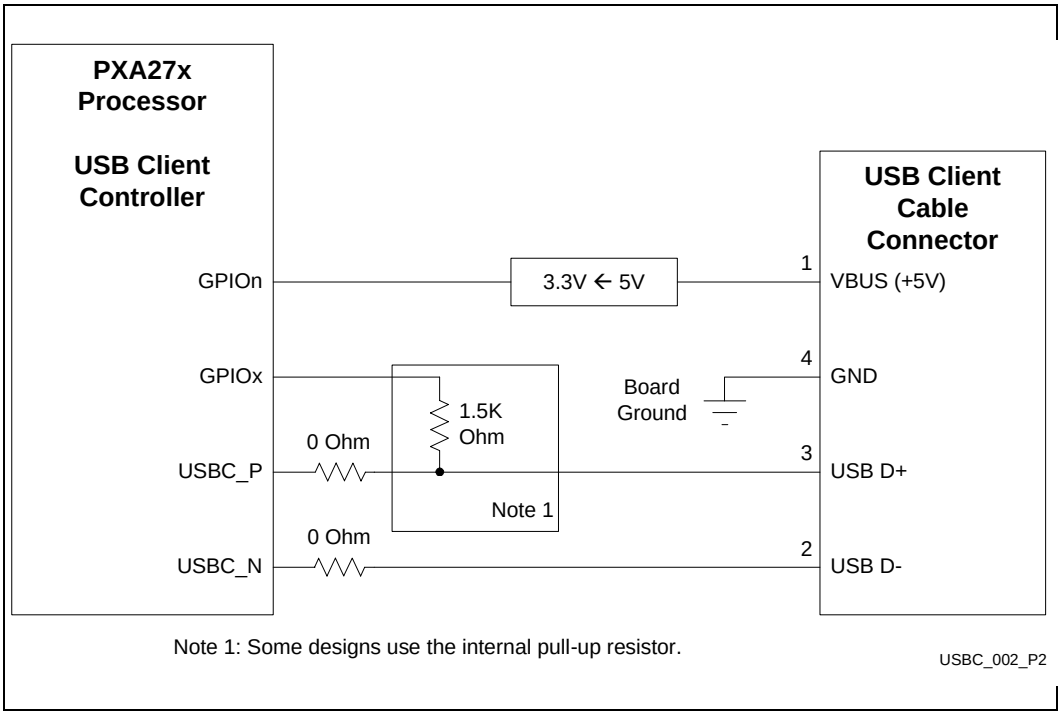
The 5.0 V to 3.3 V voltage divider is required since the PXA27x processor GPIO pins cannot exceed 3.3 V. This voltage divider is implemented in a number of ways:

- 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 more inexpensive 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.

12.4.1.1 Operation if GPIOn and GPIOx are Different Pins

See Figure 12-2 for illustration of a self-powered USB Client device using two different GPIO signals for USB cable detection.

Figure 12-2. Self Powered Device when GPIOn and GPIOx are Different Pins



Any GPIO signal is defined as GPIOn and GPIOx; however, GPIOn must be a GPIO which brings the PXA27x 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 1.1*.

The configuration of using separate GPIOs for GPIOn and GPIOx lets the PXA27x processor enter sleep or deep sleep mode with a USB cable connected or disconnected to the USB Client interface. Before entering sleep or deep sleep mode, the software must read the GPIOn pin to determine if a USB cable is connected or not.

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.
- Configure the GPIOn pin to detect a wake-up event.

The PXA27x processor is then put into sleep or deep sleep mode.

If a USB cable is detected, software must configure the GPIOx pin as an input to allow the USB D+ line to float. This looks like a virtual disconnect to the host controller although the USB cable remains physically attached to the PXA27x processor USB Client interface.

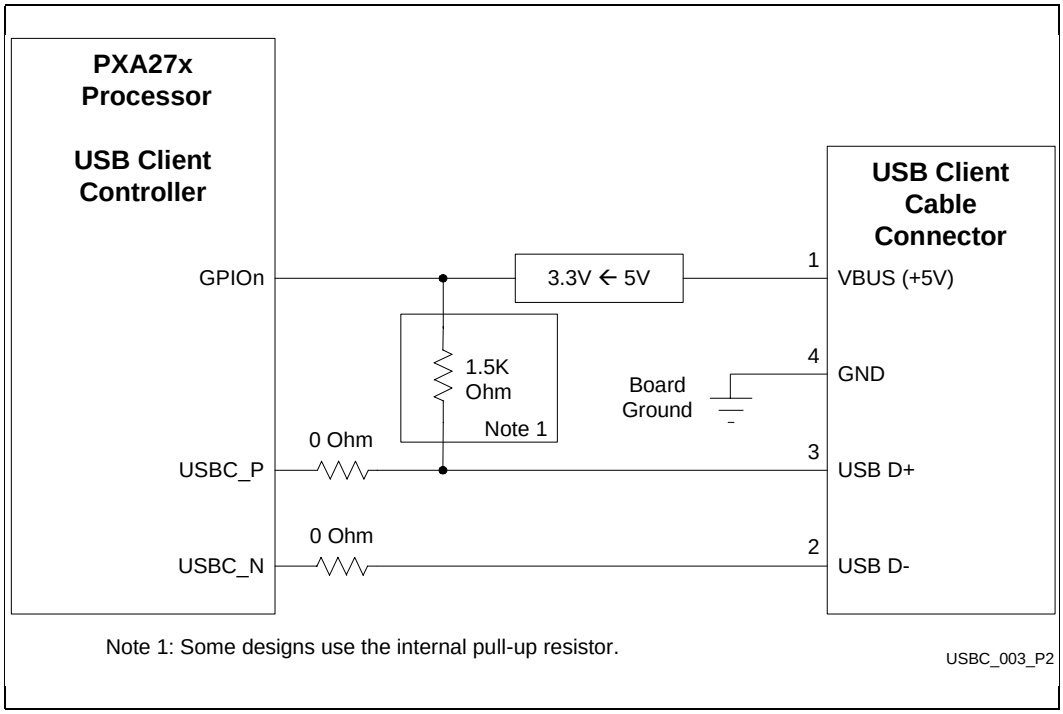
The PXA27x processor is then put into sleep or deep sleep mode.

When the PXA27x processor wakes-up, software must drive the GPIOx signal high pulling USB D+ high indicating to the host controller that a fast USB client device has been (virtually) connected.

12.4.1.2 Operation if GPIOn and GPIOx are the Same Pin

See Figure 12-3 for illustration of a self-powered USB Client device utilizing one GPIO signal for detection.

Figure 12-3. Self-Powered Device when GPIOn and GPIOx are Same Pins



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 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 is connected, USB D+ is pulled high using the 1.5 KΩ resistor to indicate to the host controller that a fast USB client device is connected. Software must enable the USB Client peripheral before the host sends the first USB command.

Note: When the single GPIOn/x configuration is in use, never put the PXA27x processor into sleep or deep sleep mode while the USB cable is connected to the USB client interface for these reasons:

- During sleep and deep sleep modes, the USB controller is in reset and does not respond to the host.
- After sleep or deep sleep mode, the USB client does not respond to its host-assigned address because the previously assigned address is not retained during sleep and deep sleep modes.

Use the following sequence to enter sleep mode when GPIOn and GPIOx are the same signal. The 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 sleep mode.

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

12.4.2 Bus-Powered Device

The PXA27x processor cannot support a bus-powered device model. When the host sends a suspend command, the device must consume less than 500 µA (refer to Section 7.2.3 of the *Universal Serial Bus Specification Revision 1.1*). The processor cannot limit its current consumption to 500 µA unless it enters sleep or deep sleep mode. If it enters either of these reduced power modes, all USB registers are reset and the USB client does not respond to its host-assigned address.

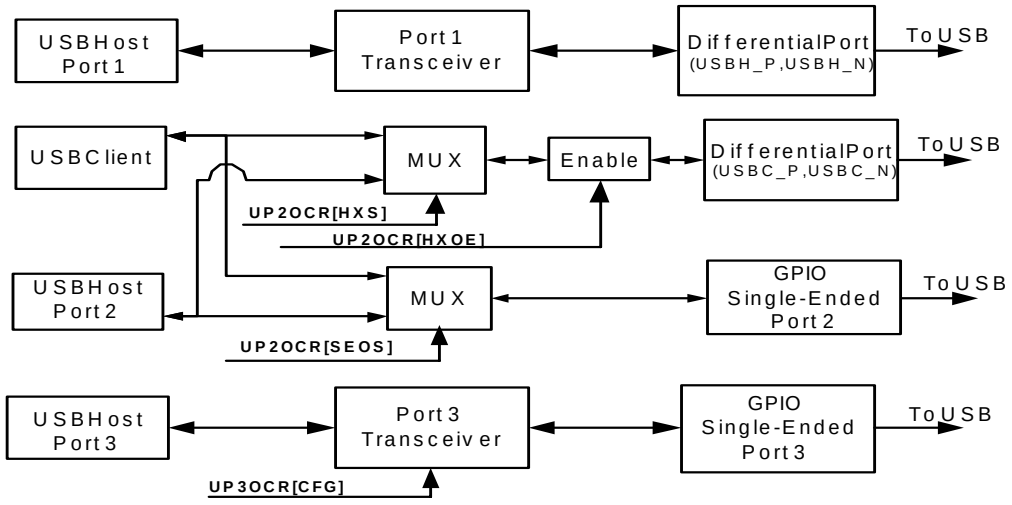
12.4.3 USB On-The-GO Transceiver Usage

See Figure 12-4 for 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 PXA27x processor does not provide:

- Direct connection to the USB Vbus
- Control of the USB Vbus

Figure 12-4. USB OTG Configurations



The USB Host port 2 transceiver is designed in accordance with the *Pull-up/Pull-down 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 PXA27x processor OTG transceiver consists of two pull-up resistors and one pull-down resistor 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 12-5 for illustration of the on-chip Host port 2 transceiver pad with the pull-up and pull-down resistors.
- As shown in Figure 12-5, SW3 is enabled for both D+ and D- when Host port 2 is being used for USB host controller data.
- As shown in Figure 12-5, 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 12-5, 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 *Pull-up/Pull-down Resistors Engineering Change Notice to the USB 2.0 Specification*.

See Table 12-2 for the list of switch settings used for the USB Host and USB Device controller I/O.

Figure 12-5. Host Port 2 OTG Transceiver

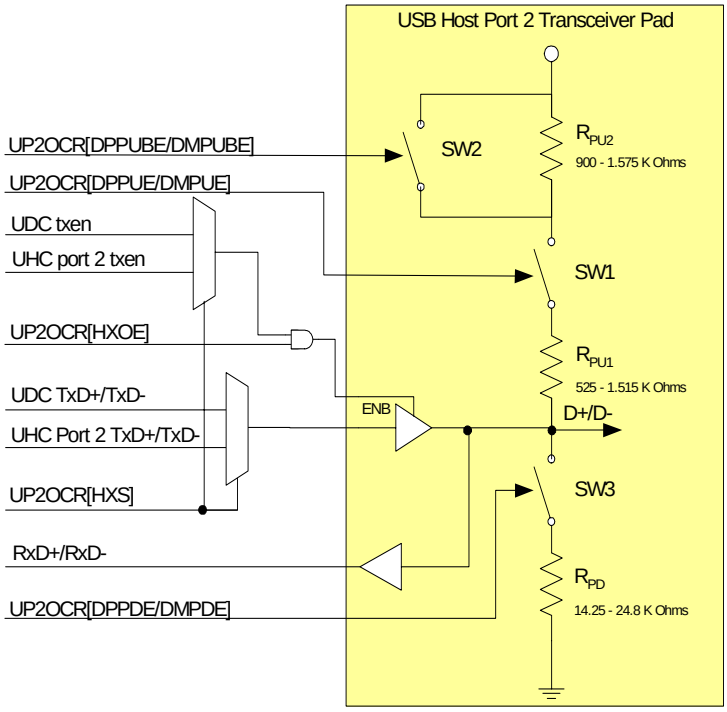


Table 12-2. Host Port 2 OTG Transceiver Switch Control Settings

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 12-5.						

Note: There are programming requirements for disabling and enabling the transceiver and pull-up/pull-down resistors for USB host port 2 and for sleep and standby mode operation. Refer to the *Intel® PXA27x Processor Family Developers Manual* for additional information.

12.4.4 Interface to External Transceiver (OTG)

If the user does not use the internal OTG transceiver, 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. 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:

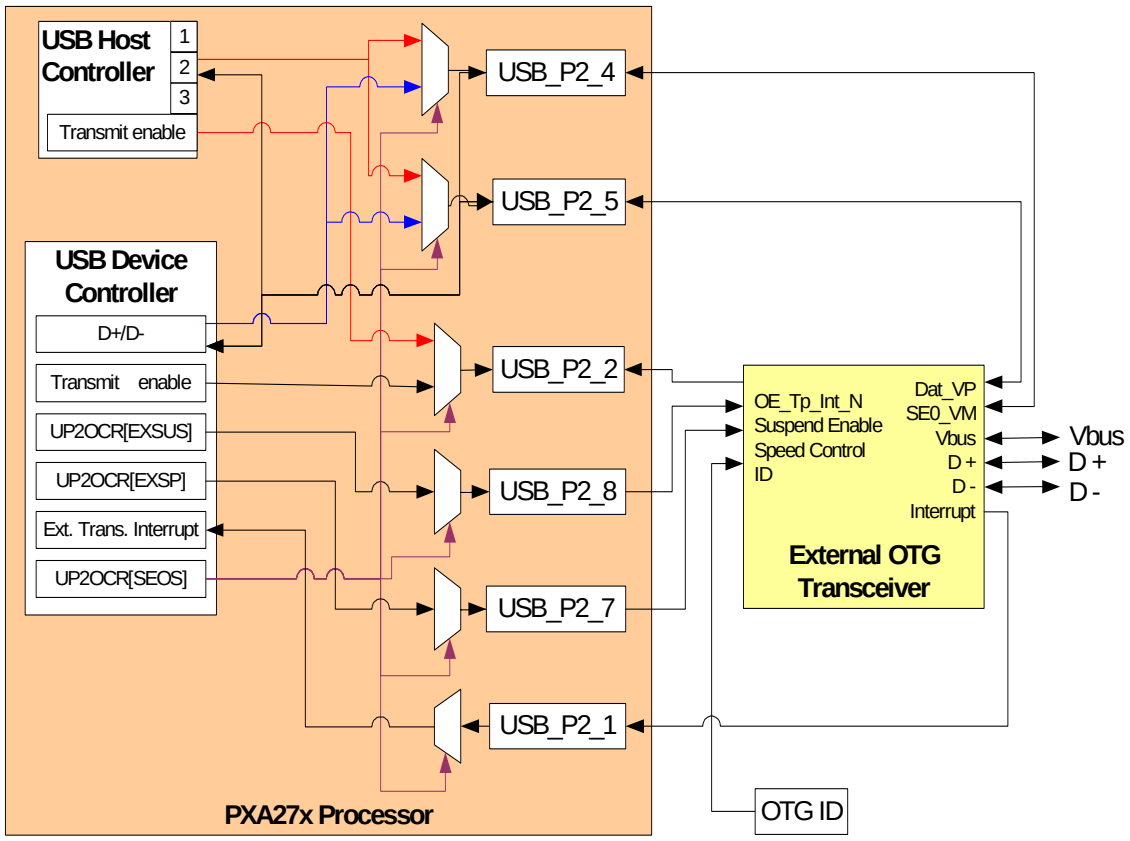
- UDC
- D+, D-, and transmit enable signals of USB host controller

In addition, "USB Port 2 Output Control Register (UP2OCR)" provides:

- 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 12-6 for illustration of the PXA27x processor OTG connections to an external transceiver.

Figure 12-6. Connection to External OTG Transceiver



12.4.5 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 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 following:

- UDC
- 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:

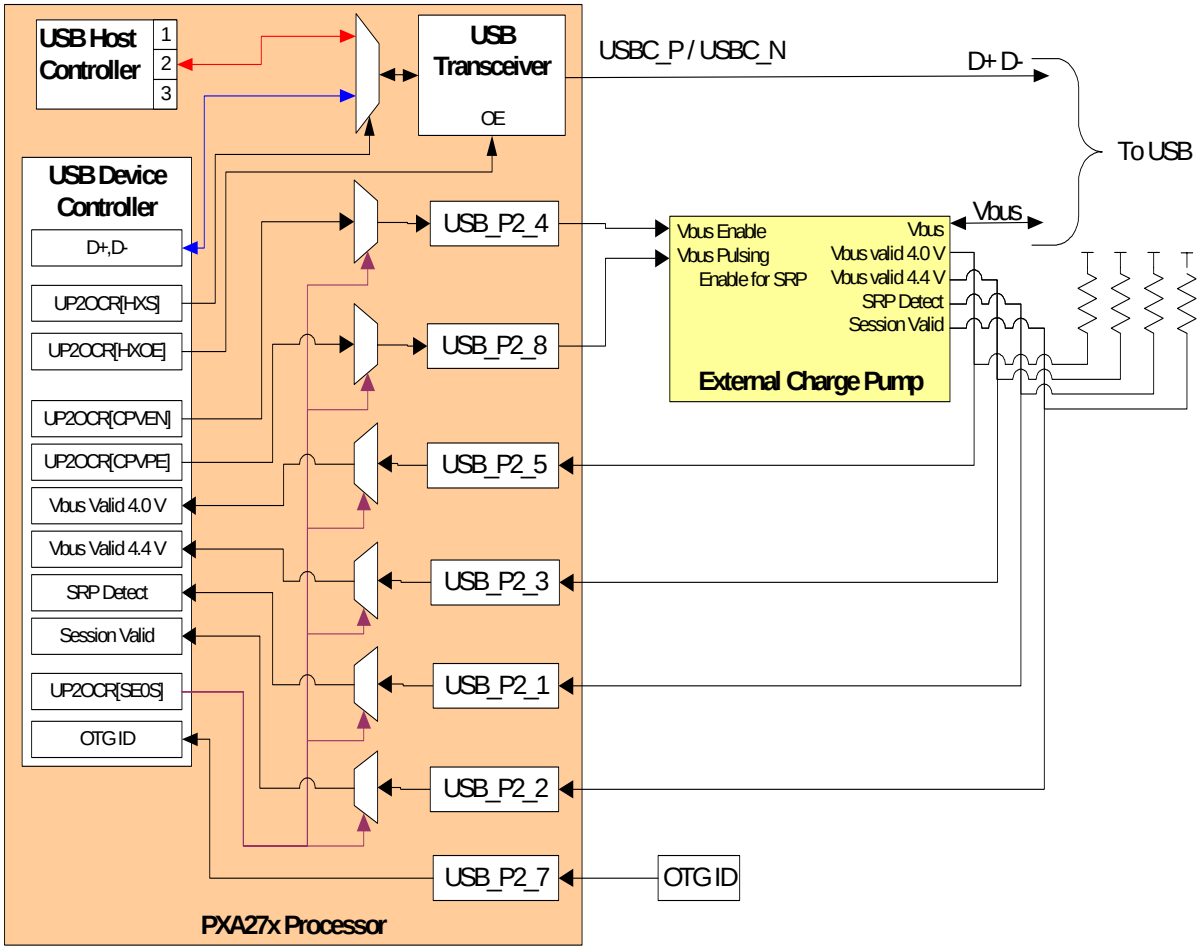
- Charge pump Vbus enable (CPVEN) control output bit
- Charge pump Vbus pulse enable (CPVPE) control output bit

.Additionally, “USB Port 2 Output Control Register (UP2OCR)” provides 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 12-7 for illustration of the PXA27x processor OTG connections to an external charge pump device.

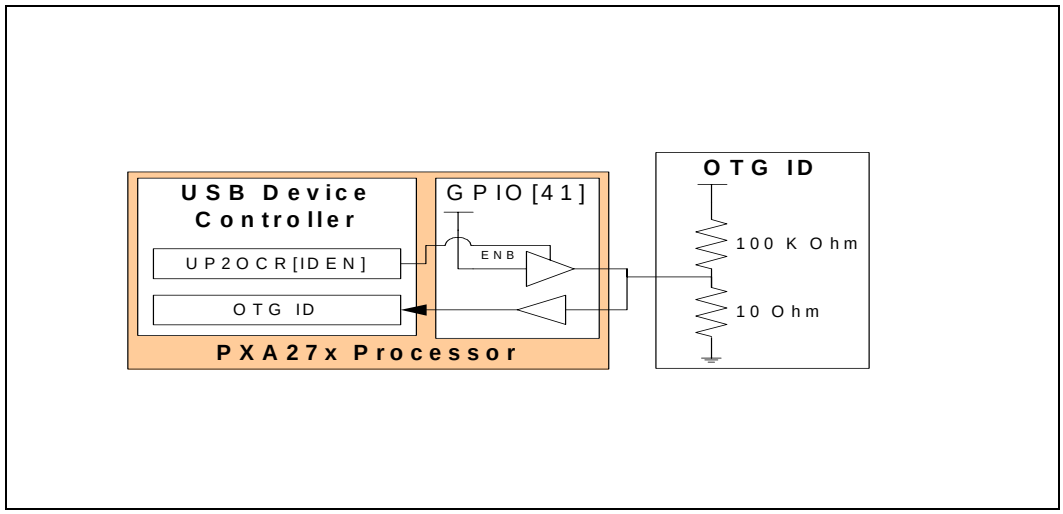
Figure 12-7. Connection to External OTG Charge Pump



12.4.6 OTG ID

The UDC provides OTG ID interface support through GPIO[41]. The UDC provides "USB Port 2 Output Control Register (UP2OCR)" *ID output enable (IDON)* to enable the output for OTG ID reading and the OTG ID interrupt input to detect changes in the OTG ID signal. When IDON is set to 1, it enables the output of GPIO[41] and driven high with a weak output driver so when the OTG ID pin on the USB connector is connected to a 100 KΩ resistor, the OTG ID input results in 1 and when the OTG ID pin is connected to a 10 Ω resistor to ground, the GPIO[41] output driver is unable to drive OTG ID to a 1, resulting in the OTG ID input being a 0. See Figure 12-8 for illustration of the PXA27x processor interface to OTG ID.

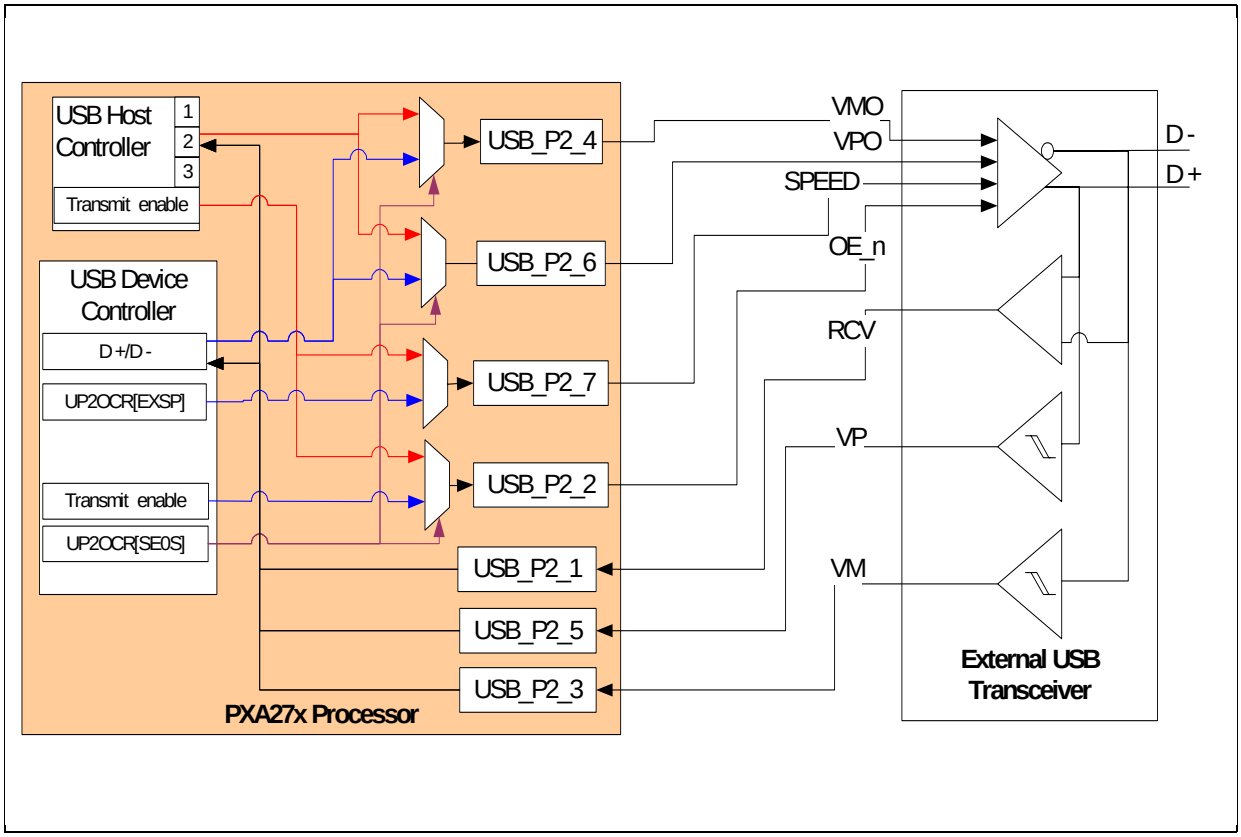
Figure 12-8. Connection to OTG ID



12.4.7 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 12-9 for illustration of the PXA27x processor connection to an external USB transceiver.

Figure 12-9. PXA27x Processor Connection to External USB Transceiver



See Table 12-3 and Table 12-4 for definitions associated with the possible combinations of data while using the External USB Transceiver mode.

Table 12-3. Output to External USB Transceiver

P2_6/P3_6	P2_4/P3_4	Result
0	0	Logic "0"
0	1	"SE0"
1	0	Logic "1"
1	1	"SE0"

Table 12-4. 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	Error

§§

AC ’97 13

This chapter describes guidelines to interface the AC ‘97 controller of Intel® PXA27x Processor Family (PXA27x processor) to an external CODEC device.

13.1 Overview

The AC ’97 controller unit connects external audio integrated circuit devices and CODECs to the PXA27x processor. The AC ’97 controller supports the AC ’97 (Audio CODEC ‘97) specification revision 2.0 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), CODEC register control, and status information.

The AC ’97 port is a bidirectional, serial, pulse code modulated (PCM) digital stream interface. For recording, 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 PXA27x processor retrieves stored audio samples and sends them to the CODEC through the AC-link. The external digital-to-analog converter (DAC) in the CODEC then converts the audio sample to an analog audio waveform. A maximum of two CODECs can be connected to the AC ’97 controller unit.

13.2 Signals

The AC ’97 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 general purpose I/Os (GPIOs). Software must reconfigure the GPIOs to use them as the AC-link. See Table 13-1 for the list and description of the AC-link pins.

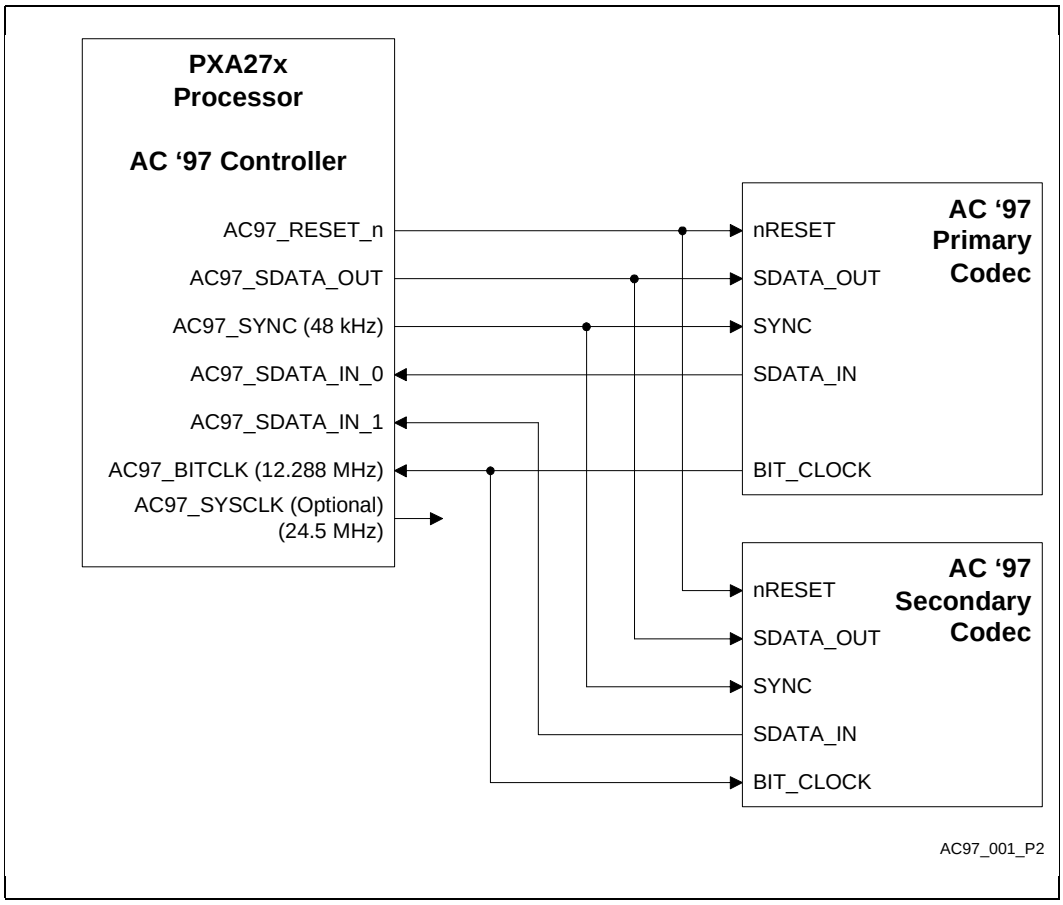
Table 13-1. External Interface to CODECs

Signal Name	Direction	Description Summary
AC97_RESET_n	Output	Active-low CODEC reset The CODEC’s registers are reset when AC97_RESET_n 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
AC97_SYSCLK	Output	Optional 24.5-MHz clock output

13.3 Block Diagram

See Figure 13-1 for the block diagram showing AC ’97 connections. The primary CODEC supplies the 12.288-MHz clock to the AC ’97. This clock is then driven into the AC ’97 controller unit on the PXA27x processor and the AC ’97 secondary CODEC.

Figure 13-1. AC ‘97 Controller to CODEC Block Diagram



13.4 Layout Notes

The AC97_RESET_n signal is asserted when the PXA27x processor enters the sleep or deep sleep modes resulting in the CODEC device being held in reset. This action results in CODEC devices that have wake-up interrupt generating capability (such as touchscreen controllers) not being able to utilize this functionality since the CODEC device is being held in reset. Also, for power consumption considerations, holding a CODEC device in reset may not be its minimal power state.

One possible solution for avoiding having an external CODEC held in reset is to use the GPIO functionality of the multiplexed GPIO113/AC97_RESET_n signal to control the CODEC reset input signal while the PXA27x processor is in sleep mode. This is accomplished by performing these steps sequentially before the PXA27x processor is in sleep mode:

1. Configure the AC97_RESET_n signal as GPIO[113] by setting the GPIO Alternate Function Register 3 Upper field, GAFR3_U[AF113] bit to 0b00.
2. Configure GPIO113 as an output by setting the GPIO Pin-Direction Register 3, GPDR3[PD113] bit.
3. Set the level of GPIO113 to output high by setting the GPIO Pin-Output Set Register 3, GPSR3[PD113] bit.
4. Set the level of GPIO[113] to output high during sleep mode by setting the Power Manager GPIO Sleep-State register, PGSR3[SS113] bit.
5. Enter sleep mode as outlined in Section 3 of the *Intel® PXA27x Processor Family Developer's Manual.*

Programming GPIO Alternate Function Register 3 Upper field, GAFR3_U[AF113], to 0b10 during the wake up sequence restores the AC97_RESET_n functionality to the GPIO113/ AC97_RESET_n signal.

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

- Use a separate power supply for the analog audio portion of the design.
- Place a digital power/ground plane keep-out underneath the analog portion. Use a separate analog ground plane. Create an island inside the keep-out. Connect the digital ground pins of the CODEC to the digital ground. Keep the two ground planes on the same layer, with at least 0.125 inches (3.135 mm) separation between the ground planes.
- Connect the two ground planes underneath the CODEC with a 0 Ω jumper. Add optional Do Not Populate 0 Ω jumpers between analog and digital ground at the power supply. To reduce excessive noise on the board, install the 0 Ω resistor.
- Do not to route digital signals underneath the analog portion. Digital traces must go over the digital ground plane, analog traces over the analog plane.
- Buffer any digital signals to or from the CODEC that go off the board (for example, if the CODEC is on a daughter card).
- Fill the areas between analog traces with copper tied to the analog ground. Fill the regions between digital traces with copper tied to the digital ground.
- 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 AC ‘97’s XTL_IN input, instead of using an external oscillator (or external crystal). This option offers part count, cost, and PCB space saving opportunities for the system designer.

§§

I²S Interface 14

This chapter describes guidelines to interface the Inter IC Sound (I²S or IIS) controller of Intel® PXA27x Processor Family (PXA27x processor) to an external CODEC device.

14.1 Overview

I²S is the name of a protocol defined by Philips Semiconductor for transferring two-channel digital audio signals (digital stereo audio) from one IC device to another. The I²S controller controls the I²S link (I²SLINK), which is a low-power four-pin serial interface for stereo audio. The I²S controller consists of buffers, status registers, control registers, serializers, and counters for transferring digitized audio between the PXA27x processor system memory and an external I²S CODEC. The I²S controller supports the normal I²S and the MSB-justified I²S formats.

The I²S controller records digitized audio by storing the samples in system memory. For playback of digitized audio or production of synthesized audio, the I²S controller retrieves digitized audio samples from PXA27x processor system memory and sends them to a CODEC through the I²SLINK. The external digital-to-analog converter in the CODEC then converts the audio samples into an analog audio waveform. The I²S data are stored to and retrieved from system memory either by the DMA controller or by programmed I/O.

For I²S systems, additional signals controls the external CODEC. Some CODECs use an L3 control bus, which requires three signals for writing bytes into the L3-bus register:

- L3_CLK
- L3_DATA
- L3_MODE

The I²S controller supports the L3 bus protocol using software control of the general-purpose I/O (GPIO) pins. The I²S controller does not provide hardware control for the L3 bus protocol.

14.2 Signals

See Table 14-1 for the list of signals between the I²S and an external CODEC device.

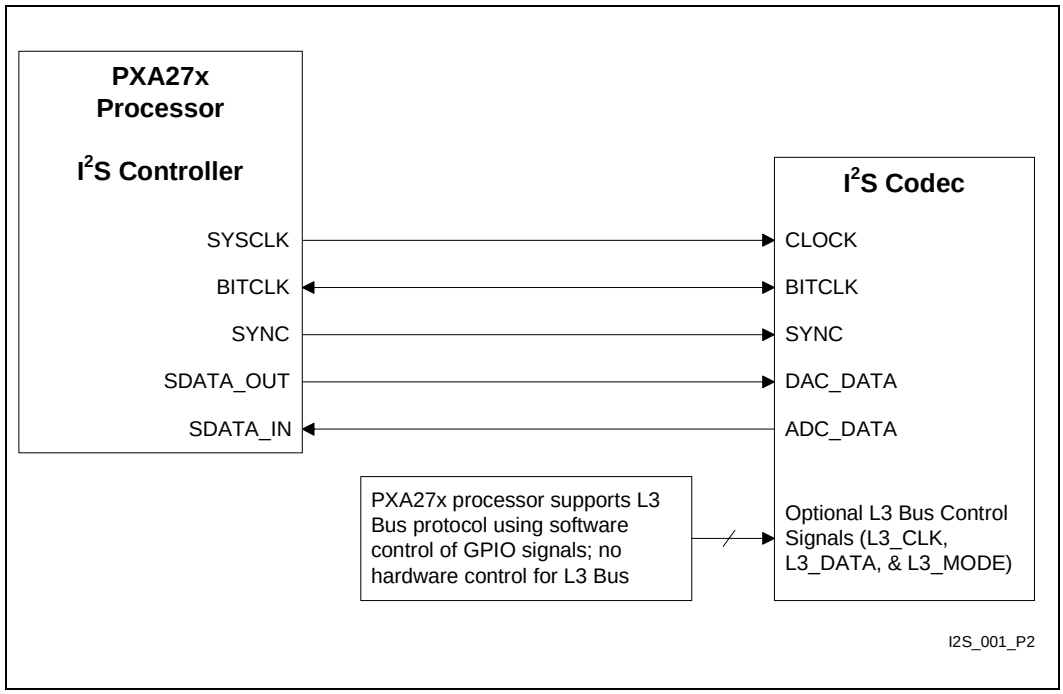
Table 14-1. I²S Controller Interface to CODEC

Name	Direction	Description
SYSCLK	Output	System Clock = BITCLK x 4, used by the CODEC only. The I²S SYSCLK generates a clock frequency between approximately 2 MHz and 12.2 MHz by dividing down the PLL clock with a programmable divisor. This frequency is always 256 times the audio sampling frequency. SYSCLK is driven out of the PXA27x processor I²S controller only if BITCLK is configured as an output. If BITCLK is supplied by the CODEC, the SYSCLK GPIO pin are used for an alternate function.
BITCLK	Input or Output	BITCLK supplies the serial audio bit rate, which is the basis for the external CODEC bit-sampling logic. BITCLK is one-quarter the frequency of SYSCLK and is 64 times the audio sampling frequency. One bit of the serial audio data sample is transmitted or received each BITCLK period. A single serial audio sample comprises a “left” and “right” signal, each containing either 8, 16, or 32 bits. BITCLK are configured either as an input or as an output. If BITCLK is an output, SYSCLK must be configured as an output.
SYNC	Output	SYNC is BITCLK divided by 64, resulting in an 8-KHz to 48-KHz signal. The state of SYNC denotes whether the current serial data samples are left or right channel data.
SDATA_OUT	Output	Serial audio output data to CODEC
SDATA_IN	Input	Serial audio input data from CODEC

14.3 Block Diagram

See Figure 14-1 for the PXA27x processor I²S controller interface block diagram.

Figure 14-1. I²S Controller Interface Block Diagram



14.4 Layout Notes

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

- Use a separate power supply for the analog audio portion of the design.
- Place a digital power/ground plane keep-out underneath the analog portion. Use a separate analog ground plane. Create an island inside the keep-out. Connect the digital ground pins of the CODEC to the digital ground. Keep the two ground planes on the same layer, with at least 0.125 inches (3.135 mm) separation between the ground planes.
- Connect the two ground planes underneath the CODEC with a 0 Ω jumper. Add optional Do-Not-Populate populate 0 Ω jumpers between analog and digital ground at the power supply. Reduce excessive noise on the board by installing the 0 Ω resistor.
- Do not route digital signals underneath the analog portion. Ensure the digital traces go over the digital ground plane and analog traces go over the analog plane.
- Buffer any digital signals to or from the CODEC that go off the board (for example, if the CODEC is on a daughter card).
- Fill the areas between analog traces with copper tied to the analog ground. Fill the regions between digital traces with copper tied to the digital ground.
- Locate the decoupling capacitors for the analog portion as close to the CODEC as possible.

14.5 Modes of Operation Overview

The I²S controller has two modes of operation:

- The PXA27x processor I²S controller supplies the BITCLK as an output signal to the external CODEC.
- The external CODEC supplies BITCLK as an input signal to PXA27x processor I²S controller.

14.5.1 PXA27x Processor Provides BITCLK signal to CODEC

In this mode of operation, the BITCLK signal is an output to the CODEC; SYSCLK must be configured as an output.

14.5.1.1 Signals

See Table 14-2 for the list of the signals between the I²S and an external CODEC device when the PXA27x processor provides BITCLK to the CODEC.

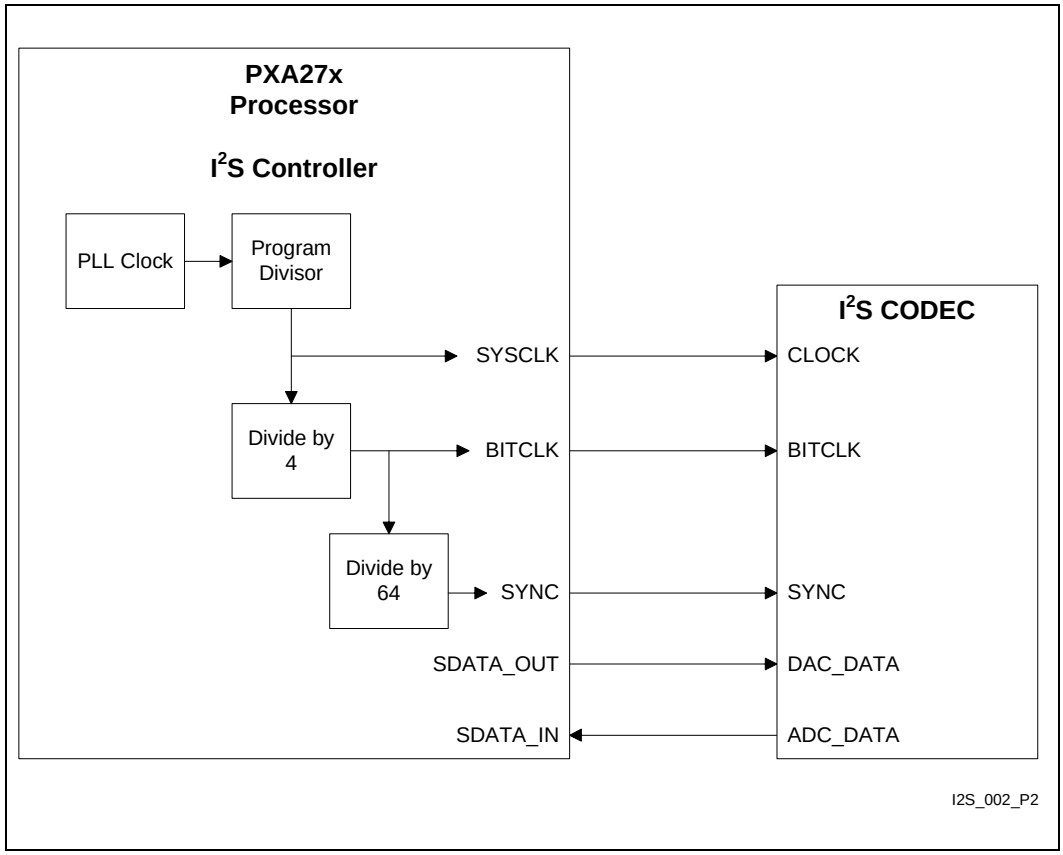
Table 14-2. I²S Controller Interface to CODEC (PXA27x Processor Providing BITCLK to CODEC)

Name	Direction	Description
SYSCLK	Output	System Clock = BITCLK x 4 used by the CODEC only SYSCLK is driven out of the I²S controller only if BITCLK is configured as an output
BITCLK	Output	Bit-rate clock = SYNC x 64 SYSCLK must be configured as an output
SYNC	Output	Left/Right identifier SYNC is BITCLK divided by 64
SDATA_OUT	Output	Serial audio output data to CODEC
SDATA_IN	Input	Serial audio input data from CODEC

14.5.1.2 Block Diagram

See Figure 14-2 for the I²S controller interface block diagram signals between the I²S and an external CODEC device when the PXA27x processor provides BITCLK to the CODEC.

Figure 14-2. PXA27x Processor Provides BITCLK



14.5.2 CODEC Provides BITCLK Signal to PXA27x Processor

In this mode of operation, the BITCLK signal is an input to the PXA27x processor from the CODEC.

14.5.2.1 Signals

See Table 14-3 for the list of the signals between the I²S and an external CODEC device when the CODEC provides BITCLK to the PXA27x processor.

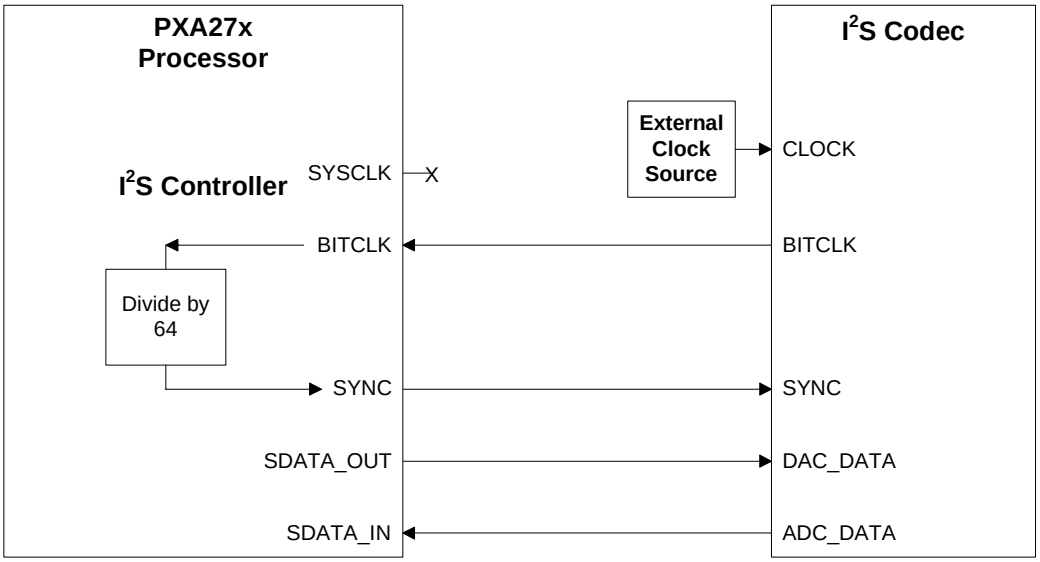
Table 14-3. I²S Controller Interface to CODEC (CODEC Providing BITCLK to the PXA27x Processor)

Name	Direction	Description
SYSCLK	---	GPIO available for alternate function use
BITCLK	Input	Bit-rate clock provided by CODEC
SYNC	Output	Left/Right identifier SYNC is BITCLK divided by 64
SDATA_OUT	Output	Serial audio output data to CODEC
SDATA_IN	Input	Serial audio input data from CODEC

14.5.2.2 Block Diagram

See Figure 14-3 for the block diagram showing I²S controller interface signals between the I²S and an external CODEC device when the PXA27x processor receives BITCLK from the CODEC.

Figure 14-3. PXA27x Processor Receives BITCLK



§§

MultiMediaCard/SD/SDIO Card Controller 15

This chapter describes guidelines to interface the MultiMediaCard (MMC)/SD/SDIO controller of Intel® PXA27x Processor Family to MMC/SD/SDIO sockets for use with MMC, SD, and SDIO card devices.

15.1 Overview

The MMC/SD/SDIO controller performs the following tasks:

- Acts as a link between the software used to access the PXA27x processor 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 systems[1]

The PXA27x processor MMC/SD/SDIO controller is based on the standards outlined in these specifications:

- *MultiMediaCard System Specification, Version 3.2*
- *SD Memory Card Specification, Version 1.01*
- *SDIO Card Specification, Version 1.0 (Draft 4.0)*

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 PXA27x processor must indicate whether to use MMC, SD Card, SDIO Card, or SPI protocol to communicate with the MMC/SD/SDIO controller.

1. Access the MMC Association's web site at **www.mmca.org** for a detailed description of the MMC system.

15.2 Signals

See Table 15-1 for the description of MMC/SD/SDIO Card controller signal functions.

Table 15-1. Multimedia Card/SD/SDIO Card Controller Interface Signal Summary

Signal	Direction (MMC/SD/SDIO Mode)	Direction (SPI Mode)	Description
MMCLK	Output	Output	MMC and SD/SDIO bus clock
MMCMD	Bidirectional	Output	MMC and SD/SDIO: Bidirectional line for command and response tokens SPI: Output for command and write data
MMDAT<0>	Bidirectional	Input	MMC and SD/SDIO: Bidirectional line for read and write data SPI: Input for response token and read data
MMDAT<1>	Bidirectional	Input	MMC and SD/SDIO: Used for SD/SDIO 4-bit data transfers and to signal SDIO interrupts to the controller SPI: Used to signal SDIO interrupts to the controller
MMDAT<2>/ MMCCS<0>	Bidirectional	Output	SD/SDIO: 4-bit data transfer SPI: CS0 chip select
MMDAT<3>/ MMCCS<1>	Bidirectional	Output	SD/SDIO: 4-bit data transfer SPI: CS1 chip select

15.3 Layout Notes

The PXA27x processor 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 15-2 for information on sockets and devices supported by the MMC/SD/SDIO controller.

Table 15-2. 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 15-3 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 PXA27x processor.

Table 15-3. 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. The PXA27x processor MMC/SD/SDIO controller supports one data bus.
	SDIO	One SDIO Card	Each SDIO Card must have its own data bus. The PXA27x processor 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	MMC/SD/SDIO controller of the PXA27x processor supports these devices: • 2 chip selects • MMCCS<0> • MMCCS<1> The specified devices must be in SPI mode.

The PXA27x processor MMC/SD/SDIO card controller communicates at a maximum data rate of 19.5 Mbps for MMC, 1-bit SD/SDIO, and SPI mode data transfers. The maximum data rate is 78 Mbps for 4-bit SD/SDIO data transfers.

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.
- Serves to protect 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 upon analysis of functionality versus component/board real estate costs.

15.4 Modes of Operation Overview

The MMC/SD/SDIO Card controller has two modes of operation:

- MMC/SD/SDIO mode

  This mode supports the MMC, SD, and SDIO Card communication protocols for data transfers; only one of the protocols is active at a time.

- SPI mode

  This mode provides the capability to communicate with MMC, SD Card, and SDIO Card devices.

15.4.1 MMC/SD/SDIO Mode Using MMC Protocol

This subsection provides guidelines for interfacing MMC/SD/SDIO card controller of the PXA27x processor to MMC devices using the MMC communications protocol.

15.4.1.1 MMC Protocol Signals

See Table 15-4 for description of the MMC signals of the MMC/SD/SDIO Card Controller used with the MMC protocol for interfacing to MMC devices.

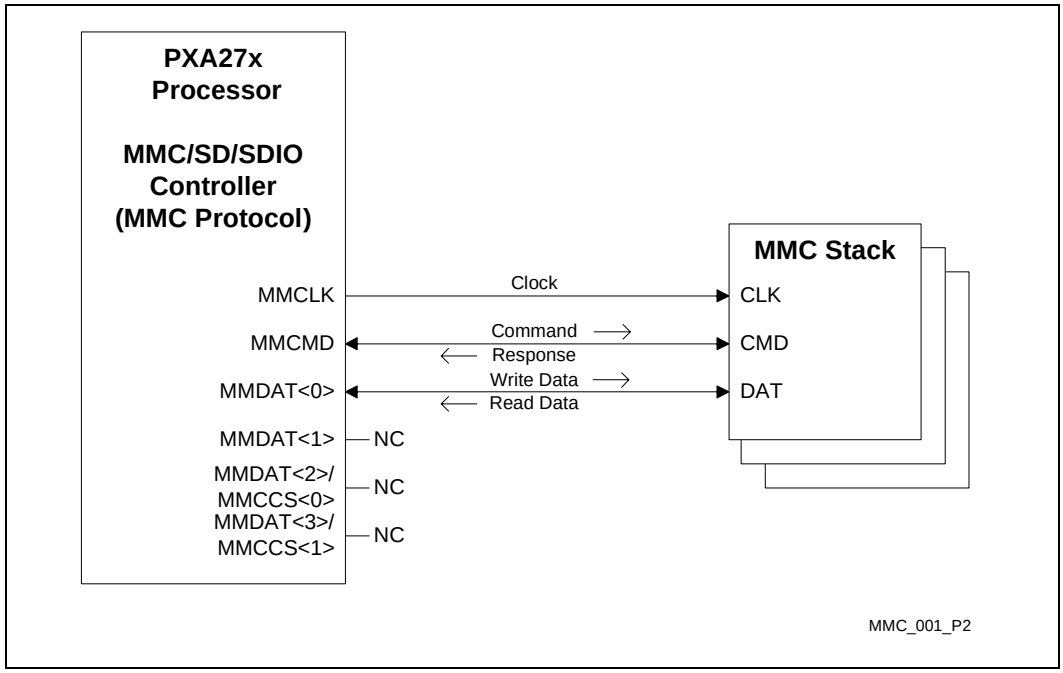
Table 15-4. Multimedia Card Protocol Interface Signals

Signal Name	Direction (MMC Protocol)	Description
MMCLK	Output	MMC bus clock
MMCMD	Bidirectional	Bidirectional signal for MMC command and response tokens
MMDAT<0>	Bidirectional	Bidirectional signal for MMC read and write data

15.4.1.2 MMC Protocol Block and Schematic Diagrams

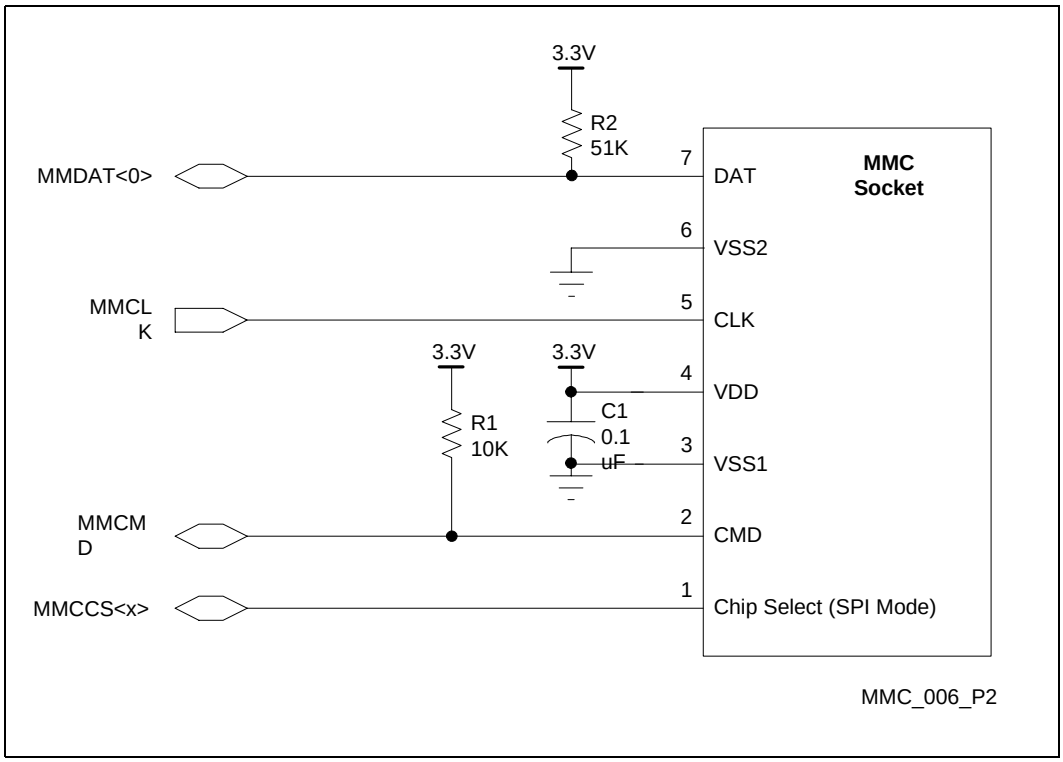
See Figure 15-1 for the block diagram showing the interface between the MMC/SD/SDIO Card controller and an MMC device stack. The block diagram shows high-level signal usage and connectivity when using the MMC communication protocol with MMC devices.

Figure 15-1. MMC Protocol Interface Block Diagram



See Figure 15-2 for the detailed schematic diagram of the interface between the MMC/SD/SDIO Card controller and an MMC device socket.

Figure 15-2. MMC Protocol Interface Schematic Diagram



15.4.1.3 MMC Protocol Layout Notes

See Table 15-5 for the list of pull-up and pull-down resistors required for MMC devices according to their respective specifications.

Table 15-5. MMC Pull-up and Pull-down Resistors

Signal	Pull-up or Pull-down	Min	Max	Remark
MMCMD	pull-up	4.7 KΩ	100 KΩ	Prevents bus floating
MMDAT<0>	pull-up	50 KΩ	100 KΩ	Prevents bus floating

The 0.1 µF capacitor (C1) shown in Figure 15-2 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, other measures must be taken to produce a card detect. Use an SD/SDIO Card socket if a card detect and write protection signal are desired even if only MMC devices are being used. The example circuit shown in Figure 15-4 supports using an MMC device with an SD/SDIO Card socket; MMDAT<1> and MMDAT<2> are not used as the MMC device does not have physical contacts to interface to these SD/SDIO Card socket signals.

15.4.2 MMC/SD/SDIO Mode Using SD or SDIO Protocols

This subsection provides guidelines for interfacing the MMC/SD/SDIO card controller of the PXA27x processor to SD card or SDIO card devices using the SD or SDIO communication protocols respectively.

15.4.2.1 SD and SDIO Protocol Signals

See Table 15-6 for description of the controller signals of the MMC/SD/SDIO Card used with the SD or SDIO protocols for interfacing to an SD Card or SDIO Card respectively.

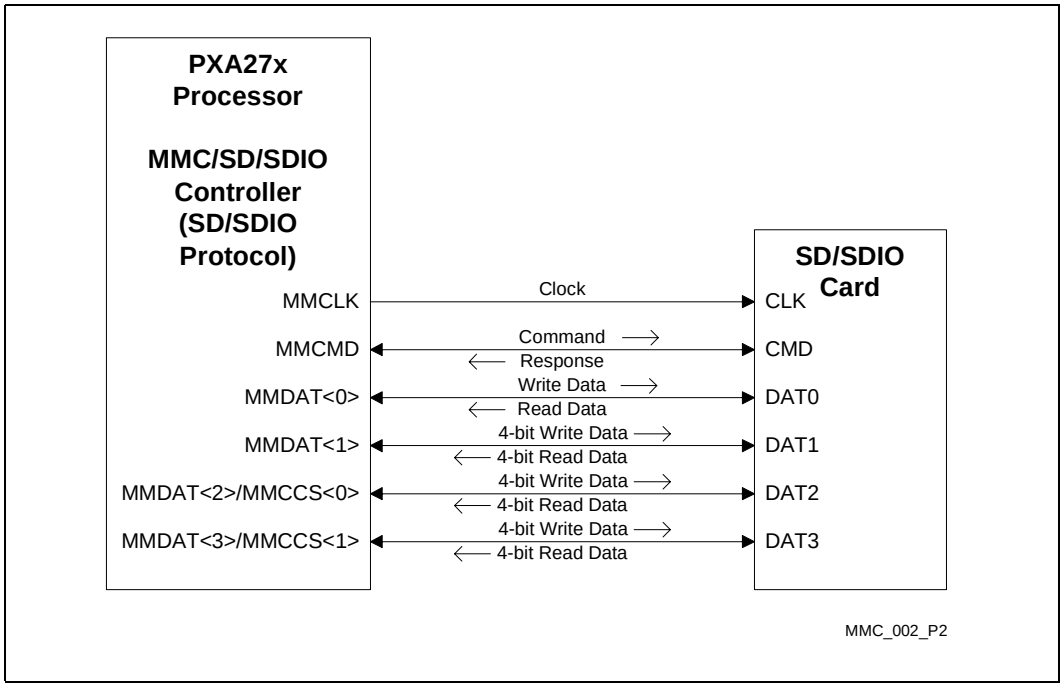
Table 15-6. SD Card and SDIO Card Protocol Interface Signals

Signal Name	Direction (SD/SDIO Protocol)	Description
MMCLK	Output	SD/SDIO bus clock
MMCMD	Bidirectional	Bidirectional signal for SD/SDIO command and response tokens
MMDAT<0>	Bidirectional	Bidirectional signal for SD/SDIO read and write data
MMDAT<1>	Bidirectional	Bidirectional signal for SD/SDIO 4-bit data transfers and to signal SDIO interrupts to the controller
MMDAT<2>/MMCCS<0>	Bidirectional	Bidirectional signal for SD/SDIO 4-bit data transfers only
MMDAT<3>/MMCCS<1>	Bidirectional	Bidirectional signal for SD/SDIO 4-bit data transfers only

15.4.2.2 SD and SDIO Protocol Block and Schematic Diagrams

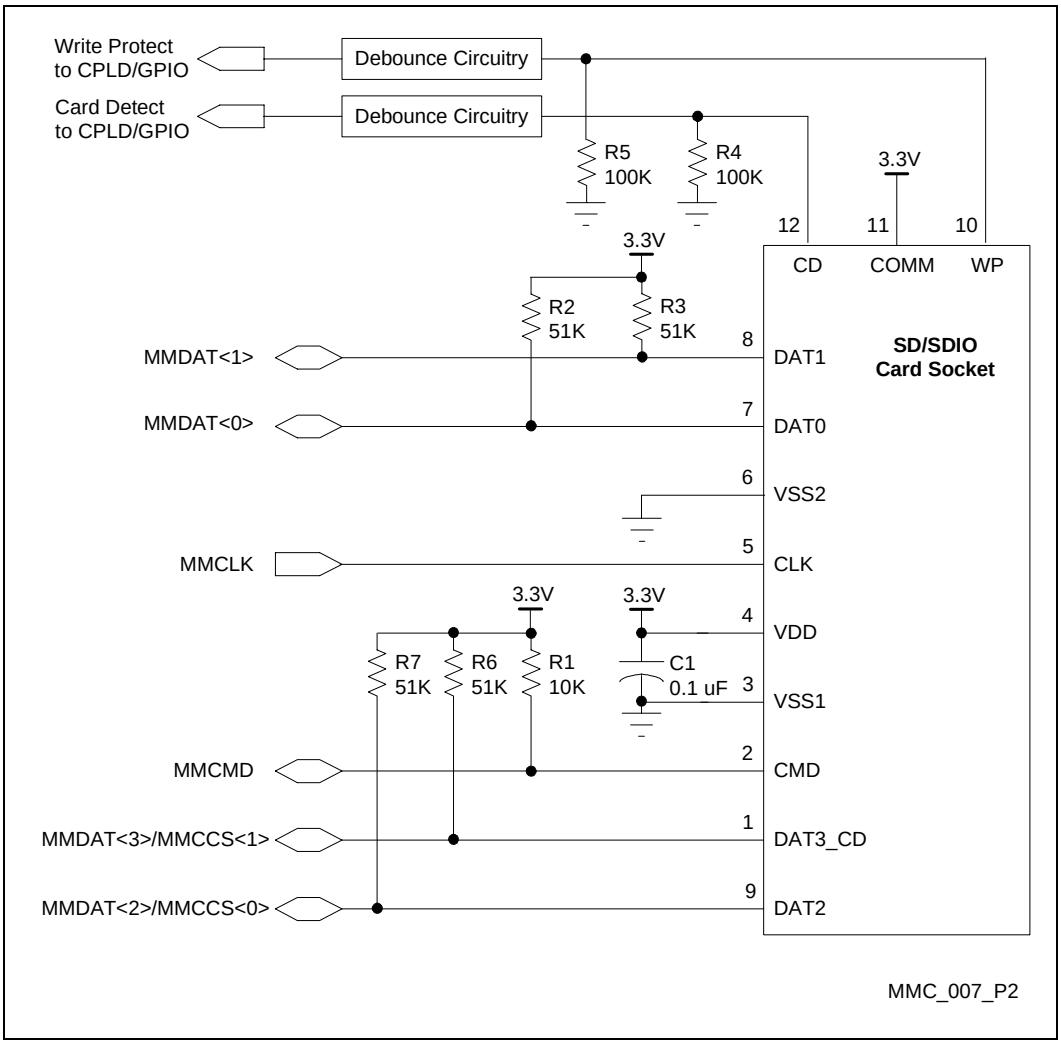
See Figure 15-3 for the block diagram showing the interface between the MMC/SD/SDIO Card controller and an SD Card or SDIO Card. The block diagram shows high level signal usage and connectivity when using the SD or SDIO communication protocols with an SD Card or SDIO Card respectively.

Figure 15-3. SD and SDIO Protocol Interface Block Diagram



See Figure 15-4 for the detailed schematic diagram of the interface between the MMC/SD/SDIO Card controller and an SD/SDIO Card socket.

Figure 15-4. SD and SDIO Protocol Interface Schematic Diagram



15.4.2.3 SD and SDIO Protocol Layout Notes

See Table 15-7 for the list of pull-up and pull-down resistors required for SD Card and SDIO Card devices according to their respective specifications.

Table 15-7. SD/SDIO Card Pull-Up and Pull-Down Resistors

Signal	Pull-up or Pull-down	Min	Max	Remark
MMCMD	pull-up	10 kΩ	100 kΩ	Prevents bus floating
MMDAT<3:0>	pull-up	10 kΩ	100 kΩ	Prevents bus floating
WP[1]	pull-up	—	—	Any value sufficient to prevent bus floating
NOTE: This resistor is shown in the specification, but the value is not specified.				

The 0.1 µF capacitor (C1) shown in Figure 15-4 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 15-4. 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 15-4, 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; the write protect tab 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 15-4, COMM is tied to VCC and WP has a pull-down resistor. This causes a rising edge to occur on the WP signal when the SD (or optional SDIO) Card's 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's CD signal when a device is inserted in the socket. In the example shown in Figure 15-4, the sockets's CD signal has a pull-down resistor. This resistor causes a rising edge to occur on the socket's 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 PXA27x processor is used as a technique for the processor to detect (using the socket's mechanical switch) that an SD/SDIO Card device insertion occurred. To wake up the PXA27x processor from sleep mode using this technique, a GPIO input signal capable of waking up the processor must be selected. Refer to the *Intel® PXA27x Processor Family Developers Manual* 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 capable of indicating to the driver software that the card is detected and write protected respectively.

15.4.3 SPI Mode with MMC, SD Card, and SDIO Card Devices

This subsection provides guidelines for interfacing the MMC/SD/SDIO card controller of the PXA27x processor to MMC, SD card, and SDIO card devices using the SPI communications protocol.

15.4.3.1 SPI Mode Signals

See Table 15-8 for description of the controller signals MMC/SD/SDIO Card used with the SPI protocol for interfacing to MMC, SD Card, and SDIO Card device sockets.

Table 15-8. SPI Protocol Interface Signals

Signal Name	Direction (SPI Mode)	Description
MMCLK	Output	SPI bus clock
MMCMD	Output	Output for SPI command token and write data
MMDAT<0>	Input	Input for SPI response token and read data
MMDAT<1>	Input	Input to signal SDIO interrupts to the controller
MMDAT<2>/**MMCCS<0>**	Output	SPI CS0 chip select
MMDAT<3>/**MMCCS<1>**	Output	SPI CS1 chip select

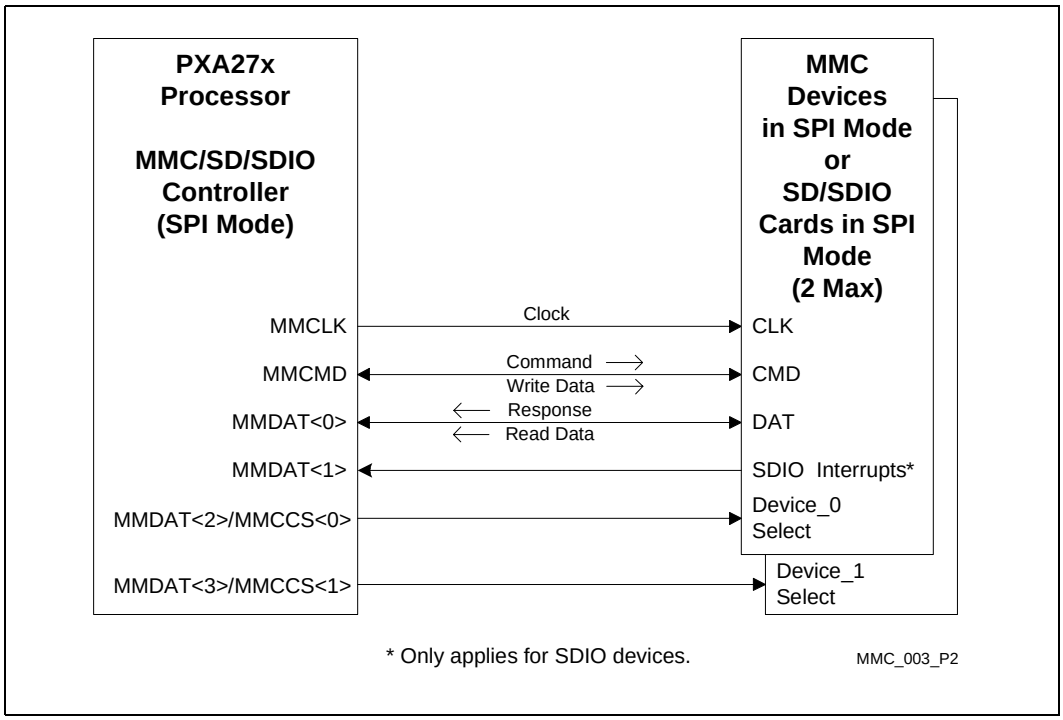
15.4.3.2 SPI Protocol Block and Schematic Diagrams

See Figure 15-5 for the block diagram 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 usage and connectivity when using the SPI communication protocol with the specified devices.

Figure 15-5. SPI Protocol Interface Block Diagram



See Figure 15-2 for illustration of an MMC socket-based schematic diagram capable of supporting SPI mode communication with an MMC device.

See Figure 15-4 for illustration of an SD/SDIO Card socket-based schematic diagram capable of supporting SPI mode communication with an SD or SDIO Card.

15.4.3.3 SPI Protocol Layout Notes

The PXA27x processor signals MMCLK, MMCMD, and MMDAT<0> are connected in parallel to both device sockets for dual device SPI mode operation. MMCCS<0> and MMCCS<1> are used as chip selects only in SPI mode and connect separately to the respective chip select signals of the socket.

§§

Baseband Interface 16

16.1 Overview

This chapter describes the interface between Intel® PXA27x Processor Family (PXA27x processor) and a cellular baseband processor. PXA27x processor provides a high performance, scalable baseband interface to Intel baseband processors described in a later revision of this document.

However, PXA27x processor also provides other standard peripheral interfaces that support connection to a number of wireless baseband subsystems from other manufacturers. For example, in applications where the RF subsystem looks like a “wireless modem” to PXA27x processor, an asynchronous UART connection supports full duplex communication up to 921.6 kbps. In applications which require a faster synchronous interface, one or two SSP ports synchronized to an external network clock provide another more flexible alternative for data rates up to 13 Mbps. The PXA27x processor also provides the capability to run timer channels synchronous with the network clock.

The PXA27x processor is configured to utilize both MSL and PCMCIA simultaneously. However, this configuration impacts the usage of the LCD and MSL as described here:

- Two of the LCD pins are required which prevents the use of 18-bit LCD.
  GPIO [87:86] have alternate functions for:
  - PCMCIA_nPCE1 and PCMCIA_nPCE2
  - L_DD[16] and L_DD[17]

  Therefore, enabling the PCMCIA signals renders the GPIO [87:86] unusable for LCD connection.
- The MSL is only used in a 1-bit configuration.
  GPIOs 48, 50 and 51 have alternate functions for the three upper most MSL data bits:
  - BB_OB_DAT[3:1] and BB_IB_DAT[3:1]
  - PCMCIA_nPCE, PCMCIA_nPIOR, and PCMCIA_nPIOW

  Therefore, enabling the GPIOs 48, 50, and 51 for PCMCIA support leaves only a single MSL data bit to use.
- The configuration allows the system to have:
  - A PCMCIA interface
  - A 1-bit MSL link
  - A 16-bit LCD connection

§§

Memory Stick Host Interface 17

17.1 Overview

Intel® PXA27x Processor Family (PXA27x processor) has a memory stick controller which allows control and communication with a memory stick. This feature provides a small, plug-able standardized form factor that allows for convenient data transfer. This is typically done through flash memory on the memory stick.

17.2 Signals

The memory stick signals are implemented through the GPIOs of the PXA27x processor. Refer to GPIO alternate function table in the GPIO section of the *Intel® PXA27x Processor Family Developers Manual* for the GPIO assignments of the memory stick signals.

The memory stick controller of the PXA27x processor controls the signals shown in Table 17-1.

Table 17-1. Memory Stick Host Interface Signal List

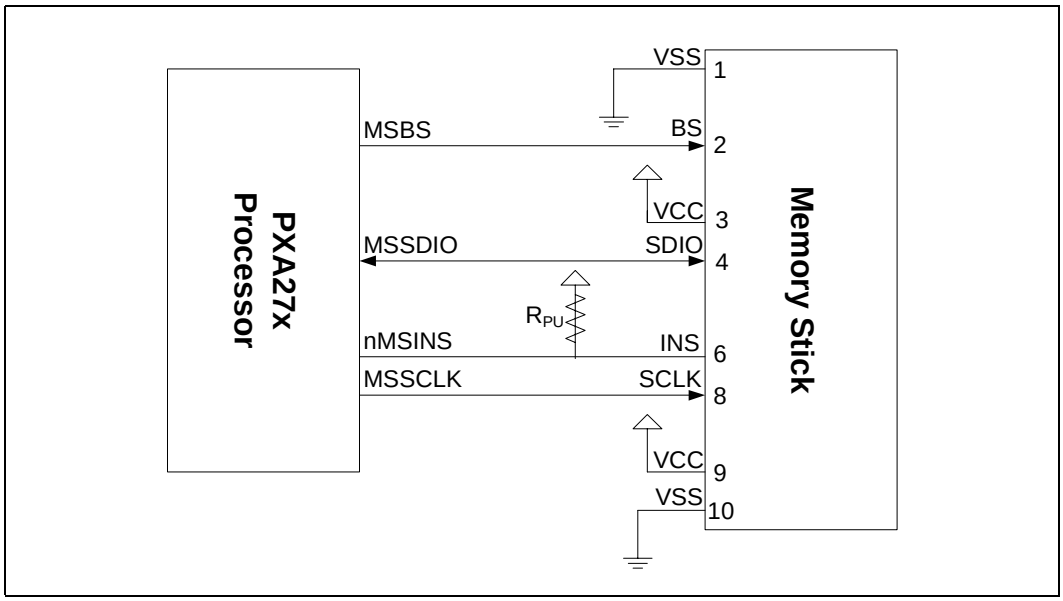
Signal Name	Type	Description
MSBS	Output	Serial protocol bus state signal Referred to as BS in the *Sony Memory Stick Standard, Format Specification Version 1.3*
MSSDIO	Bidirectional	Serial protocol data signal Referred to as SDIO in the *Sony Memory Stick Standard, Format Specification Version 1.3*
nMSINS	Input	Stick insertion/extraction detect- terminal Referred to as INS in the *Sony Memory Stick Standard, Format Specification Version 1.3*
MSSCLK	Output	Serial protocol clock signal Referred to as SCLK in the *Sony Memory Stick Standard, Format Specification Version 1.3*

17.3 Schematic/Block Diagram

The implementation of a memory stick socket is specified by the memory stick standard. Refer to the *Sony* Memory Stick Standard, Format Specification Version 1.3* for more information.

Figure 17-1 is a typical implementation of a memory stick socket with the PXA27x processor. Per the *Sony Memory Stick Standard, Format Specification Version 1.3*, the pull-up resistor R_{PU} must be present to allow the memory stick host controller to detect an insertion or removal of a memory stick. This pull-up resistor is external to both the memory stick socket and the PXA27x processor. As this resistor is connected between VCC and VSS, use a high value resistor. A typical value used is 1.0 MΩ or higher.

Figure 17-1. Memory Stick Implementation Block Diagram



17.4 Layout Notes

The maximum switching frequency of the memory stick is 20 MHz. While it is important to use good layout practices, it does allow for some flexibility in the design. Follow all general layout practices. The *Sony Memory Stick Standard, Format Specification Version 1.3* incorporates several design concepts that make a memory stick application more robust, such as dual pins for both VSS and VCC. Use of these concepts helps avoid any problems.

SCLK switches at a much higher frequency than any of the other signals. To help avoid spurious transactions of the other signals, ensure SCLK is physically and electrically isolated from BS and SDIO.

Each VSS and VCC pin from the socket must be separately routed to the appropriate power/ground plane. Do not simply connect these pins at the socket.

The *Sony Memory Stick Standard, Format Specification Version 1.3* requires that when a memory stick is inserted VSS is the first pin to make contact. The specification also requires INS to be the last pin to make contact upon insertion.

§§

Keypad Interface 18

This chapter describes the procedures for interfacing with the keypad controller of Intel® PXA27x Processor Family (PXA27x processor).

18.1 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 direct keypad interface supports the direct keys and the rotary encoders used to implement keys such as scroll keys 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 in the keypad section of the *Intel® PXA27x Processor Family Developers Manual*.

The direct keypad block supports eight input pins, whereas the matrix keypad block supports eight output and eight input pins. 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.

All references to registers are documented in the *Intel® PXA27x Processor Family Developers Manual* unless otherwise noted. The following information provides examples for specific configurations. This section has not attempted to document all possible configuration, just those configuration that are more commonly used.

18.2 Signals

This section describes the keypad interface signals. See Table 18-1 for summary of the signals.

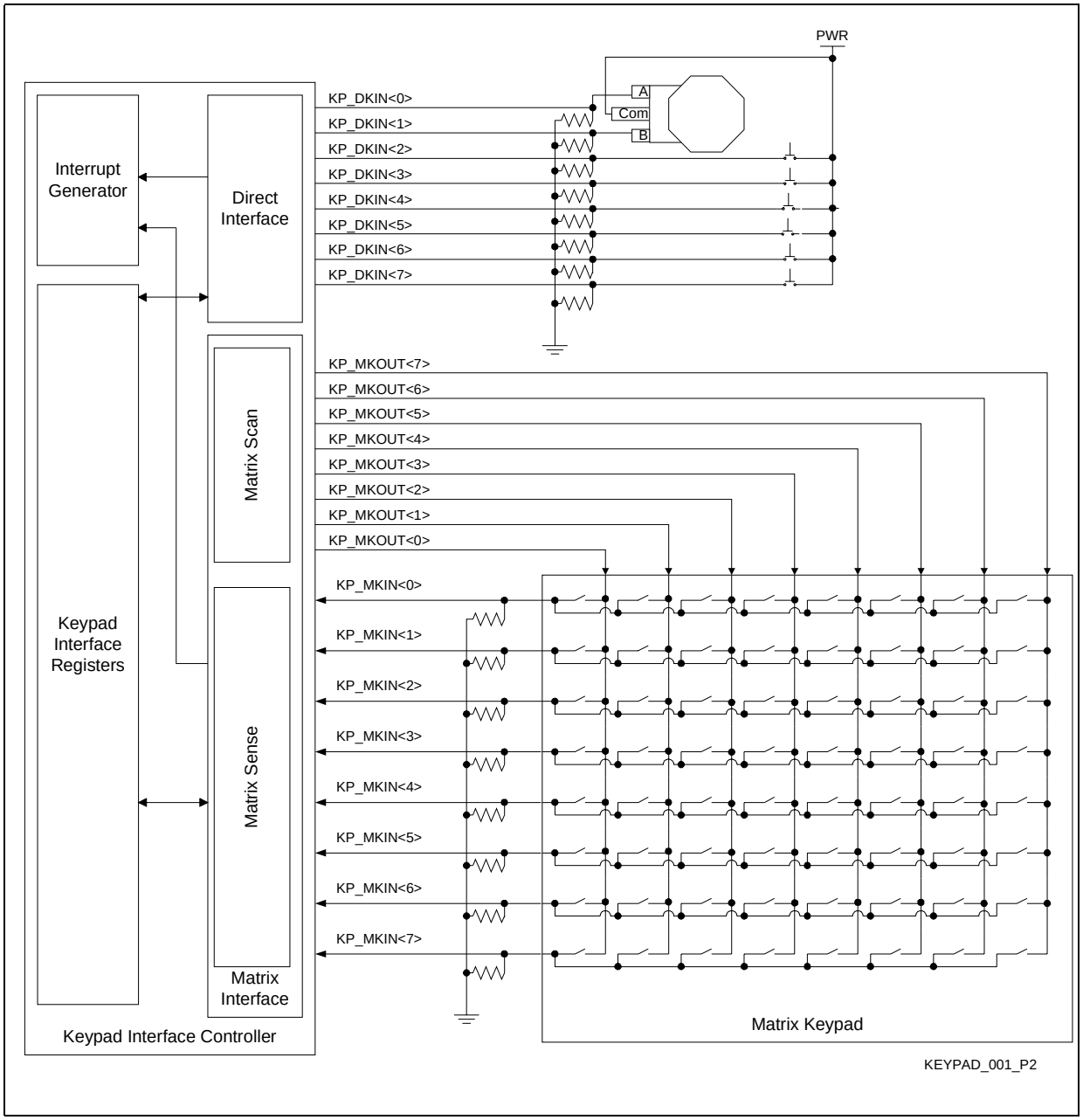
Table 18-1. Interface Signals Summary

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.
KP_MKOUT<7:0>	Output	Matrix Key Outputs – The keypad interface sends scan signals 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.

18.3 Block Diagram

See Figure 18-1 for illustration of one of many possible configurations when interfacing to the PXA27x processor keypad controller. This specific configuration shows the keypad controller interfaced to the maximum size matrix keypad of 64 keys and 6 direct keys with a single rotary encoder.

Figure 18-1. Keypad Interface Block Diagram



18.4 Layout Notes

This section has information on the best method for connecting to the keypad interface with a pull-down resistor and appropriate implementation during standby and sleep mode.

18.4.1 Recommended Pull-down Resistors

The recommended pull-down resistor value to minimize power usage is 100 KΩ. The recommendation is not to have a pull-up resistor because the value needed to overpower the internal pull-down resistance during stand-by and sleep mode would consume excessive power during normal operation mode.

18.4.2 Alternate Function During Standby and Sleep Mode

During the transition into standby or sleep mode, the matrix keypad and direct key signals change to their respective GPIO function. The operation of these pins are determined by their corresponding GPIO settings.

18.4.3 Reduce Power During Standby and Sleep Mode

Using the appropriate GPIO Pin-Direction Register (GPDR), the KP_DKIN and KP_MKIN signals, interfacing to the keypad controller, must be configured as input signals to eliminate excessive power usage caused by driving pull-down resistors. The KP_MKOUT signals used to interface to a matrix keypad must be configured as output signals to eliminate excessive power usage caused by having floating inputs on the KP_MKOUT signals.

18.4.4 Using the Keypad Signals to Wake-up from Standby and Sleep Mode

It is possible to use the keypad signals to wake up the PXA27x processor from standby or sleep modes by configuring the Power Manager Keyboard Wake-Up Enable Register (PKWR). After wake-up, 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.

18.4.5 How to Enable Specific Combinations of Direct Keys

The enabling bit field KPC[DKN] implies that all KP_DKINx signals must be enabled sequentially from KP_DKIN0 to KP_DKIN7. 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 PXA27x processor package by using the PXA27x processor GPIO Alternate Function Registers (GAFR0/1/2/3). By not mapping a specific direct input signal to the PXA27x processor package, a value of zero is read in and the input is not left floating. Leaving the input floating causes excessive use of power.

For a given configuration of the PXA27x processor 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 is guaranteed a logic 0 on them all the time, denoting no activity on direct keys 2 and 3.
- The rest of the direct key are utilized as inputs KP_DKIN<1:0> and KP_DKIN<7:4> and connected with rotary encoders or direct keys. By specifying the number of direct keys in the Keypad Control register as 8 (KPC[DKN] = “111”), a logic 0 on direct key inputs 2 and 3 is guaranteed and no activity is detected on them.

For details about GPIO configuration, refer to the GPIO section of the *Intel® PXA27x Processor Family Developers Manual*.

18.4.6 Interfacing to a Matrix Keypad

When interfacing to a matrix keypad, the KP_MKINx and KP_MKOUTx signals must be connected sequentially from KP_MKIN0 to KP_MKIN7 and from KP_MKOUT0 to KP_MKOUT7. The number of columns in the keypad signifies the highest order KP_MKOUTx signal minus one to be used. The number of rows in the keypad signifies the highest order KP_MKINx signal minus one to be used.

18.5 Modes of Operation Overview

The following subsections are examples showing how to use the keypad controller with either one, two or no rotary encoders. Some example have different size matrix keypads.

18.5.1 Keypad Matrix and Direct Keys and No Rotary Encoder

This example discusses one of many possible configurations when interfacing to the PXA27x processor keypad controller. This specific configuration shows the keypad controller interfaced to a 4 x 4 matrix keypad of 16 keys, 8 direct keys with no rotary encoder.

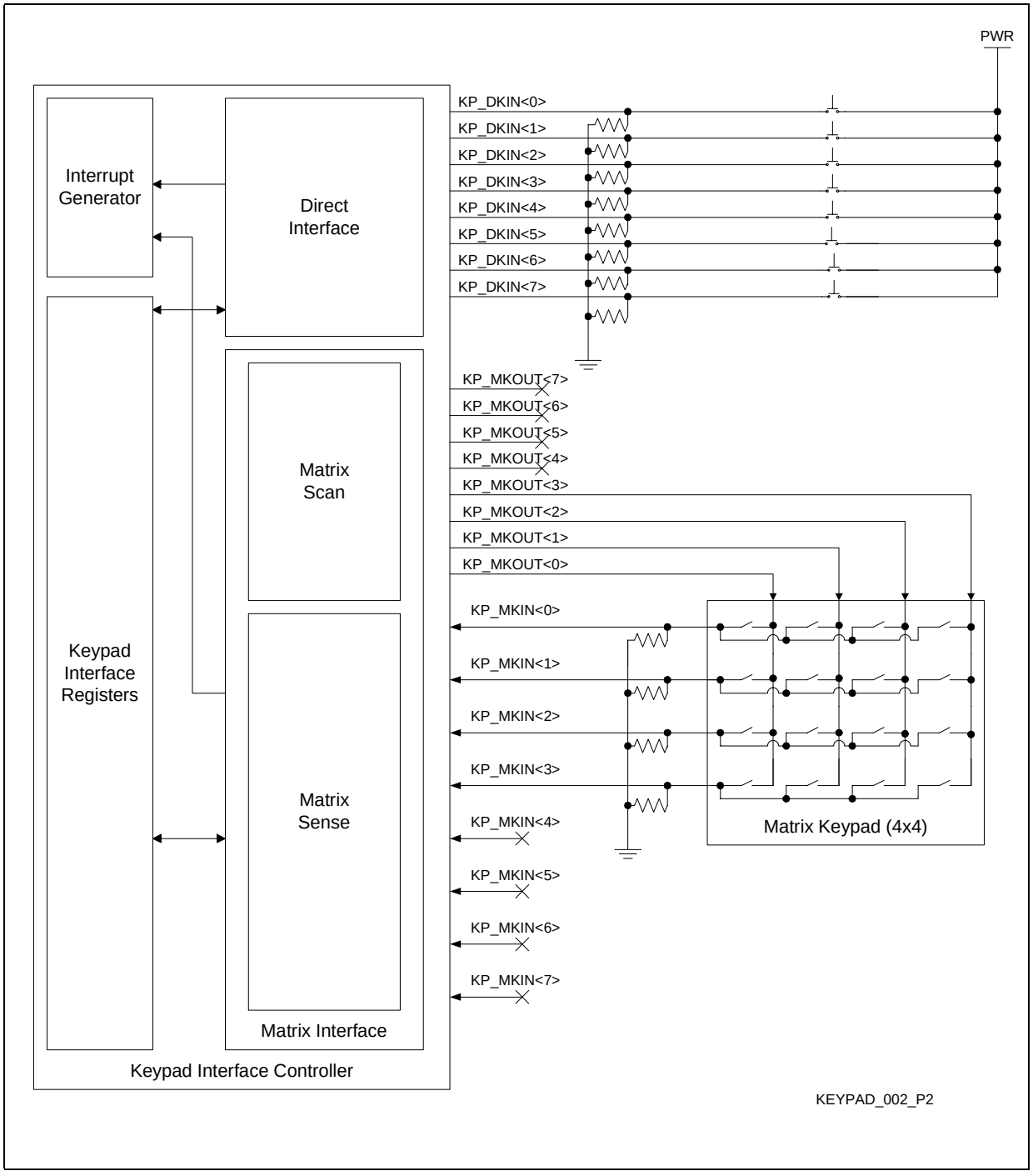
18.5.1.1 Signals

The signal definitions do not change, just the signal usage. See Figure 18-2 for detailed signal information.

18.5.1.2 Block Diagram

See Figure 18-2 for example of a possible configuration (with no rotary encoder) when interfacing to the PXA27x processor keypad controller. The recommended value for the pull-down resistors is 100 KΩ. The voltage reference must be at the same level as VCC_IO.

Figure 18-2. Keypad Matrix and Direct Keys Block Diagram (with No Rotary Encoder)



18.5.2 Keypad Matrix and Direct Keys with One Rotary Encoder

This example discusses one of many possible configurations when interfacing to the PXA27x processor keypad controller. This specific configuration shows the keypad controller interfaced to a 3 x 4 matrix keypad of 12 keys, 6 direct keys, and 1 rotary encoder.

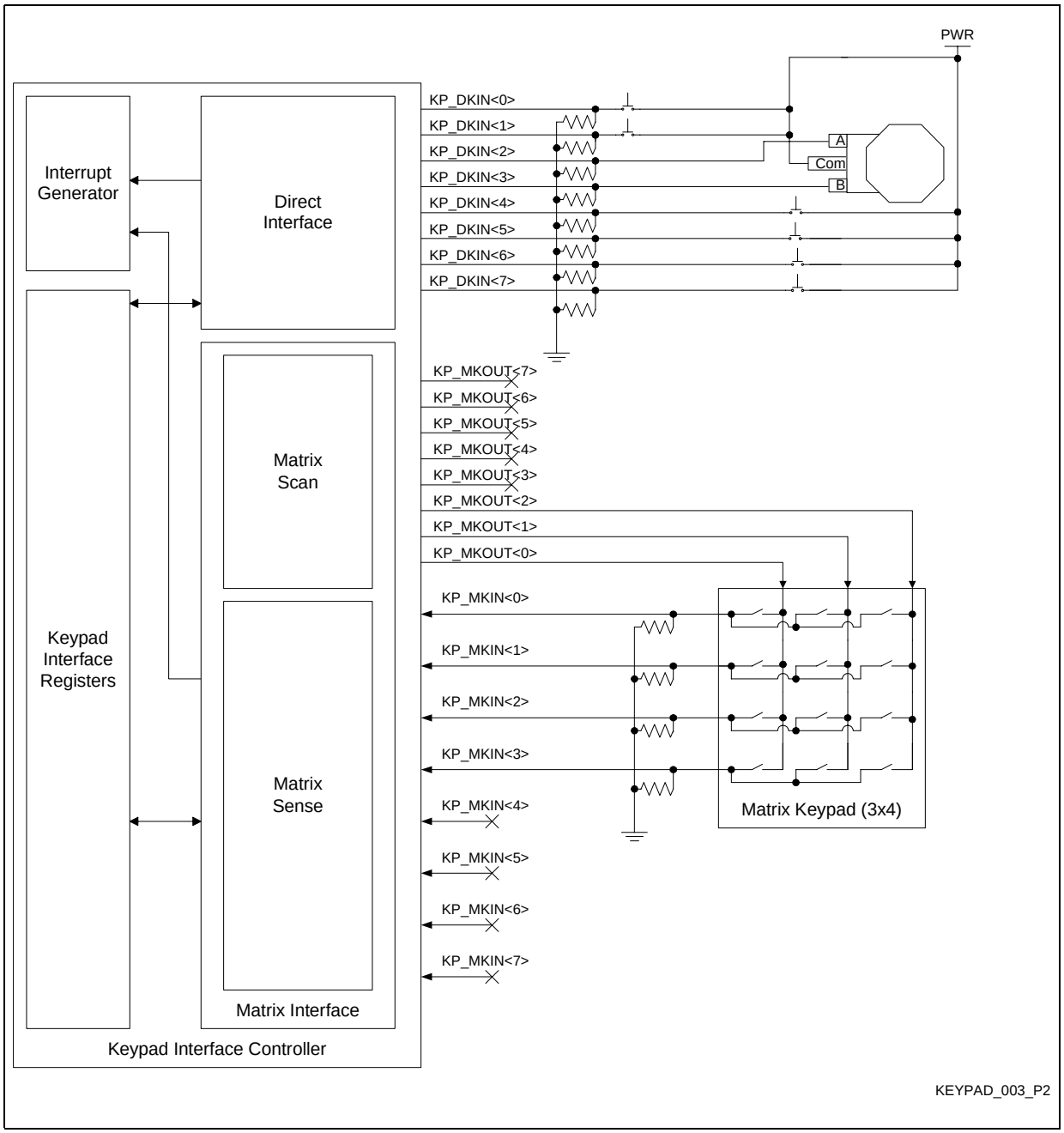
18.5.2.1 Signals

The signal definitions do not change, just the signal usage. See Figure 18-3 for detailed signal information.

18.5.2.2 Block Diagram

See Figure 18-3 for example of a possible configuration (with of one rotary encoder) when interfacing to the PXA27x processor keypad controller. The recommended value for the pull-down resistors is 100 KΩ. The voltage reference must be at the same level as VCC_IO.

Figure 18-3. Keypad Matrix and Direct Keys Block Diagram (with One Rotary Encoder)



18.5.3 Keypad Matrix and Direct Keys with Two Rotary Encoders

This example discusses one of many possible configurations when interfacing to the PXA27x processor keypad controller. This specific configuration shows the keypad controller interfaced to a 4 x 4 matrix keypad of 16 keys, four direct keys and two rotary encoders.

The use of two rotary encoders limits the number of direct key signals (KP_DKINx) to four. Therefore, it is possible to use the matrix keypad to add additional direct key switches if the full matrix keypad is not used. This implementation takes advantage of the internal debounce logic for addition direct keys and eliminates the requirement of pull-down resistors and external debounce logic need to support additional direct keys.

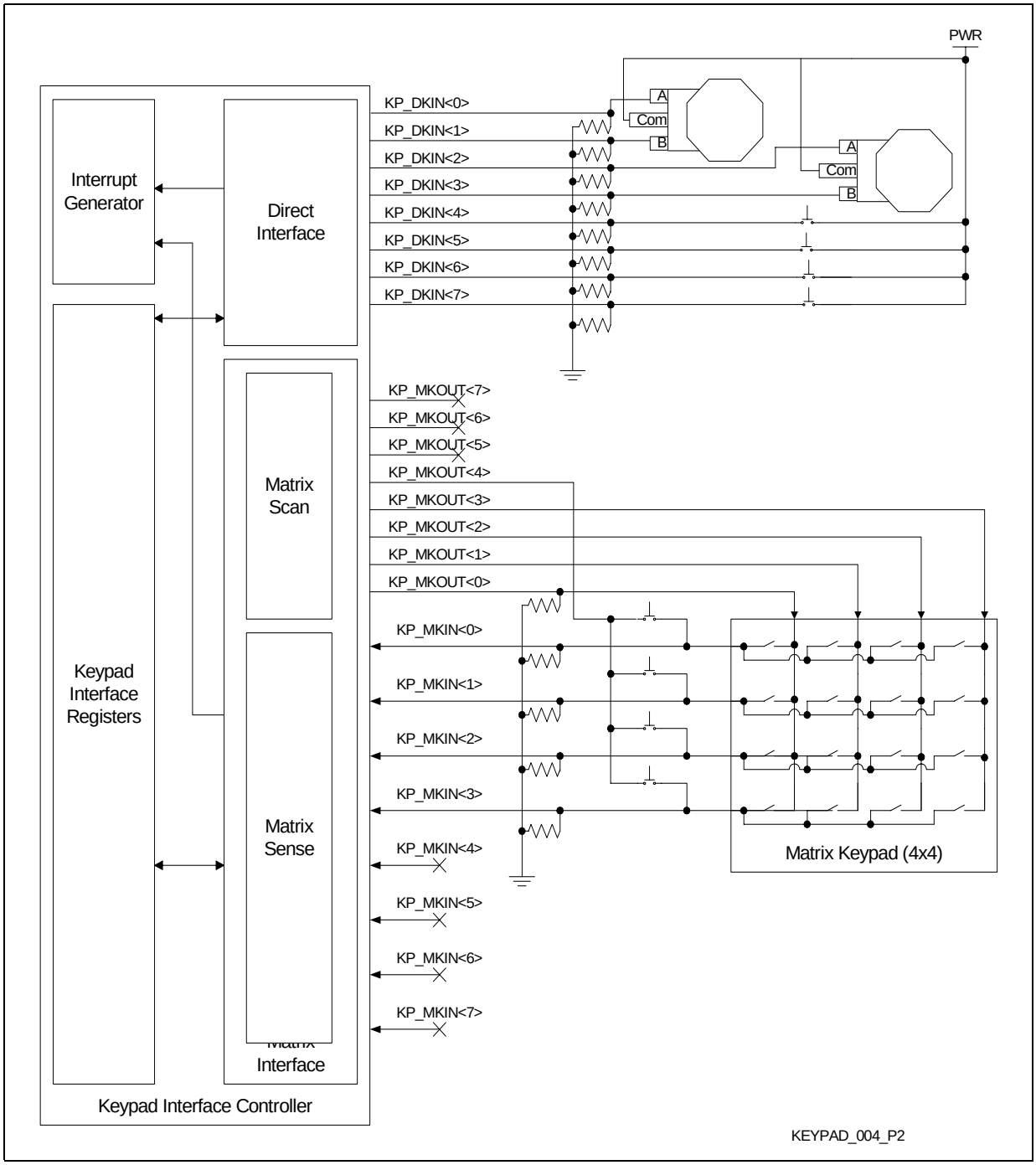
18.5.3.1 Signals

The signal definitions do not change, just the signal usage. See Figure 18-4 for detailed signal information.

18.5.3.2 Block Diagram

See Figure 18-4 for example of a possible configuration (with two rotary encoders) when interfacing to the PXA27x processor keypad controller. The recommended value for the pull-down resistors is 100 KΩ. The voltage reference must be at the same level as VCC_IO.

Figure 18-4. Keypad Matrix and Direct Keys Block Diagram (with Two Rotary Encoders)



§§

USIM Controller Interface 19

This chapter describes the universal subscriber identity module (USIM) of Intel® PXA27x Processor Family (PXA27x processor) controller interface guidelines.

19.1 Overview

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

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

The USIM interface is composed of eight major components:

- Registers
- Card interface
- Receiver first in/first out (FIFO)
- Transmitter FIFO
- Receiver
- Transmitter
- Clock generator
- Baud-rate generator

The transmitter converts the byte stored in its FIFO to serial output as specified by the standard. The receiver performs the reverse action and stores the data in the Receiver FIFO. FIFOs holds up to 16 bytes. The receiver FIFO 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, FIFO output is either inverted or left as is, according to the session convention. In this manner, the program does not have to perform 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 19-1.

19.2 Signals

This section describes the PXA27x processor I/O signals used by the USIM interface. See Figure 19-1 for illustration of the connectivity between the PXA27x processor and the external USIM card.

19.2.1 PXA27x Processor USIM Interface Signals

See Table 19-1 for the list of all PXA27x processor signals between the USIM interface and the USIM card.

Table 19-1. PXA27x Processor Interface Signals Summary (Sheet 1 of 2)

Name	Type	Description
UIO	Bidirectional	USIM I/O Data. UIO – receive and transmit data connection The bidirectional signal is connected directly to the off-chip USIM card. When the PXA27x processor transmits a logic 0, the UIO signal is driven low. For transmitting a logic 1, the UIO line is pulled-up by the 20 kΩ pull-up resistor. The USIM card also acts as a pull-down on the UIO line. **NOTE:** Leave the UIO signal disconnected if USIM controller is never intended to be used.
UVS0[1]	Output	Select for power transistor controlling voltage level supplied for card Active high signal, assertion of UVS0 must ground (0V) VCC on the USIM card VCC power-supply signal
nUVS1[1]	Output	Select for power transistor controlling voltage level supplied for card nUVS1 assertion selects 1.8 V on the USIM card VCC power-supply signal
nUVS2[1]	Output	Select for power transistor controlling voltage level supplied for card nUVS2 assertion selects 3.0 V on the USIM card VCC power-supply signal
UCLK	Output	Clock supplied to the USIM card This signal connects directly to the card CLK signal. The card cannot use any other clock.
nURST	Output	Reset signal to card The card is reset when this output is asserted
UDET[2]	Output	USIM Detection for card present
UEN[2]	Output	USIM Enable for VCC_USIM connection

Table 19-1. PXA27x Processor Interface Signals Summary (Sheet 2 of 2)

Name	Type	Description
VCC_USIM	Input	USIM Pads VCC connection (1.8 V or 3.0 V)
VSS_USIM	Input	USIM Pad ground connections

NOTES:

1. Exercise care when placing the PXA27x processor into deep-sleep power mode. The voltage control pins are powered from the VCC_IO power domain. In deep-sleep mode, the power is turned off. Powering off the voltage select pins (UVS0,nUVS1,nUVS2) causes shorting of the power supplies that are switched to provide power to the card. To prevent these electrical problems:
   a. Turn off the external power supply before (or at the same time as) the processor enters deep-sleep power mode using USIM[USCCR] bit field.
   b. Configure GPIOs correctly prior to entering deep-sleep mode using software writes to USIM[USCCR]. Refer to the GPIO chapter in the *Intel® PXA27x Processor Family Developers Manual* for more details.
2. The control and status of this signal is defined using the Power Manager USIM Card Control/Status Register (PUCR) in the Clocks and Power Manager Unit chapter of the *Intel® PXA27x Processor Family Developers Manual*.
3. Voltage level of pads is set off chip. The pads work at either 1.8 V or 3.0 V.

19.2.2 USIM Card Interface Signals

See Table 19-2 for the list of the five USIM card signals which connect between the PXA27x processor USIM interface and the USIM card.

Table 19-2. USIM Card Signals

Card Signal	Function	Signal Direction	PXA27x Processor Signal
I/O	Data input/output (bidirectional) pad with a pull-up resistor Refer to *ISO Standard 7816-3* for pad specifications.	USIM card <-> PXA27x processor	UIO
CARD_RST	USIM card reset	PXA27x processor –> 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.	PXA27x processor –> 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_USIM
GND	Mutual USIM card, PXA27x processor USIM interface, and VCC ground reference voltage	Power supply –> Card	VSS_USIM

19.3 Block Diagram

See Figure 19-1 and Figure 19-2 for illustrations of the alternate connectivity scenarios between the PXA27x processor and external USIM card. Figure 19-1 shows configurations supporting both 1.8 V and 3.0 V USIM cards. Figure 19-2 uses nUEN signal to allow the USIM controller to wake up from sleep and standby power modes in response to a falling or rising edge of UDET. This configuration only supports 1.8 V or 3.0 V USIM cards without any additional logic.

Refer to Section 3.8.1.15, "Power Manager USIM Card Control/Status Register (PUCR)," in the Clocks and Power Manager Unit section of the *Intel® PXA27x Processor Family Developers Manual* for information on control/status programmability.

Figure 19-1. Connectivity USIM Card and PXA27x Processor USIM Interface using UVSx signals

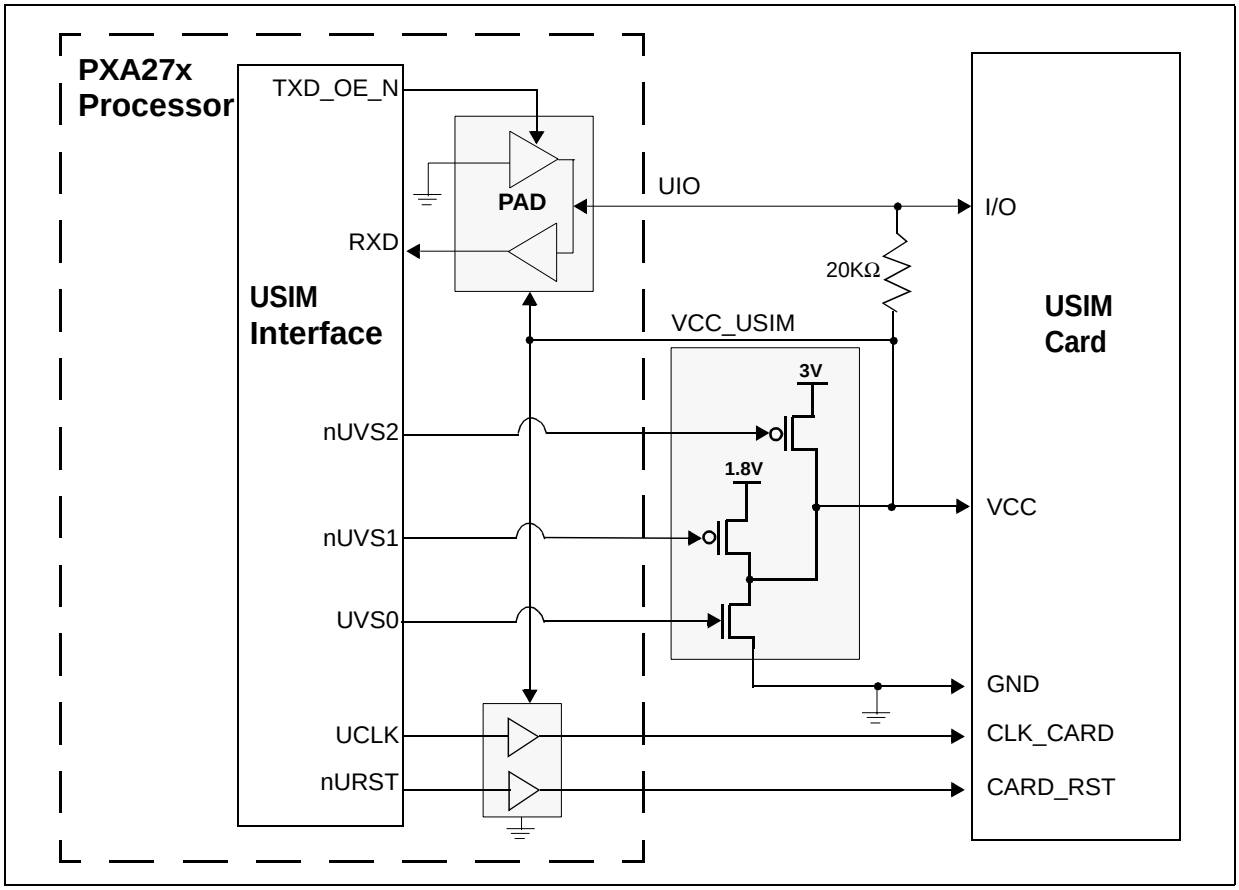
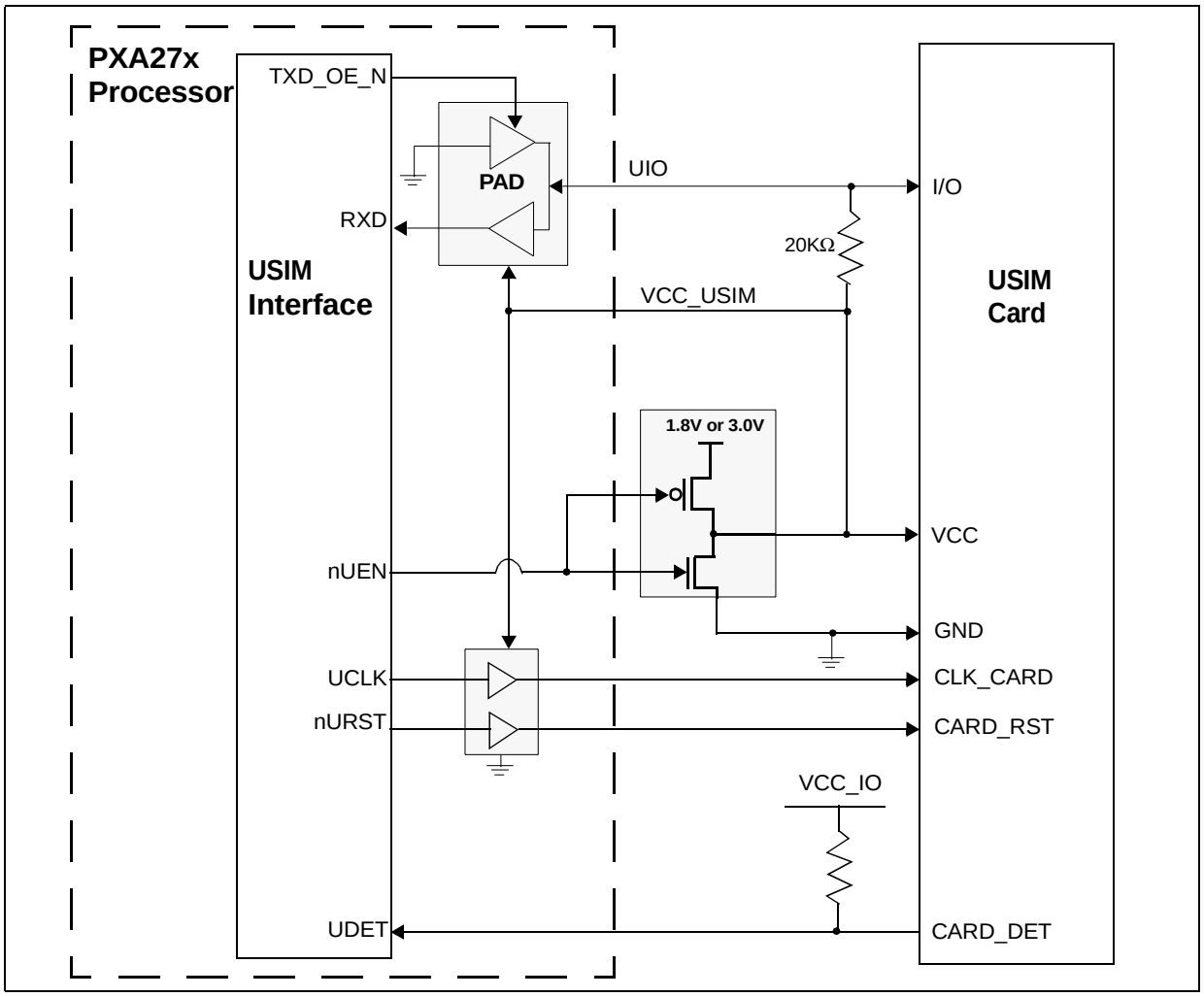


Figure 19-2. Connectivity USIM Card and PXA27x Processor USIM Interface Using nUEN



19.4 Layout Notes

Voltage level of the PXA27x processor pads is set off-chip (externally to PXA27x processor). The pads work at either 1.8 V or 3.0 V. When the USIM card and the PXA27x processor operate at different voltage levels, the I/O voltage level of the USIM card must be at a voltage level the PXA27x processor supports.

Warning: Exercise care when placing the PXA27x processor into deep sleep mode. The voltage control signals are powered from the VCC_IO power domain. In deep sleep mode, this power is turned off. Powering off the voltage select signals (UVS0, nUVS1, and nUVS2) causes shorting of the power supplies which are switched to provide power to the card. To prevent these electrical problems, turn off the external power supply before (or at the same time) the PXA27x processor enters deep sleep mode and correctly configures GPIOs prior to entering deep sleep mode.

§§

Universal Serial Bus Host Interface 20

This chapter describes guidelines for:

- Interfacing the Universal Serial Bus (USB) host controller of the Intel® PXA27x Processor Family (PXA27x processor) to a USB host cable connector
- Attaching USB client devices to the USB host controller of the PXA27x processor

20.1 Overview

The Universal Serial Bus (USB) is a bus cable that supports serial-data exchange between a host computer and a variety of simultaneously accessible peripherals. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol. Peripherals are attached, configured, used, and detached, while the host and other peripherals 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 20-1 for description of the four USB Host controller signals.

Table 20-1. USB Host Controller Interface Signals Summary

Signal Name	Direction	Description
USBHPWR<3:1>	Input	Signal indicates an over-current fault condition on the USB power supply
USBH_P<3:1>	Bidirectional	Data positive differential signal (USB D+)
USBH_N<3:1>	Bidirectional	Data negative differential signal (USB D-)
USBHPEN<3:1>	Output	Signal controls an external power switching device that supplies power to USB peripherals

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.*

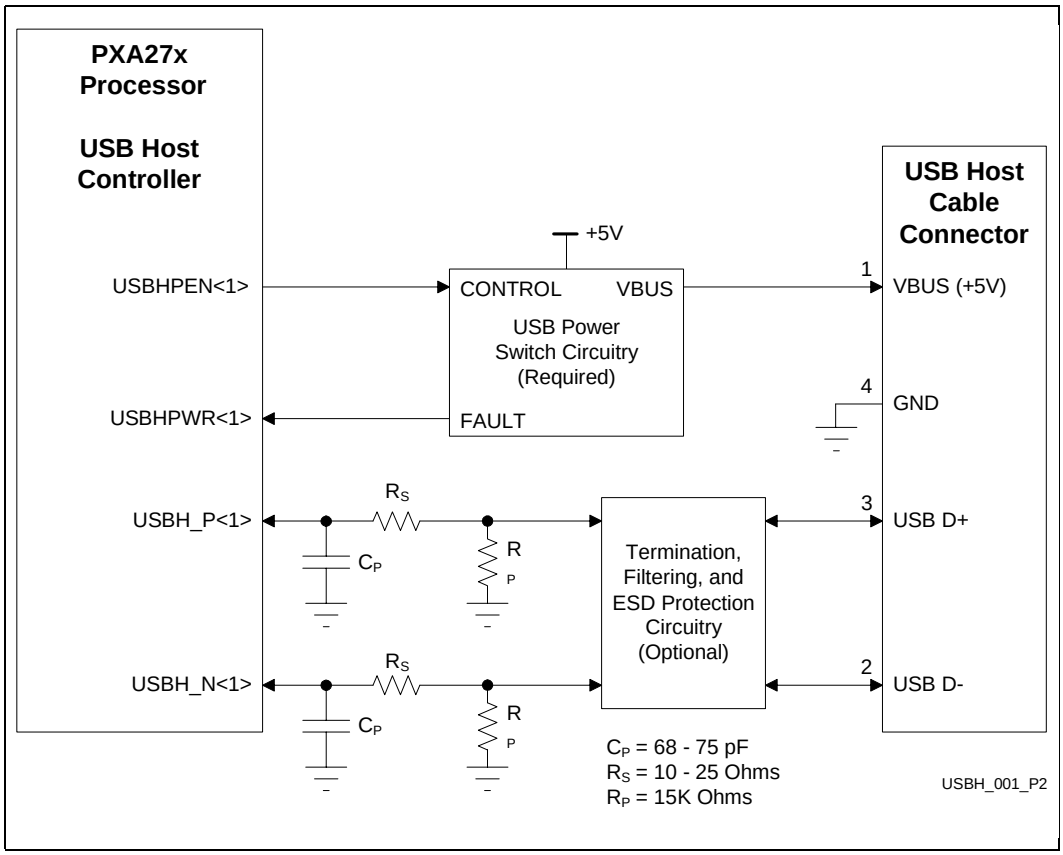
20.3 Block Diagrams

20.3.1 Block Diagram for USB Host Differential Connection (Port 1 or Port 2)

The Host Controller Port 1 is only used with a differential connection. The Host Controller Port 2 also is used with a differential connection.

See the block diagram in Figure 20-1 for USB Host Port 2 differential connections, except connections to USB Host Port 2 of the PXA27x processor.

Figure 20-1. USB Host (Port 1 or Port 2) Differential Connections Block Diagram



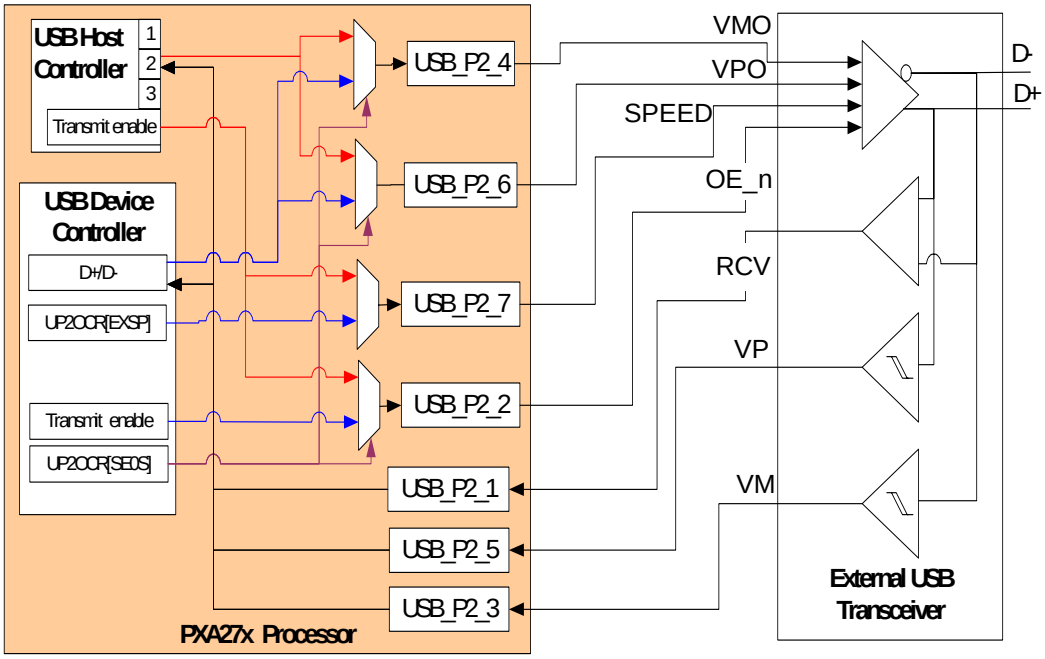
20.3.2 Block Diagrams for USB Host Port 2 (Differential or Single-Ended)

The Host Controller Port 2 is used with a single-ended connection or a differential connection. Host Controller 2 must be used either in a single-ended connection or a differential connection, but not in both connections.

See the block diagram in Figure 20-2 for USB Host Port 2 single-ended connection.

Refer to Chapter 12, "'USB Client Controller," for more information on single-ended or differential connections.

Figure 20-2. PXA27x Processor Host 2 Single-Ended Connection to External Transceiver

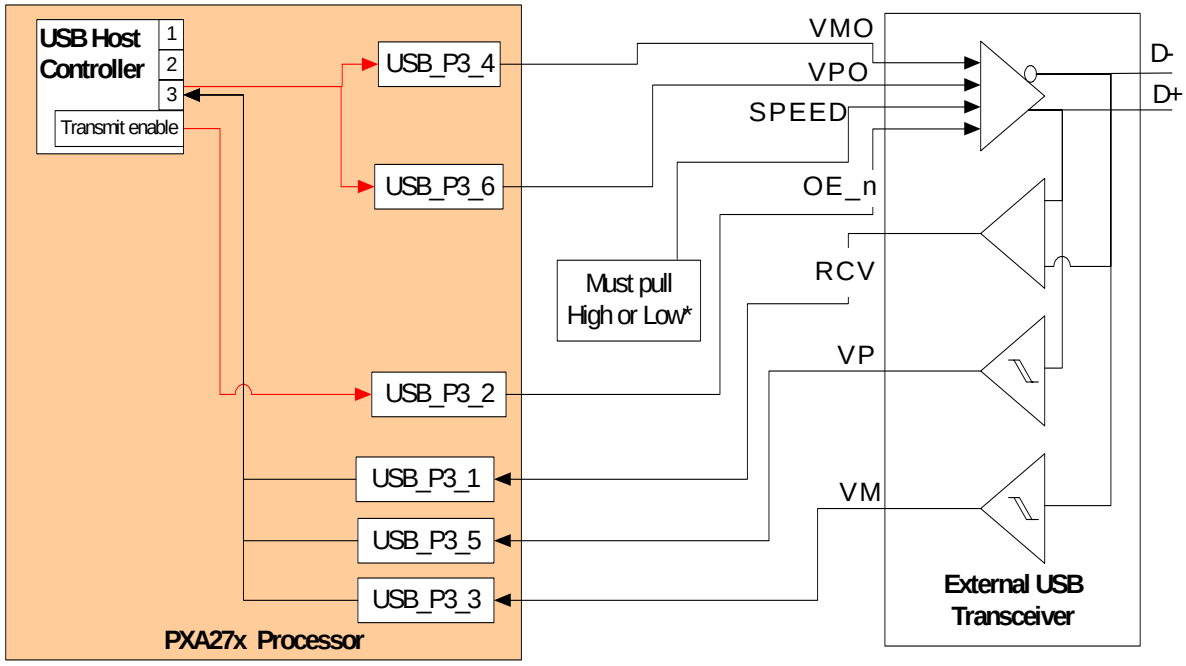


20.3.3 Block Diagram for USB Host Single-Ended Connection (Port 3)

The Host Controller Port 3 is only used with a single-ended connection. See the block diagram in Figure 20-3 for USB Host single-ended connection.

Refer to Chapter 12, "'USB Client Controller," for more information on single-ended connection.

Figure 20-3. PXA27x Processor Host 3 Connection to External USB Transceiver



Note: The "SPEED" signal does not exist for PXA27x Host Port 3. The "SPEED" signal of the external device must be tied high or low. The system designer must program PXA27x 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 PXA27x processor does not have 5.0 V tolerant inputs. System designers must provide an external device to interface the USBHPENx and USBHPWRx pins to the power supply and over current detection circuits.

The C_P and R_S components are required by the USB Host controller for USBH_P<3:1> and USBH_N<3:1> signal compliance with the USB specification and must be placed as close as possible to the PXA27x processor USBH_P<3:1> and USBH_N<3:1> ball pads.

The R_P pull-down resistors shown on the USBH_P<3:1> and USBH_N<3:1> 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.

USBH_P<3:1> and USBH_N<3:1> are differential pair signals. Use these recommended layout guidelines:

- 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)).

§§

Real Time Clock Interface 21

21.1 Overview

The real time clock (RTC) interface of Intel® PXA27x Processor Family (PXA27x processor) 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 details in the *Intel® PXA27x Processor Family Developers Manual*.

The RTC signal is implemented through the GPIOs. Therefore, the hardware considerations necessary for the signal are the same as that of the GPIOs. Refer to Part II: Chapter 24, "General Purpose Input/Output Interfaces," of this document for information regarding the proper hardware implementation of the real time clock signal.

21.2 Signals

The RTC signal is implemented through the GPIOs of the PXA27x processor. Refer to GPIO alternate function table in the GPIO chapter of the *Intel® PXA27x Processor Family Developers Manual* for the GPIO assignments of the RTC HZ_CLK signal.

See Table 21-1 for the description of the signal controlled by the RTC controller of the PXA27x processor.

Table 21-1. RTC Interface Signal List

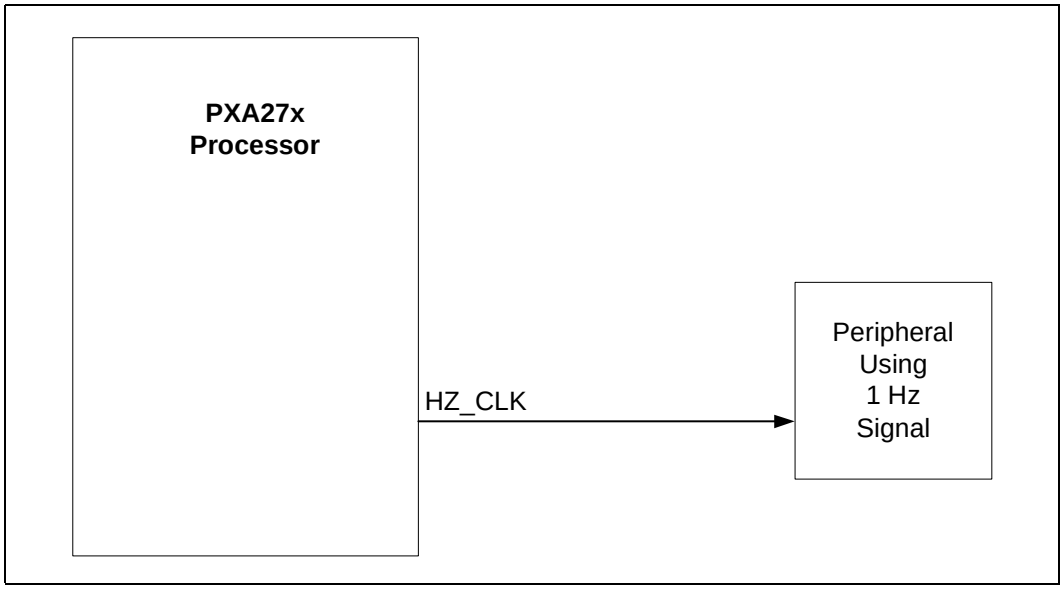
Signal Name	Type	Description
HZ_CLK	Output	1-Hz clock generated by the RTC trimmer section

21.3 Block Diagram

Refer to Part II: Chapter 24, “General Purpose Input/Output Interface,” of this document for information regarding the proper hardware implementation of the real time clock signal.

See Figure 21-1 for illustration of a typical application of the HZ_CLK. The connection is directly between the PXA27x processor and peripheral using the HZ_CLK signal.

Figure 21-1. Example HZ_CLK Block Diagram



21.4 Layout Notes

Refer to Part II: Chapter 24, “General Purpose Input/Output Interface,” of this document for information regarding the proper layout practices for the real time clock signal.

The switching frequency of HZ_CLK is 1.0 Hz. Therefore, layout and routing considerations are far less stringent as for other GPIOs and are somewhat relaxed. However, for best results, adhere to all GPIO layout recommendations.

§§

OS Timer Interface 22

This chapter describes procedures for interfacing the OS Timer controller to Intel® PXA27x Processor Family (PXA27x processor).

22.1 Overview

The operating system timers block provides a set of timer channels that allow software to generate timed interrupts (or wakeup events). In the PXA27x processor, these interrupts are generated by two sets of timer channels:

- One set provides one counter and four match registers and is clocked from a 3.25-MHz clock. This block maintains the four Intel® PXA25x processor compatible timer channels.
- The other set (additional to the Intel® PXA25x processor) provides eight counters and eight match registers and clocked from any of the following:
  - 32.768 KHz timer lock
  - 13 MHz clock
  - An externally supplied clock that provides a wide range of timer resolutions

All references to registers are documented in the *Intel® PXA27x Processor Family Developers Manual* unless otherwise noted.

22.2 Signals

See Table 22-1 for the list of signals used to interface to the OS Timer.

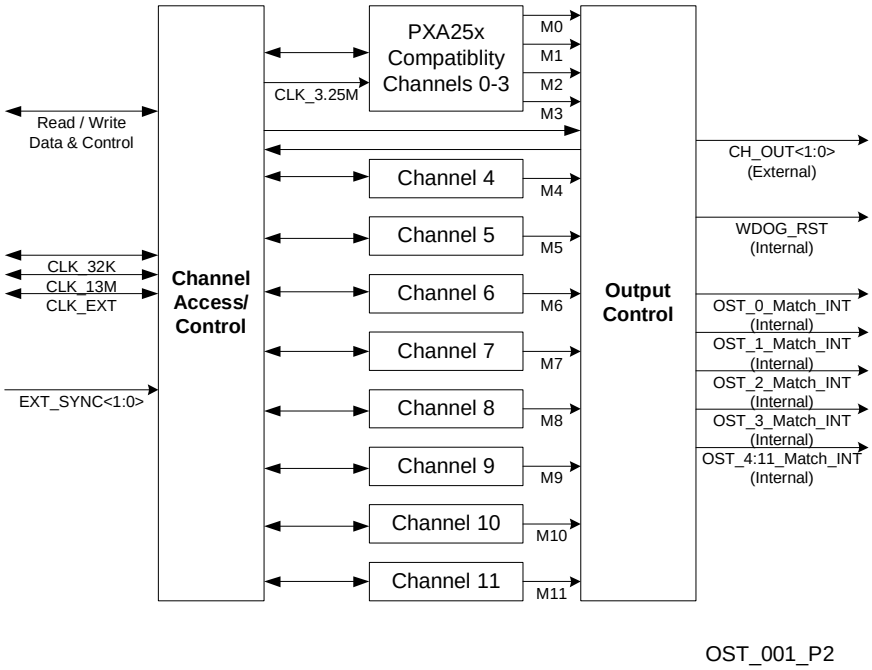
Table 22-1. OS Timer Interface Signals

Name	Type	Description
EXT_SYNC<0>	Input	External Sync 0 This input provides a reset for the OS Timer channels enabled for use
EXT_SYNC<1>	Input	External Sync 1 This input provides a reset for the OS Timer channels enabled for use
CHOUT<0>	Output	Channel Out 0 Periodic clock output from OS Timer channel 11
CHOUT<1>	Output	Channel Out 1 Periodic clock output from OS Timer channel 10

22.3 Block Diagram

See the block diagram of the OS Timer Controller in Figure 22-1.

Figure 22-1. OS Timer Block Diagram



OST_001_P2

22.3.1 Channel Access/Control Block

This block controls reads and writes to registers within the operating system timers block. 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.

22.3.2 PXA25x Compatibility Channels 0-3 Block

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

22.3.3 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 for network sychronization

22.3.4 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

22.4 Layout Notes

The EXT_SYNCx signals are clocked using a two-stage sychnronizer. Therefore, depending on the setting of the OMCRx[CRES] bitfield, the signal must remained asserted for three clock periods of the source clock.

Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specifications* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for all AC timing information.

§§

Pulse-Width Modulator Interface 23

23.1 Overview

The pulse-width modulator (PWM) interface of Intel® PXA27x Processor Family (PXA27x processor) contains four signals that are configured to generate periodic output signals. The configuration of the PWMs is accomplished through software and is described in detail in the *Intel® PXA27x Processor Developers Manual*.

The PWM signals are implemented through GPIOs. Therefore, the hardware considerations necessary for the PWM signals are the same as that of the GPIOs. Refer to Chapter 24, "General Purpose Input/Output Interfaces" of this document for information regarding the proper hardware implementation of the pulse-width modulator signals.

23.2 Signals

The PWM signals are implemented through the GPIOs of the PXA27x processor. Refer to GPIO alternate function table in the GPIO chapter of the *Intel® PXA27x Processor Developers Manual* for the GPIO assignments of the PWM signals.

See Table 23-1 for the list of signals controlled by the PWM controller of the PXA27x processor.

Table 23-1. 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

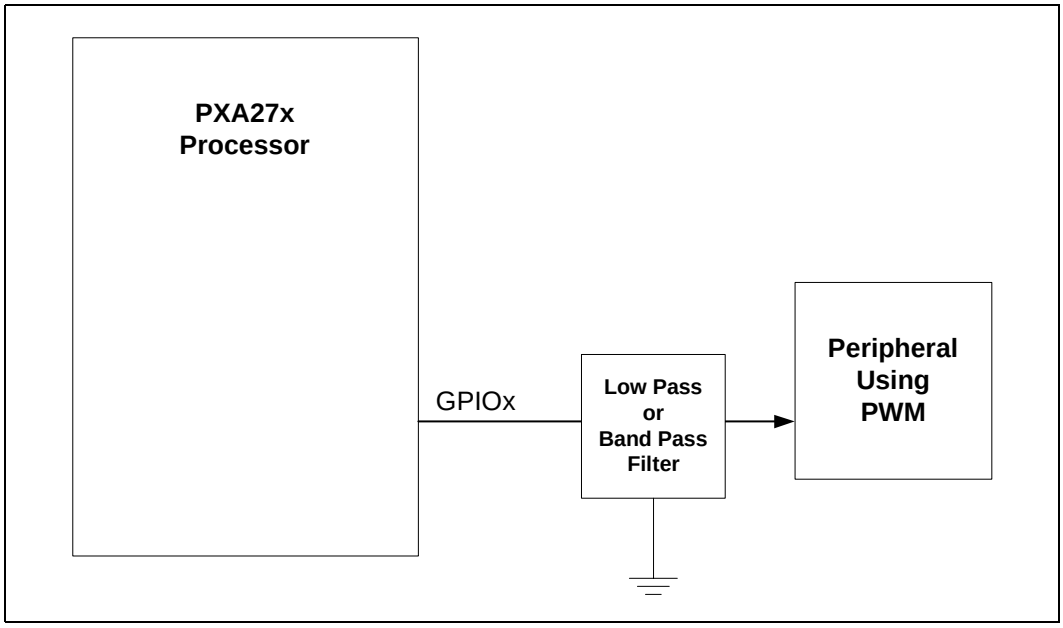
23.3 Block Diagram

Refer to Chapter 24, “General Purpose Input/Output Interfaces” of this document for information regarding the proper hardware implementation of the pulse-width modulator (PWM) signals.

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 this is 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 filter in Figure 23-1.

Figure 23-1. PWM Block Diagram For Applications Requiring a Filter



23.4 Layout Notes

Refer to Chapter 24, “General Purpose Input/Output Interfaces” of this document for information regarding the proper layout practices for the PWM signals.

The maximum switching frequency of the PWM signals is approximately 1.6 MHz. Therefore, layout and routing considerations are not as stringent as for other GPIOs and are somewhat relaxed. However, for best results, adhere to all GPIO layout recommendations.

A typical use for pulse-width modulated outputs is the contrast control for LCD panels. In this situation, the panel backlight inverter is a serious source of noise. To ensure the noise is not induced into the contrast control or the processor, follow all recommendations for shielding the inverter and separating the inverter as far as possible from the contrast control line (PWM output.)

§§

General Purpose Input/Output Interfaces 24

24.1 Overview

The general purpose input/outputs (GPIOs) have a variety of uses:

- Operates as programmable inputs or outputs
- Acts as sources of interrupts to the processor
- Causes a wakeup event from sleep/deep sleep (only some GPIOs have this functionality)

Many of the GPIOs have alternate functions assigned to them that are configured with use of the peripheral controllers of Intel® PXA27x Processor Family (PXA27x processor).

Refer to Section 24 of the *Intel® PXA27x Processor Developers Manual* for possible alternate function assignments.

24.2 Signals

There are 119 GPIOs in discrete packages and 121 GPIOs in Intel® PXA27x Processor Family as described in GPIO table in the GPIO chapter of the *Intel® PXA27x Processor Developers Manual*. When selected to be used as GPIOs, all GPIOs are configured and function identically. If selected to be used with an alternate function, the limitations and functionality of the GPIOs differ and are dependant upon the limitations and constraints of the chosen alternate function. This chapter only describes the design considerations of the GPIOs. Regarding considerations for possible use as an alternate function, refer to the appropriate chapter.

Table 24-1. GPIO Interface Signal List

Signal Name	Type	Description
GPIO<120:0>	Input or Output	GPIOs (119 in discrete packages and 121 in Intel® PXA27x Processor Family with Intel® Folded-SCSP) are programmed as inputs or outputs. The signals are configurable using registers that select whether the signal functions as a GPIO or as one of several alternate functions.

Power considerations hold greater importance to the GPIOs as there are many GPIOs and not all are necessarily required in every design. GPIOs that are incorrectly configured causes increased power draw, possibly beyond system specifications. All GPIOs, whether required in the design or not, must be properly configured. For minimal power consumption, configure all unused GPIOs as outputs.

Do not use the GPIOs to drive large loads. The current drive capabilities are sufficient to drive a small LED without buffers, but as they are designed to drive signals, avoid using GPIOs to drive significant loads. Refer to *Intel® PXA270 Processor Electrical, Mechanical, and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for the highest allowable current available from the GPIOs.

There are no pull-up or pull-down resistors connected to any of the GPIOs when PSSR[RDH] is cleared, which is the state required for normal operation. These pull-ups and pull-downs only operate in sleep/deep sleep mode. Any pull-ups or pull-downs required for normal operation must be externally provided. The possibility of conflicting pull-ups and pull-downs must be considered when using external pull-up or pull-down resistors in sleep/deep sleep mode. Failure to properly account for this results in incorrect operation or excess power usage.

Note: nRESET_GPIO (GPIO<1>) has an internal pull-up that is active following any reset the exception is when exit from sleep or deep sleep and until PSSR[RDH] is cleared.

A portion of the GPIO pins have alternate functions with bidirectional signals. For these signals, the direction of the pin is controlled by the peripheral directly overriding the GPIO direction settings for these pins. These pins are:

- MMCMD,
- MMDAT<3:0>
- MSSDIO
- SSPSCLK1
- SSPSCLK2
- SSPSCLK3
- SSPSFRM
- SSPSFRM2
- SSPSFRM3
- L_DD<17:0>
- I^2C pins:
  - PWR_SDA
  - PWR_SCL
  - SDA
  - SCL

For all other signals, the GPIO Direction Register must be correctly configured for the GPIO function.

Special care must be taken with any GPIO that is used for generating resets. The power manager directly overrides the function of GPIO<10:2> and configures these signals as GPIOs for use as GPIO reset signals. If the particular system implementation uses this function, the GPIOs must be designed such that this does not cause conflicts on the GPIOs. Refer to Section 3 of the *Intel® PXA27x Processor Developers Manual* for information regarding this function.

24.3 Block Diagram/Schematic

For many uses, especially slow switching or static applications, the GPIOs are directly connected to the corresponding signal of the peripheral. Buffers and terminating resistors are not required.

24.4 Layout Notes

The GPIOs are not fast-switching devices and do not switch any faster than 10 MHz. This allows greater flexibility in the design and layout of the GPIO signals on the PCB. Guidelines and techniques, similar to those used with the GPIOs of the Intel® PXA250 and PXA210 processors, are applicable for the PXA27x processor.

The 10 MHz maximum switching frequency only applies to the GPIO pins when configured as GPIOs. When configured with an alternate function, the maximum switching speed exceeds 10 MHz, depending upon the alternate function in use. Refer to the appropriate section in the *Intel® PXA27x Processor Developers Manual* for more information on the maximum switching speed for GPIOs when configured with an alternate function.

Be certain not to run any GPIO signals, especially reset and interrupt related signals, in close proximity to clock or address/data lines. The higher frequency of these signals results in spurious transitions on the GPIO signals that causes extraneous resets or interrupts. Separate these signals both spatially on the layer as well as between layers to avoid this problem.

§§

Interrupt Interface 25

This chapter describes the procedures for interfacing with the interrupt controller of Intel® PXA27x Processor Family (PXA27x processor).

25.1 Overview

The interrupt controller interfaces to both internal and external peripheral interrupt request. The means of interfacing an external peripheral interrupt request is through the GPIO signals. All the GPIO signals are configured to generate an interrupt on a rising edge, falling edge or both edges. Refer to the Interrupt and GPIO sections in the *Intel® PXA27x Processor Family Developers Manual* for enabling interrupts through GPIO signals and for setting the edge of the appropriate GPIO interrupt.

To configure the GPIO<120:0> signals as interrupt signals, perform these steps:

1. Program the GPIO pin direction using the GPDRx register.
2. Program the GPIO edge detect using the GRERx or GFERx registers.
3. Program the GPIO alternate function using the GAFRx_x register.
4. Determine if the GPIO interrupt generates an IRQ or FIQ interrupt using the ICLR register.
5. Configure the priority of the GPIO interrupt using the IPR8 – IPR10 registers.
6. Unmask the GPIO interrupts using the ICMR<10:8> register.

Note: GPIO<120:119> are only available in the Intel® PXA27x Processor Family.

All references to registers are documented in the *Intel® PXA27x Processor Family Developers Manual* unless otherwise noted.

25.2 Signals

See Table 25-1 for description of the signals associated with the GPIO Unit. All GPIO unit signals are programmable as interrupt signals and, therefore, are used by the interrupt controller.

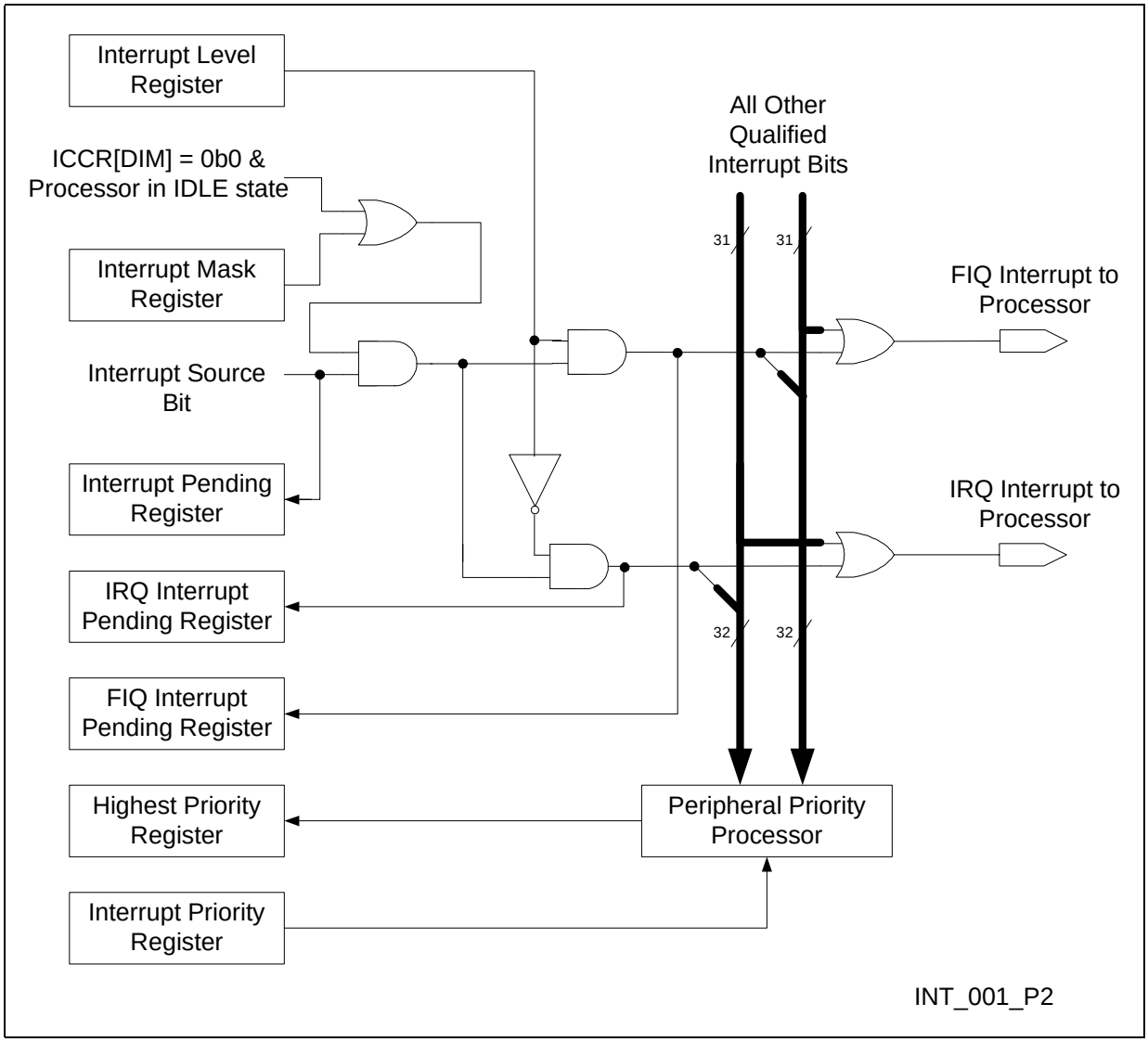
Table 25-1. GPIO Unit I/O Signal

Signal Name	Type	Description
GPIO<0>	Input/Output	Causes an independent first-level interrupt Configured to generate an interrupt by setting ICPR8 bit field. This signal contains an internal resistive pull-down that are enabled during power-on, hardware, watchdog, and GPIO resets and are disabled when PSSR[RDH] is clear.
GPIO<1>	Input/Output	Causes an independent first-level interrupt Configured to generate an interrupt by setting ICPR9 bit field. This signal contains an internal resistive pull-up that are enabled during power-on, hardware, watchdog, and GPIO resets and are disabled when PSSR[RDH] is clear.
GPIO<120:2>	Input/Output	Causes an second-level interrupt These signals cause an interrupt if an edge is detected and ICPR10 bit field is set.

25.3 Block Diagram

See Figure 25-1 for 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 31 times for the additional 31 possible interrupts represented. Figure 25-1 shows all other qualified interrupt bits.

Figure 25-1. Interrupt Controller Block Diagram



25.4 Layout Notes

All GPIO input signals are received through a two-stage synchronizer to eliminate meta-stable problems that result from clocking an asynchronous signal into a synchronous digital input. Interrupt signals received using GPIO pins are acknowledged when the function is properly configured and GPIO pin are asserted for greater than 154 ns (2 x 1/13 MHz) during run and idle mode. A pulse less that 77 ns (1/13 MHz) on a GPIO pin cannot be detected, whereas a pulse with a period between 77 ns and 154 ns is undetermined and must be avoided.

During standby mode, GPIO<15:0> signal and those signals associated with the keypad interface are programmable to generate an interrupt after wake-up. To guarantee that the interrupt is acknowledged after assertion of a GPIO signal wake-up event, the GPIO pin must be configured for receiving an interrupt.

In addition, the minimum time the GPIO signal must be asserted is determined by the PCFR[OPDE] bit:

- If the PCFR[OPDE] bit is not set, the GPIO signal must be asserted for a minimum of 1 ms to transition from the wake-up event to locking the PLLs.
- If the PCFR[OPDE] bit is set, the GPIO signal must be asserted for a minimum of 7 ms to transition from wake-up event to locking the PLLs.

Considerations must be made to avoid crosstalk caused by running signals too close to asynchronous signals, such as interrupt signals. Induced noise as a result of an adjacent signal causes a spurious interrupt if amplitude of the adjacent signal is great enough and it meets the setup and hold timing requirements on both edges.

§§

JTAG Debug 26

This chapter describes the boundary-scan (JTAG) features of Intel® PXA27x Processor Family (PXA27x processor). The boundary-scan interface provides a means of driving and sampling the external pins of the processor, regardless of the state of the core. This function tests the processor's 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.

26.1 Overview

The boundary-scan interface intercepts each external connection in the processor using a boundary-scan cell. The boundary-scan cells combine to form a serial shift register, the boundary-scan register.

The interface is controlled through five dedicated test access port (TAP) pins as described in Table 26-1:

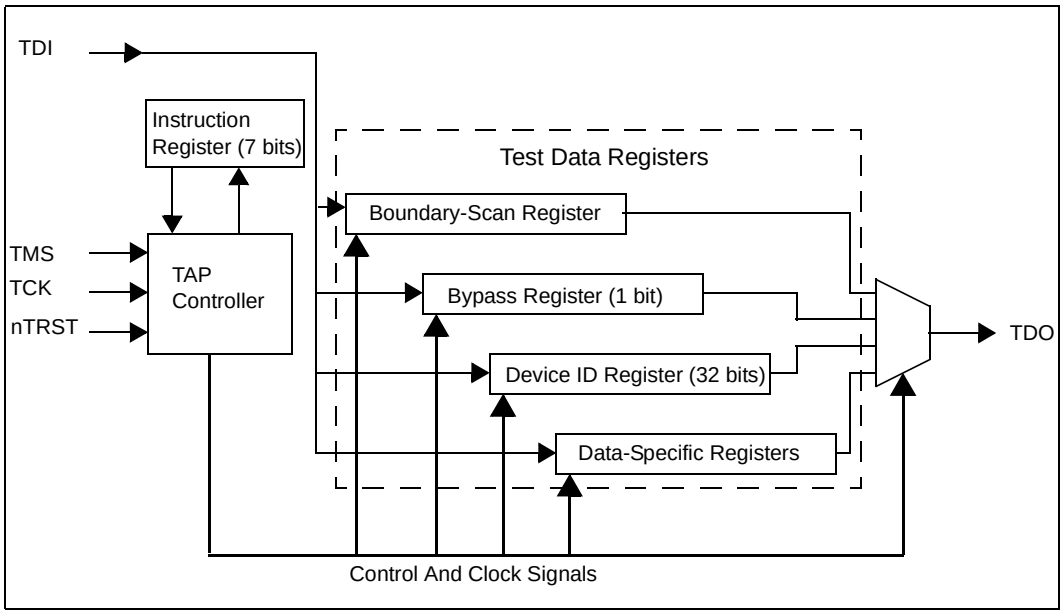
- TDI
- TMS
- TCK
- nTRST
- TDO

The boundary-scan test-logic elements include:

- TAP pins
- TAP controller
- Instruction register
- Boundary-scan register
- Bypass register
- Device identification register
- Data-specific register(s)

See Figure 26-1 for illustration of all the above elements.

Figure 26-1. Test Access Port (TAP) Block Diagram



26.2 Features

The boundary-scan interface complies with *IEEE Standards 1149.1-1990*, *IEEE Standards 1149.1a-1993*, and *IEEE Standard Test Access Port and Boundary-Scan Architecture*, with support for:

- Board-level boundary-scan connectivity testing
- Connection to software debugging tools through the JTAG interface
- In-system programming of programmable memory and logic devices on the PCB

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

26.3 Signal Descriptions

The TAP interface is controlled through five dedicated pins: TDI, TMS, TCK, nTRST, and TDO. See Table 26-1 for description of these pins.

Table 26-1. TAP Controller Pin Definitions

Signal Name	Direction	Description
TCK	Input	Test Clock — clock input for the TAP controller and instruction and test data registers
TMS	Input	Test Mode Select — controls operation of the TAP controller The TMS input is pulled high when it is not being driven. TMS is sampled on the rising edge of TCK.
TDI	Input	Test Data In — 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 when it is not being driven.
TDO	Output	Test Data Out — serial data output Data at TDO is clocked out on the falling edge of TCK. It provides an inactive (high-impedance) state during non-shift operations to support parallel connection of TDO outputs at the board or module level.
nTRST	Input	Test Reset — provides asynchronous initialization of the JTAG test logic Asserting this pin puts the TAP controller in the *Test-Logic-Reset* state. An external source must drive nTRST before or at the same time as the hardware nRESET pin for correct TAP controller and device operation.

26.4 Operation

This section describes the operation of the JTAG interface and TAP controller implemented in the PXA27x processor.

26.4.1 TAP Controller Reset

The boundary-scan interface includes a synchronous finite state machine, the TAP controller (see Figure 26-3). In order to force the TAP controller into the correct state, a reset pulse must be applied to the nTRST pin. This forces the TAP controller into the *Test-Logic-Reset* state (TLRS).

A clock on TCK is not necessary to reset the TAP controller.

To use the boundary-scan interface, these requirements must be met:

- 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.
- For JTAG TAP operation, the nBATT_FAULT and nVCC_FAULT pins must always be driven high (de-asserted). An active low signal on either pin puts the device into sleep mode, which powers down all JTAG circuitry.

The tasks of a pulse or DC level nTRST reset are:

- Selection of the system mode (the boundary-scan chain does not intercept the signals that pass between the pads and the core).
- Selection of the idcode instruction. If TCK is pulsed, the contents of the ID register are clocked out of TDO.

If JTAG is not used, at a minimum, the nTRST signal must be connected to the nRESET signal to cause a reset on nTRST at power-up. TCK must be grounded.

Refer to the ARM* *Multi-ICE System Design Considerations, Application Note 72 (*ARM DAI 0072A) for additional information on the JTAG interface.

26.4.2 Pull-Up Resistors

The *IEEE 1149.1* standard effectively requires that TDI, TMS, and nTRST have internal pull-up resistors. Leave the TDI and TMS pins left unconnected when not in use. TCK must be tied low in cases where JTAG is not used.

Scan Chain

PXA27x has 4 scan chains, controlled from a single JTAG style TAP controller. These are referred to as scan chains 0,1, 2, and 3 and are arranged as shown in Figure 26-2, “PXA27x Scan Chain Arrangement”. The scan chains are selected by a TAP controller instruction.

Scan Chain

0 This allows access to the PXA27x core. The scan chain’s functions allow inter-macrocell testing (EXTEST), and allow the core’s test patterns to be applied serially (INTEST). The order of the scan chain (from **TDI** to **TDO**) sequentially:

1. Data bus bits 0 through 3
2. Control signals (order to be determined)
3. Address bus bits 31 through 0

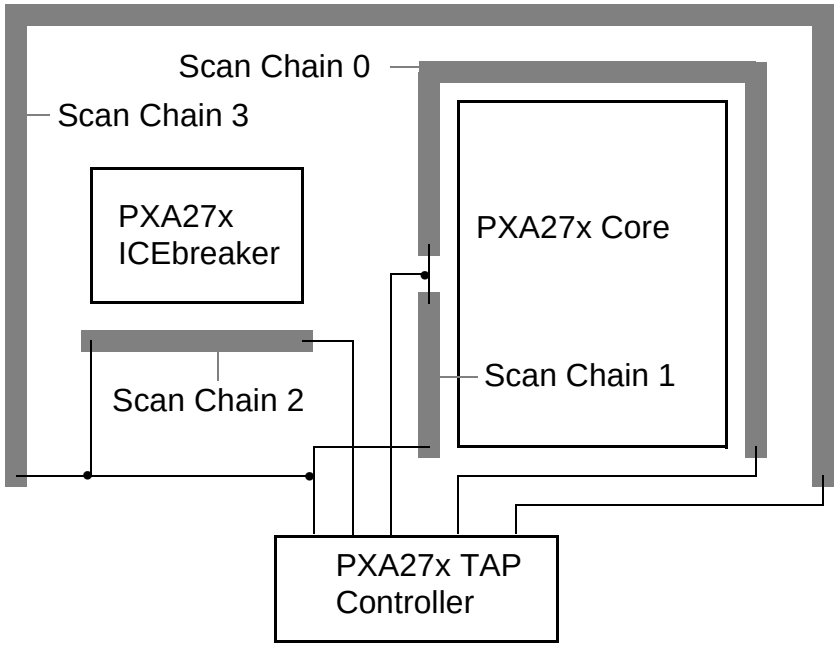
1 This is a small scan chain which only allows access to the core’s data bus. There are 32 scan cells in this chain. This scan chain is used during debug to insert instructions into the processor’s pipeline and capture the internal state as it is written. The order of the scan chain is (from **TDI** to **TDO**): data bus bits 0 through 31.

2 This is a scan chain around the PXA27x ICEbreaker macrocell. This allows the watchpoint registers to be programmed and tested.

3 This is a scan chain around the whole of the PXA27x. The scan chain allows the PXA27x core to be exercised (INTEST) and allows inter-device testing at a board level (EXTEST). The order of the scan chain is to be determined.

Note: Scan Chains 0 and 1 are not fully JTAG compliant in that data cannot be moved around the chains without affecting the scan cell outputs. Use these scan chains only in debug state when the core is not being clocked.conform

Figure 26-2. PXA27x Scan Chain Arrangement



26.4.3 JTAG Instruction Register and Instruction Set

The seven-bit instruction register (IR) holds instruction codes shifted in through the TDI pin. Instruction codes in this register are used to select the specific test operation to be performed and the test-data register to be accessed. These instructions are either mandatory, optional, user-defined, or private, as set forth in the *IEEE 1149.1* standard.

The most-significant bit of the IR is connected to TDI and the least-significant bit is connected to TDO. TDI is shifted into the IR on each rising edge of TCK as long as TMS remains asserted.

When nTRST is asserted, idcode becomes the default instruction.

The PXA27x processor supports the mandatory public boundary-scan instructions, optional public instructions, user-defined instructions, and private instructions listed in Table 26-2. The processor does *not* support the *IEEE 1149.1* optional public instructions runbist, intest, and usercode. See Table 26-3 for description of the supported instructions in detail.

Table 26-2. IEEE 1149.1 Boundary-Scan Instruction Set

Instruction Code	Instruction Type	Instruction Name	Instruction Code	Instruction Type	Instruction Name
0b000_0000	mandatory public	extest	0b000_1010 - 0b000_1111	private	private
0b000_0001	mandatory public	sample/preload	0b001_0000	user defined	dbgtx
0b000_0010	user defined	dbgrx	0b001_0001- 0b011_0101	private	private
0b000_0011	private	private	0b011_0110	user defined	flashload
0b000_0100	optional public	clamp	0b011_0111	user defined	flashprogram
0b000_0101 - 0b000_0110	private	private	0b011_1000 - 0b111_1101	private	private
0b000_0111	user defined	ldic	0b111_1110	optional public	idcode
0b000_1000	optional public	highz	0b111_1111	mandatory public	bypass
0b000_1001	user defined	dcsr			

Table 26-3. IEEE 1149.1 Boundary-Scan Instruction Descriptions (Sheet 1 of 2)

Instruction	Opcode	Description
extest IEEE 1149.1 required	0b000_0000	The extest instruction initiates testing of external circuitry, typically board-level interconnections and off-chip circuitry. The extest instruction connects the boundary-scan register between TDI and TDO in the Shift-DR state only. When extest is selected, output signal pin values are driven by values shifted into the boundary-scan register and change only on the falling-edge of TCK in the Update-DR state. When extest is selected, all system input-pin states are loaded into the boundary-scan register on the rising edge of TCK in the Capture-DR state. Values shifted into input latches in the boundary-scan register are never used by the processor's internal logic.
sample/ preload IEEE 1149.1 required	0b000_0001	The sample/preload instruction performs two functions: • When the TAP controller is in the Capture-DR state, the sample instruction executes on the rising edge of TCK and provides a snapshot of the component's normal operation without interfering with that operation. The instruction causes boundary-scan register cells to sample data entering and leaving the processor. • When the TAP controller is in the Update-DR state, the preload instruction occurs on the falling edge of TCK. This instruction causes the data held in the boundary-scan cells to be transferred to the slave register cells. Typically, the slave-latched data is then applied to the system outputs by means of the extest instruction.
dbgrx	0b000_0010	Refer to Chapter 26, "Software Debug," in the *Intel® PXA27x Processor Family Developers Manual*.
clamp	0b000_0100	The clamp instruction allows the states of the signals driven from the PXA27x processor pins to be determined from the boundary-scan register while the bypass register is selected as the serial path between TDI and TDO. Signals driven from the component pins do not change while the clamp instruction is selected.
ldic	0b000_0111	Refer to Section 26.4.6.3, "Downloading Code into the Instruction Cache," in the *Intel® PXA27x Processor Family Developers Manual*.
highz	0b000_1000	The highz instruction floats all three-statable output and I/O pins. When this instruction is active, the bypass register is connected between TDI and TDO. This register is accessed using the JTAG TAP throughout the device operation. The bypass register is also accessed with the **bypass** instruction.
dcsr	0b000_1001	Refer to Chapter 26, "Software Debug," in the *Intel® PXA27x Processor Family Developers Manual*.
dbgtx	0b001_0000	Refer to Chapter 26, "Software Debug," in the *Intel® PXA27x Processor Family Developers Manual*.

Table 26-3. IEEE 1149.1 Boundary-Scan Instruction Descriptions (Sheet 2 of 2)

Instruction	Opcode	Description
flashload	0b011_0110	The flashload instruction is for loading values that are programmed into an external flash device. This instruction mimics the sample/preload instruction, but only on a subset of boundary-scan register cells, the flash data register, needed to program an external flash device to reduce flash programming time. Flash values intended for programming are loaded during the Shift-DR state.
flashprogram	0b011_0111	The flashprogram instruction for programming the values into an external flash device. This instruction mimics the extest instruction, but uses only on a subset of boundary-scan register cells, specifically the flash data register. This subset reduces flash programming time. When flashprogram is selected in the Update-IR state, relevant output signal pins are driven by values shifted into the flash data register from a previous flashload instruction. The signals not listed in the flash data register are not affected.
idcode IEEE 1149.1 optional	0b111_1110	The idcode instruction is used with the device identification (ID) register. It connects the ID register between TDI and TDO in the Shift-DR state. When selected, idcode parallel-loads the hard-wired identification code (32 bits) on TDO into the ID register on the rising edge of TCK in the Capture-DR state. **NOTE:** The ID register is not altered when data is shifted in on TDI.
bypass IEEE 1149.1 required	0b111_1111	The bypass instruction selects the bypass register between TDI and TDO pins in the Shift-DR state, effectively bypassing the processor's test logic. 0b0 is captured in the Capture-DR state. While this instruction is in effect, no other test data registers have any effect on the operation of the system. Test data registers with both test and system functionality perform their system functions when this instruction is selected.

26.4.4 Test Data Registers

These subsections describe the test data registers:

Part II: Section 26.4.4.1 — Bypass Register
Part II: Section 26.4.4.2 — Boundary-Scan Register
Part II: Section 26.4.4.3 — Data-Specific Registers
Part II: Section 26.4.4.4 — Flash Data Register
Part II: Section 26.4.4.5 — Intel XScale® Data Registers
Part II: Section 26.5.1 — JTAG Device Identification (ID) Register

26.4.4.1 Bypass Register

The single-bit bypass register is selected as the path between TDI and TDO to allow the PXA27x processor to be bypassed during boundary-scan testing. This allows for more rapid movement of test data to and from other board-level components that performs JTAG test operations.

When the bypass, highz, or clamp instruction is the current instruction in the instruction register, serial data is transferred from TDI to TDO in the Shift-DR state with a delay of one TCK cycle.

A logic 0 is loaded from the parallel input of the bypass register in the *Capture-DR* state. There is no parallel output from the bypass register.

26.4.4.2 Boundary-Scan Register

The boundary-scan register consists of a set of serially connected cells around the periphery of the PXA27x processor at the interface between the core logic and the PXA27x processor I/O pins. This register isolates the pins from the core logic and then drive or monitor the pins. The connected boundary-scan cells make up a shift register.

See Table 26-4 for the list of the PXA27x processor I/O pins that are not part of the boundary-scan register.

Table 26-4. I/O Pins Excluded from Boundary-Scan Register

Pin	Reason for Exclusion
PXTAL_IN PXTAL_OUT TXTAL_IN TXTAL_OUT PWR_EN ALL VDD/VSS pins	Prevents disruption of processor clocks and power
PWR_OUT nRESET_OUT	Prevents disruption of power and unintentional reset for external components
TCK TMS TDI TDO nTRST	Prevents disruption of the TAP-controller JTAG interface

The boundary scan logic powers down when nBATT_FAULT, nVCC_FAULT, or nRESET is asserted (low) and when the PXA27x processor device is in sleep or deep sleep. Refer to Chapter 3, "Clocks and Power Manager Unit," in the *Intel® PXA27x Processor Family Developers Manual* for details of sleep and deep-sleep modes. Thus, nBATT_FAULT, nVCC_FAULT, and nRESET must be driven high (de-asserted) for any instruction that uses the boundary-scan register.

The boundary-scan register is selected as the register to be connected between TDI and TDO only during the sample/preload and extest instructions. Values in the boundary-scan register are used but are not changed during the clamp instruction.

During normal (system) operation, straight-through connections between the core logic and pins are maintained, and normal system operation is unaffected. This is also the case when the sample/preload instruction is selected.

In test mode when extest is the currently selected instruction, values are applied to the output pins independently of the actual values on the input pins and core logic outputs. In the PXA27x processor, all of the boundary-scan cells include update registers. Refer to the *IEEE 1149.1* standard for more information on the update registers.

The values stored in the boundary-scan register after power-up are not defined. The values previously clocked into the boundary-scan register are not guaranteed to be maintained across a JTAG reset (from forcing nTRST low or entering the *Test-Logic-Reset* state).

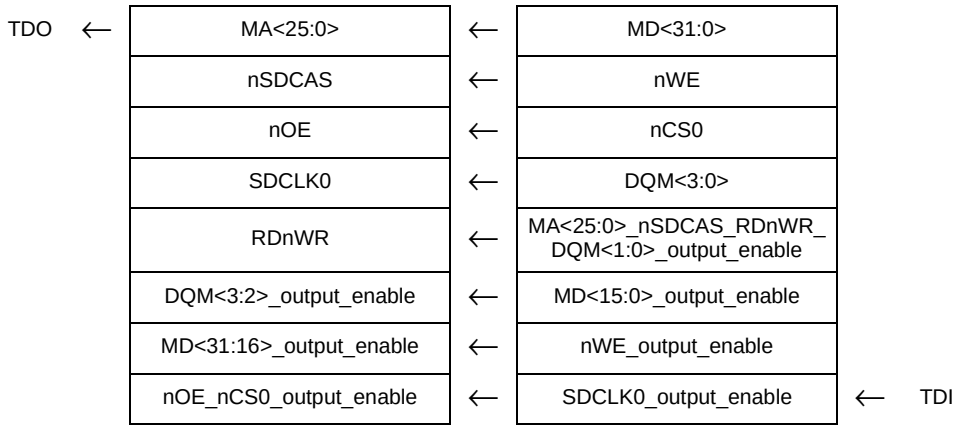
26.4.4.3 Data-Specific Registers

Data-specific registers are used for software debugging and to initialize the processor instruction cache. For more information, refer to these subsections in Chapter 26, “Software Debug,” of the *Intel® PXA27x Processor Family Developers Manual*:

- Section 26.5.2, “Debug Control and Status Register (DCSR)”
- Section 26.4.6.1.1, “SEL_DCSR JTAG Command and Register”
- Section 26.4.6.3.1 “LDIC JTAG Data Register”
- Section 26.4.6.1.2, “DBG_TX JTAG Command and Register”
- Section 26.4.6.1.4, “DBG_RX Data Register”

26.4.4.4 Flash Data Register

The flash data register is a subset of the boundary-scan register. This subset of cells pertinent to flash programming facilitates shorter programming times using JTAG. The output signals and pins required for external flash programming are ordered from TDI to TDO as:



For instructions that utilize the flash data register, nBATT_FAULT, nVCC_FAULT, and nRESET must be de-asserted.

26.4.4.5 Intel XScale® Data Registers

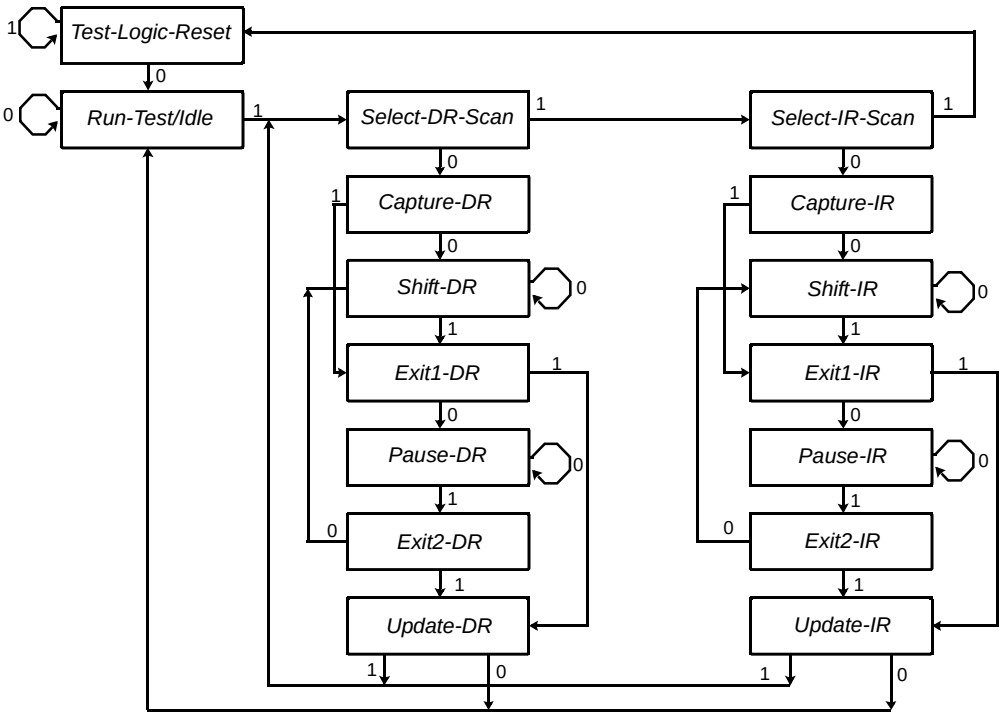
Intel XScale® Technology data registers are not documented here. They are used in conjunction with the user-defined JTAG instructions that are described in the *Intel XScale® Core Developer’s Manual*.

26.4.5 Test Access Port (TAP) Controller

The TAP controller is a 16-state, synchronous, finite state machine that controls the sequence of test logic operations. The TAP is controlled using a bus master that is an automatic test equipment or a programmable logic device that interfaces with the TAP. The TAP controller changes state only in response to power-up or a rising edge of TCK. The value of the TMS input signal at a rising edge of TCK controls the sequence of state changes. The TAP controller is automatically initialized on power up. It is also initialized by applying a high signal level on the TMS input for five TCK periods.

The following subsections describe the behavior of the TAP controller and other test logic in each controller state. See Figure 26-3 for illustration of the state transitions that occur in the TAP controller. For more information on the TAP states and the public instructions, refer to the *IEEE 1149.1* standard.

Figure 26-3. TAP Controller State Diagram



NOTE: All state transitions are based on the value of TMS.

26.4.5.1 Test-Logic-Reset State

In this state, test logic is disabled to allow the PXA27x processor to operate normally. No matter what state the controller is in, the PXA27x processor enters the Test-Logic-Reset state when the TMS input is held high for at least five rising edges of TCK. The controller remains in this state while TMS is high. Asserting nTRST forces the TAP controller to enter the Test-Logic-Reset state.

If the controller exits the Test-Logic Reset controller state as a result of an erroneous low signal on the TMS line on the rising edge of TCK (for example, a glitch due to external interference), it (controller) returns to the Test-Logic-Reset state after three rising edges of TCK with the TMS line high. Test logic operation does not disturb on-chip logic application as the result of such an error.

26.4.5.2 Run-Test/Idle State

The TAP controller enters the Run-Test/Idle state between scan operations. The controller remains in this state as long as TMS is held low. Instructions that do not call functions that execute in the Run-Test/Idle state do not generate any activity in the test logic while the controller is in the Run-Test/Idle state. The instruction register and all test data registers retain their current states. When TMS is high on the rising edge of TCK, the controller moves to the Select-DR-Scan state.

26.4.5.3 Select-DR-Scan State

The Select-DR-Scan state is a temporary controller state. The test data register selected by the current instruction retains its previous state. When the controller is in the Select-DR-Scan state and TMS is held low on the rising edge of TCK, the controller moves into the Capture-DR state, and a scan sequence for the selected test data register is initiated. If TMS is held high on the rising edge of TCK, the controller moves into the Select-IR-Scan state.

The current instruction does not change while the TAP controller is in this state.

26.4.5.4 Capture-DR State

When the controller is in the Capture-DR state and the current instruction is sample/preload, the boundary-scan register captures input-pin data on the rising edge of TCK. Test data registers that do not have parallel input are not changed. If the sample/preload instruction is not selected during this state, the boundary-scan register cells retain their previous states.

The current instruction does not change while the TAP controller is in this state.

If TMS is high on the rising edge of TCK, the controller enters the Exit1-DR state. If TMS is low on the rising edge of TCK, the controller enters the Shift-DR state.

26.4.5.5 Shift-DR State

In the Shift-DR controller state, the test data register, which is connected between TDI and TDO as a result of the current instruction, shifts data one bit position nearer to its serial output on each rising edge of TCK. Test data registers that the current instruction selects but does not place in the serial path retain their previous values during this state.

The current instruction does not change while the TAP controller is in this state.

If TMS is high on the rising edge of TCK, the controller enters the Exit1-DR state. If TMS is low on the rising edge of TCK, the controller remains in the Shift-DR state.

26.4.5.6 Exit1-DR State

Exit1-DR is a temporary controller state. When the TAP controller is in the Exit1-DR state and TMS is held high on the rising edge of TCK, the controller enters the Update-DR state, which terminates the scanning process. If TMS is held low on the rising edge of TCK, the controller enters the Pause-DR state.

The current instruction does not change while the TAP controller is in this state. The test data register selected by the current instruction retains its previous value during this state.

26.4.5.7 Pause-DR State

The Pause-DR state allows the test controller to temporarily halt the shifting of data through the test data register in the serial path between TDI and TDO. The test data register selected by the current instruction retains its previous value during this state. The current instruction does not change in this state.

The controller remains in this state as long as TMS is low. When TMS goes high on the rising edge of TCK, the controller moves to the Exit2-DR state.

26.4.5.8 Exit2-DR State

Exit2-DR State is a temporary state. If TMS is held high on the rising edge of TCK, the controller enters the Update-DR state, which terminates the scanning process. If TMS is held low on the rising edge of TCK, the controller enters the Shift-DR state.

The current instruction does not change while the TAP controller is in this state. The test data register selected by the current instruction retains its previous value during this state.

26.4.5.9 Update-DR State

The boundary-scan register is provided with a latched parallel output. This output prevents changes at the parallel output while data is shifted in response to the extest or sample/preload instructions. When the boundary-scan register is selected while the TAP controller is in the Update-DR state, data is latched onto the boundary-scan register's parallel output from the shift register path on the falling edge of TCK. The data held at the latched parallel output does not change unless the controller is in this state.

While the TAP controller is in this state, the test data register selected by the current instruction retains its previous value. The current instruction does not change while the TAP controller is in this state.

When the TAP controller is in this state and TMS is held high on the rising edge of TCK, the controller enters the Select-DR-Scan state. If TMS is held low on the rising edge of TCK, the controller enters the Run-Test/Idle state.

26.4.5.10 Select-IR-Scan State

Select-IR-Scan is a temporary controller state. In this state, the test data register selected by the current instruction retains its previous value. If TMS is held low on the rising edge of TCK in this state, the controller moves into the Capture-IR state, and a scan sequence for the instruction register is initiated. If TMS is held high on the rising edge of TCK, the controller moves to the Test-Logic-Reset state.

The current instruction does not change in this state.

26.4.5.11 Capture-IR State

When the controller is in the Capture-IR state, the shift register contained in the instruction register loads the fixed value 0b000_0001 on the rising edge of TCK. See Table 26-2 for the instruction associated with this value.

The test data register selected by the current instruction retains its previous value during this state. The current instruction does not change in this state. In the Capture-IR state, holding TMS high on the rising edge of TCK causes the controller to enter the Exit1-IR state. If TMS is held low on the rising edge of TCK, the controller enters the Shift-IR state.

26.4.5.12 Shift-IR State

When the controller is in the Shift-IR state, the shift register contained in the instruction register is connected between TDI and TDO and shifts data one bit position nearer to its serial output on each rising edge of TCK. The test data register selected by the current instruction retains its previous value during this state. The current instruction does not change.

If TMS is held high on the rising edge of TCK, the controller enters the Exit1-IR state. If TMS is held low on the rising edge of TCK, the controller remains in the Shift-IR state.

26.4.5.13 Exit1-IR State

Exit1-IR is a temporary state. If TMS is held high on the rising edge of TCK, the controller enters the Update-IR state, which terminates the scanning process. If TMS is held low on the rising edge of TCK, the controller enters the Pause-IR state. The test data register selected by the current instruction retains its previous value during this state. The current instruction does not change and the instruction register retains its state.

26.4.5.14 Pause-IR State

The Pause-IR state allows the test controller to temporarily halt the shifting of data through the instruction register. The test data register selected by the current instruction retains its previous value during this state. The current instruction does not change and the instruction register retains its state.

The controller remains in this state as long as TMS is held low. When TMS goes high on the rising edges of TCK, the controller moves to the Exit2-IR state.

26.4.5.15 Exit2-IR State

Exit2-IR is a temporary state. If TMS is held high on the rising edge of TCK, the controller enters the Update-IR state, which terminates the scanning process. If TMS is held low on the rising edge of TCK, the controller enters the Shift-IR state.

The test data register selected by the current instruction retains its previous value during this state. The current instruction does not change and the instruction register retains its state.

26.4.5.16 Update-IR State

The instruction shifted into the instruction register is latched onto the parallel output from the shift register path on the falling edge of TCK. After the instruction is latched, it becomes the current instruction. The test data register selected by the current instruction retains its previous value.

If TMS is held high on the rising edge of TCK, the controller enters the Select-DR-Scan state. If TMS is held low on the rising edge of TCK, the controller enters the Run-Test/Idle state.

26.5 Register Descriptions

This section describes the registers used in JTAG testing.

26.5.1 JTAG Device Identification (ID) Register

The read-only ID register, defined in Table 26-5, is for reading the 32-bit device identification code. No programmable supplementary identification code is provided.

When the idcode instruction is selected, the ID register is selected as the serial path between TDI and TDO. The 32-bit device identification code is loaded into the ID register during the Capture-DR state from its parallel inputs.

Table 26-5. JTAG Device Identification (ID) Register

JTAG access only	JTAG Device Identification (ID) Register	PXA27x Processor Test Controller

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Version				Part Number																				JEDEC Code							
Reset	†	†	†	†	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	1

Bits	Access	Name	Description
31:28	R	Version	The version number changes with each new revision of silicon. Processor stepping: 0b0000 = A0 0b0001 = A1 0b0010 = B0 0b0011 = B1 0b0100 = C1
27:12	R	Part Number	The part number of the PXA27x processor is subject to change: 0b1001_0010_0110_0101 (0x9265)
11:0	R	JEDEC Code	The JEDEC code is the manufacturer identification number: 0b0000_0001_0011 (0x013)
† These values reflect the actual production identification and revision numbers embedded in the PXA27x processor.			

26.5.2 JTAG Test Data Registers

Refer to Section 26.4.4, “Test Data Registers,” for details of the JTAG test registers.

26.5.3 Debug Registers

Refer to Chapter 26, “Software Debug,” of the *Intel® PXA27x Processor Family Developers Manual* for detailed descriptions of the debug registers accessible through co-processor instructions.

26.6 Test Register Summary

See Table 26-6 for the list of the registers used in testing and references the description for each register.

Table 26-6. Test Register Summary

Register	Reference Page
JTAG Device Identification (ID) Register	26-15
Bypass Register	26-7
Boundary-Scan Register	26-8
Data-Specific Registers	26-9
Flash Data Register	26-9
Intel XScale® Data Registers	26-9
Software debug registers	Chapter 26, “Software Debug,” *Intel® PXA27x Processor Family Developers Manual*

§§

Intel® Quick Capture Technology 27

This chapter describes the guidelines for connecting camera image sensors and sensor modules to the Intel® PXA27x processor's quick capture interface. The PXA27x processor supports a wide variety of operating modes, data widths, formats, and clocking schemes. Only a subset of these modes are described in this chapter.

27.1 Overview

The quick capture interface is intended for use in a PDA or mobile phone application requiring image capture capability. Some common usage scenarios include:

- Capturing simple still images, sharing images using email, and sending images to a web photofinisher
- Using images for "Pictures-as-Information"
- Capturing video clips
- Providing 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 "megapixel" resolution
- 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 fps

27.2 Feature List

The functions of the quick capture interface:

- Acquiring both data and control signals from a camera image sensor
- Formatting of the data appropriately prior to being routed to memory through DMA

The features of the quick capture interface include:

- Parallel interface support for 8, 9, and 10 bits
- Serial interface support for 4-bit and 5-bit data bus connections
- Support for ITU-R BT.656 Start-of-Active-View (SAV) and End-of-Active View (EAV) embedded signaling
- Pre-processed capture modes, such as RGB and YCbCr
- Raw capture modes, such as RGGB and CMYG
- Programmable vertical & horizontal resolutions up to 2048 x 2048
- Two 8-entry (by 64-bit) and one 16-entry (by 64-bit) FIFOs
- Programmable sensor clock output from 196.777 KHz to 52 MHz
- Programmable interface timing signals for both internal and external synchronization
- Programmable interrupts for FIFO overflow, End-of-Line (EOL), and End-of-Frame (EOF)

27.3 Signals

See Table 27-1 for the list of signals used by the quick capture interface.

Table 27-1. Signal Descriptions for Quick Capture Technology

Signal Name	Type	Description
CIF_DD[9:0]	Input	Data lines to transmit 4,5,6,7,8,9 or 10 bits at a time
CIF_MCLK	Output	Programmable output clock used by the camera capture sensor
CIF_PCLK	Input	Pixel clock used by the quick capture interface of the camera to clock the pixel data into the input FIFO
CIF_LV	I/O	Line start or alternate synchronization signal used by the sensor to signal line read-out or as an external horizontal synchronization
CIF_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 are typically met through the use of standard GPIOs. A couple of common examples are “sensor reset” or “sensor power-down .”

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.

27.4 Block Diagram

See Figure 27-1 for illustration of a typical 8-bit master parallel connection between the PXA27x processor and an image sensor module. It is important to note that master mode refers to the case where the sensor module drives the line and frame synchronization signals. Slave mode is the case where the PXA27x processor drives the line and frame synchronization signals. See Figure 27-2 for an interface options summary.

Figure 27-1. Block Diagram for 8-bit Master Parallel Interface

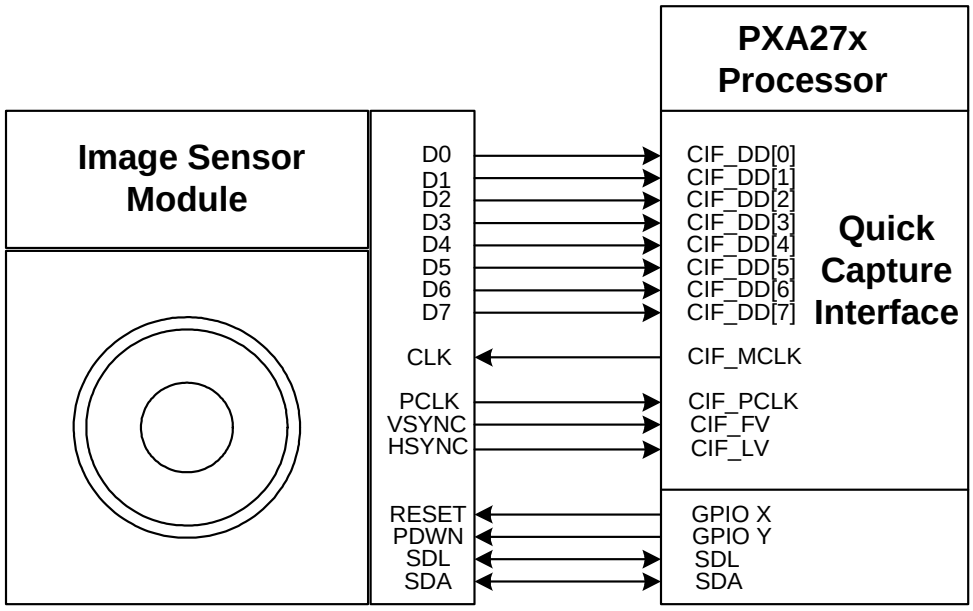
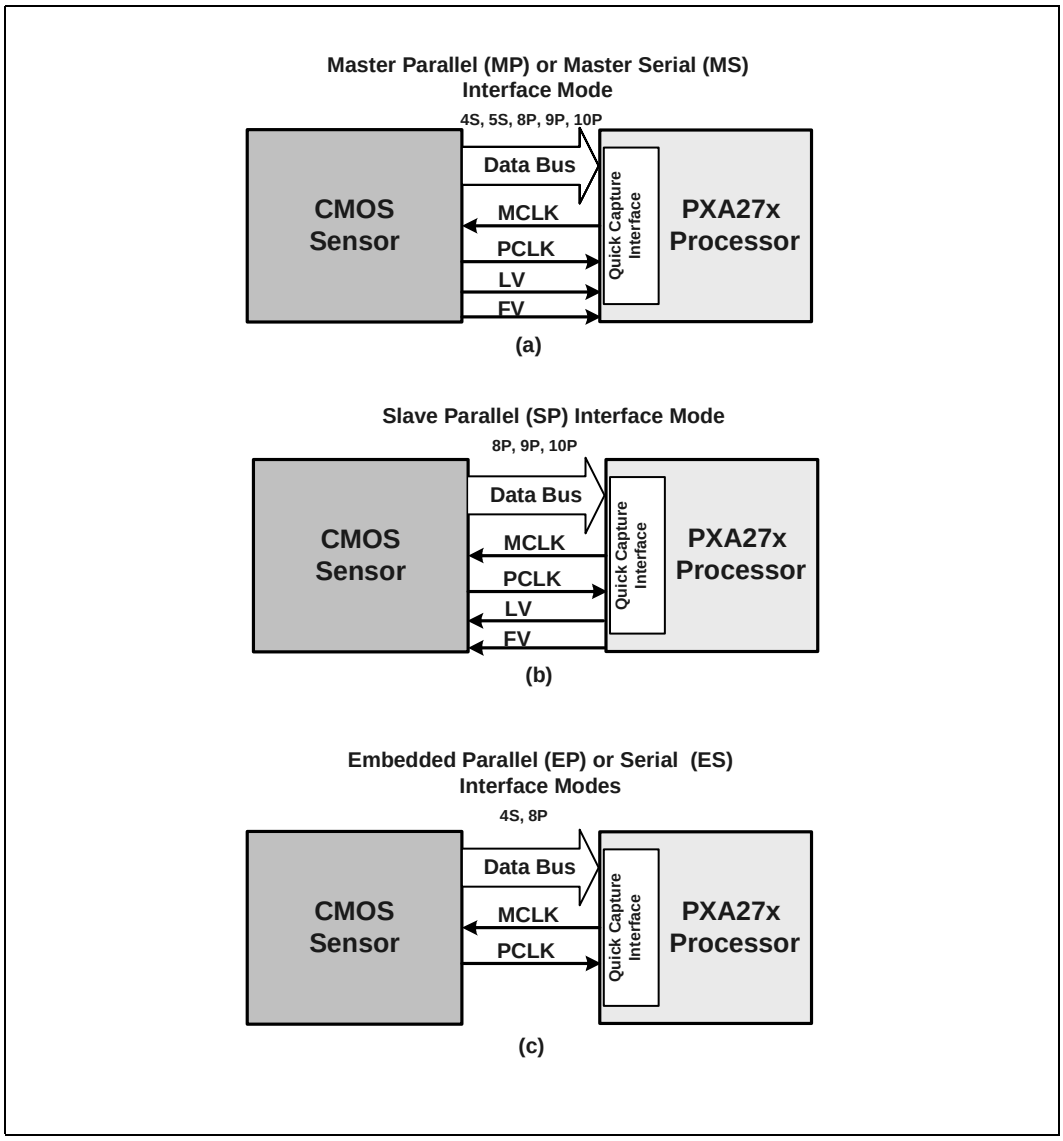


Figure 27-2. Interface Options Summary



§§

PXA27x DVK Block Diagram A

This chapter contains the DVK (formerly NBMMNS2BVS DVK and formerly NBMMNS3BVS DVK) block diagrams of Intel® PXA27x Processor Family (PXA27x processor).

Figure A-1. System Overview Block Diagram

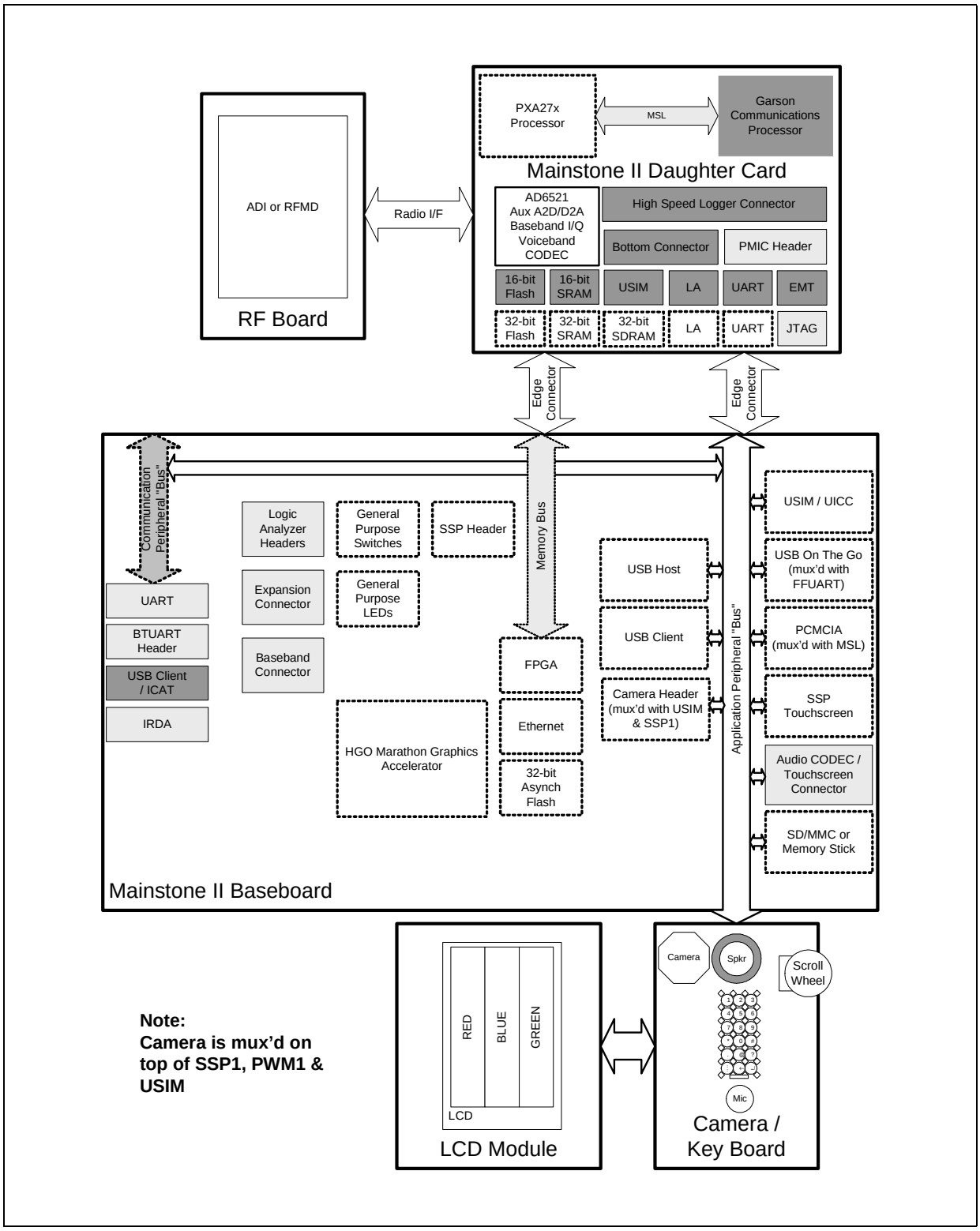


Figure A-2. Main Board Block Diagram

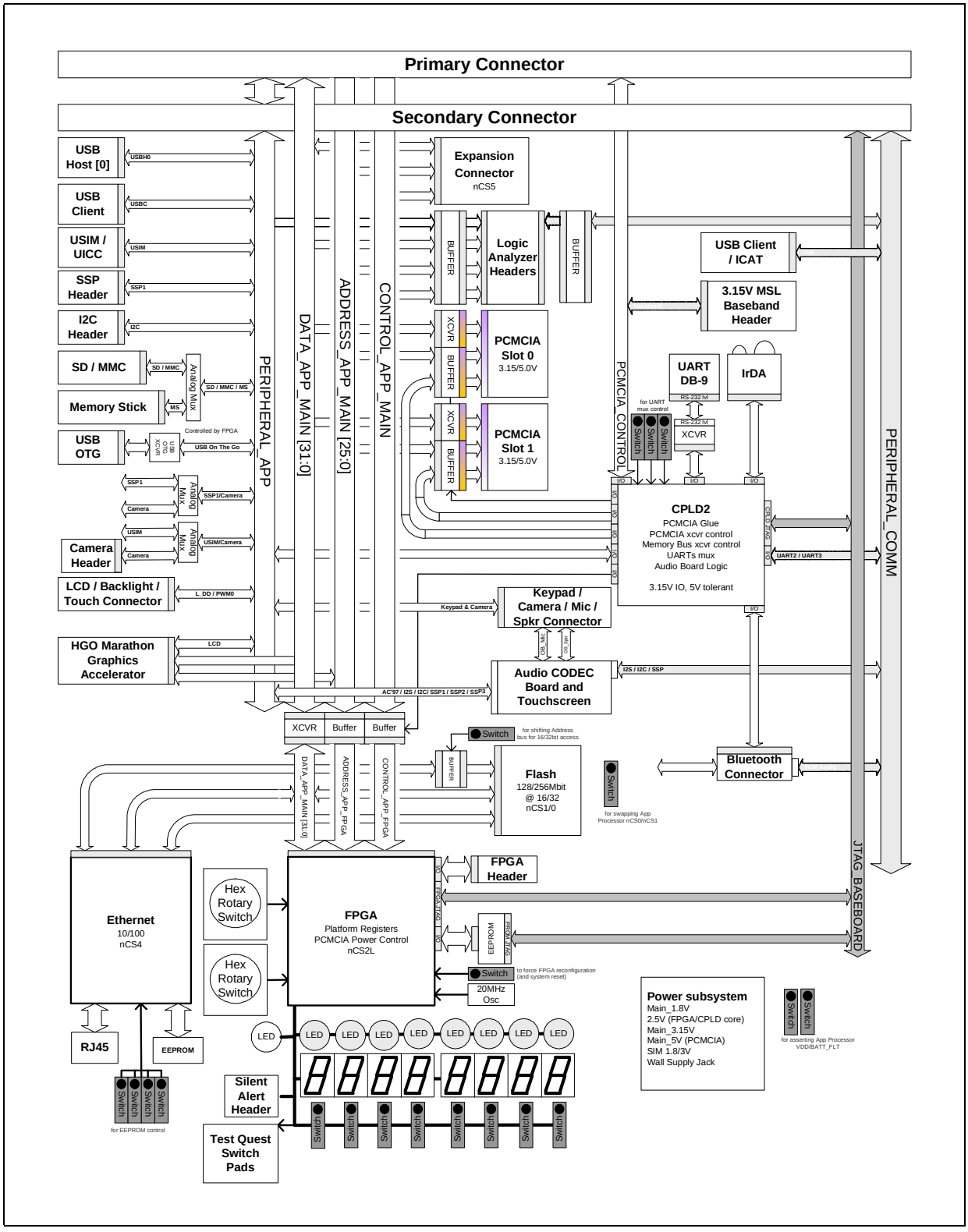
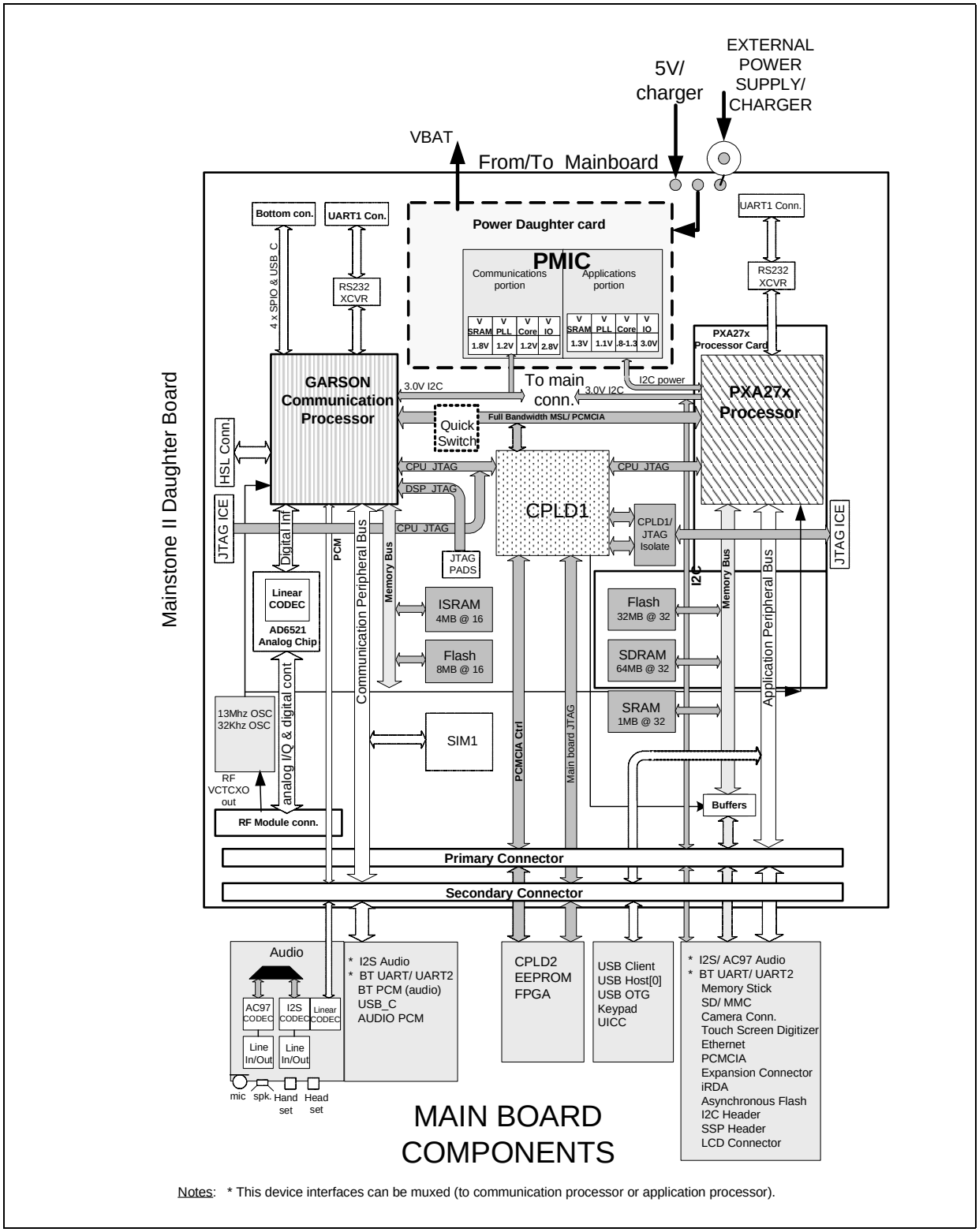


Figure A-3. Daughter Card Block Diagram



Notes: * This device interfaces can be muxed (to communication processor or application processor).

Figure A-4. Liquid Crystal Display Block Diagram

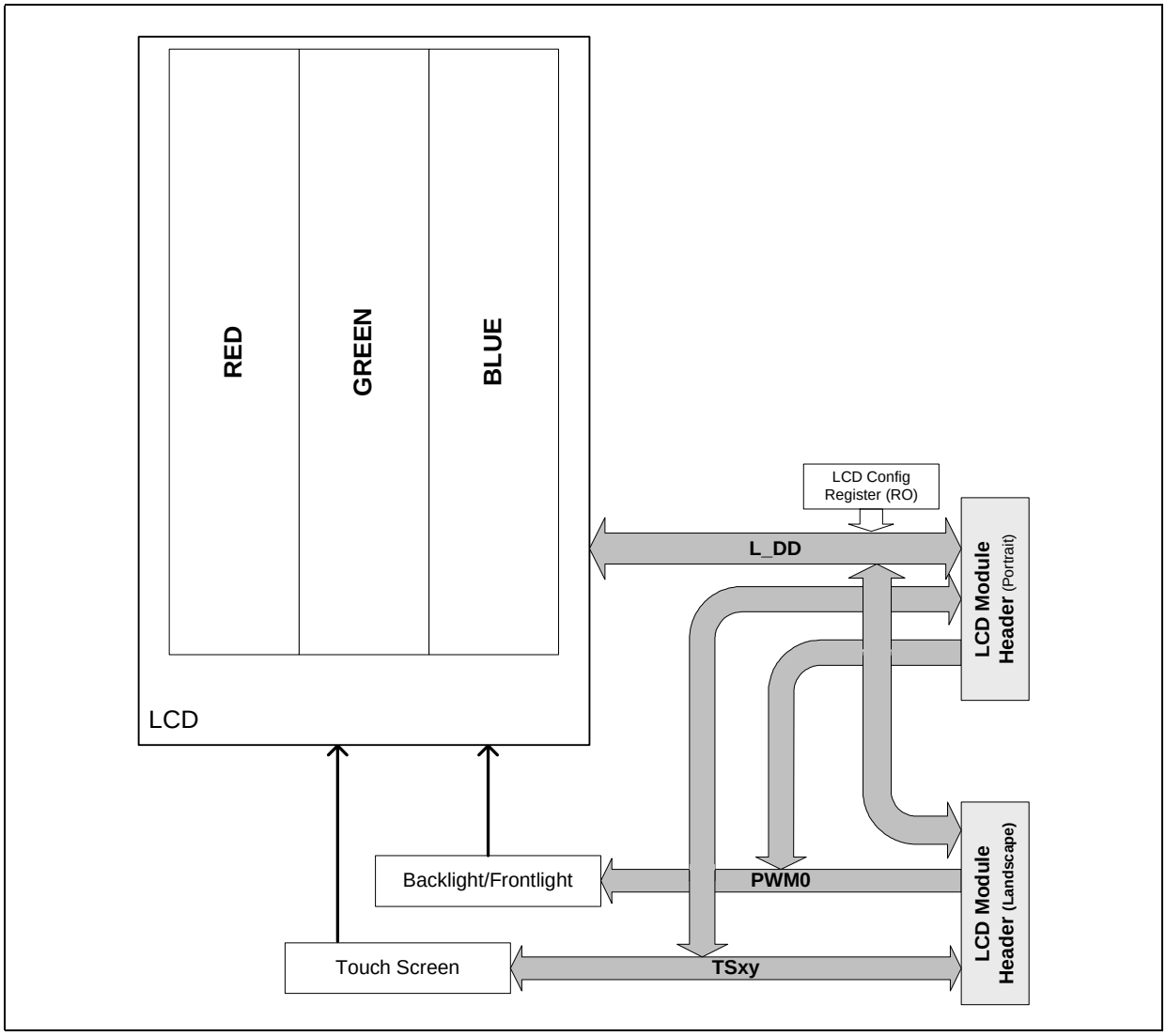


Figure A-5. Audio Module Block Diagram

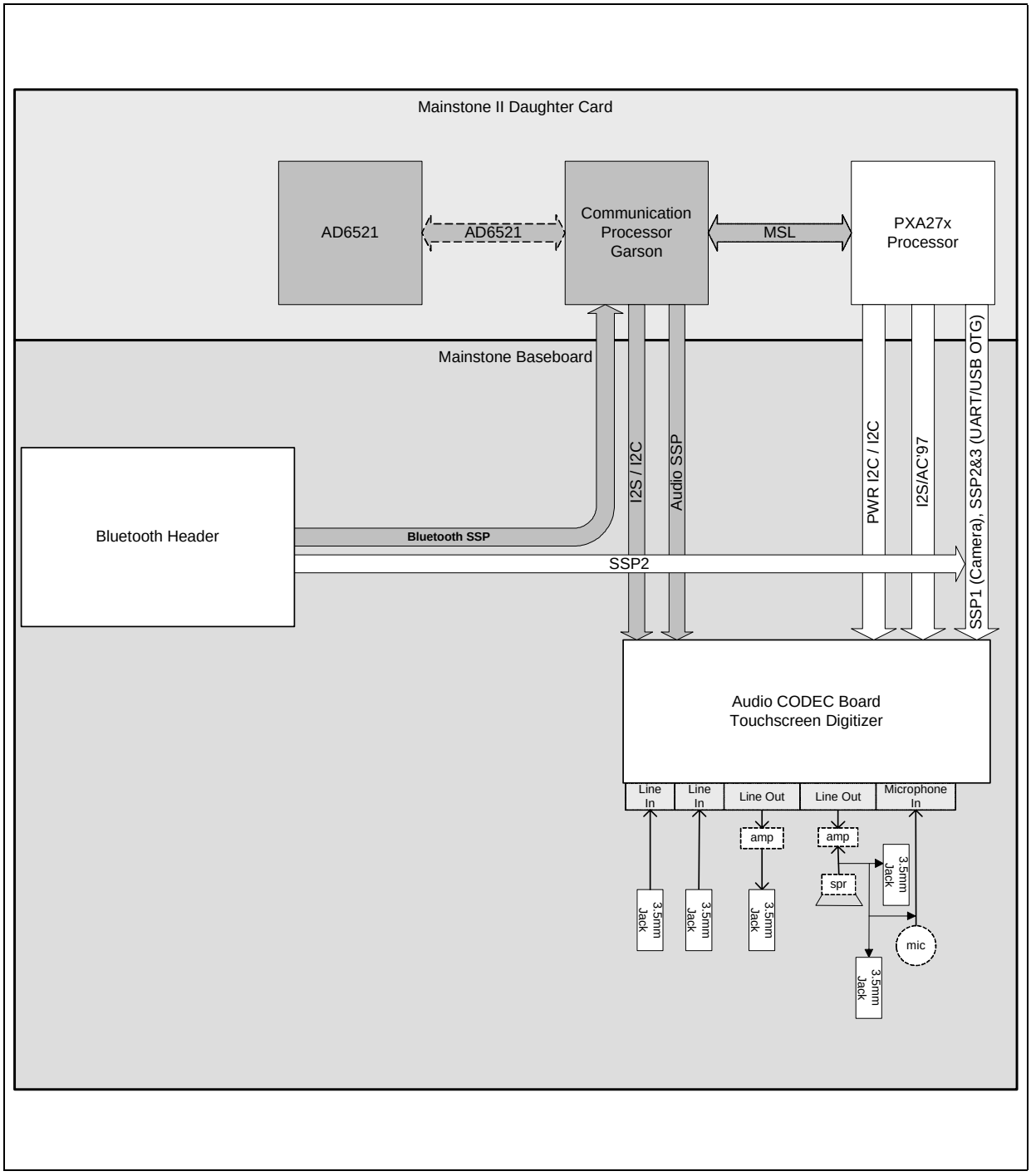


Figure A-6. Keyboard Block Diagram

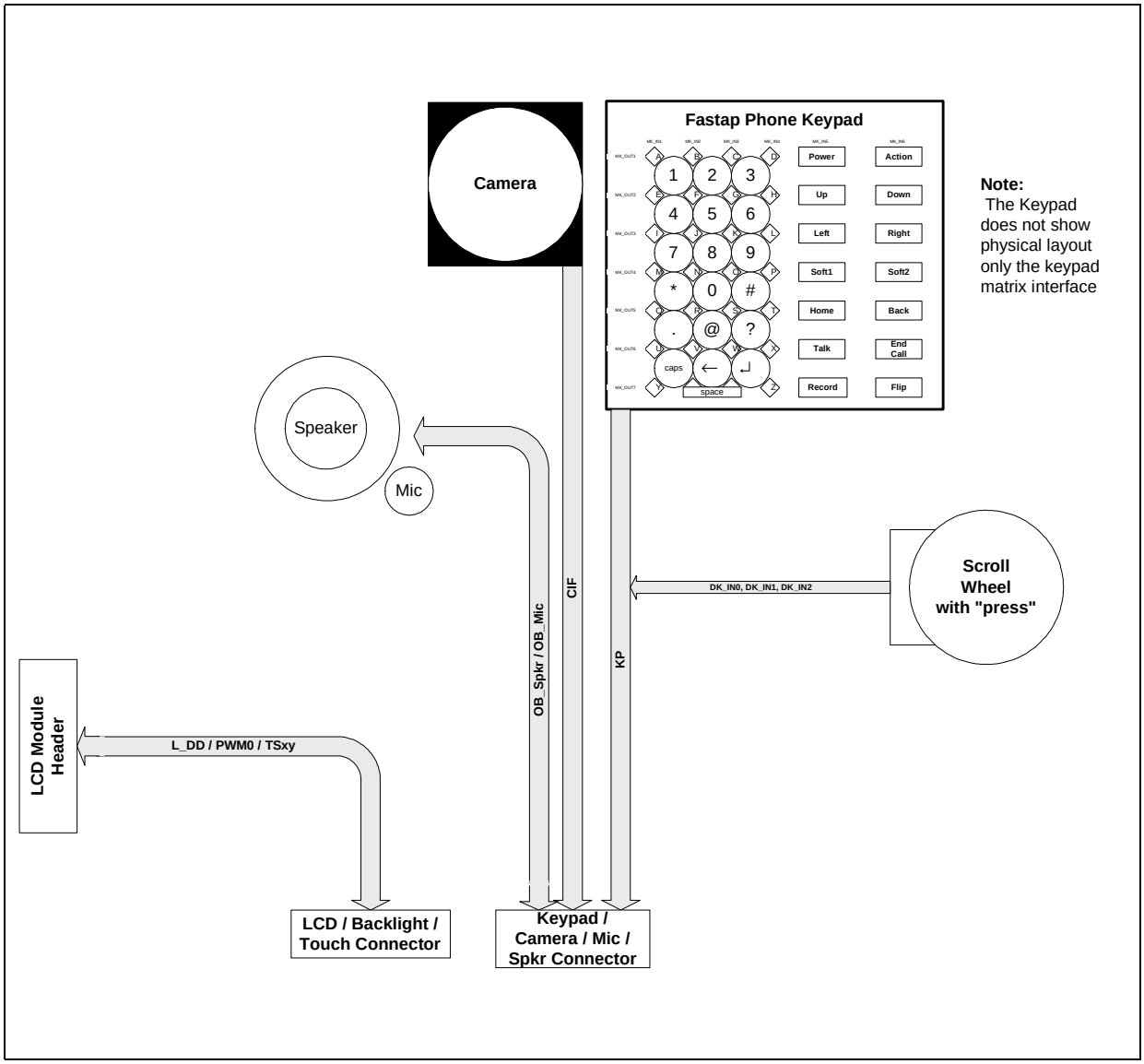
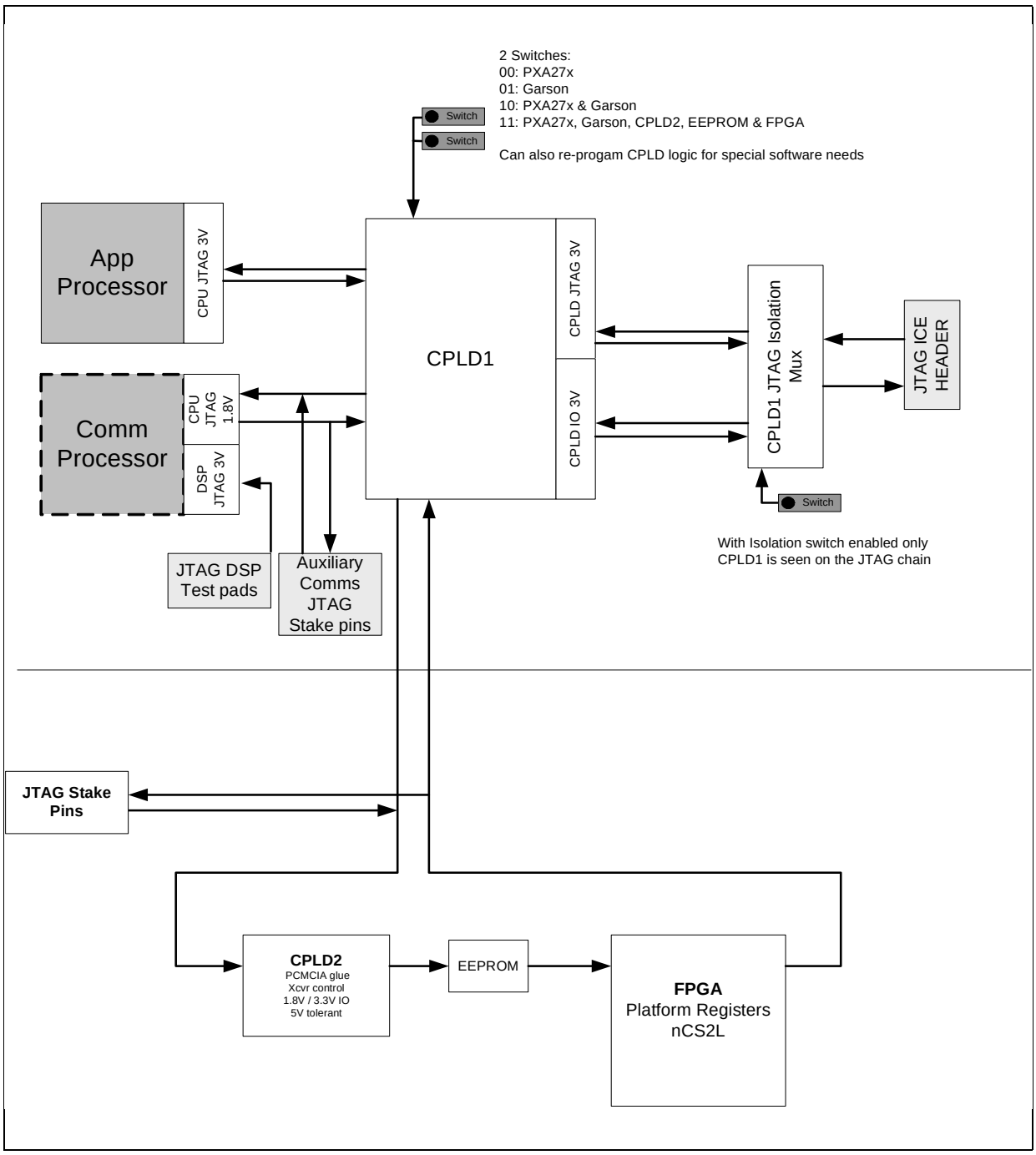


Figure A-7. JTAG Block Diagram



§§

PXA27x Processor Developer's Kit (DVK) B

This chapter lists documents for:

- Intel® PXA27x Processor Developer's Kit (formerly NBMMNS3BVS DVK). Refer to Table B-1.
- Intel® PXA27x Processor Developer's Kit (formerly NBMMNS2BVS DVK). Refer to Table B-2.
- Intel® PXA27x Processor Developer's Kit Schematics. Refer to Table B-3.

Table B-1. Processor Developer's Kit (formerly NBMMNS3BVS DVK)

Order #	Description
	Intel® NBMMNS3BVS DVK for Intel® Personal Internet Client Architecture Quick Start Guide
	Intel® NBMMNS3BVS PXA27x DVK for Intel® Personal Internet Client Architecture Parts List
	Intel® NBMMNS3BVS DVK for Intel® Personal Internet Client Architecture Schematics
	Intel® NBMMNS3BVS and NBMMNS2BVSDVK PMIC LDO Card for Intel® Personal Internet Client Architecture Specification Update
	Intel® NBMMNS2BVS and NBMMNS3BVS DVK Daughter Cards for Intel® PCA Specification Update
	Intel® NBMMNS3BVS Developers Kit for Intel® Personal Internet Client Architecture User's Guide

Table B-2. Processor Developer's Kit (formerly NBMMNS2BVS DVK)

Order #	Description
	Intel® NBMMNS2BVS PXA27x DVK for Intel® Personal Client Architecture Schematics
	Intel® NBMMNS2BVS PXA27x DVK for Intel® Personal Internet Client Architecture Parts List
	Intel® PXA27x DVK Main Board for Intel® Personal Internet Computing Architecture Specification Update
	Intel® NBMMNS2BVS and NBMMNS3BVS DVK Daughter Cards for Intel® Personal Internet Client Architecture Specification Update
	Intel® NBMMNS3BVS and NBMMNS2BVS DVK PMIC (LDO) Card for Intel® Personal Internet Computing Architecture Specification Update
	Intel® NBMMNS2BVS DVK for Intel® Personal Internet Client Architecture Quick Start Guide
	Intel Power Manager for NBMMNS2BVS DVK and NBMMNS2BVGS DVK Application Note
	Intel Diagnostics for NBMMNS2BVS DVK for Intel® Personal Internet Client Architecture User's Guide
	Upgrading Processor Cards for NBMMNS2BVS DVK and NBMMNS2BVGS DVK Quick Start Guide

Table B-3. Processor Developer's Kit

Order #	Description
	Intel® PXA27x Processor Developer's Kit Schematics
	Intel® PXA27x Processor Developer's Kit Parts List
	Intel® PXA27x Processor Developer's Kit Quick Start Guide
	Board Bring Up (BBU) Program for the Intel® PXA270 Processor Developer's Kit Release Notes
	Board Bring Up (BBU) Program for the Intel® PXA27x Processor MultiChip Product Developer's Kit Release Notes
	Power-On Self-Test (POST) for the Intel® PXA270 Processor Developer's Kit Release Notes
	Power-On Self-Test (POST) for the Intel® PXA27x Processor MultiChip Product Developer's Kit Release Notes
	Intel® PXA27x Processor Developer's Kit User's Guide
	Upgrading Processor Cards for the Intel® PXA270 Processor and Intel® PXA27x Processor MultiChip Product Quick Start Guide
	Intel® PXA27x Processor Developer's Kit Main Board Specification Update
	Intel® PXA27x Processor Developer's Kit Daughter Card Specification Update
	Intel® PXA27x Processor Developer's Kit PMIC (LDO) Card Specification Update
	Diagnostics for the Intel® PXA27x Processor Developer's Kit User's Guide
	Intel Power Manager for the Intel® PXA27x Processor Developer's Kit Application Note
	Intel® PXA27x Processor Developer's Kit Board Support Package for Microsoft Windows* Mobile* 2003 for Pocket PC* Release Notes
	Intel® PXA27x Processor Developer's Kit Board Support Package for Microsoft Windows* Mobile* 2003 for Smartphone* Release Notes
	Intel® PXA27x Processor Developer's Kit Board Support Package for Microsoft Windows* Mobile* 2003 for Pocket PC* User's Guide
	Intel® PXA27x Processor Developer's Kit Board Support Package for Microsoft Windows* Mobile* 2003 for Smartphone* User's Guide
	Intel® PXA27x Processor Developer's Kit Board Support Package for Palm OS* Release Notes
	Intel® PXA27x Processor Developer's Kit Board Support Package for Symbian OS* Release Notes
	Intel® PXA27x Processor Developer's Kit Board Support Package for Palm OS* User's Guide
	Intel® PXA27x Processor Developer's Kit Board Support Package for Symbian OS* User's Guide

§§

PXA27x DVK Bill-of-Materials C

For the Bills of Materials (BOM) that correspond to the current revision of Intel® PXA27x Processor Family (PXA27x processor) DVK, refer to Appendix B , “PXA27x Processor Developer’s Kit (DVK).”

To locate a specific parts list, refer to the appropriate tables in Appendix B:

- Intel® NBMMNS3BVS PXA27x DVK for Intel® Personal Internet Client Architecture Parts List (Table B-1).
- Intel® NBMMNS2BVS PXA27x DVK for Intel® Personal Internet Client Architecture Parts List (Table B-2).
- Intel® PXA27x Processor Developer’s Kit Parts List (Table B-3).

§§

Intel® PXA27x Processor and Intel® PXA25x Processor Differences D

D.1 Introduction

This appendix describes the changes and enhancements found in the Intel® PXA27x Processor Family (PXA27x processor) compared to the Intel® PXA25x processor. The differences are explained separately for each peripheral.

D.2 System Architecture

The system architecture has received these improvements in the PXA27x processor:

- The addition of the Concan SIMD engine that allows for significant calculation performance improvements.
- Performance monitor coprocessor for evaluation and tuning of system performance.

D.3 Clock and Power Manager

The PXA27x processor provides significant enhancements in the clock and power manager units relative to the PXA25x processor. These enhancements reduce power consumption and provide new functionality to give system designers the flexibility to design more powerful systems with even better power efficiency.

D.3.1 Clock Manager

The PXA27x processor uses a 13-MHz master clock instead of the 3.6864-MHz clock used by the PXA25x processor. PXA27x processor provides two internal PLLs. One PLL generates all peripheral timing and the other generates the processor core timing. The PXA25x processor requires three internal PLLs to generate core and peripheral timing.

In addition, the PXA27x processor provides these features that are not present in the PXA25x processor:

- Three clock-speed controls to adjust frequency: turbo mode, divisor mode, and fast-bus mode
- Switchable clock source
- Functional clock gating
- Larger core clock operating frequency range (33 - 403 MHz)

D.3.2 Power Manager

The PXA27x processor provides new low power modes not available in the PXA25x processor. See Table D-1 for the description of the new power modes.

Table D-1. PXA27x Processor Operating Modes not Supported by the PXA25x Processor

Operating Mode	Description
Standby mode	The clocks to the CPU are disabled and the CPU is placed in a low leakage state, but context is retained. All external power supplies are enabled. Each internal SRAM bank are placed in a low power mode where state is retained, but no activity is allowed. The PLLs are disabled and peripheral operation is suspended. An interrupt assertion causes the transition back to Normal mode.
Deep-sleep mode	All internal power domains except VCC_RTC and VCC_OSC are powered down. All clock sources except the Real Time Clock and Power Manager are disabled and the external power supplies are disabled. The active internal power domains are powered from one of three internal regulators driven from the backup battery pin, VCC_BATT. Recovery is initiated by external and select internal wake-up events and requires a system reboot.

The PXA27x processor power manager provides support for additional wake-up event sources from new units including the baseband interface, keypad, and USB controllers.

The PXA27x processor power manager also provides a programmable I^2C-based external regulator interface, which is not on the PXA25x processor, to support dynamic voltage and frequency management.

D.4 Internal Memory

The internal memory controller has been added to the PXA27x processor. This controller does not exist in the PXA25x processor. The features of the internal memory controller are listed in the internal memory chapter of the *Intel® PXA27x Processor Family Developers Manual*.

D.5 DMA Controller

The DMA controllers in the PXA27x processor and PXA25x processor are similar with these exceptions:

- Features added to the DMA controller of the PXA27x processor:
  - Additional 16 DMA channels for a total of 32 DMA channels
  - Support fly-by transfers
- Enhancements to the DMA controller of the PXA27x processor:
  - Descriptor Compare and Branching Mode
  - Byte boundary alignment for source and target addresses (down from 64-bit aligned)
  - End of Receive status/control bits for transfers from the internal peripherals
- Support for big endian transfers to and from any DMA devices has been removed.
- Signals added to the PXA27x processor DMA controller:
  - DVAL<1:0> - Data Valid for fly-by transfers
- Changes to the registers within the PXA25x processor DMA controller were required to support the new features in the PXA27x processor DMA controller:
  - FLYCNFG registers have been added to support fly-by transfers to external SDRAM.
  - These registers have been added to support the additional internal peripheral device requests:
    - DRCMR38 - DRCMR67
  - These registers have been added to support the additional 16 DMA channels:
    - DCMD16 - DCMD31
    - DDADR16 - DDADR31
    - DSADR16 - DSADR31
    - DTADR16 - DTADR31
  - The DDADRx, DCMDx, and DCSRx registers have additional bits added to support the descriptor branch and compare mode and the end of receive control and status functionality.
  - The DINT register had additional bit added to support the additional 16 DMA channels.
  - The DRCMRx register has the CHLNUM bitfield increased by 1.

Note: The registers that remains unchanged are the DSADRx and DTADRx registers.

D.6 Memory Controller

The memory controllers in the PXA27x processor and PXA25x processor are similar with the following exceptions.

- Features added to the PXA27x processor memory controller:
  - — Support for low power SDRAM includes:
    - ◆ Support for 1.8 volt memory I/O
    - ◆ Additional MRS register for low power SDRAM support
  - — Support 128 Mbytes/256 Mbytes partitions
  - — Programmable Buffer Strength on memory I/O signals
- Enhancement to the memory controller of the PXA27x processor:
  - — SDCLK<0> divide down option by 4 with respect to CLK_MEM
- Features removed from the memory controller of the PXA27x processor:
  - — Support for Synchronous Mask ROM (SMROM)
- Signals removed from the memory controller of the PXA27x processor:
  - — SDCKE<0>
  - — BOOT_SEL<2:1>

Changes to the registers within the memory controller of the PXA25x processor were required to support the new features and removal of support for the SMROM in the memory controller of the PXA27x processor:

- — The MDCNFX register has been added to support larger SDRAM partitions.
- — The SCNTR0 - BSNTR3 registers have been added to support programmable buffer strength.
- — The MDMRSLP register has been added to support low power SDRAM MRS command.
- — The MDREFR register has additional bit added to support SDCLK0 divide by 4 option.
- — The SXCNFG and BOOT_DEF registers have additional bits removed as a result of not supporting SMROM in the PXA27x processor memory controller.
- — The registers that remain unchanged are:
  - ◆ MDCNFG
  - ◆ MDMRS
  - ◆ MSC0 - MSC2
  - ◆ MCMEM0
  - ◆ MCMEM1
  - ◆ MCATT0
  - ◆ MCATT1
  - ◆ MCIO0
  - ◆ MCIO1

D.7 LCD Controller

The LCD controllers in the PXA27x processor and PXA25x processor are similar with these exceptions:

- Features added to the PXA27x processor memory controller:
  - Support for these Display modes:
    - Up to 16777216 colors (24 bits) in active color mode
    - A total of 16777216 colors (24 bits) in passive color mode
    - Up to 24-bit per pixel single-panel color displays
    - LCD panel with internal Frame buffer
  - Larger output FIFOs (64-entry by 24 bits)
  - Three 256-entry by 25-bits internal color-palette RAMs (one for each overlay and Base) programmable to be automatically loaded at the beginning of each frame
  - Command data RAM (16 x 9 bits) to hold command data
  - Additional support for pixel depths of 18, 19, 24 and 25 bpp RGB formats
  - No direct support for 12 bpp RGB format (indirectly supported)
  - One base layer plus two overlays for single-panel displays; maximum size of each overlay is equal the display size
  - Programmable transparency for overlays
  - Integrated seven-channel DMA (one channel for base plane, one channel for Overlay 1 and three channels for Overlay 2, one channel for the hardware cursor, and one channel for the command data)
  - Hardware support for color-space conversion from YCbCr to RGB for video streams
  - Support for hardware cursor for single-panel display
  - Programmable pixel clock from 48.75 MHz to 50.8 KHz (97.5 MHz/2 to 26 MHz/512)
  - Six 16 x 64-bit Input FIFOs: one for base channel, one for Overlay 1, three for Overlay 2, and one for the hardware cursor; one 4 x 52-bit Input FIFO for Command data for panels with internal Frame buffer
  - Four additional pins
- Changes to the registers within the PXA25x processor memory controller were required to support the new features and removal of support for the PXA27x processor LCD controller:
  - LCCR0 expanded capabilities for command functions.
  - LCCR3 expanded for number of bits per pixel and palette format capabilities.
  - LCCR4 register has been added to support transparency control and palette data formats.
  - LCCR5 register has been added to support interrupts for the new overlays.
  - OVL1C1, OVL1C2, OVL2C1 and OVL2C2 registers added to support overlays.
  - CCR register has been added to support cursor.
  - CMDCR register has been added to support frame buffer panels.

— TCR register has an additional field to enhance TMED support.

— FDADR1-FDADR6 registers added to support new overlays and cursor.

— FBR1-FBR6 registers added to support new overlays and cursor.

— FSADR1-FSADR6 registers added to support new overlays and cursor.

— FIDR1-FIDR6 registers added to support new overlays and cursor.

— LDCMD1-LDCMD6 registers added to support new overlays and cursor.

— LCDBSCNTR register has been added to support programmable output buffer strength.

— PRSR register has been added to support reading data from frame buffer panels.

— LCSR0 register has been expanded to support more channels and read status for frame buffer panels.

— LCSR1 register has been added to support the new overlays.

— The registers that remains unchanged are: LCCR1, LCCR2, TRGBR, FDADDR0, FBR0, FSADR0, FIDR0, LDCMD0 and LIIDR.

D.8 SSP Serial Port

The PXA27x processor provides three SSPs whereas the PXA25x processor provides a single SSP.

The PXA27x processor SSP controller retains compatibility with the original protocols implemented in the PXA25x processor and adds support for a new Programmable Serial Protocol (PSP). The PSP supports register programmable frame sync in addition to programmable start and stop delays.

The PXA27x processor SSP controller also provides greater flexibility than the PXA25x processor SSP controller through these enhancements:

- Transfer rates up to 13 Mbps
- 4-bit to 32-bit serial data transfers
- Master or slave operation for both clock and frame sync signals
- Flexible clock source selection from the 13-MHz master clock, the network clock input, or the dedicated SSP external clock input

D.9 I²C Bus Interface Unit

The I²C controllers in the PXA27x processor and PXA25x processor are similar with these exceptions:

- PXA27x processor contains a second I²C controller targeted for use with a power management IC.
- For the PWR_I²C controller only, these restrictions apply:
  - PWR_I²C supports standard-mode operation of 40 Kbps and fast-mode operation of 100 Kbps. When used with voltage change sequencer, PWR_I²C operates in standard-mode. When used outside of the voltage change sequencer, there are no restrictions to the use of the I²C.
  - When the Voltage Change Sequencer is operating, (see Section 3.5.1.14 of the *Intel® PXA27x Processor Family Developers Manual*), the PWR_I²C registers are not writable and reads unknown values. When Voltage Change Sequencer is operating, writing to PCMDx and PVCR results in unpredictable operation.
- When the PWR_I²C is used by the Voltage Change Sequencer, these restrictions apply:
  - The sequencer only allows master-transmitter operations to a single, predefined slave.
  - The sequencer does not send interrupts to the CPU, so interrupts other than “IDBR Transmit Empty” are ignored and causes failure of the transmission.
  - Fast mode of operation is not possible.

D.10 UARTs

The UART controllers in the PXA27x processor and the PXA25x processor are similar with these exceptions:

- The UARTs in the PXA27x processor is selected to use all 32 bits of the Peripheral Bus.
- There is an option in the PXA27x processor that is enabled where the DMA removes any trailing bytes in the receive FIFO. The PXA25x processor did not have this option and the processor performed this task.
- An autoflow mode has been added for the PXA27x processor.
- Auto-baud functionality has been added for the PXA27x processor.
- Registers FFFOR, BTFOR, and STFOR are new for FIFO occupancy information.
- Registers FFABR, FFACR, BTABR, BTACR, STABR, and STACR are new to support auto-baud functionality.
- Registers RBR, THR, IIR, FCR, and MCR have new bits.

D.11 Fast Infrared Communication Port

The FIR controllers in the PXA27x processor and PXA25x processor are similar with these exceptions:

- The FICP in PXA27x processor is selected to use all 32 bits of the Peripheral Bus.
- The FICP IRTXD and IRRXD signals are additionally multiplexed with the BTUART's BTTXD and BTRXD on the PXA27x processor. The IRTXD and IRRXD signals are only multiplexed with the STUART's STTXD and STRXD on the PXA25x meaning only one of these two interfaces (FICP or STUART) is used at any one time. The additional multiplexing on the PXA27x processor allows for any two of the three interfaces (FICP, STUART, and BTUART) to be used at any one time.
- There is an option in PXA27x processor which is enabled where the DMA removes any trailing bytes in the FICP FIFO. The PXA25x processor did not have this option and the processor performed this task.
- Register ICFOR is new.
- Register ICDR now utilizes all 32 bits; the PXA25x processor only uses 8 of the available 32 bits.
- Registers ICCR2 and ICSR0 have new bits.

D.12 USB Device Controller

The USB Device controllers (UDC) in the PXA27x processor and PXA25x processor are similar with these exceptions:

- Support for 24 Endpoints on the PXA27x processor compared to support for 16 Endpoints on the PXA25x processor.
- Endpoints are configurable on PXA27x processor UDC unlike the PXA25x processor UDC.
- PXA27x processor supports the Interrupt OUT EndPoint unlike the PXA25x processor.
- Maximum packet size for Isochronous Endpoints is increased to 1023 bytes for the PXA27x processor from 256 bytes on the PXA25x processor.
- The majority of the UDC registers have moved (address changed), have been renamed, and new registers have been added primarily because of the additional endpoints and endpoint programmability supported by the UDC.
- Register UDCCR has new bits, name changes on some bits, and new functionality on some bits.
- Register UDCCSR0 has new functionality on some bits.
- Registers UDCCSRx have new bits and new functionality on some bits.

D.13 AC ‘97

The AC ‘97 controllers in the PXA27x processor and PXA25x processor are similar with the exception of these features:

- An optional output AC97_SYSCLK (approximately 24.5 MHz) clock signal is available on the PXA27x processor.
- Clean shutdown functionality has been added with receive trailing bytes being handled differently in the PXA27x processor.
- The following registers have been changed:
  - Register GCR now utilizes bit 24 and some bit names have changed.
  - Register GSR has a new bit 3 and some bit names have changed.
  - Registers POCR, PICR, MCCR, MOCR, and MICR have new bit 1.
  - Registers POSR, PISR, MCSR, MOSR, and MISR have bit 4 functionality changes and bit 2 is new.

D.14 I^2S (Inter IC Sound Controller)

The I^2S controllers in the PXA27x processor and PXA25x processor are similar with the exception of these features:

- Features added to the PXA27x processor I^2S controller:
  - Clean startup functionality has been added.
  - Clean shutdown functionality has been added.
  - Serial audio clocks and sampling frequencies have changed slightly.
  - Register SASR0 has a new bit.

D.15 MultiMediaCard/SD/SDIO Controller

The MMC/SD/SDIO Card controllers in the PXA27x processor and PXA25x processor are similar with the exception of these features.

Features added to the PXA27x processor MMC/SD/SDIO card controller:

- Secure Digital (SD) and Secure Digital I/O communication protocols support available for PXA27x processor.
- One and three byte data transfers are supported for PXA27x processor.
- Three signals (MMDAT1, MMDAT2, and MMDAT3) have been added for 4-bit SD Card and SDIO Card protocol data support; the PXA25x processor only had one data signal.
- Registers MMC_STRPCL and MMC_STAT have new bits and functionality of other bits have changed from the PXA25x processor implementation.
- Registers MMC_RDWAIT and MMC_BLKS_REM have been added to the PXA27x processor.
- Register MMC_CMDAT has an additional bit.

D.16 Baseband/Multimedia Interface

The Baseband/Multimedia Interface has been added to the PXA27x processor. This controller does not exist in the PXA25x processor.

The features of the Baseband/Multimedia Interface are listed in the Baseband and/Multimedia Interface chapter of the *Intel® PXA27x Processor Family Developers Manual*.

D.17 Memory Stick Host Controller

The Memory Stick Host Controller has been added to the PXA27x processor. This controller does not exist in the PXA25x processor.

The features of the Memory Stick Host Controller are listed in the Memory Stick Host controller chapter of the *Intel® PXA27x Processor Family Developers Manual*.

D.18 Keypad Interface

The keypad controller has been added to the PXA27x processor. This controller does not exist in the PXA25x processor.

The features of the keypad controller are listed in the Keypad Interface chapter of the *Intel® PXA27x Processor Family Developers Manual*.

D.19 Universal Subscriber ID Interface

The Universal Subscriber ID Interface has been added to the PXA27x processor. This controller does not exist in the PXA25x processor.

The features of the Universal Subscriber ID Interface are listed in the *Intel® PXA27x Processor Family Developers Manual* Universal Subscriber Interface chapter.

D.20 Universal Serial Bus Host Controller

The Universal Serial Bus Host controller has been added to the PXA27x processor. This controller does not exist in the PXA25x processor.

The features of the Universal Serial Bus Host controller are listed in the Universal Serial Bus Host controller chapter of the *Intel® PXA27x Processor Family Developers Manual*.

D.21 Real-Time Clock (RTC)

The real-time clock (RTC) controller of the PXA27x processor is similar to that of the PXA25x processor except for these changes:

- Features added to the PXA27x processor real time clock:
  - — Timer Section
    - ◆ Resolution is now required to be one second
  - — Wristwatch Section
    - ◆ User-programmable, free-running counter containing time of the day in terms of hours, minutes, seconds, day of week, week of month, day of month, month, and year
    - ◆ User-programmable alarm registers to generate alarms in terms of hours, minutes, seconds, day of week, week of month, day of month, month, and year
    - ◆ Resolution of one second
  - — Stopwatch Section
    - ◆ Programmable counter register that contains the time elapsed between two events in terms of hours, minutes, seconds, and hundredths of a second
    - ◆ Two user-programmable alarm registers to generate alarms in terms of hours, minutes, seconds, and hundredths of a second
    - ◆ Resolution of one 100^{th} of a second
  - — Periodic Interrupt Section
    - ◆ Programmable alarm register to generate periodic interrupts at regular intervals
    - ◆ Resolution of one millisecond
- Changes to the registers within the PXA25x processor operating system timers were required to support the new features of the PXA27x processor operating system timers:
  - — The RTSR register has 12 new alarm and alarm enable bits.
  - — The following registers have been created:
    - ◆ Two Wristwatch Day Alarm registers
    - ◆ Two Wristwatch Year Alarm registers
    - ◆ Two Stopwatch Alarm registers
    - ◆ A single Periodic Interrupt Alarm register
    - ◆ A single RTC Day Counter register
    - ◆ A single RTC Year Counter register
    - ◆ A single Stopwatch Counter register
    - ◆ A single Periodic Interrupt Counter register

D.22 Operating System Timers

The Operating System timers in the PXA27x processor and PXA25x processor are similar with these exceptions:

- Features added to the PXA27x processor operating system timers:
  - — Eight independent channels, each consisting of:
    - ◆ Counter
    - ◆ Match Register
    - ◆ Control Register
  - — Independent clock for each counter, selectable by software
    - ◆ 32.768-KHz clock for low power
    - ◆ 13 MHz-clock for high accuracy
    - ◆ Externally supplied clock for network synchronization
  - — Counter resolutions of $1/32768^{th}$ of a second, 1 ms, 1 second, and 1 µs
  - — Periodic and one-shot timers
  - — Two external synchronization events
  - — Operation during reduced-power mode (standby, sleep, deep sleep)
- Changes to the registers within the PXA25x processor operating system timers were required to support the new features of the PXA27x processor operating system timers:
  - — OMCR4 - OMCR11, OSMR4 - OSMR11, and OSCR4 - OSCR11 registers have been added to support the extra eight independent channels.
  - — OIER and OSSR registers have additional bitfields added to support the extra eight independent channels.
  - — The only register that remains unchanged is the OWER register.

D.23 Pulse-Width Modulator Controller

The PWM controllers in the PXA27x processor and PXA25x processor are similar with these exceptions:

- Features added to the PWM controller of the PXA27x processor:
  - — There are now four PWMs available (instead of two).
  - — The PWM signals are now based off a 13-MHz clock signal.
- Changes to the registers within the PXA25x processor interrupt controller were required to support the new features of the PXA27x processor PWM controller:
  - — The names of the following registers were changed to allow for greater clarity of function:
    - PWM_CTRLx changed to PWMCRx
    - PWM_DUTYx changed to PWMDCRx
    - PWM_PERVALx changed to PWMPCRx
  - — Extra registers were added to the PXA27x processor to support the extra PWMs. Previously, there were two sets of registers for each PWM in the PXA25x processor (for example, PWM_DUTY0 and PWM_DUTY1). Now, there are four sets of registers in the PXA27x processor (for example, PWMDCR0, PWMDCR1, PWMDCR2, and PWMDCR3).

D.24 General-Purpose I/O Unit

The GPIO controllers in the PXA27x processor and PXA25x processor are similar with these exceptions.

- Features added to the PXA27x processor GPIO controller:
  - There are now 119 GPIOs in discrete packages and 121 GPIOs in Intel® PXA27x Processor Family, whereas in the PXA25x processor, there are 81 GPIOs.
  - Some alternate functions of the PXA27x processor have signals that are bidirectional. When these alternate configurations are selected, the signal becomes bidirectional and the value of GPDR for that GPIO is ignored.
  - The power manager is able to override the configuration of GPIO<10:2> for GPIO reset (when PCFR[GPR_EN] is set).
- Changes to the registers within the PXA25x processor interrupt controller were required to support the new features of the PXA27x processor GPIO controller:
  - Extra registers were added to the PXA27x processor to support the extra GPIOS. Where there were three registers for a function in the PXA25x processor (for example, GPDR0, GPDR1, and GPDR2). There are now four sets of registers in the PXA27x processor (for example, GPDR0, GPDR1, GPDR2, and GPDR3).
  - The mapping of the alternate functions to the GAFR registers has changed from that of the PXA25x processor.

D.25 Interrupt Controller

The interrupt controllers in the PXA27x processor and PXA25x processor are similar with these exceptions.

- Features added to the PXA27x processor interrupt controller:
  - Priority mechanism to indicate highest priority interrupt.
  - Accessibility to interrupt control/status registers using the coprocessor interface.
- Changes to the registers within the PXA25x processor interrupt controller were required to support the new features of the PXA27x processor interrupt controller:
  - All register that existed in the PXA25x processor have additional bits to support the 10 additional interrupt signals in the PXA27x processor controller.
  - IPRx registers have been added to the PXA27x processor interrupt controller to support the new interrupt priority scheme.

Note: The only register that remains unchanged is the ICCR register.

D.26 Debug/Test

This subsection describes separate changes to the software debug module and the hardware TAP controller (also referred to as the Test Interface from the PXA25x processor).

D.26.1 Software Debug Module

There were no changes to the software debug module.

D.26.2 Test Interface

The TAP controllers in the PXA27x processor and PXA25x processor are similar with the exceptions of the following changes.

- Features added to the PXA27x processor TAP controller:
  - — JTAG instructions added to shorten the boundary scan chain while programming flash (flash load, flash program)
- Enhancements to the PXA27x processor TAP controller:
  - — Instruction register increased from 5-bits to 7-bits in length

The part numbers found in the JTAG ID Code registers for the specific processors:

- PXA250 0b1001_0010_0110_0100 [0x9264]
- PXA210 0b1001_0010_0110_1100 [0x926C]
- PXA27x processor 0b1001_0010_0110_0101 [0x9265]

§§

Companion Components for PXA27x Processor

E

E.1 Introduction

This appendix describes the companion components which, at the time of being documented, have been determined to be compatible with Intel® PXA27x Processor Family (PXA27x processor). Because specifications for external components could change, it is the engineer's responsibility to confirm that any components used to meet the specifications in the latest release of *Intel® PXA270 Processor Electrical, Mechanical and Thermal Specification* and *Intel® PXA27x Processor Family Electrical, Mechanical, and Thermal Specification* for the PXA27x processor, whether they are from this list, contained in the Bill of Materials (BOM) for either Intel reference platforms or Intel development platforms.

E.2 System Architecture

N/A

E.3 Clock and Power Manager

This section describes the crystals/oscillators and PMIC devices determined to be compatible with PXA27x processor.

E.3.1 Crystals/Oscillators

See Table E-1 for the list of manufacturers and part numbers for crystal and oscillator devices. All devices are listed in alphabetic order by manufacturer.

Table E-1. Crystal Devices

Item #	Manufacturer	Part Number	Description
1	Fox Electronics	NC26S/NC38	32-KHz crystal
2	Precision Devices	L213000XFCD10BX	13-MHz crystal, LP9000 SMD 3.0mm

E.3.2 PMIC Devices

See Table E-2 for the list of manufacturers of PMIC devices. All manufacturers are listed in alphabetic order.

Disclaimer:This PMIC vendor list is not actively maintained and may does not include every possible PXA27x compatible power IC. It is only intended to provide possible starting points for customer power IC selection. Customers are recommended to research and contact other companies to investigate other possible solutions.

Table E-2. Manufacturers of PMIC Devices

Item #	Manufacturer
1	Advanced Analogic Technology
2	Analog Devices
3	Analog Tech
4	Austria MIcrosystems
5	Dialog Semiconductor
6	Empirion
7	Epson
8	International Rectifier
9	Intersil
10	Linear Technologies
11	Maxim
12	Micrel
13	National Semiconductor
14	ON Semiconductor
15	Panasonic
16	Philips
17	RichTek
18	Rohm
19	SemTech
Board Vendors	
1	Accelent
2	Applied Data Systems
3	Fairchild
4	Sophia Systems
5	Stellcom

E.4 Internal Memory

N/A

E.5 DMA Controller

N/A

E.6 Memory Controller

E.7 LCD Controller

E.8 SSP Serial Port

E.9 I^2C Bus Interface Unit

E.10 UARTs

E.11 Fast Infrared Communication Port

E.12 USB Device (OTG) Controller

This section describes the transceivers needed for USB OTG support determined to be compatible with the PXA27x processor.

E.12.1 USB On-The-Go Transceivers

See Table E-3 for the list of manufacturers and part numbers for USB OTG support. All devices are listed in alphabetic order by manufacturer.

Table E-3. USB OTG Transceivers

Item #	Manufacturer	Part Number	Description
1	Analogic Tech		USB OTG Charge Pump Regulated Fractional Charge Pump Reverse Load Protection Rower Good Flag SRP detection Flag SRP Ready Flag Output Short Circuit and Thermal Protection Under Voltage Protection Less than 1 µA consumed while disabled 100 mA Version Designed to allow operation with output Capacitance as low as 3.3 µF 16-pin 4x4mm QFN package
2	Advanced Analogic Technology	AAT3125	USB Charge Pump
3	Micrel	MIC2550	USB On-The-Go Transceiver
4	Sipex	5301	USB On-The-Go Transceiver
5	Philips	ISP1301	USB On-The-Go Transceiver Integrates OTG analog requirements, dedicated for ASIC solution USB Full speed/low speed transceiver Built-in Charge pump outputs VBUS voltage 4.4 to 5.25 V at > 8 mA (tunable by external cap) Built-in VBUS threshold comparators Supports data-line and VBUS pulsing session request HNP command and status registers Serial I^2C interface minimizes pin count Interrupt on status change VBAT power supply: 3.0 to 3.6 V for USB ports and digital logic VDD_LDG power supply: 1.65 to 3.6V for low power digital I/O interface Built-in ESD protection Available in small (4x4 mm2) HVQFN24 package

E.13 AC ‘97

E.14 I2S (Inter IC Sound Controller)

E.15 MultiMediaCard/SD/SDIO Controller

E.16 Intel(R) Mobile Scalable Link

E.17 Memory Stick Host Controller

E.18 Keypad Interface

E.19 Universal Subscriber ID Interface

E.20 Universal Serial Bus Host Controller

E.21 Real-Time Clock (RTC)

E.22 Operating System Timers

E.23 Pulse-Width Modulator Controller

E.24 General-Purpose I/O Unit

E.25 Interrupt Controller

E.26 JTAG Interface

§§

Glossary

3G. An industry term used to describe the next, still-to-come generation of wireless applications. It represents a move from circuit-switched communications (where a device user has to dial in to a network) to broadband, high-speed, packet-based wireless networks which are always on. The first generation of wireless communications relied on analog technology, followed by digital wireless communications. The third generation expands the digital capabilities by including high-speed connections and increased reliability.

802.11. Wireless specifications developed by the IEEE, outlining the means to manage packet traffic over a network and ensure that packets do not collide, which could result in the loss of data, when travelling from device to device.

8PSK. 8-phase shift key modulation scheme. Used in the EDGE standard.

AC-97. AC-link standard serial interface for modem and audio

ACK. Handshake packet indicating a positive acknowledgment.

Active device. A device that is powered and is not in the suspended state.

Air interface. The RF interface between a mobile cellular handset and the base station

AMPS. Advanced Mobile Phone Service. A term used for analog technologies, the first generation of wireless technologies.

Analog. Radio signals that are converted into a format that allows them to carry data. Cellular phones and other wireless devices use analog in geographic areas with insufficient digital networks.

ARM* V5te. An ARM* architecture designation indicating the processor is conforms to ARM* architecture version 5, including “Thumb” mode and the “El Segundo” DSP extensions.

Asynchronous Data. Data transferred at irregular intervals with relaxed latency requirements.

Asynchronous RA. The incoming data rate, Fsi, and the outgoing data rate, Fso, of the RA process are independent (i.e., there is no shared master clock). See also Rate Adaptation.

Asynchronous SRC. The incoming sample rate, Fsi, and outgoing sample rate, Fso, of the SRC process are independent (i.e., there is no shared master clock). See also Sample Rate Conversion.

Audio device. A device that sources or sinks sampled analog data.

Automatic Scan (AS). Scan signals are set when the AS bit is set.

AWG#. The measurement of a wire’s cross-section, as defined by the American Wire Gauge standard.

Babble. Unexpected bus activity that persists beyond a specified point in a (micro) frame.

Backlight Inverter. A device to drive cold cathode fluorescent lamps used to illuminate LCD panels.

Bandwidth. The amount of data transmitted per unit of time, typically bits per second (bps) or bytes per second (Bps). The size of a network “pipe” or channel for communications in wired networks. In wireless, it refers to the range of available frequencies that carry a signal.

Base Station. The telephone company's interface to the Mobile Station

BGA. Ball Grid Array

BFSK. Binary frequency shift keying. A coding scheme for digital data.

Bit. A unit of information used by digital computers. Represents the smallest piece of addressable memory within a computer. A bit expresses the choice between two possibilities and is typically represented by a logical one (1) or zero (0).

Bit Stuffing. Insertion of a "0" bit into a data stream to cause an electrical transition on the data wires, allowing a PLL to remain locked.

Blackberry. A two-way wireless device (pager) made by Research In Motion (RIM) that allows users to check e-mail and voice mail translated into text, as well as page other users of a wireless network service. It has a miniature "qwerty" keyboard that can be used by your thumbs, and uses SMS protocol. A Blackberry user must subscribe to the proprietary wireless service that allows for data transmission.

Bluetooth. A short-range wireless specification that allows for radio connections between devices within a 30-foot range of each other. The name comes from 10th-century Danish King Harald Blatand (Bluetooth), who unified Denmark and Norway.

BPSK. Binary Phase Shift Keying. A means of encoding digital data into a signal using phase-modulated communications.

bps. Transmission rate expressed in bits per second.

Bps. Transmission rate expressed in bytes per second.

BTB. Branch Target Buffer

BTS. Base Transmitter Station

Buffer. Storage used to compensate for a difference in data rates or time of occurrence of events, when transmitting data from one device to another.

Bulk Transfer. One of the four USB transfer types. Bulk transfers are non-periodic, large bursty communication typically used for a transfer that can use any available bandwidth and can also be delayed until bandwidth is available. See also Transfer Type.

Bus Enumeration. Detecting and identifying USB devices.

Byte. A data element that is eight bits in size.

Capabilities. Those attributes of a USB device that are administrated by the host.

CAS. Cycle Accurate Simulator

CAS-B4-RAS. See CBR.

CBR. CAS Before RAS. Column Address Strobe Before Row Address Strobe. A fast refresh technique in which the DRAM keeps track of the next row it needs to refresh, thus simplifying what a system would have to do to refresh the part.

CDMA (Code Division Multiple Access). U.S. wireless carriers Sprint PCD and Verizon use CDMA to allocate bandwidth for users of digital wireless devices. CDMA distinguishes between multiple transmissions carried simultaneously on a single wireless signal. It carries the transmissions on that signal, freeing network room for the

wireless carrier and providing interference-free calls for the user. Several versions of the standard are still under development. CDMA should increase network capacity for wireless carriers and improve the quality of wireless messaging. CDMA is an alternative to GSM.

CDPD. Cellular Digital Packet Data Telecommunications companies can use DCPD to transfer data on unused cellular networks to other users. IF one section, or “cell” of the network is overtaxed, DCPD automatically allows for the reallocation of services.

Cellular. Technology that senses analog or digital transmissions from transmitters that have areas of coverage called cells. As a user of a cellular phone moves between transmitters from one cell to another, the users’ call travels from transmitter to transmitter uninterrupted.

Circuit Switched. Used by wireless carriers, this method lets a user connect to a network or the Internet by dialing in, such as with a traditional phone line. Circuit switched connections are typically slower and less reliable than packet-switched networks, but are currently the primary method of network access for wireless users in the U.S.

CF. Compact Flash memory and I/O card interface

Characteristics. Those qualities of a USB device that are unchangeable; for example, the device class is a device characteristic.

Client. Software resident on the host that interacts with the USB System Software to arrange data transfer between a function and the host. The client is often the data provider and consumer for transferred data.

CML. Current Mode Logic

CODEC. Coder/Decoder transforms analog data into a digital bit stream (coder) and digital signals into analog data (decoder). All digital audio, modem, microphone input (MIC-in), CODEC register control, and status information are communicated over the AC-link.

Configuring Software. Software resident on the host software that is responsible for configuring a USB device. This may be a system configuration or software specific to the device.

Control Endpoint. A pair of device endpoints with the same endpoint number that are used by a control pipe. Control endpoints transfer data in both directions and, therefore, use both endpoint directions of a device address and endpoint number combination. Thus, each control endpoint consumes two endpoint addresses.

Control Pipe. Same as a message pipe.

Control Transfer. One of the four USB transfer types. Control transfers support configuration/command/status type communications between client and function. See also Transfer Type.

CRC. See Cyclic Redundancy Check.

CSP. Chip Scale Package

CTE. Coefficient of Thermal Expansion

CTI. Computer Telephony Integration

Clear-To-Send (CTS). When low, indicates the modem or data set is ready to exchange data.

Cyclic Redundancy Check (CRC). A check performed on data to check if an error has occurred in transmitting, reading, or writing the data. The result of a CRC is typically stored or transmitted with the checked data. The stored or transmitted result is compared to a CRC calculated for the data to determine if an error has occurred.

D-cache. Data cache

DECT. The Digital European Cordless Telecommunications standard

Default Address. An address defined by the USB Specification and used by a USB device when it is first powered or reset. The default address is 00H.

Default Pipe. The message pipe created by the USB System Software to pass control and status information between the host and a USB device's endpoint zero.

Device. A logical or physical entity that performs a function. The actual entity described depends on the context of the reference. At the lowest level, "device" may refer to a single hardware component, as in a memory device. At a higher level, it may refer to a collection of hardware components that perform a particular function, such as a USB interface device. At an even higher level, device may refer to the function performed by an entity attached to the USB; for example, a data/FAX modem device. Devices may be physical, electrical, addressable, and logical. When used as a non-specific reference, a USB device is either a hub or a function.

Device Address. A seven-bit value representing the address of a device on the USB. The device address is the default address (00H) when the USB device is first powered or the device is reset. Devices are assigned a unique device address by the USB System Software.

Device Endpoint. A uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the host and device. See also Endpoint Address.

Device Resources. Resources provided by USB devices, such as buffer space and endpoints. See also Host Resources and Universal Serial Bus Resources.

Device Software. Software that is responsible for using a USB device. This software may or may not also be responsible for configuring the device for use.

DMA. Direct Memory Access

Downstream. The direction of data flow from the host or away from the host. A downstream port is the port on a hub electrically farthest from the host that generates downstream data traffic from the hub. Downstream ports receive upstream data traffic.

DQPSK. Differential Quadrature Phase Shift Keying a modulation technique used in TDMA.

Driver. When referring to hardware, an I/O pad that drives an external load. When referring to software, a program responsible for interfacing to a hardware device, that is, a device driver.

DSP. Digital Signal Processing

Data Set Ready (DSR). When low, indicates the modem or data set is ready to send signal and establish a communications link with a UART.

DSTN (Double-layer Supertwist Nematic). A passive LCD panel that uses two display layers to counteract the color shifting that occurs with conventional supertwist displays. See STN.

Data Terminal Ready (DTR). When low, signals the modem or data set that the UART is ready to communicate.

Dual band mobile phone. A phone that supports both analog and digital technologies by picking up analog signals when digital signals fade. Most mobile phones are not dual-band.

Digital Volt Meter. This equipment is used for measuring voltage across a series resistor, preferable for high current power supplies. For low power supplies, the voltage drop is too small for detection by the DVM.

DWORD. Double Word. A data element that is two words (that is, four bytes or 32 bits) in size.

Dynamic Insertion and Removal. The ability to attach and remove devices while the host is in operation.

E2PROM. See Electrically Erasable Programmable Read Only Memory.

EAS. External Architecture Specification

EAV. End of Active Video

EDGE. Enhanced Data GSM Environment. A faster version of the GSM standard. It is faster because it can carry messages using broadband networks that employ more bandwidth than standard GSM networks.

EEPROM. See Electrically Erasable Programmable Read Only Memory.

Electrically Erasable Programmable Read Only Memory (EEPROM). Non-volatile re-writable memory storage technology.

End User. The user of a host.

Endpoint. See Device Endpoint.

Endpoint Address. The combination of an endpoint number and an endpoint direction on a USB device. Each endpoint address supports data transfer in one direction.

Endpoint Direction. The direction of data transfer on the USB. The direction can be either IN or OUT. IN refers to transfers to the host; OUT refers to transfers from the host.

Endpoint Number. A four-bit value between 0H and FH, inclusive, associated with an endpoint on a USB device.

Envelope Detector. An electronic circuit inside a USB device that monitors the USB data lines and detects certain voltage related signal characteristics.

EOF. End-of-(micro) Frame

EOL. End-of-Line. A special character or sequence of characters that marks the end of a line, just before the new line character.

EOP. End-of-Packet

EOTD. Enhanced Observed Time Difference

ETM. Embedded Trace Macrocell. The ARM* real-time trace capability

External Port. See Port.

Eye pattern. A representation of USB signaling that provides minimum and maximum voltage levels as well as signal jitter.

FAR. Fault Address Register. Part of the ARM* architecture.

False EOP. A spurious, usually noise-induced event that is interpreted by a packet receiver as an EOP.

FDD. The Mobile Station transmits on one frequency; the Base Station transmits on another frequency

FDM. Frequency Division Multiplexing. Each Mobile station transmits on a different frequency (within a cell).

FDMA. Frequency Division Multiple Access. An analog standard that lets multiple users access a group of radio frequency bands and eliminates interference of message traffic.

FHSS. See Frequency Hopping Spread Spectrum.

FIFO. First In/First Out is a buffering scheme in which the first byte of data that enters the buffer is also the first byte of data retrieved by the serial port (controlled by UART chip)

FIQ. Fast Interrupt Request. See Interrupt Request.

FIR. Fast Infrared

FICP. Fast Infrared Communications Port is a low-voltage controller that can be connected directly to the device's transmit, receive, and shutdown logic signals of the transceiver**.** The controller operates at half-duplex and is based on 4-Mbps and four-position pulse modulation, providing a very fast data rate.

Flash Memory. Flash is a type of erasable, rewritable memory that holds its content when power is off. The low-cost, low-power and high-density memory chip, with high-speed architecture, is highly reliable.

Frame. A 1 millisecond time base established on full-/low-speed buses.

Frame Pattern. A sequence of frames that exhibit a repeating pattern in the number of samples transmitted per frame. For a 44.1 KHz audio transfer, the frame pattern could be nine frames containing 44 samples followed by one frame containing 45 samples.

Frequency Hopping Spread Spectrum. A method by which a carrier spreads out packets of information (voice or data) over different frequencies. For example, a phone call is carried on several different frequencies so that when one frequency is lost another picks up the call without breaking the connection.

Fs. See Sample Rate.

FS-CSP. Folded Stacked-Chip Scale Package. Method for escape routing for copper-defined land pads that involves routing signals on the four inner rows of balls in the chip scale package from the top layer to the inner PCB layers for routing way from the package.

FSR. Fault Status Register. Part of the ARM* architecture.

Full-duplex. Computer data transmission occurring in both directions simultaneously.

Full-speed. USB operation at 12 Mbps. See also Low-speed and High-speed.

Function. A USB device that provides a capability to the host, such as an ISDN connection, a digital microphone, or speakers.

GMSK. Gaussian Minimum Shift Keying. A modulation scheme used in GSM.

GPRS. General Packet Radio Service. A technology that sends packets of data across a wireless network at speeds up to 114 Kbps. Unlike circuit-switched networks, wireless users do not have to dial in to networks to download information; GPRS wireless devices are "always on" in that they can send and receive data without dial-ins. GPRS works with GSM.

GPS. Global Positioning System allows location of an object anywhere on earth using a constellation of satellites orbiting the earth.

GPIO. General Purpose Inputs/Outputs

GSM. Global System for Mobile Communications. A standard for how data is coded and transferred through the wireless spectrum. The European wireless standard, also used in parts of Asia, GSM is an alternative to CDMA. GSM digitizes and compresses data and sends it across a channel with two other streams of user data. GSM is based on TDMA technology.

Hamming Distance. The distance (number of bits) between encoded values that can change without causing a decode into the wrong value.

Handshake Packet. A packet that acknowledges or rejects a specific condition. For examples, see ACK and NAK.

HDML. Handheld Device Markup Language. HDML uses hypertext transfer protocol (HTTP) to display text versions of web pages on wireless devices. Unlike WML, HDML is not based on XML. HDML does not allow scripts, while WML uses a variant of JavaScript. Web site developers using HDML must re-code their web pages in HDML to be viewed on the smaller screen sizes of handheld devices.

HARP. Windows CE standard development platform spec (Hardware Adaptation Reference Platform)

HD-CSP. High Density Chip Scale Package. Method for escape routing for copper-defined land pads that involves routing signals on the two inner rows of balls in the chip scale package from the top layer to the inner PCB layers for routing way from the package.

High Density Interconnect (HDI). PCB fabrication technology required for designing of escape routing, which is the layout of package signals from underneath the chip scale package to other components on the PCB. The design involves routing signals from the inner rows of balls on the ball pitch packages away from the packages.

High-bandwidth Endpoint. A high-speed device endpoint that transfers more than 1024 bytes and less than 3073 bytes per microframe.

High-speed. USB operation at 480 Mbps. See also Low-speed and Full-speed.

Host. The host computer system where the USB Host controller is installed. This includes the host hardware platform (CPU, bus, and so forth) and the operating system in use.

Host Controller. The host's USB interface. See UHC.

Host Controller Driver (HCD). The USB software layer that abstracts the Host controller hardware. The Host controller driver provides an SPI for interaction with a Host controller. The Host controller driver hides the specifics of the Host controller hardware implementation.

Host Resources. Resources provided by the host, such as buffer space and interrupts. See also Device Resources and Universal Serial Bus Resources.

HSTL. High-Speed Transceiver Logic

Hub. A USB device that provides additional connections to the USB.

Hub Tier. One plus the number of USB links in a communication path between the host and a function.

IMMU. Instruction Memory Management Unit, part of the Intel® XScale™ core.

I-Mode. A Japanese wireless service for transferring packet-based data to handheld devices created by NTT DoCoMo. I-Mode is based on a compact version of HTML and does not currently use WAP.

I-cache. Instruction cache

IBIS. I/O Buffer Information. Specification is a behavioral description of the I/O buffers and package characteristics of a semiconductor device. IBIS models use a standard format to make it easier to import data into circuit simulation software packages.

iDEN. Integrated Digital Enhanced Network. A technology that allows users to access phone calls, two-way radio transmissions, paging and data transmissions from one wireless device. iDEN was developed by Motorola and based on TDMA.

Interrupt Request (IRQ). A hardware signal that allows a device to request attention from a host. The host typically invokes an interrupt service routine to handle the condition that caused the request.

Interrupt Transfer. One of the four USB transfer types. Interrupt transfer characteristics are small data, non-periodic, low-frequency, and bounded-latency. Interrupt transfers are typically used to handle service needs. See also Transfer Type.

I/O Request Packet. An identifiable request by a software client to move data between itself (on the host) and an endpoint of a device in an appropriate direction.

IrDA. Infrared Development Association

IRP. See I/O Request Packet.

IRQ. See Interrupt Request.

ISI. Inter-signal interference. Data ghosting caused when multi-path delay causes previous symbols to interfere with the one currently being processed.

ISM. Industrial, Scientific, and Medical band. Part of the wireless spectrum that is less regulated, such as 802.11.

Isochronous Data. A stream of data whose timing is implied by its delivery rate.

Isochronous Device. An entity with isochronous endpoints, as defined in the USB Specification, that sources or sinks sampled analog streams or synchronous data streams.

Isochronous Sink Endpoint. An endpoint that is capable of consuming an isochronous data stream that is sent by the host.

Isochronous Source Endpoint. An endpoint that is capable of producing an isochronous data stream and sending it to the host.

Isochronous Transfer. One of the four USB transfer types. Isochronous transfers are used when working with isochronous data. Isochronous transfers provide periodic, continuous communication between host and device. See also Transfer Type.

Jitter. A tendency toward lack of synchronization caused by mechanical or electrical changes. More specifically, the phase shift of digital pulses over a transmission medium.

kbps. Transmission rate expressed in kilobits per second. A measurement of bandwidth in the U.S.

kBps. Transmission rate expressed in kilobytes per second.

KPC. Keypad Interface Control

LCD. Liquid Crystal Display

Little Endian. Method of storing data that places the least significant byte of multiple-byte values at lower storage addresses. For example, a 16-bit integer stored in little endian format places the least significant byte at the lower address and the most significant byte at the next address.

LOA. Loss of bus activity characterized by an SOP without a corresponding EOP.

Low-speed. USB operation at 1.5 Mbps. See also Full-speed and High-speed.

LSb. Least significant bit

LSB. Least significant Byte

LVDS. Low-voltage Differential Signal

MAC. Multiply Accumulate Unit

Mbps. Transmission rate expressed in megabits per second.

MBps. Transmission rate expressed in megabytes per second.

MC. Media Center. A combination digital set-top box, video and music jukebox, personal video recorder and an Internet gateway and firewall that hooks up to a broadband connection.

MCI. Memory Controller Interface

Message Pipe. A bidirectional pipe that transfers data using a request/data/status paradigm. The data has an imposed structure that allows requests to be reliably identified and communicated.

Microframe. A 125 microsecond time base established on high-speed buses.

MMC. Multimedia Card. Small form factor memory and I/O card

MMX Technology. The Intel® MMX™ technology comprises a set of instructions that are designed to greatly enhance the performance of advanced media and communications applications. Refer to Chapter 10 of the Intel Architecture Software Developers Manual, Volume 3: System Programming Guide, Order #245472.

Mobile Station. Cellular Telephone handset

M-PSK. Multilevel Phase Shift Keying. A convention for encoding digital data in which there are multiple states.

MMU. Memory Management Unit. Part of the Intel XScale®core.

MSb. Most significant bit

MSB. Most significant byte

MSL. Mobile Scalable Link

NAK. Handshake packet indicating a negative acknowledgment.

Non Return to Zero Invert (NRZI). A method of encoding serial data in which ones and zeroes are represented by opposite and alternating high and low voltages where there is no return to zero (reference) voltage between encoded bits. Eliminates the need for clock pulses.

NRZI. See Non Return to Zero Invert.

Object. Host software or data structure representing a USB entity.

OFDM. See Orthogonal Frequency Division Multiplexing.

Orthogonal Frequency Division Multiplexing. A special form of multi-carrier modulation. In a multi-path channel, most conventional modulation techniques are sensitive to inter-symbol interference unless the channel symbol rate is small compared to the delay spread of the channel. OFDM is significantly less sensitive to inter-symbol interference, because a special set of signals is used to build the composite transmitted signal. The basic idea is that each bit occupies a frequency-time window that ensures little or no distortion of the waveform. In practice, it means that bits are transmitted in parallel over a number of frequency-nonselective channels.

OTG. See USB OTG.

Packet. A bundle of data organized in a group for transmission. Packets typically contain three elements: control information (for example, source, destination, and length), the data to be transferred, and error detection and correction bits. Packet data is the basis for packet-switched networks, which eliminate the need to dial-in to send or receive information, because they are “always on.”

Packet Buffer. The logical buffer used by a USB device for sending or receiving a single packet. This determines the maximum packet size the device can send or receive.

Packet ID (PID). A field in a USB packet that indicates the type of packet, and by inference, the format of the packet and the type of error detection applied to the packet.

Packet Switched Network. Networks that transfer packets of data.

PCMCIA. Personal Computer Memory Card Interface Association (PC Card)

PCD. Pixel Clock Divider

PCS. Personal Communications Services. An alternative to cellular, PCD works like cellular technology because it sends calls from transmitter to transmitter as a caller moves. But PCS uses its own network, not a cellular network, and offers fewer “blind spots” than cellular, where calls are not available. PCS transmitters are generally closer together than their cellular counterparts.

PDA. Personal Digital Assistant. A mobile handheld device that gives users access to text-based information. Users can synchronize their PDAs with a PC or network; some models support wireless communication to retrieve and send e-mail and get information from the Internet.

Phase. A token, data, or handshake packet. A transaction has three phases.

Phase Locked Loop (PLL). A circuit that acts as a phase detector to keep an oscillator in phase with an incoming frequency.

Physical Device. A device that has a physical implementation; for example, speakers, microphones, and CD players.

PID. See Packet ID or Process ID.

PIO. Programmed Input/Output

Pipe. A logical abstraction representing the association between an endpoint on a device and software on the host. A pipe has several attributes; for example, a pipe may transfer data as streams (stream pipe) or messages (message pipe). See also Stream Pipe and Message Pipe.

PLL. See Phase Locked Loop.

PM. Phase Modulation

PMIC. Power Management Integrated Circuit is a highly integrated device with both required and optional features to support the nine power domains on the PXA27x processor, as well as dynamic voltage management features.

Polling. Asking multiple devices, one at a time, if they have any data to transmit.

POR. See Power On Reset

Port. Point of access to or from a system or circuit. For the USB, the point where a USB device is attached.

Power On Reset (POR). Restoring a storage device, register, or memory to a predetermined state when power is applied.

Process ID. Process Identifier.

Programmable Data Rate. Either a fixed data rate (single-frequency endpoints), a limited number of data rates (32 KHz, 44.1 KHz, 48 KHz, …), or a continuously programmable data rate. The exact programming capabilities of an endpoint must be reported in the appropriate class-specific endpoint descriptors.

Protocol. A specific set of rules, procedures, or conventions relating to format and timing of data transmission between two devices.

PSP. Programmable Serial Protocol

PWM. Pulse-Width Modulator

QBS. Qualification By Similarity. A technique allowed by JEDEC for part qualification when target parameters are fully understood and data exist to warrant omitting a specific test.

QAM. Quadrature Amplitude Modulation. A coding scheme for digital data.

QPSK. Quadrature Phase Shift Keying. A convention for encoding digital data into a signal using phase-modulated communications.

RA. See Rate Adaptation.

Radio Frequency Device. These devices use radio frequencies to transmit data. One typical use is for bar code scanning of products in a warehouse or distribution center, and sending that information to an ERP database.

Rate Adaptation. The process by which an incoming data stream, sampled at Fs i, is converted to an outgoing data stream, sampled at Fs o, with a certain loss of quality, determined by the rate adaptation algorithm. Error control mechanisms are required for the process. Fs i and Fs o can be different and asynchronous. Fs i is the input data rate of the RA; Fs o is the output data rate of the RA.

Request. A request made to a USB device contained within the data portion of a SETUP packet.

Retire. The action of completing service for a transfer and notifying the appropriate software client of the completion.

RGBT. Red, Green, Blue, Transparency

ROM. Read Only Memory

Root Hub. A USB hub directly attached to the Host controller. This hub (tier 1) is attached to the host.

Root Port. The downstream port on a Root Hub.

RTC. Real-Time Clock

RTS. Request-To-Send. When low, signals the modem or data set that the UART is ready to exchange data.

SA-1110. StrongARM* based applications processor for handheld products

Intel® StrongARM* SA-1111. Companion chip for the Intel® SA-1110 processor

SAD. Sum of absolute differences

Sample. The smallest unit of data on which an endpoint operates; a property of an endpoint.

Sample Rate (Fs). The number of samples per second, expressed in Hertz (Hz).

Sample Rate Conversion (SRC). A dedicated implementation of the RA process for use on sampled analog data streams. The error control mechanism is replaced by interpolating techniques. Service A procedure provided by a System Programming Interface (SPI).

Satellite Phone. Phones that connect callers by satellite. Users have a world-wide alternative to terrestrial connections. Typical use is for isolated users, such as crews of deep-sea oil rigs with phones configured to connect to a satellite service.

SAV. Start of Active Video

SAW. Surface Acoustic Wave filter.

SD. Secure Digital. A non-volatile and very small memory card with high-storage capacity.

SDIO. See Secure Digital I/O

SDRAM. Synchronous Dynamic Random Access Memory

Secure Digital I/O (SDIO). Protocol provides high-speed data I/O with low-power consumption. The card supports multiple I/O functions, interrupts, and read/write operations.

SEO. Single-Ended Zero. The USB uses differential signals (D+ and D-) to transmit and receive data. The SEO can be set to control the multiplexors.

Service Interval. The period between consecutive requests to a USB endpoint to send or receive data.

Service Jitter. The deviation of service delivery from its scheduled delivery time.

Service Rate. The number of services to a given endpoint per unit time.

SIMD. Single Instruction Multiple Data. A parallel processing architecture.

Smart Phone. A combination of a mobile phone and a PDA, which allow users to communicate as well as perform tasks; such as, accessing the Internet and storing contacts in a database. Smart phones have a PDA-like screen.

SMROM. Synchronous Mask ROM

SMS. Short Messaging Service. A service through which users can send text-based messages from one device to another. The message can be up to 160 characters and appears on the screen of the receiving device. SMS works with GSM networks.

SOC. System-On-Chip

SOF. See Start-of-Frame.

SOP. Start-of-Packet

SPI. See System Programming Interface. Also, see Serial Peripheral Interface protocol.

SPI. Serial Peripheral Interface

Split transaction. A transaction type supported by host controllers and hubs. This transaction type allows full- and low-speed devices to be attached to hubs operating at high-speed.

Spread Spectrum. An encoding technique patented by actress Hedy Lamarr and composer George Antheil, which broadcasts a signal over a range of frequencies.

SRAM. Static Random Access Memory

SRC. See Sample Rate Conversion.

SSE. Streaming SIMD Extensions

SSE2. Streaming SIMD Extensions 2: for Intel Architecture machines, 144 new instructions, a 128-bit SIMD integer arithmetic and 128-bit SIMD double precision floating point instructions, enabling enhanced multimedia experiences.

SRP. Session Request Protocol

SSP. Synchronous Serial Port

SSTL. Stub Series Terminated Logic

Stage. One part of the sequence composing a control transfer; stages include the Setup stage, the Data stage, and the Status stage.

Start-of-Frame (SOF). The first transaction in each (micro)frame. An SOF allows endpoints to identify the start of the (micro)frame and synchronize internal endpoint clocks to the host.

STN. Supertwist Nematic. A passive LCD display using the technique of twisting light rays to improve the quality of the LCD screens. Passive LCD displays apply either full-on voltage or full-off voltage (on or off) during each frame refresh. By intelligently turning the pixel on and off each frame a partial intensity is affected on each pixel. This process is known as “dithering.”

Stream Pipe. A pipe that transfers data as a stream of samples with no defined USB structure

SWI. Software Interrupt

Synchronization Type. A classification that characterizes an isochronous endpoint’s capability to connect to other isochronous endpoints.

Synchronous RA. The incoming data rate, Fsi, and the outgoing data rate, Fso, of the RA process are derived from the same master clock. There is a fixed relation between Fsi and Fso.

Synchronous SRC. The incoming sample rate, Fsi, and outgoing sample rate, Fso, of the SRC process are derived from the same master clock. There is a fixed relation between Fsi and Fso.

System Programming Interface (SPI). A defined interface to services provided by system software.

TAP. Test Access Port. The boundary-scan interface provides a means of driving and sampling the external pins of the processor: testing the processor's electrical connections to the circuit board and the integrity of the circuit board connections between devices. The interface is controlled through five dedicated TAP pins.

TC. Temperature Cycling

TCK. Test Clock

TDD. Time Division Duplexing. The Mobile Station and the Base Station transmit on same frequency at different times.

TDM. See Time Division Multiplexing.

TDMA. Time Division Multiple Access

TDMA. Protocol allows multiple users to access a single radio frequency by allocating time slots for use to multiple voice or data calls. TDMA breaks down data transmissions, such as a phone conversation, into fragments and transmits each fragment in a short burst, assigning each fragment a time slot. With a cell phone, the caller would not detect this fragmentation. TDMA works with GSM and digital cellular services.

TDR. See Time Domain Reflectometer.

Termination. Passive components attached at the end of cables to prevent signals from being reflected or echoed.

Three-state. A high-impedance state in which the output is floating and is electrically isolated from the buffer's circuitry.

Time Division Multiplexing (TDM). A method of transmitting multiple signals (data, voice, and/or video) simultaneously over one communications medium by interleaving a piece of each signal one after another.

Time Domain Reflectometer (TDR). An instrument capable of measuring impedance characteristics of the USB signal lines.

TFT. Thin Film Transistor. An active LCD panel in which each pixel is controlled by individual transistors that allows the voltage applied to each pixel to be precisely controlled. This permits faster response time and greater contrast compared to passive LCD panels.

Time-out. The detection of a lack of bus activity for some predetermined interval.

Token Packet. A type of packet that identifies what transaction is to be performed on the bus.

TPV. Third Party Vendor

Transaction. The delivery of service to an endpoint; consists of a token packet, optional data packet, and optional handshake packet. Specific packets are allowed/required based on the transaction type.

Transaction translator. A functional component of a USB hub. The Transaction Translator responds to special high-speed transactions and translates them to full/low-speed transactions with full/low-speed devices attached on downstream facing ports.

Transfer. One or more bus transactions to move information between a software client and its function.

Transfer Type. Determines the characteristics of the data flow between a software client and its function. Four standard transfer types are defined: control, interrupt, bulk, and isochronous.

TS. Thermal Shock

Turn-around Time. The time a device needs to wait to begin transmitting a packet after a packet has been received to prevent collisions on the USB. This time is based on the length and propagation delay characteristics of the cable and the location of the transmitting device in relation to other devices on the USB.

UART. Universal Asynchronous Receiver/Transmitter serial port

UICC. Universal Integrated Circuit Card, formerly SIM card. A small electronic device about the size of a credit card that contains an embedded 8-bit microprocessor. The card stores a mathematical algorithm that encrypts voice and data transmissions. The card also identifies the caller to the mobile network as being a legitimate caller.

UDC. Universal Serial Bus Device Controller. The UDC, which consists of peripheral bus interface, endpoint memory, endpoint control, and USB interface, provides interface options to a host of peripheral devices at full speed.

UHC. Universal Serial Bus Host Controller. The UHC in conjunction with the UHC driver serially transfers data between a shared-memory data structure and the USB controller.

Universal Serial Bus Driver (USBD). The host resident software entity responsible for providing common services to clients that are manipulating one or more functions on one or more Host controllers.

Universal Serial Bus Resources. Resources provided by the USB, such as bandwidth and power. See also Device Resources and Host Resources.

Upstream. The direction of data flow towards the host. An upstream port is the port on a device electrically closest to the host that generates upstream data traffic from the hub. Upstream ports receive downstream data traffic.

USBD. See Universal Serial Bus Driver.

USB-IF. USB Implementers Forum, Inc. is a nonprofit corporation formed to facilitate the development of USB compliant products and promote the technology.

USB OTG. The USB On-The-Go provides 'dual-role peripheral' capability: it can act as either host or peripheral depending on how users connect the cable to its unique mini-AB receptacle. When the dual-role device is connected to the mini-A plug, it turns into a host. When the mini-B plug is connected instead, the device becomes a peripheral.

Universal Subscriber Identity Module (USIM). The USIM controller is an interface for a GSM mobile handset that supports communication with SmartCards.

VBI. Vertical Blanking Interval. Also known as the “backporch”.

Virtual Device. A device that is represented by a software interface layer. An example of a virtual device is a hard disk with its associated device driver and client software that makes it able to reproduce an audio.WAV file.

VLIO. Variable Latency Input/Output Interface

YUV. A method of characterizing video signals typically used in digital cameras and PAL television specifying luminance and chrominance.

WAP. Wireless Application Protocol. WAP is a set of protocols that lets users of mobile phones and other digital wireless devices access Internet content, check voice mail and e-mail, receive text of faxes and conduct transactions. WAP works with multiple standards, including CDMA and GSM. Not all mobile devices support WAP.

W-CDMA. Wideband CDMA. A third generation wireless technology under development that allows for high-speed, high-quality data transmission. Derived from CDMA, W-CDMA digitizes and transmits wireless data over a broad range of frequencies. It requires more bandwidth than CDMA, but offers faster transmission because it optimizes the use of multiple wireless signals, instead of one, as does CDMA.

Wireless LAN. A wireless LAN uses radio frequency technology to transmit network messages through the air for relatively short distances, like across an office building or a college campus. A wireless LAN can serve as a replacement for, or an extension to, a traditional wired LAN.

Wireless Spectrum. A band of frequencies where wireless signals travel carrying voice and data information.

Word. A data element that is four bytes (32 bits) in size.

WML. Wireless Markup Language. A version of HDML based on XML. Wireless applications developers use WML to re-target content for wireless devices.

§§

Index

A

AC ‘97
II:D-9, II:E-5
AC ’97 II:13-1
Achieve Minimum Power
Usage During All Power Modes I:5-2
Usage During Deep Sleep I:5-2
Achieve Minimum Power Usage
During Idle/13M/Run/Turbo I:5-3
During Sleep I:5-3
During Standby I:5-3
Active Color
12-bit per pixel Mode II:7-16
16-bit per pixel Mode II:7-18
18-bit per pixel Mode II:7-20
Alternate Bus Master
Block Diagram II:6-33
Interface II:6-30
Layout Notes II:6-33
Signals II:6-32
Alternate Function During Standby and Sleep Mode II:18-4

B

Backlight Inverter II:7-4
Baseband Interface II:16-1
Baseband/Multimedia Interface
II:D-10
Block Diagram
II:3-4, II:4-1, II:5-2, II:5-4, II:5-5, II:6-5, II:8-3, II:11-2, II:12-2, II:13-2, II:14-3, II:14-5, II:14-6, II:18-3, II:18-7, II:18-9, II:18-11, II:19-4, II:21-2, II:22-2, II:23-2, II:25-3, II:27-3
Block Diagram/Schematic II:24-3
Block Diagram for
USB Host Differential Connection (Port 1 or Port 2) II:20-2
USB Host Single-Ended Connection (Port 3) II:20-4
Block Diagrams II:20-2
Block Diagrams for
USB Host Port 2 (Differential or Single-Ended) II:20-3
Bluetooth UART II:10-5
Block Diagram II:10-5
Signals II:10-5
Boundary-Scan Register II:26-8
Bus Host Controller, Universal Serial II:D-10
Bus-Powered Device II:12-5
Bypass Register II:26-7

C

Capture-DR State II:26-11
Capture-IR State II:26-13
Cautions I:4-4
Channel Access/Control Block II:22-2
Channels 4 - 11 Blocks II:22-3
Clock and Power Manager
II:D-1, II:E-1
Clock Interface II:3-6
Clock Interface Signals II:3-1
Clock Manager
II:D-1
Clocks and Power Interface II:3-1
CODEC, BITCLK Signal II:14-6
Companion Components, PXA27x Processor II:E-1
Contrast Voltage II:7-3
Controller, USB II:D-10
Crystals/Oscillators
II:E-1

D

Data-Specific Registers II:26-9
Debug Registers II:26-16
Debug/Test
II:D-15
Design Check List I:3-1
Differences
PXA27x and PXA25x Processors II:D-1
Digital-to-Analog Converter II:9-2
DMA Controller
II:D-3, II:E-3
DMA Controller Interface II:5-1
Document Organization and Overview I:1-1

E

Example Power Supply Utilizing Minimal Regulators I:4-2
Exit1
DR State II:26-12
IR State II:26-13

Exit2

DR State II:26-12
IR State II:26-14
External 13.00-MHz Clock II:3-8
External 32.768-KHz Clock II:3-7
External Clock Enable Configuration Scheme II:8-5
External Clock Source Configuration Scheme II:8-4

F

Fast Infrared Communication Port II:D-8, II:E-3
Fast Infrared Interface II:11-1
Feature List II:27-2
Features, JTAG II:26-2
FFUART
Block Diagram II:10-4
Layout Notes II:10-4
Flash
Block Diagram II:6-16
Data Register II:26-9
Layout Note II:6-16
Memory Interface II:6-15
Memory Signals II:6-15
Flow-Through DMA Transfers II:5-5
Fly-By DMA Transfers II:5-3
FS-CSP Escape Routing I:2-8
Full Function UART II:10-3
Full Function UART Signals II:10-3
Functional Overview, PXA27x I:1-2

G

General PCB Characteristics I:2-1
General Purpose Input/Output Interfaces II:24-1
General-Purpose I/O Unit II:D-14, II:E-5

H

Handling
Shipping Media I:2-10
HD-CSP Escape Routing I:2-7
Host Controller, Universal Serial Bus II:D-10

I

I2C Bus Interface Unit II:D-7, II:E-3
I2S (Inter IC Sound Controller) II:D-9, II:E-5
I2S Interface II:14-1
ID Interfacev, Universal Subscriber II:E-5
Instructions
TRISTATE II:26-6
Intel XScale® Data Registers II:26-9
Intel(R) Mobile Scalable Link II:E-5
Intel® Flash Memory Design Guidelines I:2-1
Intel® Quick Capture Technology II:27-1
Interface to
External Charge Pump Device (OTG) II:12-9
External Transceiver (OTG) II:12-8
External USB Transceiver (non-OTG) II:12-12
Interface, Universal Subscriber ID II:E-5
Interfacing to
Matrix Keypad II:18-5
Inter-Integrated Circuit II:9-1
Internal Clock Enable Configuration Design II:8-5
Internal Memory II:D-2, II:E-2
Internal SRAM II:4-1
Interrupt Controller II:D-14, II:E-5
Interrupt Interface II:25-1
Introduction II:D-1, II:E-1
Introduction to Part I I:1-1
Introduction to Part II II:1-1

J

JTAG Debug II:26-1
JTAG Device Identification Register II:26-15
JTAG Instruction Register and Instruction Set II:26-5
JTAG Interface II:E-5
JTAG Test Data Registers II:26-16

K

Keypad Interface II:18-1, II:D-10, II:E-5
Keypad Matrix and Direct Keys
No Rotary Encoder II:18-6
One Rotary Encoder II:18-8
Two Rotary Encoders II:18-10
Keypad Signals to Wake-up from Standby and Sleep Mode II:18-4

L

Layout Notes II:3-6, II:4-2, II:5-2, II:5-4, II:5-6, II:7-3, II:7-7, II:7-9, II:7-11, II:7-13, II:7-15, II:7-17, II:7-19, II:7-21, II:7-23, II:8-6, II:9-4, II:12-2, II:13-3, II:14-3, II:15-2, II:17-2, II:18-4, II:19-5, II:20-

5, II:21-2, II:22-3, II:23-2, II:24-3, II:25-4
LCD Controller
II:D-5, II:E-3
LCD Interface II:7-1

M

Measure Current Directly with a Current Meter in Series I:5-1
Measure Voltage Across a Series Resistor I:5-1
Measurement Guidelines I:5-1
Memory Controller
II:D-4, II:E-3
Layout Notes II:6-6
Routing Guidelines for 0.5mm and 0.65 mm Ball Pitch II:6-6
Memory Stick Host Controller
II:D-10, II:E-5
Memory Stick Host Interface II:17-1
Minimum Board Stack-up Configuration used for Signal Integrity II:6-8
Mixed Voltage Design Considerations I:4-1
MMC Protocol
Block and Schematic Diagrams II:15-5
Layout Notes II:15-6
Signals II:15-4
MMC/SD/SDIO Mode
Using MMC Protocol II:15-4
Using SD or SDIO Protocols II:15-7
Modes of Operation II:5-3
Modes of Operation Overview
II:6-9, II:7-6, II:14-4, II:15-4, II:18-6
Modes of Operations II:3-6
MultiMediaCard/SD/SDIO Controller
II:15-1, II:D-9, II:E-5

N

nBATT_FAULT II:3-4
nVDD_FAULT II:3-3

O

On-Chip Oscillator
13.000-MHz Crystal II:3-7
32.768-KHz Crystal II:3-7
Operating System Timers
II:D-12, II:E-5
Operation II:26-3
GPIOn and GPIOx are Different Pins II:12-3
GPIOn and GPIOx are Same Pin II:12-4
OS Timer Interface II:22-1
OTG ID II:12-11
Other Uses of I2C II:9-3
Output Control II:22-3
Overview
I:4-1, I:5-1, II:3-1, II:4-1, II:5-1, II:6-1, II:7-1, II:8-1, II:9-1, II:10-1, II:11-1, II:12-1, II:13-1, II:14-1, II:15-1, II:16-1, II:17-1, II:18-1, II:19-1, II:20-1, II:21-1, II:22-1, II:23-1, II:24-1, II:25-1, II:26-1, II:27-1

P

Package and Pins
II:2-1
package types II:2-2
Package Introduction I:1-3
Package to Board Assembly Process I:2-10
Panel Connector II:7-5
Passive Color
Dual-Scan Mode II:7-14
Single-Scan Mode II:7-12
Passive Monochrome
Dual-Scan Mode II:7-10
Single-Scan Double-Pixel Mode II:7-8
Single-Scan Mode II:7-6
Pause
DR State II:26-12
IR State II:26-13
PC Card
Interface II:6-24
Layout Notes II:6-30
PC Card Signals II:6-26
PCB
Component Placement I:2-4
Design Guidelines I:2-1
Escape Routing I:2-6
Keep-out Zones I:2-9
Layer Assignment (Stackup) I:2-2
PC-Card Block Diagrams II:6-27
PMIC Devices
II:E-2
Power Enable (PWR_EN) II:3-3
Power Interface
II:3-8
Power Manager
II:D-2
Power Manager I2C
Clock (PWR_SCL) II:3-3
Data (PWR_SDA) II:3-3
Power Manager Interface Control Signals II:3-2
Power Measurements I:5-1
Power Supplies II:3-8
Power Supply Decoupling Requirements I:2-10
Preconditioning and Moisture Sensitivity I:2-11
Pull-Up Resistors II:26-4
Pull-Ups and Pull-Downs II:9-4
Pulse-Width Modulator Controller

II:D-13, II:E-5
Pulse-Width Modulator Interface II:23-1
PXA Processor Developer's Kit (DVK) II:B-1
PXA27x Compatibility Channels 0-3 Block II:22-2
PXA27x DVK
Bill-of-Materials II:C-1
Block Diagram II:A-1
PXA27x Processor
BITCLK signal to CODEC II:14-4
USIM Interface Signals II:19-2

R

Real Time Clock Interface II:21-1
Real-Time Clock
II:D-11, II:E-5
Recommended
Mobile Handset Dimensions I:2-9
Pull-down Resistors II:18-4
Reduce Power During Standby and Sleep Mode II:18-4
Register Descriptions II:26-15
Required Power Supplies I:4-1
ROM
Block Diagram II:6-18
Interface II:6-17
Layout Notes II:6-18
Signals II:6-17
Run-Test/Idle State II:26-11

S

Schematics/Block Diagram
II:7-3, II:7-7, II:7-9, II:7-11, II:7-13, II:7-15, II:7-17, II:7-19, II:7-21, II:7-23, II:9-2, II:17-1
SD and SDIO Protocol
Block and Schematic Diagrams II:15-8
Layout Notes II:15-10
Signals II:15-7
SDCLK and SDCAS Routing Guidelines II:6-7
SDRAM
Interface II:6-9
Layout Notes II:6-12
Memory Block Diagram II:6-11
Signals II:6-9
Select-DR-Scan State II:26-11
Select-IR-Scan State II:26-13
Self-Powered Devices II:12-2
Serial Bus Host Controller, Universal II:D-10
Shift-DR State II:26-11
Shift-IR State II:26-13
Signal Descriptions II:26-3
Signal Pin Descriptions I:1-3
Signal Routing and Buffering II:7-4
Signals II:11-1
II:3-1, II:4-1, II:5-1, II:5-3, II:5-5, II:6-3, II:7-2, II:7-6, II:7-8, II:7-10, II:7-12, II:7-14, II:7-16, II:7-18, II:7-20, II:7-22, II:8-2, II:9-1, II:10-2, II:11-1, II:12-1, II:13-1, II:14-2, II:14-4, II:14-6, II:15-2, II:17-1, II:18-2, II:18-6, II:18-8, II:18-10, II:19-2, II:20-1, II:21-1, II:22-1, II:23-1, II:24-1, II:25-2, II:27-2
UART signals named and described II:10-2
Silicon Daisy Chain Evaluation Units I:2-10
Smart Panel II:7-22
Software Debug Module
II:D-15
Specific Combinations of Direct Keys II:18-4
SPI Mode Signals II:15-11
SPI Mode with MMC, SD Card, and SDIO Card Devices II:15-11
SPI Protocol
Block and Schematic Diagrams II:15-12
Layout Notes II:15-12
SRAM
Block Diagram II:6-20
Interface II:6-18
Layout Notes II:6-21
Signals II:6-19
SSP Port Interface II:8-1
SSP Serial Port
II:D-6, II:E-3
Standard SSP Configuration Scheme II:8-3
Standard UART II:10-6
Block Diagram II:10-6
Signals II:10-6
Subscriber ID Interface, Universal II:E-5
System Architecture
II:D-1, II:E-1
System Bus, Signal Routing Guidelines II:6-6
System Memory Interface II:6-1
System Power Enable (SYS_EN) II:3-3

T

TAP Controller Reset II:26-3
Test
Features II:26-2
I/O signals II:26-3
Overview II:26-1
Pull-up resistors II:26-4
TAP controller II:26-10
Test Access Port Controller II:26-10
state diagram II:26-10
Test Data Registers II:26-7
Test Interface
II:D-15
Test Register Summary II:26-16
Test-Logic-Reset State II:26-11

Thermal Considerations I:2-10
Tray Specifications I:2-11
TRISTATE II:26-6
Types of UARTs II:10-3

U

UART
    FIFO mode II:10-1
    non-FIFO mode II:10-1
UART Interfaces II:10-1
UARTs
    II:D-7, II:E-3
Universal Asynchronous Receiver/Transmitter II:10-1
Universal Serial Bus Device Controller
    II:12-1
Universal Serial Bus Host Controller
    II:D-10, II:E-5
Universal Serial Bus Host Interface II:20-1
Universal Subscriber ID Interface
    II:D-10, II:E-5
Universal Subscriber ID Interfacev II:E-5
Universal Subscriber Identity Module
    Card Interface Description II:19-3
Update-DR State II:26-12
Update-IR State II:26-14
USB Client
    Signals II:12-1
USB Client Controller II:12-1
USB Device Controller II:D-8
    II:D-8, II:E-3
    UDC registers II:12-11
USB On-The-GO
    Transceiver Usage II:12-6
USB On-The-Go
    Transceivers II:E-4
USB On-The-Go Transceivers II:E-4
USIM
    Card Interface Signals II:19-3
    Controller Interface II:19-1

V

Variable Latency Input/Output Interface II:6-22
VLIO
    Block Diagram II:6-24
    Memory Layout Notes II:6-24
    Memory Signals II:6-23

§§