



Intel® *Intel® PXA255 Processor*

Design Guide

March, 2003

Order Number: 278694-001



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS 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 products are not intended for use in medical, life saving, or life sustaining 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® PXA255 Processors and PXA210 processors 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 your local Intel sales office or your distributor to obtain the latest specifications and before placing your 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, 2003

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 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, RemoteExpress, Screamline, Shiva, SmartDie, Solutions960, Sound Mark, StorageExpress, The Computer Inside, The Journey Inside, This Way In, TokenExpress, Trillium, Vivonic, and VTune 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

1	Introduction	1-1
1.1	Functional Overview	1-1
1.2	Package Information	1-2
1.2.1	Package Introduction	1-2
1.2.2	Signal Pin Descriptions	1-3
2	System Memory Interface	2-1
2.1	Overview	2-1
2.2	SDRAM Interface	2-3
2.3	SDRAM memory wiring diagram	2-4
2.4	SDRAM Support	2-5
2.5	SDRAM Address Mapping	2-6
2.6	Static Memory	2-7
2.6.1	Overview	2-7
2.6.2	Boot Time Defaults	2-8
2.6.3	Flash Memory	2-9
2.6.3.1	Synchronous Intel® StrataFlash® Memory Reset	2-9
2.6.4	SRAM / ROM / Flash / Synchronous Fast Flash Memory Options	2-9
2.6.5	Variable Latency I/O Interface Overview	2-10
2.6.6	External Logic for PCMCIA Implementation	2-12
2.6.7	DMA / Companion Chip Interface	2-15
2.7	System Memory Layout Guidelines	2-18
2.7.1	System Memory Topologies (Min and Max Simulated Loading)	2-18
2.7.2	System Memory Recommended Trace Lengths	2-19
3	LCD Display Controller	3-1
3.1	LCD Display Overview	3-1
3.2	Passive (DSTN) Displays	3-1
3.2.1	Typical Connections for Passive Panel Displays	3-2
3.2.1.1	Passive Monochrome Single Panel Displays	3-2
3.2.1.2	Passive Monochrome Single Panel Displays, Double-Pixel Data	3-3
3.2.1.3	Passive Monochrome Dual Panel Displays	3-3
3.2.1.4	Passive Color Single Panel Displays	3-4
3.2.1.5	Passive Color Dual Panel Displays	3-4
3.3	Active (TFT) Displays	3-5
3.3.1	Typical connections for Active Panel Displays	3-6
3.4	PXA255 processor Pinout	3-7
3.5	Additional Design Considerations	3-8
3.5.1	Contrast Voltage	3-8
3.5.2	Backlight Inverter	3-8
3.5.3	Signal Routing and Buffering	3-8
3.5.4	Panel Connector	3-9
4	USB Interface	4-1
4.1	Self Powered Device	4-1
4.1.1	Operation if GPIOn and GPIOx are Different Pins	4-1
4.1.2	Operation if GPIOn and GPIOx are the Same Pin	4-2

4.2	Bus Powered Device	4-2
5	MultiMediaCard (MMC)	5-1
5.1	Schematics	5-1
5.1.1	Signal Description	5-1
5.1.2	How to Wire	5-2
5.1.2.1	SDCard Socket	5-4
5.1.2.2	MMC Socket	5-4
5.1.3	Simplified Schematic	5-5
5.1.4	Pull-up and Pull-down	5-5
5.2	Utilized Features	5-6
6	AC97	6-1
6.1	Schematics	6-1
6.2	Layout	6-2
7	I2C	7-1
7.1	Schematics	7-1
7.1.1	Signal Description	7-1
7.1.2	Digital-to-Analog Converter (DAC)	7-2
7.1.3	Other Uses of I2C	7-2
7.1.4	Pull-Ups and Pull-Downs	7-3
7.2	Utilized Features	7-4
8	Power and Clocking	8-1
8.1	Operating Conditions	8-1
8.2	Electrical Specifications	8-2
8.2.1	Power Supply Connectivity	8-2
8.3	Example Form Factor Reference Design Power Delivery Example	8-7
8.3.1	Power System	8-7
8.3.1.1	Power System Configuration	8-8
8.3.2	CORE Power	8-9
8.3.3	PLL Power	8-9
8.3.4	I/O 3.3 V Power	8-9
8.3.5	Peripheral 5.5 V Power	8-9
9	JTAG/Debug Port	9-1
9.1	Description	9-1
9.2	Schematics	9-1
9.3	Layout	9-2
A	SA-1110/Processor Migration	A-1
A.1	SA-1110 Hardware Migration Issues	A-2
A.1.1	Hardware Compatibility	A-2
A.1.2	Signal Changes	A-2
A.1.3	Power Delivery	A-4
A.1.4	Package	A-4
A.1.5	Clocks	A-4
A.1.6	UCB1300	A-4
A.2	SA-1110 to PXA255 Processor Software Migration Issues	A-5
A.2.1	Software Compatibility	A-5

A.2.2	Address space	A-5
A.2.3	Page Table Changes	A-6
A.2.4	Configuration registers	A-6
A.2.5	DMA	A-6
A.3	Using New PXA255 Processor Features	A-7
A.3.1	Intel® XScale™ Microarchitecture	A-7
A.3.2	Debugging	A-8
A.3.3	Cache Attributes	A-8
A.3.4	Other features	A-8
A.3.5	Conclusion	A-8
B	Example Form Factor Reference Design Schematic Diagrams	B-1
B.1	Notes	B-1
B.2	Schematic Diagrams	B-1
C	BBPXA2xx Development Baseboard Schematic Diagram	C-1
C.1	Schematic Diagram	C-1
D	PXA250 Processor Card Schematic Diagram	D-1
D.1	Schematic Diagram	D-1

Figures

1-1	Processor Block Diagram	1-2
1-2	PXA255 Processor	1-13
2-1	General Memory Interface Configuration	2-2
2-2	SDRAM Memory System Example.....	2-4
2-3	Flash Memory Reset Using State Machine	2-9
2-4	Flash Memory Reset Logic if Watchdog Reset is Not Necessary	2-9
2-5	32-Bit Variable Latency I/O Read Timing (Burst-of-Four, One Wait Cycle Per Beat).....	2-11
2-6	Expansion Card External Logic for a Two-Socket Configuration.....	2-13
2-7	Expansion Card External Logic for a One-Socket Configuration.....	2-14
2-8	Alternate Bus Master Mode	2-16
2-9	Variable Latency I/O	2-17
2-10	CS, CKE, DQM, CLK, MA minimum loading topology.....	2-18
2-11	CS, CKE, DQM, CLK, MA Maximum Loading Topology	2-18
2-12	MD Minimum Loading Topology.....	2-18
2-13	MD maximum loading topology	2-19
3-1	Single Panel Monochrome Passive Display Typical Connection	3-3
3-2	Passive Monochrome Single Panel Displays, Double-Pixel Data Typical Connection.....	3-3
3-3	Passive Monochrome Dual Panel Displays Typical Connection	3-4
3-4	Passive Color Single Panel Displays Typical Connection	3-4
3-5	Passive Color Dual Panel Displays Typical Connection.....	3-5
3-6	Active Color Display Typical Connection.....	3-7
4-1	Self Powered Device	4-1
5-1	Processor MMC and SDCard Signal Connections.....	5-3
5-2	Processor MMC to SDCard Simplified Signal Connection	5-5
6-1	AC97 connection	6-1
7-1	Linear Technology DAC with I2C Interface	7-2
7-2	Using an Analog Switch to Allow a Second CF Card	7-3
7-3	I ² C Pull-Ups and Pull-Downs	7-3
8-1	Example Form Factor Reference Design Power System Design.....	8-8
9-1	JTAG/Debug Port Wiring Diagram	9-1

Tables

1-1	Related Documentation.....	1-1
1-2	Pin & Signal Descriptions for the PXA255 Processor	1-3
1-3	Pin Description Notes.....	1-12
1-4	PXA255 Processor Pinout — Ballpad Number Order	1-14
2-1	Memory Address Map	2-3
2-2	SDRAM Memory Types Supported by the Processor	2-5
2-3	Normal Mode Memory Address Mapping.....	2-6
2-4	Processor Compatibility Mode Address Line Mapping.....	2-7
2-5	Valid Booting Configurations Based for the PXA255 processor	2-8
2-6	BOOT_SEL Definitions	2-8
2-7	SRAM / ROM / Flash / Synchronous Fast Flash AC Specifications.....	2-10
2-8	Variable Latency I/O Interface AC Specifications	2-11
2-9	Card Interface (PCMCIA or Compact Flash) AC Specifications	2-14
2-10	Minimum and Maximum Trace Lengths for the SDRAM Signals	2-19
3-1	LCD Controller Data Pin Utilization	3-1

3-2	Passive Display Pins Required	3-2
3-3	Active Display Pins Required	3-6
3-4	PXA255 processor LCD Controller Ball Positions	3-7
5-1	MMC Signal Description	5-1
5-2	SDCard Socket Signals	5-2
5-3	MMC Controller Supported Sockets and Devices	5-2
5-4	SDCard Pull-up and Pull-down Resistors	5-6
5-5	MMC Pull-up and Pull-down Resistors	5-6
7-1	I2C Signal Description	7-1
8-1	Voltage, Temperature, and Frequency Electrical Specifications	8-1
8-2	Absolute Maximum Ratings	8-2
8-3	PXA255 Processor VCCN vs. VCCQ	8-2

Revision History

Date	Revision	Description
January 2003	-001	Initial Release

This document presents design recommendations, board schematics, and debug recommendations for the Intel® PXA255 Processor (PXA255 processor). The PXA255 processor is a 32-bit device.

The guidelines presented in this document ensure maximum flexibility for board designers, while reducing the risk of board-related issues. Use the schematics in [Appendix B, “Example Form Factor Reference Design Schematic Diagrams”](#) as a reference for your own design. While the included schematics cover a specific design, the core schematics remain the same for most PXA255 processor based platforms. Consult the debug recommendations when debugging a processor based system. To ensure the correct implementation of the debug port (refer to [Section 9](#) for more information), these debug recommendations should be understood before completing a board design, in addition to other debug features.

Table 1-1. Related Documentation

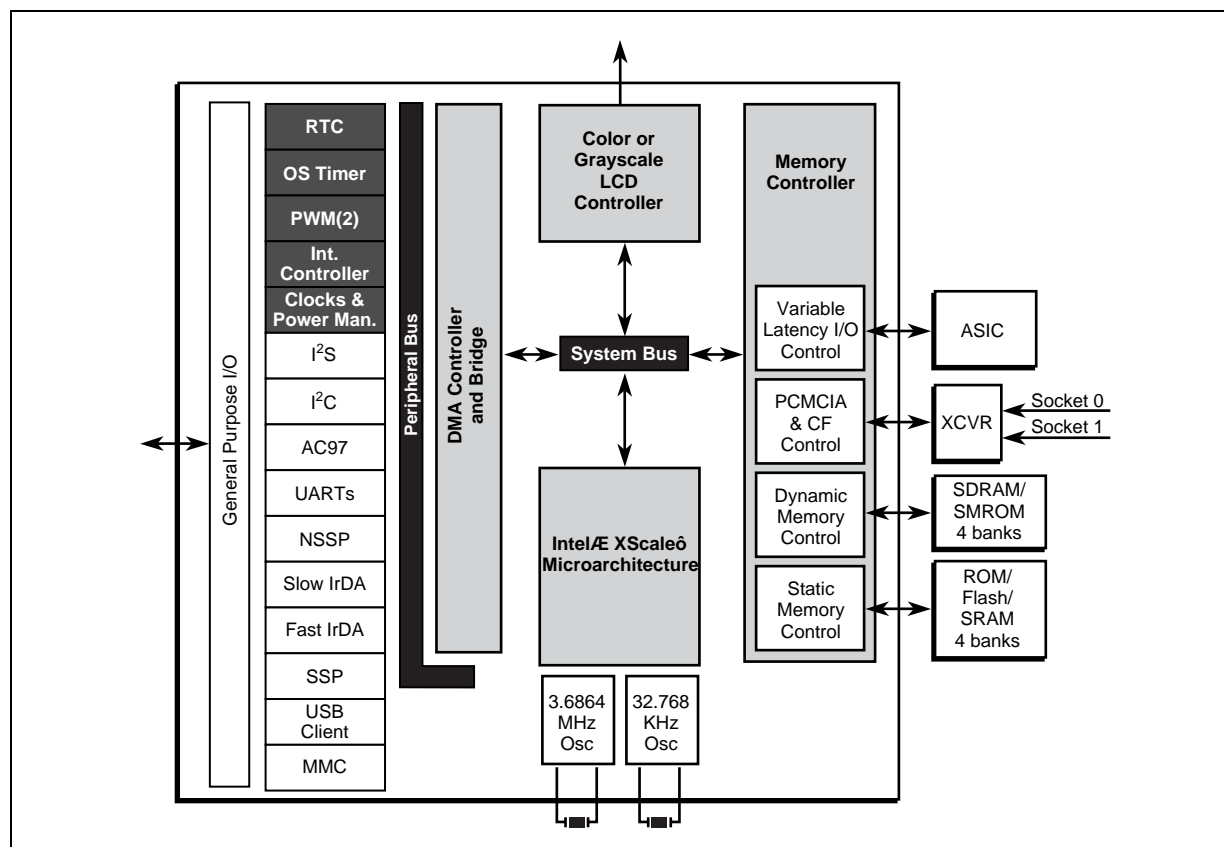
Document Title	Order Number
<i>Intel® PXA255 Processor Developer's Manual</i>	278693
<i>Intel® PXA255 Processor Electrical, Mechanical, and Thermal Specification</i>	278695

1.1 Functional Overview

The PXA255 processor is an integrated system-on-a-chip design based on the Intel® XScale™ microarchitecture. The PXA255 processor integrates the Intel® XScale™ microarchitecture core with many peripherals to let you design products for the handheld market.

[Figure 1-1 on page 1-2](#) is a block diagram of the processor.

Figure 1-1. Processor Block Diagram



The PXA255 processor package is: 256 pin, 17x17 mBGA – 32-bit functionality.

Section 1.2.1, “Package Introduction” contains a breakdown of the features supported by the PXA255 processor.

1.2 Package Information

This section describes the package, pinout, and signal descriptions.

1.2.1 Package Introduction

Features of the PXA255 processor are:

- Core frequencies supported - 100 MHz - 400 MHz
- System memory interface
 - 100 MHz SDRAM
 - 4 MB to 256 MB of SDRAM memory
 - Support for 16, 64, 128, or 256 Mbit DRAM technologies

- 4 Banks of SDRAM, each supporting 64 MB of memory
- Clock enable (1 CKE pin is provided to put the entire SDRAM interface into self refresh)
- Supports as many as 6 static memory devices (SRAM, Flash, or VLIO)
- PCMCIA/Compact Flash card control pins
- LCD Controller pins
- Full Function UART
- Bluetooth UART
- Hardware UART
- MMC Controller pins
- SSP Pins
- Network SSP Pins
- USB Client Pins
- AC'97 Controller Pins
- Standard UART Pins
- I²C Controller pins
- PWM pins
- 15 dedicated GPIOs pins
- Integrated JTAG support

1.2.2 Signal Pin Descriptions

Table 1-2 defines the signal descriptions for the processor.

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 1 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
Memory Controller Pins				
MA[25:0]	OCZ	Memory address bus. (output) Signals the address requested for memory accesses.	Driven Low	Driven Low
MD[15:0]	ICOCZ	Memory data bus. (input/output) Lower 16 bits of the data bus.	Hi-Z	Driven Low
MD[31:16]	ICOCZ	Memory data bus. (input/output) Used for 32-bit memories.	Hi-Z	Driven Low
nOE	OCZ	Memory output enable. (output) Connect to the output enables of memory devices to control data bus drivers.	Driven High	Note [4]
nWE	OCZ	Memory write enable. (output) Connect to the write enables of memory devices.	Driven High	Note [4]
nSDCS[3:0]	OCZ	SDRAM CS for banks 3 through 0. (output) Connect to the chip select (CS) pins for SDRAM. For the PXA255 processor nSDCS0 can be Hi-Z, nSDCS1-3 cannot.	Driven High	Note [5]
DQM[3:0]	OCZ	SDRAM DQM for data bytes 3 through 0. (output) Connect to the data output mask enables (DQM) for SDRAM.	Driven Low	Driven Low

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 2 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
nSDRAS	OCZ	SDRAM RAS. (output) Connect to the row address strobe (RAS) pins for all banks of SDRAM.	Driven High	Driven High
nSDCAS	OCZ	SDRAM CAS. (output) Connect to the column address strobe (CAS) pins for all banks of SDRAM.	Driven High	Driven High
SDCKE[0]	OC	Synchronous Static Memory clock enable. (output) Connect to the CKE pins of SMROM. The memory controller provides control register bits for deassertion.	Driven Low	Driven Low
SDCKE[1]	OC	SDRAM and/or Synchronous Static Memory clock enable. (output) Connect to the clock enable pins of SDRAM. It is deasserted during sleep. SDCKE[1] is always deasserted upon reset. The memory controller provides control register bits for deassertion.	Driven low	Driven low
SDCLK[0]	OC	Synchronous Static Memory clock. (output) Connect to the clock (CLK) pins of SMROM. It is driven by either the internal memory controller clock, or the internal memory controller clock divided by 2. At reset, all clock pins are free running at the divide by 2 clock speed and may be turned off via free running control register bits in the memory controller. The memory controller also provides control register bits for clock division and deassertion of each SDCLK pin. SDCLK[0] control register assertion bit defaults to on if the boot-time static memory bank 0 is configured for SMROM.		
SDCLK[1]	OCZ	SDRAM Clocks (output) Connect SDCLK[1] and SDCLK[2] to the clock pins of SDRAM in bank pairs 0/1 and 2/3, respectively. They are driven by either the internal memory controller clock, or the internal memory controller clock divided by 2. At reset, all clock pins are free running at the divide by 2 clock speed and may be turned off via free running control register bits in the memory controller. The memory controller also provides control register bits for clock division and deassertion of each SDCLK pin. SDCLK[2:1] control register assertion bits are always deasserted upon reset.	Driven Low	Driven Low
SDCLK[2]	OC		Driven Low	Driven Low
nCS[5]/ GPIO[33]	ICOCZ	Static chip selects. (output) Chip selects to static memory devices such as ROM and Flash. Individually programmable in the memory configuration registers. nCS[5:0] can be used with variable latency I/O devices.	Hi-Z - Note [1]	Note [4]
nCS[4]/ GPIO[80]	ICOCZ			
nCS[3]/ GPIO[79]	ICOCZ			
nCS[2]/ GPIO[78]	ICOCZ			
nCS[1]/ GPIO[15]	ICOCZ			
nCS[0]	ICOCZ	Static chip select 0. (output) Chip select for the boot memory. nCS[0] is a dedicated pin.	Driven High	Note [4]
RD/nWR	OCZ	Read/Write for static interface. (output) Signals that the current transaction is a read or write.	Driven Low	Holds last state
RDY/ GPIO[18]	ICOCZ	Variable Latency I/O Ready pin. (input) Notifies the memory controller when an external bus device is ready to transfer data.	Hi-Z - Note [1]	Note [3]

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 3 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
L_DD[8]/ GPIO[66]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. Memory Controller alternate bus master request. (input) Allows an external device to request the system bus from the Memory Controller.	Hi-Z - Note [1]	Note [3]
L_DD[15]/ GPIO[73]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. Memory Controller grant. (output) Notifies an external device that it has been granted the system bus.	Hi-Z - Note [1]	Note [3]
MBGNT/ GP[13]	ICOCZ	Memory Controller grant. (output) Notifies an external device that it has been granted the system bus.	Hi-Z - Note [1]	Note [3]
MBREQ/ GP[14]	ICOCZ	Memory Controller alternate bus master request. (input) Allows an external device to request the system bus from the Memory Controller.	Hi-Z - Note [1]	Note [3]
PCMCIA/CF Control Pins				
nPOE/ GPIO[48]	ICOCZ	PCMCIA output enable. (output) Reads from PCMCIA memory and to PCMCIA attribute space.	Hi-Z - Note [1]	Note [5]
nPWE/ GPIO[49]	ICOCZ	PCMCIA write enable. (output) Performs writes to PCMCIA memory and to PCMCIA attribute space. Also used as the write enable signal for Variable Latency I/O.	Hi-Z - Note [1]	Note [5]
nPIOW/ GPIO[51]	ICOCZ	PCMCIA I/O write. (output) Performs write transactions to PCMCIA I/O space.	Hi-Z - Note [1]	Note [5]
nPIOR/ GPIO[50]	ICOCZ	PCMCIA I/O read. (output) Performs read transactions from PCMCIA I/O space.	Hi-Z - Note [1]	Note [5]
nPCE[2]/ GPIO[53]	ICOCZ	PCMCIA card enable 2. (output) Selects a PCMCIA card. nPCE[2] enables the high byte lane and nPCE[1] enables the low byte lane. MMC clock. (output) Clock signal for the MMC Controller.	Hi-Z - Note [1]	Note [5]
nPCE[1]/ GPIO[52]	ICOCZ	PCMCIA card enable 1. (outputs) Selects a PCMCIA card. nPCE[2] enables the high byte lane and nPCE[1] enables the low byte lane.	Hi-Z - Note [1]	Note [5]
nIOIS16/ GPIO[57]	ICOCZ	IO Select 16. (input) Acknowledge from the PCMCIA card that the current address is a valid 16 bit wide I/O address.	Hi-Z - Note [1]	Note [5]
nPWAIT/ GPIO[56]	ICOCZ	PCMCIA wait. (input) Driven low by the PCMCIA card to extend the length of the transfers to/from the PXA255 processor.	Hi-Z - Note [1]	Note [5]
PSKTSEL/ GPIO[54]	ICOCZ	PCMCIA socket select. (output) Used by external steering logic to route control, address, and data signals to one of the two PCMCIA sockets. When PSKTSEL is low, socket zero is selected. When PSKTSEL is high, socket one is selected. Has the same timing as the address bus.	Hi-Z - Note [1]	Note [5]
nPREG/ GPIO[55]	ICOCZ	PCMCIA Register select. (output) Indicates that the target address on a memory transaction is attribute space. Has the same timing as the address bus.	Hi-Z - Note [1]	Note [5]

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 4 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
LCD Controller Pins				
L_DD(7:0)/ GPIO[65:58]	ICOCZ	LCD display data. (outputs) Transfers pixel information from the LCD Controller to the external LCD panel.	Hi-Z - Note [1]	Note [3]
L_DD[8]/ GPIO[66]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. Memory Controller alternate bus master request. (input) Allows an external device to request the system bus from the Memory Controller.	Hi-Z - Note [1]	Note [3]
L_DD[9]/ GPIO[67]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. MMC chip select 0. (output) Chip select 0 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
L_DD[10]/ GPIO[68]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. MMC chip select 1. (output) Chip select 1 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
L_DD[11]/ GPIO[69]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. MMC clock. (output) Clock for the MMC Controller.	Hi-Z - Note [1]	Note [3]
L_DD[12]/ GPIO[70]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. RTC clock. (output) Real time clock 1 Hz tick.	Hi-Z - Note [1]	Note [3]
L_DD[13]/ GPIO[71]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. 3.6864 MHz clock. (output) Output from 3.6864 MHz oscillator.	Hi-Z - Note [1]	Note [3]
L_DD[14]/ GPIO[72]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. 32 kHz clock. (output) Output from the 32 kHz oscillator.	Hi-Z - Note [1]	Note [3]
L_DD[15]/ GPIO[73]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. Memory Controller grant. (output) Notifies an external device it has been granted the system bus.	Hi-Z - Note [1]	Note [3]
L_FCLK/ GPIO[74]	ICOCZ	LCD frame clock. (output) Indicates the start of a new frame. Also referred to as Vsync.	Hi-Z - Note [1]	Note [3]
L_LCLK/ GPIO[75]	ICOCZ	LCD line clock. (output) Indicates the start of a new line. Also referred to as Hsync.	Hi-Z - Note [1]	Note [3]
L_PCLK/ GPIO[76]	ICOCZ	LCD pixel clock. (output) Clocks valid pixel data into the LCD's line shift buffer.	Hi-Z - Note [1]	Note [3]
L_BIAS/ GPIO[77]	ICOCZ	AC bias drive. (output) Notifies the panel to change the polarity for some passive LCD panel. For TFT panels, this signal indicates valid pixel data.	Hi-Z - Note [1]	Note [3]
Full Function UART Pins				
FFRXD/ GPIO[34]	ICOCZ	Full Function UART Receive. (input) MMC chip select 0. (output) Chip select 0 for the MMC Controller.	Hi-Z - Note [1]	Note [3]

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 5 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
FFTXD/ GPIO[39]	ICOCZ	Full Function UART Transmit. (output) MMC chip select 1. (output) Chip select 1 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
FFCTS/ GPIO[35]	ICOCZ	Full Function UART Clear-to-Send. (input)	Hi-Z - Note [1]	Note [3]
FFDCD/ GPIO[36]	ICOCZ	Full Function UART Data-Carrier-Detect. (input)	Hi-Z - Note [1]	Note [3]
FFDSR/ GPIO[37]	ICOCZ	Full Function UART Data-Set-Ready. (input)	Hi-Z - Note [1]	Note [3]
FFRI/ GPIO[38]	ICOCZ	Full Function UART Ring Indicator. (input)	Hi-Z - Note [1]	Note [3]
FFDTR/ GPIO[40]	ICOCZ	Full Function UART Data-Terminal-Ready. (output)	Hi-Z - Note [1]	Note [3]
FFRTS/ GPIO[41]	ICOCZ	Full Function UART Request-to-Send. (output)	Hi-Z - Note [1]	Note [3]
Bluetooth UART Pins				
BTRXD/ GPIO[42]	ICOCZ	Bluetooth UART Receive. (input)	Hi-Z - Note [1]	Note [3]
BTTXD/ GPIO[43]	ICOCZ	Bluetooth UART Transmit. (output)	Hi-Z - Note [1]	Note [3]
BTCTS/ GPIO[44]	ICOCZ	Bluetooth UART Clear-to-Send. (input)	Hi-Z - Note [1]	Note [3]
BTRTS/ GPIO[45]	ICOCZ	Bluetooth UART Data-Terminal-Ready. (output)	Hi-Z - Note [1]	Note [3]
Standard UART and ICP Pins				
IRRXD/ GPIO[46]	ICOCZ	IrDA receive signal. (input) Receive pin for the FIR function. Standard UART receive. (input)	Hi-Z - Note [1]	Note [3]
IRTXD/ GPIO[47]	ICOCZ	IrDA transmit signal. (output) Transmit pin for the Standard UART, SIR and FIR functions. Standard UART transmit. (output)	Hi-Z - Note [1]	Note [3]
HWUART Pins				
HWTXD/ GPIO[48]	ICOCZ	Hardware UART Transmit Data.	Pulled High Note [1]	Note [3]
HWRXD/ GPIO[49]	ICOCZ	Hardware UART Receive Data.	Pulled High Note [1]	Note [3]
HWCTS/ GPIO[50]	ICOCZ	Hardware UART Clear-To-Send.	Pulled High Note [1]	Note [3]
HWRTS/ GPIO[51]	ICOCZ	Hardware UART Request-to-Send.	Pulled High Note [1]	Note [3]
MMC Controller Pins				
MMCMD	ICOCZ	Multimedia Card Command. (bidirectional)	Hi-Z	Hi-Z

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 6 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
MMDAT	ICOCZ	Multimedia Card Data. (bidirectional)	Hi-Z	Hi-Z
nPCE[2]/ GPIO[53]	ICOCZ	PCMCIA card enable 2. (outputs) Selects a PCMCIA card. Bit one enables the high byte lane and bit zero enables the low byte lane. MMC clock. (output) Clock signal for the MMC Controller.	Hi-Z - Note [1]	Note [5]
L_DD[9]/ GPIO[67]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. MMC chip select 0. (output) Chip select 0 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
L_DD[10]/ GPIO[68]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. MMC chip select 1. (output) Chip select 1 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
L_DD[11]/ GPIO[69]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. MMC clock. (output) Clock for the MMC Controller.	Hi-Z - Note [1]	Note [3]
FFRXD/ GPIO[34]	ICOCZ	Full Function UART Receive. (input) MMC chip select 0. (output) Chip select 0 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
FFTXD/ GPIO[39]	ICOCZ	Full Function UART Transmit. (output) MMC chip select 1. (output) Chip select 1 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
MMCCLK/ GP[6]	ICOCZ	MMC clock. (output) Clock signal for the MMC Controller.	Hi-Z - Note [1]	Note [3]
MMCCS0/ GP[8]	ICOCZ	MMC chip select 0. (output) Chip select 0 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
MMCCS1/ GP[9]	ICOCZ	MMC chip select 1. (output) Chip select 1 for the MMC Controller.	Hi-Z - Note [1]	Note [3]
SSP Pins				
SSPCLK/ GPIO[23]	ICOCZ	Synchronous Serial Port Clock. (output)	Hi-Z - Note [1]	Note [3]
SSPSFRM/ GPIO[24]	ICOCZ	Synchronous Serial Port Frame. (output)	Hi-Z - Note [1]	Note [3]
SSPTXD/ GPIO[25]	ICOCZ	Synchronous Serial Port Transmit. (output)	Hi-Z - Note [1]	Note [3]
SSPRXD/ GPIO[26]	ICOCZ	Synchronous Serial Port Receive. (input)	Hi-Z - Note [1]	Note [3]
SSPEXTCLK/ GPIO[27]	ICOCZ	Synchronous Serial Port External Clock. (input)	Hi-Z - Note [1]	Note [3]
Network SSP pins				
NSSPCLK/ GPIO[81]	ICOCZ	Network Synchronous Serial Port Clock.	Pulled High Note [1]	Note [3]
NSSPSFRM/ GPIO[82]	ICOCZ	Network Synchronous Serial Port Frame Signal.	Pulled High Note [1]	Note [3]

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 7 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
NSSPTXD/ GPIO[83]	ICOCZ	Network Synchronous Serial Port Transmit.	Pulled High Note [1]	Note [3]
NSSPRXD/ GPIO[84]	ICOCZ	Network Synchronous Serial Port Receive.	Pulled High Note [1]	Note [3]
USB Client Pins				
USB_P	IAOAZ	USB Client Positive. (bidirectional)	Hi-Z	Hi-Z
USB_N	IAOAZ	USB Client Negative pin. (bidirectional)	Hi-Z	Hi-Z
AC97 Controller and I ² S Controller Pins				
BITCLK/ GPIO[28]	ICOCZ	AC97 Audio Port bit clock. (input) AC97 clock is generated by Codec 0 and fed into the PXA255 processor and Codec 1. AC97 Audio Port bit clock. (output) AC97 clock is generated by the PXA255 processor. I²S bit clock. (input) I2S clock is generated externally and fed into PXA255 processor. I²S bit clock. (output) I2S clock is generated by the PXA255 processor.	Hi-Z - Note [1]	Note [3]
SDATA_IN0/ GPIO[29]	ICOCZ	AC97 Audio Port data in. (input) Input line for Codec 0. I²S data in. (input) Input line for the I ² S Controller.	Hi-Z - Note [1]	Note [3]
SDATA_IN1/ GPIO[32]	ICOCZ	AC97 Audio Port data in. (input) Input line for Codec 1. I²S system clock. (output) System clock from I ² S Controller.	Hi-Z - Note [1]	Note [3]
SDATA_OUT/ GPIO[30]	ICOCZ	AC97 Audio Port data out. (output) Output from the PXA255 processor to Codecs 0 and 1. I²S data out. (output) Output line for the I ² S Controller.	Hi-Z - Note [1]	Note [3]
SYNC/ GPIO[31]	ICOCZ	AC97 Audio Port sync signal. (output) Frame sync signal for the AC97 Controller. I²S sync. (output) Frame sync signal for the I ² S Controller.	Hi-Z - Note [1]	Note [3]
nACRESET	OC	AC97 Audio Port reset signal. (output)	Driven Low	Driven Low
I ² C Controller Pins				
SCL	ICOCZ	I²C clock. (bidirectional)	Hi-Z	Hi-Z
SDA	ICOCZ	I²C data. (bidirectional).	Hi-Z	Hi-Z
PWM Pins				
PWM[1:0]/ GPIO[17:16]	ICOCZ	Pulse Width Modulation channels 0 and 1. (outputs)	Hi-Z - Note [1]	Note [3]
DMA Pins				
DREQ[1:0]/ GPIO[19:20]	ICOCZ	DMA Request. (input) Notifies the DMA Controller that an external device requires a DMA transaction. DREQ[1] is GPIO[19]. DREQ[0] is GPIO[20].	Hi-Z - Note [1]	Note [3]
GPIO Pins				
GPIO[1:0]	ICOCZ	General Purpose I/O. Wakeup sources on both rising and falling edges on nRESET.	Hi-Z - Note [1]	Note [3]
GPIO[14:2]	ICOCZ	General Purpose I/O. More wakeup sources for sleep mode.	Hi-Z - Note [1]	Note [3]

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 8 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
GPIO[22:21]	ICOCZ	General Purpose I/O. Additional General Purpose I/O pins.	Hi-Z - Note [1]	Note [3]
Crystal and Clock Pins				
PXTAL	IA	3.6864 Mhz crystal input. No external lops are required.	Note [2]	Note [2]
PEXTAL	OA	3.6864 Mhz crystal output. No external lops are required.	Note [2]	Note [2]
XTAL	IA	32 Khz crystal input. No external lops are required.	Note [2]	Note [2]
TEXTAL	OA	32 Khz crystal output. No external lops are required.	Note [2]	Note [2]
L_DD[12]/ GPIO[70]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. RTC clock. (output) Real time clock 1 Hz tick.	Hi-Z - Note [1]	Note [3]
L_DD[13]/ GPIO[71]	ICOCZ	LCD display data. (output) Transfers the pixel information from the LCD Controller to the external LCD panel. 3.6864 MHz clock. (output) Output from 3.6864 MHz oscillator.	Hi-Z - Note [1]	Note [3]
L_DD[14]/ GPIO[72]	ICOCZ	LCD display data. (output) Transfers pixel information from the LCD Controller to the external LCD panel. 32 kHz clock. (output) Output from the 32 kHz oscillator.	Hi-Z - Note [1]	Note [3]
48MHz/GP[7]	ICOCZ	48 MHz clock. (output) Peripheral clock output derived from the PLL. NOTE: This clock is only generated when the USB unit clock enable is set.	Hi-Z - Note [1]	Note [3]
RTCCLK/ GP[10]	ICOCZ	Real time clock. (output) 1 Hz output derived from the 32kHz or 3.6864MHz output.	Hi-Z - Note [1]	Note [3]
3.6MHz/GP[11]	ICOCZ	3.6864 MHz clock. (output) Output from 3.6864 MHz oscillator.	Hi-Z - Note [1]	Note [3]
32kHz/GP[12]	ICOCZ	32 kHz clock. (output) Output from the 32 kHz oscillator.	Hi-Z - Note [1]	Note [3]
Miscellaneous Pins				
BOOT_SEL [2:0]	IC	Boot select pins. (input) Indicates type of boot device.	Input	Input
PWR_EN	OC	Power Enable for the power supply. (output) When negated, it signals the power supply to remove power to the core because the system is entering sleep mode.	Driven High	Driven low while entering sleep mode. Driven high when sleep exit sequence begins.
nBATT_FAULT	IC	Main Battery Fault. (input) Signals that main battery is low or removed. Assertion causes PXA255 processor to enter sleep mode or force an Imprecise Data Exception, which cannot be masked. PXA255 processor will not recognize a wakeup event while this signal is asserted. Minimum assertion time for nBATT_FAULT is 1 ms.	Input	Input
nVDD_FAULT	IC	VDD Fault. (input) Signals that the main power source is going out of regulation. nVDD_FAULT causes the PXA255 processor to enter sleep mode or force an Imprecise Data Exception, which cannot be masked. nVDD_FAULT is ignored after a wakeup event until the power supply timer completes (approximately 10 ms). Minimum assertion time for nVDD_FAULT is 1 ms.	Input	Input

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 9 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
nRESET	IC	Hard reset. (input) Level sensitive input used to start the processor from a known address. Assertion causes the current instruction to terminate abnormally and causes a reset. When nRESET is driven high, the processor starts execution from address 0. nRESET must remain low until the power supply is stable and the internal 3.6864 MHz oscillator has stabilized.	Input	Input. Driving low during sleep will cause normal reset sequence and exit from sleep mode.
nRESET_OUT	OC	Reset Out. (output) Asserted when nRESET is asserted and deasserts after nRESET is deasserted but before the first instruction fetch. nRESET_OUT is also asserted for “soft” reset events: sleep, watchdog reset, or GPIO reset.	Driven low during any reset sequence - driven high prior to first fetch.	Driven Low
JTAG and Test Pins				
nTRST	IC	JTAG Test Interface Reset. Resets the JTAG/Debug port. If JTAG/Debug is used, drive nTRST from low to high either before or at the same time as nRESET. If JTAG is not used, nTRST must be either tied to nRESET or tied low.	Input	Input
TDI	IC	JTAG test data input. (input) Data from the JTAG controller is sent to the PXA255 processor using this pin. This pin has an internal pull-up resistor.	Input	Input
TDO	OCZ	JTAG test data output. (output) Data from the PXA255 processor is returned to the JTAG controller using this pin.	Hi-Z	Hi-Z
TMS	IC	JTAG test mode select. (input) Selects the test mode required from the JTAG controller. This pin has an internal pull-up resistor.	Input	Input
TCK	IC	JTAG test clock. (input) Clock for all transfers on the JTAG test interface.	Input	Input
TEST	IC	Test Mode. (input) Reserved. Must be grounded.	Input	Input
TESTCLK	IC	Test Clock. (input) Reserved. Must be grounded.	Input	Input
Power and Ground Pins				
VCC	SUP	Positive supply for internal logic. Must be connected to the low voltage supply on the PCB.	Powered	Note [6]
VSS	SUP	Ground supply for internal logic. Must be connected to the common ground plane on the PCB.	Grounded	Grounded
PLL_VCC	SUP	Positive supply for PLLs and oscillators. Must be connected to the common low voltage supply.	Powered	Note [6]
PLL_VSS	SUP	Ground supply for the PLL. Must be connected to common ground plane on the PCB.	Grounded	Grounded
VCCQ	SUP	Positive supply for all CMOS I/O except memory bus and PCMCIA pins. Must be connected to the common 3.3v supply on the PCB.	Powered	Note [7]

Table 1-2. Pin & Signal Descriptions for the PXA255 Processor (Sheet 10 of 10)

Pin Name	Type	Signal Descriptions	Reset State	Sleep State
VSSQ	SUP	Ground supply for all CMOS I/O except memory bus and PCMCIA pins. Must be connected to the common ground plane on the PCB.	Grounded	Grounded
VCCN	SUP	Positive supply for memory bus and PCMCIA pins. Must be connected to the common 3.3v or 2.5v supply on the PCB.	Powered	Note [7]
VSSN	SUP	Ground supply for memory bus and PCMCIA pins. Must be connected to the common ground plane on the PCB.	Grounded	Grounded

Table 1-3. Pin Description Notes

Note	Description
[1]	GPIO Reset Operation: Configured as GPIO inputs by default after any reset. The input buffers for these pins are disabled to prevent current drain and the pins are pulled high with 10K to 60K internal resistors. The input paths must be enabled and the pull-ups turned off by clearing the Read Disable Hold (RDH) bit described in Section 3.5.7, "Power Manager Sleep Status Register (PSSR)" on page 3-27 of the <i>Intel® PXA255 Processor Developer's Manual</i> . Even though sleep mode sets the RDH bit, the pull-up resistors are not re-enabled by sleep mode.
[2]	Crystal oscillator pins: These pins are used to connect the external crystals to the on-chip oscillators. Refer to Section 3.3.1, "32.768 kHz Oscillator" on page 3-4 and Section 3.3.2, "3.6864 MHz Oscillator" on page 3-4 in the <i>Intel® PXA255 Processor Developer's Manual</i> for details on sleep mode operation.
[3]	GPIO Sleep operation: During the transition into sleep mode, the state of these pins is determined by the corresponding PGSRn. See Section 3.5.10, "Power Manager GPIO Sleep State Registers (PGSR0, PGSR1, PGSR2)" and Section 4.1.3.2, "GPIO Pin Direction Registers (GPDR0, GPDR1, GPDR2)" on page 4-8 in the <i>Intel® PXA255 Processor Developer's Manual</i> . If selected as an input, this pin does not drive during sleep. If selected as an output, the value contained in the Sleep State Register is driven out onto the pin and held there while the PXA255 processor is in sleep mode. GPIOs configured as inputs after exiting sleep mode cannot be used until PSSR[RDH] is cleared.
[4]	Static Memory Control Pins: During sleep mode, these pins can be programmed to either drive the value in the Sleep State Register or to be placed in Hi-Z. To select the Hi-Z state, software must set the FS bit in the Power Manager General Configuration Register. If PCFR[FS] is not set, then during the transition to sleep these pins function as described in [3], above. For nWE, nOE, and nCS[0], if PCFR[FS] is not set, they are driven high by the Memory Controller before entering sleep. If PCFR[FS] is set, these pins are placed in Hi-Z.
[5]	PCMCIA Control Pins: During sleep mode: Can be programmed either to drive the value in the Sleep State Register or to be placed in Hi-Z. To select the Hi-Z state, software must set PCFR[FP]. If it is not set, then during the transition to sleep these pins function as described in [3], above.
[6]	During sleep, this supply may be driven low. This supply must never be high impedance.
[7]	Remains powered in sleep mode.

Intel® PXA255 Processor Design Guide

Table 1-4. PXA255 Processor Pinout — Ballpad Number Order (Sheet 1 of 3)

Ball #	Signal	Ball #	Signal	Ball #	Signal
A1	VCCN	F7	GPIO[10]	L13	GPIO[2]
A2	L_DD[13]/GPIO[71]	F8	FFRTS/GPIO[41]	L14	VSSQ
A3	L_DD[12]/GPIO[70]	F9	SSPCLK/GPIO[23]	L15	TEXTAL
A4	L_DD[11]/GPIO[69]	F10	FFDTR/GPIO[40]	L16	TX TAL
A5	L_DD[9]/GPIO[67]	F11	VCC	M1	MA[14]
A6	L_DD[7]/GPIO[65]	F12	GPIO[9]	M2	MD[21]
A7	GPIO[11]	F13	BOOT_SEL[2]	M3	MA[15]
A8	L_BIAS/GPIO[77]	F14	GPIO[8]	M4	VCCN
A9	SSPRXD/GPIO[26]	F15	VSSQ	M5	MD[1]
A10	SDATA_OUT/GPIO[30]	F16	NSSP_CLK/GPIO[81]	M6	MD[6]
A11	SDA	G1	MA[0]	M7	MD[7]
A12	FFDCD/GPIO[36]	G2	VSSN	M8	DQM[0]
A13	FFRXD/GPIO[34]	G3	nSDCS[2]	M9	MD[8]
A14	FFCTS/GPIO[35]	G4	nWE	M10	MD[15]
A15	BTCTS/GPIO[44]	G5	nOE	M11	VCCQ
A16	SDATA_IN1/GPIO[32]	G6	nSDCS[1]	M12	GPIO[22]
B1	DQM[1]	G7	VCC	M13	nPREG/GPIO[55]
B2	DQM[2]	G8	VSSQ	M14	VCCN
B3	L_DD[15]/GPIO[73]	G9	VCC	M15	VSSN
B4	GPIO[14]	G10	VSSQ	M16	nIOIS16/GPIO[57]
B5	GPIO[13]	G11	TESTCLK	N1	MD[22]
B6	GPIO[12]	G12	TEST	N2	VSSN
B7	L_DD[3]/GPIO[61]	G13	BOOT_SEL[1]	N3	MA[16]
B8	L_PCLK/GPIO[76]	G14	VCCQ	N4	MD[0]
B9	SSPEXTCLK/GPIO[27]	G15	GPIO[7]	N5	VCCN
B10	FFRI/GPIO[38]	G16	BOOT_SEL[0]	N6	MD[4]
B11	FFDSR/GPIO[37]	H1	MA[2]	N7	VCCN
B12	USB_N	H2	MA[1]	N8	nCS[0]
B13	BTRXD/GPIO[42]	H3	MD[16]	N9	VCCN
B14	BTRTS/GPIO[45]	H4	VCCN	N10	MD[13]
B15	IRRXD/GPIO[46]	H5	MD[17]	N11	VCCN
B16	MMDAT	H6	MA[3]	N12	DREQ[0]/GPIO[20]
C1	RDY/GPIO[18]	H7	VSSQ	N13	VCCN
C2	VSSN	H8	VSS	N14	DREQ[1]/GPIO[19]
C3	L_DD[14]/GPIO[72]	H9	VSS	N15	GPIO[21]
C4	VSSQ	H10	VCC	N16	nPWAIT/GPIO[56]
C5	L_DD[8]/GPIO[66]	H11	nTRST	P1	MA[17]

Table 1-4. PXA255 Processor Pinout — Ballpad Number Order (Sheet 2 of 3)

Ball #	Signal	Ball #	Signal	Ball #	Signal
C6	VCCQ	H12	TCK	P2	MA[19]
C7	L_DD[2]/GPIO[60]	H13	TMS	P3	VCCN
C8	VSSQ	H14	GPIO[6]	P4	MA[25]
C9	BITCLK/GPIO[28]	H15	TDI	P5	MA[23]
C10	VCCQ	H16	TDO	P6	MD[24]
C11	VSSQ	J1	MA[7]	P7	MD[26]
C12	USB_P	J2	VSSN	P8	MD[27]
C13	VCCQ	J3	MA[6]	P9	nCS[2]/GPIO[78]
C14	VSSQ	J4	MD[18]	P10	MD[29]
C15	IRTXD/GPIO[47]	J5	MA[5]	P11	MD[12]
C16	VSS	J6	MA[4]	P12	MD[31]
D1	SDCLK[2]	J7	VCC	P13	nPOE/GPIO[48]
D2	SDCLK[0]	J8	VSS	P14	nPCE[1]/GPIO[52]
D3	RDnWR	J9	VSS	P15	VSSN
D4	VCCN	J10	VSSQ	P16	nPSKTSEL/GPIO[54]
D5	L_DD[10]/GPIO[68]	J11	GPIO[5]	R1	MA[18]
D6	L_DD[5]/GPIO[63]	J12	GPIO[4]	R2	VSSN
D7	L_DD[1]/GPIO[59]	J13	nRESET	R3	MA[20]
D8	L_LCLK/GPIO[75]	J14	VSSQ	R4	VSSN
D9	SSPTXD/GPIO[25]	J15	PLL_VCC	R5	MA[22]
D10	nACRESET	J16	PLL_VSS	R6	VSSN
D11	SCL	K1	MA[8]	R7	MD[25]
D12	PWM[1]/GPIO[17]	K2	MA[9]	R8	VSSN
D13	BTTXD/GPIO[43]	K3	MD[19]	R9	MD[10]
D14	MMCMD	K4	VCCN	R10	VSSN
D15	VCCQ	K5	MA[10]	R11	MD[30]
D16	NSSP_TXD_RXD/ GPIO[84]	K6	MA[11]	R12	VSSN
E1	nSDRAS	K7	VSSQ	R13	nCS[4]/GPIO[80]
E2	VSSN	K8	VCC	R14	VSSN
E3	SDCKE[1]	K9	VSSQ	R15	nPIOW/GPIO[51]
E4	SDCKE[0]	K10	VCC	R16	nPCE[2]/GPIO[53]
E5	L_DD[6]/GPIO[64]	K11	nRESET_OUT	T1	VSS
E6	L_DD[4]/GPIO[62]	K12	nBATT_FAULT	T2	VCCN
E7	L_DD[0]/GPIO[58]	K13	nVDD_FAULT	T3	MD[23]
E8	L_FCLK/GPIO[74]	K14	GPIO[3]	T4	MA[21]
E9	SSPSFRM/GPIO[24]	K15	PXTAL	T5	MA[24]
E10	SDATA_IN0/GPIO[29]	K16	PEXTAL	T6	MD[3]

Table 1-4. PXA255 Processor Pinout — Ballpad Number Order (Sheet 3 of 3)

Ball #	Signal	Ball #	Signal	Ball #	Signal
E11	SYNC/GPIO[31]	L1	MA[12]	T7	MD[5]
E12	PWM[0]/GPIO[16]	L2	VSSN	T8	nCS[1]/GPIO[15]
E13	FFTXD/GPIO[39]	L3	MA[13]	T9	nCS[3]/GPIO[79]
E14	VCCQ	L4	MD[20]	T10	MD[9]
E15	NSSP_TXD_RXD/ GPIO[83]	L5	MD[2]	T11	MD[11]
E16	NSSP_FRAME/GPIO[82]	L6	VCC	T12	MD[14]
F1	nSDCS[0]	L7	DQM[3]	T13	nCS[5]/GPIO[33]
F2	nSDCS[3]	L8	MD[28]	T14	nPWE/GPIO[49]
F3	nSDCAS	L9	VCC	T15	nPIOR/GPIO[50]
F4	VCCN	L10	GPIO[0]	T16	VCCN
F5	SDCLK[1]	L11	PWR_EN		
F6	VSSQ	L12	GPIO[1]		

This section describes the design guidelines for the system memory interface.

2.1 Overview

The external memory bus interface for the processor supports:

- 100 MHz SDRAM at 3.3 V
- 100 MHz SDRAM at 2.5 V
- Synchronous and asynchronous Burst mode and Page mode Flash
- Synchronous Mask ROM (SMROM)
- Page Mode ROM
- SRAM
- SRAM-like Variable Latency I/O (VLIO)
- PCMCIA expansion memory
- Compact Flash

Use the memory interface configuration registers to program the memory types. Refer to [Figure 1-1, “Processor Block Diagram” on page 1-2](#) for the block diagram of the Memory Controller configuration. Refer to [Figure 2-1, “Memory Address Map” on page 2-3](#) for the processor memory map. Refer to [Table 2-3, “Normal Mode Memory Address Mapping” on page 2-6](#) for alternate mode address mapping.

Figure 2-1. General Memory Interface Configuration

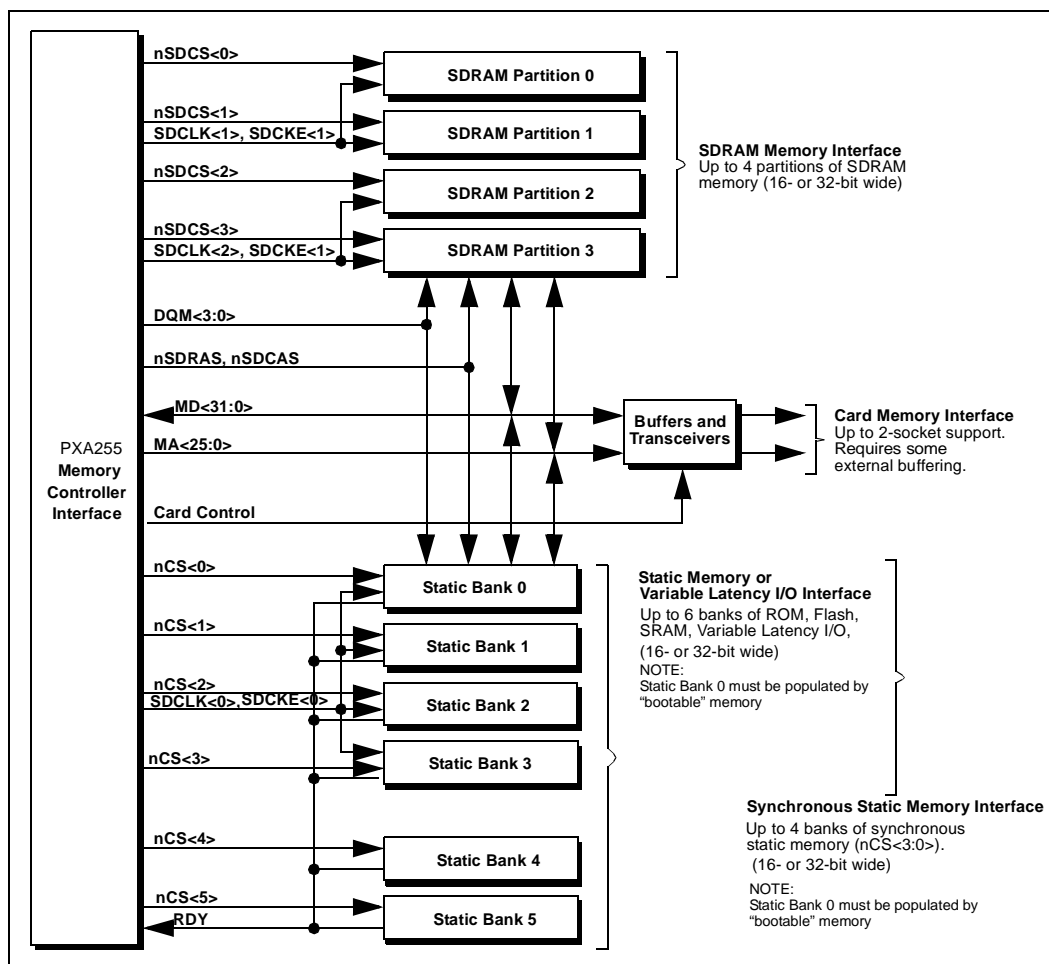


Table 2-1. Memory Address Map

Memory Address	Description
0x6000 0000	Reserved Address Space
0x5C00 0000	Reserved Address Space
0x5800 0000	Reserved Address Space
0x5400 0000	Reserved Address Space
0x5000 0000	Reserved Address Space
0x4C00 0000	Reserved Address Space
0x4800 0000	Memory Mapped Registers (Memory Ctl)
0x4400 0000	Memory Mapped Registers (LCD)
0x4000 0000	Memory Mapped Registers (Peripherals)
0x3000 0000	PCMCIA/CF – Slot 1
0x2000 0000	PCMCIA/CF – Slot 0
0x1C00 0000	Reserved Address Space
0x1800 0000	Reserved Address Space
0x1400 0000	Static Chip Select 5
0x1000 0000	Static Chip Select 4
0x0C00 0000	Static Chip Select 3
0x0800 0000	Static Chip Select 2
0x0400 0000	Static Chip Select 1
0x0000 0000	Static Chip Select 0

2.2 SDRAM Interface

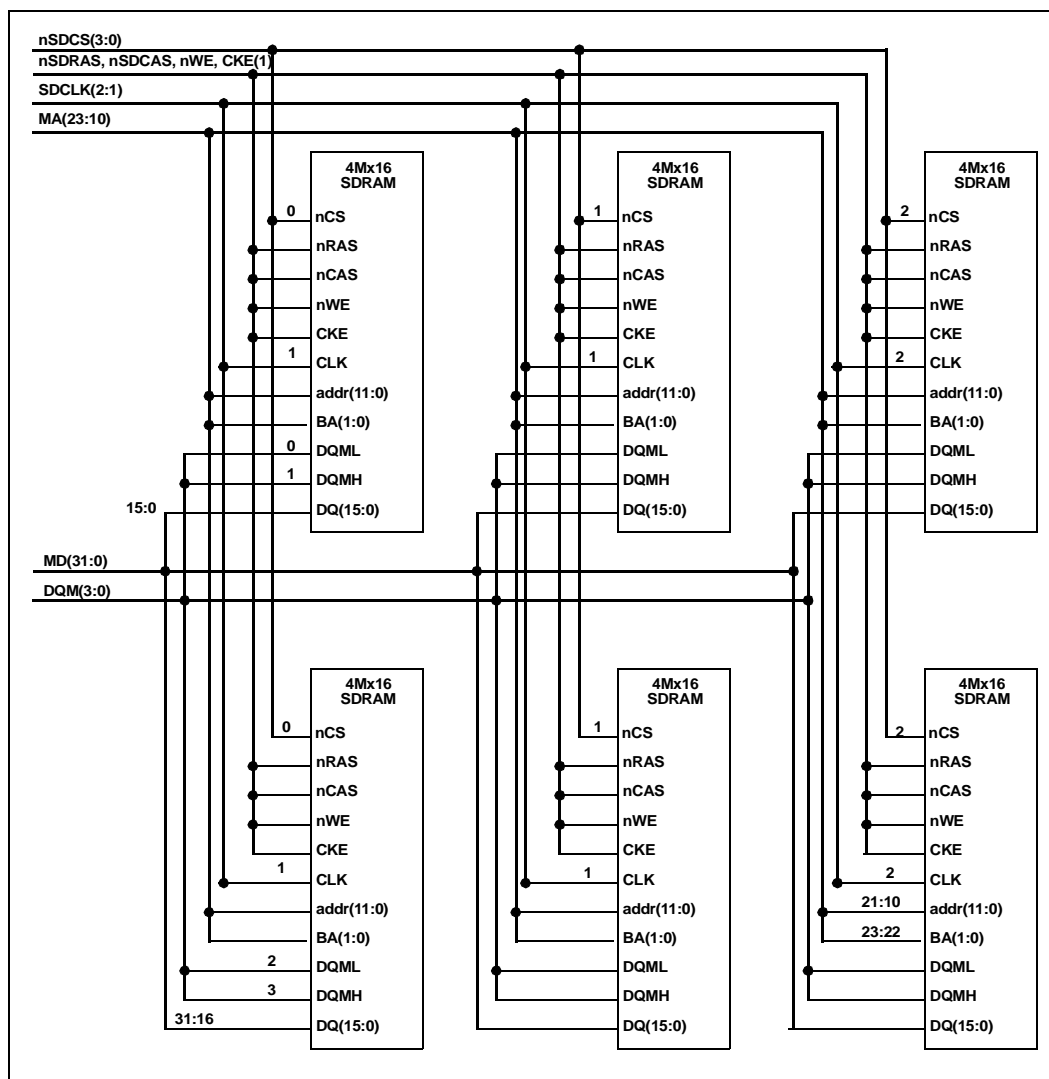
The processor supports an SDRAM interface at a maximum frequency of 100 MHz. The SDRAM Interface supports four 16-bit or 32-bit wide partitions of SDRAM. Each partition is allocated 64 MBytes of the internal memory map. However, the actual size of each partition is dependent on the particular SDRAM configuration used. The four partitions are divided into two partition pairs: the 0/1 pair and the 2/3 pair. Both partitions within a pair (for example, partition 0 and partition 1) must be identical in size and configuration; however, the two pairs can be different. For example, the 0/1 pair can be 100 MHz SDRAM on a 32-bit data bus, while the 2/3 pair can be 50 MHz SDRAM on a 16-bit data bus.

Note: For proper SDRAM operation above 50 MHz, 22 ohm series resistors should be placed on the memory address lines. System designers must run the appropriate high speed design simulations to determine if a different value resistor is needed.

2.3 SDRAM memory wiring diagram

Figure 2-2, “SDRAM Memory System Example” on page 2-4 is a wiring diagram example that shows a system using 1Mword x 16-bit x 4-bank SDRAM devices for a total of 48 Mbytes. Refer to Section 2.5, “SDRAM Address Mapping” on page 2-6 to determine the individual SDRAM component address.

Figure 2-2. SDRAM Memory System Example



2.4 SDRAM Support

Table 2-2 shows the SDRAM memory types and densities that are supported by the processor.

Table 2-2. SDRAM Memory Types Supported by the Processor

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

2.5 SDRAM Address Mapping

SDRAM Address Mapping is shown in Table 2-3 and Table 2-4.

Table 2-3. Normal Mode Memory Address 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	Technology		A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
1Mx16	16Mbit	1x11x8				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2Mx8	16Mbit	1x11x9				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
4Mx4	16Mbit	1x11x10				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1x12x8			BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1x12x9			BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1x12x10			BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1x12x11			BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1x13x8		BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1x13x9		BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1x13x10		BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1x13x11		BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2Mx32	64Mbit	2x11x8			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		2x11x9			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		2x11x10			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
4Mx16/4Mx32	64Mbit/128Mbit	2x12x8		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8Mx8/8Mx16	64Mbit/128Mbit	2x12x9		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16Mx4/16Mx8	64Mbit/128Mbit	2x12x10		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
32Mx4	128Mbit	2x12x11		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8Mx32	256Mbit	2x13x8	BS1	BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16MX16	256Mbit	2x13x9	BS1	BS0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Table 2-4. Processor Compatibility Mode Address Line 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	Technology		A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	
1Mx16	16Mbit	1x11x8				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
2Mx8	16Mbit	1x11x9				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
4Mx4	16Mbit	1x11x10				BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		1x12x8			A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		1x12x9			A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		1x12x10			A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		1x12x11			A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		1x13x8		A12	A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		1x13x9		A12	A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		1x13x10		A12	A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		1x13x11		A12	A11	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
2Mx32	64Mbit	2x11x8			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		2x11x9			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
		2x11x10			BS1	BS0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
4Mx16/4Mx32	64Mbit/128Mbit	2x12x8		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
8Mx8/8Mx16	64Mbit/128Mbit	2x12x9		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
16Mx4/16Mx8	64Mbit/128Mbit	2x12x10		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
32Mx4	128Mbit	2x12x11		BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
8Mx32	256Mbit	2x13x8	A12	BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
16Mx16	256Mbit	2x13x9	A12	BS1	BS0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	

2.6 Static Memory

2.6.1 Overview

The processor external memory bus interface supports the following static memory types:

- Synchronous and asynchronous Burst mode and Page mode Flash
- Synchronous Mask ROM (SMROM)
- Page Mode ROM
- SRAM
- SRAM-like Variable Latency I/O (VLIO)
- PCMCIA expansion memory
- Compact Flash

Memory types are programmable through the memory interface configuration registers.

Six chip selects control the static memory interface, nCS<5:0>. All are configurable for nonburst 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 that it allows the data ready input signal (RDY) to insert a variable number of memory-cycle wait states. The data bus width for each chip select region may be programmed to be 16-bit or 32-bit. nCS<3:0> are also configurable for Synchronous Static Memory.

Note: Chip selects CS[5:1] should have external 10K pull up resistors.

For SRAM and variable latency I/O implementations, DQM<3:0> signals are used for the write byte enables, where DQM<3> corresponds to the MSB. The processor supplies 26-bits of byte address for access of up to 64 Mbytes per chip select. However, when the address is sent out on the MA pins, MA reflects the actual address, not the byte address. The lower one or two internal address bits are truncated appropriately.

2.6.2 Boot Time Defaults

Table 2-5 shows valid booting configurations for the PXA255 processor, while Table 2-6 shows boot selection definitions. See Section 6.10.2 “Boot Time Defaults” in the *Intel® PXA255 Processor Developer’s Manual* for more detailed descriptions of these Boot Time Configurations.

Table 2-5. Valid Booting Configurations Based for the PXA255 processor

Valid Booting Configurations
000
001
100
101
110
111

Table 2-6. BOOT_SEL Definitions

BOOT_SEL			Boot From . . .
2	1	0	
0	0	0	Asynchronous 32-bit ROM
0	0	1	Asynchronous 16-bit ROM
1	0	0	1 32-bit Synchronous Mask ROM (64 Mbits) 2 16-bit Synchronous Mask ROMs = 32-bits (32 Mbits each)
1	0	1	1 16-bit Synchronous Mask ROM (64 Mbits)
1	1	0	2 16-bit Synchronous Mask ROMs = 32-bits (64 Mbits each)
1	1	1	1 16-bit Synchronous Mask ROM (32 Mbits)

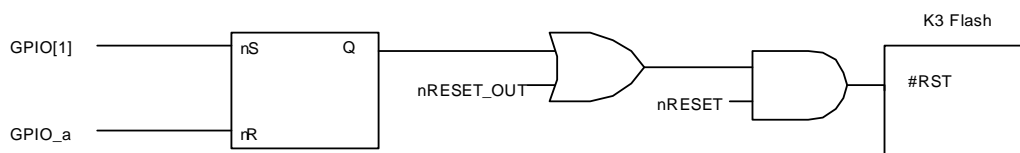
2.6.3 Flash Memory

The PXA255 processor supports Flash memory on all six static chip selects and synchronous operation on nCS[3:0]. If synchronous Flash memory is used in the system, the traces for SDCLK0, address, data and control should be limited to 3 inches. If the traces must exceed 3 inches, 47 ohm series termination resistors must be included.

2.6.3.1 Synchronous Intel® StrataFlash® Memory Reset

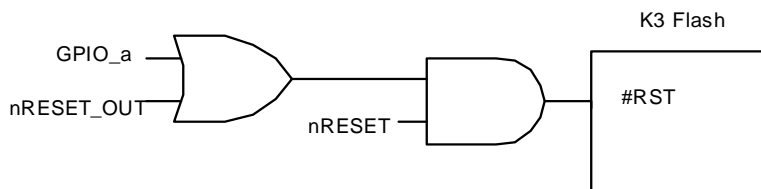
The PXA250 nRESET_OUT pin must be connected to the K3 #RST pin for Hardware reset, Watchdog reset and sleep mode to work properly. GPIO reset however does not reset the contents of the memory controller configuration register. If GPIO reset operation is required, a state machine is necessary between nRESET, nRESET_OUT, GPIO[1], and #RST to ensure that #RST is asserted during Hardware reset, Watchdog reset, and sleep mode, and not asserted during GPIO reset. [Figure 2-3, “Flash Memory Reset Using State Machine” on page 2-9](#) shows the required logic. GPIO_a is an unused GPIO that is driven low by software during the initialization sequence and left high during normal operation. After this is completed, then enable GPIO Reset.

Figure 2-3. Flash Memory Reset Using State Machine



If Watchdog reset is not necessary, a secondary GPIO can control nRESET_OUT using the equation $\#RST = nRESET \& (nRESET_OUT \mid GPIO_a)$. This allows sleep mode entry to reset the flash memory while keeping it in synchronous mode during a GPIO reset. [Figure 2-4, “Flash Memory Reset Logic if Watchdog Reset is Not Necessary” on page 2-9](#) shows the required logic. GPIO_a is an unused GPIO that is kept high during normal operation and driven low before sleep mode entry.

Figure 2-4. Flash Memory Reset Logic if Watchdog Reset is Not Necessary



2.6.4 SRAM / ROM / Flash / Synchronous Fast Flash Memory Options

[Table 2-7](#) contains the AC specification for SRAM / ROM / Flash / Synchronous Fast Flash.

Table 2-7. SRAM / ROM / Flash / Synchronous Fast Flash AC Specifications

Symbol	Description	MEMCLK					Units Notes
		99.5	118.0	132.7	147.5	165.9	
SRAM / ROM / Flash / Synchronous Fast Flash (WRITES) (Asynchronous)							
tromAS	MA(25:0) setup to nOE, nSDCAS (as nADV) asserted	10	8.5	7.5	6.8	6	ns, 1
tromAH	MA(25:0) hold after nCS, nOE, nSDCAS (as nADV) de-asserted	10	8.5	7.5	6.8	6	ns, 1
tromASW	MA(25:0) setup to nWE asserted	30	25.5	22.5	20.4	18	ns, 3
tromAHW	MA(25:0) hold after nWE de-asserted	10	8.5	7.5	6.8	6	ns, 1
tromCES	nCS setup to nWE asserted	20	17	15	13.6	12	ns, 2
tromCEH	nCS hold after nWE de-asserted	10	8.5	7.5	6.8	6	ns, 1
tromDS	MD(31:0), DQM(3:0) write data setup to nWE asserted	10	8.5	7.5	6.8	6	ns, 1
tromDSWH	MD(31:0), DQM(3:0) write data setup to nWE de-asserted	20	17	15	13.6	12	ns, 2
tromDH	MD(31:0), DQM(3:0) write data hold after nWE de-asserted	10	8.5	7.5	6.8	6	ns, 1
tromNWE	nWE high time between beats of write data	20	17	15	13.6	12	ns, 2

NOTES:

1. This number represents 1 MEMCLK period
2. This number represents 2 MEMCLK periods

2.6.5 Variable Latency I/O Interface Overview

Both reads and writes for VLIO differ from SRAM in that the PXA255 processor samples the data-ready input, RDY. The RDY signal is level sensitive and goes through a two-stage synchronizer on input. When the internal RDY signal is high, the I/O device is ready for data transfer. This means that for a transaction to complete at the minimum assertion time for either nOE or nPWE (RDF+1), the RDY signal must be high two clocks prior to the minimum assertion time for either nOE or nPWE (RDF-1). Data will be latched on the rising edge of MEMCLK once the internal RDY signal is high and the minimum assertion time of RDF+1 has been reached. Once the data has been latched, the address may change on the next rising edge of MEMCLK or any cycles thereafter. The nOE or nPWE signal de-asserts one MEMCLK after data is latched. Before a subsequent data beat, nOE or nPWE remains deasserted 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 deassertion. Refer to [Figure 2-5](#) for 32-Bit Variable Latency I/O read timing and [Figure 2-8](#) for Variable Latency I/O Interface AC Specifications

Figure 2-5. 32-Bit Variable Latency I/O Read Timing (Burst-of-Four, One Wait Cycle Per Beat)

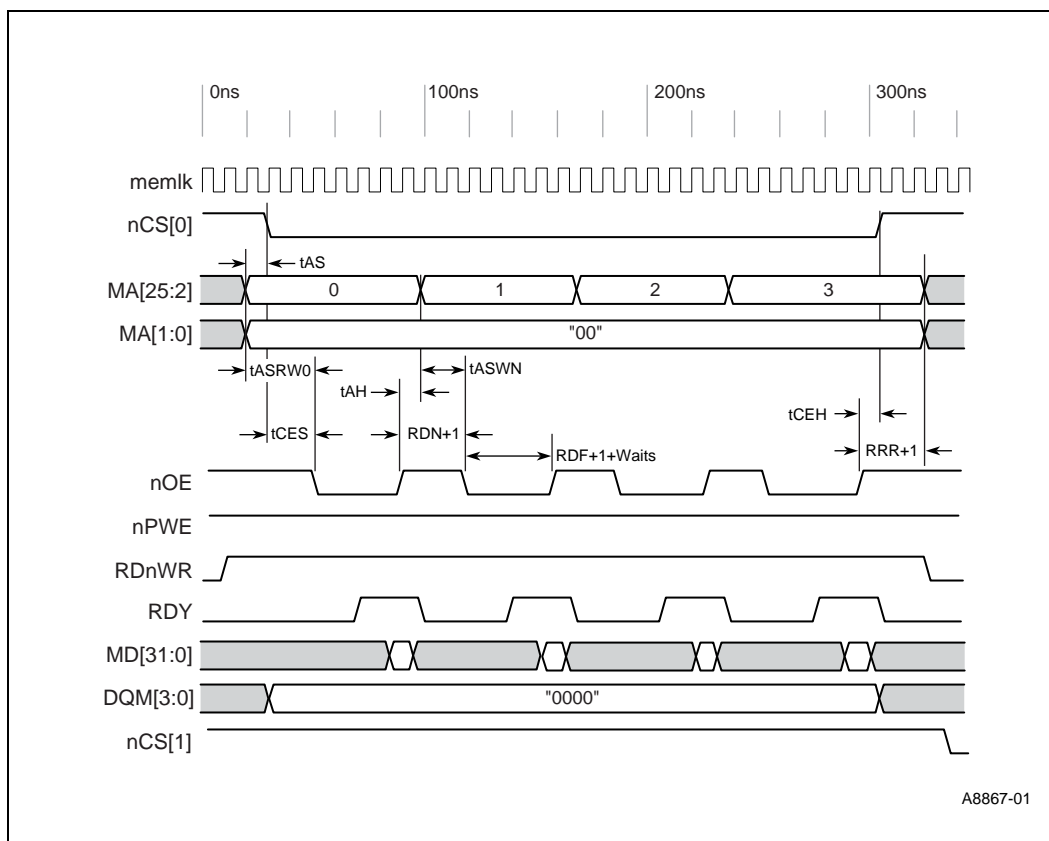


Table 2-8. Variable Latency I/O Interface AC Specifications (Sheet 1 of 2)

Symbol	Description	MEMCKLK					Units Notes
		99.5	118.0	132.7	147.5	165.9	
Variable Latency IO Interface (VLIO) (Asynchronous)							
tvlioAS	MA(25:0) setup to nCS asserted	10	8.5	7.5	6.8	6	ns, 1
tvlioASRW	MA(25:0) setup to nOE or nPWE asserted	10	8.5	7.5	6.8	6	ns, 1
tvlioAH	MA(25:0) hold after nOE or nPWE de-asserted	10	8.5	7.5	6.8	6	ns, 1
tvlioCES	nCS setup to nOE or nPWE asserted	20	17	15	13.6	12	ns, 2
tvlioCEH	nCS hold after nOE or nPWE de-asserted	10	8.5	7.5	6.8	6	ns, 1
tvlioDSW	MD(31:0), DQM(3:0) write data setup to nPWE asserted	10	8.5	7.5	6.8	6	ns, 1
tvlioDSWH	MD(31:0), DQM(3:0) write data setup to nPWE de-asserted	20	17	15	13.6	12	ns, 2
tvlioDHW	MD(31:0), DQM(3:0) hold after nPWE de-asserted	10	8.5	7.5	6.8	6	ns, 1

Table 2-8. Variable Latency I/O Interface AC Specifications (Sheet 2 of 2)

Symbol	Description	MEMCLK					Units Notes
		99.5	118.0	132.7	147.5	165.9	
tvlioDHR	MD(31:0) read data hold after nOE de-asserted	0	0	0	0	0	ns
tvlioRDYH	RDY hold after nOE, nPWE de-asserted	0	0	0	0	0	ns
tvlioNPWE	nPWE, nOE high time between beats of write or read data	20	17	15	13.6	12	ns, 2

NOTES:

1. This number represents 1 MEMCLK period
2. This number represents 2 MEMCLK periods

2.6.6 External Logic for PCMCIA Implementation

The PXA255 processor requires external glue logic to complete the PCMCIA socket interface. Figure 2-6, “Expansion Card External Logic for a Two-Socket Configuration” on page 2-13 and Figure 2-7, “Expansion Card External Logic for a One-Socket Configuration” on page 2-14 show general solutions for one and two socket configurations. Use GPIO or memory-mapped external registers to control the PCMCIA interface’s reset, power selection (V_{CC} and V_{pp}), and drive enables. These diagrams show the logical connections necessary to support hot insertion capability. For dual-voltage support, level shifting buffers are required for all the processor input signals. Hot insertion capability requires each socket be electrically isolated from the other and from the remainder of the memory system. If one or both of these features are not required, you may eliminate some of the logic shown in these diagrams. The processor allows either 1-socket or 2-socket solutions. In the 1-socket solution, only minimal glue logic is required (typically for the data transceivers, address buffers, and level shifting buffers.) To achieve this some of the signals are routed through dual-duty GPIO pins. The nOE of the transceivers is driven through the PSKTSEL pin, which is not needed in the one-socket solution. The DIR pin of the transceiver is driven through the RDnWR pin. A GPIO is used for the three-state signal of the address and nPWE lines. These signals are used for memories other than the card interface and must be three-stated.

Note: For 2.5 V VCCN, 5 V to 2.5 V level shifters are required.

In the 2-socket solution, all pins assume their normal duties and glue logic is necessary for proper operation of the system. The pull-ups shown are included for compliance with *PC Card Standard - Volume 2 - Electrical Specification*. Remove power from these pull-ups during sleep to avoid unnecessary power consumption. Refer to Table 2-9 for the PCMCIA or compact Flash card interface AC specifications.

Figure 2-6. Expansion Card External Logic for a Two-Socket Configuration

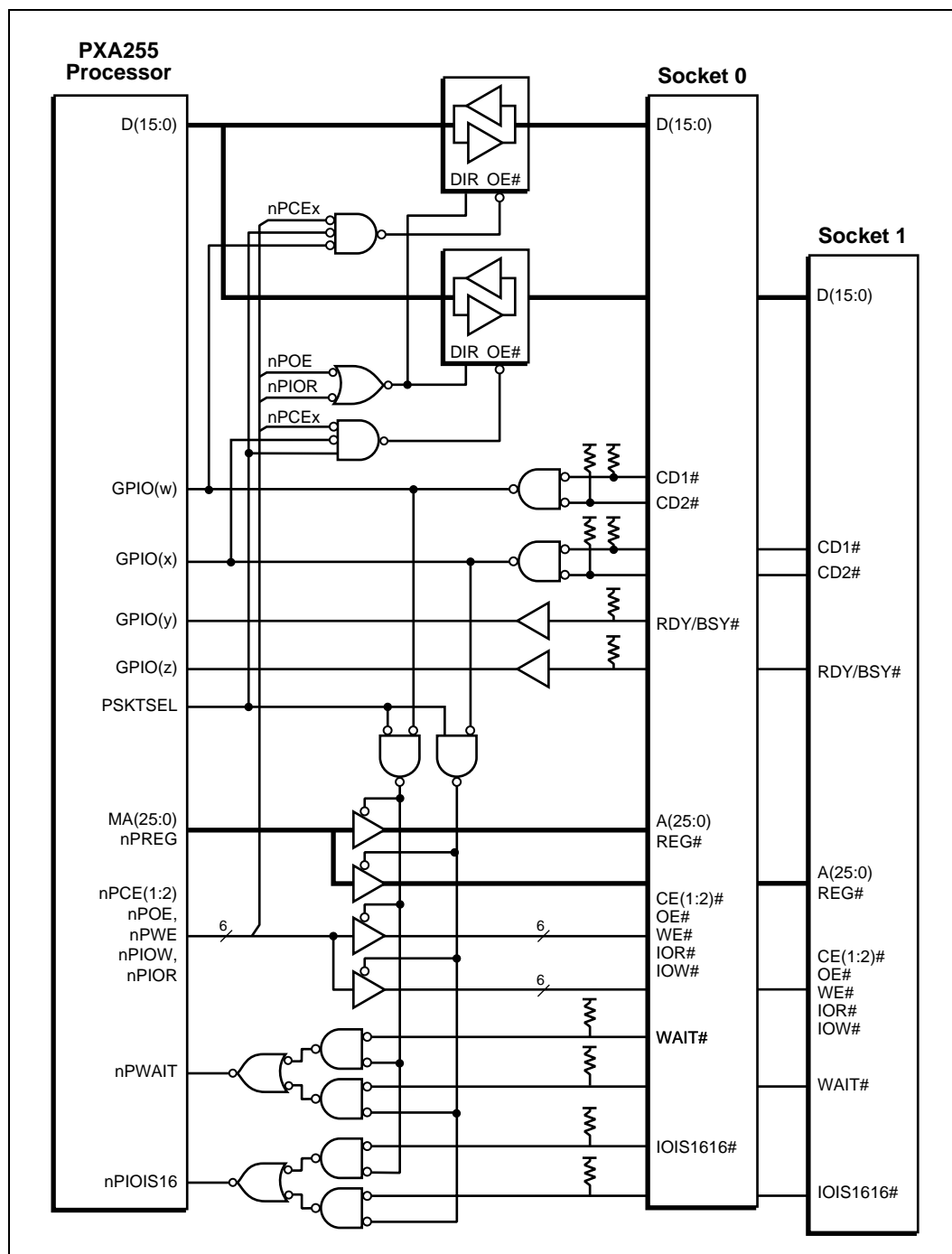


Figure 2-7. Expansion Card External Logic for a One-Socket Configuration

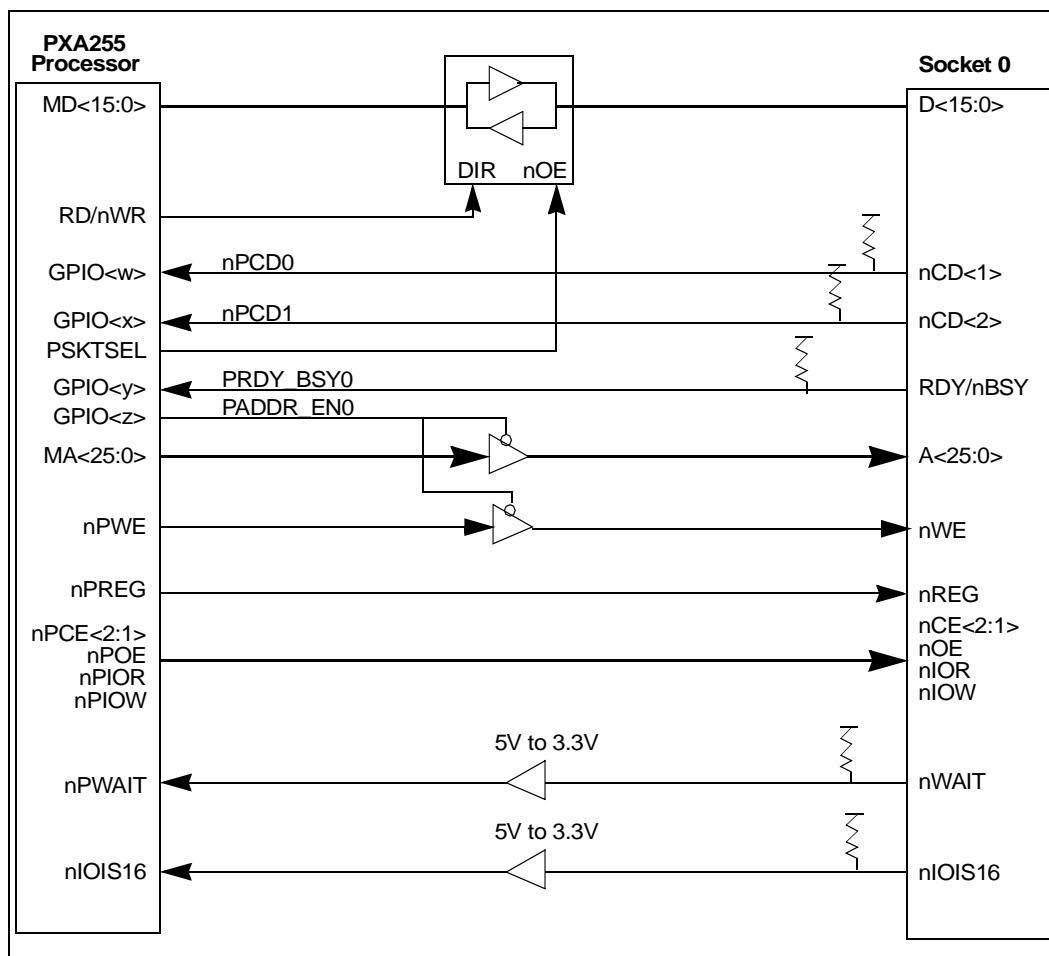


Table 2-9. Card Interface (PCMCIA or Compact Flash) AC Specifications (Sheet 1 of 2)

Symbol	Description	MEMCKLK					Units Notes
		99.5	118.0	132.7	147.5	165.9	
Card Interface (PCMCIA or Compact Flash) (Asynchronous)							
tcardAS	MA(25:0), nPREG, PSKTSEL, nPCE setup to nPWE, nPOE, nPIOW, or nPIOR asserted	20	17	15	13.6	12	ns, 1
tcardAH	MA(25:0), nPREG, PSKTSEL, nPCE hold after nPWE, nPOE, nPIOW, or nPIOR de-asserted	10	8.5	7.5	6.8	6	ns, 1
tcardDS	MD(31:0) setup to nPWE, nPOE, nPIOW, or NPIOR asserted	10	8.5	7.5	6.8	6	ns, 1

Table 2-9. Card Interface (PCMCIA or Compact Flash) AC Specifications (Sheet 2 of 2)

Symbol	Description	MEMCKLK					Units Notes
		99.5	118.0	132.7	147.5	165.9	
tcardDH	MD(31:0) hold after nPWE, nPOE, nPIOW, or nPIOR de-asserted	10	8.5	7.5	6.8	6	ns, 1
tcardCMD	nPWE, nPOE, nPIOW, or nPIOR command assertion	30	25.5	22.5	20.4	18	ns, 1

NOTE:

1. These numbers are minmums. They can be much larger based on the programmable Card Interface timing registers.

2.6.7 DMA / Companion Chip Interface

Connect a companion chip to the processor via:

- Alternate Bus Master Mode
- Variable Latency I/O
- Flow through DMA

These connections are illustrated in [Figure 2-8](#) and [Figure 2-9](#).

Figure 2-8. Alternate Bus Master Mode

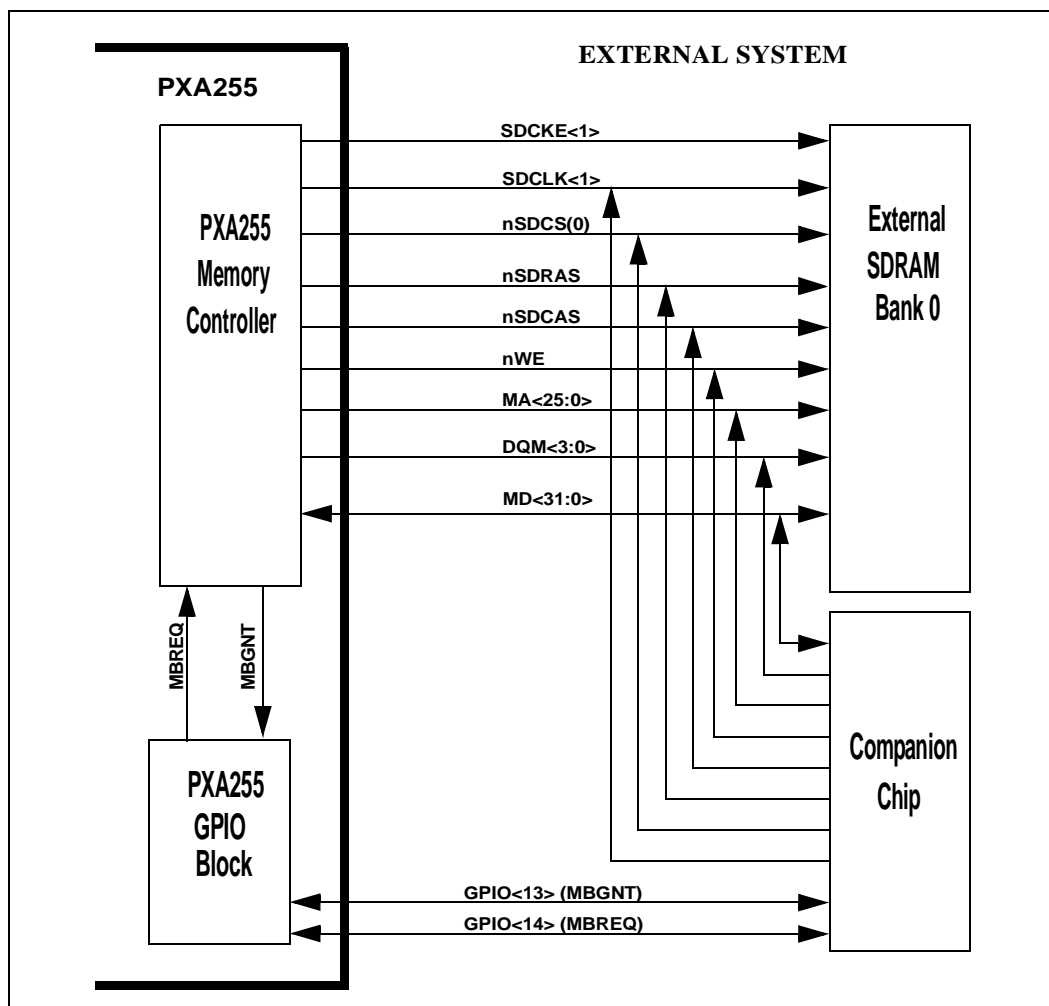
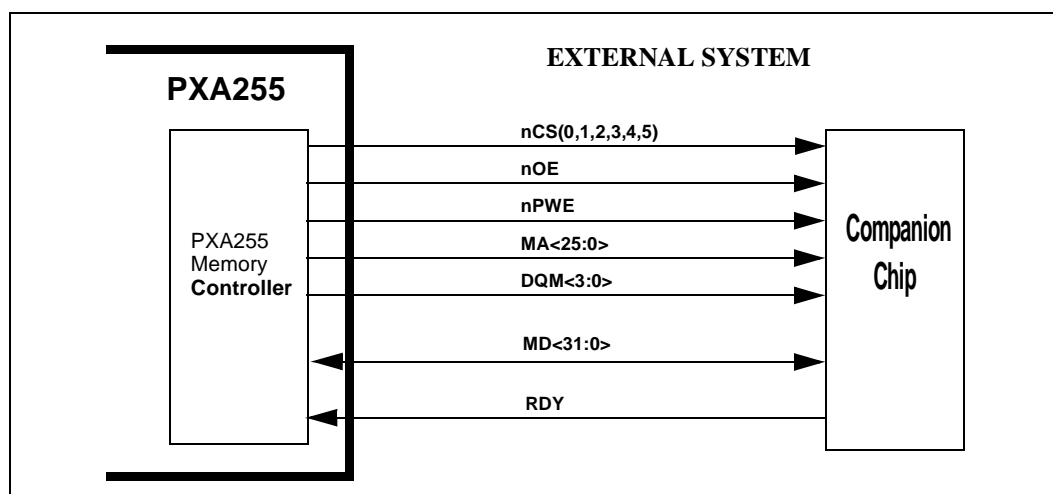


Figure 2-9. Variable Latency I/O



2.7 System Memory Layout Guidelines

2.7.1 System Memory Topologies (Min and Max Simulated Loading)

Figure 2-10, Figure 2-11, Figure 2-12, and Figure 2-13 are the topologies that were simulated to develop the trace length recommendations in Section 2.7.2. These topologies are for reference only.

Figure 2-10. CS, CKE, DQM, CLK, MA minimum loading topology

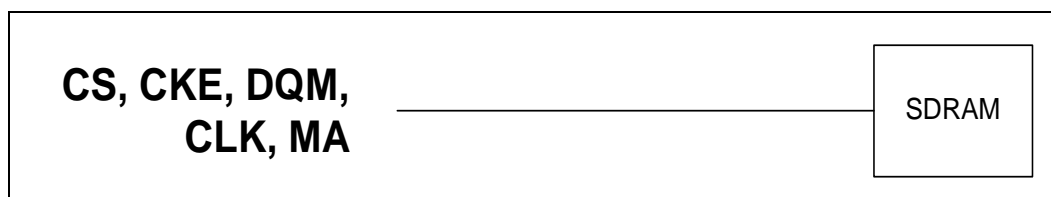


Figure 2-11. CS, CKE, DQM, CLK, MA Maximum Loading Topology

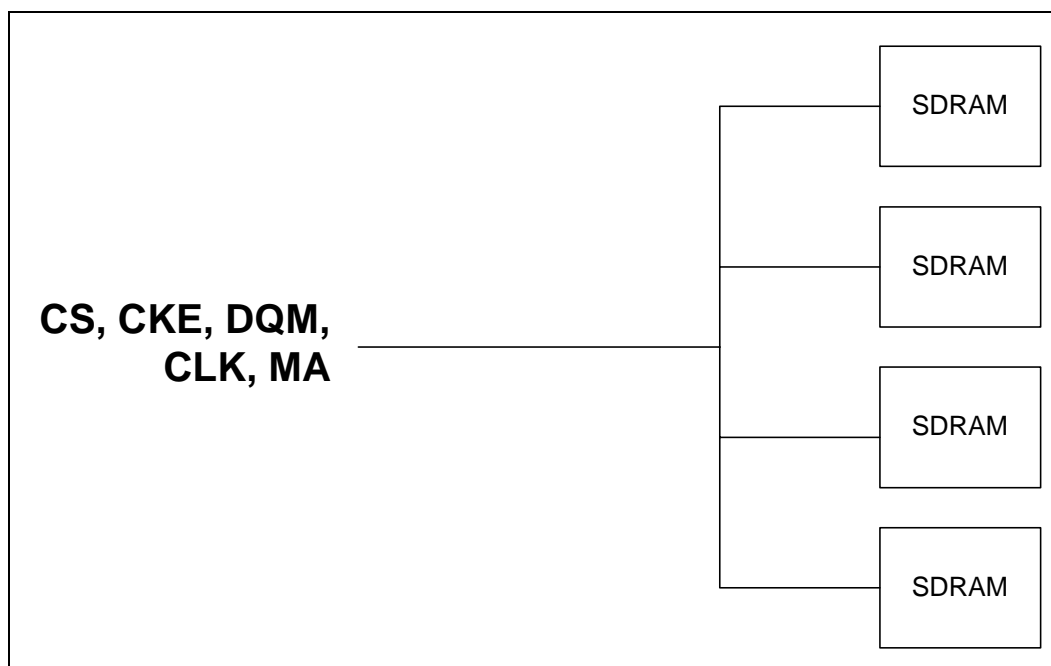


Figure 2-12. MD Minimum Loading Topology

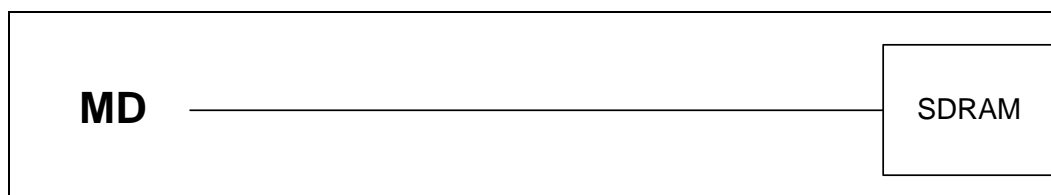
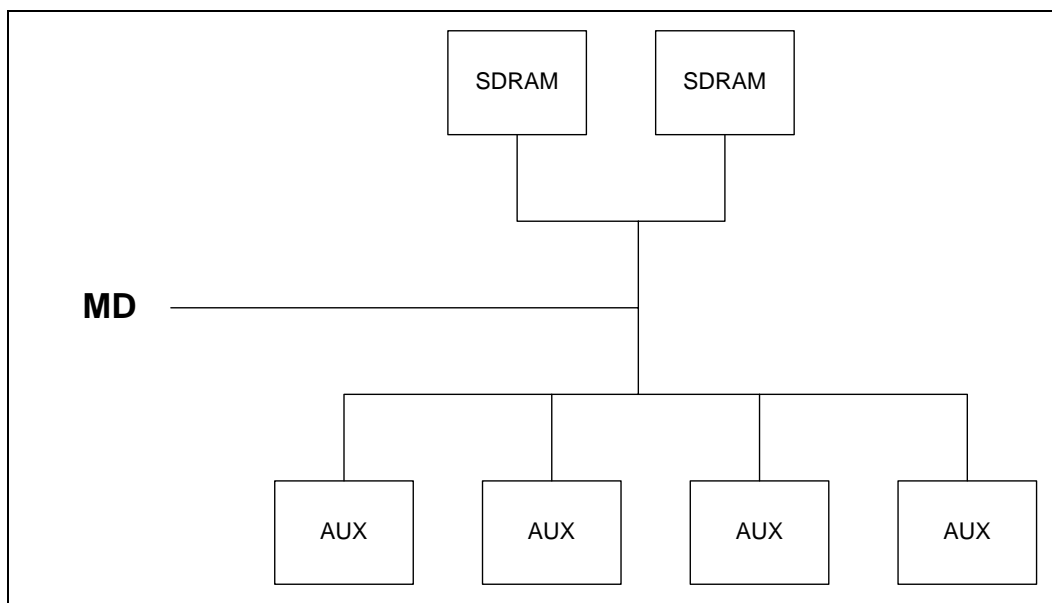


Figure 2-13. MD maximum loading topology



2.7.2 System Memory Recommended Trace Lengths

Table 2-10 details the minimum and maximum trace lengths that were simulated for the processor. These trace lengths are not the absolute trace lengths that will work given the loading conditions. The trace lengths in Table 2-10 are measured from the processor to the individual component pins. The board impedance for the simulations was 60ohm +/- 10%.

Table 2-10. Minimum and Maximum Trace Lengths for the SDRAM Signals

Signal	Min Trace Length	Max Trace Length
CS, CKE, DQM	0.75 in	4.5 in
CLK	1.0 in	4.25 in
MA	1.0 in	4.5 in
MD	1.0 in	4.25 in

This chapter describes sample hardware connections from the PXA255 processor to various types of LCD controllers. Active (TFT) as well as passive (DSTN) displays are discussed as well as single and dual panel displays. These should not be considered the only possible ways to connect an LCD panel to the PXA255 processor, but should serve as a reference to assist with hardware design considerations. Other panels, for example panels without L_FCLK or L_LCLK, have been successfully connected to the PXA255 processor.

3.1 LCD Display Overview

The PXA255 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.

Note: Names used for “LCD Panel Pin” are representative names and may not match those on all LCD panels. Refer to the LCD panel reference documentation for the actual name.

3.2 Passive (DSTN) Displays

Several different types of passive displays are available in both color and monochrome. These may be single or dual panel 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® PXA255 Processor Developer's 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.

Table 3-1 describes the number of L_DD pins required for the various types of passive displays, as well as which LCD data pins are used for which panel (upper or lower).

Table 3-1. LCD Controller Data Pin Utilization (Sheet 1 of 2)

Color/ Monochrome Panel	Single/ Dual Panel	Double-Pixel Mode	Screen Portion	Pins
Monochrome	Single	No	Whole	L_DD<3:0>
Monochrome	Single	Yes	Whole	L_DD<7:0> ¹
Monochrome	Dual	No	Top	L_DD<3:0>
			Bottom	L_DD<7:4>
Color	Single	N/A	Whole	L_DD<7:0>

Table 3-1. LCD Controller Data Pin Utilization (Sheet 2 of 2)

Color/ Monochrome Panel	Single/ Dual Panel	Double-Pixel Mode	Screen Portion	Pins
Color	Dual	N/A	Top	L_DD<7:0>
			Bottom	L_DD<15:8>

NOTE: 1. Double pixel data mode (DPD) = 1.

For passive displays, the pins described in Table 3-2 are required connections between the PXA255 processor and your LCD panel.

Table 3-2. Passive Display Pins Required

PXA255 processor Pin	LCD Panel Pin	Pin Type ¹	Definition
L_DD	DU_x, DL_x	Output	Data lines used to transmit either four or eight data values at a time to the LCD display. For monochrome displays, each pin value represents a single pixel; for passive color, groupings of three pin values represent one pixel (red, green, and blue data values). Either the bottom four pins (L_DD<3:0>), the bottom 8 pins (L_DD<7:0>) or all 16 pixel data pins (L_DD<15:0>) will be used as shown in Table 3-1
L_PCLK	Pixel_Clock	Output	Pixel Clock - used by the LCD display to clock the pixel data into the line shift register.
L_LCLK	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	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 PXA255 processor).

NOTES:

1. "Pin Type" is in reference to the PXA255 processor. Therefore, outputs are pins that drive a signal from the processor to another device.
2. Vcon is a signal external to the PXA255 processor. Please refer to "Contrast Voltage" on page 8

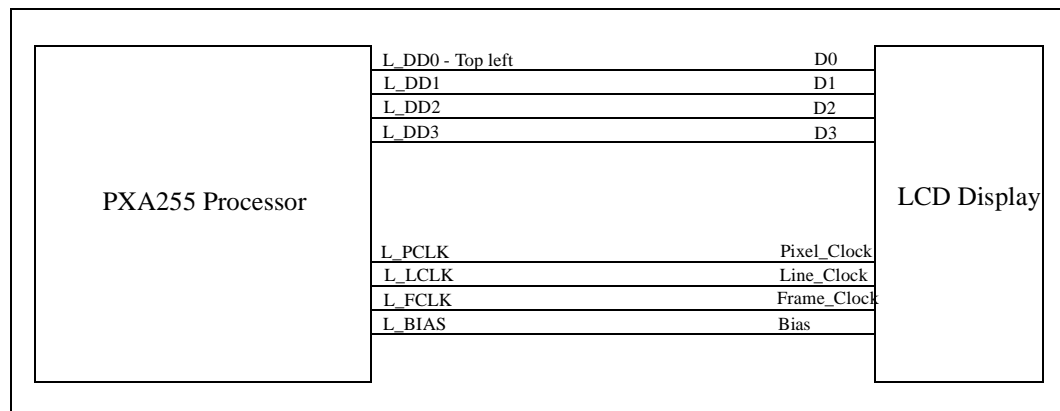
3.2.1 Typical Connections for Passive Panel Displays

The following diagrams are typical connections and serve as a guide for designing systems which contain passive LCD displays. Panels differ on which is the panel's least significant bit (Refer to the LCD panel reference documentation for the least significant bit.) Each figure indicates the top-left pixel (1,1) bit. Dual panels indicate the top-left pixel (1,n/2) of the upper and lower panels and color passive panels show the top-left-pixel color bits.

3.2.1.1 Passive Monochrome Single Panel Displays

Figure 3-1 is a typical single-panel monochrome passive display connection.

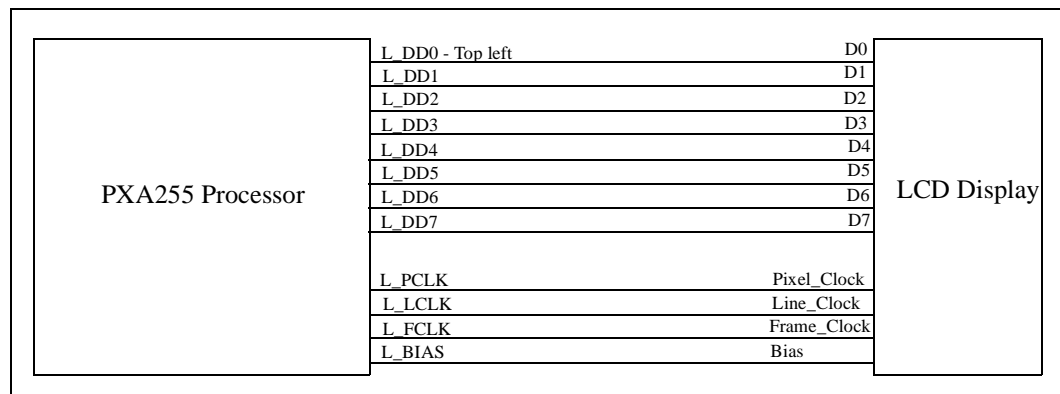
Figure 3-1. Single Panel Monochrome Passive Display Typical Connection



3.2.1.2 Passive Monochrome Single Panel Displays, Double-Pixel Data

Figure 3-2 shows typical connections for a single-panel monochrome passive display using double-pixel data mode.

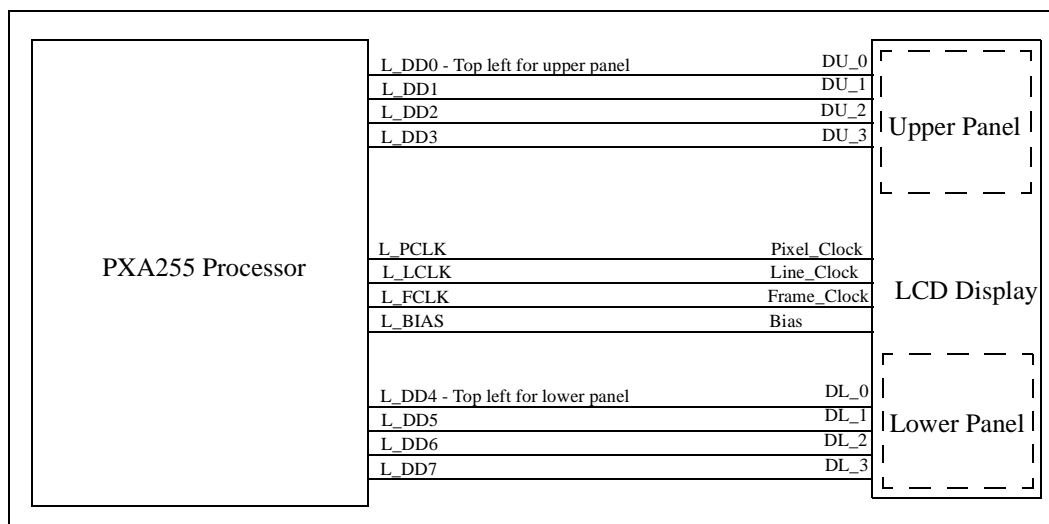
Figure 3-2. Passive Monochrome Single Panel Displays, Double-Pixel Data Typical Connection



3.2.1.3 Passive Monochrome Dual Panel Displays

Figure 3-3 is a typical dual-panel monochrome passive display connection.

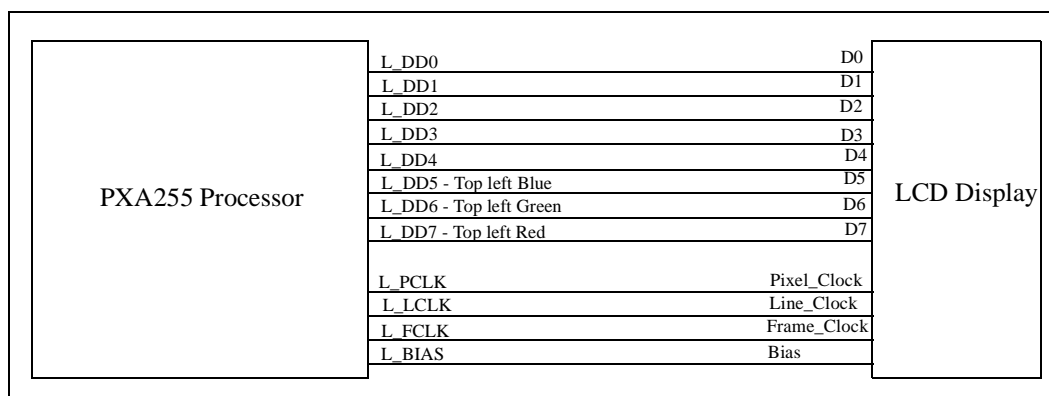
Figure 3-3. Passive Monochrome Dual Panel Displays Typical Connection



3.2.1.4 Passive Color Single Panel Displays

Figure 3-4 is a typical single-panel color passive display connection.

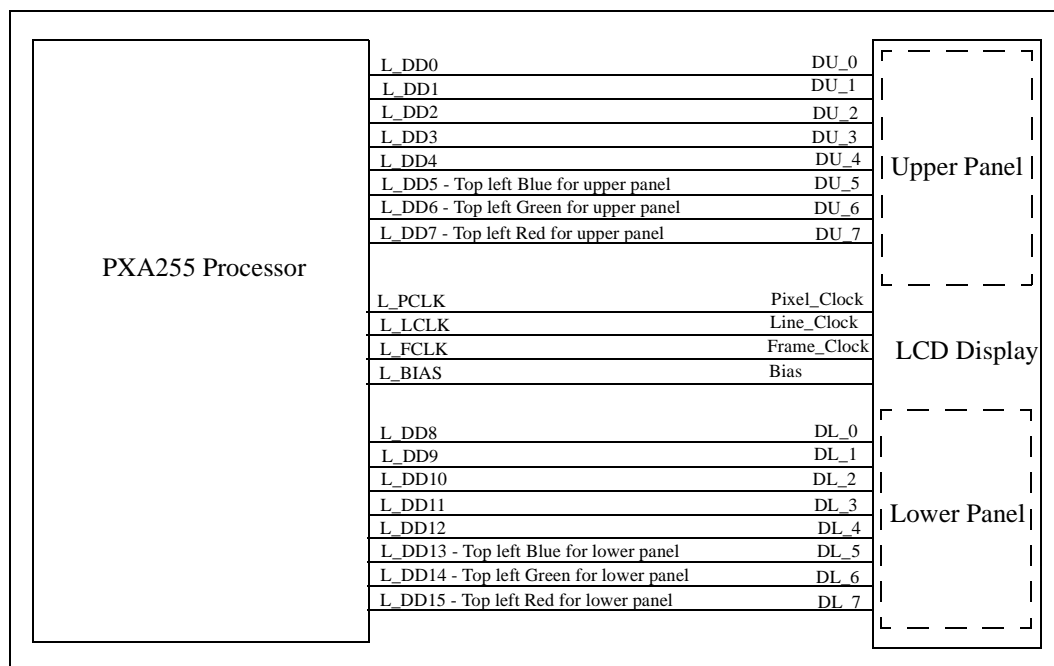
Figure 3-4. Passive Color Single Panel Displays Typical Connection



3.2.1.5 Passive Color Dual Panel Displays

Figure 3-5 is a typical dual-panel color passive display connection.

Figure 3-5. Passive Color Dual Panel Displays Typical Connection



3.3 Active (TFT) Displays

Because data is sent to the panel as raw 16-bit pixel data, active displays require 16 data pins in order to transfer the pixel data from the controller. All 16 data lines are also required to drive one pixel value. The 16 bits of data describe the intensity level of the red, green, and blue for each pixel. Typically, this is formatted as 5 bits for red, 6 bits for green, and 5 bits for blue, but this can vary by display and is controlled by the software writing to the frame buffer. Refer to the display datasheet to ensure that the correct the PXA255 processor LCD data lines are connected to the correct LCD panel data lines.

Many active displays actually have more than 16 data lines - usually 18 (6 of each color). For these panels it is recommended that the most significant lines of the panel lines are connected to the data lines from the PXA255 processor. This maintains the panel's full range of colors but increases the granularity of the color spectrum with an insufficient number of data lines. All unused panel data lines can be tied either high or low. Other options include tying the LSB of red and blue to the next bit, R1 or B1.

For active displays, connect the pins described in [Table 3-3](#) between the PXA255 processor and the LCD panel.

Table 3-3. Active Display Pins Required

PXA255 processor Pin	LCD Panel Pin	Pin Type ¹	Definition
L_DD<15:0>	R<4:0>,G<5:0>, B<4:0>	Output	Data lines used to transmit the 16 bit data values to the LCD display.
L_PCLK	Clock	Output	Pixel Clock - used by the LCD display to clock the pixel data into the line shift register. In active mode this clock transitions constantly.
L_LCLK	Horizontal Sync	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. Also signals the panel to start a new line.
L_FCLK	Vertical Sync	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	DE (Data Enable)	Output	AC biases used in active mode as a data enable signal when data should be latched by the pixel clock from the data lines.
N/A	Vcon ²	N/A	Contrast Voltage - Adjustable voltage input to LCD panel - external voltage circuitry is required (no pin available on the PXA255 processor).

NOTES:

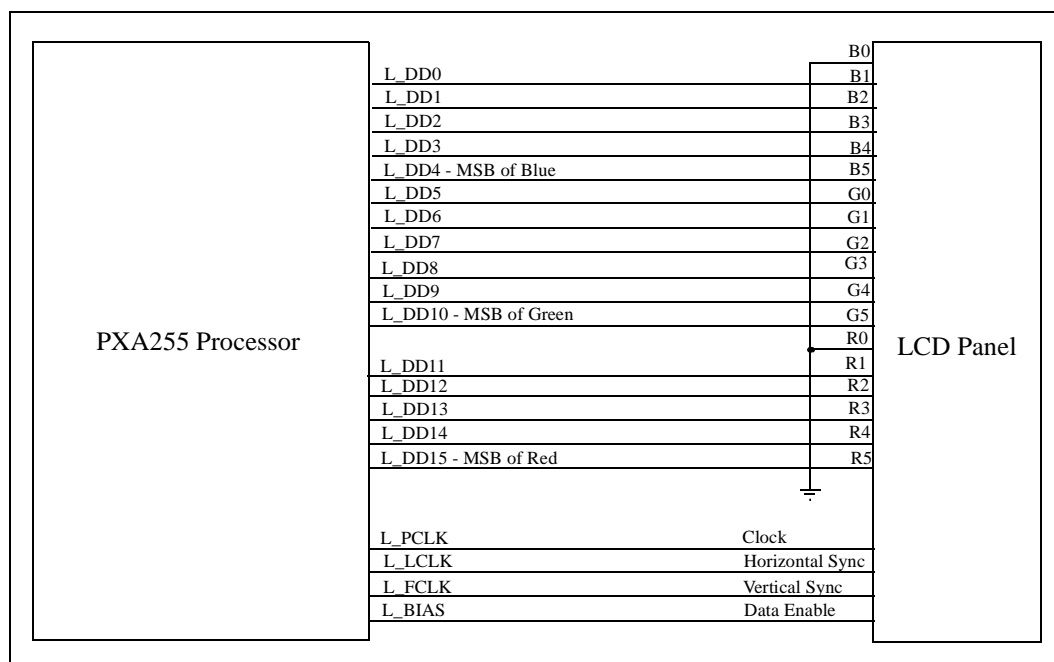
1. In reference to the PXA255 processor. Therefore, outputs are pins that drive a signal from the PXA255 processor to another device.
2. Vcon is a signal external to the PXA255 processor. Please refer to [Section 3.5.1, "Contrast Voltage" on page 8](#).

3.3.1 Typical connections for Active Panel Displays

Figure 3-6, “Active Color Display Typical Connection” on page 7 shows a typical connection for an active panel display and should serve as a guide for designing systems which contain active LCD displays. The MSB of each color is indicated. The panel is 18-bit, with the LSB of red and blue tied to ground.

Note: This example shows 6 red, 6 green and 6 blue bits on the LCD panel. However, different active display panels might have more or different data lines. Consult the LCD panel manufacturer's datasheet for the actual data lines.

Figure 3-6. Active Color Display Typical Connection



3.4 PXA255 processor Pinout

Table 3-4 describes the ball positions for the LCD controller on the PXA255 processor.

Table 3-4. PXA255 processor LCD Controller Ball Positions (Sheet 1 of 2)

Pin Name	Ball Position
L_DD0	E7
L_DD1	D7
L_DD2	C7
L_DD3	B7
L_DD4	E6
L_DD5	D6
L_DD6	E5
L_DD7	A6
L_DD8	C5
L_DD9	A5
L_DD10	D5
L_DD11	A4
L_DD12	A3

Table 3-4. PXA255 processor LCD Controller Ball Positions (Sheet 2 of 2)

Pin Name	Ball Position
L_DD13	A2
L_DD14	C3
L_DD15	B3
L_FCLK	E8
L_LCLK	D8
L_PCLK	B8
Bias	A8

3.5 Additional Design Considerations

3.5.1 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 via a voltage source on the system board. The contrast voltage is adjusted via a variable resistor on the circuit board.

The required voltage range and current capabilities vary between panel manufacturers. Consult the datasheet for your panel to determine the variable voltage circuit design. Ensure that the contrast voltage is stable, otherwise visual artifacts might result. Possible contrast voltage circuits are often suggested by the panel manufacturers.

3.5.2 Backlight Inverter

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

- Use a shielded backlight inverter
- Physically locate the inverter as far away from the LCD data lines and system board as possible, usually located with the LCD panel

If power consumption is an issue, chose a backlight inverter that can be disabled through software. This lets you save power by automatically disabling the backlight if no activity occurs within a preset period of time

3.5.3 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 PXA255 processor is 166 MHz, resulting in a maximum pixel clock rate of 83 MHz. Thus, use of 100 MHz design considerations are sufficient to ensure LCD panel signal integrity.

However, typical transfer rates are considerably less than 83 Mhz. For example, an 800x600 color active display running at 75 Hz requires a transfer rate of approximately 36 MHz. To determine this, calculate the number of pixels ($800 \times 600 = 480,000$) and multiply by the screen refresh rate (75 Hz). Since active panels 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 you may use a lower refresh rate to reduce transmission rates even more.

Passive displays often do require refresh rates greater than 75 Hz, which transfers more pixels each clock cycle. For instance, a color passive display with 8 data lines transfers $2 \frac{2}{3}$ pixels' worth of data each clock cycle. This divides the transmission rate by $2 \frac{2}{3}$. Further reductions in the transfer rate come by using dual panel displays which use twice as many data lines to transfer data - halving the rate again.

Generally, this gives you lower transfer rates to even large 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 to be 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. There are other factors related to choosing a refresh rate for an LCD system, most significant is the impact on system bandwidth.

If you must use excessively long or poorly routed signals, one possible solution is to add buffers between the PXA255 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 major concern.

When mounting the LCD panel, it is critical to shield the touchscreen control lines, if present. Noise from the LCD panel and its control signals can become injected into the touchscreen control lines, causing spurious touch interrupts or loss of resolution.

3.5.4 Panel Connector

Most LCD panels are connected to the system board via a connector, instead of being directly mounted on the system board. This increases flexibility and ease of manufacture. Typically the manufacturer of the panel recommends a particular connector for the panel. Follow the panel manufacturer's recommendation.

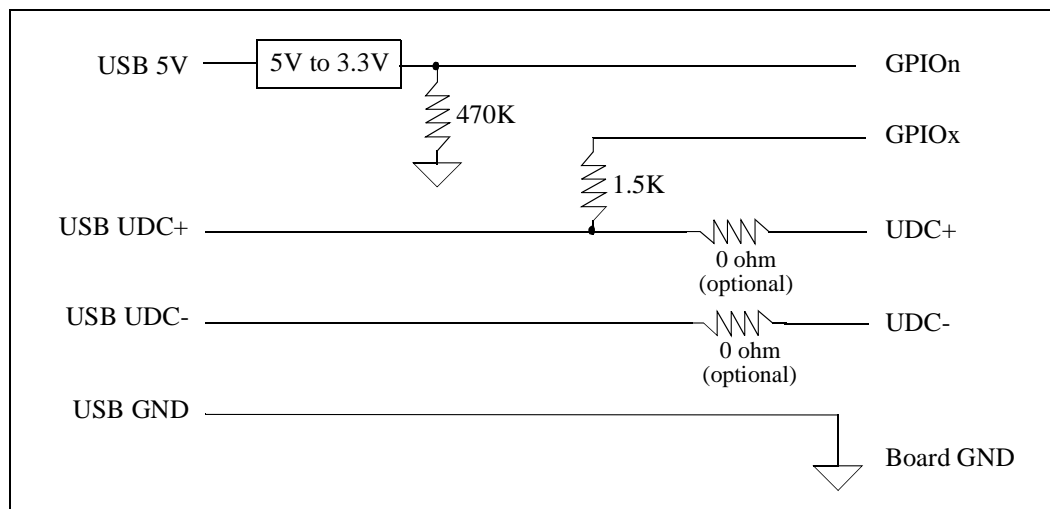
4.1 Self Powered Device

Figure 4-1 shows the USB interface connection for a self-powered device. The 0 ohm resistors are optional, and if not used, then connect USB UDC+ directly to the device UDC+ and connect USB UDC- directly to device UDC-. The device UDC+ and UDC- pins match the impedance of a USB cable, 90 ohms, without the use of external series resistors. You may install 0 ohm resistors on your board to compensate for minor differences between the USB cable and your board trace impedance.

The 5 to 3.3 voltage divider is required since the device GPIO pins cannot exceed 3.3 V. This voltage divider can be implemented in a number of ways. The most robust and expensive solution is to use a MAX6348 Power-On-Reset device. This solution produces a very clean signal edge and minimizes signal bounce. The more inexpensive solution is to use a 3.3 V line buffer with 5 V tolerant inputs. This solution does not reduce signal bounce, so software must compensate by reading the GPIO signal after it stabilizes. A third solution is to implement a signal bounce minimization circuit that is 5 V tolerant, but produces a 3.3 V signal to the GPIO pin.

Note: If GPIO_{on} and GPIO_x are the same pin, never put the device to sleep while the USB cable is connected to the device. During sleep, the USB controller is in reset and will not respond to the host; after sleep, the device will not respond to its host-assigned address.

Figure 4-1. Self Powered Device



4.1.1 Operation if GPIO_{on} and GPIO_x are Different Pins

Any GPIO pins can be defined as GPIO_{on} and GPIO_x. GPIO_{on} should be a GPIO which can bring the device out of sleep. Out of reset, configure GPIO_x as an input that causes the UDC+ line to float. GPIO_{on} is configured as an input that causes an interrupt whenever a rising or falling edge is detected. When an interrupt occurs, software must read the GPIO_{on} pin to determine if the cable is connected or not. GPIO_{on} is 1 if the cable is connected or 0 if the cable is disconnected. If a USB

connect is detected, then software enables the UDC peripheral and drives a 1 onto the GPIOx pin to indicate to the host PC a fast USB device is connected. If a USB disconnect is detected, then software must configure the GPIOx pin as an input, configure the GPIOn pin to detect a wakeup event, and then put the part into sleep mode.

Also, at any time, you may use software to put the part into sleep mode. Before entering sleep mode, configure the GPIOx pin as an input to cause the UDC+ line to float. This looks like a disconnect to the host PC. The device can then be put into sleep mode. When the device becomes active, software must drive a 1 onto the GPIOx pin to indicate to the host PC a fast USB device has been connected.

4.1.2 Operation if GPIOn and GPIOx are the Same Pin

Out of reset, GPIOn is configured as an input and configured to cause an interrupt whenever a rising or falling edge is detected. When an interrupt occurs, software must read the GPIOn pin to determine if the cable is connected or not. This pin is 1 if the cable is connected or 0 if the cable is disconnected. If the USB cable is connected, then software must enable the UDC peripheral before the host sends the first USB command. If the USB cable is not connected, then software must configure the GPIOn pin to detect a wakeup event, and then put the part into sleep mode.

4.2 Bus Powered Device

The processor cannot support a bus powered device model. When the host sends a suspend, the device is required to consume less than 500 μ A (Section 7.2.3 of the USB spec version 1.1). The processor cannot limit its current consumption to 500 μ A unless it enters sleep mode. If it enters sleep mode, all USB registers are reset and it does not respond to its host-assigned address.

The MultiMediaCard (MMC) is a low cost data storage and communication media. The MMC supports the translation protocol from a standard MMC or Serial Peripheral Interface (SPI) bus to an application bus.

The MMC controller in the processor is compliant with *The MultiMediaCard System Specification, Version 2.1*. The only exception is one and three byte data transfers are not supported. The MMC controller is capable of communicating with a card in MMC or SPI mode. Your application is responsible for specifying the MMC controller communication mode.

5.1 Schematics

The MultiMediaCard (MMC) controller on the processor supports MMC and SDCard devices. (The MMC controller does not support SDCard nibble mode.) This section presents several options on how to connect each type of device to the controller.

5.1.1 Signal Description

MMC controller signal functions are described in [Table 5-1](#).

Table 5-1. MMC Signal Description

Signal Name	Input/Output	Description
MMCLK	Output	Clock signal to MMC
MMCMD	BiDirectional	Command line
MMDAT	BiDirectional	Data line
MMCCS0	Output	Chip Select 0
MMCCS1	Output	Chip Select 1

The MMCLK, MMCCS0, and MMCCS1 signals are routed through alternate functions within the processor general purpose input/output (GPIO) module. Each of these signals can be programmed to a particular GPIO pin.

The signals defined in *The MultiMediaCard System Specification* for an MMC device are CLK, CMD, and DAT which correspond to the MMCLK, MMCMD, and MMDAT in the processor, respectively. The two chip selects in the controller are for the MMC SPI mode and correspond to the reserved pin of two different devices, defined in the specification.

The signals defined in the *Physical Layer Specification* of the *SD Memory Card Specifications* for an SDCard device are CLK, CMD, and DAT0-DAT3. The obvious difference is the number of DAT signals. In addition, the socket for an SDCard contains mechanical switches for write protect (WP) and card detect (CD). For an SDCard to be connected to the MMC controller, only one data line, DAT0, is used. Otherwise the signal mapping remains the same as an MMC device. The WP and CD switches on the socket are discussed in [Section 5.1.2, “How to Wire”](#) on page 5-2.

5.1.2 How to Wire

Notice in the example schematic ([Figure 5-1, “Processor MMC and SDCard Signal Connections”](#) on [page 5-3](#)) an SDCard socket is used. The signals on the socket are defined in [Table 5-2](#).

Table 5-2. SDCard Socket Signals

Signal Name	Pin #
DAT3	1
CMD	2
VSS1	3
VDD	4
CLK	5
VSS2	6
DAT0	7
DAT1	8
DAT2	9

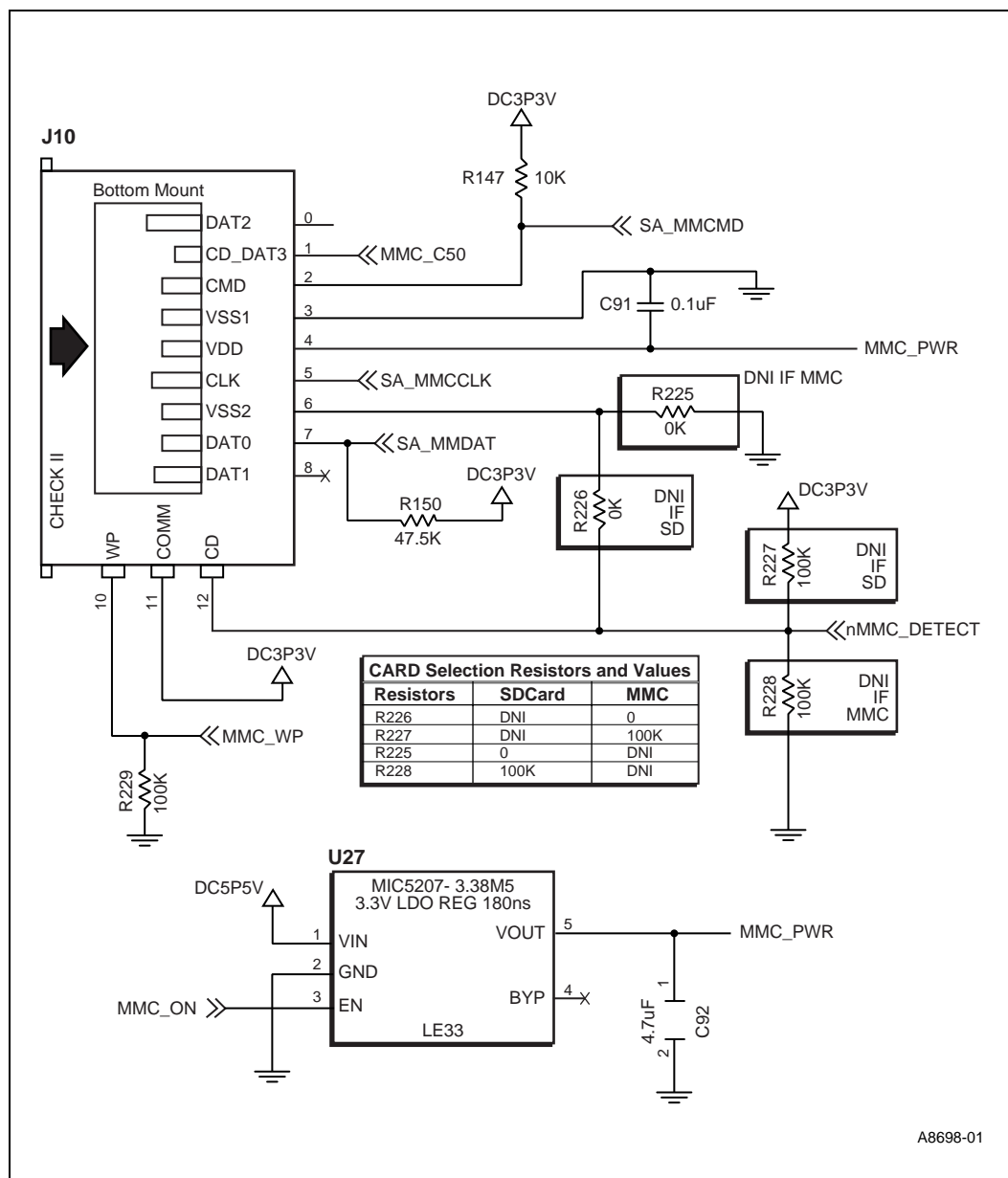
As stated previously, the PXA255 processor MMC controller can be connected to either an MMC device or an SDCard device, but you are limited to which device installs in which socket. Refer to [Table 5-3](#) for information on sockets and device supported by the MMC controller.

Table 5-3. MMC Controller Supported Sockets and Devices

Sockets	Devices Supported
SDCard socket	SDCard device MMC device
MMC socket	MMC device

[Figure 5-1](#) is a schematic that supports both MMC and SDCard devices. In the schematic, the signals SA_MMCLK, SA_MMCMMD, and SA_DAT correspond to the processor signals MMCLK, MMCMMD, and MMDAT, respectively. These three signals are also directly connected to the socket.

Figure 5-1. Processor MMC and SDCard Signal Connections



MMC_CS0, which corresponds to the processor MMCCS0 signal, is connected to the socket at pin 1. This connection is the SPI mode chip select and is available on both MMC and SDCard. This pin is also labeled DAT3. DAT3 is only used with an SDCard in SDCard mode and not available on the processor MMC controller.

The signals DAT1 and DAT2 are not connected because these are specific to SDCard operation in SDCard mode.

Three other signals shown on the connector are COMM and the mechanical switches write protect (WP) and card detect (CD). When a device is inserted in the example schematic ([Figure 5-1](#)), WP may be and CD is connected to COMM via a mechanical switch inside the socket

SDCard devices have a write protect tab. Depending on the position of the tab, the WP signal may or may not be connected to the COMM signal. Connect the WP signal to a CPLD or other device capable of indicating to the driver software that the card is write protected. In this example, COMM is tied to a VCC and WP has a pull-down resistor. This causes a rising edge when the tab is in the write protect position and the WP signal remains low when the tab is in the read/write position.

The CD signal, MMC_DETECT, indicates to the MMC controller when a card is installed. It is used for both an SDCard socket and an MMC socket. Since the MMC socket does not have the mechanical CD switch, other measures must be taken to produce a card detect. Thus, the SDCard and MMC cases are discussed separately.

Note: While this schematic shows two ways to create a card detect, it is recommended that an SDCard socket be used if a card detect and write protection signal are desired even if only MMC devices are being used.

5.1.2.1 SDCard Socket

When using [Figure 5-1, “Processor MMC and SDCard Signal Connections”](#) on page 5-3 as a template for your SDCard circuit design, all resistors labeled “DNI IF SD” should not be installed and all resistors labeled “DNI IF MMC” should be installed in the circuit. Removing R226 and inserting R225 causes the VSS2 signal on pin 6 to be tied to ground. Also, the SDCard needs a pull-down resistor in position R228.

SDCard sockets have a card detect switch internal to the socket. The CD signal is physically connected to the COMM signal. Connect the CD signal to a CPLD or other device capable of indicating to the driver software that a card has been inserted in the socket. In this example, COMM is tied to a VCC and CD has a pull-down resistor. This causes a rising edge on CD when a card is inserted while the CD signal remains low if no card is in the socket.

5.1.2.2 MMC Socket

When using [Figure 5-1, “Processor MMC and SDCard Signal Connections”](#) on page 5-3 as a template for your MMC circuit design, all resistors labeled “DNI IF MMC” should not be installed and all resistors labeled “DNI IF SD” should be installed in the circuit. This causes the VSS2 signal on pin 6 to be pulled-up through resistor R227.

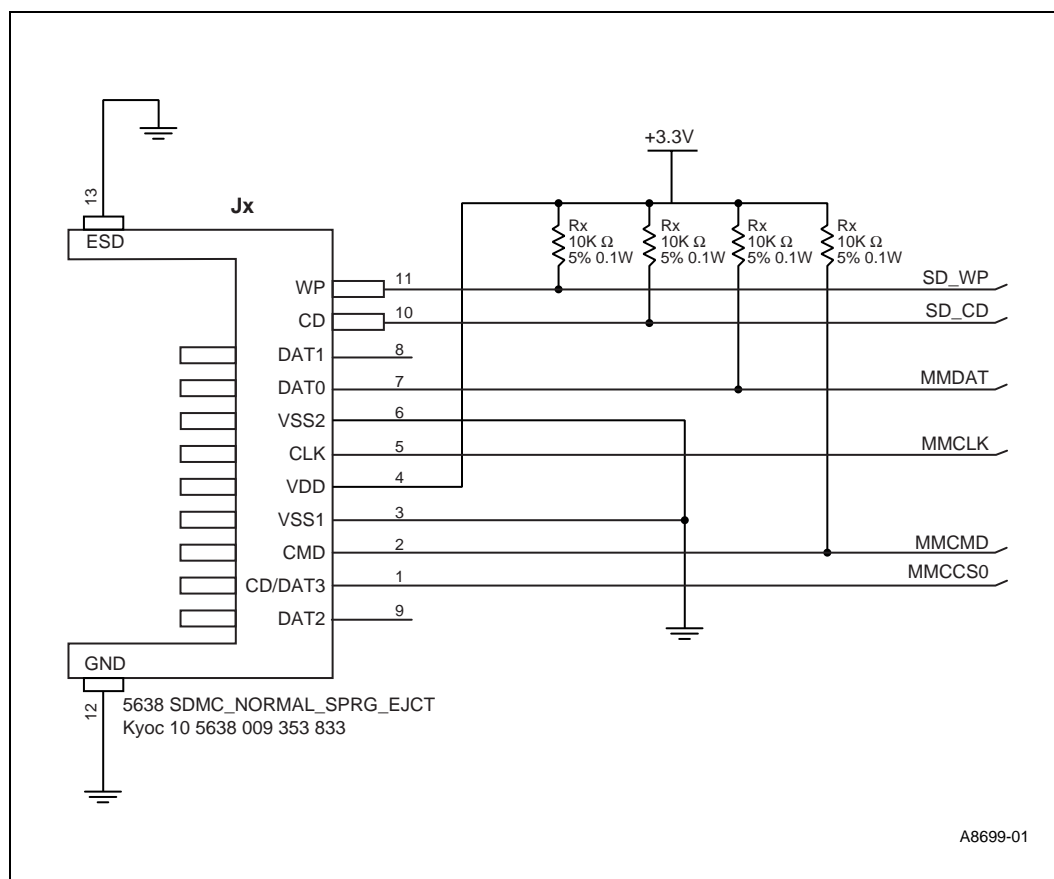
Unlike SDCard sockets, MMC sockets do not have a card detect or write protect switch. In order to implement this, a pull-up is placed on the VSS2 signal (pin 6 of the socket.) Since VSS2 and VSS1 are connected internally on the MMC device, the signal called nMMC_DETECT on the schematic is driven low when the MMC device is inserted.

Warning: Connecting VSS2 to something other than the power supply ground violates *The MultiMediaCard System Specification, Version 2.1*. Because the MMC specification does not state that VSS1 and VSS2 must be connected internal to the MMC device, the design in Figure 5-1 may not work with all MMC devices. Use caution when using the card detection method shown in Figure 5-1.

5.1.3 Simplified Schematic

Figure 5-2 shows another SDCard socket. In this case, all processor signals are connected to the socket. This socket does not have a common signal for the write protect and card detect and are connected to the two tabs shown on the left side of the diagram. Inserting a card into the socket may cause the write protect signal and will cause the card detect signal to change states and must be interpreted by the CPLD software.

Figure 5-2. Processor MMC to SDCard Simplified Signal Connection



5.1.4 Pull-up and Pull-down

Table 5-4 and Table 5-5 show the pull-up and pull-down resistors required for SDCard and MMC devices according to their respective specifications.

Table 5-4. SDCard Pull-up and Pull-down Resistors

Signal	Pull-up or Pull-down	Min	Max	Remark
CMD	pull-up	10k Ω	100k Ω	Prevents bus floating
DAT0-DAT3	pull-up	10k Ω	100k Ω	Prevents bus floating
WP ¹	pull-up	—	—	Any value sufficient to prevent bus floating
NOTE: 1. This resistor is shown in the specification but the value is not specified				

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

Signal	Pull-up or Pull-down	Min	Max	Remark
CMD	pull-up	4.7k Ω	100k Ω	Prevents bus floating
DAT	pull-up	50k Ω	100k Ω	Prevents bus floating

5.2 Utilized Features

The processor MultiMediaCard controller has these features:

- Data transfer rates as fast as 20 Mbps
- A16 bit response FIFO
- Dual receive data FIFOs
- Dual transmit FIFOs
- Support for two MMCs in either MMC or SPI mode

The sample schematics in this section support MMC and SDCard and are configured to use MMC or SPI mode.

The processor MultiMediaCard controller and the MMC device have the same maximum data rate, 20 Mbps, so their communication rates are compatible. However, because the maximum processor MultiMediaCard controller data rate is 20 Mbps and the maximum SDCard data rate is 25 Mbps, SDCard devices are not utilized to their fullest extent.

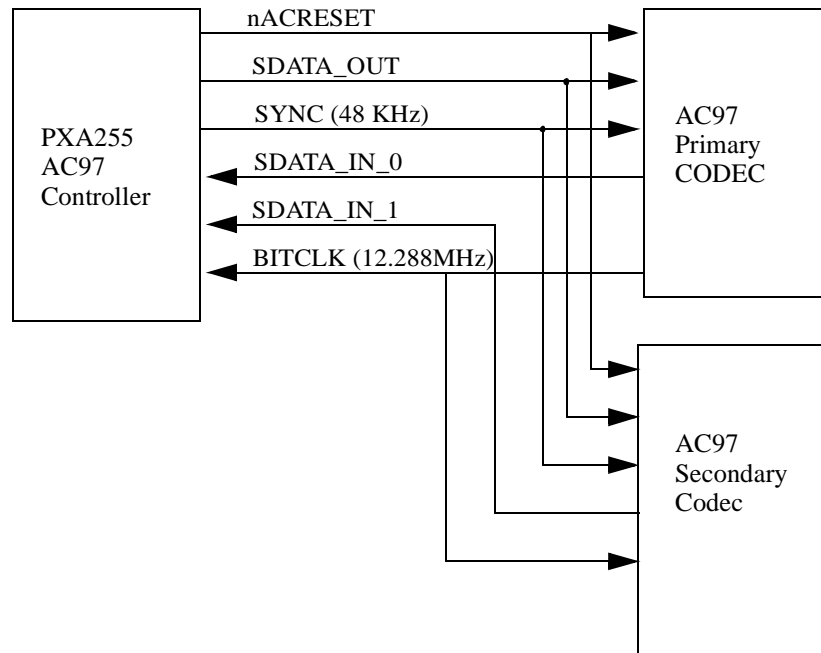
The circuit designs presented in this guide ([Figure 5-1](#) and [Figure 5-2](#)) only show support for one SDCard or MMC device, but the processor MultiMediaCard controller handles as many as two devices.

The AC97 controller unit (ACUNIT) connects audio chips and codecs to the processor. It uses a six-wire interface to transmit and receive data from AC97 2.0 compliant codecs. The AC97 port is a bidirectional, serial PCM digital stream. A maximum of two codecs may be connected to the ACUNIT.

6.1 Schematics

The schematics for an AC97 connection are shown in [Figure 6-1](#). The primary codec supplies the 12.288 MHz clock to the AC97. This clock is then driven into the ACUNIT on the processor and the AC97 Secondary Codec.

Figure 6-1. AC97 connection



6.2 Layout

Because of the analog/digital nature of the codecs, it is important that proper mixed-signal layout procedures be followed. Intel recommends you follow the layout recommendations given in your Codec datasheet. 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. You can 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 1/8 of an inch separation between them.
- Connect the two ground planes underneath your codec with a 0 ohm jumper. Add optional Do Not Populate 0 ohm jumpers between analog and digital ground at the power supply. Excessive noise on the board may be reduced by installing the 0 ohm resistor.
- Do not 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 your 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 Inter-Integrated Circuit (I²C) bus interface unit lets the PXA255 processor serve as a master and slave device residing on the I²C bus. The I²C bus is a serial bus developed by Philips Corporation consisting of a two-pin interface. SDA is the serial data line and SCL is the serial clock line.

Using the I²C bus lets the processor interface to other I²C 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 I²C bus interface unit is a peripheral device that resides on the processor internal bus. Data is transmitted to and received from the I²C bus via a buffered interface. Control and status information is relayed through a set of memory-mapped registers. Refer to the I²C Bus Specification for complete details on I²C bus operation.

7.1 Schematics

The I²C bus is used by many different applications. This reference guide presents two possible methods for using the I²C 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.

7.1.1 Signal Description

The I²C bus interface unit signals are SDA and SCL. [Table 7-1](#) describes the function of each signal.

Table 7-1. I²C Signal Description

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

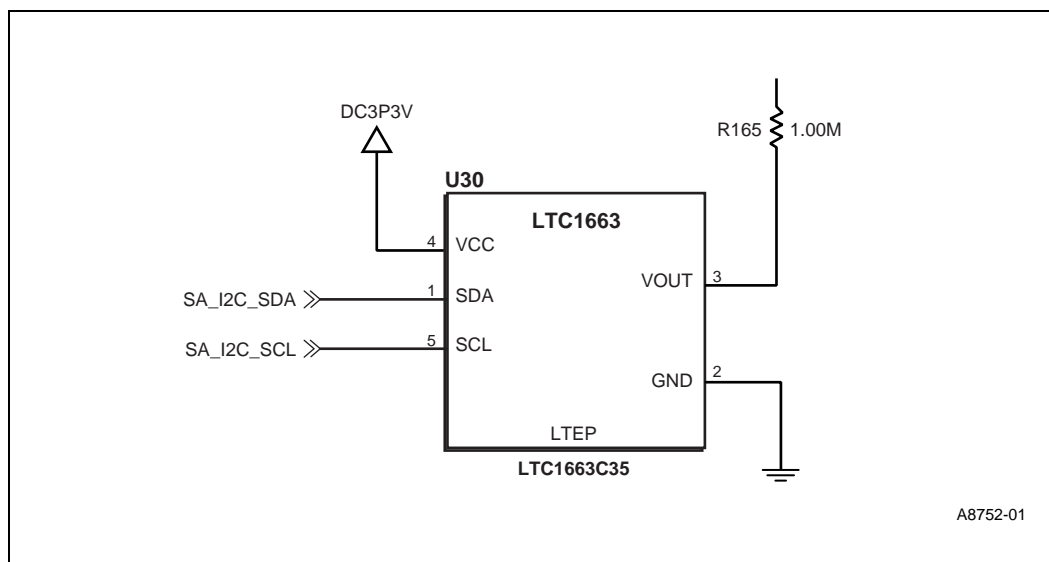
The I²C bus serial operation uses an open-drain, wired-AND bus structure, which allows multiple devices to drive the bus lines and to communicate status about events such as arbitration, wait states, error conditions and so on. For example, when a master drives the clock (SCL) line during a data transfer, it transfers a bit on every instance that the clock is high. When the slave is unable to accept or drive data at the rate that the master is requesting, the slave can hold the clock line low between the high states to insert a wait interval. The master's clock can only be altered by a slow slave peripheral keeping the clock line low or by another master during arbitration.

The I²C bus lets you design a multi-master system; meaning more than one device can initiate data transfers at the same time. To support this feature, the I²C bus arbitration relies on the wired-AND connection of all I²C interfaces to the I²C bus. Two masters can drive the bus simultaneously provided they are driving identical data. The first master to drive SDA high while another master drives SDA low loses the arbitration. The SCL line consists of a synchronized combination of clocks generated by the masters using the wired-AND connection to the SCL line.

7.1.2 Digital-to-Analog Converter (DAC)

Figure 7-1 shows the schematic for connecting the I²C interface to a Linear Technology micropower DAC. The DAC output is connected to the buck converter feedback path and is controlled by the I²C bus interface unit. The DAC can modify the voltage of the feedback path, which effects the processor core voltage.

Figure 7-1. Linear Technology DAC with I²C Interface



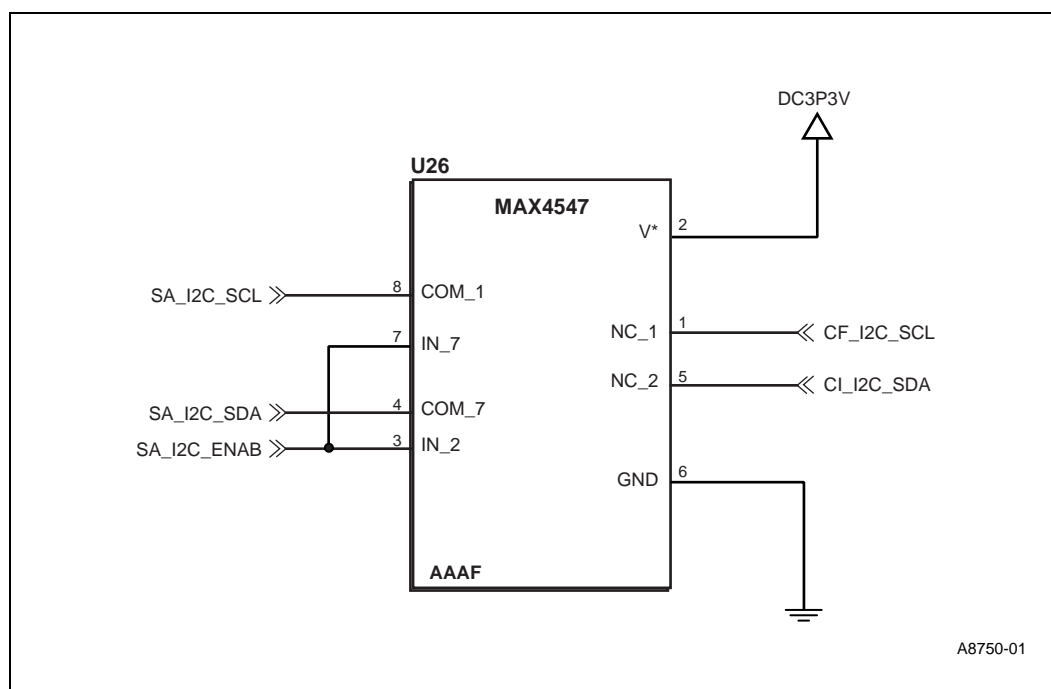
The signals SA_I2C_SDA and SA_I2C_SCL correspond to the processor signals SDA and SCL, respectively.

7.1.3 Other Uses of I²C

Figure 7-2 shows the I²C signals passing through an analog switch to a compact flash 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 you decide not to use a CF card, a different application using a CF card socket could be designed to utilize the I²C bus interface unit. If this alternate function is used, the I²C bus can be enabled to the CF socket by asserting the signal SA_I2C_ENAB shown in the diagram. If the user decides to use a CF Card, negate the SA_I2C_ENAB signal so the I²C 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 7-2. Using an Analog Switch to Allow a Second CF Card



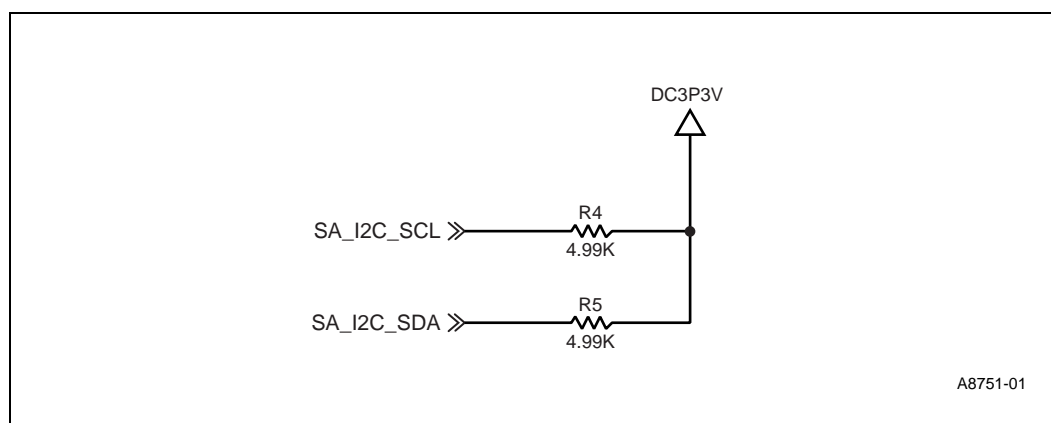
7.1.4 Pull-Ups and Pull-Downs

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 200pF, so the pull-up device is a resistor as shown in Figure 7-3.

Figure 7-3. I²C Pull-Ups and Pull-Downs



The actual value of the pull-up is system dependant 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.

7.2 Utilized Features

The processor I²C bus interface unit is compatible with the two pin interface developed by Phillips Corporation. A complete list of features and capabilities can be found in the *I²C Bus Specification*.

8.1 Operating Conditions

Table 8-1 shows voltage, frequency, and temperature specifications for the PXA255 processor for four different ranges. The temperature specification for each range is constant; the frequency range is operation voltage dependent. On a prototype design, the VCC/PLL_VCC regulator should have a range from 0.95 V to 1.65 V. PLL_VCC and VCC must be connected together on the board or driven by the same supply.

Table 8-1. Voltage, Temperature, and Frequency Electrical Specifications

Symbol	Description	Min	Typical	Max
t_{CE}	Extended Case Temperature	-40° C	—	100° C
t_{CC}	Commercial Case Temperature	0° C	—	85° C
V_{VSS}	VSS, VSSN, VSSQ Voltage	-0.3 V	0 V	0.3 V
V_{VCCQ}	VCCQ	3.0 V	3.3 V	3.6V
V_{VCCN_H}	VCCN @ 3.3V	3.0 V	3.3 V	3.6V
Low Voltage Range				
V_{VCC_L}	VCC, PLL_VCC Voltage, Low Range	0.95 V	1.0 V	1.32 V
f_{TURBO_L}	Turbo Mode Frequency, Low Range	99.5 MHz	—	132.7 MHz
f_{SDRAM_L}	External Synchronous Memory Frequency, Low Range	—	—	66.4 MHz
Medium Voltage Range				
V_{VCC_M}	VCC, PLL_VCC Voltage, Mid Range	0.95 V	1.0 V	1.32 V
f_{TURBO_M}	Turbo Mode Frequency, Mid Range	99.5 MHz	—	199.1 MHz
f_{SDRAM_M}	External Synchronous Memory Frequency, Mid Range	—	—	99.5 MHz
High Voltage Range				
V_{VCC_H}	VCC, PLL_VCC Voltage, High Range	1.045 V	1.1 V	1.43 V
f_{TURBO_H}	Turbo Mode Frequency, High Range	99.5 MHz	—	298.7 MHz
f_{SDRAM_H}	External Synchronous Memory Frequency, High Range	—	—	99.5 MHz
Peak Voltage Range				
V_{VCC_P}	VCC, PLL_VCC Voltage, Peak Range	1.235 V	1.3 V	1.65 V
f_{TURBO_P}	Turbo Mode Frequency, Peak Range	99.5 MHz	—	398.2 MHz
f_{SDRAM_P}	External Synchronous Memory Frequency, Peak Range	—	—	99.5 MHz
NOTE: When VCCN=2.5 V, the I/O signals that are supplied by VCCN are 2.5 V tolerant only. Do not apply 3.3 V to any pin supplied by VCCN in this case.				

8.2 Electrical Specifications

Table 8-2 provides the Absolute Maximum ratings for the processor. These parameters may not be exceeded or the part may be permanently damaged. Operation at absolute maximum ratings is not guaranteed.

Table 8-2. Absolute Maximum Ratings

Symbol	Description	Min	Max
T_S	Storage Temperature	-40° C	125° C
V_{SS_O}	Offset Voltage between any two VSS pins (VSS, VSSQ, VSSN)	-0.3 V	0.3 V
V_{CC_O}	Offset Voltage between any of the following pins: VCCQ, VCCN	-0.3 V	0.3 V
V_{CC_HV}	Voltage Applied to High Voltage Supplies (VCCQ, VCCN)	VSS-0.3 V	VSS+4.0 V
V_{CC_LV}	Voltage Applied to Low Voltage Supplies (VCC, PLL_VCC)	VSS-0.3 V	VSS+1.45 V
V_{IP}	Voltage Applied to non-Supply pins except XTAL pins	VSS-0.3 V	max of VCCQ+0.3 V, VSS+4.0 V
V_{IP_X}	Voltage Applied to XTAL pins (PXTAL, PEXTAL, TXTAL, TEXTAL)	VSS-0.3 V	max of VCC+0.3 V, VSS+1.45 V
V_{ESD}	Maximum ESD stress voltage, Human Body Model; Any pin to any supply pin, either polarity, or Any pin to all non-supply pins together, either polarity. Three stresses maximum.		2000 V
I_{EOS}	Maximum DC Input Current (Electrical Overstress) for any non-supply pin		5 mA

8.2.1 Power Supply Connectivity

The PXA255 processor requires two or three externally-supplied voltage levels. VCCQ requires high voltage, VCCN requires high or medium voltage, and VCC and PLL_VCC require low voltage. PLL_VCC must be separated from other low voltage supplies. Depending on the availability of independent regulator outputs and the desired memory voltage, VCCQ may have to be separated from VCCN. VCCN does not have to be separated at the board level.

Table 8-3. PXA255 Processor VCCN vs. VCCQ (Sheet 1 of 6)

Pin	Pin Count	Signal Description and Comments	Power Supply
MA(25:0)	26	Main Memory Address Bus	VCCN
MD(31:0)	16	Main Memory Data Bus	VCCN
nOE	1	Main Memory Bus Output Enable	VCCN
nWE	1	Main Memory Bus Write Enable	VCCN
nSDRAS	1	Main Memory Bus RAS	VCCN
nSDCAS	1	Main Memory Bus CAS	VCCN

Table 8-3. PXA255 Processor VCCN vs. VCCQ (Sheet 2 of 6)

Pin	Pin Count	Signal Description and Comments	Power Supply
DQM(3:0)	2	Main Memory Bus SDRAM byte selects	VCCN
nSDCS(3:0)	2	Main Memory Bus SDRAM chip selects	VCCN
SDCKE(1:0)	2	Main Memory Bus SDRAM clock enable	VCCN
SDCLK(2:0)	1	Main Memory Bus SDRAM clocks	VCCN
RD/nWR	1	CC Steering Signal	VCCN
CS(0)	1	Static chip select	VCCN
GP15	1	Active low chip select 1	VCCN
GP18	1	External Bus Ready	VCCN
GP19	1	External Bus Master Request	VCCN
GP20	1	External Bus Master Request	VCCN
GP21	1	General Purpose I/O pin	VCCN
GP22	1	General Purpose I/O pin	VCCN
GP33	1	Active low chip select 5	VCCN
GP48	1	Output Enable for Card Space	VCCN
		HWUART transmit	
GP49	1	Write Enable for Card Space	VCCN
		HWUART receive	
GP50	1	I/O Read for Card Space	VCCN
		HWUART clear to send	
GP51	1	I/O Write for Card Space	VCCN
		HWUART request to send	
GP52	1	Card Enable for Card Space	VCCN
GP53	1	Card Enable for Card Space	VCCN
		MMC CLock	
GP54	1	MMC CLock	VCCN
		Socket Select for Card Space	
GP55	1	Card Address bit 26	VCCN
GP56	1	Wait signal for Card Space	VCCN
GP57	1	Bus Width select for I/O Card Space	VCCN
GP78	1	Active low chip select 2	VCCN
GP79	1	Active low chip select 3	VCCN

Table 8-3. PXA255 Processor VCCN vs. VCCQ (Sheet 3 of 6)

Pin	Pin Count	Signal Description and Comments	Power Supply
GP80	1	Active low chip select 4	VCCN
GP81	1	Network SSP Clock	VCCQ
GP82	1	Network SSP Frame	VCCQ
GP83	1	Network SSP TXD/RXD	VCCQ
GP84	1	Network SSP TXD/RXD	VCCQ
MMCMD	1	MMC Command	VCCQ
MMDAT	1	MMC Data	VCCQ
AC_RESET_n	1	ac97 RESET	VCCQ
UDC+	1	USB client high differential signal	VCCQ
UDC-	1	USB client low differential signal	VCCQ
SCL	1	I2C Clock	VCCQ
SDA	1	I2C Bidirectional Data	VCCQ
nRESET	1	Hardware reset	VCCQ
nRESET_OUT	1	Reset output	VCCQ
BOOT_SEL[2:0]	3	ROM Width Select (16/32)	VCCQ
PWR_EN	1	power enable	VCCQ
nBATT_FAULT	1	Battery Fault	VCCQ
nVDD_FAULT	1	VDD Fault	VCCQ
nTRST	1	JTAG Reset	VCCQ
TDI	1	JTAG Data In	VCCQ
TDO	1	JTAG Data Out	VCCQ
TMS	1	JTAG Mode Select	VCCQ
TCK	1	JTAG Clock	VCCQ
TESTCLK	1	TEST Clock	VCCQ
TEST	1	TEST mode	VCCQ
GP0	1	Reserved for sleep wakeup	VCCQ
GP1	1	Active low GP_reset	VCCQ
GP2	1	General Purpose I/O pin	VCCQ
GP3	1	General Purpose I/O pin	VCCQ
GP4	1	General Purpose I/O pin	VCCQ
GP5	1	General Purpose I/O pin	VCCQ
GP6	1	MMC Clock	VCCQ

Table 8-3. PXA255 Processor VCCN vs. VCCQ (Sheet 4 of 6)

Pin	Pin Count	Signal Description and Comments	Power Supply
GP7	1	48 mhz clock output	VCCQ
GP8	1	MMC Chip Select 0	VCCQ
GP9	1	MMC Chip Select 1	VCCQ
GP10	1	real time clock (1Hz)	VCCQ
GP11	1	3.6 MHz oscillator out	VCCQ
GP12	1	32 KHz out	VCCQ
GP13	1	memory controller grant	VCCQ
GP14	1	Alternate Bus Master Request	VCCQ
GP16	1	PWM0 output	VCCQ
GP17	1	PWM1 output	VCCQ
GP23	1	SSP clock	VCCQ
GP24	1	SSP Frame	VCCQ
GP25	1	SSP transmit	VCCQ
GP26	1	SSP receive	VCCQ
GP27	1	SSP ext_clk	VCCQ
GP28	1	AC97 bit_clk	VCCQ
		I2S bit_clk	
		I2S bit_clk	
		AC97 bit_clk	
GP29	1	AC97 Sdata_in0	VCCQ
		I2S Sdata_in	
GP30	1	I2S Sdata_out	VCCQ
		AC97 Sdata_out	
GP31	1	I2S sync	VCCQ
		AC97 sync	
GP32	1	I2S sysclk	VCCQ
		AC97 Sdata_in1	
GP34	1	FFUART receive	VCCQ
		MMC Chip Select 0	
GP35	1	FFUART Clear to send	VCCQ
GP36	1	FFUART Data carrier detect	VCCQ
GP37	1	FFUART data set ready	VCCQ

Table 8-3. PXA255 Processor VCCN vs. VCCQ (Sheet 5 of 6)

Pin	Pin Count	Signal Description and Comments	Power Supply
GP38	1	FFUART Ring Indicator	VCCQ
GP39	1	MMC Chip Select 1	VCCQ
		FFUART transmit data	
GP40	1	FFUART data terminal Ready	VCCQ
GP41	1	FFUART request to send	VCCQ
GP42	1	BTUART receive data	VCCQ
		HWUART receive data	
GP43	1	BTUART transmit data	VCCQ
		HWUART transmit data	
GP44	1	BTUART clear to send	VCCQ
		HWUART clear to send	
GP45	1	BTUART request to send	VCCQ
		HWUART request to send	
GP46	1	ICP receive data	VCCQ
		STD_UART receive data	
GP47	1	STD_UART transmit data	VCCQ
		ICP transmit data	
GP58	1	LCD data pin 0	VCCQ
GP59	1	LCD data pin 1	VCCQ
GP60	1	LCD data pin 2	VCCQ
GP61	1	LCD data pin 3	VCCQ
GP62	1	LCD data pin 4	VCCQ
GP63	1	LCD data pin 5	VCCQ
GP64	1	LCD data pin 6	VCCQ
GP65	1	LCD data pin 7	VCCQ
GP66	1	LCD data pin 8	VCCQ
		Alternate Bus Master Request	
GP67	1	LCD data pin 9	VCCQ
		MMC Chip Select 0	
GP68	1	MMC Chip Select 1	VCCQ
		LCD data pin 10	

Table 8-3. PXA255 Processor VCCN vs. VCCQ (Sheet 6 of 6)

Pin	Pin Count	Signal Description and Comments	Power Supply
GP69	1	MMC_CLK	VCCQ
		LCD data pin 11	
GP70	1	Real Time clock (1Hz)	VCCQ
		LCD data pin 12	
GP71	1	3.6MHz Oscillator clock	VCCQ
		LCD data pin 13	
GP72	1	32 KHz clock	VCCQ
		LCD data pin 14	
GP73	1	LCD data pin 15	VCCQ
		memory controller grant	
GP74	1	LCD Frame clock	VCCQ
GP75	1	LCD line clock	VCCQ
GP76	1	LCD Pixel clock	VCCQ
GP77	1	LCD AC Bias	VCCQ
PXTAL	1	3.6Mhz Crystal input	0.8 * VCC
PEXTAL	1	3.6Mhz Crystal output	0.8 * VCC
XTAL	1	32khz Crystal input	0.8 * VCC
TEXTAL	1	32khz Crystal output	0.8 * VCC

8.3 Example Form Factor Reference Design Power Delivery Example

8.3.1 Power System

Features of the example form factor reference design power system (example in [Figure 8-1](#), “Example Form Factor Reference Design Power System Design” on page 8-8) are:

- A standard-size cylindrical single-cell Li+ 3.6 V battery with a 1.8 Ahr capacity
- Battery temperature monitoring thermistor during charge cycles
- Battery voltage monitoring
- Charger supply voltage fault monitoring
- Low battery interrupt signal to the microprocessor.
- Provide power gating switches for the CF, LCD, backlight, RS-232, MMC, Radio, and audio amplifier subsystems
- Provide a high-efficiency 5.5 V supply rail for LCD and other devices

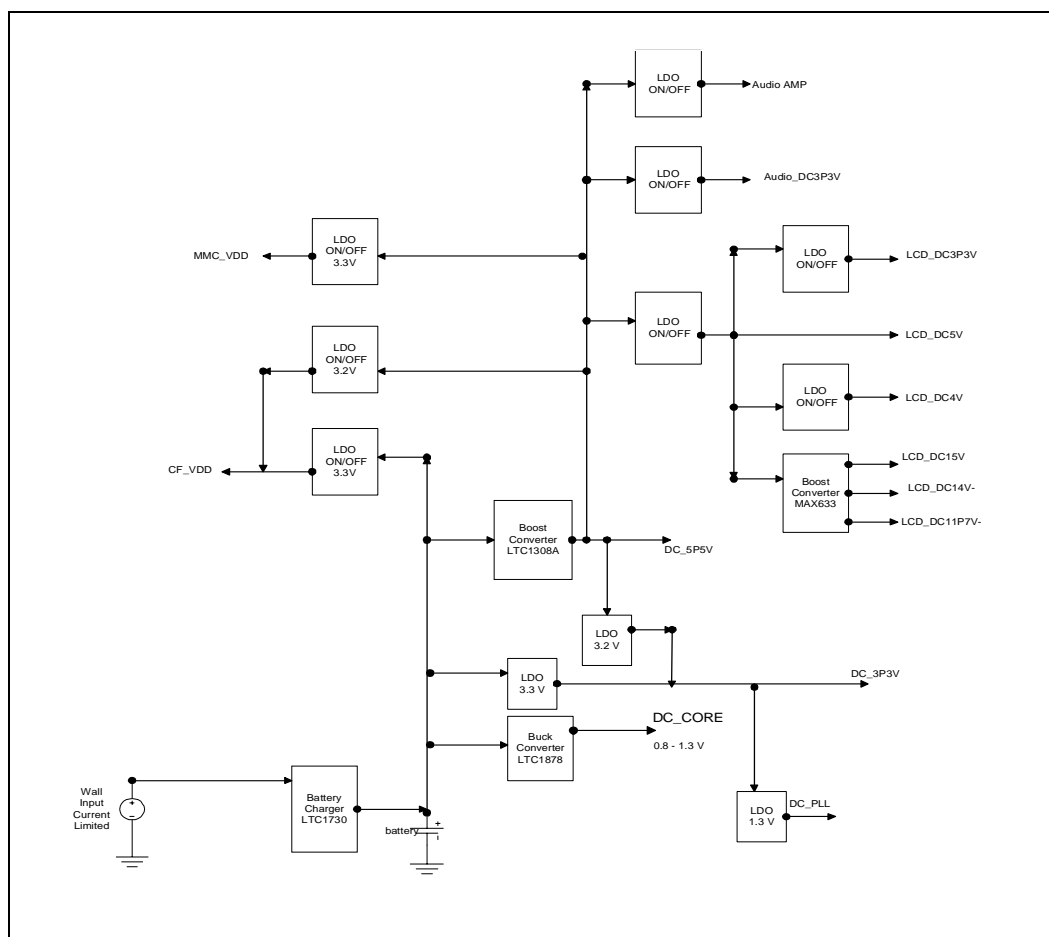
- Provide a high-efficiency 3.3 V supply rail for I/O and general system power
- Provide a high-efficiency 1.0 V – 1.3 V Core/PLL supply for the microprocessor
- Provide separate, clean power rails for the LCD and Audio subsystems
- A small, low-cost, low-heat dissipation pulse-charge method for the battery

8.3.1.1 Power System Configuration

The example form factor reference design power system design is described in [Figure 8-1, “Example Form Factor Reference Design Power System Design”](#) on page 8-8. Note that there are four main power rails in the system:

- 3.3 V I/O power
- 1.0-1.3 V Core/PLL power
- 5.5 V power

Figure 8-1. Example Form Factor Reference Design Power System Design



8.3.2 CORE Power

The example form factor reference design has a variable 1.0 V – 1.3 V core power supply for the processor. This voltage varies depending on the performance required by the application. A Linear Technologies LTC1878 buck converter is chosen for this application. The power is drawn directly from the Li+ battery. This device operates at 550 kHz and can supply up to 1 A at 1.0 V and 800 mA at 1.3 V with up to 95% efficiency. The device is turned on/off by the SA_PWR_EN signal directly from the processor.

The required output voltage is statically adjusted by selecting the value of the feed-back resistor. Ultimately, output voltage can be changed using software control of the Linear Technologies LTC1663 DAC. This DAC is controllable via the standard I2C bus, and can modify the voltage of the feedback path of the buck converter, which effects a change in the output voltage.

8.3.3 PLL Power

DC_PLL supplies power to the three PLLs within the processor. This pin requires a 1.0 V to 1.3 V nominal supply at an expected 20 mA load.

8.3.4 I/O 3.3 V Power

A simple LDO linear regulator supplies the 3.3V rail. The Analog Devices ADP3335 is chosen for its very low drop-out – 200 mV at 500 mA and 110 mV at 200 mA. So typically, the input cut-off voltage for this device is about $3.3\text{ V} + 0.11\text{ V} = 3.41\text{ V}$. The power is drawn directly from the Li+ battery. For a 3.6 V battery, this device has a 82% efficiency. There are four zones of operation for the Li+ battery:

- 4.1 – 3.8 V zone 10% of the time;
- 3.7 – 3.6 V zone at 70% of the time;
- 3.5 – 3.4 V at 10% of the time; and
- 3.4 – 3.1 V at 10% of the time.

The ADP3335 operates in zone 1,2, 3, and cutoffs in zone 4.

The overall efficiency is:

$$0.1(3.3/4.0) + 0.7(3.3/3.6) + 0.1(3.3/3.4) = 0.0825 + 0.642 + 0.097 = 0.82$$

To access the energy in zone 4 use the second LDO linear regulator in a parallel configuration with the ADP3335 and set it to output 3.2 V. Input to this regulator is 5.5 V from the boost converter. When the battery voltage drops below 3.5 V, the ADP3335 drops-out and the second regulator takes over.

8.3.5 Peripheral 5.5 V Power

The example form factor reference design provides a 5.5 V rail to supply power to LCD, Audio amplifier and Radio modules. An LT1308A boost converter is used. This device supplies up to 1 A at 5.5 V while operating at 600 kHz with up to 90% efficiency at rated load and 3.6 V input.

In addition, a low battery voltage detect circuit has an open-drain output. The detect voltage is set at 3.45 V by a resistor divider circuit. When the battery drops below 3.45 V the output transitions to a logic low. This output signal is used as a processor interrupt.

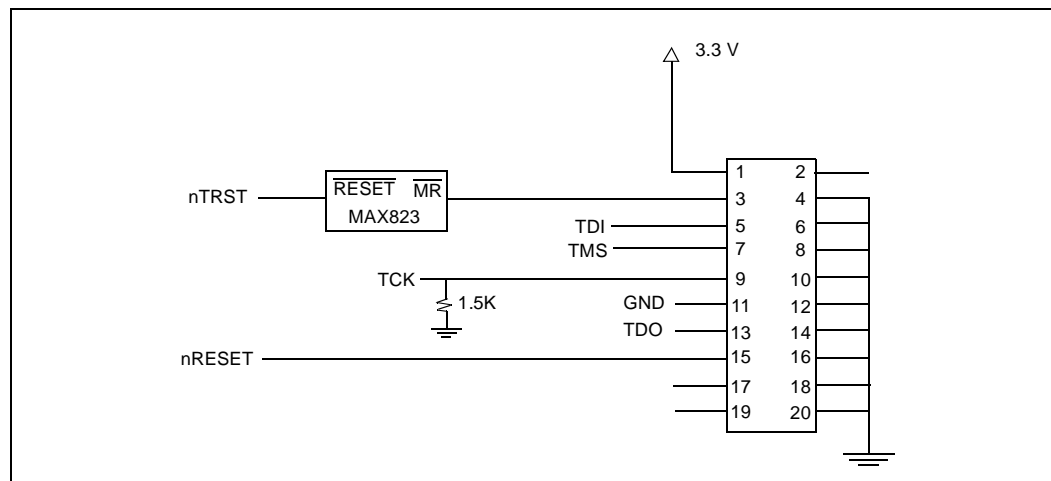
9.1 Description

The JTAG/Debug port is essentially several shift registers, with the destination controlled by the TMS pin and data I/O with TDI/TDO. nTRST provides initialization of the test logic. JTAG is testable via the IEEE 1149.1. Many use JTAG to control the address/data bus for Flash programming. JTAG is also a hardware debug port.

9.2 Schematics

All JTAG pins, except for nTRST and TCK, are directly connected. TCK is not driven internally and so you must add an external pull-up or pull-down resistor. Intel recommends adding a 1.5 k pull-down resistor to TCK. nTRST must be asserted during power-on. Asserting nRESET or nTRST must not cause the other reset signal to assert. Also, use an external pull-up resistor on nTRST to prevent spurious resets of the JTAG port when disconnected. The circuit in [Figure 9-1](#) drives nTRST. It uses a reset IC on nTRST to ensure that nTRST is reset at power-on. nRESET must be directly connected to the CPU nRESET. Do not connect pins 17 and 19 – they are special purpose functions and not used.

Figure 9-1. JTAG/Debug Port Wiring Diagram



If you are not utilizing either JTAG or the hardware debug functions, it is highly recommended that you design in a JTAG/debug port on your system anyway. This greatly facilitates board debug, startup, and software development. During final production you would not have to populate the JTAG connector.

9.3 Layout

Use the JTAG/Debug the port layout recommendations given in ARM's application note, *Multi-ICE System Design Considerations, Application Note 72*. The recommended connector is a 2x10-way, 2.54 mm pitch pin header, shown in [Figure 9-1](#).

If board space is critical, use a small form-factor receptacle with a smaller pitch. Then use a cable interface that has a wire “dongle” with a 2.54 mm pitch pin header on one end and the smaller pitch connector on the other.

Place the JTAG/Debug connector as close as possible to the PXA255 processor to minimize signal degradation.

If you follow these design recommendations, a JTAG bridge board is not required. Essentially, the JTAG bridge board for the example form factor reference design uses a 220 ohm resistor to tie nTRST high so that the JTAG logic can be brought out of reset (otherwise it would not come out of reset since nTRST is open-drain).

The PXA255 processor represents the next generation follow-on to Intel® StrongARM® SA-1110 product. This appendix highlights the migration path needed to change an SA-1110 design to one that uses the processor.

The majority of application code running on the SA-1110 will directly run on the processor, but there are substantial differences in hardware implementation and low-level coding, especially device configuration, that need to be noted.

The processor has numerous new hardware and software features that substantially benefit a handheld product design. The processor can be considered a superset of the SA-1110, but with so many new features direct socket compatibility is impractical.

For a detailed analysis of the differences between these products, refer to the full specifications of each device:

- SA-1110 Advanced Developers Manual, Order# 278240 [<http://developer.intel.com>]
- Intel® PXA255 Processor Developer's Manual, Order# 278693
- ARM® Architecture Reference Manual, Order# ARM DDI 0100D-10
- Intel 80200 Developers' manual, Order# 273411-002 [<http://developer.intel.com>]

Or later updated versions of any of the above.

This appendix is separated into sections that focus on three different issues:

1. SA-1110 hardware migration issues
 - Hardware Compatibility
 - Signal Changes
 - Power Delivery
 - Package
 - Clocks
 - UCB1300
2. SA-1110 software migration issues
 - Software Compatibility
 - Address space
 - Page Table Changes
 - Configuration registers
 - DMA
3. Using new features in the processor
 - Intel® XScale™ microarchitecture
 - Debugging

- Cache attributes
- Other Features
- Conclusion

A.1 SA-1110 Hardware Migration Issues

A.1.1 Hardware Compatibility

The majority of the features provided in the SA-1110 are also provided in the PXA255 processor. However, with the additional functionality of the PXA255 processor, the two devices are not pin compatible and cannot occupy the same socket.

There has been an effort to ensure Companion Devices that take advantage of SA-1110 memory interface access works with the PXA255 processor. The memory controls for taking over the memory bus such as those exercised by the SA-1111, are included in the PXA255 processor memory bus interface however, there are some issues.

One difference in particular is the way PXA255 processor toggles the A1 and A0 address lines. The SA-1110 toggled A1 and A0 regardless of the size of the data bus. With PXA255 processor, if the data bus is set to 16-bit, then A0 does not toggle and if the data bus is set to 32-bit, then neither A1 nor A0 toggles.

There is a big difference in manufacturing technology between the SA-1110 and the PXA255 processor. The most significant change being from a 0.35 micron CMOS technology to a finer lithography of 0.18 microns. Aside from a potential impact to signal edge rates this allows for lower processor voltage operation.

A.1.2 Signal Changes

There are two pins that control SA-1110 boot-up:

- ROM select pin that selects a 16 or 32-bit interface
- Synchronous Mask ROM enable pin that selects a synchronous or asynchronous ROM access

The PXA255 processor has three pins that select eight different boot select options (see [Table A-1](#)). The subset of these options that are SA-1110 equivalent are not compatible with the PXA255 processor pin polarities, so these pins must be selected afresh when designing with the PXA255 processor.

Table A-1. PXA255 Processor Boot Select Options (Sheet 1 of 2)

Boot Select Pins			Boot Location
2	1	0	
0	0	0	Asynchronous 32-bit ROM
0	0	1	Asynchronous 16-bit ROM
0	1	0	Synchronous 32-bit Flash
0	1	1	Synchronous 16-bit Flash

Table A-1. PXA255 Processor Boot Select Options (Sheet 2 of 2)

Boot Select Pins			Boot Location
2	1	0	
1	0	0	(1) Synchronous 32-bit Mask ROM (64 Mbit) (2) Synchronous 16-bit Mask ROM = 32bits (32 Mbit)
1	0	1	(1) Synchronous 16-bit Mask ROM (64 Mbit)
1	1	0	(2) Synchronous 16-bit Mask ROM = 32bits (64 Mbit)
1	1	1	(1) Synchronous 16-bit Mask ROM (32 Mbit)

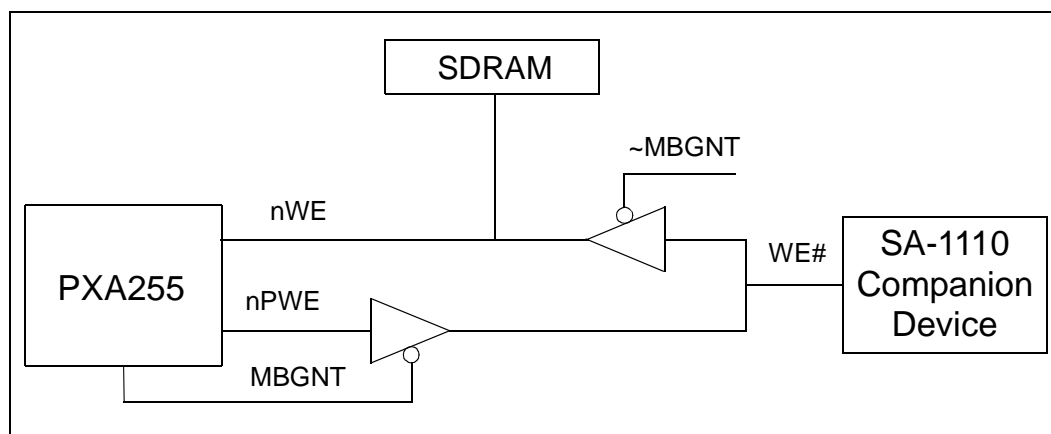
The power fault (VDD_FAULT) and battery fault (BATT_FAULT) pins that drive the SA-1110 sleep mode are negated with respect to the PXA255 processor. You must invert these signals or change the design to make sure that these signals are negated with respect to the SA-1110 design.

The PXA255 processor treats variable latency IO differently than the SA-1110. The difference occurs only when a static chip select is configured to support variable latency IO, i.e. the bus cycle is to be extended by a value on the RDY pin. In this configuration, the SDRAM refresh cycle retains the use of the nWE pin to allow the memory bus to be held for an indeterminate time. During any variable latency IO cycle, the PCMCIA pin nPWE is used to write to an external device instead of the nWE pin.

Note: Holding the bus for extended periods is not recommended because it interferes with the LCD DMA and prevents an LCD panel refresh.

This change in write enables only causes an issue if an external companion bus master device has a single write enable pin and requires variable latency IO to be accessed. As shown in Figure A-1, the write enable to the companion master has to be gated to differentiate between a case where the PXA255 processor uses the WE to write to the companion and a case where the companion uses the WE to write into SDRAM memory. Gating the WE pin with the Bus Grant signal (as shown) segregates the two different memory bus cycle types. If the companion bus master has both a WE input pin and a WE output pin to SDRAM, this logic is unnecessary.

Figure A-1. Write Enable Control Pins



A.1.3 Power Delivery

Although both products are tolerant to 3.3 V inputs and outputs, there is a difference in the supply voltage that drives the transistors of the microprocessor megacell. The PXA255 processor takes advantage of lower supply voltages to offer substantial power consumption savings. A design using SA-1110 has a supply voltage of 1.55 V to 1.75 V. The PXA255 processor is rated to 1.4 V maximum.

Drive the PXA255 processor core voltage pins at a lower voltage than the SA-1110 to reduce overall power consumption. The choice of voltage impacts the maximum upper frequency of operation so check the PXA255 processor documentation for the correct voltages as they are application dependent.

Also notice that the PXA255 processor supports independent power sources for Core, IO, Memory Bus, phase lock loops (PLLs), and a backup battery. It is recommended that these be independent power sources.

A.1.4 Package

The SA-1110 and the PXA255 processors are similar but not identical. The ball pitch of 1 mm is the same and the body outlines are both 17x17 mm but the heights are different. The PXA255 processor contains 4-layers within the package making it fractionally thicker than the SA-1110 2-layer package.

A.1.5 Clocks

The crystal inputs for the PXA255 processor are at the same frequency as those for the SA-1110:

- High frequency input of 3.6864 MHz
- Slow real-time clock source of 32.768 KHz.

The input frequency requirements are relatively low, such that any crystal that is an AT-cut style with a certain amount of shunt capacitance will work for both products.

The actual PLL design and process technology is different between the two products, such that a marginal SA-1110 design may not work with the PXA255 processor. Please refer to the product specifications of each device for further details.

You can program GPIO pins to generate various clocks in both the SA-1110 and the PXA255 processors. For example, these are often used in audio codec designs to generate clocks. The inter-relationships of some of these clocks have changed from the SA-1110 to the PXA255 processor. You may need to select different GPIO pins and program different configuration registers to provide similar functionality.

A.1.6 UCB1300

The SA-1110 supports a unique serial protocol for communication with the Philips UCB product family: UCB1100, 1200 and 1300. This serial interface is not available on the PXA255 processor. Instead the PXA255 processor supports several industry standard Audio codec Interfaces. You may also use I²S/I²C combinations and an AC'97 interface.

If an SA-1110 design utilizes this UCB interface then an alternative choice of components is necessary for the PXA255 processor.

A.2 SA-1110 to PXA255 Processor Software Migration Issues

The difficulty of migrating software from the SA-1110 to the PXA255 processor depends on the amount of hardware and software interaction. SA-1110 applications running under an Operating System, which use device driver interfaces, should move seamlessly between the two devices.

There is one exception; any application that explicitly uses the Read Buffer to prefetch external memory data into the SA-1110. This buffer does not exist on the PXA255 processor and register #9 in Coprocessor #15 that was used to access it are not compatible to software.

As the Read Buffer prefetching activity was deemed to be a hint rather than an instruction, applications can simply delete all references to the Read Buffer and still function correctly. They may not even suffer a performance penalty, as the PXA255 processor 'hit-under-miss' cache feature can turn the entire data space into a prefetchable region without any explicit software direction.

Alternately, as a patch for software that cannot be modified, all applications must be limited to User Mode execution, whereupon an Exception can be generated for all Coprocessor activity. Such an exception manager needs to filter out the Read Buffer coprocessor calls, or convert them to PXA255 processor PLD instructions that can preload a data cache value.

There are major software difference within the device initialization/configuration software and device drivers, such as low-level code that controls the hardware.

The PXA255 processor has enhanced functionality and extra instructions not found in the SA-1110. The PXA255 processor software is not backward compatibility to the SA-1110. Once code is compiled for the PXA255 processor it is unlikely to run on the SA-1110.

A.2.1 Software Compatibility

Because the PXA255 processor uses Intel® XScale™ microarchitecture, the PXA255 processor has a different pipeline length relative to the SA-1110. This effects code performance when migrating between the two devices varies because of the number of clock cycles needed for execution. Any application that relies on specific cycle counts, or has specific timing components, will show a difference in performance.

The PXA255 processor features: larger caches, Branch target buffering, and faster multiplication, and so many applications run faster than the SA-1110 when running at the same clock frequency.

A.2.2 Address space

The physical address mapping of gross memory regions is not compatible between the PXA255 processor and SA-1110. For example, on the PXA255 processor, static chip selects 4 and 5 are lower in memory than PCMCIA, on the SA-1110 they are higher in the memory space.

Changes of this kind could be managed by the Operating System remapping virtual memory pages to new physical addresses. This assumes that the Operating System has basic support for virtual memory, but not if this could be managed by initialization code modifications effecting the same change.

More significantly, memory-mapped registers may have different names, new addresses and different functionality. This impacts all device drivers and register-level firmware, that at a minimum, requires re-mapping register address and changing the default configuration.

A.2.3 Page Table Changes

There are differences in the virtual memory Page Table Descriptors between the SA-1110 and the PXA255 processors that impact software execution speed. A new bit has been added to differentiate ARM* compliant operation modes from some features Intel includes such as access to the Mini-Data-Cache.

If any software attempts to explicitly control page table modifications, normally the domain of the Operating System, then that software may need annotation to allow for the extra opportunities the PXA255 processor offers.

Any SA-1110 code that explicitly uses the Mini-Data-Cache is executed correctly, but its ability to utilize a different cache is lost without a page table bit being changed. The impact here is performance not functionality.

A.2.4 Configuration registers

There are numerous device configuration changes in the PXA255 processor. You must now select the configuration options for clock speeds such as Turbo Mode. This requirement is not found on the SA-1110.

You must choose memory clocks, LCD clock rates, audio clocks and interfaces, which GPIOs are actually connected to hardware, and many more. There are no easy solutions here, the device space of the PXA255 processor is very diverse and a number of selections must be made in software to select your particular hardware functionality.

All software that controls registers will need to be updated. If a switch is connected to a GPIO, then the software that reads the switch register will differ between the PXA255 processor and SA-1110. This applies to software that controls Configuration registers in Coprocessor space and also to a number of the memory-mapped registers. Many functions such as memory timing configuration are done through these registers. For example the registers to access USB have a different name, address and function. Any code that directly accesses the USB hardware registers will need to be rewritten.

A.2.5 DMA

The SA-1110 contains a 6-channel DMA controller that pipes data between the serial channels and memory. The PXA255 processor provides a far more substantial 16-channel chained DMA controller that can be configured to do much more than the SA-1110, including memory-to-memory block moves.

Clearly there are changes in software required to take advantage of this new asset. However this also implies changes necessary to maintain similar functionality to the SA-1110. To configure a DMA channel you no longer set it to a specific serial port, instead you map it to the specific source or destination address of the serial port FIFO. You must configure other parameters for address incrementing and memory width that differ between the PXA255 processor and SA-1110.

Any device driver using the SA-1110 DMA controller, or application that takes direct advantage of DMA, will need to be modified. The impact of this varies as some Operating Systems and many device drivers have ignored the SA-1110 DMA in favor of programmed I/O.

Some have argued to remove the required interrupt management code as they only move small blocks. Operating systems have excluded DMA to guarantee they can manage the real-time behavior of different threads. Other software providers saw the serial ports as so slow that DMA performance was unnecessarily complex.

The performance benefit of the PXA255 processor DMA controller is one of the most significant improvements over the SA-1110, particularly in the area of memory-to-memory moves. Changing code on the PXA255 processor to utilize the new DMA functionality will significantly enhance applications.

A.3 Using New PXA255 Processor Features

This appendix doesn't attempt to discuss all the differences between the PXA255 processor and the SA-1110 or it would become quite substantial. There are numerous significant advantages that the PXA255 processor has to offer, all of which potentially require changes in hardware, firmware or software development tools. This section lists just a few of the chief additional benefits of the PXA255 processor. However, refer to the product specifications for further details. This list is not comprehensive.

A.3.1 Intel® XScale™ Microarchitecture

The PXA255 processor is a system on a chip that includes Intel's new microprocessor megacell. This includes Intel® Superpipelined Technology and a new optimized cache architecture that allows program execution to continue despite data cache misses.

The PXA255 processor supports:

- ARM® Architecture v5 instructions, including ARM's Thumb extensions
- DSP Extensions but not ARM's optional Vector Floating Point instructions

Appendix A in the Intel 80200 Developers' manual, Order# 273411-002 [<http://developer.intel.com>], is the best guide to the new capabilities available in the Intel® XScale™ Megacell's Instruction Set.

Using a software development toolset that takes specific advantage of Intel® XScale™ microarchitecture, and Intel® Media Processing Technology could give you substantial performance benefits.

The PXA255 processor offers increased performance at similar clock rates, and also a wider range of operating clock rates at lower voltages. The overall benefit is more work done for less battery power.

A.3.2 Debugging

New PXA255 processor hardware creates new debugging possibilities. You can use the JTAG test port to download programs into a dedicated memory area to act as a debug monitor. Applications can be inspected and performance data, such as cache hit rates, can be measured via a dialog over JTAG. These features offer developers far more visibility inside a PXA255 processor system improving time to market.

A.3.3 Cache Attributes

The PXA255 processor has twice the instruction cache and four times the data cache of the SA-1110. The Caches can be locked for optimized code or data and for reliability the caches are now covered by parity protection.

To take advantage of cache locking software, data must be selected and specifically loaded and locked into cache.

To take advantage of new features such as Write-Through mode for external IO buffers, page tables will need to be revisited in boot software.

A.3.4 Other features

As mentioned before, the PXA255 processor DMA controller is highly versatile. With 16 channels it can be utilized as:

- Several serial ports in parallel
- A general-purpose memory move capability
- A fast interface for external companion devices

Additional software is required to access these benefits.

A similar story is true for AC'97, I2C, GPIOs, MMC and others. The benefits are substantial, but new hardware and software will be necessary to effectively use the PXA255 processor.

A.3.5 Conclusion

Although most application software will migrate unchanged between SA-1110 and the PXA255 processors, the underlying microarchitectures are radically different. The PXA255 processor being considerably more ornate and capable than the SA-1110. The majority of hardware and low-level software from any SA-1110 design will need to be redesigned. with a view to taking full advantage of the new features that the PXA255 processor brings to the handheld market.

Example Form Factor Reference Design

Schematic Diagrams B

B.1 Notes

The example form factor reference design schematics in this appendix have known issues and assumptions that need to be assessed for each board design. This appendix documents the issues that have been discovered and provides revision data for the schematics. This appendix also points out some of the design specific assumptions that were made in designing this board.

Page 4: The schematic supports the SA1110 legacy mode for addressing the SDRAM. See SDRAM section of the design guidelines to explain normal vs. legacy addressing modes.

Page 10: U26 allows the CF card to access the I²C interface.

Page 11: The MMC slot has resistor jumpers to select SDCARD and/or MMC card support. If the users wants to support both SCARD and MMC, populate the resistor values in the SD column of the CARD selection table. If MMC card support only populate the resistors under the MMC column in the table.

Page 11: The JTAG port on this board assumes that it is connected to a specific JTAG bridge board. If you follow this design guide, you should not need the bridge board. See the JTAG section of this document for more details on this bridge board implementation.

Page 11: J19 pin 9 requires a 1.5 k pull down.

Page 13: U33, U34, U35, U40, and surrounding circuits are to support a legacy sharp LCD. Refer to that vendor's design guidelines if you are integrating a different LCD display.

Page 14: U36, U37, U39, J13, and surrounding circuits including the resistor divider network in the upper left hand corner support a legacy Sharp LCD. Refer to that vendor's design guidelines if you are integrating a different LCD display.

Page 14: J14 is the Toshiba LCD connector. Confirm the pinout before layout.

B.2 Schematic Diagrams

The example form factor reference design schematics for the PXA255 processor are on the following pages. The PXA255 processor is a drop in replacement and is fully compatible with these schematics.

Example Form Factor Reference Design for PXA250

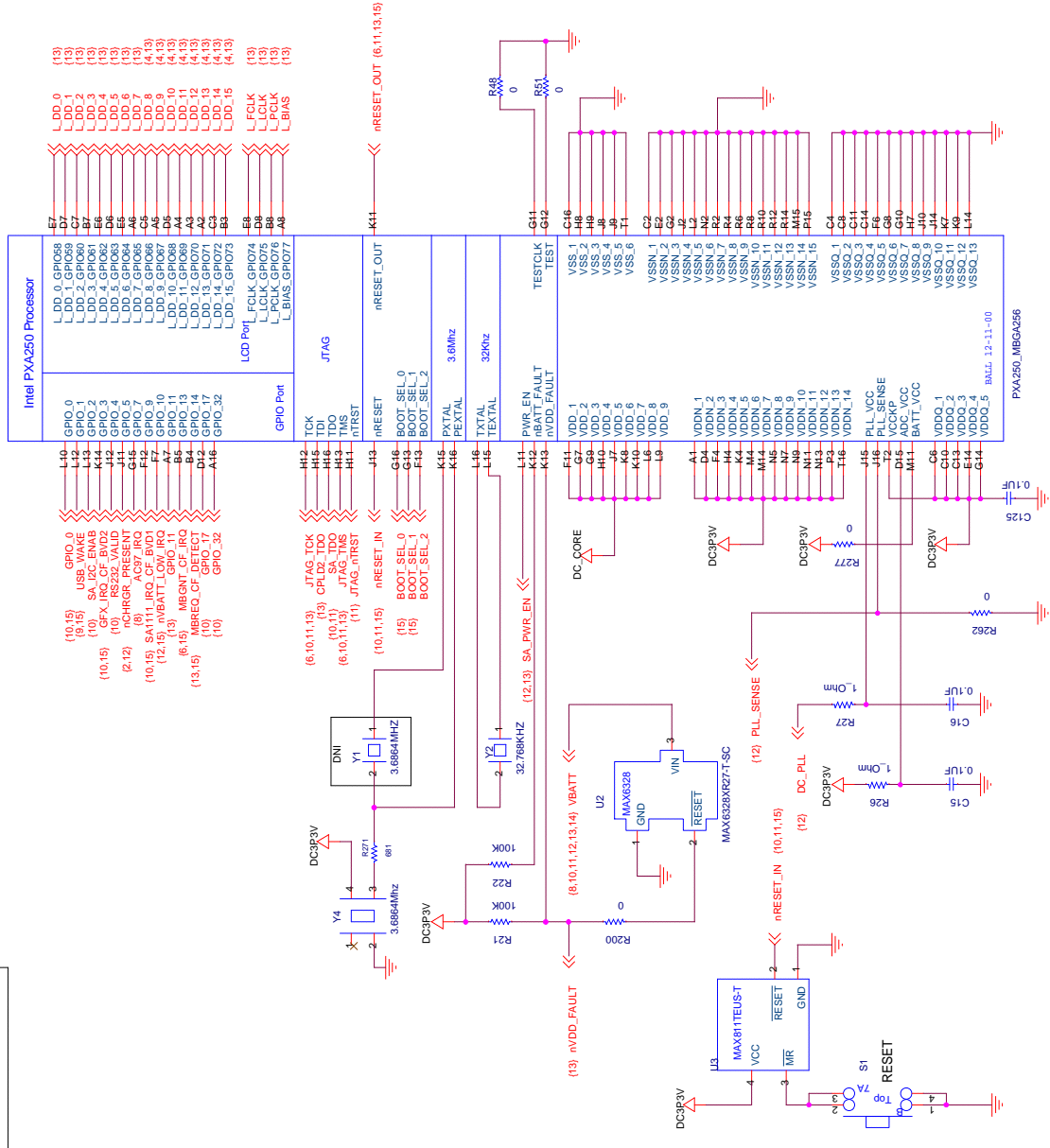
Page Description

- 1 Cover Sheet
- 2-3 PXA250 Processor
- 4 SDRAM, System Configuration Register
- 5 Intel Flash Memory (BGA)
- 6 Buffer, CPLD, Board Control Register
- 7 Transceivers
- 8 Audio Codec, Audio AMP
- 9 Headset Jack, Microphone, Stereo Jack, Speaker, Dual Axis Accelerometer, IrDA, USB
- 10 Basestation Hdr, RS232 Xcvr, CF Socket, Function Switches
- 11 SD socket, Radio Header, JTAG Port
- 12 Boost Buck Power , Battery Header, Battery On Jumper
- 13 LCD CPLD, LCD Power, Back Light Connector
- 14 LCD Connectors, 9 Channel Buffer, Touch Screen Connector
- 15 Bus Connectors, 2-140 Pin
- 16 Schematic Revision Page

Rev 2.07

PXA250 Processor Reference Design

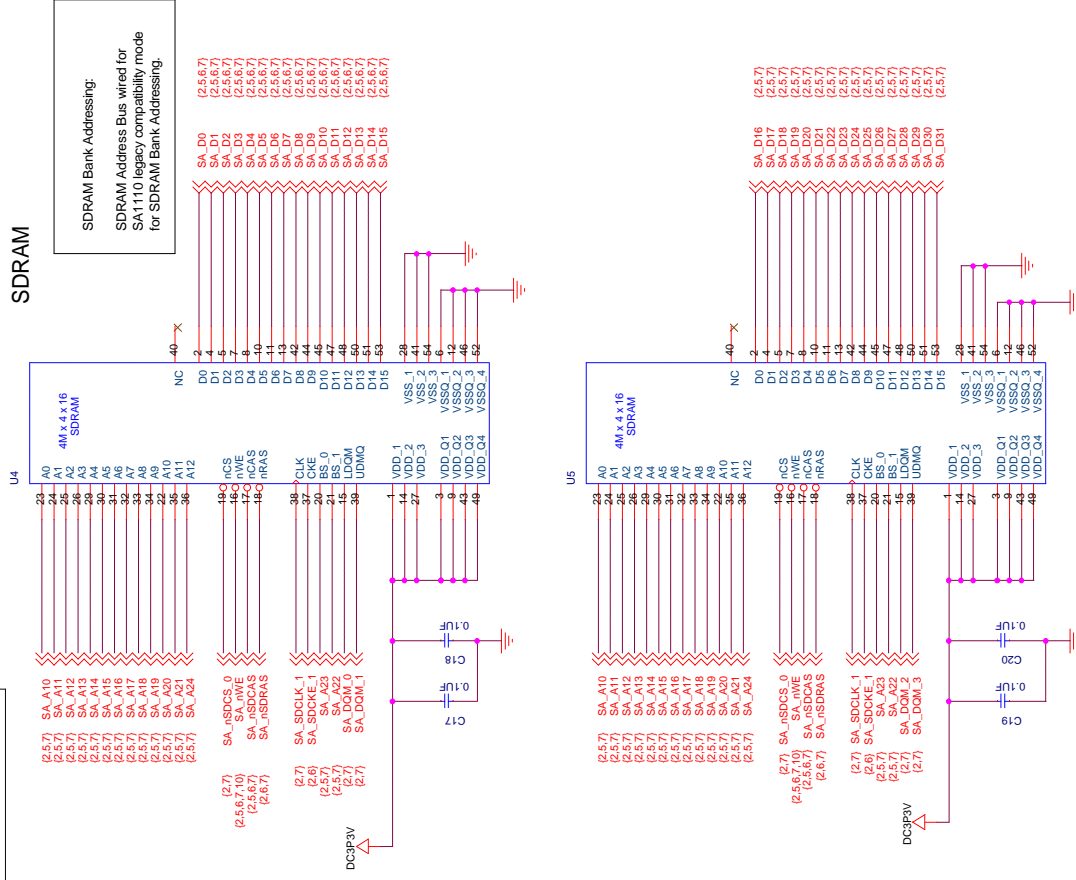
Size B	Rev 2.07
Date: Tuesday, February 05, 2002	
Sheet	1 of 16



PXA250 Processor Reference Design

Size B	Rev 2.07
Date: Tuesday, February 05, 2002	Sheet 3 of 16

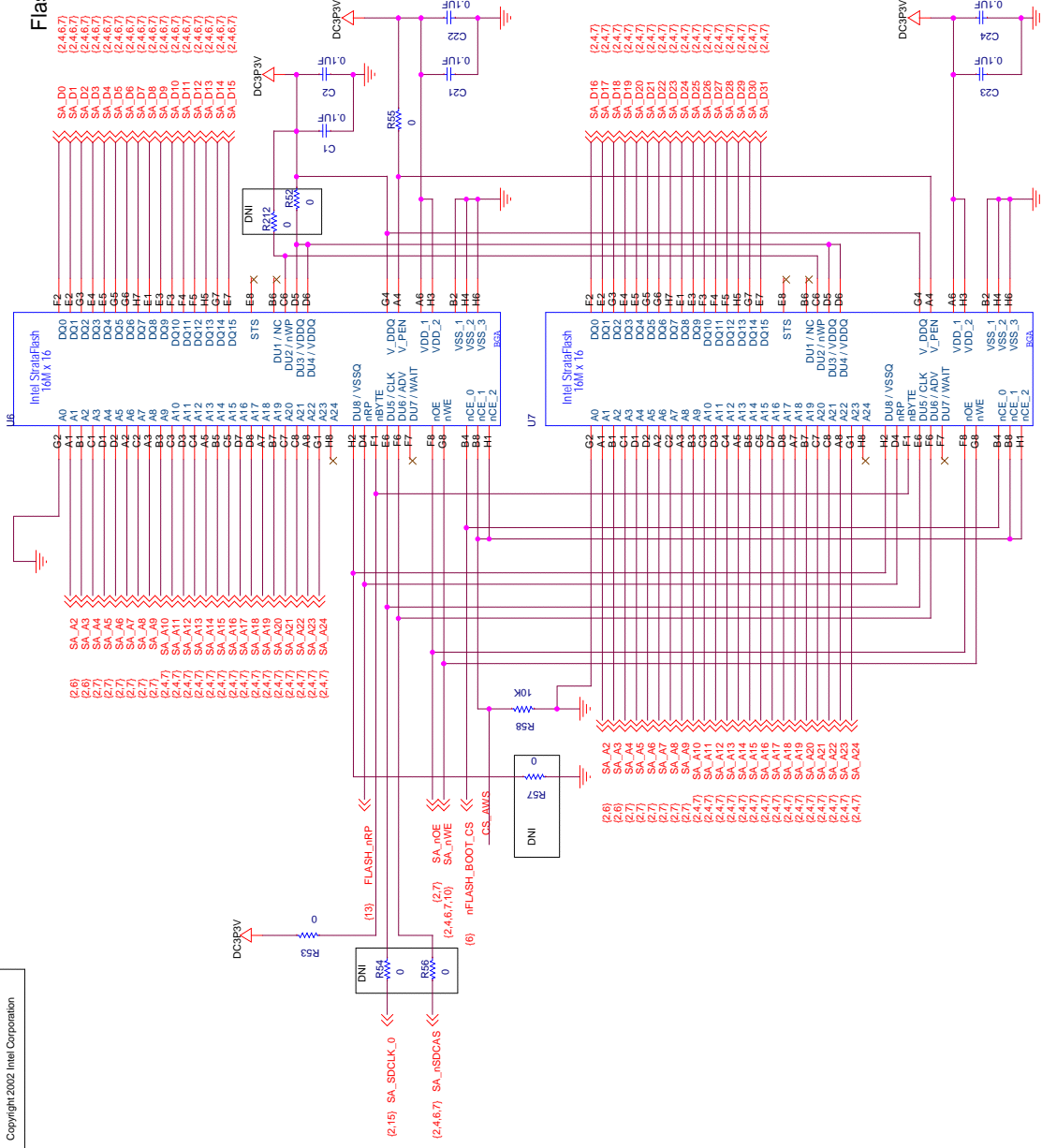
SYSTEM CONFIGURATION REGISTER



PXA250 Processor Reference Design	
Size B	Rev 2.07
Date: Tuesday, February 05, 2002	Sheet 4 of 16

Flash Memory

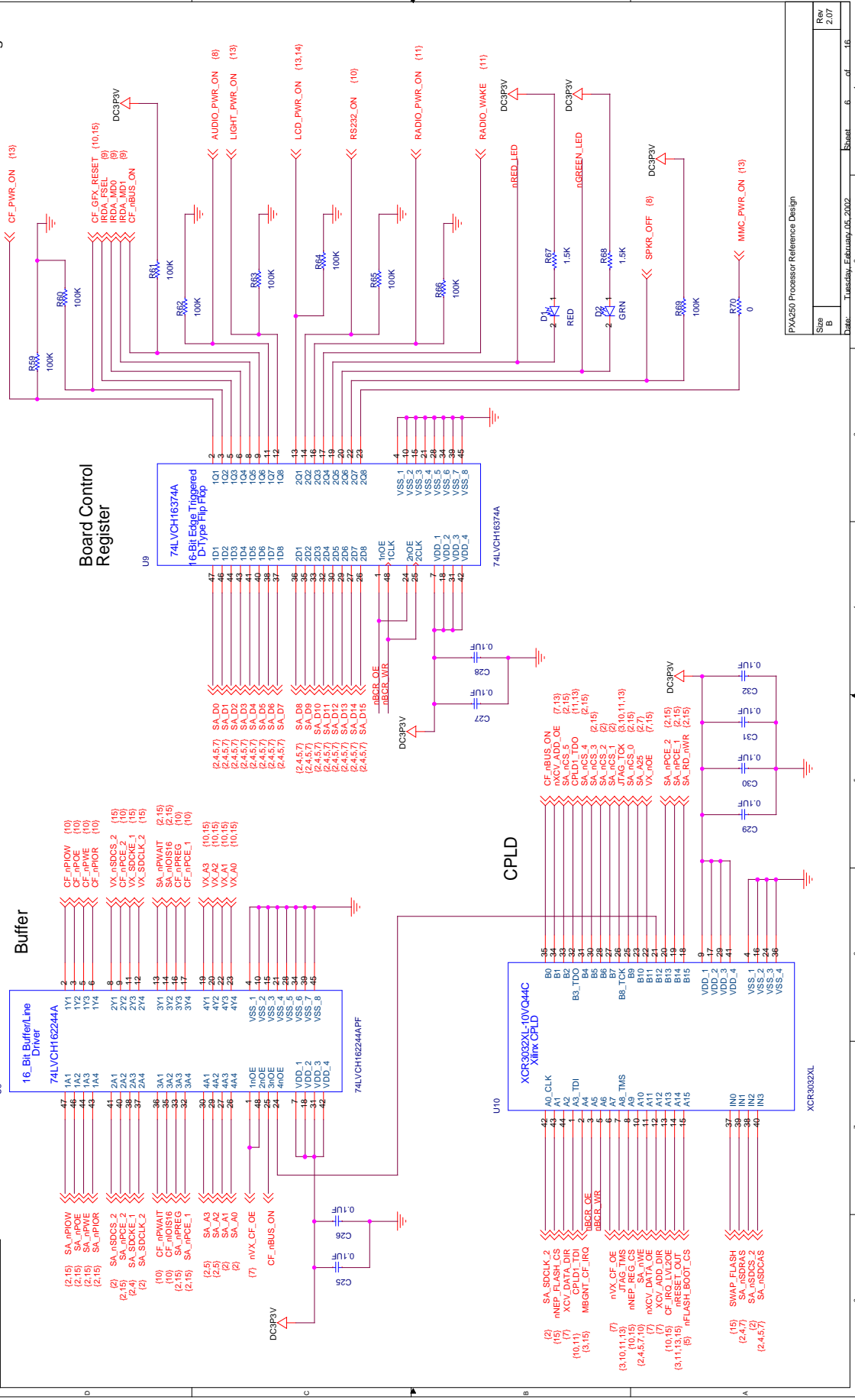
Resistor	StrataFlash	Synchronous StrataFlash
R52	DNI	IN
R53	IN	DNI
R54	DNI	IN
R56	DNI	IN
R57	DNI	IN
R212	DNI	IN



PXA250 Processor Reference Design

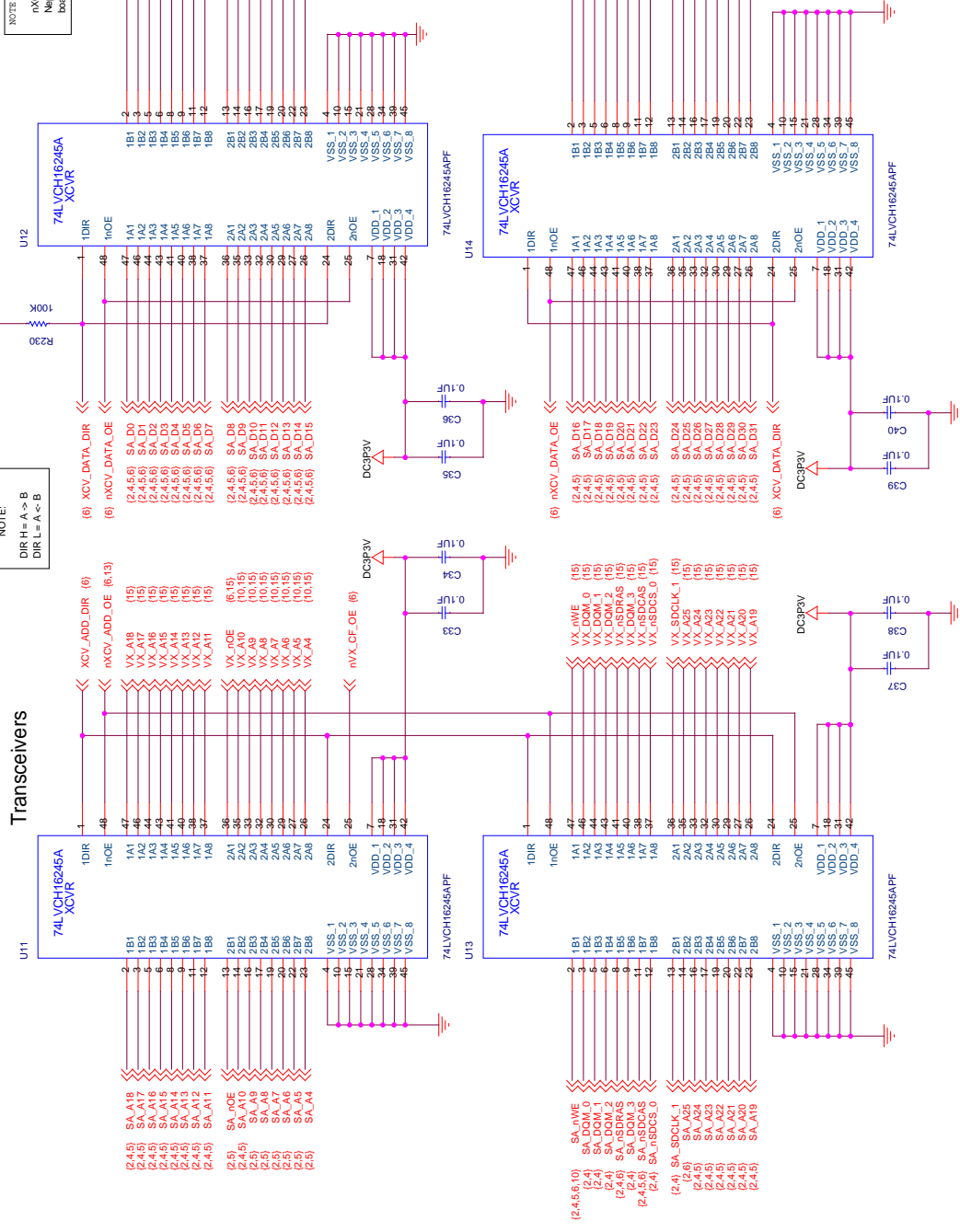
Size	Rev
B	2.07

Date: Tuesday, February 05, 2002 Sheet 5 of 16

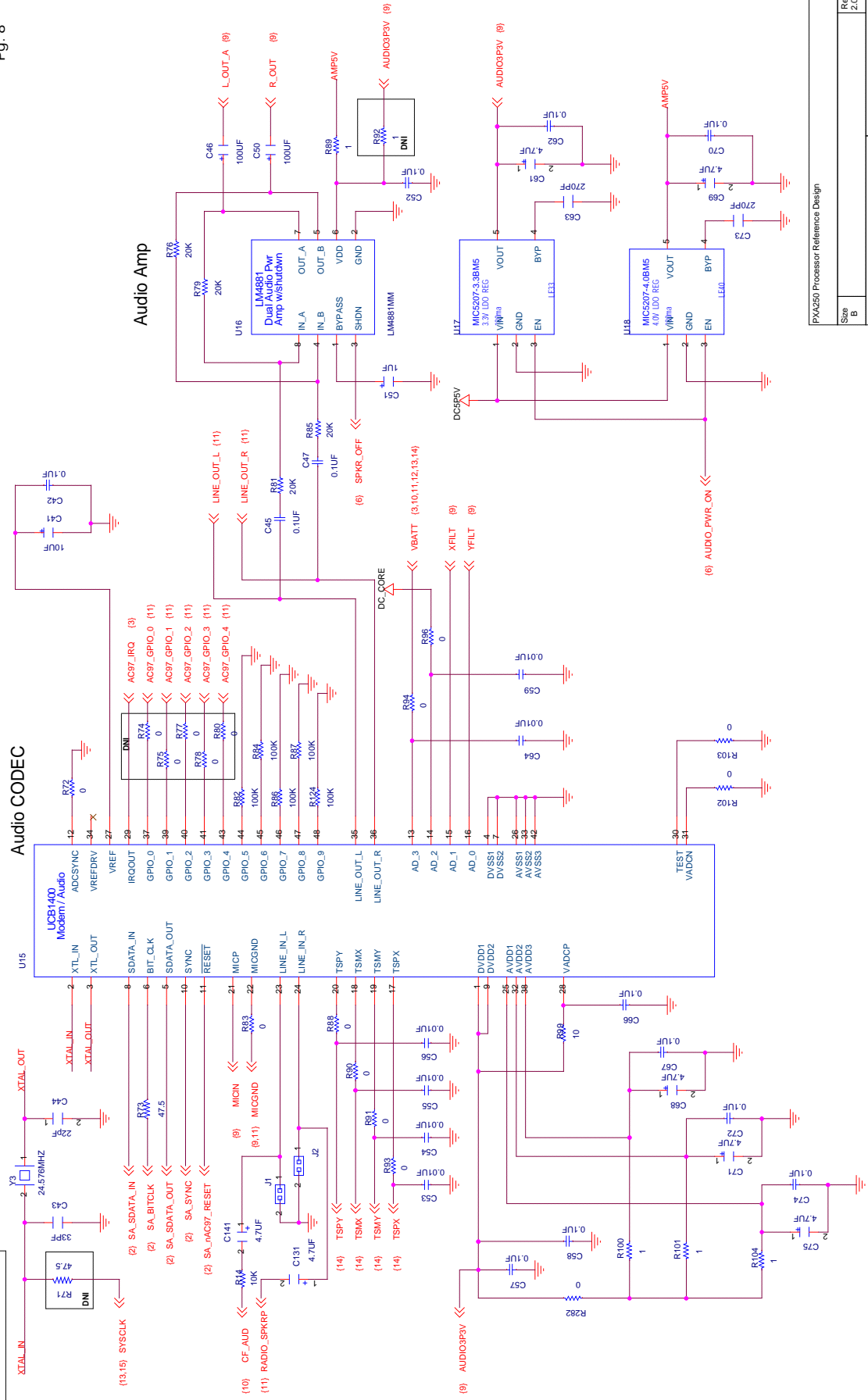


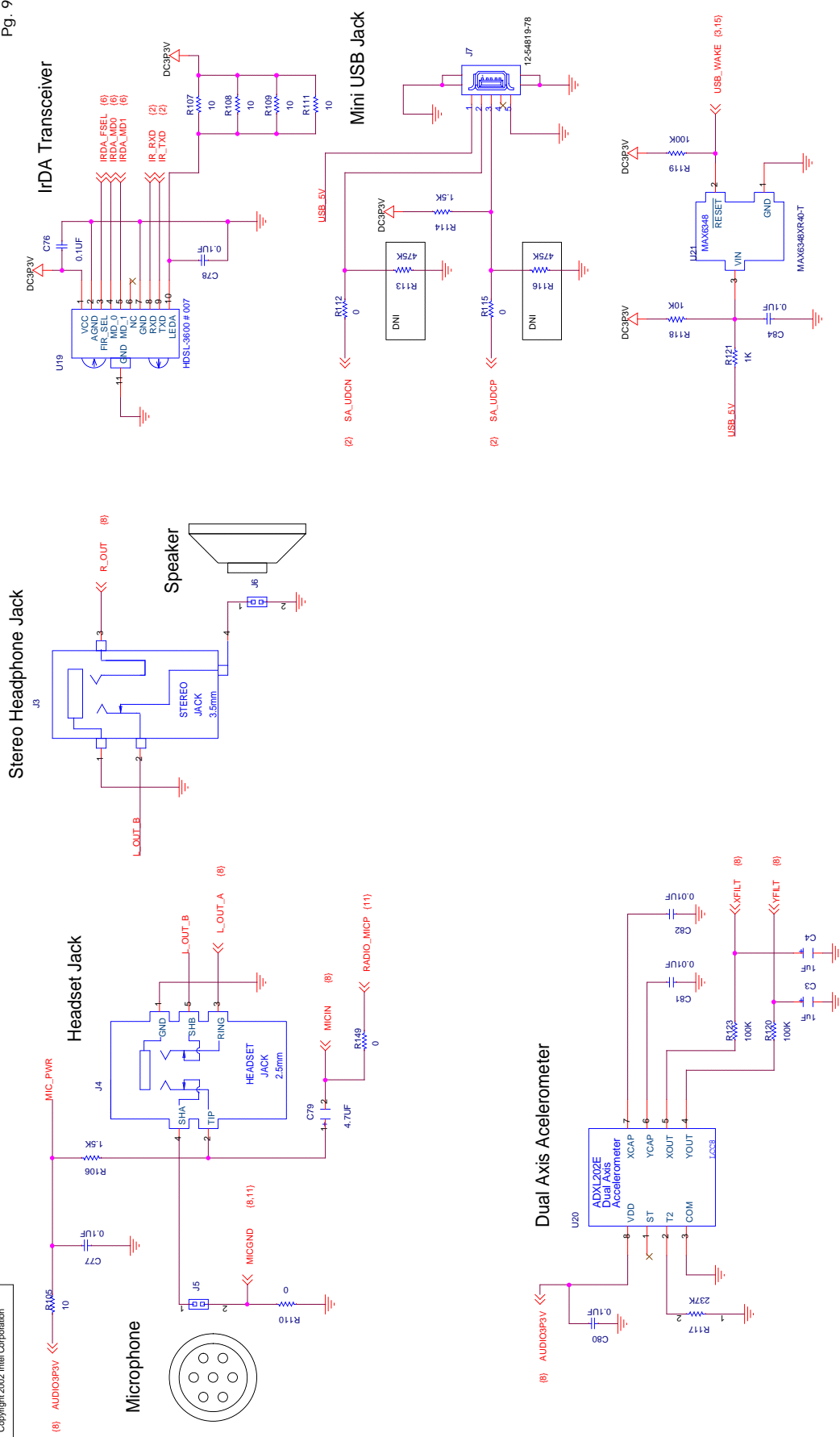
Transceivers

NOTE:
DIR H = A > B
DIR L = A < B



NOTE:
nXCV_ADD_OE is Low when
Neponset or Graphics
boards are present.





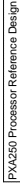
PVA250 Processor Reference Design

Size	Rev
B	2.07

Date: Tuesday, February 05, 2002

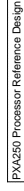
Sheet 9 of 16

Momentary Switches

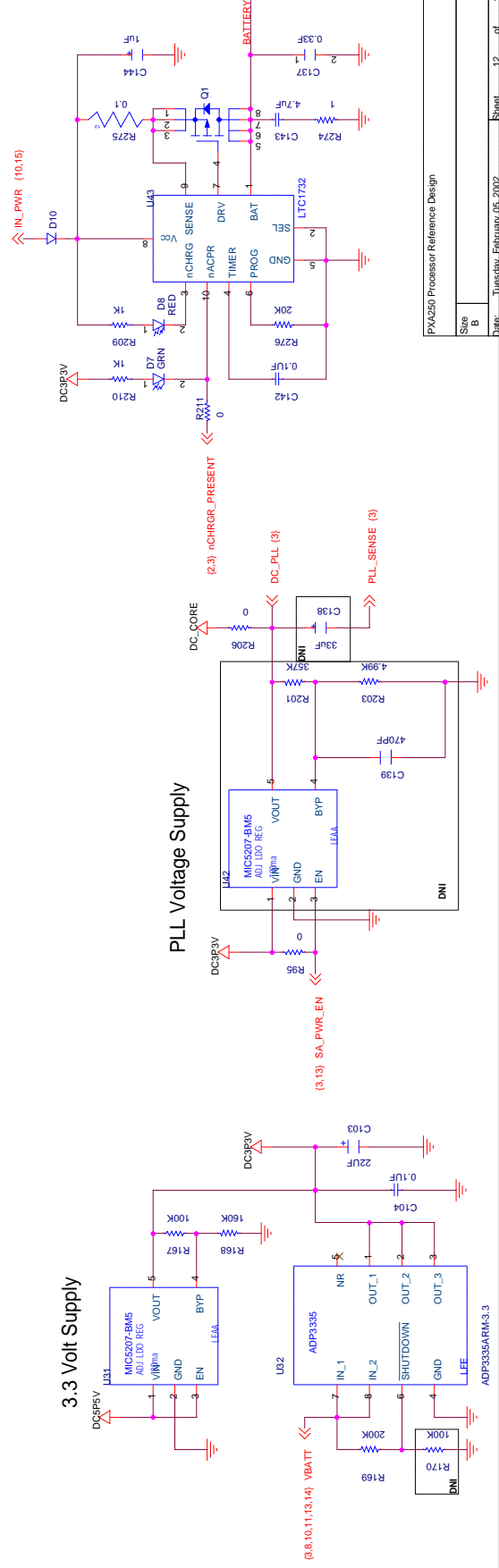
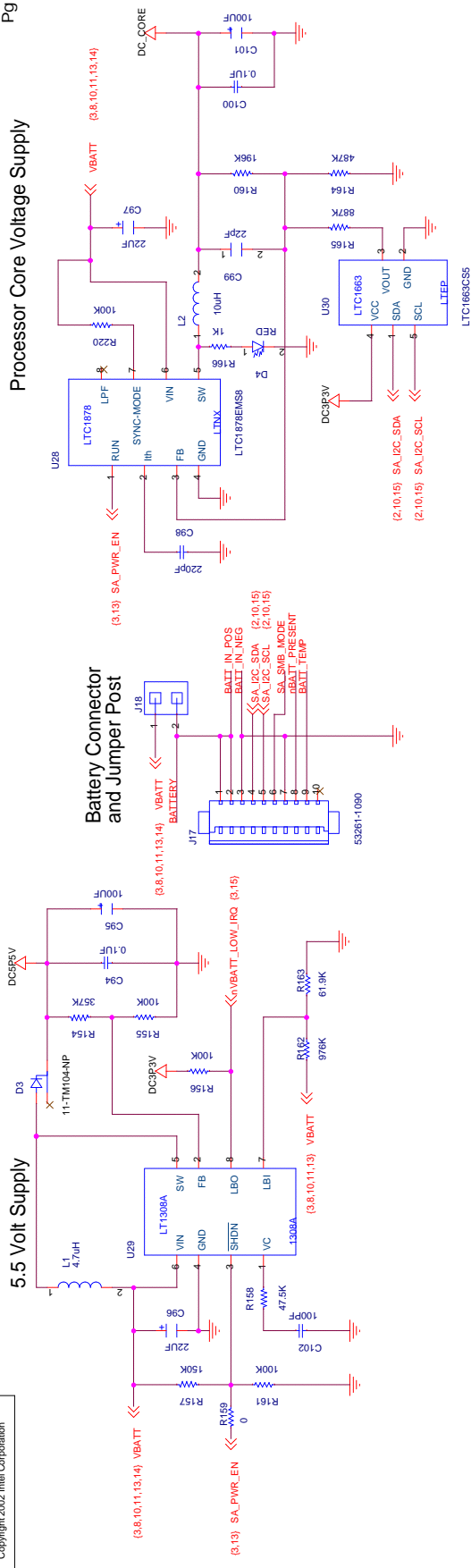


Size B	Rev 2.07
-----------	-------------

Date:	Tuesday, February 05, 2002	Sheet	10	of
-------	----------------------------	-------	----	----



Size B	Rev 2.07
Date: Tuesday, February 05, 2002	Sheet 11 of 16





Note: On board Back Light Inverter for Toshiba Display

BL-0204

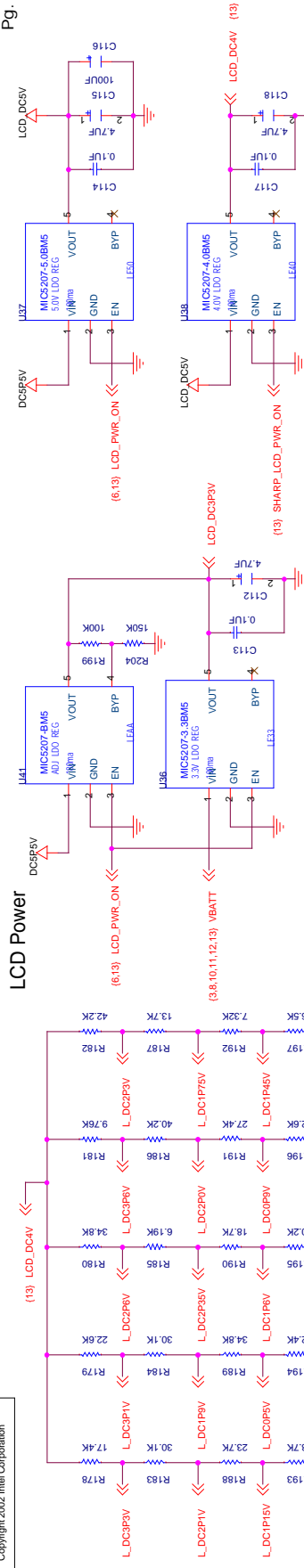
H	CN2-1
L	CN2-2

ELECTRIC INVERTER

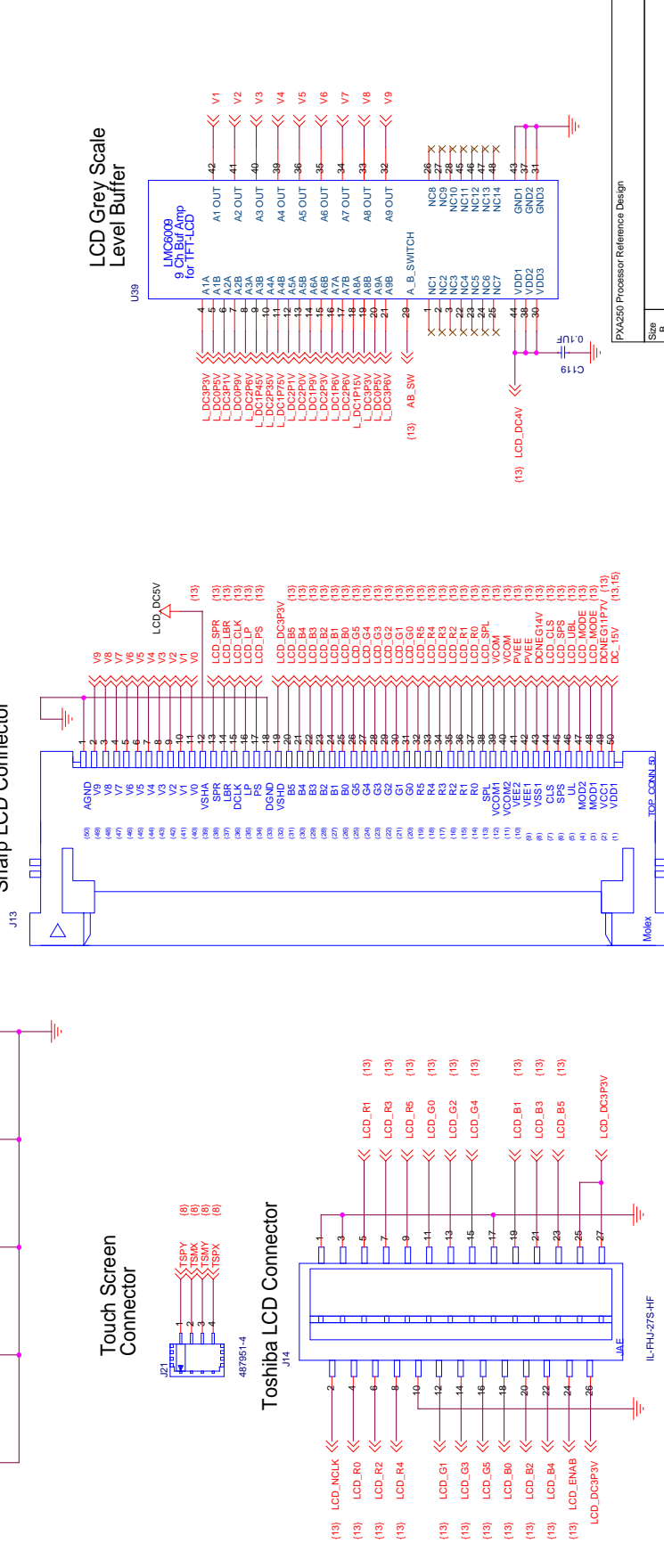
PXA250 Processor Reference Design

Rev	2.07
-----	------

Date: Tuesday, February 05, 2002

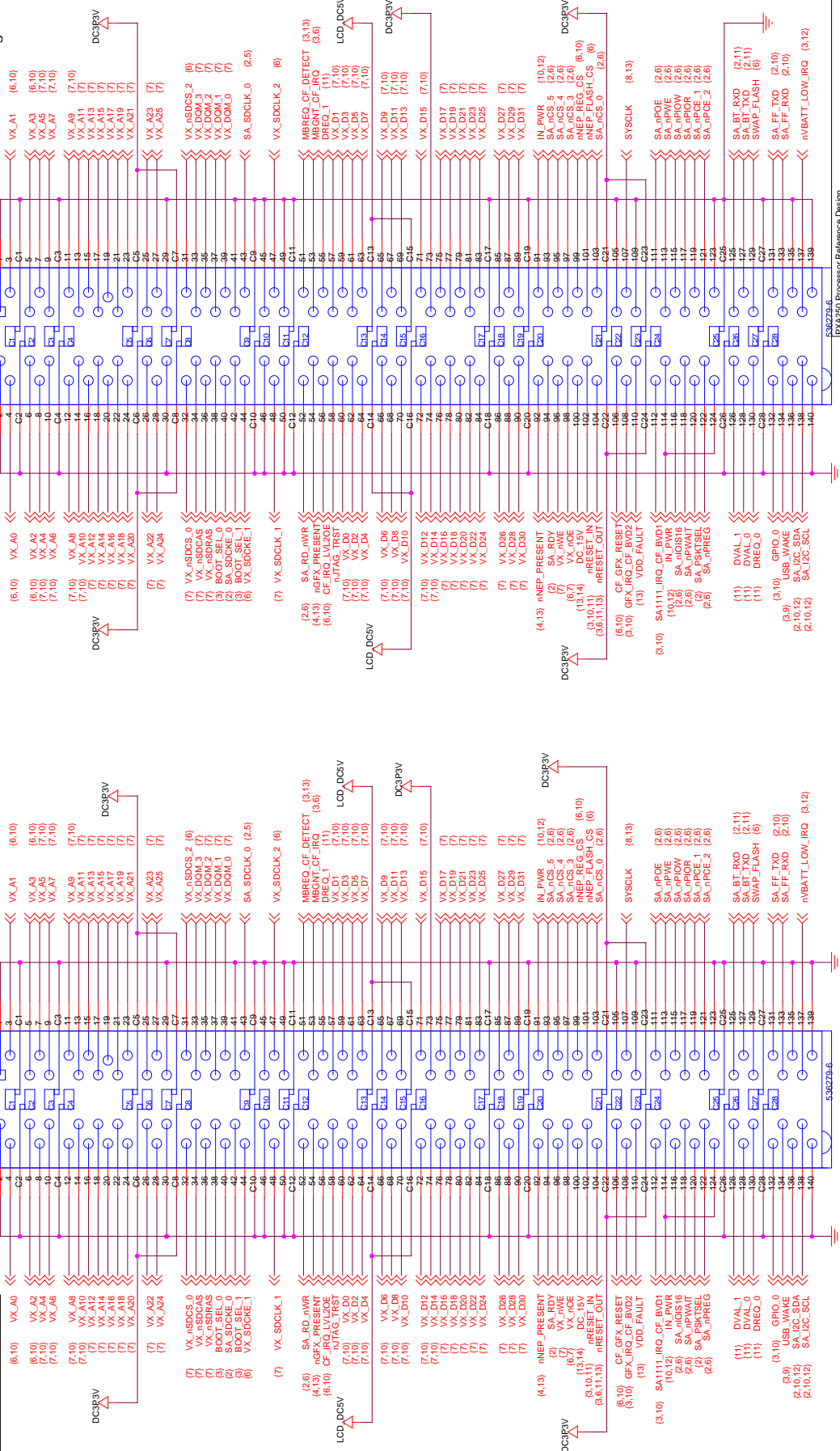


Sharp LCD Connector



PXA250 Processor Reference Design

Size B	Rev 2.07
Date: Tuesday, February 05, 2002	Sheet 14 of 16



Expansion Header 1

Expansion Header 2

Size B	Rev 2.07
-----------	-------------

Revision Tracking Changes
R 2.00 - 09-21-01 - Initial Release
R 2.01 - 09-25-01 - CHRGR_PRESENT signal changed to nCHRGR_PRESENT to properly represent active low signal - Pages 2, 3 & 12

- | | | |
|--------|----------|--|
| R.2.01 | 09-25-01 | - CHRG8 PRESENT signal changed to nCHRG8_PRESENT to properly represent active low signal - Pages 2, 3 & 12 |
| R.2.01 | 09-25-01 | - INTRG PRESENT |
| R.2.01 | 09-25-01 | R281 - nCHRG8_PRESENT 100k pull-up added - Page 2 |
| R.2.01 | 09-25-01 | R282 - U1A SEVEN pin removed from DNI test - Page 3 |
| R.2.01 | 09-25-01 | C6, C7, C8, R283, R284, R285, R286, R287, R288, R289, R290, R291, R292, R293, R294, R295, R296, R297, R298, R299, R300, R301, R302, R303, R304, R305, R306, R307, R308, R309, R310, R311, R312, R313, R314, R315, R316, R317, R318, R319, R320, R321, R322, R323, R324, R325, R326, R327, R328, R329, R330, R331, R332, R333, R334, R335, R336, R337, R338, R339, R340, R341, R342, R343, R344, R345, R346, R347, R348, R349, R350, R351, R352, R353, R354, R355, R356, R357, R358, R359, R360, R361, R362, R363, R364, R365, R366, R367, R368, R369, R370, R371, R372, R373, R374, R375, R376, R377, R378, R379, R380, R381, R382, R383, R384, R385, R386, R387, R388, R389, R390, R391, R392, R393, R394, R395, R396, R397, R398, R399, R400, R401, R402, R403, R404, R405, R406, R407, R408, R409, R410, R411, R412, R413, R414, R415, R416, R417, R418, R419, R420, R421, R422, R423, R424, R425, R426, R427, R428, R429, R430, R431, R432, R433, R434, R435, R436, R437, R438, R439, R440, R441, R442, R443, R444, R445, R446, R447, R448, R449, R450, R451, R452, R453, R454, R455, R456, R457, R458, R459, R460, R461, R462, R463, R464, R465, R466, R467, R468, R469, R470, R471, R472, R473, R474, R475, R476, R477, R478, R479, R480, R481, R482, R483, R484, R485, R486, R487, R488, R489, R490, R491, R492, R493, R494, R495, R496, R497, R498, R499, R500, R501, R502, R503, R504, R505, R506, R507, R508, R509, R510, R511, R512, R513, R514, R515, R516, R517, R518, R519, R520, R521, R522, R523, R524, R525, R526, R527, R528, R529, R530, R531, R532, R533, R534, R535, R536, R537, R538, R539, R540, R541, R542, R543, R544, R545, R546, R547, R548, R549, R550, R551, R552, R553, R554, R555, R556, R557, R558, R559, R560, R561, R562, R563, R564, R565, R566, R567, R568, R569, R570, R571, R572, R573, R574, R575, R576, R577, R578, R579, R580, R581, R582, R583, R584, R585, R586, R587, R588, R589, R590, R591, R592, R593, R594, R595, R596, R597, R598, R599, R600, R601, R602, R603, R604, R605, R606, R607, R608, R609, R610, R611, R612, R613, R614, R615, R616, R617, R618, R619, R620, R621, R622, R623, R624, R625, R626, R627, R628, R629, R630, R631, R632, R633, R634, R635, R636, R637, R638, R639, R640, R641, R642, R643, R644, R645, R646, R647, R648, R649, R650, R651, R652, R653, R654, R655, R656, R657, R658, R659, R660, R661, R662, R663, R664, R665, R666, R667, R668, R669, R670, R671, R672, R673, R674, R675, R676, R677, R678, R679, R680, R681, R682, R683, R684, R685, R686, R687, R688, R689, R690, R691, R692, R693, R694, R695, R696, R697, R698, R699, R700, R701, R702, R703, R704, R705, R706, R707, R708, R709, R710, R711, R712, R713, R714, R715, R716, R717, R718, R719, R720, R721, R722, R723, R724, R725, R726, R727, R728, R729, R730, R731, R732, R733, R734, R735, R736, R737, R738, R739, R740, R741, R742, R743, R744, R745, R746, R747, R748, R749, R750, R751, R752, R753, R754, R755, R756, R757, R758, R759, R760, R761, R762, R763, R764, R765, R766, R767, R768, R769, R770, R771, R772, R773, R774, R775, R776, R777, R778, R779, R780, R781, R782, R783, R784, R785, R786, R787, R788, R789, R790, R791, R792, R793, R794, R795, R796, R797, R798, R799, R800, R801, R802, R803, R804, R805, R806, R807, R808, R809, R810, R811, R812, R813, R814, R815, R816, R817, R818, R819, R820, R821, R822, R823, R824, R825, R826, R827, R828, R829, R830, R831, R832, R833, R834, R835, R836, R837, R838, R839, R840, R841, R842, R843, R844, R845, R846, R847, R848, R849, R850, R851, R852, R853, R854, R855, R856, R857, R858, R859, R860, R861, R862, R863, R864, R865, R866, R867, R868, R869, R870, R871, R872, R873, R874, R875, R876, R877, R878, R879, R880, R881, R882, R883, R884, R885, R886, R887, R888, R889, R890, R891, R892, R893, R894, R895, R896, R897, R898, R899, R900, R901, R902, R903, R904, R905, R906, R907, R908, R909, R910, R911, R912, R913, R914, R915, R916, R917, R918, R919, R920, R921, R922, R923, R924, R925, R926, R927, R928, R929, R930, R931, R932, R933, R934, R935, R936, R937, R938, R939, R940, R941, R942, R943, R944, R945, R946, R947, R948, R949, R950, R951, R952, R953, R954, R955, R956, R957, R958, R959, R960, R961, R962, R963, R964, R965, R966, R967, R968, R969, R970, R971, R972, R973, R974, R975, R976, R977, R978, R979, R980, R981, R982, R983, R984, R985, R986, R987, R988, R989, R990, R991, R992, R993, R994, R995, R996, R997, R998, R999, R1000, R1001, R1002, R1003, R1004, R1005, R1006, R1007, R1008, R1009, R1010, R1011, R1012, R1013, R1014, R1015, R1016, R1017, R1018, R1019, R1020, R1021, R1022, R1023, R1024, R1025, R1026, R1027, R1028, R1029, R1030, R1031, R1032, R1033, R1034, R1035, R1036, R1037, R1038, R1039, R1040, R1041, R1042, R10 |





BBPXA2xx Development Baseboard Schematic Diagram

C

C.1 Schematic Diagram

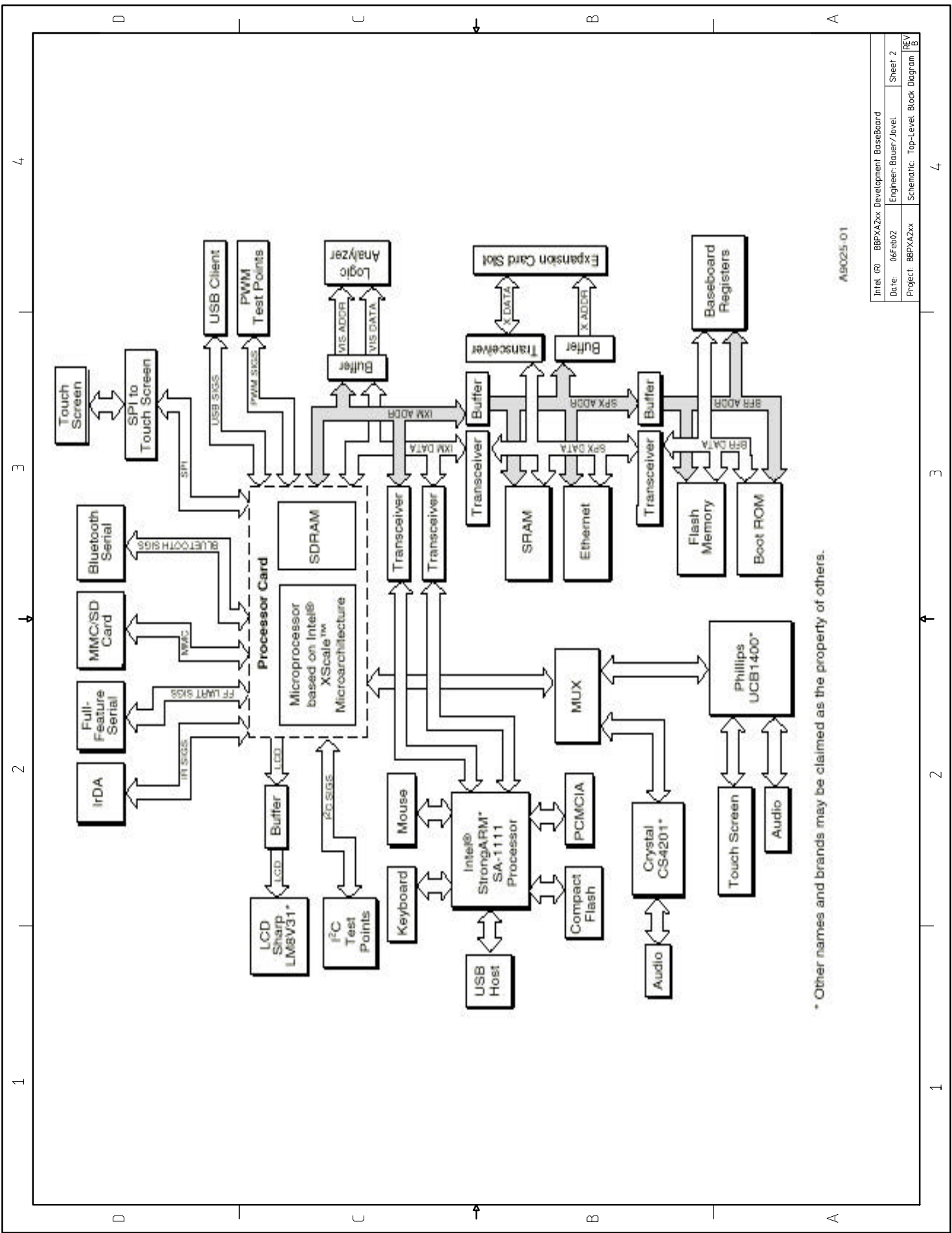
The BBPXA2xx schematic is on the following pages.

BBPXA2xx

Table of Contents

Page	Function	Page	Function	Page	Function
2	Top level block diagram	18	SWITCHES	34	LOGIC ANALYZER
3	Processor Card Connector	19	SPII SPROM	35	RESET & FAULT SWITCHES
4	Data Buffers/Transceivers	20	SA-1111 INTERFACE	36	3.3V & 5V SUPPLY
5	Data Buffers/Transceivers	21	CF & PCMCIA I/O	37	POWER SUPPLY INPUT
6	Address BUFFERS/TRANSCIEVERS	22	CF & PCMCIA PU	38	SPARES
7	SPX ADDRESS BUFFERS	23	SA-1111 USB KB/MS		
8	CONTROL BUFFERS/TRANSCIEVERS	24	BUFFER CONTROL LOGIC		
9	X ADDRESS BUFFERS	25	LCD BUFFERS		
10	BFR ADDRESS BUFFERS	26	IrDA & FF SERIAL & USB		
11	SRAM	27	CS4201 CODEC		
12	FLASH	28	UCB1400 CODEC		
13	ROM	29	BB TOUCH SCREEN & GPIO HEADER		
14	REGISTER & CONTROL	30	CODEC MUX & SPII 2.5V		
15	HEX DISPLAY HIGH	31	SDCARD & BT SERIAL		
16	HEX DISPLAY LOW	32	10Mbps ETHERNET		
17	LEDs & HEX SWITCHES	33	XPANSION PORT		

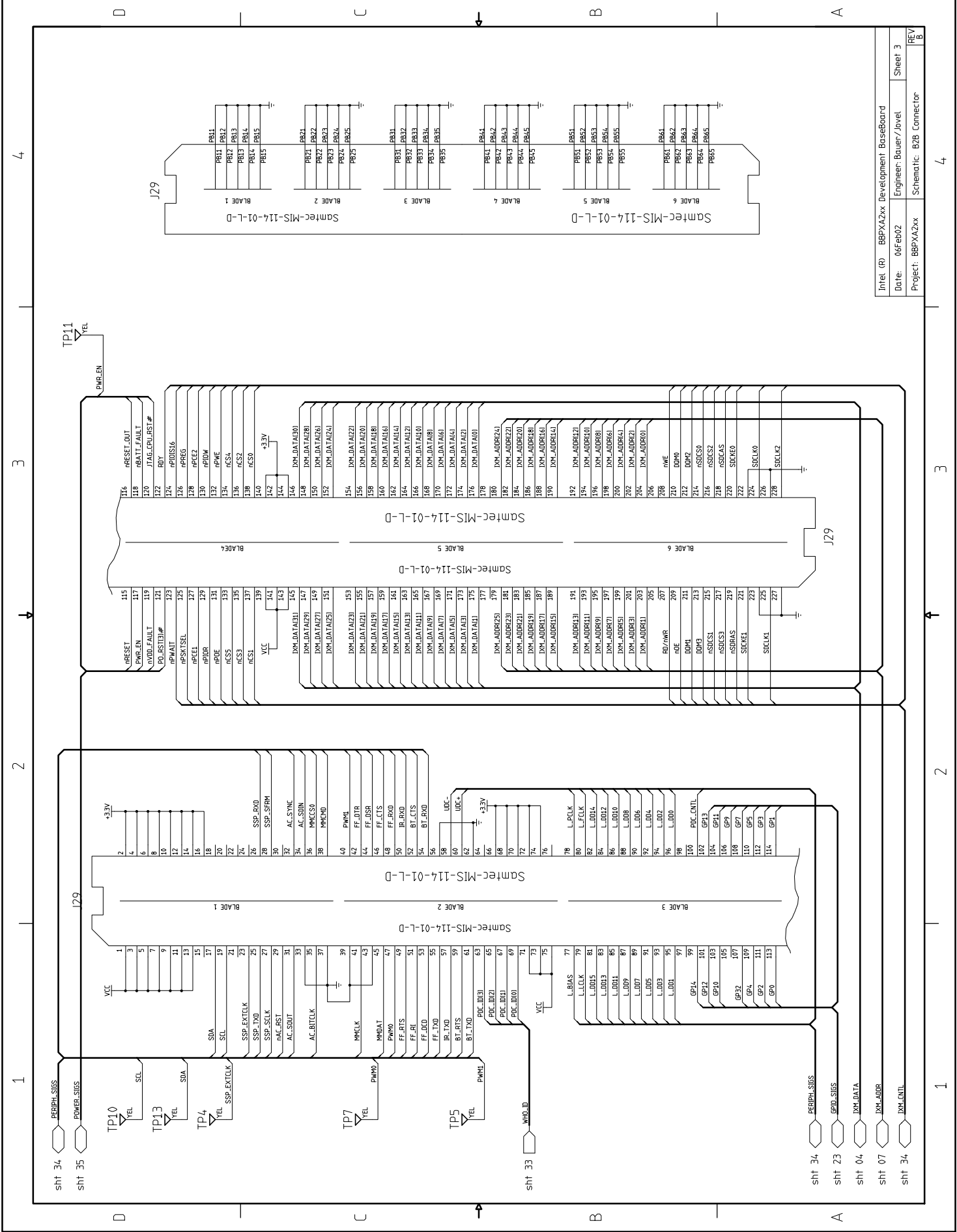
Intel (P)	BBPXA2xx	Development BaseBoard
Date:	06feb02	Engineer: Bauer/Jovel
Project:	BBPXA2xx	Schematic: BBPXA2xx
REV	B	



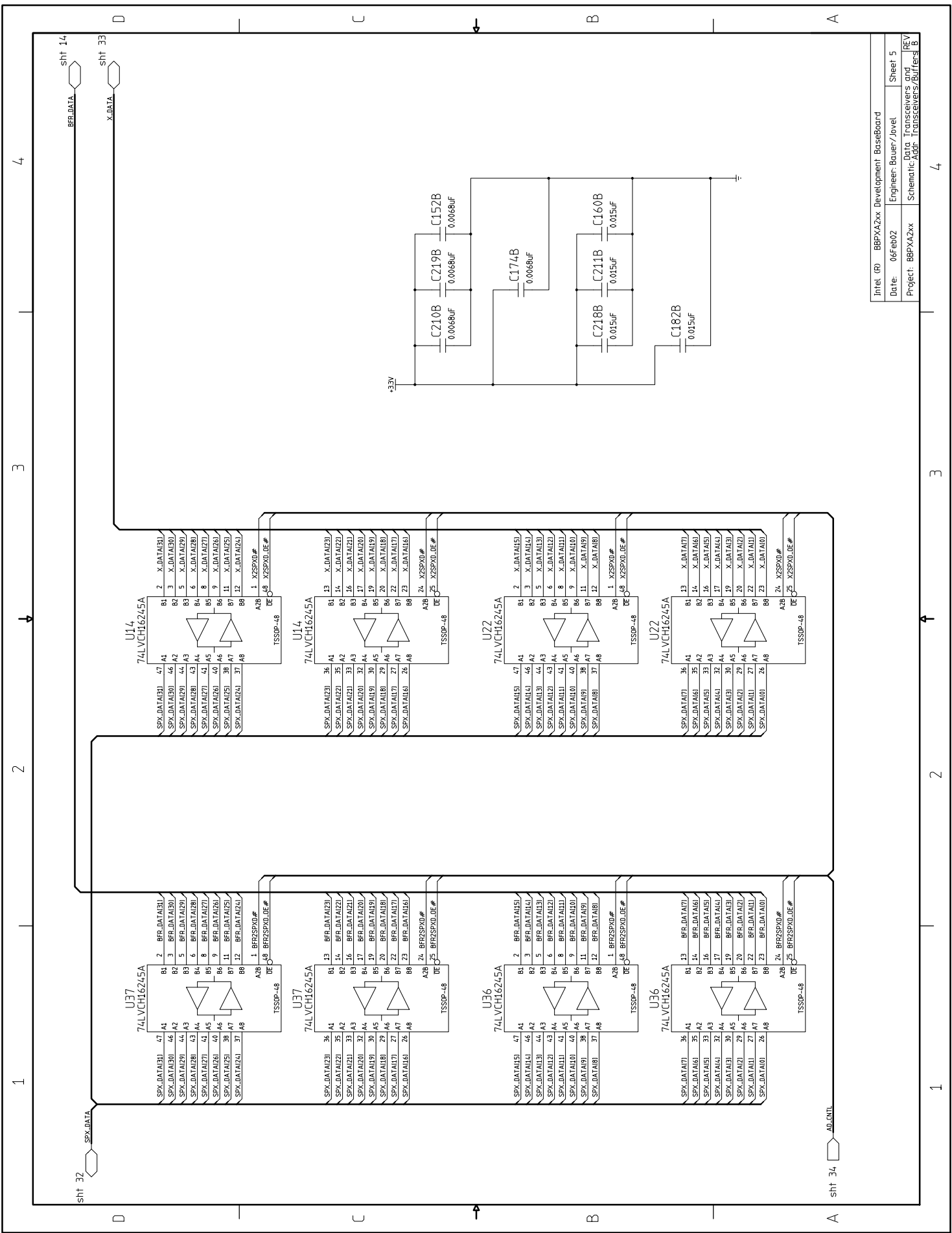
* Other names and brands may be claimed as the property of others.

AS025_01

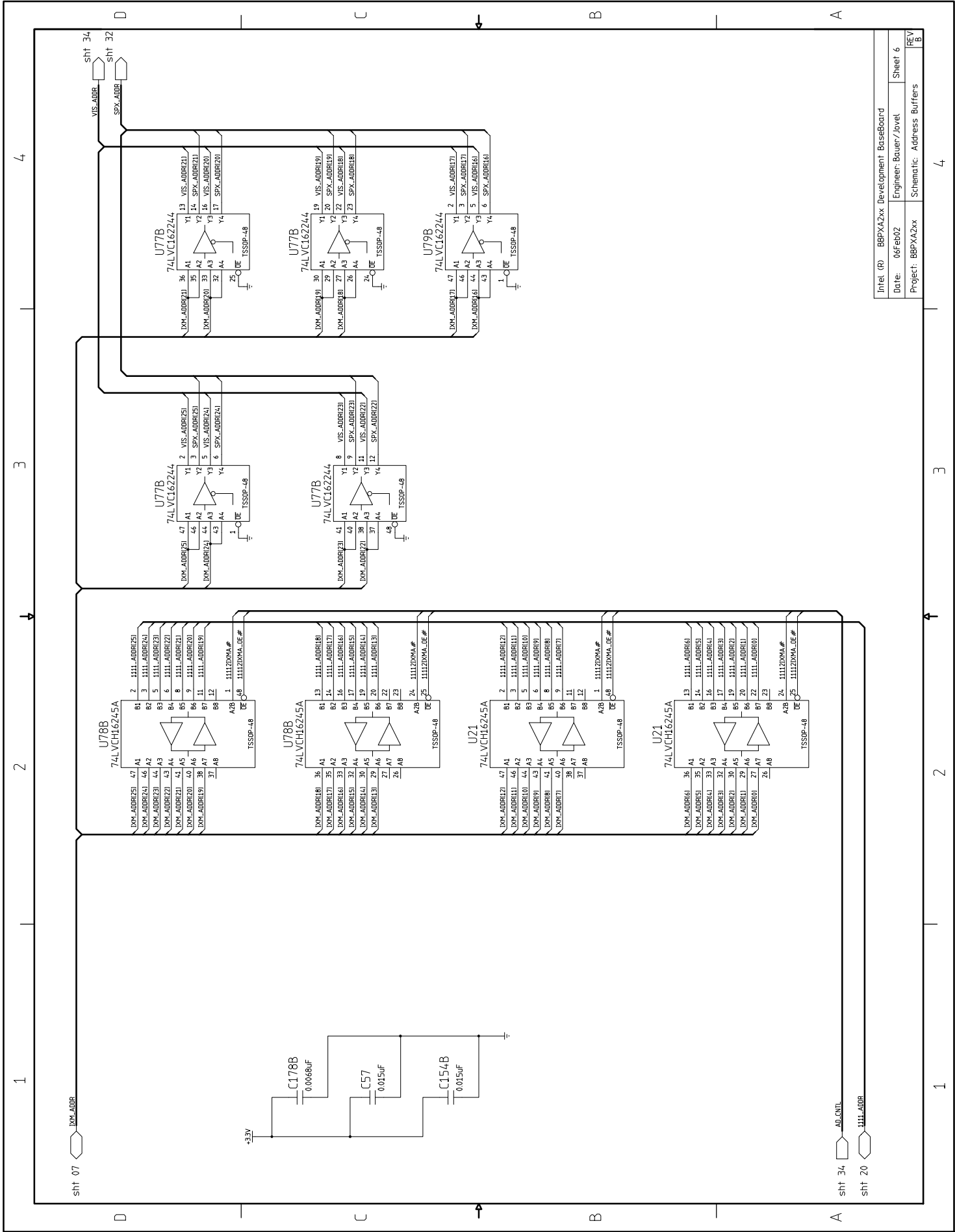
Intel (R)	BBPX2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPX2xx	Schematic: Top-Level Block Diagram
REV	B	



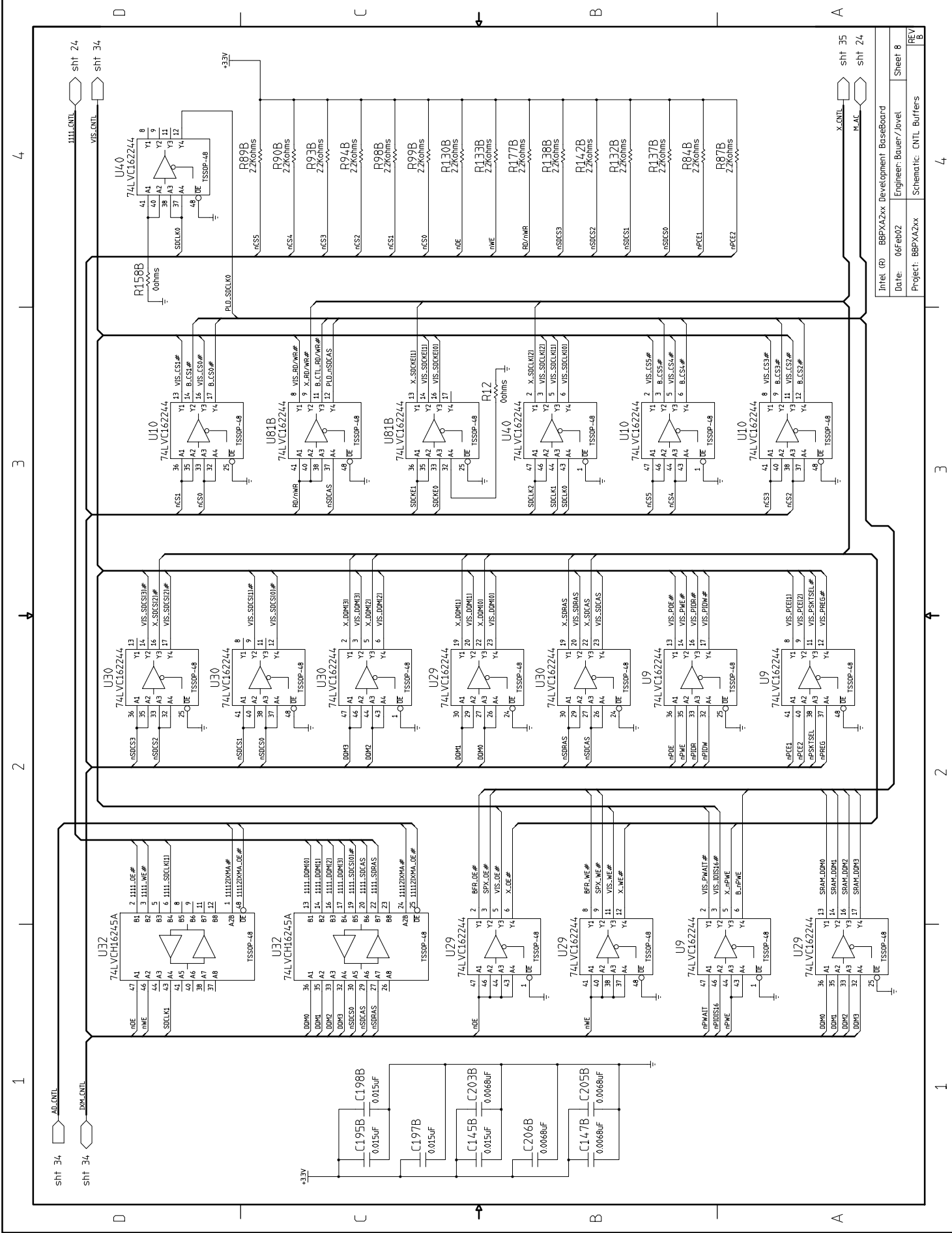
Intel (R) B8PXA2xx Development BaseBoard	
Date: 06Feb02	Engineer: Bauer/Jovel
Project: B8PXA2xx	Schematic: B2B Connector
	REV B



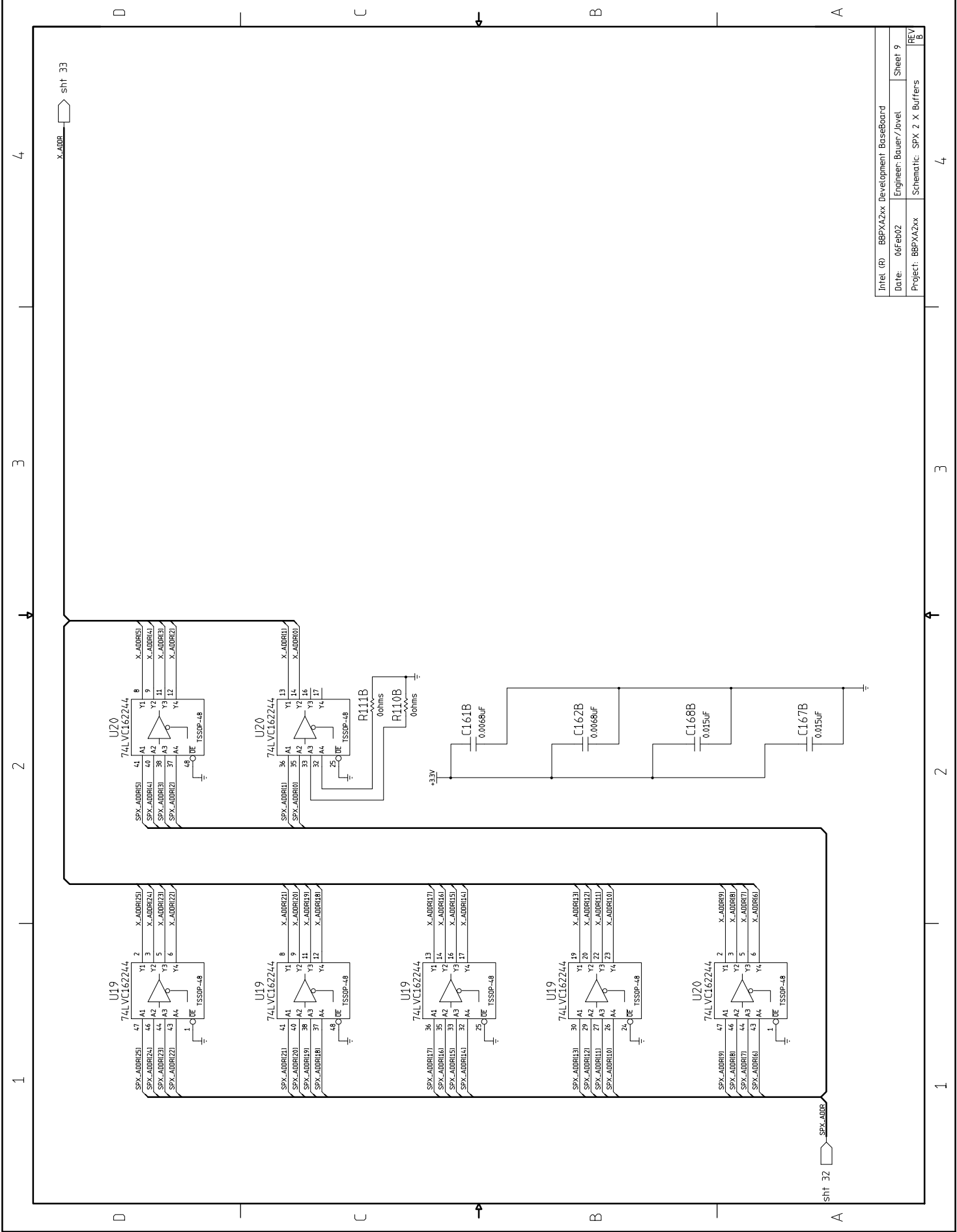
Intel (R)	BBPX2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPX2xx	Schematic: Data Transceivers and Buffers
		REV B



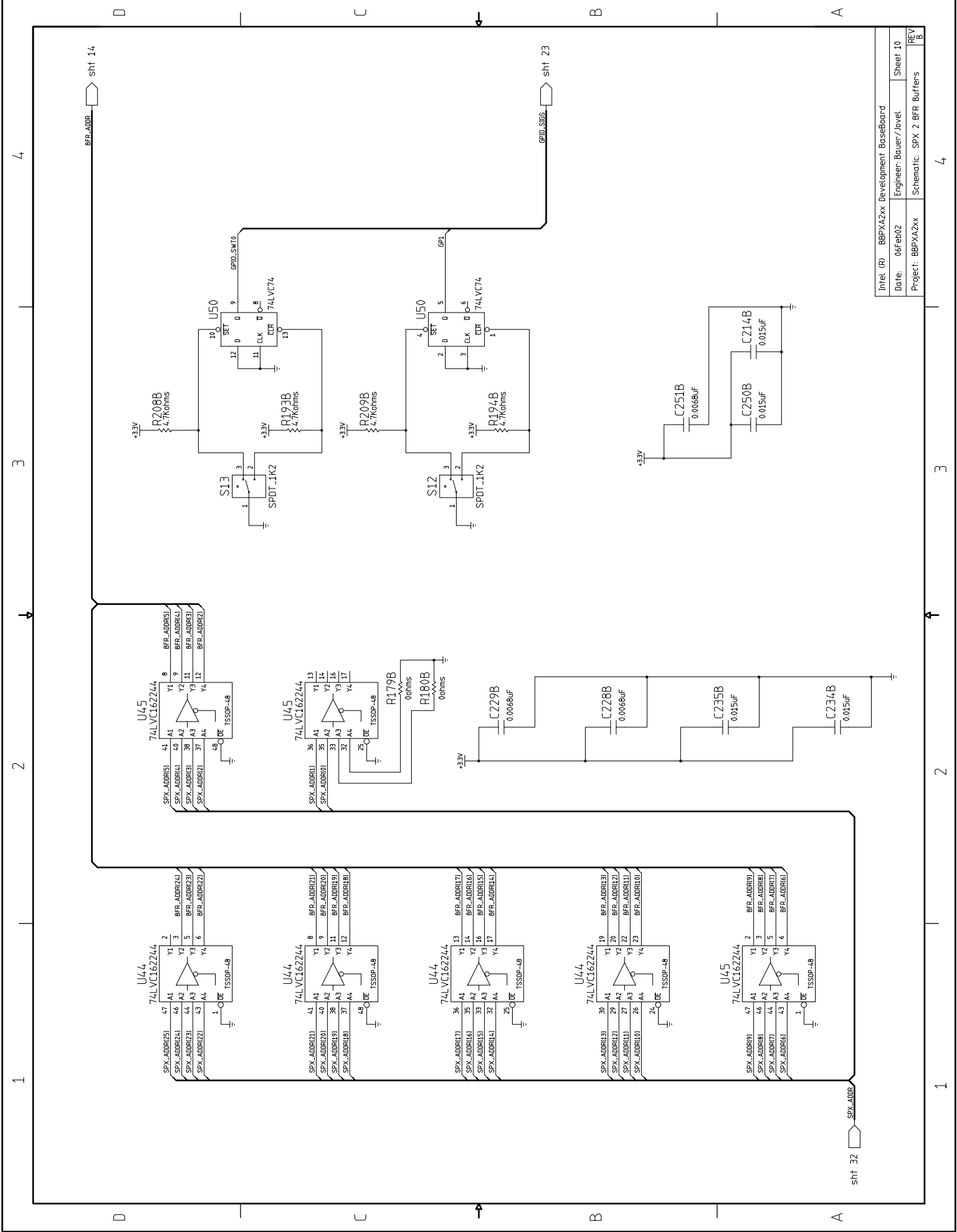
Intel (R) BBPXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 6
Project: BBPXA2xx	Schematic: Address Buffers	REV B



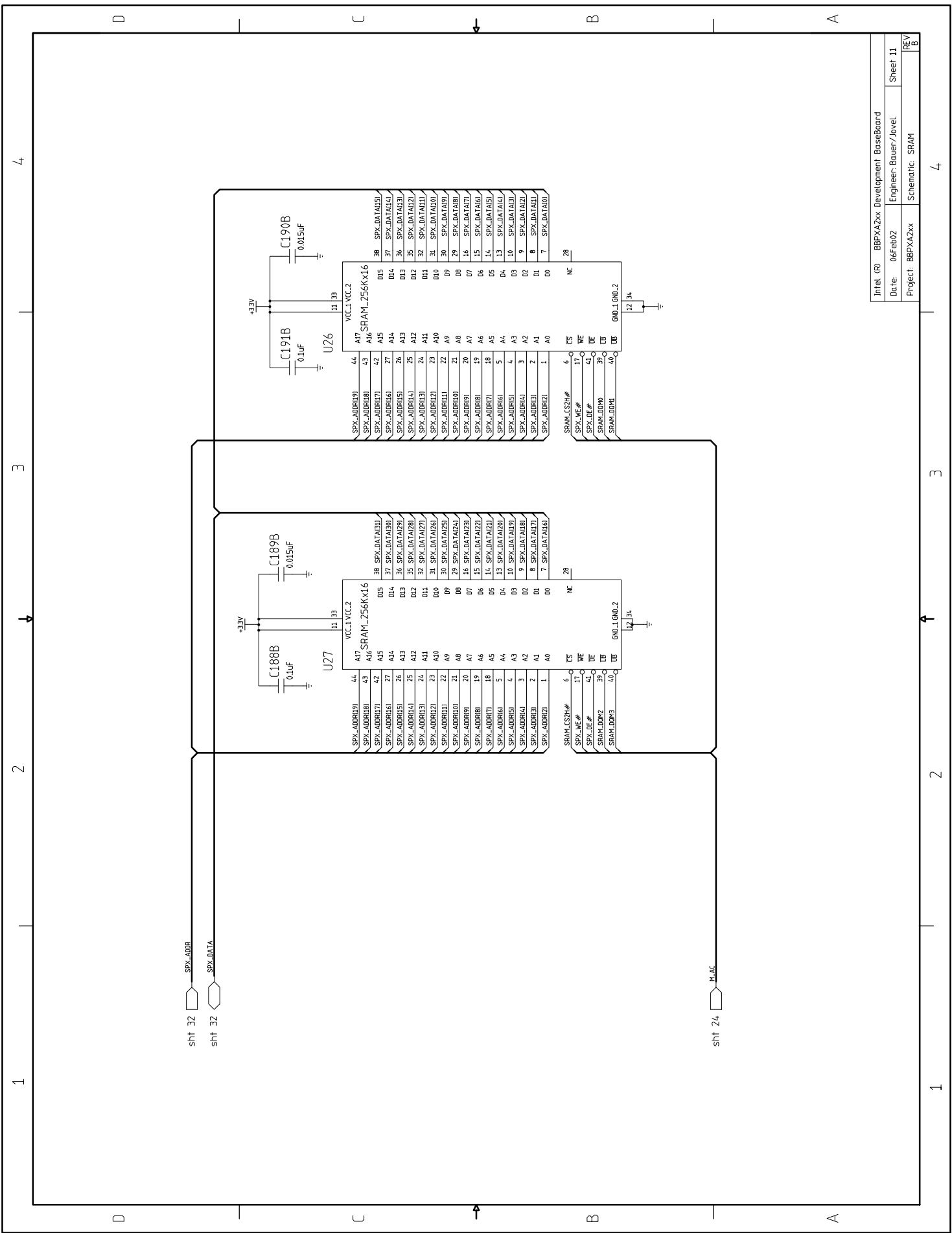
Intel (P)	BBPXA2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPXA2xx	Schematic: CNTL Buffers
REV	B	



Intel (R)	BBPX2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPX2xx	Schematic: SPX 2 X Buffers
REV	B	

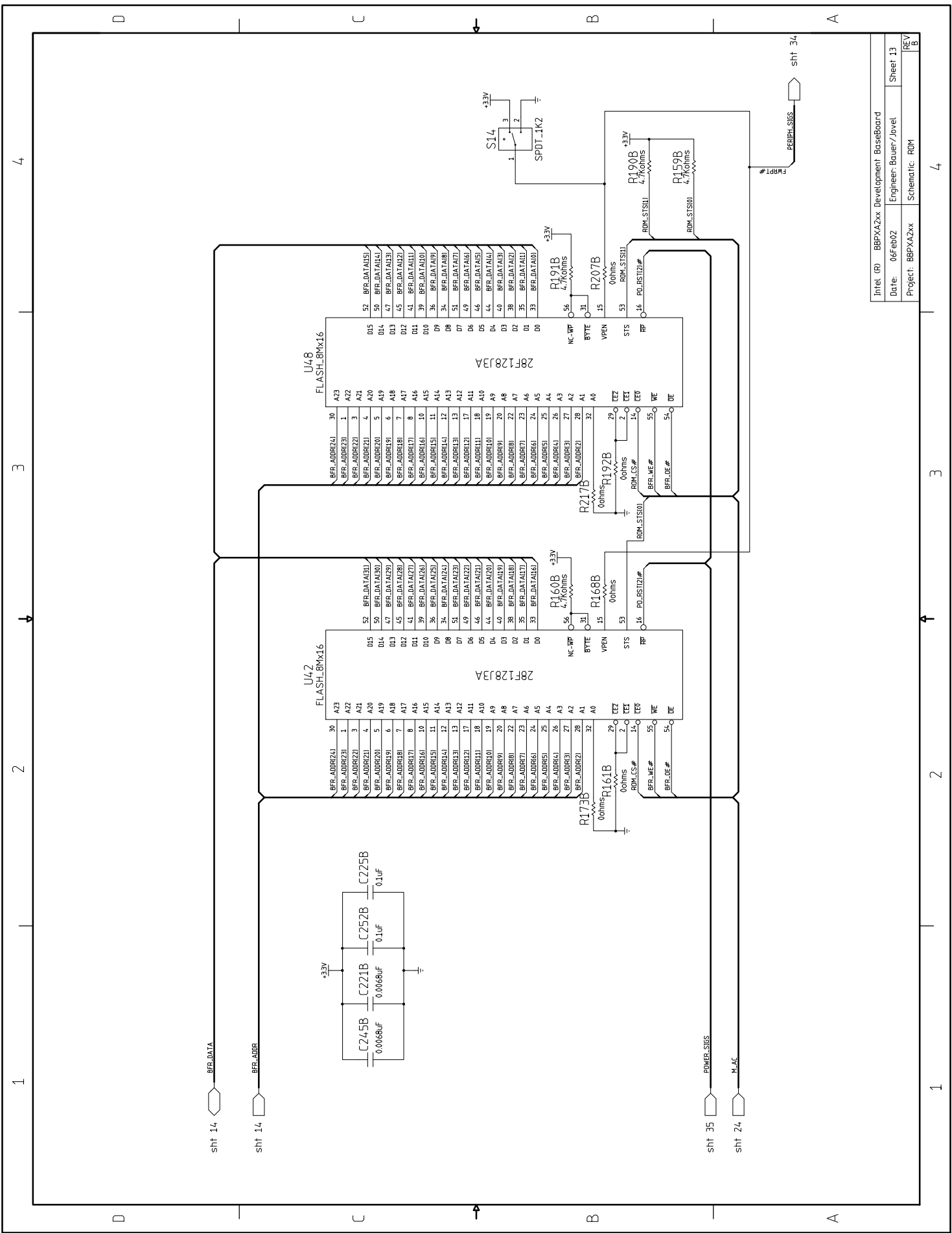


Intel (R)	BBPXA2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPXA2xx	Schematic: SPX 2 BFR Buffers
REV	B	Sheet 10

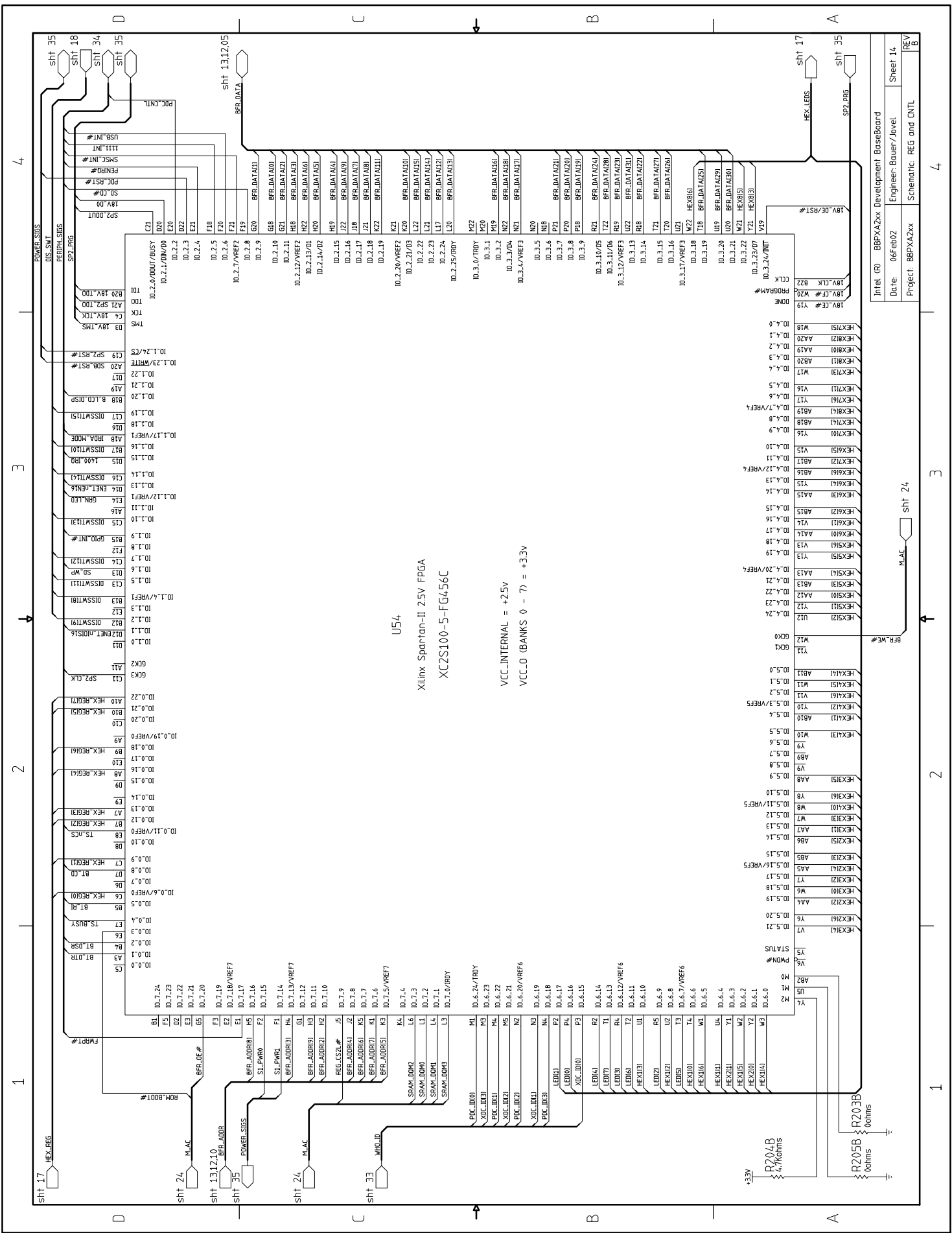


Intel (R) B8PXA2xx Development BaseBoard			
Date:	06Feb02	Engineer:	Bauer/Jovel
Project:	B8PXA2xx	Schematic:	SRAM
REV	B		

Sheet 11			
Intel (R) B8PXA2xx Development BaseBoard			
Date:	06Feb02	Engineer:	Bauer/Jovel
Project:	B8PXA2xx	Schematic:	SRAM
REV	B		



Intel (R) BBPXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 13
Project: BBPXA2xx	Schematic: ROM	REV B



U54

Xilinx Spartan-II 2.5V FPGA

XC2S100-5-FG456C

VCC_INTERNAL = +2.5V

VCC_0 (BANKS 0 - 7) = +3.3V

Intel (R) B8PXA2xx Development BaseBoard

Date: 06Feb02 Engineer: Bauer/Jovel

Project: B8PXA2xx Schematic: REG and CNTL

REV B

sh1 24

MAC

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

sh1 24

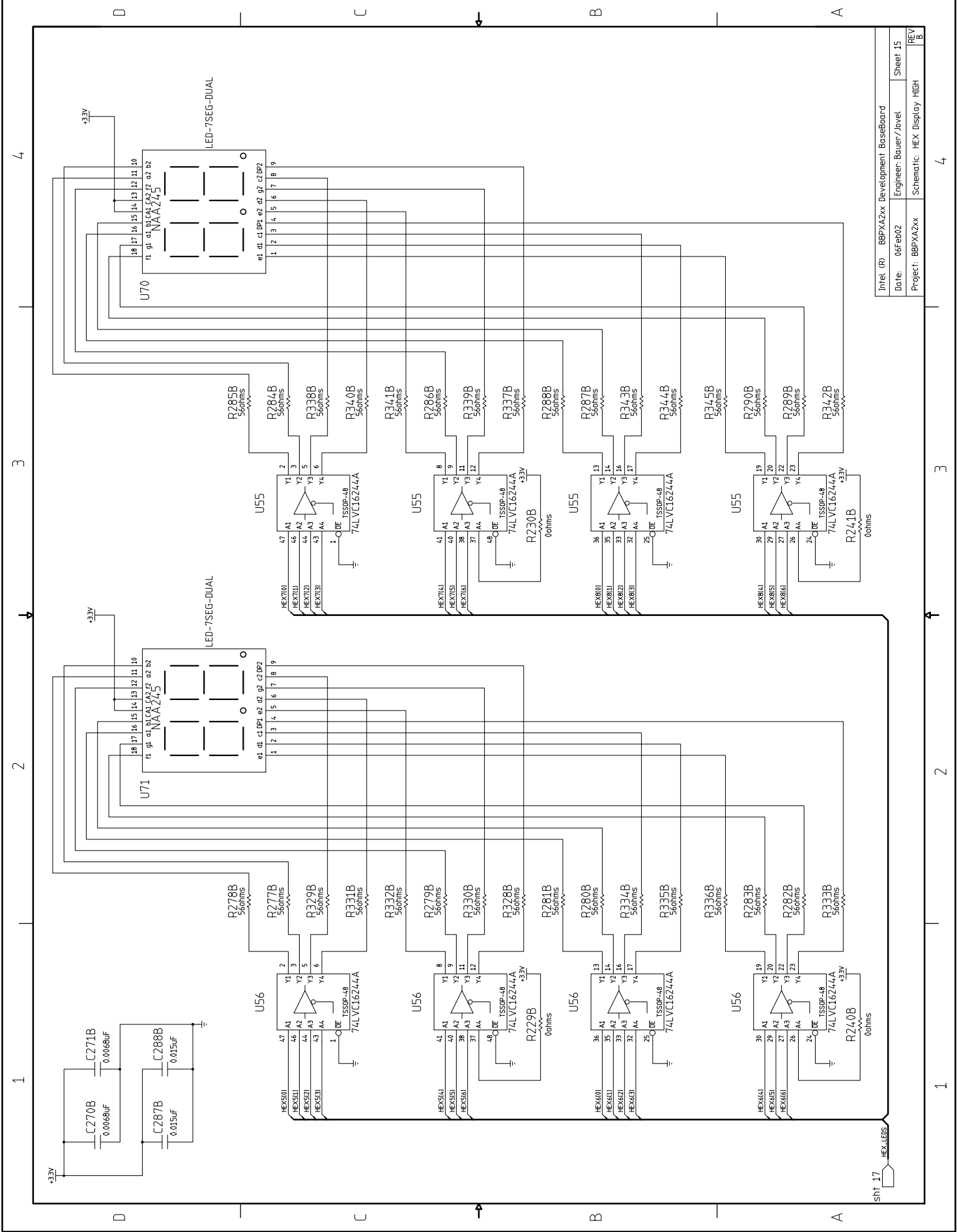
sh1 24

sh1 24

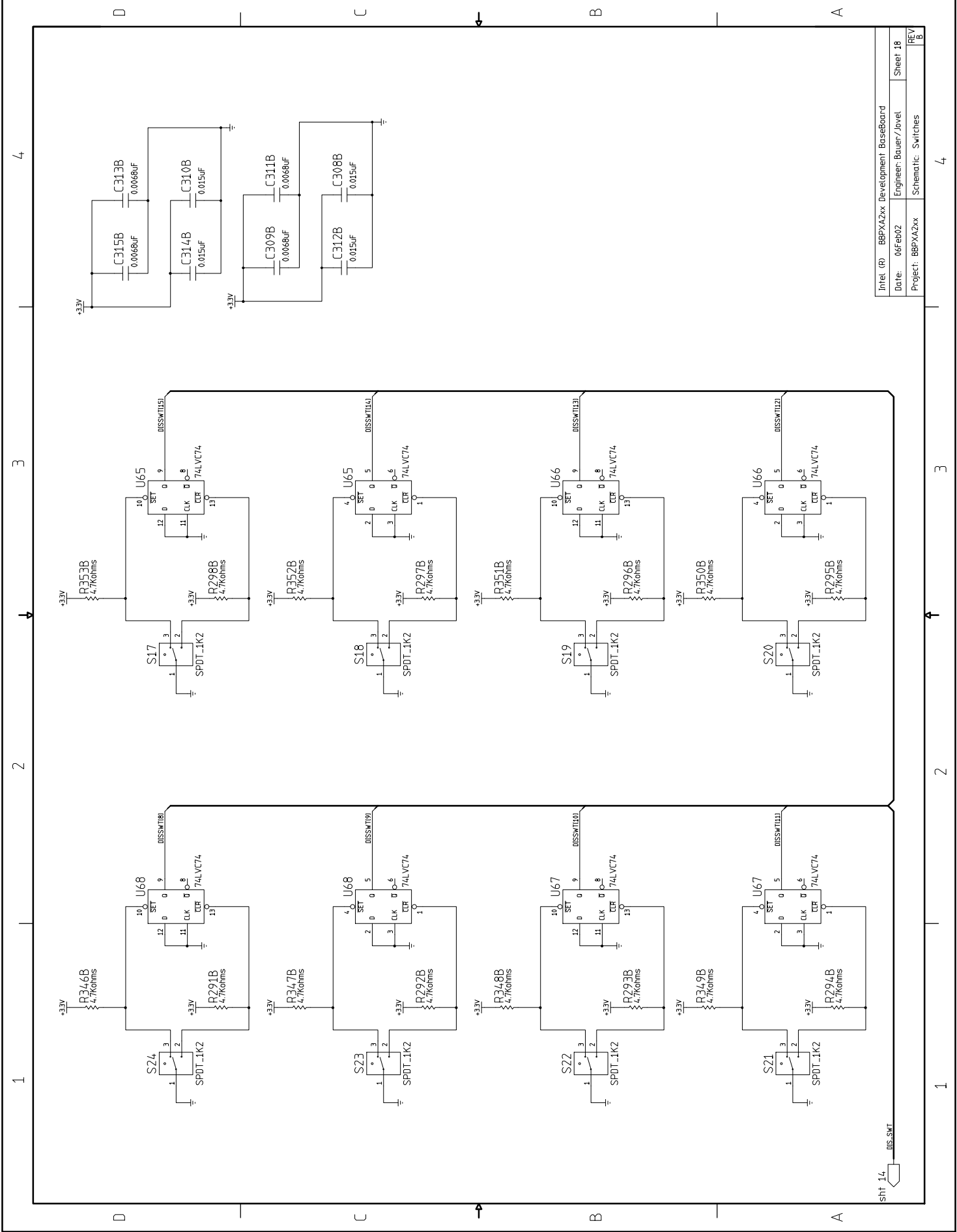
sh1 24

sh1 24

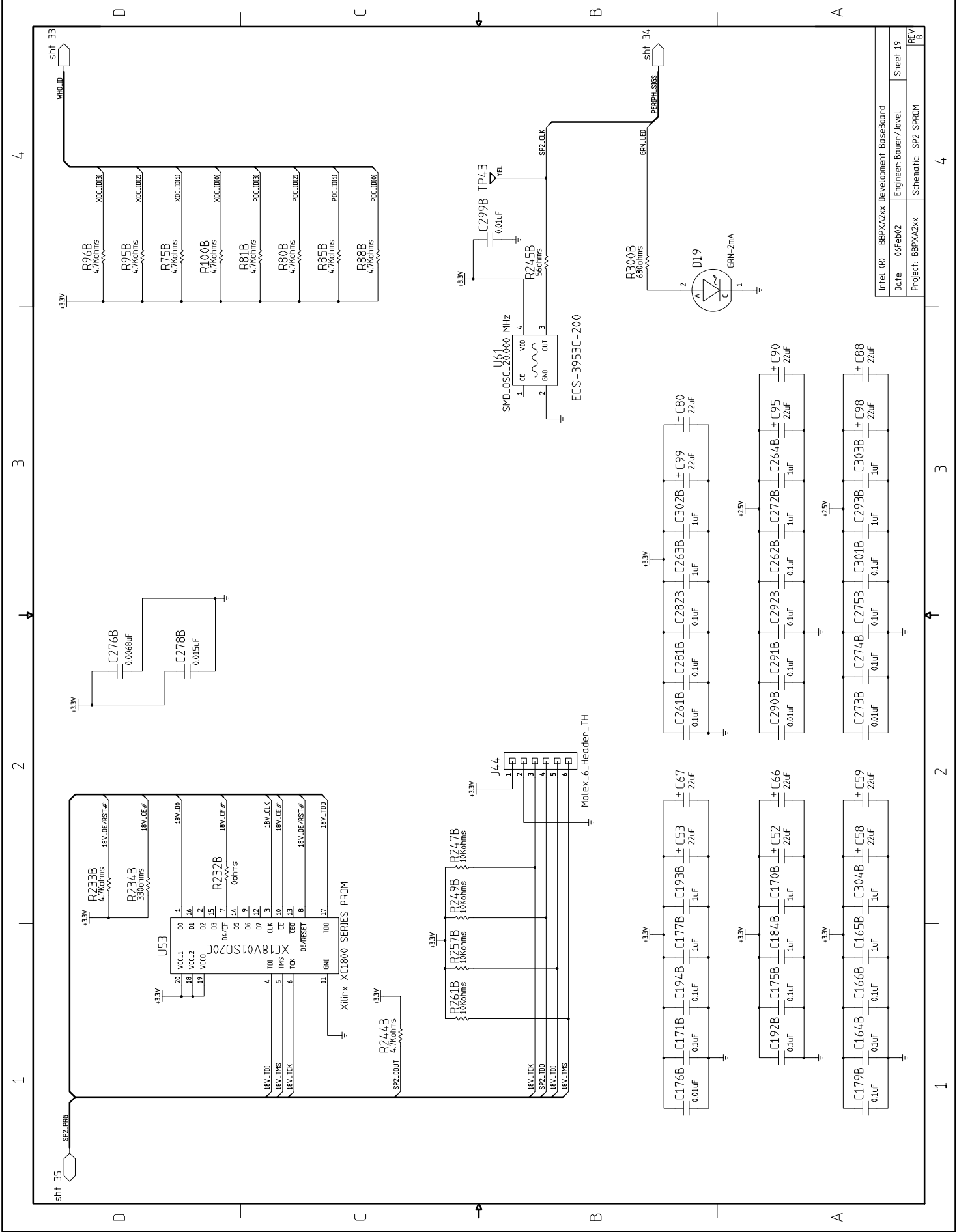
sh1 24



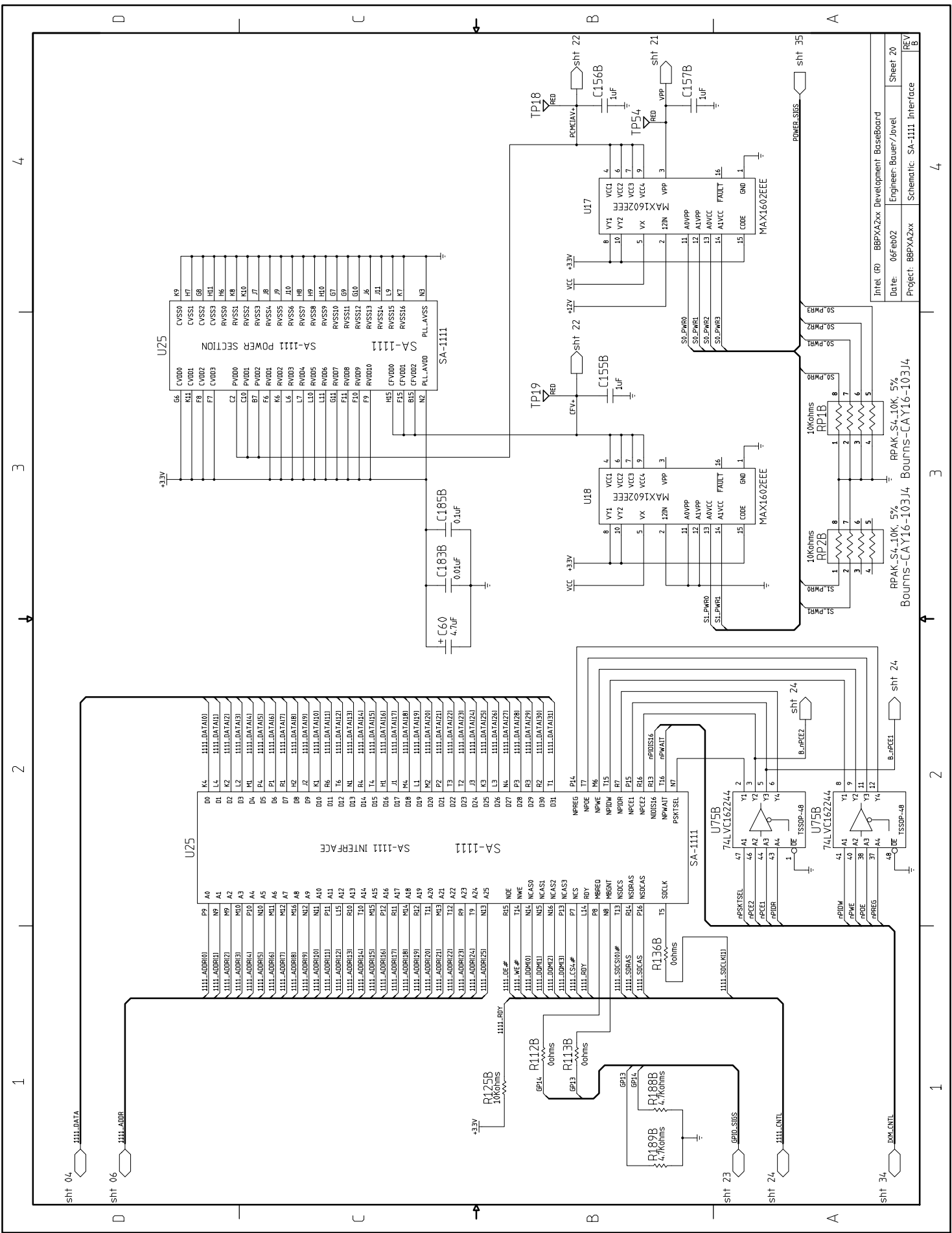
Intel (R)	BBPXA2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPXA2xx	Schematic: HEX Display High
REV	B	



Intel (P)	BBPXA2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPXA2xx	Schematic: Switches
REV	B	



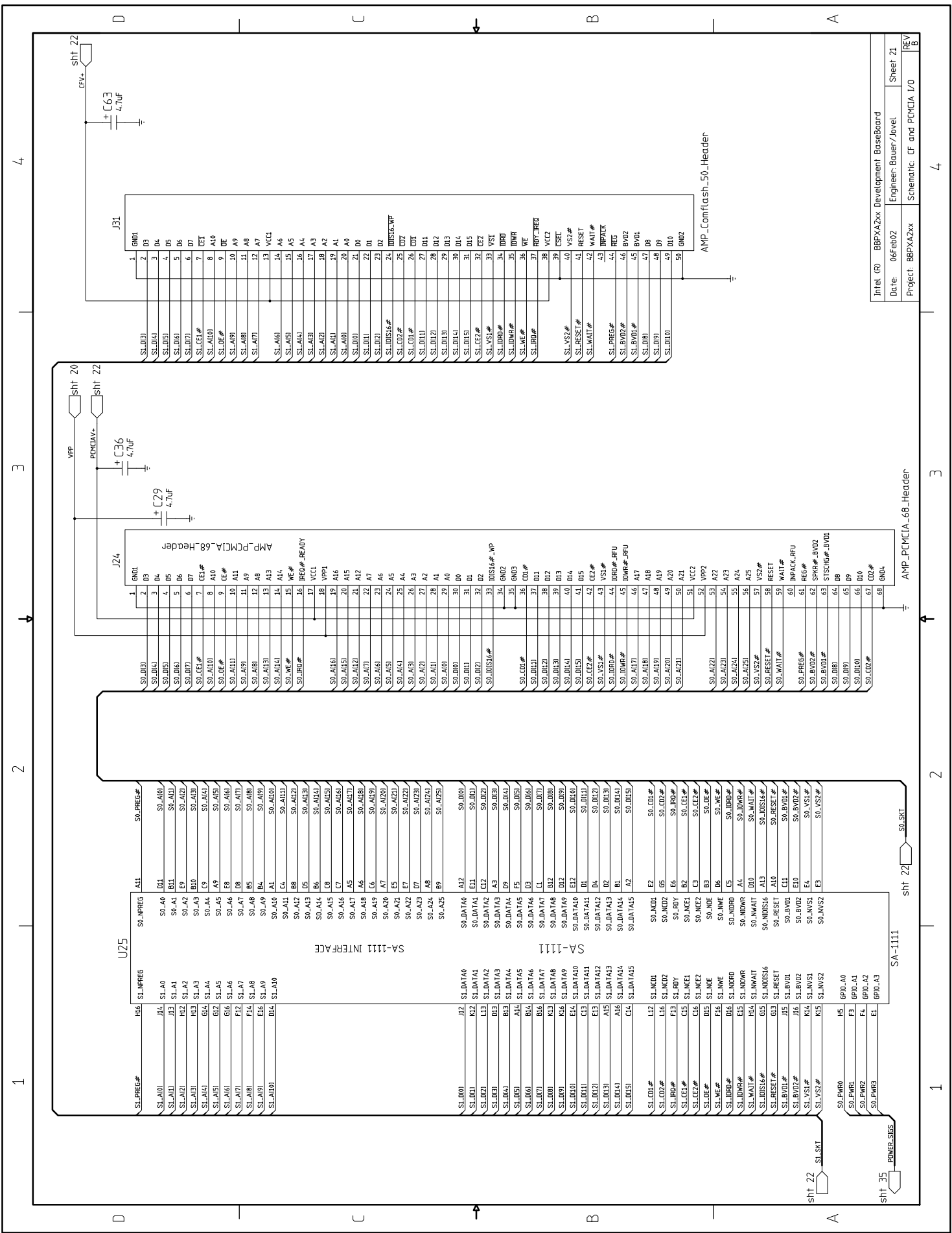
Intel (P)	BBPXA2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPXA2xx	Schematic: SP2_SPROQM
REV	B	Sheet 19



Intel (P) B8PXA2xx Development BaseBoard
Date: 06Feb02
Project: B8PXA2xx
Schematic: SA-1111 Interface
Sheet 20
REV B

RPAK-S4.10K, 5%
Bourns-CAY16-103J4

RPAK-S4.10K, 5%
Bourns-CAY16-103J4



Intel (R) B8PXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 21
Project: B8PXA2xx	Schematic: CF and PCMCIA I/O	REV B

CD2#			
GND4			

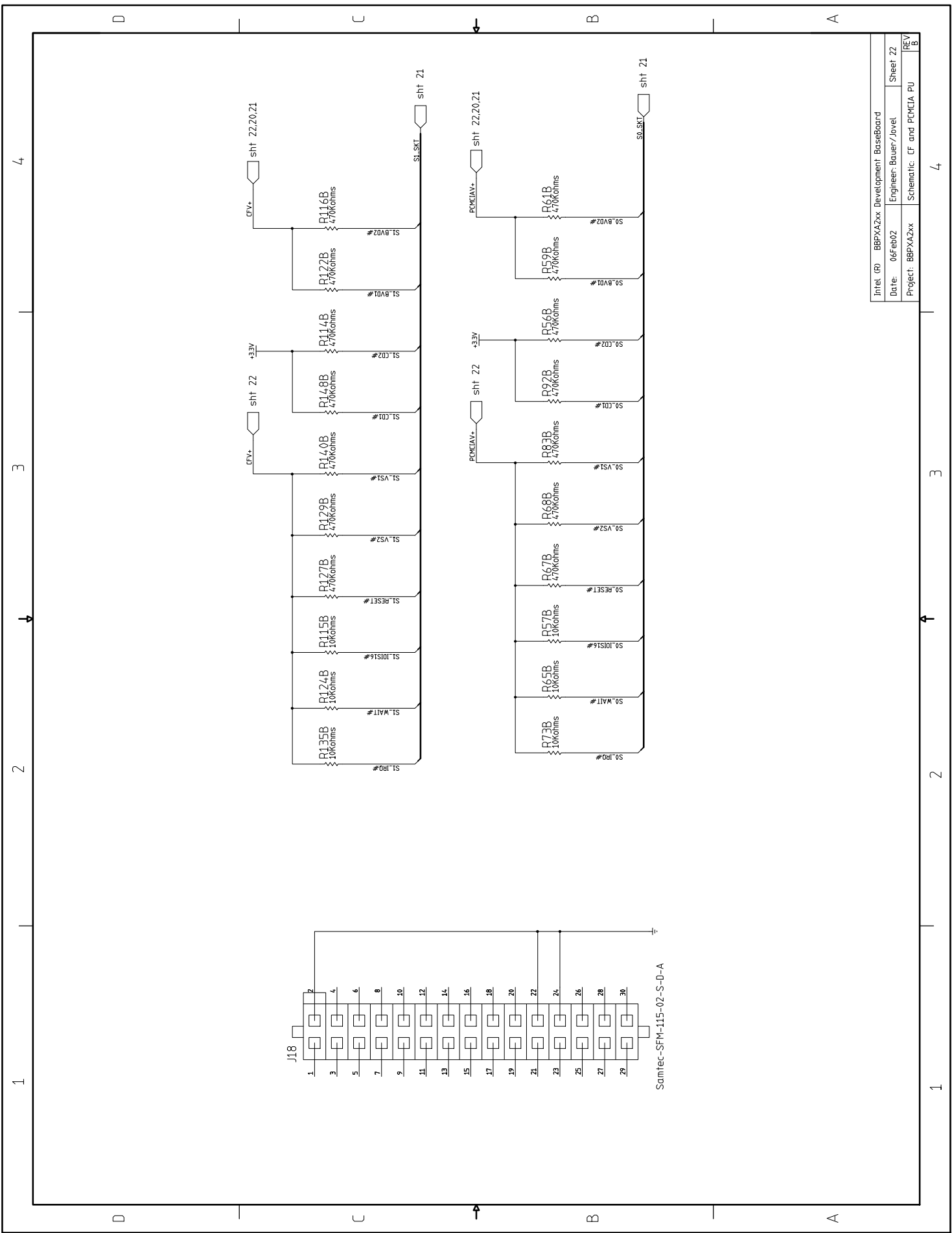
AMP_PCMCIA_68_Header

AMP_PCMCIA_50_Header			
1	2	3	4

SA-1111
GPIO_A3
1

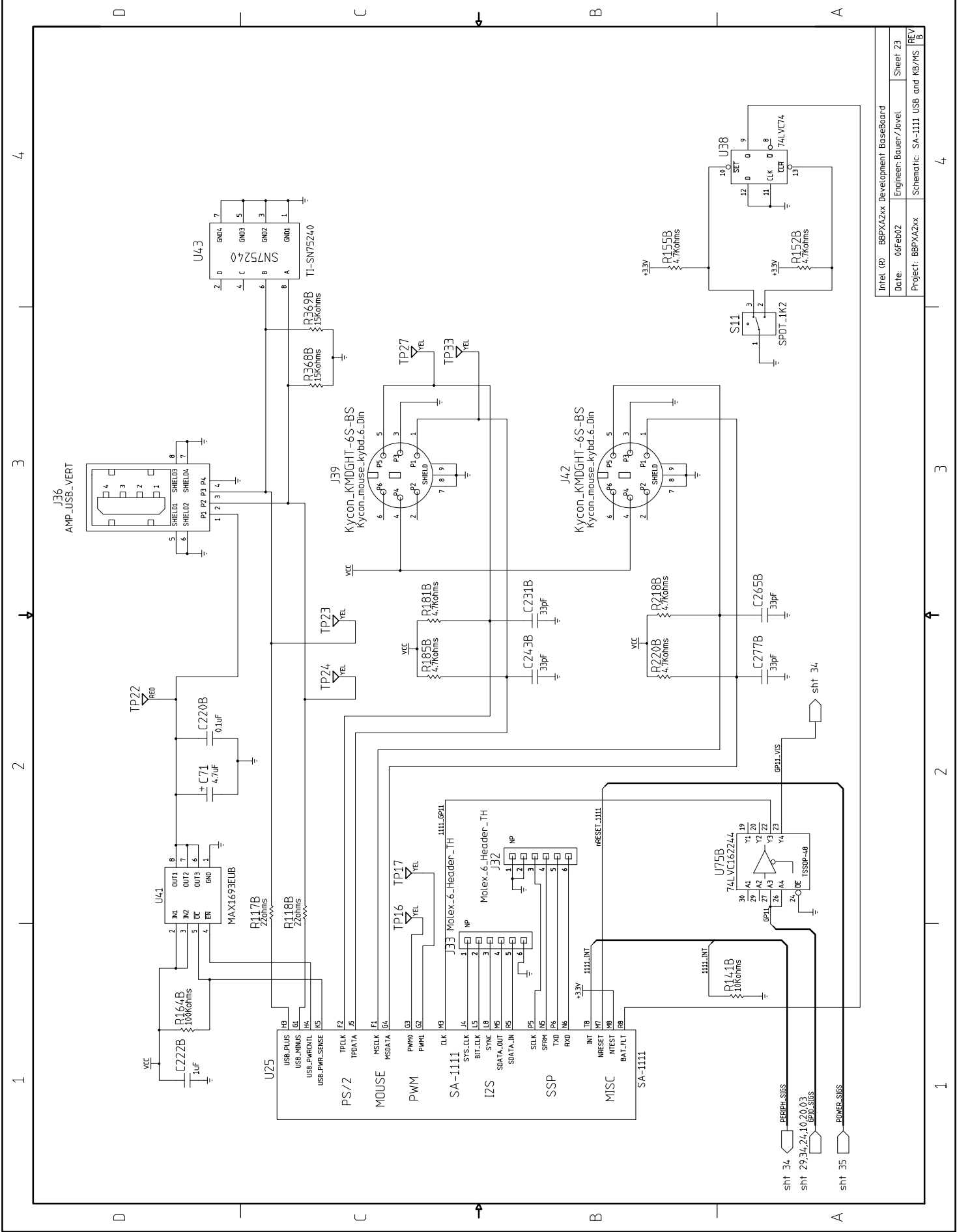
EWMS-05

SA-1111			
1	2	3	4

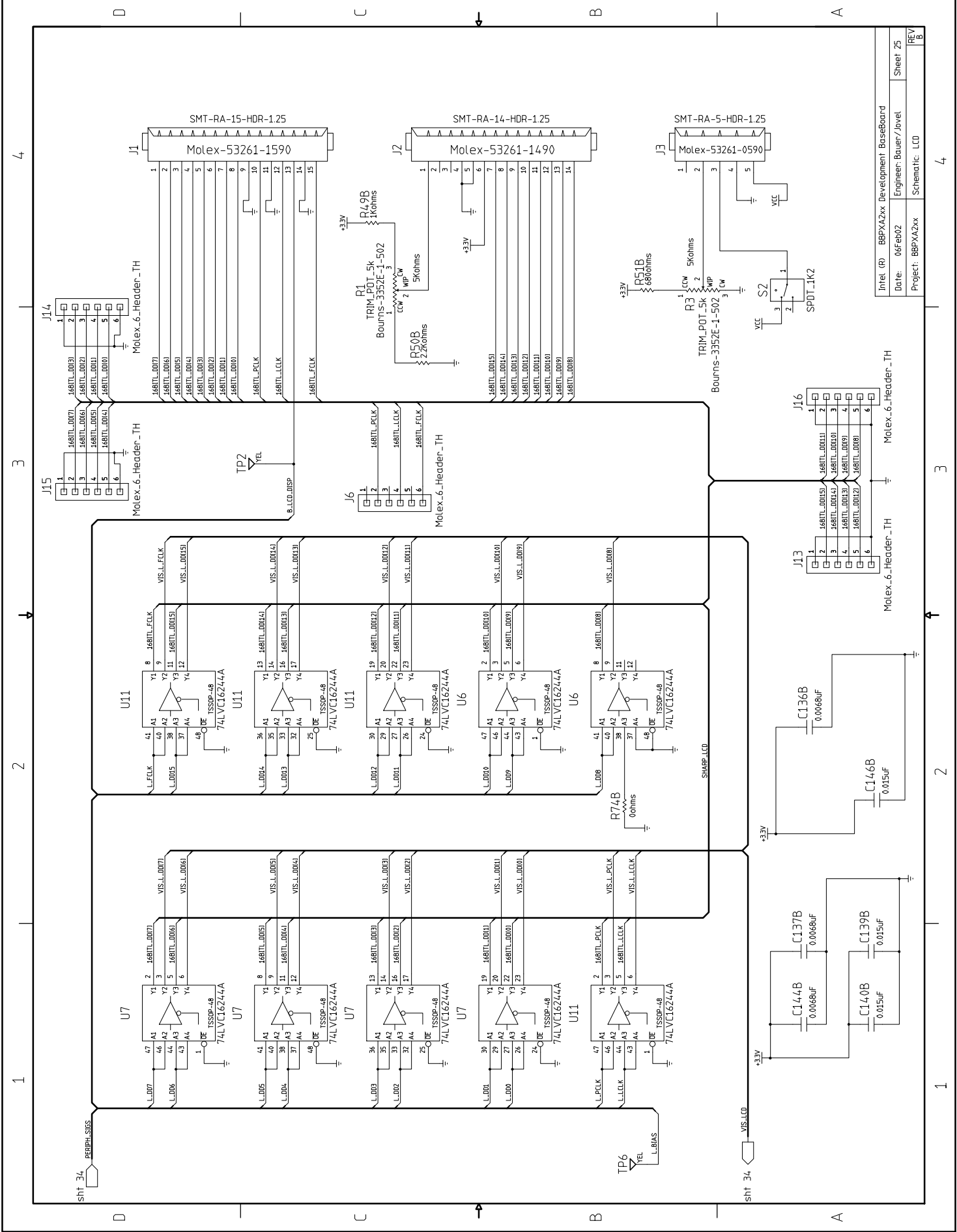


Intel (R) B8PXA2xx Development BaseBoard			
Date:	06Feb02	Engineer:	Bauer/Jovel
Project:	B8PXA2xx	Schematic:	CF and PCMCIA PU
		REV	B

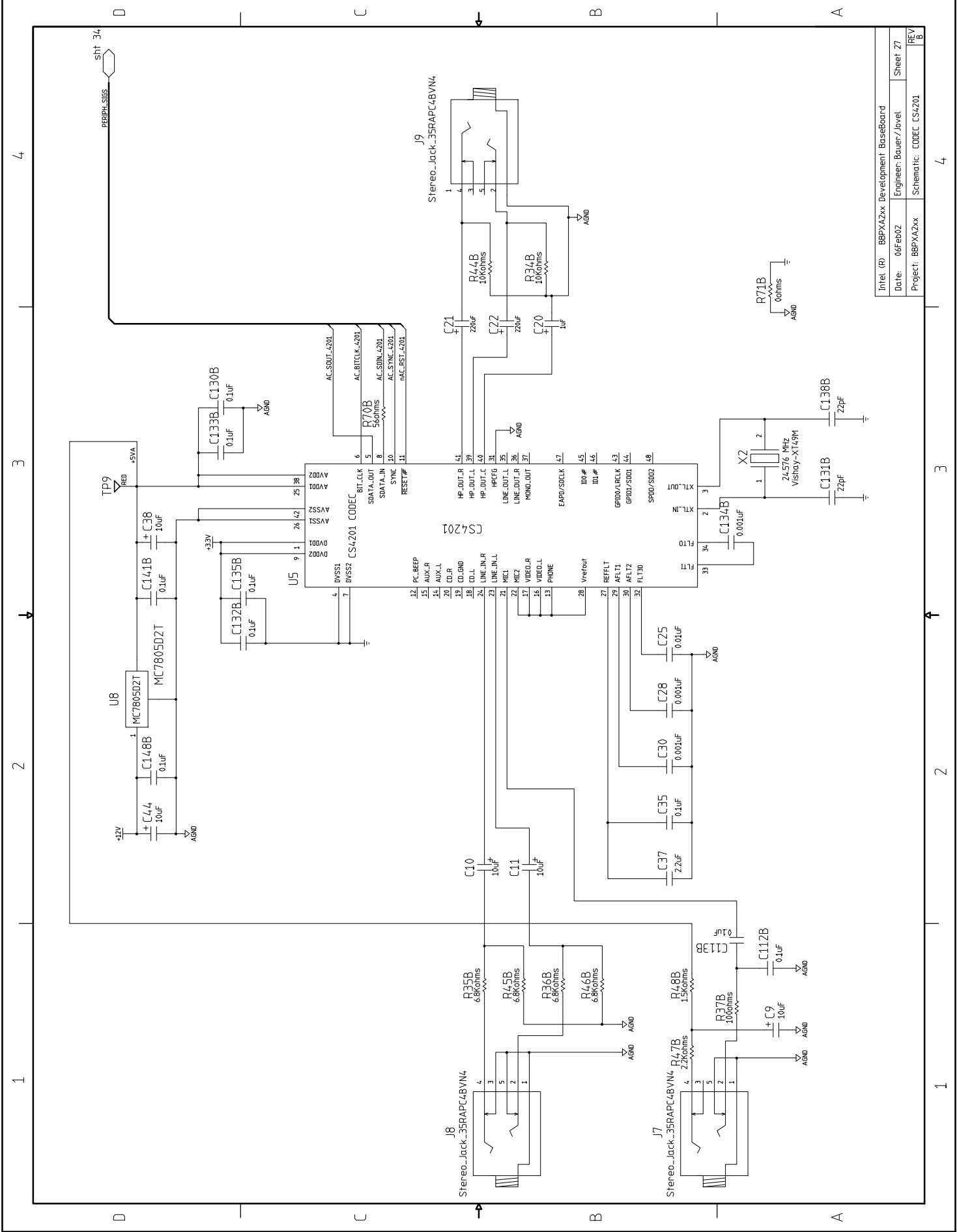
Samtec-SFM-115-02-S-D-A



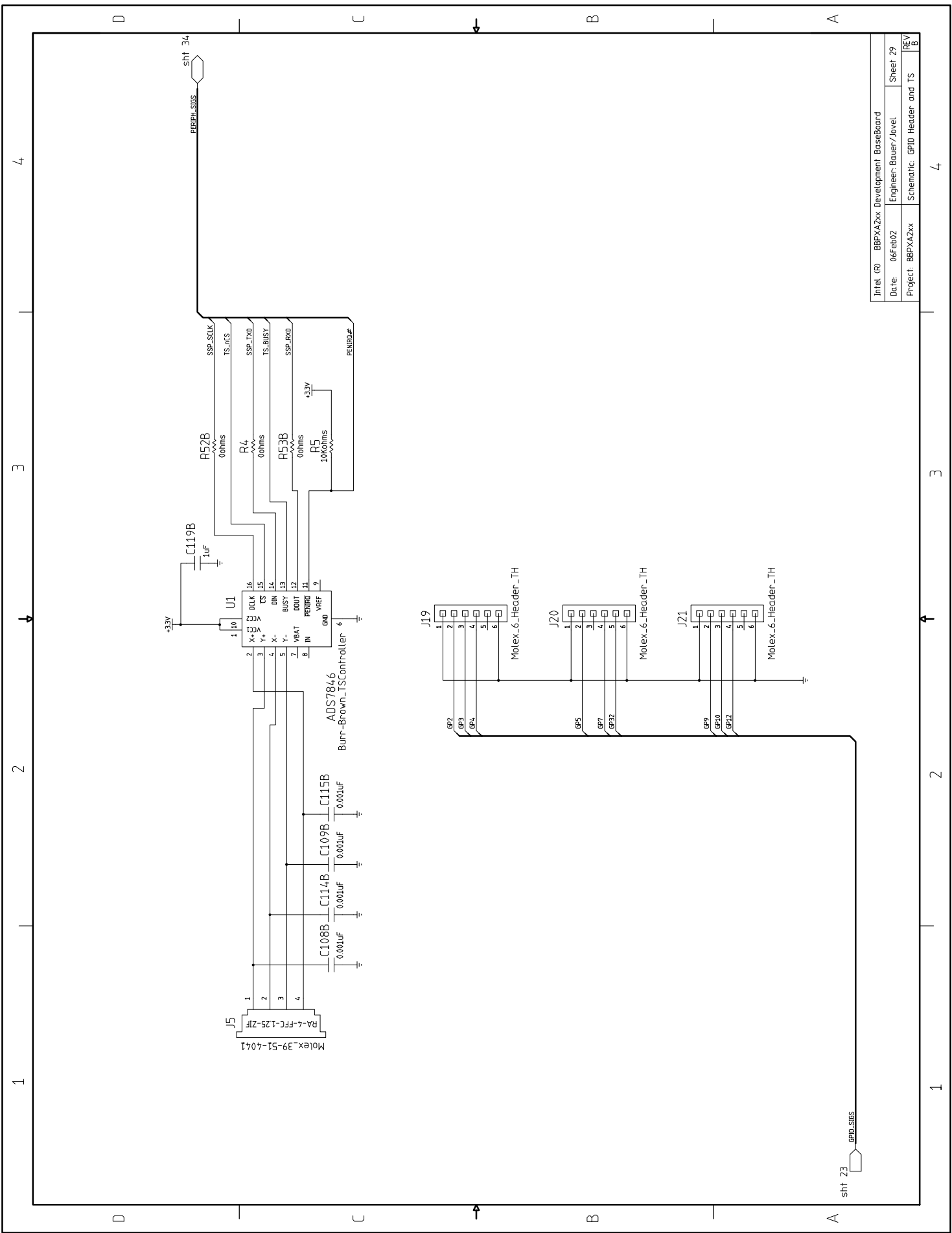
Intel (R) B8PXA2xx Development BaseBoard	Sheet 23
Date: 06Feb02	Engineer: Bauer/Jovel
Project: B8PXA2xx	Schematic: SA-1111 USB and KB/MS
REV B	



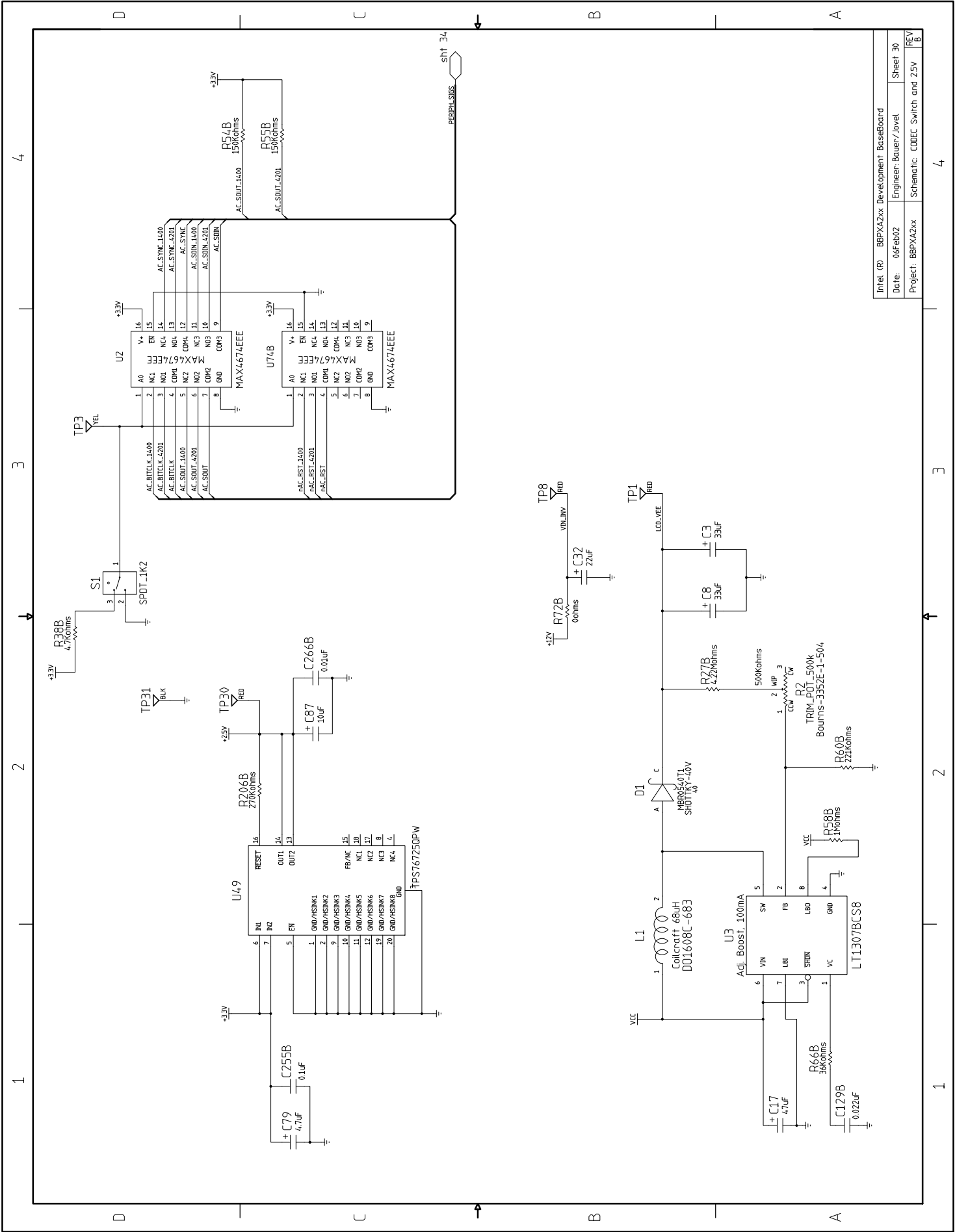
Intel (P) B8PXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 25
Project: B8PXA2xx	Schematic: LCD	REV B



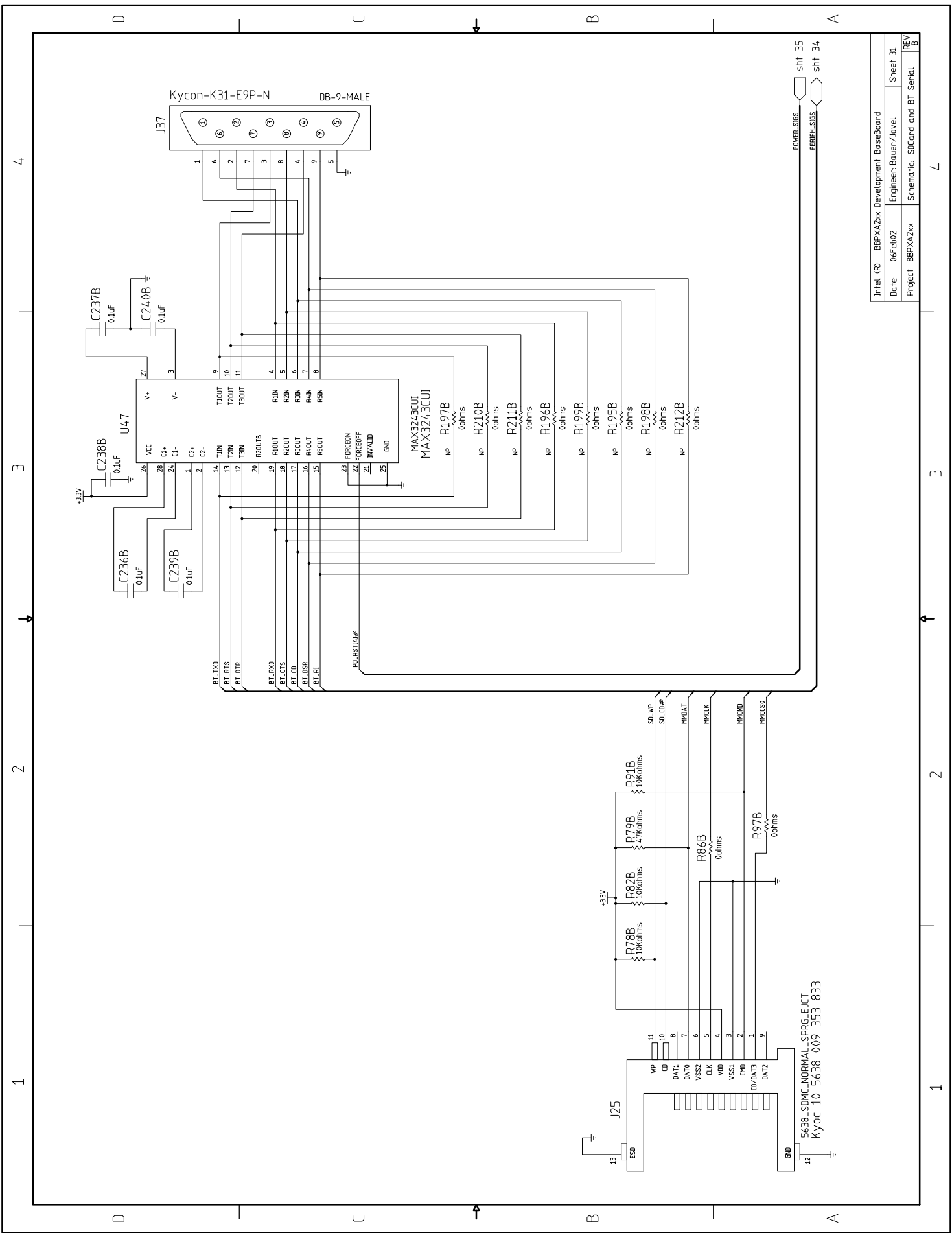
Intel (R) BBPXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 27
Project: BBPXA2xx	Schematic: CODEC_CS4201	REV B



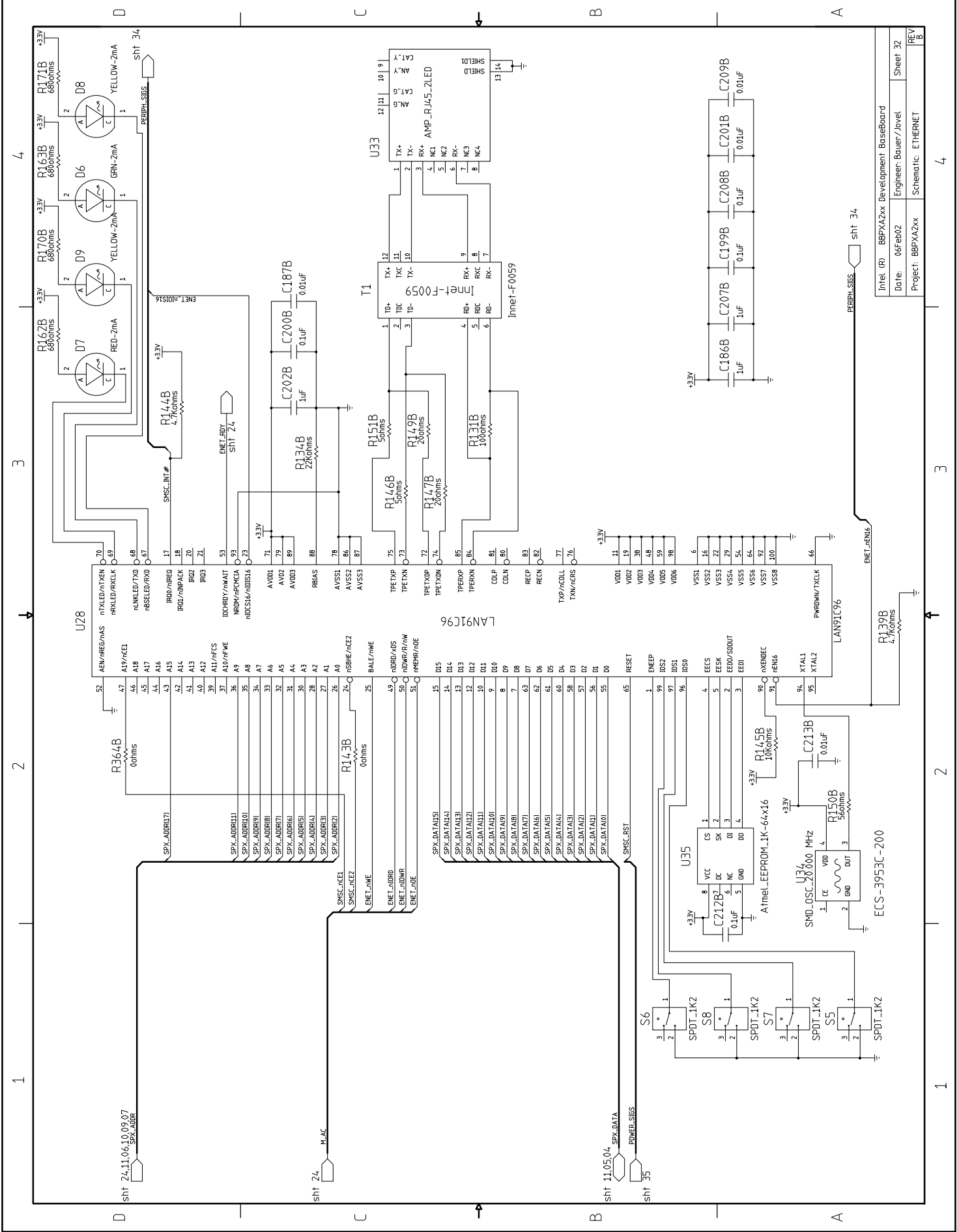
Intel (R) BBPX2xx Development BaseBoard			
Date:	06Feb02	Engineer:	Bauer/Jovel
Project:	BBPX2xx	Schematic:	GP10 Header and TS
		REV	B



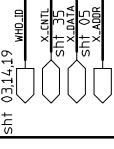
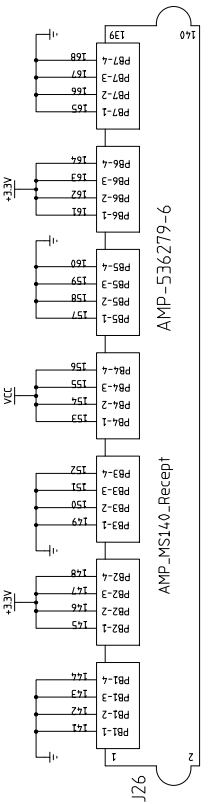
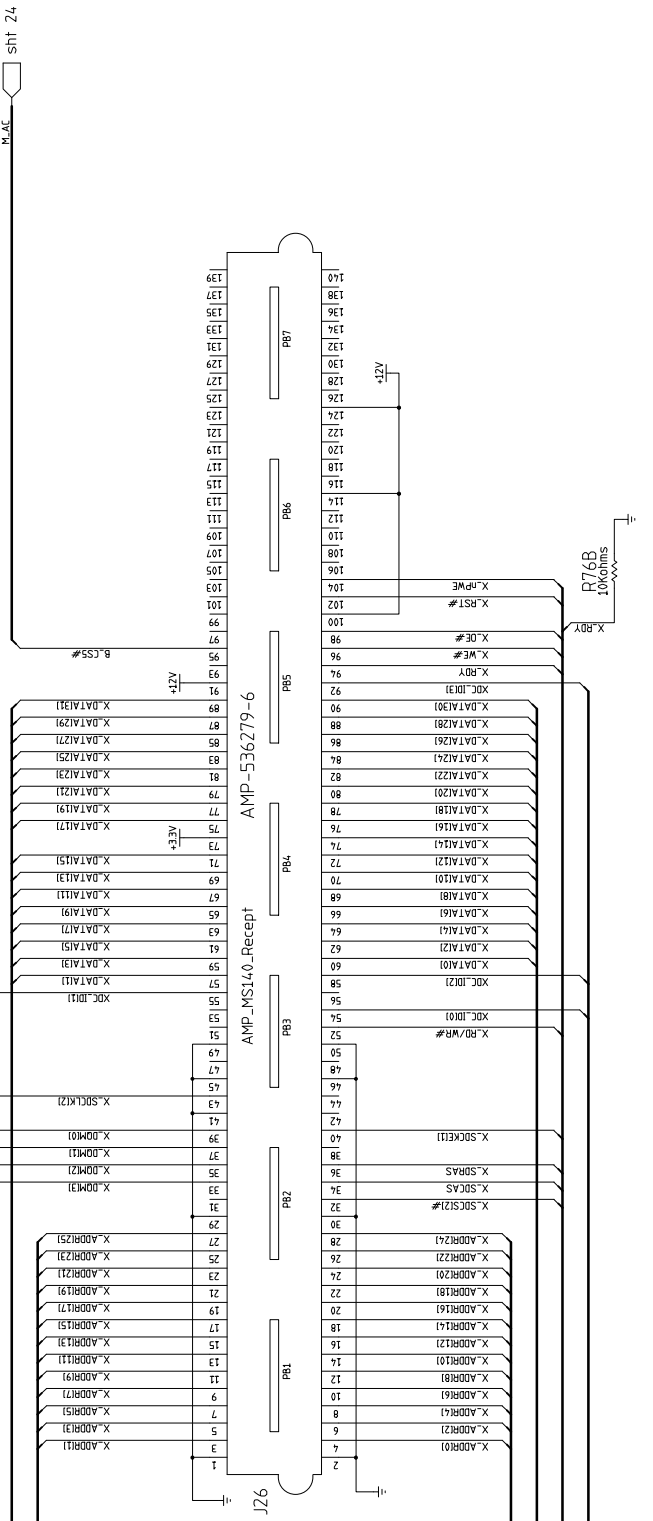
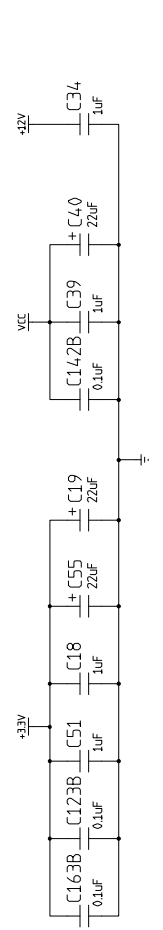
Intel (R) BBPXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 30
Project: BBPXA2xx	Schematic: CODEC Switch and 2.5V	REV B

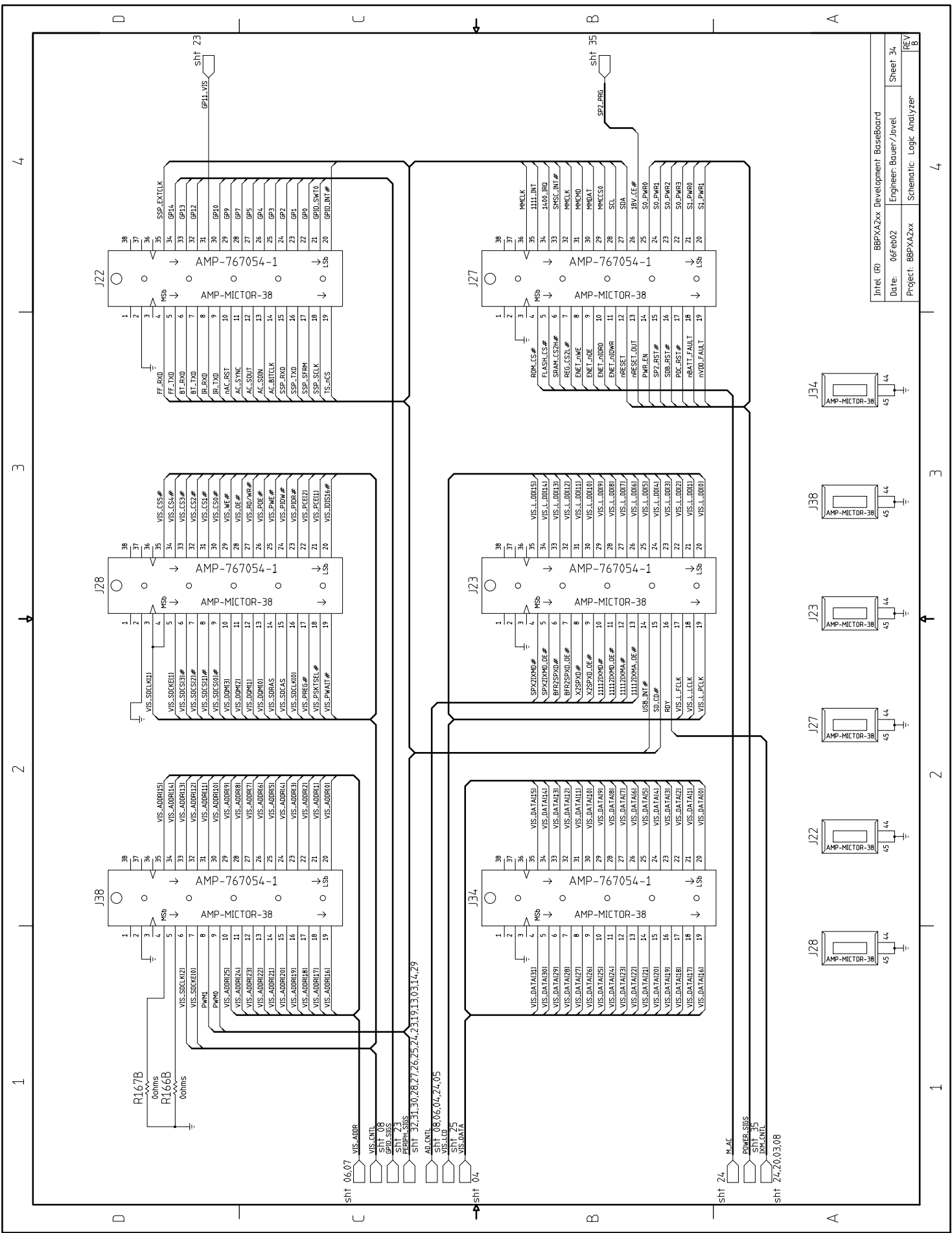


Intel (R) B8PXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 31
Project: B8PXA2xx	Schematic: SDCard and BT Serial	REV B

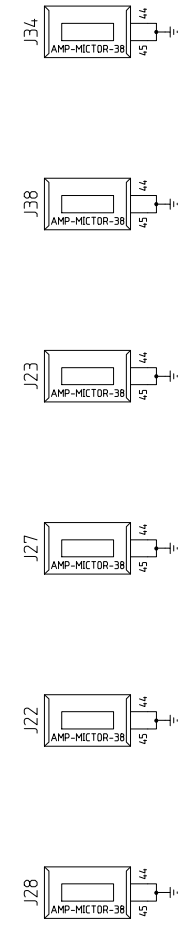


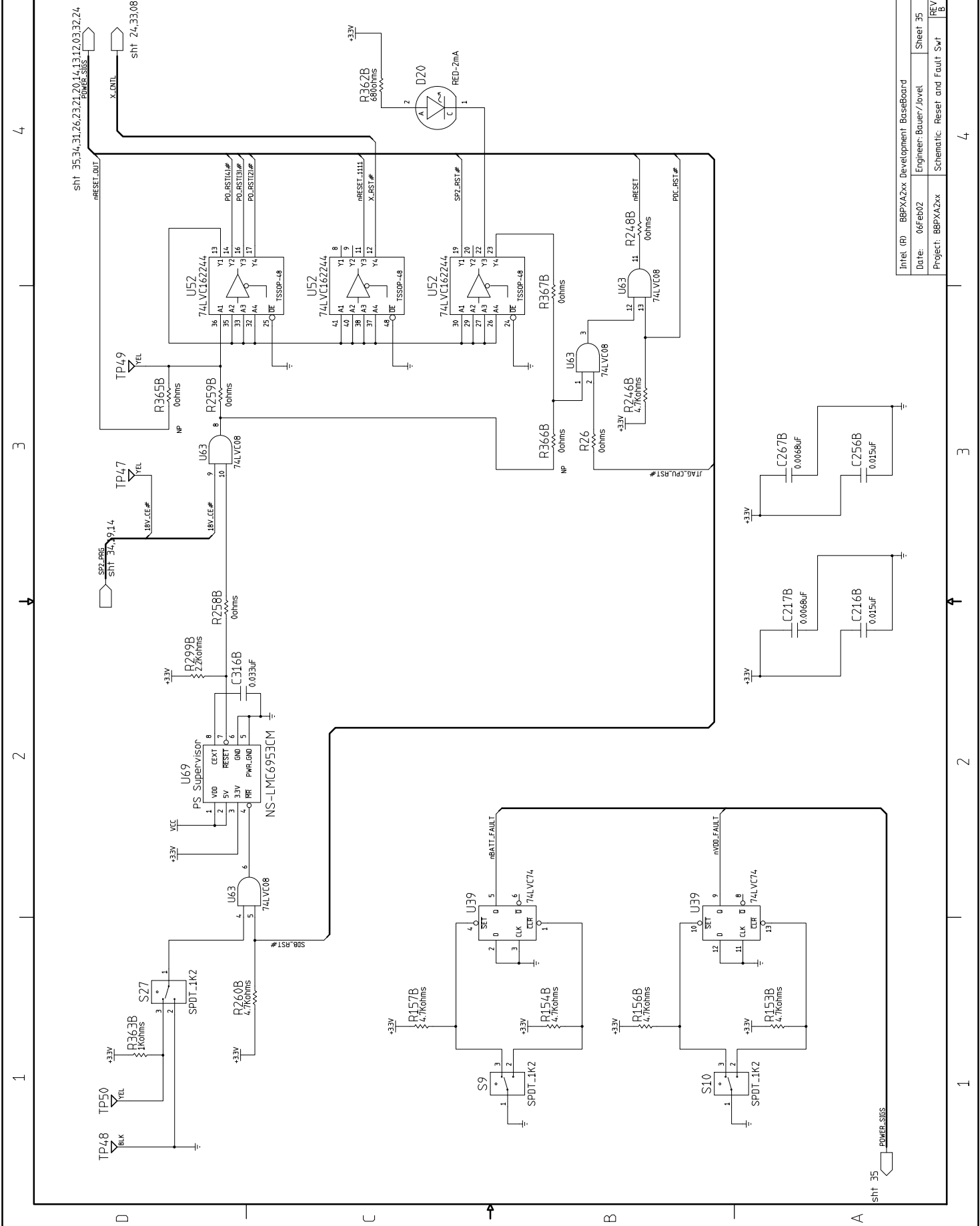
Intel (R) BBPXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 32
Project: BBPXA2xx	Schematic: ETHERNET	REV B



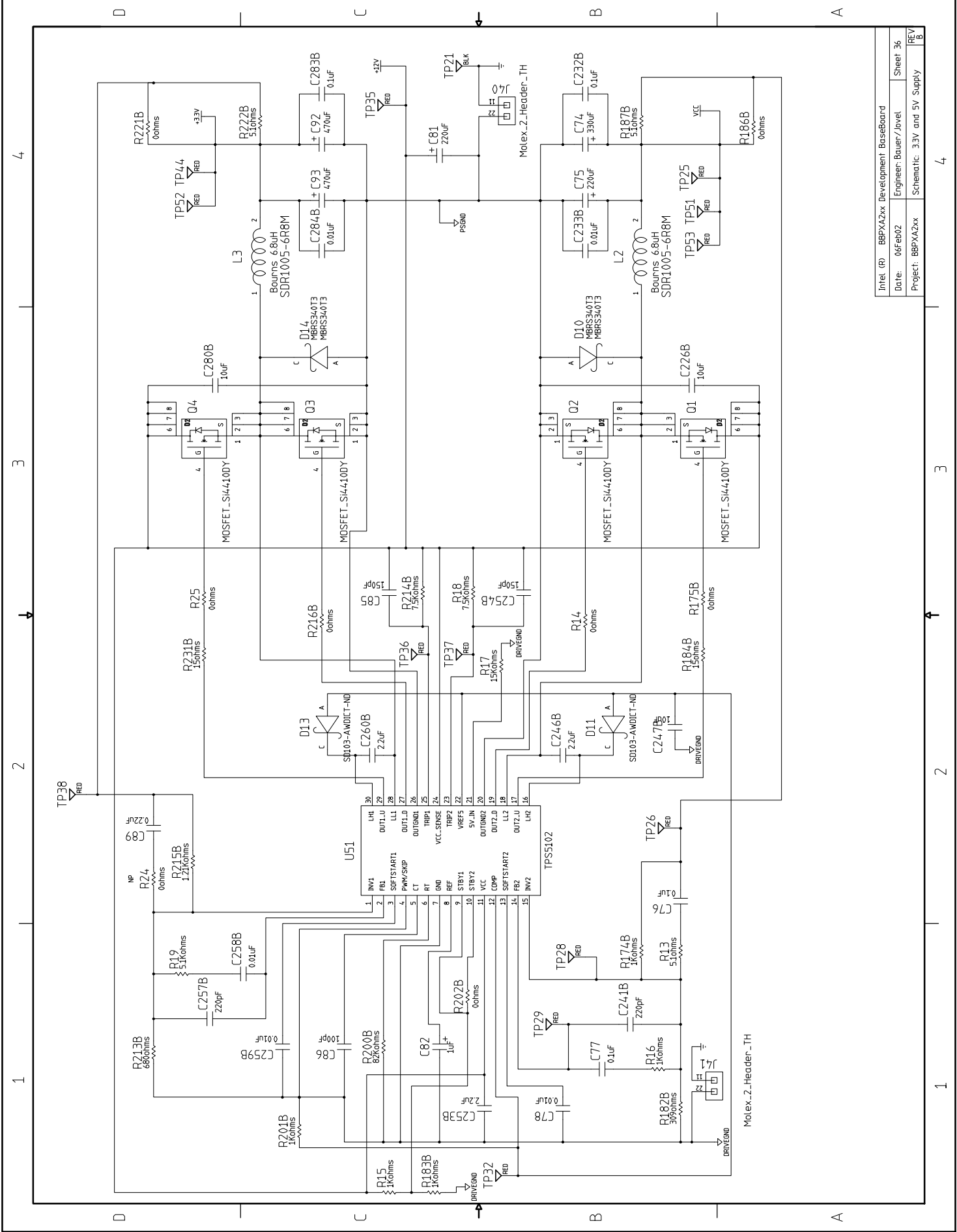


Intel (R)	BBPXA2xx Development BaseBoard
Date:	06Feb02
Project:	BBPXA2xx
Engineer:	Bauer/Jovel
Schematic:	Logic Analyzer
Sheet:	34
REV	B

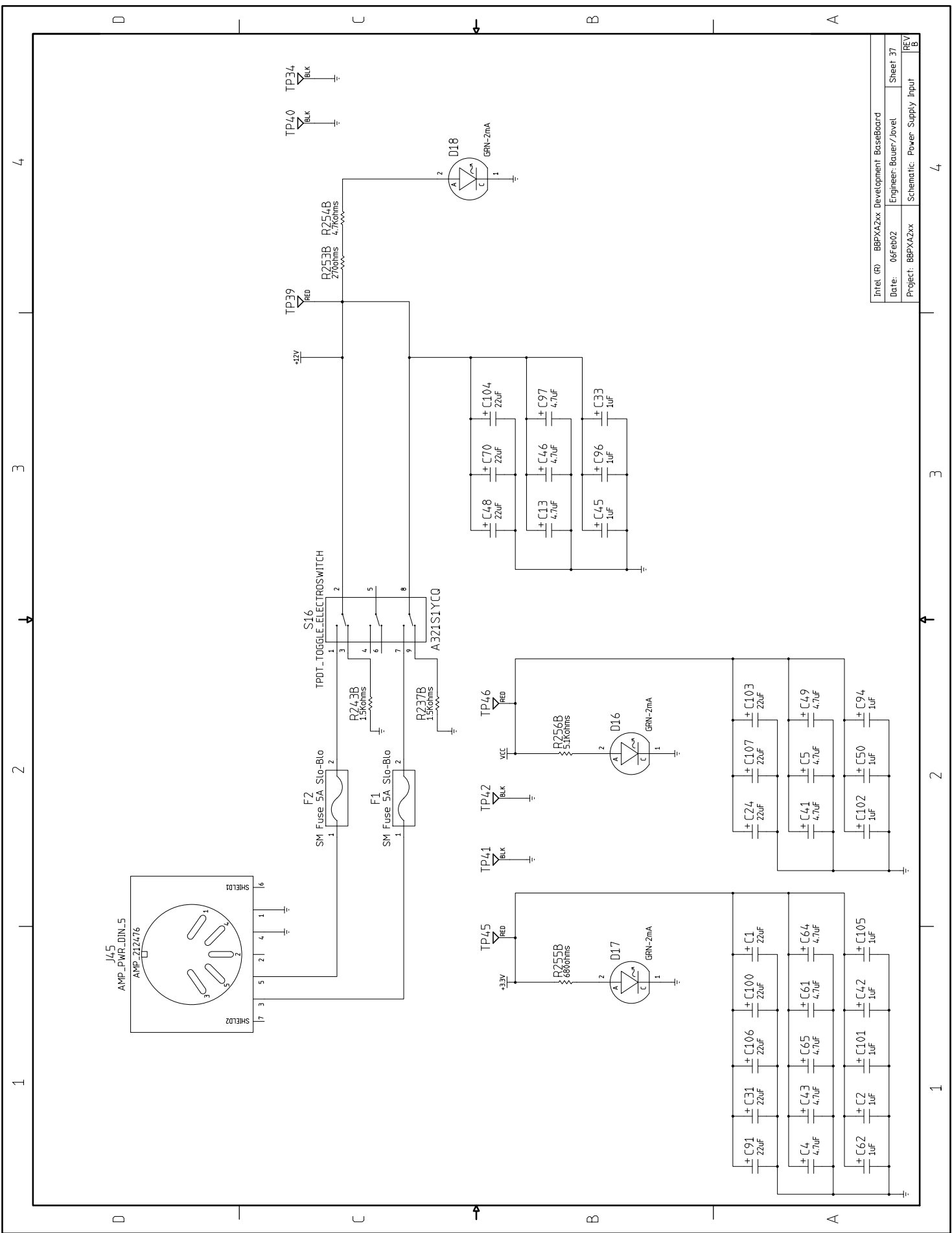




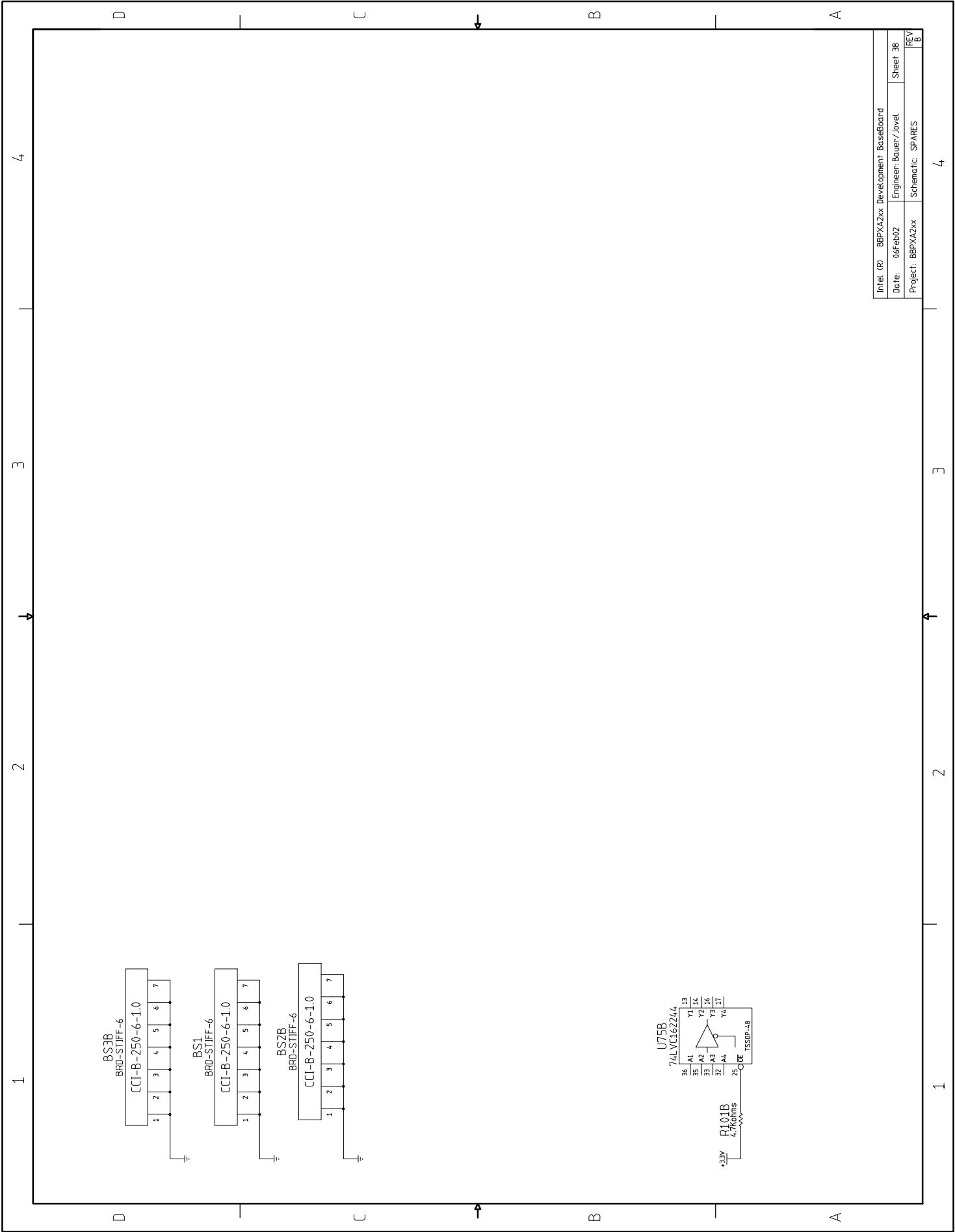
Intel (R) B8PXA2xx Development BaseBoard		
Date: 06Feb02	Engineer: Bauer/Jovel	Sheet 35
Project: B8PXA2xx	Schematic: Reset and Fault Svt	REV B



Intel® BBPX2xx Development BaseBoard			
Date:	06Feb02	Engineer:	Bauer/Jovel
Project:	BBPX2xx	Schematic:	3.3V and 5V Supply
REV	B		



Intel (P)	BBPXA2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPXA2xx	Schematic: Power Supply Input
REV	B	



Intel (R)	BBPX2xx	Development BaseBoard
Date:	06Feb02	Engineer: Bauer/Jovel
Project:	BBPX2xx	Schematic: SPARES
REV	B	



PXA250 Processor Card Schematic Diagram

D

D.1 Schematic Diagram

The DCPXA250 processor card schematic is on the following pages. The PXA255 processor is a drop in replacement for the Intel® PXA250 processor and is fully compatible with these schematics.

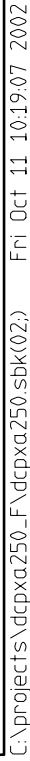
Intel(R) DCPXA250 Processor Card

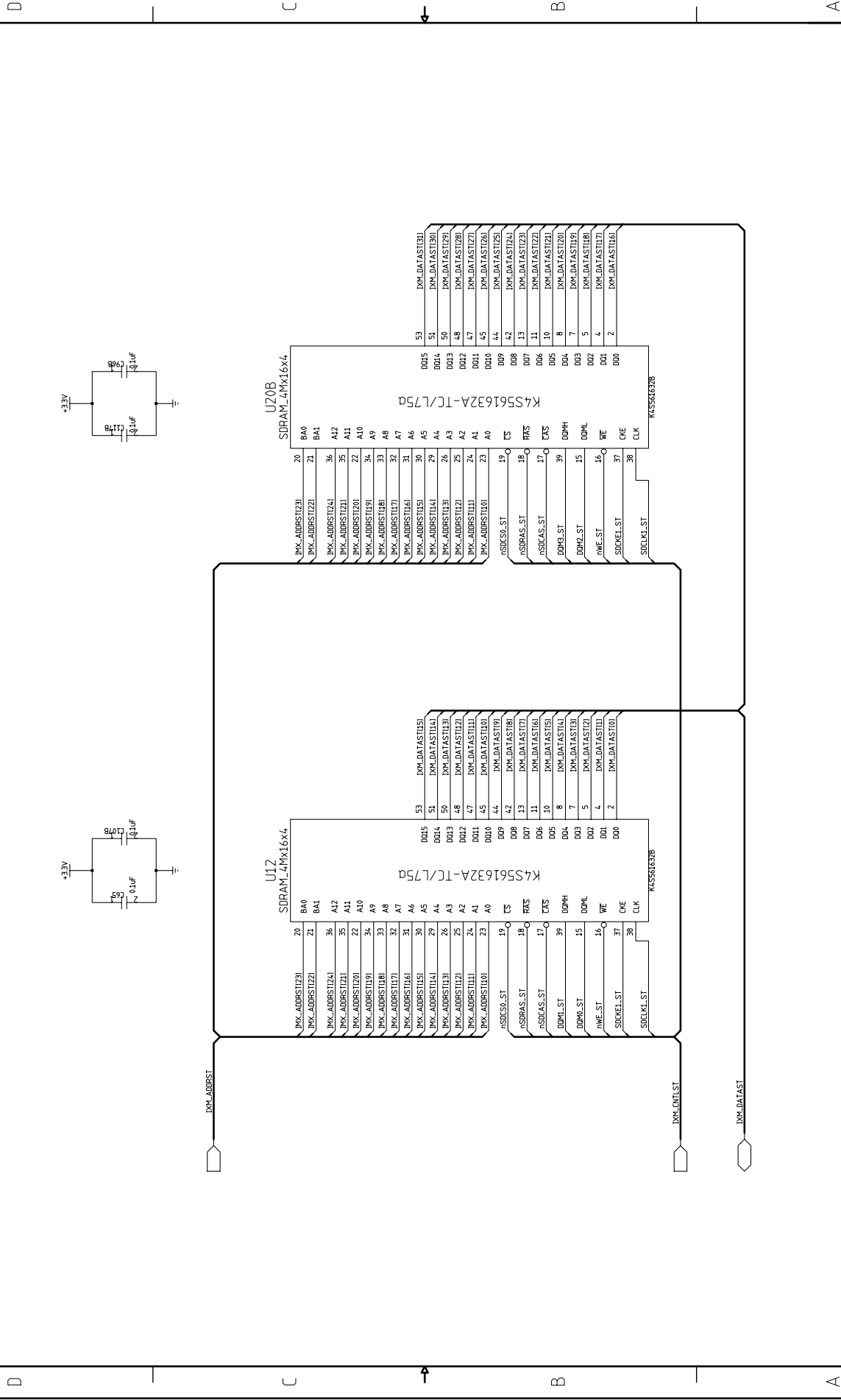
32-bit version

TABLE OF CONTENTS

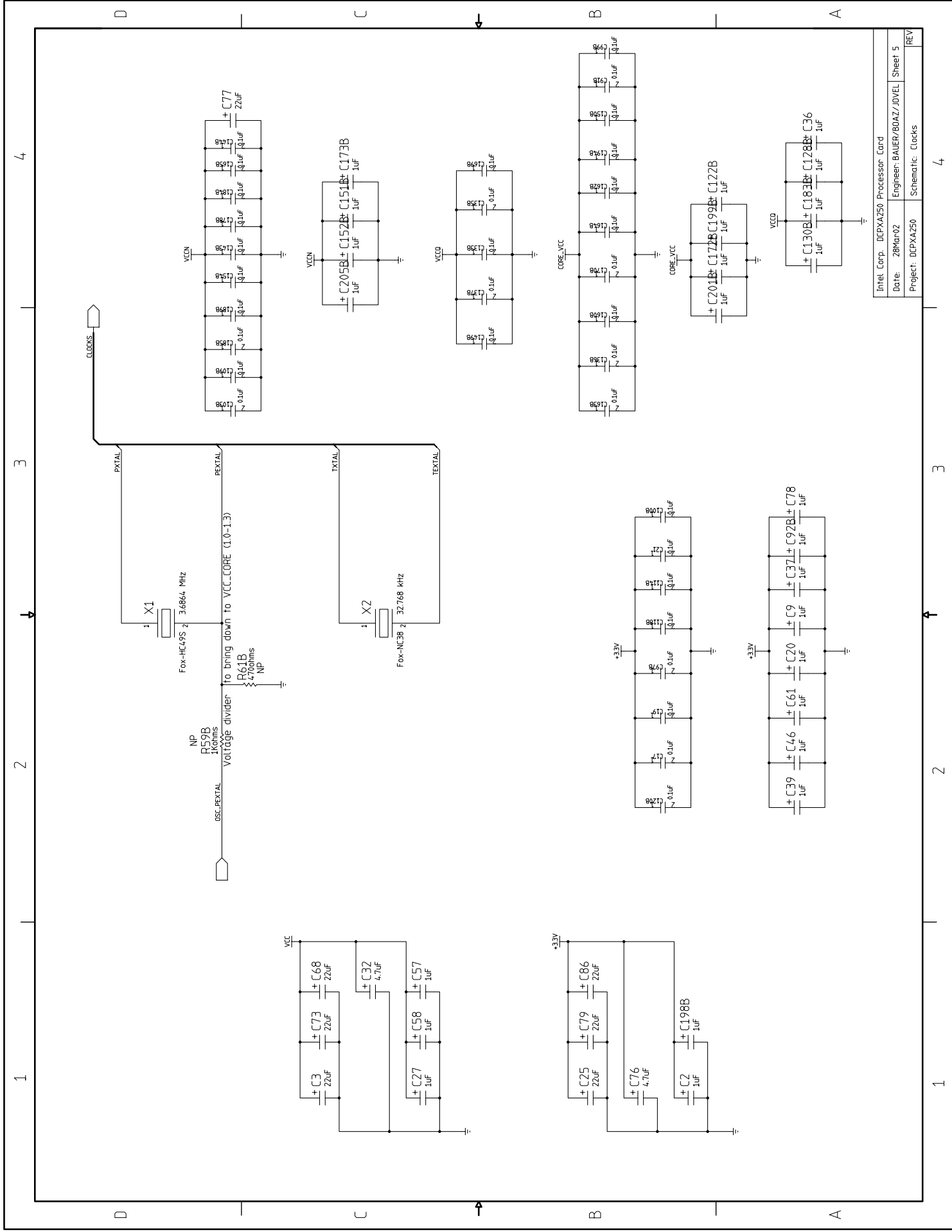
PAGE	FUNCTION
1	TITLE PAGE
2	PROCESSOR --- PXA250
3	SDRAM
4	CONNECTOR (MEMORY and I/O SIGNALS)
5	CLOCKS
6	JTAG CONNECTOR & PLL and CORE VOLTAGE REGULATORS
7	BYPASS CAPS
8-10	Bus Terminators
11-12	Data, Address and Control Transceivers
13	CPLD
14	FLASH

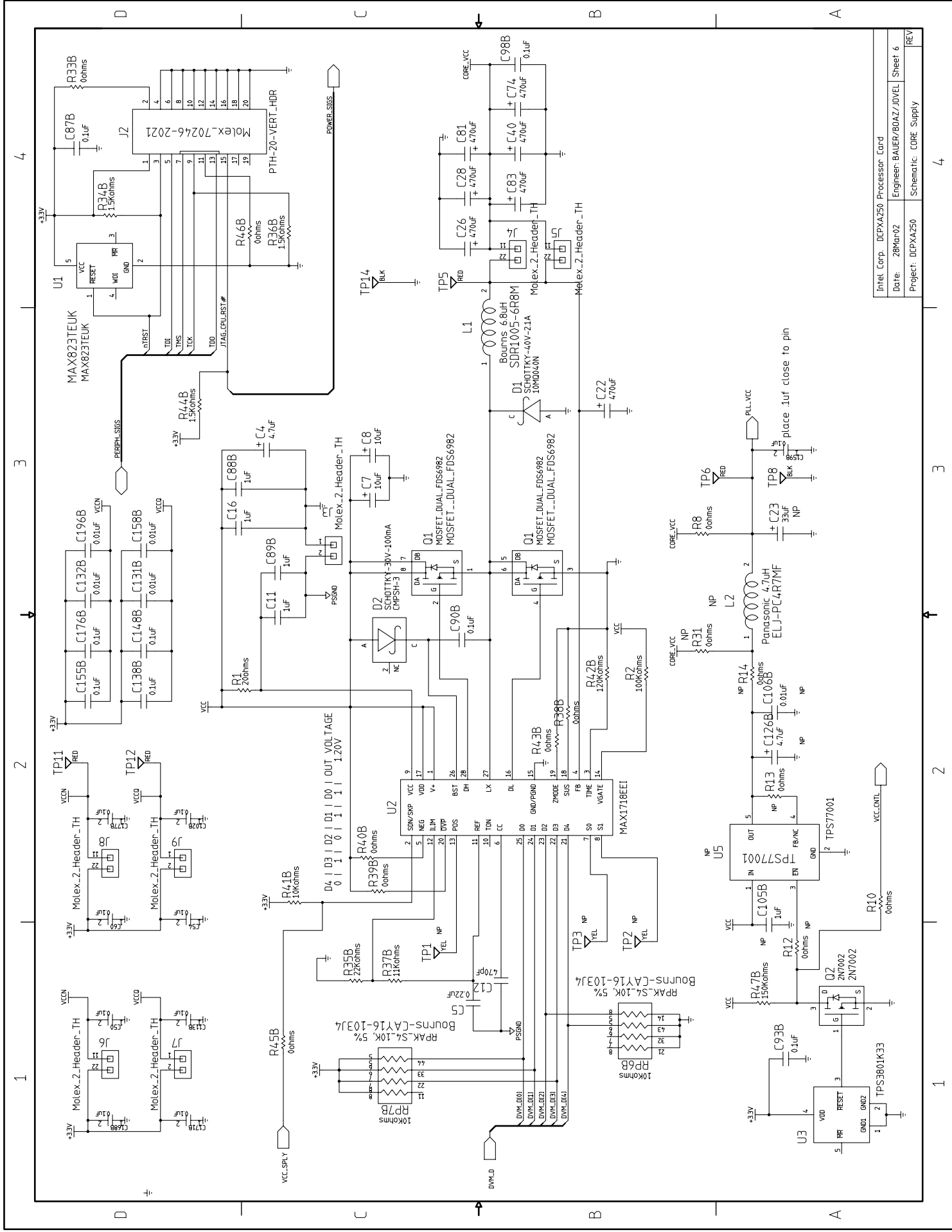
Intel Corp. DCPXA250 Processor Card			
Date:	28Mar02	Engineer: BAUER/BOAZ/JOVEL	Sheet 1
Project:	DCPXA250	Schematic: Title Page	REV





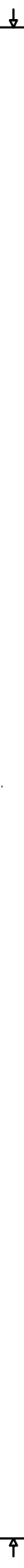
Intel Corp.	DCPXA250 Processor Card
Date:	28Mar02
Project:	DCPXA250
Engineer:	BAUER/BOAZ/JVEL
Schematic:	SDRAM
Sheet:	3
REV	





Intel Corp. DCPXA250 Processor Card	Sheet 6
Date: 28Mar02	Engineer: BAUER/BOAZ/JOVEL
Project: DCPXA250	Schematic: CORE Supply
REV	

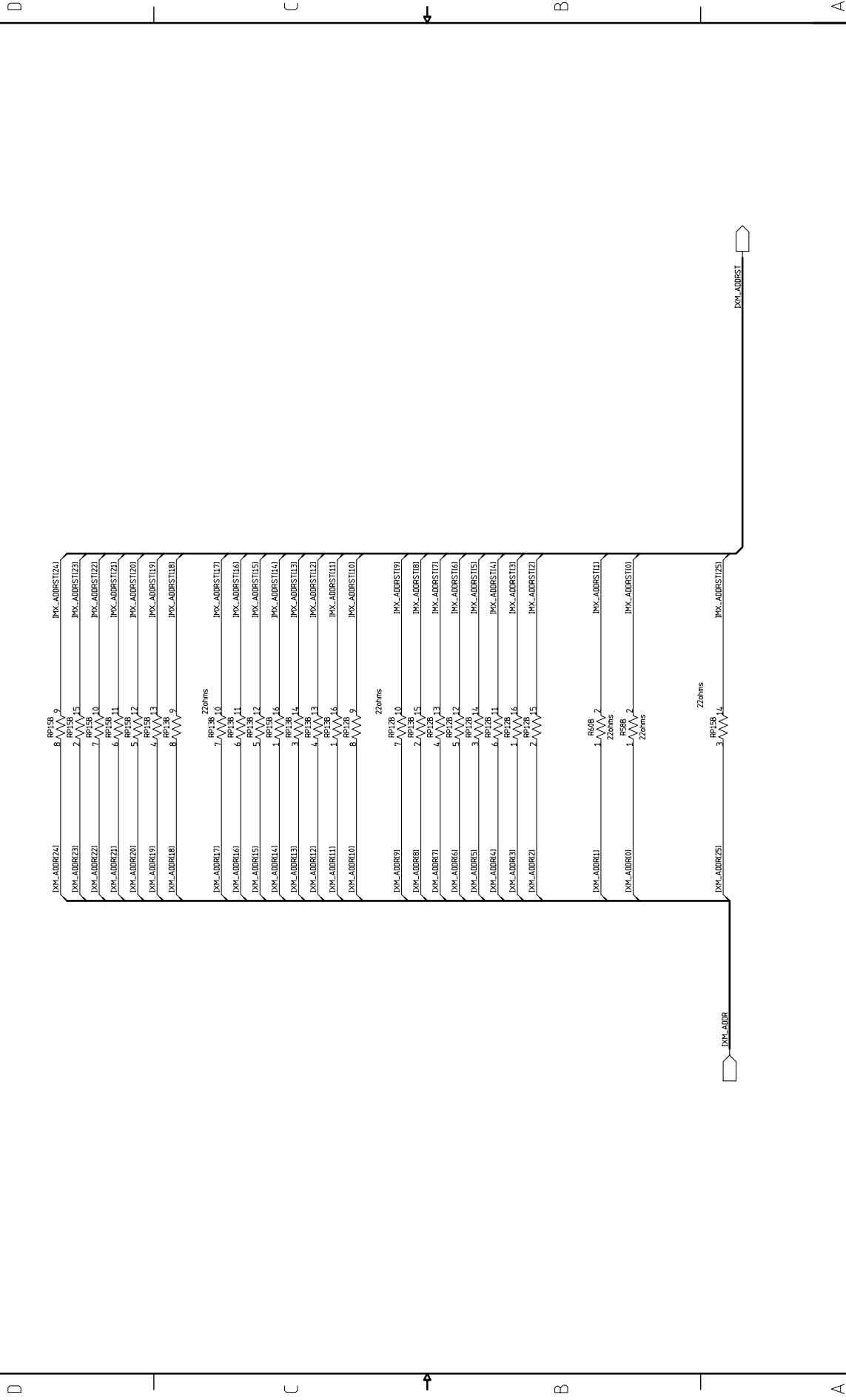




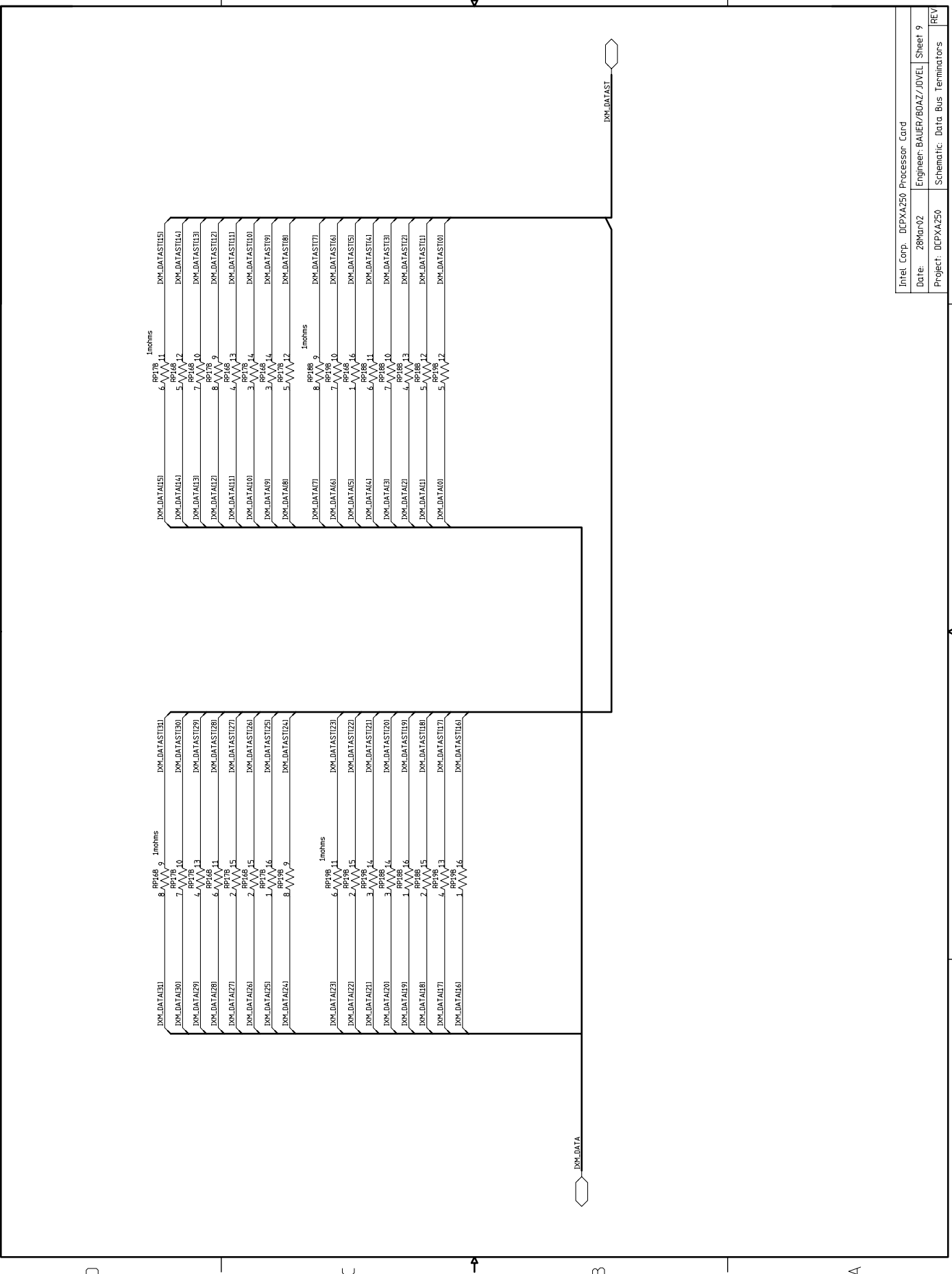
∞

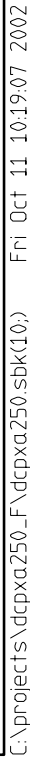


A vertical number line with tick marks at 1, 2, 3, and 4. A point is marked with a dot and labeled 'a' at the position 2.5.



Intel Corp. DCPXA250 Processor Card			
Date:	28Mar02	Engineer: BAUER/BOAZ/JVEL	Sheet 8
Project:	DCPXA250	Schematic: Address Terminators	REV





1

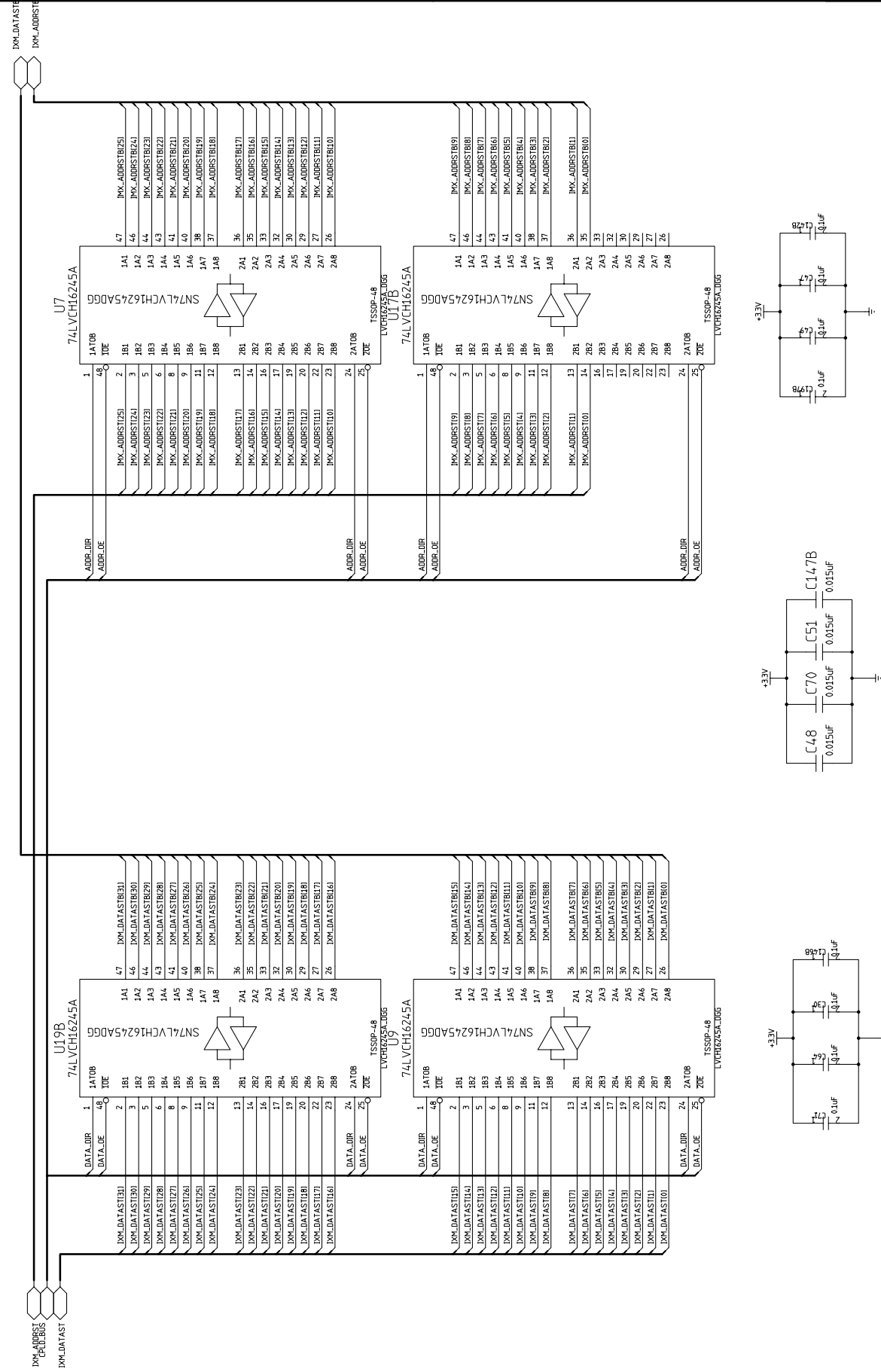
2

3

4

DATA BUFFERS

ADDRESS BUFFERS



Intel Corp. DCPXA250 Processor Card
Date: 28Mar02
Project: DCPXA250
Engineer: BAUER/BOAZ/JVEL
Schematic: Memory Bus Transceivers
Sheet 11
REV

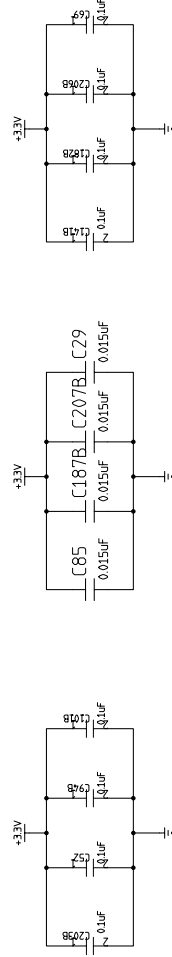
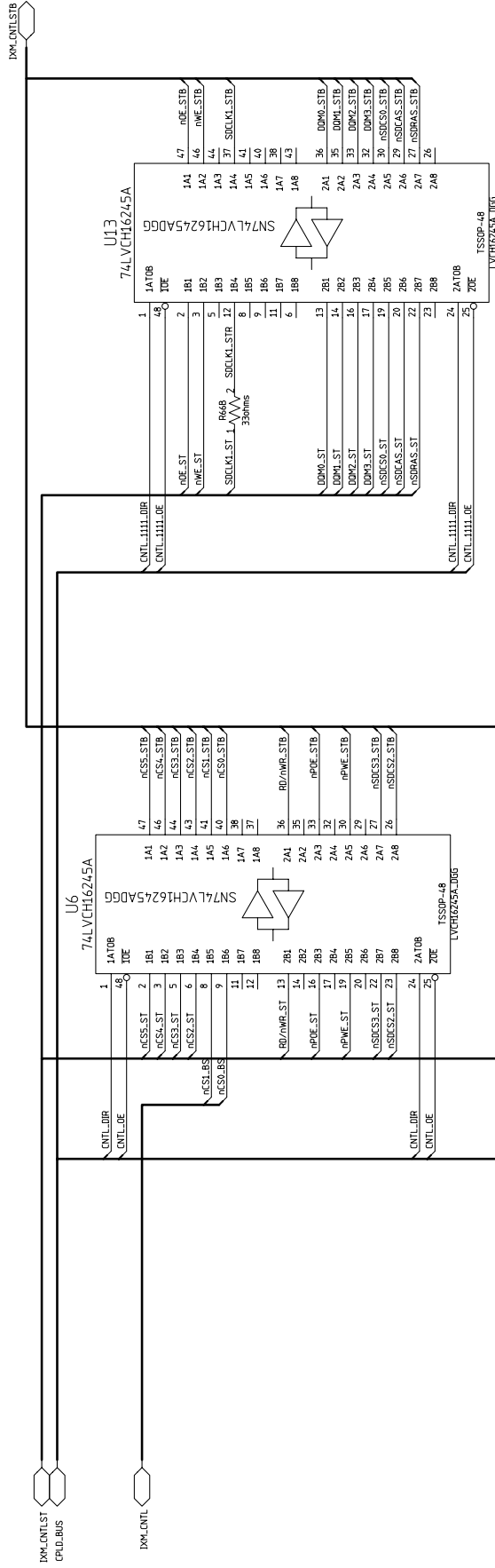
1

2

3

4

CONTROL BUFFERS



Intel Corp. DCPXA250 Processor Card	REV
Date: 28Mar02	Sheet 12
Project: DCPXA250	Schematic: CNTL Transceivers

