



Embedded Pentium[®] Processor- Based POS Terminal Sample Design

Application Note

November 1998

Order Number: 273218-001





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 Pentium® processor 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.

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, 1998

*Third-party brands and names are the property of their respective owners.

Contents

1.0	Introduction.....	7
1.1	Key Terms	7
1.2	Related Documents.....	7
2.0	Embedded Applications Overview.....	8
2.1	Influence of PC Technology	8
2.2	Embedded Pentium Processors.....	8
2.3	Operating Systems for POS Applications.....	9
3.0	Point-of-Sale Terminal.....	9
3.1	Proprietary Operating System Versus Off-the-Shelf	9
3.2	POS Implementation	10
3.3	Windows CE	11
4.0	Embedded Pentium Processors.....	11
4.1	Pentium Processor with MMX Technology.....	14
5.0	POS Terminal Design Overview.....	15
5.1	Core Components	15
5.2	Processor Assembly Components	16
6.0	Functional Description of Hardware	16
6.1	Processor Assembly.....	17
6.1.1	Processor	17
6.1.2	In-Target Probe (ITP)	17
6.1.3	Voltage Regulator.....	17
6.1.4	Intel 430TX PCIsset	17
6.1.4.1	82439TX System Controller (MTXC).....	17
6.1.4.2	Intel 82371AB PCI ISA IDE Xcelerator (PIIX4)	18
6.1.4.3	IDE / Floppy.....	19
6.1.4.4	USB Host Controller	19
6.1.5	Cache	21
6.2	Video	22
6.2.1	Chips and Technologies 69000 HiQVideo Accelerator with Integrated Memory22	
6.2.2	High Performance Integrated Memory	22
6.2.3	HiQColor Technology	22
6.2.4	Versatile Panel Support.....	22
6.2.5	Television NTSC/PAL Flicker Free Output.....	23
6.2.5.1	Design Note.....	23
6.3	Audio	26
6.3.1	AC '97 Audio	26
6.3.2	Ensoniq AudioPCI 97 ES1371 Digital Controller	27
6.3.2.1	Design Note.....	27
6.3.3	Analog Devices AC '97 AD1819 SoundPort Codec	27
6.3.3.1	Design Note.....	27
6.4	SMSC FDC37B78x Ultra I/O	29



6.4.1	Floppy Disk Controller (FDC)	29
6.4.2	Serial Port Controller	29
6.4.3	Infrared Interface	29
6.4.3.1	Design Note	29
6.4.4	Parallel Port Controller	30
6.4.5	Keyboard and Mouse	30
6.4.5.1	Design Note for Ultra I/O	30
6.4.6	Additional Serial Ports with SMSC 37C669 Super I/O	31
6.4.6.1	Design Note	31
6.5	Tritech Microelectronics TR88L803 Touch Screen Controller	32
6.5.1	Design Note	32
6.6	PCMCIA (Strataflash & PC Card) Socket	33
6.6.1	Texas Instruments PCI1221 PC Card (PCMCIA) Controller	33
6.6.1.1	Slot A: PCMCIA Socket with Texas Instruments TPS2206 PC Card Power-Interface Switch with Reset	33
6.6.1.2	Slot B: Intel StrataFlash Memory	33
6.6.1.3	Design Note	34
6.7	ISA PCI Expansion Cards	35
6.7.1	Design Note	35
6.8	Clocking	36
6.8.1	Cypress Semiconductor CY2280	36
6.8.2	Cypress Semiconductor CY2309	36
6.8.3	Clocking for 430TX Chipset	36
6.8.4	Clocking for 440BX Chipset	37
7.0	Design Considerations	37
8.0	Windows CE	38
8.1	Windows CE for Embedded Applications	39
8.2	Windows CE Environment	39
8.2.1	Kernel	39
8.2.2	Modularity	39
8.2.3	OEM Adaptation Layer (OAL)	40
8.3	Windows CE Build Information	40
8.3.1	Getting Started	40
8.3.2	Building a Windows CE Operating System Configuration	40
8.3.3	Building a New Project	41
8.3.3.1	Create a New Platform Directory	41
8.3.3.2	Create a New Project Directory	41
8.3.3.3	Create a New Command Prompt Build Window	42
8.3.3.4	Build the Windows CE Operating System Image for a New Project	42
8.3.3.5	Adding Files or Applications to the Windows CE Operating System Image	42
9.0	Windows CE Device Drivers	43
9.1	Dynamic Link Libraries (DLLs)	43
9.2	Windows CE Device Driver Model	43
9.3	Interaction with Application Software	44
9.4	Incorporating Device Drivers	44
9.4.1	NE2000 ISA Network Interface Cards (NIC)	45



9.4.1.1 NE2000 NIC Configuration 45

9.4.2 ACTiSYS IR 2000B 46

9.4.3 Audio 46

A. Schematics 48

A.1 POS Terminal Baseboard Schematics 48

Figures

1 Stand-alone POS Terminal 9

2 Back End and Front End System 10

3 Pentium Processor with MMX Technology Block Diagram 13

4 POS Terminal Baseboard Block Diagram 15

5 Processor Assembly Block Diagram 16

6 Fence Example 38

7 Device Driver Architecture 44

Tables

1 Related Intel Documents 7

2 Related Specifications and Technical Papers 8

3 General Purpose Input and Output Pins 21

4 Power and Ground Requirements 24

5 AD1819 Codec Filter Pins and Implementations 28

6 SYSOPT Pin Settings 30

7 SYSOPT Setting Options 31

8 SYSOPT Settings 31

9 SYSOPT Setting Options 32

10 Interrupt Implementations 34

11 Multifunction (MFUNC[6:0]) Board Connections 35

12 Board Interrupt Configurations 36

Revision History

Revision	Date	Notes
001	November 1998	First release of this document.



1.0 Introduction

This application note provides a sample point-of-sale (POS) terminal design using the Pentium® processor and provides recommendations to enable shorter design cycles. Also described are the various peripherals commonly used in POS terminals and their interface to a Pentium processor platform.

The tested operating systems are: QNX*, DOS, BeOS*, Windows 95*, Windows 98*, Windows NT*, Windows 3.1*, and Windows CE*. Since Windows CE is a relatively new operating system, drivers have been tested and configuration information for pertinent drivers are provided.

Caution: The design has not been implemented in hardware. This document is for reference only. Customers are responsible for validating designs created using the information in this document.

1.1 Key Terms

PIIX4E refers to the Intel 82371EB PCI ISA IDE Xcelerator.

POS Terminal refers to point-of-sale terminal.

Design Features are items that allow the designer to fully use the capabilities of the embedded Pentium® processor with MMX™ technology and the Intel® 440TX AGPset, and the Pentium® II processor with the Intel® 430TX PCIset.

Design Notes are items that should be considered but may not apply to your design.

1.2 Related Documents

Table 1. Related Intel Documents

Document	Order Number
<i>Embedded Pentium® Processor Family Developer's Manual</i>	273204
<i>Embedded Pentium® Processor with MMX™ Technology datasheet</i>	273214
<i>Intel® 430TX PCIset: 82439TX System Controller (MTXC) datasheet</i>	290559
<i>Intel® 430TX PCIset Design Guide</i>	290613
<i>Intel® 82371EB PCI-to-ISA/IDE Xcelerator (PIIX4E) Specification Update</i>	290635
<i>Intel® 82371AB PCI-to-ISA/IDE Xcelerator (PIIX4) datasheet</i>	290562
<i>Embedded Pentium® Processor with MMX™ Technology Flexible Motherboard Design Guidelines</i>	273206



Table 2. Related Specifications and Technical Papers

Document	URL
PCI Local Bus Specification, Revision 2.1	http://www.pcisig.com/specs.html
Designing with Flash Memory in Windows CE Applications	http://developer.intel.com/design/flash/isf/wince/wince1.htm
Understanding Modularity in Microsoft Windows CE	http://www.microsoft.com/windowsce/embedded/techpapers/wce20/modularity.asp
Introducing the Windows CE Embedded Toolkit	http://www.microsoft.com/windowsce/embedded/techpapers/wce20/intro_etc.asp
CHIPS 69000 HiQVideo Accelerator with Integrated Memory datasheet	http://www.chips.com/design/graphics/mobilegraphics/datashts/ds_69000.htm
Application Notes for the HiQVideo Family	http://www.chips.com/design/graphics/mobilegraphics/applnots/

2.0 Embedded Applications Overview

2.1 Influence of PC Technology

Embedded systems come in a variety of forms and include POS terminals, industrial equipment, and telecommunication equipment.

High performance embedded processors are becoming increasingly popular in embedded applications, and PC technology is quickly filling the performance needs. Using desktop processors in the embedded space with standard operating systems allows embedded designers to get to market quickly. This discussion will address the use of embedded processors in POS terminals.

POS terminals, widely used in supermarkets, department stores, restaurants, convenience stores, and banks are PC-like platforms with peripherals added to address various needs. The main function of a POS is to process transactions including receipt generation, scanning, weighing, and inventory management. A POS can be a stand-alone system or used in conjunction with back-office systems for inventory management, training, and advertising. POS terminal users may need to look up products, re-index merchandise by product code, description or stock number, or track layaway and backorders. When discounts are offered, the POS terminal can support multiple price levels.

2.2 Embedded Pentium Processors

The embedded Pentium processor is inherited from the desktop PC, along with selected chipsets, and have been made available to the embedded systems designer. Intel offers the Pentium processor at 100, 133 and 166 MHz, the Pentium® processor with MMX™ technology at 166, 200 and 233 MHz, and the low-power embedded Pentium processor with MMX technology at 166 and 266 MHz.

Applications using the embedded Pentium processor family provide scalability to designs and offer a wide performance range and an easily alterable platform.

2.3 Operating Systems for POS Applications

In POS applications using the embedded Pentium processor, frequently used operating systems include DOS, QNX, Linux, BeOS, VxWorks, pSOS, Windows 3.1, Windows NT, Windows 95 and Windows 98. Windows 95 and Windows NT meet the increasing performance needs of many embedded applications. Windows NT, Windows 95, and Windows 98 have the advantage of having a large base of applications readily available, and features such as Desktop Management Interface (DMI), and support for interactivity.

Another operating system solution that is gaining momentum in POS applications is Windows CE, which can be used in smaller applications that require a ROM-able operating system.

3.0 Point-of-Sale Terminal

Figure 1. Stand-alone POS Terminal



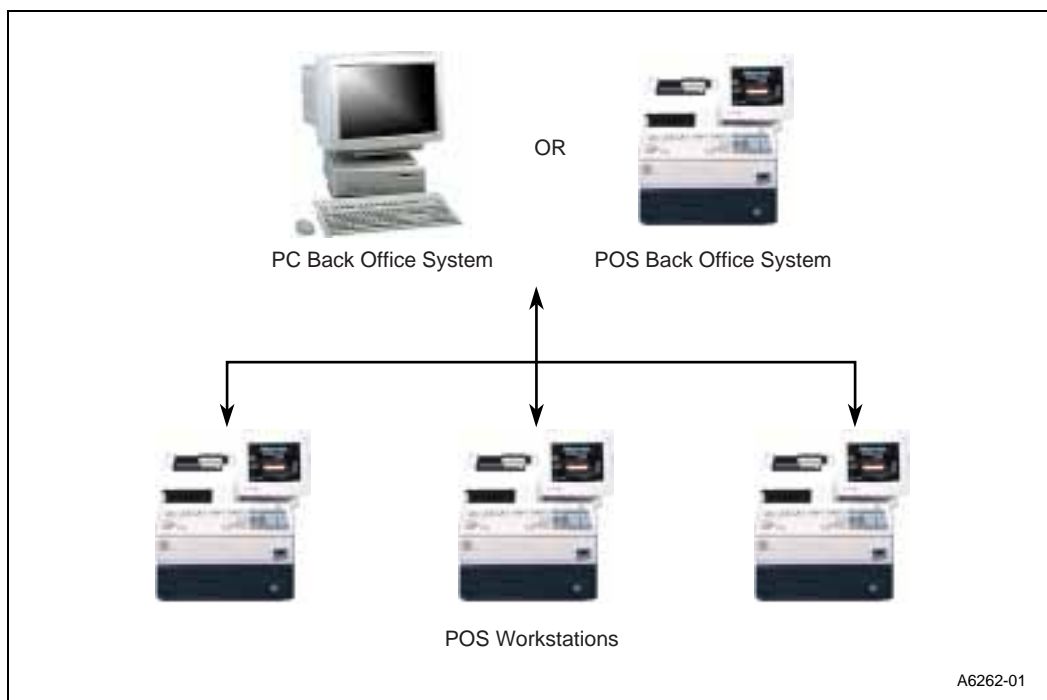
3.1 Proprietary Operating System Versus Off-the-Shelf

A typical POS implementation can include a high performance back end system and a front end system with less functionality. In a traditional POS implementation, a company may have to develop and train people on two or more systems. At the machine level, or front end, where functionality is limited, applications may be developed based on proprietary operating systems. At the back end or supervisory level, applications are developed on open systems based on Windows 95 and Windows NT operating systems. Because these levels are built on different platforms the systems may not talk to each other, look the same, or share databases, graphics and other application files.

Alternatively, a POS implementation can have an embedded Pentium processor with MMX technology, or a Pentium II processor-based back end system with Windows NT as the operating system. This will enable POS system designers to add functions such as inventory management and disaster recovery. A front end POS terminal with an embedded Pentium processor, a Pentium processor with MMX technology, or a Pentium II processor platform running Windows 95, Windows 98, Windows NT, or Windows CE operating system can communicate with the back end

or supervisory level systems. This system allows the sharing of databases, graphics and applications. Refer to Figure 2. By using an off-the-shelf operating system, the added benefits are shorter design cycles and quick time-to-market.

Figure 2. Back End and Front End System



3.2 POS Implementation

POS terminals are PC-like platforms adapted for retail and service environments like fast food restaurants. They may include liquid-crystal display (LCD) touch screens, which are especially important at fast food restaurants where numerous items have to be immediately selected.

Also, unlike a desktop system, a POS terminal can have an expanded number of serial and Universal Serial Bus (USB) ports. These additional serial ports are usually needed for peripherals such as bar code scanners, credit card readers, magnetic stripe readers, and pin pads.

USB keyboards, mice, and receipt printers can be used as hot plug-and-play peripherals. Some of the other peripherals common in POS terminals include cash drawers, digital scales, and pole displays.

In most cases, POS terminals must be available at all times. For this reason, hardware monitoring and the capability for fault recovery are crucial. Since a POS terminal is usually always on, the processor's power management features should be used. This will enable the terminal to be placed in a low-power mode when it is not being used.

This sample design tested peripherals and Windows CE drivers to provide support for: Infrared (IrDA), flash memory, PCMCIA cards, AC '97-compliant audio codec, modem, LCD touch screen, and network interface cards. This sample design also discusses the components used for a PCMCIA card, touch screen controller, LCD/touch screen, additional serial ports, infrared, and AC '97 audio codec.

3.3 Windows CE

All software for Windows CE is based on the Win32* application programming interface (API). Developers can write programs for Windows CE using familiar tools such as Visual C/C++*, or Visual Basic*. Windows CE helps in moving some embedded applications away from proprietary systems so products have a shorter development cycle, fewer costs and less user training. Since Windows CE is ROM-able, a hard disk drive may not be necessary.

Because Windows CE, Windows 95 and Windows NT are based on the same Microsoft technologies, the same development tools and utilities can be used on both systems allowing faster development and a similar look and feel.

For an application with reduced functionality, Windows CE can be used as the core OS since it requires a small footprint. A Windows CE-based device with the kernel, the file system, and the registry can fit into 512 Kbytes ROM, making it a candidate for some embedded systems.

Windows CE must be configured to meet the specific needs of the application. However, it is unique in that functionality must be added to it in the form of installable device drivers. Windows CE has numerous components allowing for different configurations. A small footprint is the key difference between Windows CE and Windows NT applications, something the developer must keep in mind.

4.0 Embedded Pentium Processors

The Pentium processor family uses bus fractioning so that the designer must take into account a 66-MHz bus while the processor clocks internally at 100, 133, 166, 200, 233, and 266 MHz. This technique allows the Pentium processor family of devices to have very good processing power internally while still being relatively easy to design into an application. Refer to the *Embedded Pentium® Processor Family Developer's Manual* (order number 273204).

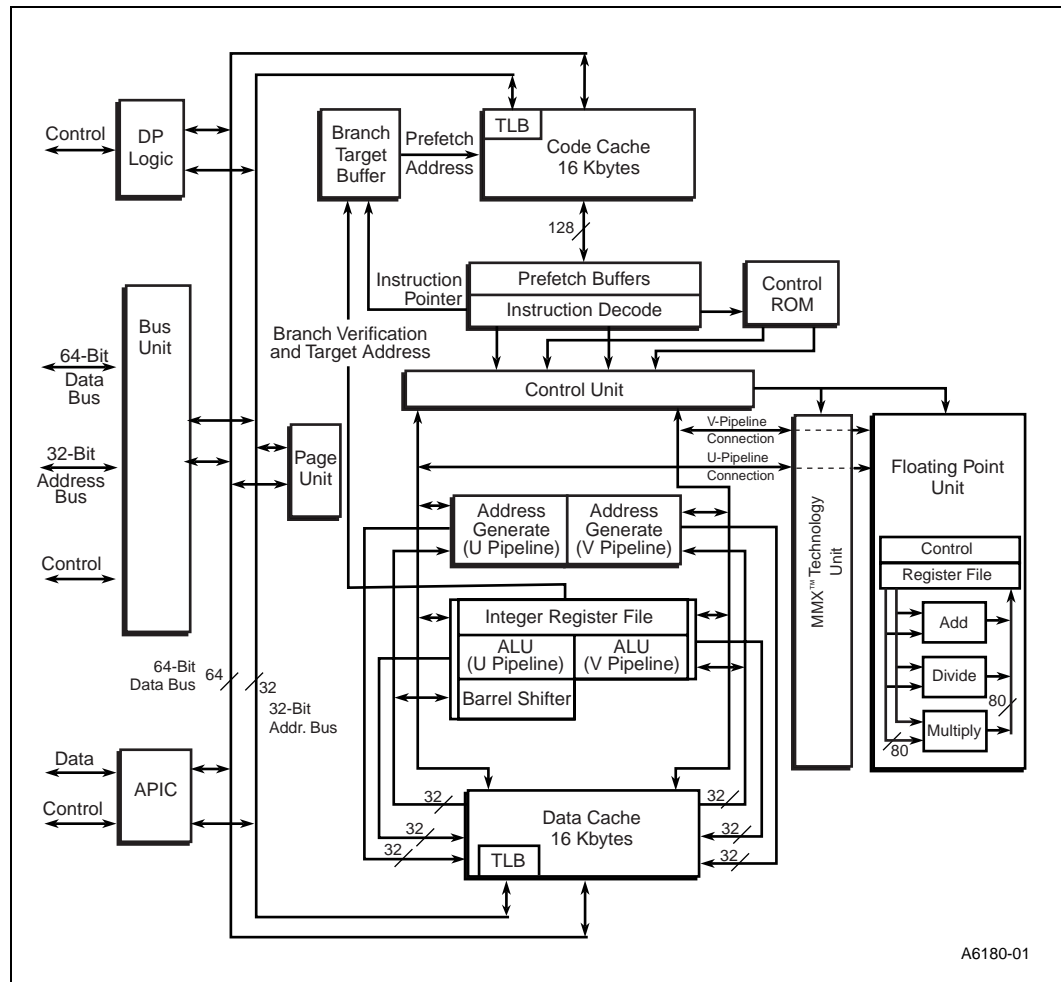
There are two pipeline integer units (U and V) contained in the Pentium processor that gives it the ability to execute two instructions per clock cycle. The U pipeline can execute floating point and integer instructions while the V pipeline executes simple integer instructions and FXCH floating point instructions. Level one (L1) cache structure is separated into code and data, unlike the Intel486™ processor which has a unified (code + data) cache structure. The separate cache are 8 Kbytes each with two-way set associativity on the classic Pentium processors, and 16 Kbytes each with four-way set associativity on the Pentium with MMX technology-enhanced processors. The MESI protocol is also used. Implementing the Pentium processor is binary-compatible all the way back to the Intel 186 processor. Code written for the Intel386™ and Intel486 processors will execute on the Intel Pentium processors without modification. All the processors in the Pentium family are 32 bits internally and have 64 bit external data bus widths. The separate data and instruction caches on the Pentium processors also have separate Translation Look-aside Buffers (TLB) to translate linear addresses into physical addresses. The cache tag scheme employs triple porting to allow two data transfers and an inquiry cycle in only one clock. By default the code cache is write protected for safety. Again a multiporting scheme is employed on the code cache to support snooping.

Another performance enhancement is the implementation of branch prediction. To support this, it was necessary to design two prefetch buffers, the first buffer prefetches code linearly while the second buffer prefetches code according to requests from the branch target buffer (BTB). This technique makes sure that in almost all circumstances the code is prefetched and available before it is needed by the processor execution unit.

The external data bus is 64 bits wide. This is necessary to maximize the data transfer rate. Both burst read and burst write back are supported by the Pentium processor. Bus cycle pipelining allows two separate bus cycles to run concurrently. Memory management unit (MMU) allows 4 Kbytes to 4 Mbyte page sizes.

Figure 2 is a block diagram of the Pentium processor with MMX technology. Refer to the *Embedded Pentium® Processor with MMX™ Technology* datasheet (order number 273214).

Figure 3. Pentium Processor with MMX Technology Block Diagram



The code cache, BTB and prefetch buffers are used to get the instructions into the Pentium processor execution units. Fetching can occur from either the code cache, or in the event of a cache miss, the instruction can be fetched directly from the external system DRAM. Referring to the block diagram in Figure 3, the decode unit decodes prefetched instructions for execution. The control ROM stores the code for the Pentium architecture in micro-code, this micro-code controls the sequence of operations in support of the Pentium architecture. Since the Pentium processor is based on CMOS technology, as the clock is slowed or stopped, the processor dissipates less and less power until at clock stop power dissipation is almost zero and comprises leakage currents through the silicon substrate. This clock stop capability makes the Pentium processor family ideal for energy efficient embedded applications.

With the many Pentium processors offered it should be a relatively easy task to choose a processor and chipset combination. With the fact that the chipsets support PCI v2.1, ISA, and IDE interfaces it is a safe bet that most if not all applications will be supported with this combination. The fact that the chipsets are 3.3 V_{CC} with low-power dissipation in their own right makes integration into an application much more straightforward.

The Pentium processor and Pentium II processor families give the applications designer a plethora of performance and power choices. Whether the application utilizes a 100 MHz or a 233 MHz based processor, the fact that the processor is fabricated on small geometry CMOS technology gives the application designer a relatively low-power, very high performance processor in a small form factor package.

The Pentium processors also contain an Advanced Programmable Interrupt Controller or APIC. This supports I/O, is 8259A compatible and provides inter-processor interrupt support in multi-processor based systems.

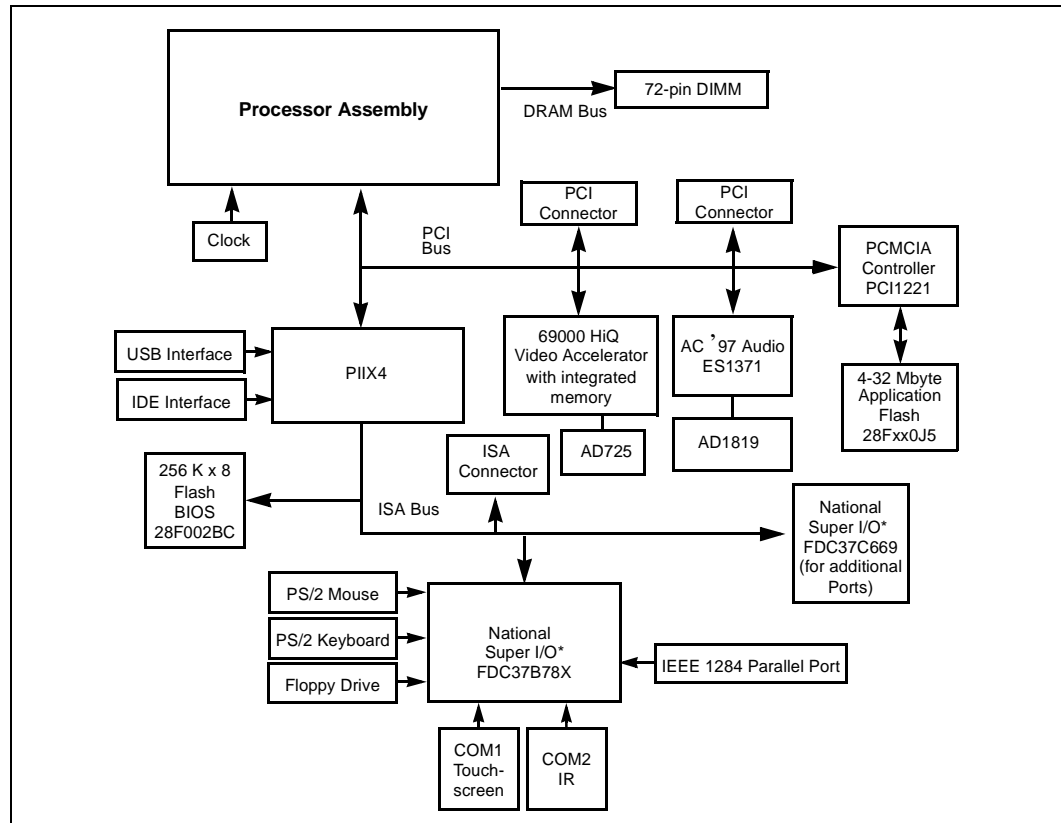
4.1 Pentium Processor with MMX Technology

MMX technology is an outstanding addition to the capabilities of the Pentium processor family. Pentium processors with MMX technology are pin compatible with the classic versions of this processor. With the addition of voltage reduction technology the additional processing capabilities brought to bear with MMX technology result in thermal specifications that are within the Pentium processor Classic guidelines. The MMX enhanced processors utilizes a new Single Instruction Multiple Data or SIMD capability specifically targeted for motion video decode and communications applications. Fifty-seven new opcodes have been added to the base code set in support of MMX technology. Compatibility with the Pentium processor is assured by careful architectural design with a performance increase and no apparent penalty in power dissipation. The BTB has also been improved on the MMX enhanced processors to increase accuracy in operation. With the addition of four prefetch buffers on the MMX enhanced processors we can now simultaneously hold four separate code streams. In the MMX family of processors we also added another stage to the pipeline with other enhancements to further boost performance. For an in depth look at the Pentium processor with MMX technology see the *Embedded Pentium Processor Family Developers Manual* (order number 273204).

5.0 POS Terminal Design Overview

Figure 4 is the block diagram for the POS terminal design.

Figure 4. POS Terminal Baseboard Block Diagram



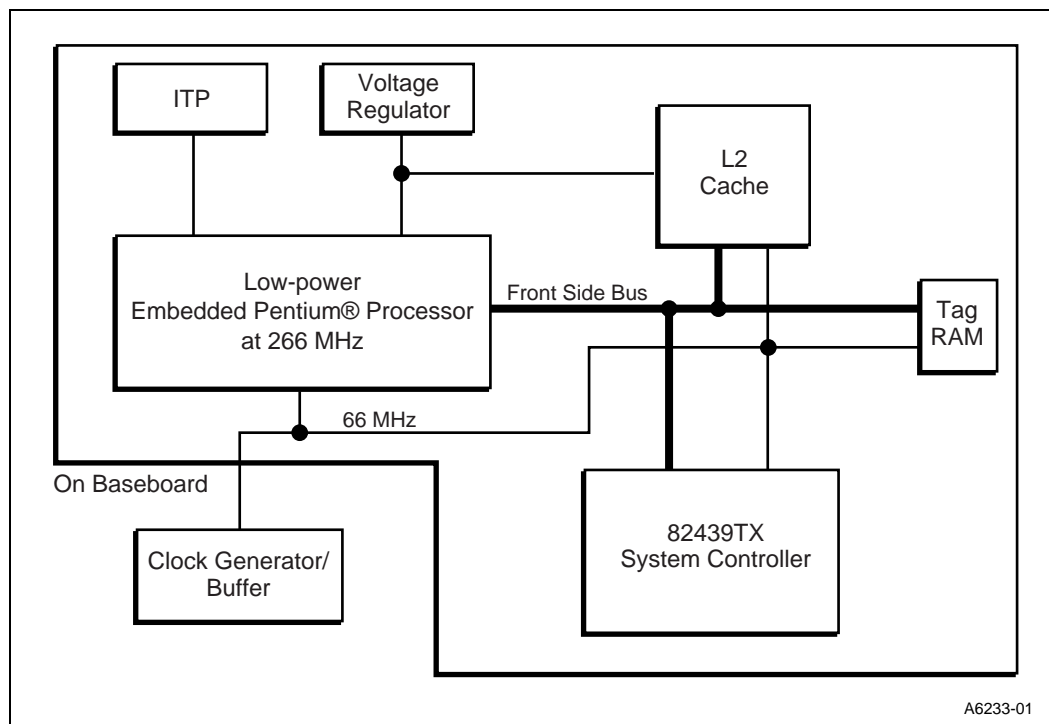
5.1 Core Components

As shown in Figure 4, the core components of this reference design are:

- Intel embedded processor assembly
- 82371AB PCI-TO-ISA/IDE Xcelerator (PIIX4)
 - IDE connector
 - Universal Serial Bus host controller
- Cache
- Chips and Technologies 69000 HiQVideo* Accelerator with Integrated Memory
- AC '97 Audio Solution
 - Ensoniq AudioPCI* ES1371
 - AD1819 SoundPort* Codec
- SMSC FDC37B78x Ultra I/O

- SMSC FDC37C669* Super I/O for additional serial ports
- TriTech Microelectronics TR88L803* Touch Screen Controller
- Texas Instruments PCI1221* PC Card (PCMCIA) Controller

Figure 5. Processor Assembly Block Diagram



5.2 Processor Assembly Components

- Low-power embedded Pentium processor at 266 MHz
- 82439TX System Controller
- Second Level (L2) Cache
- Clock Generator/Buffer
- In-Target Probe (ITP)
- Voltage Regulator

6.0 Functional Description of Hardware

For more information about the POS terminal design, please refer to the schematics located in Appendix A.

6.1 Processor Assembly

The processor assembly is designed to aid in the design of the processor, the 82439TX system controller, and the level two cache interface. Integrating these three components resolves many of the high-speed design issues. This allows the POS vendor to focus on the added functions of the peripheral components described in this document.

Note: Although many design issues are resolved by the processor assembly, line lengths for PCI system and the memory bus must also be taken into consideration.

6.1.1 Processor

This reference design supports the Pentium processor with MMX technology at 266 MHz. This processor is available in a plastic ball grid array (PBGA) package and, if power consumption is a concern, can be operated at 166 MHz to save power and solve thermal issues that sometimes arise with certain designs.

This design provides scalability to a Pentium II processor module through an interposer card. For more information on using the Pentium II processor, refer to Intel's Developer Site at: <http://developer.intel.com>.

6.1.2 In-Target Probe (ITP)

The processor assembly comes populated with an ITP, also called a Pentium debug port. This is a 20-pin interface that communicates directly to the processor, allowing access to the processor's registers and signals. Using this debug utility will allow easier debugging of low-level code like BIOS and driver development.

The connectors for the ITP are provided by AMP Incorporated (part number 104068-1). Debuggers can be purchased through a variety of vendors. For information on implementing ITP in another design, refer to the *Embedded Pentium® Processor Family Developer's Manual* (order number 273204).

6.1.3 Voltage Regulator

The processor assembly comes equipped with a Linear Technologies voltage regulator (part number LTC 1438C6-ADJ).

6.1.4 Intel 430TX PCIset

The Intel 430TX PCIset (430TX) consists of the 82439TX System Controller (MTXC) and the 82371AB PCI ISA IDE Xcelerator (PIIX4). The 430TX supports both mobile and desktop architectures. The 430TX forms a Host-to-PCI bridge and provides second level cache control and a full function 64-bit data path to main memory.

6.1.4.1 82439TX System Controller (MTXC)

The MTXC integrates the L2 cache and main memory DRAM control functions and provides bus control to transfers between the CPU, L2 cache, main memory, and the PCI Bus. The L2 cache controller supports a write back cache policy for L2 cache sizes of 256 Kbytes and 512 Kbytes. Cacheless designs are also supported. The L2 cache memory can be implemented with pipelined burst SRAMs or DRAM. An external Tag RAM is used for the address tag and an internal TAG

RAM for the cache line status bits. For the MTXC DRAM controller, six rows are supported for up to 256 Mbytes of main memory. The MTXC is highly integrated by including the Data Path into the same BGA chip. Using the snoop ahead feature, the MTXC allows PCI masters to achieve full PCI bandwidth. For increased system performance, the MTXC integrates posted write and read prefetch buffers. The 430TX integrates many Power Management features that enable the system to save power when the system resources become idle.

Design Note

When using Pentium® II processor Mobile Module: EMC-2 at 266 MHz with an Intel 440BX AGPset through an interposer card, the MAB#9 signal must be pulled high to 3.3 V through a 10 K Ω resistor. This is to disable the advance graphics port (AGP) on the 440BX Chipset.

The MAB#12 signal is jumpered on the board to allow 66-MHz and 100-MHz bus speed for the 440BX Chipset.

Note: The Intel 440TX PCIset does not support AGP.

6.1.4.2 Intel 82371AB PCI ISA IDE Xcelerator (PIIX4)

The 82371AB PCI ISA IDE Xcelerator (PIIX4) is a multi-function PCI device implementing a PCI-to-ISA bridge function, a PCI IDE function, a Universal Serial Bus host/hub function, and an Enhanced Power Management function.

This device is compatible with both the Intel 440BX and 430TX chipsets, providing an easy processor upgrade path. The PIIX4 also allows complete Plug and Play compatibility. It supports two IDE connectors (Ultra DMA/33) to be used with up to four IDE devices in bus master mode.

As a PCI-to-ISA bridge, PIIX4 integrates many common I/O functions found in ISA-based PC systems—two 82C37 DMA Controllers, two 82C59 Interrupt Controllers, an 82C54 Timer/Counter, and a Real Time Clock. In addition to compatible transfers, each DMA channel supports Type F transfers. PIIX4 also contains full support for both PC/PCI and Distributed DMA protocols implementing PCI-based DMA. The Interrupt Controller has edge- or level-sensitive programmable inputs and fully supports the use of an external I/O Advanced Programmable Interrupt Controller (APIC) and Serial Interrupts. Chip select decoding is provided for BIOS, Real Time Clock, Keyboard Controller, second external microcontroller, as well as two Programmable Chip Selects. PIIX4 provides full plug-and-play compatibility. PIIX4 can be configured as a Subtractive Decode bridge or as a Positive Decode bridge. This allows the use of a subtractive decode PCI-to-PCI bridge such as the Intel 380FB PCIset which implements a PCI/ISA docking station environment.

PIIX4 supports Enhanced Power Management, including full Clock Control, Device Management for up to 14 devices, and Suspend and Resume logic with Power On Suspend, Suspend to RAM or Suspend to Disk. It fully supports Operating System Directed Power Management via the Advanced Configuration and Power Interface (ACPI) specification. PIIX4 integrates both a System Management Bus (SMBus) Host and Slave interface for serial communication with other devices. Refer to the *82371AB PCI-TO-ISA/IDE Xcelerator (PIIX4)* datasheet (order number 290562) for further information.

Design Note

The Intel 82371AB is the PCI south bridge. It connects to the processor assembly via the PCI bus. The Intel 82371AB contains the PCI and ISA interrupt controller, along with various ISA legacy functions such as a DMA controller, a bus master IDE Interface, and ISA bus interface, an ISA bus clock control, a XD bus control, a USB interface and a BIOS ROM interface.

- PCI: Refer to the PCI section for more information on PCI signals on the board. The PIIX4 is PCI device #7 (IDSEL connected to pin AD18 through a 220 Ω resistor per PCI Specification, v2.1).
- ISA: Refer to the ISA section for more information on ISA signals on the board. The PIIX4's interrupt control unit provides interrupt handling to all ISA devices on the board including SMSC FDC37B78x Ultra I/O (floppy disk, serial ports, parallel port control), and SMSC 37C669 Super I/O (2 serial ports).

6.1.4.3 IDE / Floppy

The PIIX4 supports two IDE connectors for up to four IDE devices providing an interface for IDE hard disks and CDROMs. Up to four IDE devices can be supported in bus master mode. The PIIX4 contains support for "Ultra DMA/33" synchronous DMA compatible devices. This design uses one IDE connector with support for up to two devices.

6.1.4.4 USB Host Controller

The PIIX4 contains a Universal Serial Bus (USB) Host Controller that is Universal Host Controller Interface (UHCI) compatible. The Host Controller's root hub has two programmable USB ports.

- X-Bus Signals
XOE# and XDIR# are connected as control signals to the X-Bus transceiver, with XOE# connected to G# and XDIR# connected to DIR of the transceiver. The internal RTC of the PIIX4 is used, therefore RTCALE and RTCCS# become general purpose outputs (GPO25 and GPO24 respectively) by programming the General Configuration Register (GENCFG) in Function 0, Offset B0h–B3h.
- Power Management Signals
SUSA# of the PIIX4 is connected to the PWR_DWN# signal of the clock generator. SUSA# is asserted during power management suspend states, POS, STR, and STD suspend states. PWR_DWN# is an active low control input to the clock generator to power down the device. CPU_STP# and PCI_STP# of the PIIX4 are asserted low to disable the processor and PCI clock outputs respectively. They are connected to CPU_STOP# and PCI_STOP# of the clock generator.
SUSC# of the PIIX4 is first inverted and then connected directly to PS_ON# of the ATX power supply. Therefore when SUSC# is asserted, during STD suspend state, all power rails including 3.3 V, 5 V, $\bar{5}$ V, 12 V, and $\bar{12}$ V power rails, are turned off. This is used for the remote-off function. The inverter described above *must* be connected to a Standby Power rail. ZZ is connected to the ZZ pin on the L2 cache on the processor assembly for low-power mode.
RSMRST# connection requires a minimum time delay of one millisecond from the rising edge of the standby power supply voltage. A simple RC circuit is used to provide this time delay ($t = RC$, in the design this is $\sim 2.7 \text{ K}\Omega * 10 \mu\text{F} \sim 2.7 \text{ ms}$) and a Schmitt trigger circuit is used to drive the signal on the board.
- USB Interface

The following are general layout guidelines for the USB interface:

- Any unused USB ports should be terminated with 15 K Ω pull-down resistors on both P+/P- data lines
- 27 Ω series resistors should be placed as close as possible to the PIIX4 (<1 inch). These series resistors are for source termination of the reflected signal.
- 47 pF caps must be placed as close to the PIIX4 as possible and on the PIIX4 side of the series resistors on the USB data lines (P0+, P1+). These caps are there for signal quality (rise/fall time) and to help minimize electromagnetic interference (EMI).
- 15 K Ω +5% pull-down resistors should be placed on the USB side of the series resistors on the USB data lines (P0+ and P1+), and are REQUIRED for signal termination by USB specification. The length of stub should be as short as possible.
- The trace impedance for the P0+, P1+ signals should be 45 Ω (to referenced ground) for each USB signal P+ or P-. The impedance is 90 Ω between the differential signal pairs P+ and P- to match the 90 Ω USB twisted pair cable impedance. Note the twisted pair characteristic impedance of 90 Ω is the series impedance of both wires, resulting in an individual wire presenting a 45 Ω impedance. The trace impedance can be controlled by carefully selecting the trace width, trace distance from power or ground planes, and physical proximity of nearby traces.
- USB data lines must be routed as 'critical signals' (i.e., hand routing preferred). The P+/P- signal pair must be routed together and not parallel with other signal traces to minimize crosstalk. Doubling the space from the P+/P- signal pair to adjacent signal traces will help to prevent crosstalk. Do not worry about crosstalk between the two P+/P- signal traces. The P+/P- signal traces must also be the same length. This will minimize the effect of common mode current on EMI. (Common mode current is caused by differential signals whose currents are not perfectly matched.)

The following are general layout guidelines for the USB power and ground lines:

- Ferrite beads are placed on each power and ground line to minimize EMI. They should be placed as close as possible to the USB connector.
- Voltage divider circuits are used to drive the status of the SUB power line to the OC[1:0]# inputs. OC[1:0]# signals are 3.3 V inputs and have a leakage current of maximum + 1 μ A.
- Fuses are used on each power line for overcurrent protection.

Refer to the *Intel® 430TX PCIset Design Guide* for sample layout topologies.

- IDE Interface

Two standard IDE interfaces are provided by the PIIX4. One IDE interface (primary) is included in this design. The primary IDE controller is connected as described in the TX Chipset design Guide. The secondary IDE controller is not used on the board. Hence all inputs to the secondary IDE controller are pulled to the inactive state with a 10 K Ω resistor. All outputs and bidirectional pins are left floating.

A Hard Drive Active LED circuit is connected to the HD_ACT# signal of the primary IDE connector. If the secondary IDE controller is to be used, an OR-gate between the HD_ACT1# and the HD_ACT2# should be used to drive the LED circuit.

Proper operation of the IDE circuit depends on the total length of the IDE bus. The total signal length of the IDE drivers to the end of the IDE cables should not exceed 18". Therefore, the PIIX4 should be located as close as possible to the IDE connectors to allow the IDE cable to be as long as possible. When the distance between the PIIX4 and the ATA connectors exceeds four inches the series termination resistors should be placed within one inch of the PIIX4. Designs that place the PIIX4 within four inches of the ATA connectors can place the series

resistors anywhere along the trace. The capacitance of each pin of the IDE connector on the host should be below 25 pF when the cables are disconnected from the host.

- Clocks
Refer to the *Intel® 430TX PCIsset Design Guide* for layout topologies.
- General Purpose Input and Output Pins
15 general purpose inputs and 15 general purpose outputs are provided on the board. A table of the pins and sharing information is below:

Table 3. General Purpose Input and Output Pins

GPO	Pin	Muxed With	GPI	Pin	Muxed With
GPO0	1	None	GPI1	2	None
GPO8	3	None	GPI2	4	REQA#
GPO9	5	GNTA#	GPI3	6	REQB#
GPO10	7	GNTB#	GPI4	8	REQC#
GPO11	9	GNTC#	GPI5	10	APICREQ#
GPO12	11	APICACK#	GPI7	12	None
GPO15	13	APICCS#	GPI13	14	None
GPO21	15	SUS_STAT2#	GPI14	16	None
GPO24	17	RTCCS#	GPI15	18	None
GPO25	19	RTCALE	GPI16	20	None
GPO26	21	KBCCS#	GPI17	22	None
GPO27	23	None	GPI18	24	None
GPO28	25	None	GPI19	26	None
GPO29	27	IRQ9OUT#	GPI20	28	None
GPO30	29	None	GPI21	30	None

NOTE:

1. All GPIO pins in the sample design are always available. Pins that are multiplexed with other signals are not used for their alternate function and are therefore always available for GPIO.

6.1.5 Cache

The level two (L2) cache concept developed from the need to have data and instructions available to the processor locally. With the Intel386 processor running at 25 MHz, there is enough time to allow for a code fetch and data to arrive from system DRAM for the processor to use before a stall occurs. With the advent of faster processors, the need for L2 cache has become even greater. Now the processor is fast enough that without cache, the processor core can stall while waiting for code and data to arrive from the system DRAM. While the processor is stalled waiting for instructions, it cannot perform other tasks.

Intel’s dual 8-Kbyte (non-MMX-enhanced) with 2-way set associativity or dual 16-Kbyte (MMX-enhanced) with 4-way set associativity level one (L1) caches on the Pentium processor(s), with the addition of a 512-Kbyte L2 cache system, greatly lessens bus utilization and increases the overall throughput of the system. The baseboard has been designed with 512 Kbyte of fast static RAM cache available. The devices are (part number: KM732V696T-13 manufactured by Samsung Electronic Company) supplied in a 100-pin Thin Quad Flatpack-leded (TQFP) surface package

mount. The devices are Synchronous Pipelined Burst SRAM architecture. The I/O is 2.5 V while V_{CC} is 3.3 VDC. The cycle time is 7.8 ns and the array is 64 K x 32 K. This architecture allows easy interface to the 440TX chipset without the need for level shifters.

6.2 Video

The Chips and Technologies 69000 HiQVideo Accelerator with integrated memory is used in this design. The 69000 solution integrates 2 Mbyte SDRAM for use in frame buffering. This memory supports up to an 83 MHz clock, allowing a high memory bandwidth.

This part supports a variety of outputs, including monochrome, color Single Panel, Single-Drive (SS), Dual-Panel, Dual-Drive (DD), and standard / high resolution, passive STN and active matrix TFT/MIM LCD, as well as EL panels and can be customized.

6.2.1 Chips and Technologies 69000 HiQVideo Accelerator with Integrated Memory

The CHIPS 69000 is a portable graphics accelerator that integrates high performance memory technology for the graphics frame buffer. Based on the HiQVideo graphics accelerator core, the 69000 combines a flat panel controller capabilities with low-power, high performance integrated memory.

6.2.2 High Performance Integrated Memory

The 69000 incorporates 2Mbyte of proprietary integrated SDRAM for the graphics/video frame buffer. The integrated SDRAM memory can support up to 83 MHz operation, increasing the available memory bandwidth for the graphics subsystem.

6.2.3 HiQColor Technology

The 69000 integrates Chips and Technologies HiQColor technology based on the CHIPS proprietary Temporal Modulated Energy Distribution (TMED) algorithm, HiQColor technology is a process that allows the display of 16.7 million true colors of STN panels without using Frame Rate Control (FRC) or dithering. TMED also reduces the need for the panel tuning associated with current FRC-based algorithms. The TMED algorithm eliminates all of the flaws (such as shimmer, Mach banding, and other motion artifacts) normally associated with dithering and FRC.

6.2.4 Versatile Panel Support

The HiQVideo family supports a wide variety of monochrome and color single-panel, single-drive (SS) and dual-panel, dual drive (DD), standard and high-resolution, passive STN, and active matrix TFT/MIM LCD, and EL panels. With HiQColor technology, up to 256 gray scales are supported on passive STN LCDs. Up to 16.7 million different colors can be displayed on passive STN LCDs and on 24-bit active matrix LCDs.

The 69000 offers a variety of programmable features to optimize display quality. Vertical centering and stretching are provided for handling modes with less than 480 lines on 480-line panels. Horizontal and vertical stretching capabilities are also available for both text and graphics modes for optimal display of VGA text and graphics modes on 800x600, 1024x768 and 1280x1024 panels.

6.2.5 Television NTSC/PAL Flicker Free Output

The 69000 uses a flicker reduction process which makes text of all fonts and sizes readable by reducing the flicker and jumping lines on the display. The 69000 uses a line buffer and digital filters to average adjacent vertical lines for odd/even display. The chip also uses both horizontal and vertical interpolation to make both graphics and text appear smooth on a standard television. This process reduces the effect of flicker on NTSC displays.

6.2.5.1 Design Note

The Chips and Technologies 69000 HiQVideo Accelerator provides VGA and LCD output on the board. Using the Analog Devices AD725 RGB to NTSC/Pal Encoder, the board also provides TV-out capability.

The 69000 is PCI device number 2 (IDSEL connected to AD13 through a 220 Ω resistor) and uses PCI interrupt INTA#.

On the board a RGB monitor is always populated. Therefore when a TV Encoder like the AD725 is added, the RGB signals must be buffered. This is accomplished by a AD8073 video amplifier with high input impedances on each color signal. They are configured for a gain of two, which is normalized by the divide by two termination scheme used for the RGB monitor. However, since the RGB signals shift to ground during horizontal sync period, the AD8073 require -5 V supply.

The board provides an interface to LCD screens with up to 36-bit color. 8-bit monochrome to 24-bit color LCD panels are supported by Panel Connector 1. 36-bit LCD panels are supported using both Panel Connector 1 and Panel Connector 2.

All Panel interface signals from the graphics controller are 3.3 V. Level-shifting buffers should be added between the outputs from the graphics controller and the inputs of a 5.0 V flat panel if the 69000's VOL and VOH do not meet the requirements for the panel's VIL and VIH. The signals involved include pins P0-P35, SHFCLK, LP, FLM and M. The included schematics do NOT include these level shifters. The design assumes 3.3 V panels.

To ensure full compatibility with older CRT displays, 3.3 V to 5.0 V level-shifting buffers are added between the 69000's HSYNC and VSYNC outputs and the display's HSYNC and VSYNC inputs.

Power and ground signals to the 69000 are as specified in the *CHIPS 69000 HiQVideo Accelerator with Integrated Memory* datasheet. Refer to Table 2 for the URL. Table 4 lists all power and ground requirements, and their implementation on the board.

Table 4. Power and Ground Requirements

Pin Name	Description	Requirement	Implementation
DACVCC	Analog power for the internal RAMDAC	DACVCC should be isolated from all other VCCs and should not be greater than CORVCC	Isolated from all other VCCs through ferrite beads with filtering caps
DACGND	Analog ground for the internal RAMDAC	DACGND should be common with digital ground but must be tightly coupled to DACVCC	Tied to analog GND
MCKVCC	Analog power for the internal memory clock synthesizer (MCLK)	MCKVCC must be at the same voltage level as CORVCC	Coupled to MCKGND through RC network
MCKGND	Analog ground for the internal memory clock synthesizer (MCLK)	MCKVCC/MCKGND pair must be INDIVIDUALLY decoupled	Tied to digital GND through ferrite bead
DCKVCC	Analog power pins for the internal dot clock synthesizer (DCLK)	DCKVCC must be at the same voltage level as CORVCC	Coupled to DCKGND through RC network
DCKGND	Analog ground pins for the internal dot clock synthesizer (DCLK)	DCKVCC/DCKGND pair must be INDIVIDUALLY decoupled	Tied to digital GND through ferrite bead
GND	Digital ground		Tied to digital GND
CORVCC	Digital power for the graphics controller internal logic (a.k.a. the "core" VCC)		
MEMGND	Embedded memory ground		Tied to 3.3 V plane with filtering caps
RGND	Internal reference GND	Should be tied to GND	Tied to digital GND
IOVCC	I/O power		Tied to 3.3 V plane with filtering caps
MEMVCC	Power for embedded memory		Tied to 3.3 V plane with filtering caps

Proper layout of the 69000 is important to getting optimal performance.

The power plane that is attached to the core power supply should be as wide as practical for high current carrying capacity, and low inductance.

Decoupling capacitors should ideally be placed as close as possible to the 69000. This means that the best decoupling will occur if the capacitors are placed directly underneath the component. If a single sided board is required and capacitors cannot be placed underneath the component then decoupling is recommended at the corners of the 69000. Placing the capacitors at the corners minimizes decoupling trace lengths from the BGA package, reducing EMI.

Avoid placing audio components near a switching power supply.

Digital signals should be isolated from analog circuitry as much as possible to keep digital and analog currents from crossing. Ground currents from digital signals are noisy and should be isolated from the audio signals. Do not run dynamic digital signals such as clock, address or data lines in or close to the analog area.

All analog circuitry should be placed as close to I/O connectors as possible and should be isolated to as small an area as possible. Analog components and circuitry should reside over a separate ground and power plane with a gap between digital and analog planes made as wide as possible (25–50 mils is the acceptable minimum, 100 mils is ideal). The two plane pairs should be separated by a 2–3 mm gap. This means using at least a four layer board with ground and power planes forming an internal high capacitive, sandwich. This creates an effective series resistance (ESR) and effective series inductance (ESL) bypass capacitor.

Analog signals should reside over, or be referenced to, the analog ground plane as much as possible. Vias should be kept to a minimum. All analog signal traces should be as wide as possible for lower impedance.

Internal power and ground planes should be solid with no traces routed on them. The analog power plane containing the voltage regulator should coincide with the analog ground plane as much as possible. Analog ground shielding should be used on the external layers, especially near a low level input (i.e., microphone).

The analog and digital planes should be connected at one point only through a 0 Ω resistor or a ferrite bead. This maintains the proper reference potential for each ground plane. On the board this should be placed as close as possible to the 69000. This allows analog and digital ground return currents from the 69000 to flow through the analog and digital ground plane respectively.

There should not be any digital or analog traces crossing the gap between the analog and digital planes.

IC leads should have pads and vias that go directly to the appropriate plane for power and ground.

A separate small digital partition should be used for the crystal oscillator. This partition serves to keep noise from coupling onto the analog signals.

For information on how to interface to LCD Panels for the 65550, 65554, 65555 and 69000 HiQVideo accelerator series, refer to their respective application notes. See Table 2 for the URL.

A jumper on VEESAFE has been added to allow VEESAFE to be driven by a scaled version of VDDSAFE (scaled by a 1 K Ω potentiometer) for newer panels that have a low voltage requirement for VEESAFE. It is important that the potentiometer be fed from VDDSAFE rather than some other voltage source so that the resulting VEESAFE will meet the same power sequencing requirements as the panel. The sequencing of VEESAFE will be the same for VDDSAFE. The current requirement for a low-voltage VEESAFE is intended to be 1 mA maximum.

- Analog Devices AD725 RGB to NTSC/PAL Encoder with Luma Trap Port
 - The Analog Devices AD725 is a RGB to NTSC/PAL Encoder which may be populated on the board to add TV OUT capability.
 - RIN, BIN, and GIN are analog inputs that should be terminated by 75 Ω resistors to ground in close proximity to the IC.
 - The AD725 is a mixed signal part. It has separate pins for analog and digital +5 V and ground power supplies. Both the analog and digital ground pins should be tied to the ground plane by a short, low inductance path. Each power pin should have a bypass capacitor of 0.1 μ F and a larger tantalum capacitor of 10 μ F.
 - The outputs have a DC bias that must be AC-coupled for proper operation. This is done on the board by placing a 220 μ F tantalum capacitor and 75 Ω resistor in a series on each output.

- On the board, a RGB monitor is always populated. Therefore, when a TV encoder like the AD725 is added, the RGB signals must be buffered. This is accomplished by an AD8073 video amplifier with high input impedances on each RGB signal. The signals are configured for a gain of two, which is normalized by the divide by two termination scheme used for the RGB monitor. However, since the RGB signals go to ground during horizontal synchronization, the AD8073 requires a -5 V supply.
- A jumper is placed on pin 5 of the AD725. Placing a jumper on pins 1–2 enables the AD725. A jumper on pins 2–3 disables the AD725.
- A jumper is also placed on pins 1 and 12. Placing a jumper on pins 1–2 enables NTSC, placing a jumper on 2–3 enables PAL. It is important that the oscillator circuit, placed on pin 3 of the AD725, also be changed when switching from NTSC to PAL (a 14.318180 MHz crystal is already placed for NTSC) the crystal must be changed from 14.318180 MHz to 17.734475 MHz for PAL.
- Proper layout of the AD725 circuitry is required for optimal performance.
- All passive components should be placed as close as possible to the AD725.
- A “fence” should be formed around the analog signals. *No* digital signals should be routed under or above the analog power and ground planes.
- The filter circuits for the video output signals (CMPS, LUMA and CRMA) should be placed as close as possible to the connector. Long lengths of closely-spaced parallel analog signals should be avoided. Wherever analog signals run in parallel, separated by less than 15 mils for longer than 250 mils, run a ground line between the video input traces of approximately 12 mils in width.
- The three analog inputs (RIN, GIN, BIN) should be terminated by a 75Ω to ground close to the respective pins.
- Layout guidelines should be followed as in the HiQVideo 69000.

6.3 Audio

The audio solution on the board is AC '97 Component Specification 1.0 compliant. It features an Ensoniq AudioPCI 97 ES1371 Digital Controller, and an Analog Devices AC '97 SoundPort Codec AD1819 connected through the 5-wire AC '97 link.

The AC '97 audio architecture provides legacy support as well as a migration path to digital audio. Sound Blaster emulation is provided, as well as MIDI data interface. The AD1819 provides an analog front end for high performance audio, modem, and DSP applications.

6.3.1 AC '97 Audio

AC '97 defines a high quality two-chip audio architecture (AC '97 digital controller and AC '97 codec) advancing the migration to digital audio, while maintaining support for analog audio sources and analog interconnect for backwards compatibility. The two-chip audio solution comprising a digital audio controller, plus a high quality analog component that includes Digital-to-Analog Converters (DAC), Analog-to-Digital Converters (ADC) mixer and I/O. The architecture supports a wide range of high quality audio solutions, from a 2-channel mix of digital and analog audio, to multi-channel digital audio. The system is capable of achieving greater than 90 dB SNR performance.

6.3.2 Ensoniq AudioPCI 97 ES1371 Digital Controller

The Ensoniq AudioPCI 97 ES1371 Digital Controller supports simultaneous stereo audio, and host-based wave table music synthesis through the integration of an AC '97 2.0-compatible link and a PCI interface, in addition to audio legacy compatibility. The controller includes digital AC-link converters, WDM digital mixing acceleration, and variable sample rate conversion.

The PCI interface handles up to two digital audio streams, modem and handset streams. The digital audio streams are sample-rate converted and mixed to a common rate before being transmitted over the AC '97 2.2-compatible link to an AC '97 codec.

The sample rates are completely independent from the incoming and outgoing streams. The controller includes a variable sample rate converter that allows instantaneous support for sample rates ranging from 7 KHz to 48 KHz, with a resolution of 1 Hz.

SoundBlaster emulation is provided through the combination of hardware digital mixing, software SoundBlaster trapping, and host generated wave table music synthesis.

The primary interface for communicating MIDI data to and from the host is the hardware MPU-401 interface. The MPU-401 interface includes a built-in 64 byte FIFO for communication to the host bus.

6.3.2.1 Design Note

The ES1371 Digital Controller is PCI device number 16 (IDSEL connector to AD27 through a 220-Ω resistor). It is also PCI bus master #2 (connected to PCI bus master Pins REQ2# and GNT3#) and uses PCI interrupt INTB# on the board.

Note: The MIDI/JOYSTICK interface is not implemented on the board.

6.3.3 Analog Devices AC '97 AD1819 SoundPort Codec

The AD1819 SoundPort codec is designed to meet all requirements of the Audio Codec, Component Specification, v1.03. The AD1819 supports multiple codec configurations (up to three per AC link), a DSP serial mode, variable sample rates, modem sample rates and filtering, and built-in Phat Stereo 3D enhancement.

The AD1819 is an analog front end for high performance audio, modem, or DSP applications.

The main architectural features of the AD1819 are the high quality analog mixer section, two channels of sigma-delta ADC conversion, two channels of sigma-delta DAC conversion and Data Direct Scrambling (D2s) rate generators. The AD1819's left channel ADC and DAC are compatible for modem applications supporting irrational sample rates and modem filtering requirements.

6.3.3.1 Design Note

The AD1819 codec is connected to the ES1371 through the 5-wire AC '97 link. RESET# of the 5-wire AC '97 link is connected to PCI RESET (RST#) on the board.

The board implements microphone, video, line, and CDROM inputs as well as line, mono and line level outputs. The AD1819 provides interfaces for PC beep, phone input, and auxiliary inputs that are not implemented on the board.

There are various filter pins on the AD1819. These pins are listed below with their implementation on the board.

Table 5. AD1819 Codec Filter Pins and Implementations

Pin Name	TQFP	I/O	Description	Board Implementation
AFILT1	29	O	Antialiasing Filter Capacitor - ADC Right Channel	270 pF ceramic capacitor-to-analog ground
AFILT2	30	O	Antialiasing Filter Capacitor - ADC Left Channel	270 pF ceramic capacitor-to-analog ground
FILT_R	31	O	AC-Coupling Filter Capacitor - ADC Right Channel	1 μF tantalum capacitor-to-analog ground
FILT_L	32	O	AC-Coupling Filter Capacitor - ADC Left Channel	1 μF tantalum capacitor-to-analog ground
RX3D	33	O	3D Phat Stereo Enhancement - Resistor	47 nF capacitor-to-analog ground
CX3D	34	I	3d Phat Stereo Enhancement -	100 nF capacitor connected to RX3D

Note: Proper board layout of the AC '97 Audio components is important to getting optimal performance. Avoid placing the audio components near a switching power supply.

To keep digital and analog signal currents from crossing, digital signals should be isolated from analog circuitry as much as possible. Ground currents from digital signals are noisy and should be isolated from the audio signals. Do not run dynamic digital signals such as clock, address or data lines in or around the analog area.

All analog circuitry should be placed as close to I/O connectors as possible and should be isolated to as small an area as possible. Analog components and circuitry should reside over a separate ground and power plane with a gap made as wide as possible (25–50 mil is decent and 100 mil is ideal) between digital and analog planes. The two plane pairs should be separated by a 2–3 mm gap. This means using at least a four-layer board with ground and power planes forming an internal high capacitive sandwich. This gives an effective low ESR and ESL bypass capacitor.

Analog signals should reside over, or be referenced to, the analog ground plane as much as possible. Vias should be kept to a minimum. All analog signal traces, especially VREFOUT and analog power lines on the AD1819, should be as wide as possible for lower impedance connections.

Internal power and ground planes should be solid with no traces routed on them. The analog power plane containing the voltage regulator should coincide with the analog ground plane as much as possible. Analog ground shielding should be used on the external layers, especially near the microphone input.

The analog and digital planes should be connected at one point only, through a 0-Ω resistor or a ferrite bead. This maintains the proper reference potential for each ground plane. On the board, this should be placed as close as possible to the AD1819 and its voltage regulator. This allows the AD1819's analog and digital ground return currents to flow through the analog and digital ground plane respectively.

Note: There should not be any digital or analog traces crossing the gap between the analog and digital planes.

IC leads should have pads and vias that go directly to the appropriate plane for power and ground. A separate small digital partition should be used for the crystal oscillator. This partition serves to keep noise from coupling onto the analog signals.

6.4 SMSC FDC37B78x Ultra I/O

The Standard Microsystems Corporation (SMSC) FDC37B78x Ultra I/O provides keyboard, mouse, and floppy disk port and control on the board. It also provides two serial ports: COM1 is shared with the touch screen interface, and COM2 is shared with the infrared interface. The ultra I/O resides on the ISA bus and provides twelve IRQ options with full 16-bit address decode.

The FDC37B78x Ultra I/O provides support for the ISA Plug-and-Play Standard, v1.0a. Through internal configuration registers, each of the FDC37B78x's logical device's I/O address, DMA channel and IRQ channel may be programmed. There are 480 I/O address location options, 12 IRQ options or serial IRQ options, and four DMA channel options for each logical device.

6.4.1 Floppy Disk Controller (FDC)

The FDC provides the interface between a host microprocessor and the floppy disk drives. The FDC integrates the functions of the Formatter/Controller, Digital Data Separator, Write Precompensation, and Data Rate Selection logic for an IBM PC XT/AT-compatible FDC. The true CMOS 765B core is 100% compatible with the IBM PC XT/AT in addition to providing data overflow and underflow protection. The FDC is also compatible with the 82077AA using SMSC's proprietary floppy disk controller core.

6.4.2 Serial Port Controller

The FDC37B78x incorporates two full function UARTs. They are compatible to the NS16450, the 16450 ACE registers, and the NS16C550A. The UARTs perform the necessary serial-to-parallel conversion on received characters and the parallel-to-serial conversion on transmitted characters. The data rates are programmable from 460.8 Kbaud to 50 baud. The character options are programmable for 1 start; 1, 1.5 or 2 stop bits; even, odd, sticky or no parity; and prioritized interrupts. Each UART contains a programmable baud rate generator that is capable of dividing the input clock or crystal by any number in the range of 1 to 65535. The second UART supports IrDA 1.0, HP-SIR, ASK-IR, and Consumer IR infrared modes of operation.

6.4.3 Infrared Interface

Infrared support is included in the SMSC FDC37B78x Ultra I/O chip and includes Consumer IR and IrDA, v1.0 support. The interface provides a two-way wireless communications port. There are several IR implementations for the second UART in this chip (logical device 5), IrDA, Consumer Remote Control, and Amplitude Shift Keyed IR. The IR transmission can use the standard UART2 TXD2 and RXD2 pins, or optional IRTX and IRRX pins.

Serial communication baud rates up to 115.2 Kbps are provided for through IrDA, v1.0. The Amplitude Shift Keyed IR allows an asynchronous baud rate of up to 19.2 Kbaud.

6.4.3.1 Design Note

COM1 is shared with the touch screen. Three 3-pin jumpers are provided to switch from COM1 to touch screen and vice versa. Placing a jumper on pins 1–2 on all three jumpers enables the touch screen. Placing a jumper on pins 2–3 enables COM1.



COM2 is shared with the Infrared interface. IRTX2 and IRRX2 of the ultra I/O directly connect TXD and RXD of the TEMIC TFDU4100 Infrared Transceiver respectively. The functionality of this COM port must be switched in the BIOS. The board uses the FDC37B78x's alternate IRTX2 and IRRX2 pins, allowing BIOS to switch from COM2 to the Infrared interface.

6.4.4 Parallel Port Controller

The parallel port controller is compatible with the IBM PC XT/AT. It supports the optional PS/2-type bi-directional parallel port (SPP) mode, the Enhanced Parallel Port (EPP) mode, and the Extended Capabilities Port (ECP) mode.

6.4.5 Keyboard and Mouse

The universal keyboard controller uses an 8042 microcontroller processor core.

6.4.5.1 Design Note for Ultra I/O

Each power pin must be individually decoupled with a 0.1 µF capacitor. Since the real time clock on the ultra I/O is not used, VBAT must be tied to ground. If not grounded, this pin may float and confuse the internal circuitry that is trying to detect whether power is valid.

The ultra I/O provides a BIOS interface. This is not implemented on the board, therefore ROMCS# must be pulled high to disable the ROM buffers and avoid interference with the boot ROM.

The SYSOPT pin is latched on the falling edge of RESET_DRV or on V_{CC} Power On Reset to determine the configuration register's base address. The SYSOPT pin is used to select the CONFIG PORT's I/O address at power-up. Once powered up, the configuration power base address can be changed through configuration registers. See Table 6 for SYSOPT settings.

Table 6. SYSOPT Pin Settings

Port Name	SYSOPT = 0 (1 KΩ pull-down resistor)	SYSOPT = 1 (10 KΩ pull-up resistor) Board Setting
CONFIG PORT	0x03F0h	0x0370h
INDEX PORT	0x03F0h	0x0370h
DATA PORT	INDEX PORT + 1	INDEX PORT + 1

Note: If using TTL RS232 drivers, use a 1 KΩ pull-down resistor. If using CMOS RS232 drivers, use 10 KΩ pull-down resistor. The board uses TTL drivers, therefore SYSOPT should be pulled down with a 10 KΩ pull-up resistor. Therefore, the CONFIG PORT and INDEX PORT for the FDC37B78x is located at 0x0370h after Reset.

The setting on SYSOPT *must* be different then SYSOPT on the SMSC FDC37C669 Super I/O in order for the two chips to configure at different addresses. On the board, FDC37C669 SYSOPT is pulled low with a 1-KΩ resistor. Therefore the FDC37C669 is located at 0x0370h in memory after reset. The two options are given in Table 7.

Table 7. SYSOPT Setting Options

Option	FDC37B78x SYSOPT Pin	FDC37C669 SYSOPT Pin
A	SYSOPT = 1 (pulled HIGH with 10 KΩ resistor)	SYSOPT = 0 (pulled LOW with 1 KΩ resistor)
B	SYSOPT = 0 (pulled LOW with 1 KΩ resistor)	SYSOPT = 1 (pulled HIGH with 10 KΩ resistor)

6.4.6 Additional Serial Ports with SMSC 37C669 Super I/O

An additional Super I/O, SMSC 37C669 is used on the board to add two extra serial ports COM3 and COM4. The parallel, IDE, and floppy interfaces are not used on this design. This configuration was chosen over separate 16550's because of cost and board space considerations.

Note: Each power pin must be individually decoupled with 0.1 μF capacitors.

All inputs to unused devices (i.e., parallel, IDE and floppy) are pulled to their inactive state. The value of the pull-up/pull-down resistor is chosen to keep the inputs within 0.5 V of the supply rails in order to minimize current consumption. Each pin has 10 μA of current leakage.

The FDC37C669 provides 11 decoded address lines, AEN, plus one chip-select input. Since the FDC37C669 can only decode the lower 11 address bits and AEN, an OR-gate is used to decode addresses A11–A15. The output of the OR-gate is low when A11–A15 are low, therefore asserting the FDC37C669 chip select. This chip select, AEN, and 11-bit address decode provides the full 16-bit ISA decode necessary.

6.4.6.1 Design Note

At the trailing edge of a hardware reset, the SYSOPT input is latched to determine the configuration base address. Refer to Table 8 for SYSOPT settings.

Table 8. SYSOPT Settings

Port Name	SYSOPT = 0 (1 KΩ pull-down resistor)	SYSOPT = 1 (10 KΩ pull-up resistor)
INDEX Base I/O Address	0x03F0	0x0370

Note: If you are using TTL RS232 drivers, use a 1 KΩ pull-down resistor. If using CMOS RS232 drivers, use a 10 KΩ pull-down resistor. The board uses TTL drivers, therefore SYSOPT should be pulled

down with a 1 KΩ resistor or pulled up with a 10 KΩ resistor. SYSOPT is pulled low on the board, therefore the FDC37C669 is located at 0x03F0h in memory after Reset.

The setting on SYSOPT *must* be different then SYSOPT on the FDC37B78x Ultra I/O in order for the two chips to configure at different addresses. The two options are listed in Table 9.

Table 9. SYSOPT Setting Options

Option	FDC37B78x SYSOPT Pin	FDC37C669 SYSOPT Pin
A	SYSOPT = 1 (pulled HIGH with 10 KΩ resistor)	SYSOPT = 0 (pulled LOW with 1 KΩ resistor)
B	SYSOPT = 0 (pulled LOW with 1 KΩ resistor)	SYSOPT = 1 (pulled HIGH with 10 KΩ resistor)

NOTE:

1. On the board, SYSOPT on the FDC37B78x is pulled HIGH and SYSOPT on the FDC37C669 is pulled LOW.

6.5 Tritech Microelectronics TR88L803 Touch Screen Controller

TriTech Microelectronics TR88L03 Touch Screen Controller is used to implement a touch screen solution. This is a fully integrated pen input processor with two multiplexed A/D input channels. The chip uses a serial interface, and there is no user configuration required.

The low-power TR88L803 contains all the circuitry required to interface the low cost 4- and 5-wire resistive digitizers to applications and provide pen-input capability. The TR88L803 uses 10-bit ADCs to resolve 1024 levels. When clocked at 4 MHz, the TR88L803 is designed to simplify pen interfacing by integrating all pen-input tasks required to present X, Y position data to the main application at 200 coordinate pairs per second. The TR88L803 provides standard asynchronous serial output or synchronous serial output. The TR88L803, with enhanced noise filtering, is ideally suited for operation in electrically noisy environments.

6.5.1 Design Note

The touch screen controller shares resources with COM1 on the board. To use either COM1 or the touch screen, the board must be jumpered correctly.

The TR88L803 is configured to use power from the serial port COM1. The Touch Screen circuit draws less than 10 mA from the serial port during Power On conditions. During IDLE/SLEEP mode, it draws less than 5 mA. A 3.3 V voltage regulator across the serial port Data Terminal Ready (DTR) and Ground (GND) pins generate a stable power source (digital VDD and analog VDD) from the 5 V to 12 V source found on DTR.

The TR88L803's serial data output is connected to the serial port TX pin through two general-purpose transistors. The transistors are properly biased to provide the necessary signal level swing for the TX pin. The negative signal level is derived with the RX pin of the serial port. RX is not used since data is unidirectional for the touch screen controller.

The touch screen controller provides two independent ADC input channels under the ADC multiplex mode. Control pin (MUX_SEL) multiplexes the internal ADC between serving the digitizer inputs (X+, X-, Y+, Y-) and the two independent ADC input channels. These channels are not implemented on the board, therefore ADC_1 and ADC_2 are tied to the analog ground through a 10-KΩ resistor and MUX_SEL is tied to ground.

The touch screen controller provides a 4-wire interface for the Dynapro touch screen.

Mixed signal layout guidelines should be followed as discussed in the LCD/Video and Audio sections.

- **LCD w/ Integrated Touch Screen:** The touch screen controller provides a four-wire interface for the Dynapro touch screen. The LCD interface consists of two connectors: Flat Panel Connector 1 and Flat Panel Connector 2. The Flat Panel Connector 1 provides an interface to LCD screens with up to 24-bit color. For 36-bit color, both Flat Panel Connector 1 and Flat Panel Connector 2 must be used.

6.6 PCMCIA (Strataflash & PC Card) Socket

This reference design uses the Texas Instruments PCI1221 PC Card Controller, supporting both an external PCMCIA card socket as well as on-board Intel® StrataFlash™ memory. The controller operates on a 3.3 V core voltage and is PCI interface compatible with 3.3 V and 5.0 V PCI signaling environments. Supporting up to five general purpose I/Os, the controller is compliant with both the PCI Local Bus Specification, v2.1 and 1995 PC Card Standards. The Fujitsu Takamisawa Americas 565P068-G/J-4V socket is compatible with PCMCIA card type I, II, and III.

6.6.1 Texas Instruments PCI1221 PC Card (PCMCIA) Controller

The Texas Instruments PCI1221 is a PCI-to-PC card controller that supports two independent PC card sockets, compliant with the 1995 PC Card Standards. The 1995 PC Card Standards retain the 16-bit PC card specification defined in PCMCIA, v2.1, and defines the new 32-bit PC card, CardBus, capable of full 32-bit data transfers at 33 MHz. The PCI1221 supports any combination of 16-bit and CardBus PC cards in the two sockets and, as required, are powered at 5 V or 3.3 V.

The PCI1221 is compliant with the PCI Local Bus Specification, v2.1 and with the PCI Bus Power Management Interface Specification. The PCI1221 can act as either a PCI master or slave device with PCI bus mastering initiated during CardBus PC card bridging transactions.

6.6.1.1 Slot A: PCMCIA Socket with Texas Instruments TPS2206 PC Card Power-Interface Switch with Reset

Slot A of the PCI1221 is connected to a PCMCIA socket for PCMCIA cards on the board. The Texas Instruments TPS2206 provides voltage regulation, overcurrent and over-temperature protection for the PCMCIA socket. The TPS2206 also accommodates 3.3 V/5 V systems by first powering up the PCMCIA card with 5 V, then polling it to determine its 3.3 V compatibility.

As well as the above, the TPS2206 also provides a reset to the PCMCIA card. This reset is triggered by a PCI RESET (RST#) on the board.

6.6.1.2 Slot B: Intel StrataFlash Memory

Capitalizing on two-bit-per-cell technology, Intel StrataFlash memory products provide 2X the bits in 1X the space. Offered in 64-Mbit (8-Mbyte) and 32-Mbit (4-Mbyte) densities, Intel StrataFlash memory devices are the first to bring reliable, two-bit-per-cell storage technology to the flash market.

Intel StrataFlash memory benefits include, more density in less space, lowest cost-per-bit NOR devices, support for code and data storage, and easy migration to future devices.



All card signals are internally buffered to allow hot insertion and removal without external buffering. The PCI1221 is register compatible with the Intel 82365SL-DF ExCA controller. The PCI1221 internal data path logic allows the host to access 8-, 16-, and 32-bit cards using full 32-bit PCI cycles for maximum performance. Independent buffering and a pipeline architecture provide high performance with sustained bursting. The PCI1221 can also be programmed to accept fast posted writes to improve system bus utilization.

General purpose inputs and outputs are provided to implement sideband functions. Many other features are designed into the PCI1221, such as socket activity LEDs.

A CMOS process is used to achieve low system power consumption while operating at PCI clock rates up to 33 MHz. Power consumption is further reduced by several low-power modes.

6.6.1.3 Design Note

The Texas Instruments PCI1221 is a PCI-to-PC card controller. The controller provides two PC card slots, Slot A is connected to a PCMCIA socket for PCMCIA cards on the board. Slot B of the PC card controller provides the interface to Intel StrataFlash memory.

The PCI1221 PC card controller is PCI bus master #3 (connected to PCI bus master pins REQ3# and GNT3#) on the board. It is also PCI device number 5 (IDSEL connected to AD16).

The PCI1221 provides the user with flexibility in determining the interrupt implementation. The interrupt mode is selected via bits [2:1] of the Device Control Register at PCI offset 92h. Other registers that must also be configured are described below. The interrupt implementations are listed in Table 10.

The board implements parallel IRQ and PCI interrupts. In this interrupt mode, the PCI1221 routes the legacy interrupts, IRQ[15:2], via the seven multifunction terminals MFUNC[6:0]. The parallel IRQ and PCI interrupt modes are selected by programming bits [2:1] of the Device Control Register at PCI offset 92h to a value of 01b. When this mode is selected, the multifunction terminals must be configured through the Multifunction Routing Register at PCI offset 8Ch.

Table 10. Interrupt Implementations

Interrupt Mode	PCI Offset 92h Bits 2:1
Parallel PCI Interrupts Only	00
Parallel IRQ and PCI Interrupts	01
Serial IRQ and Parallel PCI Interrupts	10
Serial IRQ and PCI Interrupts	11

The PCI interrupts INTA# and INTB#, are available only on terminals MFUNC0 and MFUNC1 respectively. A maximum of five IRQ interrupts may be implemented through multifunction terminals MFUNC[6:2].

The multifunction terminals are connected on the board as listed in Table 11.

Table 11. Multifunction (MFUNC[6:0]) Board Connections

Terminal Pin	Connection on Board
MFUNC0	INTC#
MFUNC1	INTD#
MFUNC2	IRQ4
MFUNC3	IRQ5
MFUNC4	IRQ9
MFUNC5	IRQ10
MFUNC6	IRQ15

6.7 ISA PCI Expansion Cards

There are two PCI expansion slots on this sample design, allowing for add-in card peripherals to enhance any design. There is also one ISA expansion slot in the design. With the variety of peripherals in this design that would ordinarily be implemented through an add-in card, two extra PCI slots and one extra ISA slot allows for functionality that has not been added.

6.7.1 Design Note

- ISA Bus/Expansion Slots

As discussed on the *Intel® 430TX PCIsset Design Guide* (order number 290613), pull-up and pull-down resistors should be placed as follows:

- 10 K Ω pull-ups on SD[15:0] and SA[19:0].
- 1 K Ω pull-up on IOCHRDY and REFRESH#.
- 10 K Ω pull-up on IRQx (10 K Ω pull-up on IRQ8, MEMR#, MEMW#, IOR#, IOW#, LA[23:17], SMEMR#, SMEMW#, SBHE#, and BALE).
- 330 Ω on ZEROWS#, MASTER#, MEMCS16#, and IOCS16#.
- 4.7 K Ω on IOCHK#.
- 5.6 K Ω pull-down resistors on DRQx.

- PCI Bus/Expansion Slots

Per the *Intel® 430TX PCIsset Design Guide*, place pull-up signal resistors on the following:

- 2.7 K Ω pull-up resistors to 5 V on PIRQ[D:A]#, SDONE, SBO#, FRAME#, TRDY#, STOP#, IRDY#, DEVSEL#, PLOCK#, PERR#, SERR#, REQ64# and ACK64# and PAR
- 10 K Ω pull-ups to 3.3 V on GNT[3:0]#, PHLD# and PHLDA#.

There are two PCI expansion slots on the board. Slot 0 is PCI device #17, Slot 1 is PCI device #18 (IDSEL connected to AD28 and AD29 through a 220 Ω resistor respectively). Each slot is connected to PCI bus master arbitration pins REQ# and GNT#. Slot 0 is connected to GNT0# and REQ0#, and Slot 1 is connected to GNT1# and REQ1#.

Note: For layout considerations when placing series resistors, refer to the *Intel® 430TX PCIsset Design Guide*.



Interrupts on the board are configured as indicated in Table 12 to minimize interrupt latency.

Table 12. Board Interrupt Configurations

Device Interrupt Pin	LCD / Video Controller	AC '97 Audio	PCMCIA†	PCI Slot 0	PCI Slot 1
0	INTA#	INTB#	INTC#	INTD#	INTA#
1	X	X	INTD#	INTA#	INTB#
2	X	X	X	INTB#	INTC#
3	X	X	X	INTC#	INTD#

† The PCMCIA controller also uses legacy ISA IRQs. Refer to the PCMCIA section on page 33 for more information.

Note: Refer to the PCI Local Bus Specification, v2.1, for routing guidelines (see Table 2).

6.8 Clocking

Improper layout of the clock lines on the board can cause cross-coupling with other signal lines. It is recommended that the clock line spacing (air gap) be at least two times the trace width of any surrounding traces. It is also strongly recommended that the clock spacing be at least four times the trace width of any strobe line.

6.8.1 Cypress Semiconductor CY2280

The CY2280 is a clock synthesizer/driver that outputs four processor clocks at 2.5 V. There are eight PCI clocks that run at one-half or one-third the processor clock frequency of 66.6 MHz and 100 MHz respectively. One of the PCI clocks is free-running. The CY2280 possesses power-down, processor stop, and PCI stop pins for power management control. The signals are synchronized on-chip and ensure glitch-free transitions on the outputs.

The CY2280 outputs are designed for low EMI emissions. For further information, please refer to the following URL: <http://www.cypress.com/cypress/prodgate/timi/cy2280.htm>.

6.8.2 Cypress Semiconductor CY2309

The CY2309 is a 3.3 V zero-delay buffer designed to distribute high speed clocks in PC systems and SDRAM modules. There are on-chip PLLs that lock to an input clock on the REF pin. The PLL feedback is on-chip and is obtained from the CLKOUT pad. All outputs have less than 200 ps of edge-to-edge jitter. The input-to-output propagation delay is guaranteed to be less than 350 ps, and the output-to-output skew is guaranteed to be less than 250 ps. For more information please refer to the following URL: <http://www.cypress.com/cypress/prodgate/timi/cy2309.html>.

6.8.3 Clocking for 430TX Chipset

The Intel 430TX chipset uses several techniques from the mobile and desktop markets to reduce power consumption and hence power dissipation in a Pentium processor-based embedded system.

In the MTXC there are three main states used in clocking.

1. Clock Stopped: CLKRUN# is being monitored for a restart, the clock is stopped.

2. Imminent Clock Stop: The system has indicated the clock is about to be halted utilising the CLKRUN# line.
3. Clock Running: The clock is running and the PCI system bus is operational.

The MTXC is a CLKRUN# master unit and behaves according to the protocols specified for a master control device. The PIIX4 companion device to the chipset controls system clocks and is the CLKRUN# central resource. In designing a system, the main system clock should originate from a master clock oscillator, and for the processor and MTXC, be from the same buffer driver output.

The signal paths must be setup to minimize clock skew and the resulting system instabilities. Trace length matching, along with proper terminations to minimize reflected energy, will result in better system performance and less overall noise that may possibly be affecting system reliability.

When the Intel 430TX chipset is used in a system application, the designer may elect to take advantage of the different clocking schemes and modes available to enhance system power and performance. There are five low power clocking modes available:

1. Chip standby: The MTXC processor and PCI busses are idle.
2. Dynamic stop clock: Assists the processor in transitioning in and out of clock stop from system run.
3. Powered On Suspend: All system phase locked loops (PLL) are shut down. The real time clock (RTC) and suspend clock (SUSCLK) are running. DRAM refresh is accomplished by using SUSCLK.
4. Suspend to Ram (STR): The processor and L2 cache are clocked off. RTC and SUSCLK are running. DRAM refreshes using SUSCLK
5. Suspend to Disk (STD): The processor and L2 cache, DRAM and PCI interface are clocked off.

Note: In clocking the main system the PCI bus clock is derived from the main processor clock and is one half the frequency of said clock.

6.8.4 Clocking for 440BX Chipset

As described in Section 6.8.3 above, except for SDRAM, the clock is synchronized with BXDCLKO and BXDCLKRD.

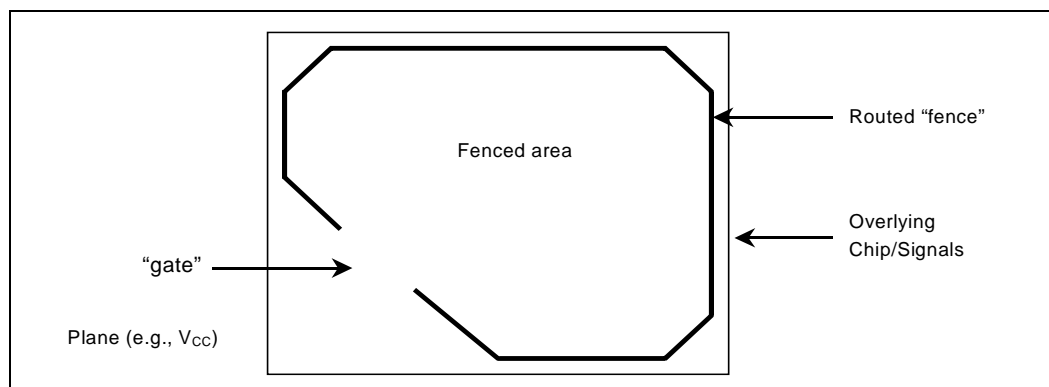
7.0 Design Considerations

There should be decoupling capacitors for every schematic page and one bulk capacitor for the entire design. This provides a short between power and ground for high frequency noise signals and reduces impedance.

- If a part is removed from the design, the outputs can be left unconnected, but the inputs should be pulled either high or low through an appropriate resistor.
- When changing jumper settings, first power down the board.
- Trace widths on all power and ground signals should be ~10 mil.

A fence is a line routed out of a plane such that a given area is isolated from the rest of the plane except at a single point of contact, conceptually the “gate” in the fence. A fence will minimize noise originating from the digital signaling onto the analog signals. This provides higher quality video or audio. An example is shown in Figure 1. The heavy black line is the routed area. The width of the gate or opening can be up to 75% of the length of the integrated circuit or signals in question. The width of the routed fence should conform to the separation routing between power planes (i.e., 25 mils minimum).

Figure 6. Fence Example



8.0 Windows CE

Windows CE can have the same look and feel as Microsoft Windows 98 or Windows NT. A variety of interfaces are supported for embedded applications. The hardware configuration for an embedded system can include the following peripherals: keyboard, mouse, touch screen, and so forth. It is often easier for users to deal with commands and keys that pertain to the desktop PC because they are familiar with them on their office or home desktop PC. A similar environment in the embedded product segment eliminates the need for retraining of personnel since they are already familiar with the look and feel of the system. Often times, it helps to be able to have little or no difference between the embedded and desktop device from a user interface perspective.

In Windows CE, the PC is used to prototype peripherals and develop device drivers; all development is done on the PC. If the target platform is also Intel Architecture, the large knowledge base for Intel Architecture can be utilized for hardware expertise, programmers, training materials, and sample code. Intel Architecture CISC is very efficient for all Windows operating systems. Since a lot of optimization effort has already been spent on the PC, reuse of PC developments can result in faster time-to-market. This is possible if the architecture of the Windows CE target is the same as the PC. The Windows CE Embedded Toolkit includes PC chipset and peripheral support, including basic VGA drivers and S3 accelerated drivers.

There are some differences between programming for the desktop and programming in Windows CE. Since Windows CE programs are required to have a small memory footprint, an important aspect of its programming is the need to manage with limited memory. Since memory usage must be minimized, having knowledge of memory allocations with Windows CE is important.

Embedded systems may stay powered-on for long periods of time. Programmers need to keep this in mind to ensure that the programs run flawlessly for many months at a time. Modifications must be made to the OEM Adaptation Layer. Since different solutions are possible for different applications, the varying need for applications can be met.

8.1 Windows CE for Embedded Applications

Windows CE is designed to be a multi-threaded, preemptive, multitasking operating system for platforms with limited resources. This is to say that multiple processes can be running simultaneously, each assigned a specific priority. Each process can have multiple threads. This allows a process to have more than one flow of execution running concurrently. Windows CE is designed as a modular operating system that can be customized to contain only the modules needed by the application. Refer to Table 2 for the URL to Microsoft's on-line Embedded Toolkit (ETK).

8.2 Windows CE Environment

8.2.1 Kernel

The Windows CE kernel contains the core operating system functionality that must be present on all Windows CE-based platforms. It includes support for memory management, process management, exception handling, multitasking and multithreading.

The kernel is provided as the file Nk.lib. The OEM Adaptation Layer must be developed to create the platform-specific Nk.exe module. For an overview of the OEM Adaptation Layer, refer to Section 8.2.3.

The kernel is designed specifically for small, fast, embedded devices. The kernel supports only a single 4 Gbyte address space (a 2 Gbyte virtual address and a 2 Gbyte physical address range). 1 Gbyte of the 2 Gbyte virtual address space is divided into 33 slots, each of which has a size of 32 Mbytes. The kernel protects each process by assigning each to a unique open slot in memory. Within each slot, memory regions are allocated for the following:

- 64 Kbytes of reserved space
- Executable files, code, and data sections
- Primary thread's stack, and the process's default heap
- Dynamic-link libraries (DLLs) loaded from RAM
- DLLs loaded from ROM

The kernel protects applications from accessing memory outside of their allocated slot by generating an exception. Applications can check for and handle such exceptions by using the "try and except" Windows CE functions. The system is limited to 32 processes, but the number of threads running in a process is limited only by the amount of available memory.

8.2.2 Modularity

Windows CE is built from a number of discrete modules, hence the size (footprint) of the operating system software can be controlled by selecting only the applicable modules. Several of these modules are also further divided into components. By selecting a minimum set of modules and components, the ROM and RAM requirements needed to support the end product can be minimized.

Refer to *Understanding Modularity in Microsoft Windows CE*. See to Table 2 for the URL.

8.2.3 OEM Adaptation Layer (OAL)

A developer can adapt Windows CE for a specific target platform by creating a thin layer of code that resides between the kernel and the target platform. To avoid confusion with the Windows NT Hardware Abstraction Layer (HAL), Windows CE refers to this interface as the OAL.

In addition to managing functions such as timing and power, the primary purpose of the OAL is to expose the target platform's hardware to the kernel. That is, each hardware interrupt request line (IRQ) is associated with one interrupt service routine (ISR). When interrupts are enabled and an interrupt occurs, the kernel calls the registered ISR for that interrupt. The ISR, the kernel mode portion of interrupt processing, is kept as short as possible. Its responsibility is primarily to direct the kernel to schedule and launch the appropriate interrupt service thread (IST). The IST, implemented in the device driver software module, gets or sends data and control codes to the hardware and acknowledges the device interrupt.

8.3 Windows CE Build Information

8.3.1 Getting Started

After installing all the necessary software (e.g., Visual C++, SDK, and ETK), get into the correct build environment for the project you are going to develop. Several are provided with Windows CE 2.10 (e.g., Maxall, Minshell). Windows CE 2.0 includes demos named DEMO1–DEMO7. After getting into the desired build environment, type “blddemo” to build the image for that project. This will place the image in the release directory. After building the image, load it into the target system using the Parallel Port Transfer Tool (PPSH) or the CE Shell Utility (CESH).

8.3.2 Building a Windows CE Operating System Configuration

The configurations included with the Windows CE Embedded Toolkit for Visual C++ contain all the modules and components needed to build different versions of a Windows CE operating system.

Included in the directory %_PUBLICROOT%\Common\Oak\Misc, a build batch file, called Blddemo.bat, automates the Build Tools required to generate the platform for your Windows CE operating system. For each configuration, a build batch file %_TGTPROJ%.bat, for example Minkern.bat, specifies the environment variables required for your Windows CE project. The following procedure describes how to configure your build environment and use Blddemo.bat to create the Windows CE operating system image for the operating system configuration.

To build a Windows CE operating system configuration:

1. If the configuration includes the Windows CE debug shell utility (CESH), invoke the Microsoft WinDbg debugging program shortcut for your platform (e.g., x86 Debugger).
2. Invoke the new command prompt build window shortcut that you created in Preparing to Build a Windows CE Operating System configuration.
3. As an optional step, modify your command prompt build window to add scroll bars. Scroll bars let you scroll back through the commands that the various batch files and tools execute. Click on the MS DOS icon in the top-left corner of the window, then click on Properties. Click on Layout and change the screen buffer size and height setting to 1000. This modification will give you a 1000-line screen buffer so you can see up to 1000 of the most recent lines in that command window. After clicking OK, select the Modify shortcut that started the window to make this change permanent.

4. Set any additional environment variables for the configuration build in your command prompt build window. For example, if you want to build the Windows CE kernel with remote debugging enabled, use the following command: `set IMGNODEBUGGER=`
To compile a configuration so that debugging messages are enabled, set the environment variable `WINCEDEBUG` to debug using the following command: `set WINCEDEBUG=debug`
For information on configuration-specific environment variables, see the corresponding description of the Windows CE operating system configuration in Section 8.3.2.
5. Enter the build batch file command to build the configuration: `blddemo`
After the build batch file is done processing, the current directory should still be the `%_WINCEROOT%` directory.
6. Verify that the build batch file command completed successfully by checking whether the `Nk.bin` file exists in the directory `%_FLATRELEASEDIR%` (by default, `\Wince210\Release`). `Nk.bin` contains the binary image of the operating system for your selected platform.

8.3.3 Building a New Project

This section provides a tutorial that describes how to build the Windows CE operating system image based on the Maxall configuration. In this section, you will:

- Create a new platform directory
- Create a new project directory
- Create a new command prompt build window
- Build the Windows CE operating system image for your new project

This tutorial assumes that you have installed the Embedded Development Kit in the default directory established by the setup program.

8.3.3.1 Create a New Platform Directory

To create a new platform directory, copy the `Wince210\Platform\Cepc` directory and all of its contents, including the subdirectories and their contents, into a new platform directory. The platform name should be no more than eight characters in length.

From a Command Prompt window, enter the following commands to copy the `\Cepc` directory from its default location and create `\MyPlat`, your new platform directory:

```
cd \Wince210\Platform
xcopy Cepc MyPlat /E /V /I
```

8.3.3.2 Create a New Project Directory

The Windows CE Embedded Toolkit for Visual C++ 5.0 (Embedded Toolkit) contains typical configurations of the Windows CE operating system. Use the Maxall configuration as the template for your MyProj build.

1. From a Command Prompt window, enter the following commands to copy the `\Maxall` directory (including all subdirectories and contents) into a new project directory named `\MyProj`. The project name should be no more than eight characters in length:

```
cd \Wince210\Public
xcopy Maxall MyProj /E /V /I
```

2. Rename the \MyProj\Maxall.bat file. The name of this batch file should always match the name of your project directory. Enter the following commands to rename the project batch file:

```
cd \Wince210\Public\MyProj
rename MAXALL.BAT MYPROJ.BAT
```

8.3.3.3 Create a New Command Prompt Build Window

The Embedded Toolkit includes a collection of shortcuts for the Minshell configuration. Each shortcut opens a command prompt build window specific for a supported processor. Use the x86 Minshell command prompt build window shortcut as a template for your command prompt build window shortcut. Your project and hardware development platform directories must first be created before you launch this shortcut.

1. In the Windows CE Embedded Development Kit program group, select the x86 Minshell shortcut.
2. Make a copy of the shortcut and rename it “x86 MyProj”.
3. In the tab shortcut for the icon's properties, modify the target command line parameters that follow the Wince.bat command. These modifications are case sensitive.

```
Wince.bat <cputype> <cpu> <os> <project> <platform>
```

For your build, the last part of the command line (starting with Wince.bat) must read as follows:

```
Wince.bat x86 I486 CE MYPROJ MYPLAT
```

4. Click OK to save the modified shortcut.

8.3.3.4 Build the Windows CE Operating System Image for a New Project

1. Invoke the x86 Build MyProj shortcut that you have previously created.
2. Enter the following command to build a complete version of Windows CE:

```
blddemo
```

After Blddemo.bat has finished processing, the current directory should still be the \Wince210 directory.

3. Be sure the build completed successfully by verifying that the Windows CE operating system image, called Nk.bin, exists in the \Wince210\Release directory. The build process sets this directory using the environment variable %_FLATRELEASEDIR%.

If Nk.bin does not exist, then the build of the Windows CE operating system image was not completed successfully. Before rebuilding, refer to Preparing Your Development Workstation for Subsequent Projects in the Windows CE Embedded Toolkit.

If Nk.bin exists, the Windows CE operating system image for MyProj was built successfully, and you can now run this image on your PC-based hardware development platform.

8.3.3.5 Adding Files or Applications to the Windows CE Operating System Image

1. Copy the file or application to the \Wince210\Public\<projectname>\Oak\Files directory.
2. Edit the project.bib file in the same directory. Add a reference to the end of the file to include the files/app formatted:

```
filename.ext $_FLATRELEASEDIR\filename.ext S
```

3. Build the image by calling the `blddemo` utility.

Note: The `filename.ext` should be replaced with the name of your file or application. The `S` signals the image builder to make the file a system file. Other switches are `H` (hidden) and `U` (uncompressed), and can be used in combinations (e.g., `SH` = hidden system file).

9.0 Windows CE Device Drivers

Device drivers are programs that provide an interface between the operating system software and platform hardware. In previous versions of Microsoft's operating system, these device drivers are implemented as special system executable files that run at the same privilege level as the operating system kernel. On Windows CE, device drivers are implemented as DLLs and run at the same privilege level as the user code.

9.1 Dynamic Link Libraries (DLLs)

DLLs are software function libraries just like their predecessor the static library. When a static library is linked to a program, the library functions used by the program are copied into the final executable image. This increases the size of the program. When a program is linked with a DLL, the library functions used by the program are *not* copied into the final executable image; only a function "stub" is copied. This function stub serves to identify the DLL and the function within the DLL that is being called upon. The actual DLL code is loaded into a separate memory space and the link between the program and the library function is created by the operating system when the function call is made. In addition, multiple programs can use DLLs simultaneously, reducing the memory requirements of each program.

9.2 Windows CE Device Driver Model

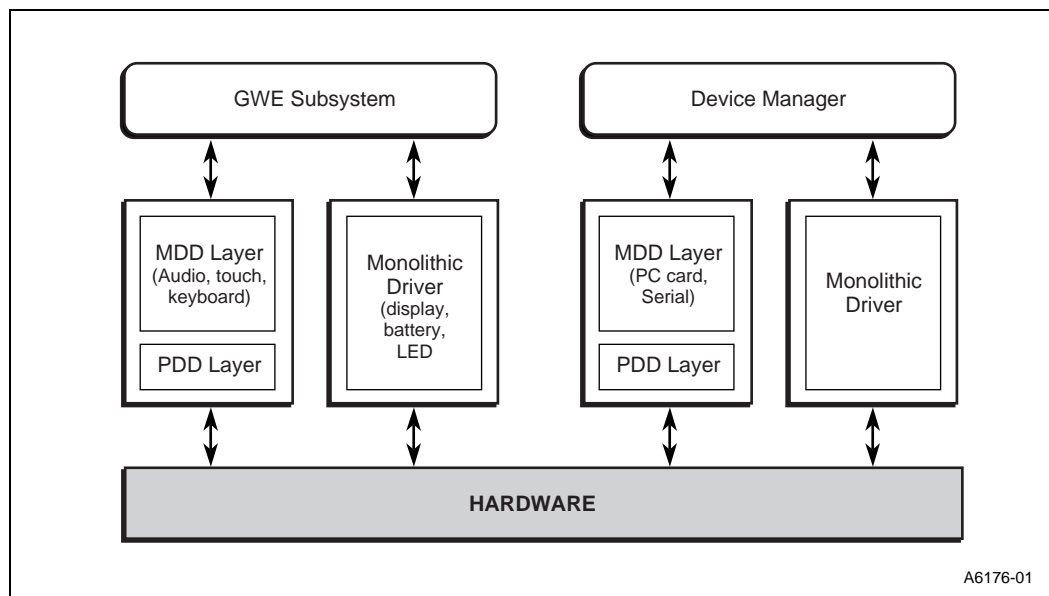
Windows CE has two types of device drivers as viewed from a software perspective: built-in and installable. Built-in drivers are typically for hardware that is native to the Windows CE target platform and are linked directly with the operating system image at build time. Some examples of this include keyboard, touch screen panel, battery, notification LED, PC card adapter, and audio hardware. Installable drivers are typically for hardware that can be added to the platform. This type of driver is loaded by the operating system and exists as a stand-alone DLL. Some examples of this include modems, printers, digital cameras and PCMCIA cards. In either case, the terms *built-in* and *installable* relate to whether or not a particular device driver is linked directly with the operating system image or loaded at run time as a stand-alone DLL.

For Windows CE drivers, Microsoft provides a layer of source code known as the Model Device Driver (MDD) component. The MDD component defines the interface to the operating system for a given type of device driver and is common to all platforms that use that type of device. The MDD also defines the interface to the platform-specific driver code. This platform-specific driver code is referred to as the Platform Device Driver (PDD) component. The PDD component implements the interface to the hardware device. These two layers are compiled, then linked together to form the complete driver DLL.

Whether or not a given type of device driver is built-in or installable is pre-determined by the MDD/OS interface defined by Microsoft for that type of device. For built-in drivers, Microsoft has defined a set of *custom* MDD/OS interfaces. For installable drivers, Microsoft has defined a set of *common* MDD/OS interfaces.

Note: Windows CE device drivers implementation is not required as MDD and PDD components. They may be implemented as a single, monolithic entity providing that the MDD/OS interface, as defined by Microsoft for a given device, is used. See Figure 2.

Figure 7. Device Driver Architecture



9.3 Interaction with Application Software

Installable device drivers are viewed as a special file in the file system by the application software. The interface exposed to the application corresponds to the standard Win32 file I/O functions such as OpenFile, ReadFile, DeviceIOControl, and others. The file system recognizes filenames as being special devices when they consist of exactly three uppercase letters, a single digit, and a colon (:). An example of this would be "COM1:", which references the first serial port available on the platform.

Built-in device drivers are accessed by application software using API calls that Microsoft has implemented in the Windows CE operating system for each device.

9.4 Incorporating Device Drivers

To include drivers in the Windows CE image, first take the compiled driver or file and place it in the following directory:

```
%_WINCEROOT%\public\maxall\files
```

Note: Maxall can be changed to the name of your project directory.

Edit the project.bib file in the same directory to include your file(s) and run blddemo.

A serial cable is needed. Windebug window must be open. All debug messages will be displayed on the windebug screen. The image must be downloaded to test the target board.

Refer to Microsoft's web site for information on new releases.

9.4.1 NE2000 ISA Network Interface Cards (NIC)

First, install the card on the embedded Pentium processor target board running Windows CE (WinCE 2.1 ETK). Along with plugging in the NIC, the following configurations must be made to ensure that the card is recognized.

The following files need to be modified prior to building a Windows CE image:

1. Registry entries in the wince.c file need to be modified.
Set the CEPC_NE2000_PCI variable to enable registry entries
2. To set up the registry correctly, modify the platform.reg file.
In WinCE210\platform\cepc\files\platform.reg
If PCI, change the bus type to 1
3. Find the following path and verify the keys:
Hkey_local_machine\comm\NE20001\Parms
4. Make sure that PCI enumeration is functional.
In Public\common\misc\wince.bat
Set CEPC_NE2000_PCI=1
Now the build will include the NE2000 driver
5. Build the driver.
Add NE2000 drivers to platform\cepc\drivers directory
Use build -c to build your image
For PCI devices, add PCI enumeration
In the wince210\public\common\oak\drivers\ceddk\test\pcienum
Use build -c to build your image
6. Debug
In the Wince.bat file, set the following environment variables to build in debug mode:
Wincedebug = debug
imgnodebugger =

9.4.1.1 NE2000 NIC Configuration

When the card is connected, run pcienum.exe. Grab the I/O address after changing wince.c Build again in the common area.

1. On the target board, press the Ctrl+Esc keys to run a program. If PCI, run pcienum.exe.
2. Run registry: NE20001
In the wince210\platform\CEPC\Files\platform.reg file make the following changes:

- a. Change Iobaseaddress = 00240
- b. Change Interruptnumber = 05
3. In the wince.c file, make the following changes. This should be in the NE2000 directory.
 - a. Change the interrupt number from 10 to 5
 - b. Change bustype to 1 if ISA
 - c. Change Iobaseaddress to 576
 - d. Type “build -c”
4. Before completing a build, delete the release directory and the maxall.bif file found in platform\CEPC\MAXALL.Bif.
 - a. Use blddemo to build the image
 - b. When the image is downloaded, use the ping application can be used to test the card.

9.4.2 ACTiSYS IR 2000B

To configure the ACTiSYS IR 2000B component:

1. Disable port 2 in the BIOS
2. Set the environment variable to use nscfir.dll instead of irsir.dll
3. Add the following to the Maxall.bat file or to the batch file that is used:
set IMGNSCFIR=1
4. To set up the registry correctly, modify the platform.reg file
5. Find the following path and verify the following keys:
[HKEY_LOCAL_MACHINE\Comm\NscIrda1\Parms]
“BoardType”=dword:0
“DongleType”=dword:4
“ConfigBase”=dword:398

Note: The above keys are important keys, however other keys should also be checked for compatibility with your system (e.g., Bus Number, and IRQ).

9.4.3 Audio

The card used in the sample design is the Creative Ensoniq AudioPCI card. Alternately this configuration is valid for equivalent parts on the board, namely the Ensoniq ES1371 and an AC '97 codec. The following steps are required to configure the digital controller.

1. The following driver files need to be included in the build:
wavepdd.cpp
wavepdd.h
ensoniq.h
2. Edit the following source files:
 - a. Makefile
 - b. Make a directory in %_WINCEROOT%\platform\cepc\drivers for the driver



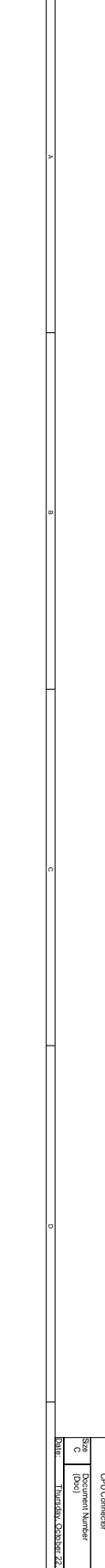
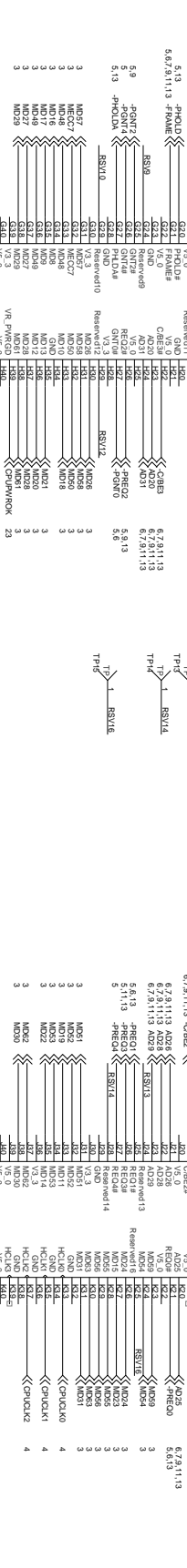
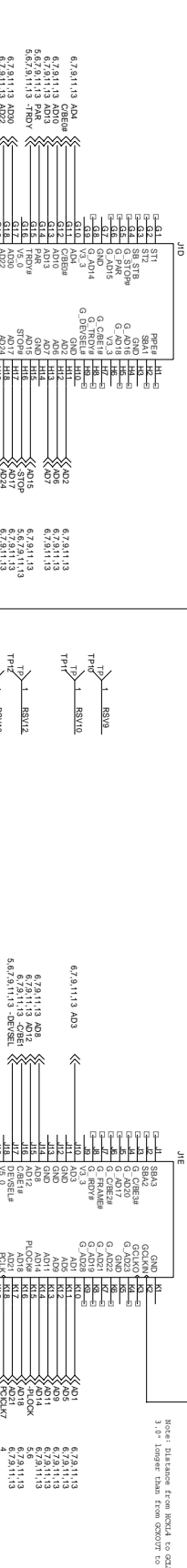
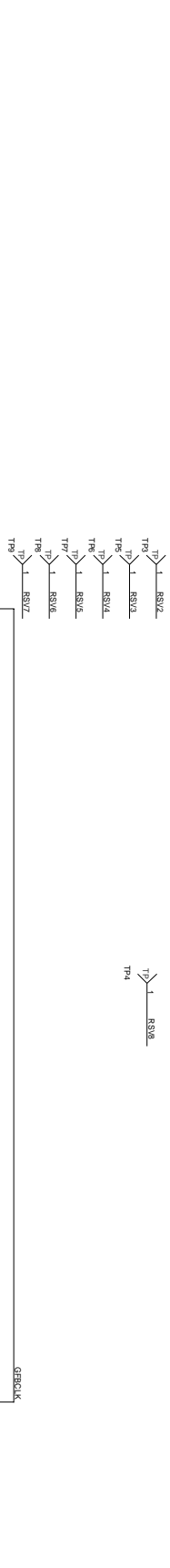
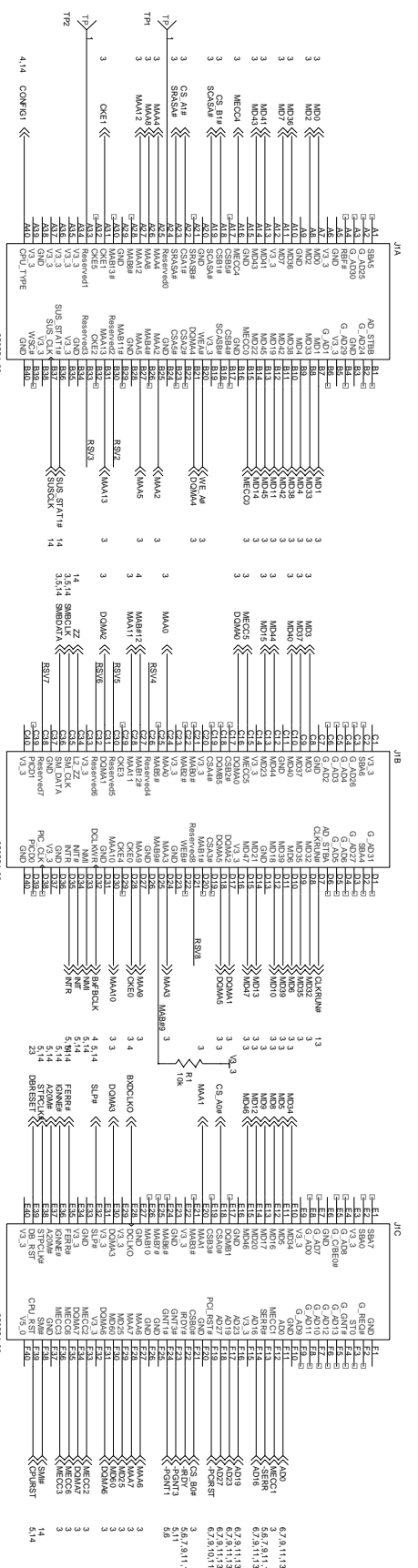
- Give it a name, we called it “ES1371WaveDev”
 - Copy files into directory
 - Edit file %_WINCEROOT%\platform\cepc\drivers\dirs to include this directory. Remove references to the WaveDev in “dirs” file
 - c. Run “blddemo” from %_WINCEROOT%
3. Set up configurations in VC++ to compile for Intel Architecture
- Load Application
- Check the Configuration list dialogue box for the desired configuration.
- a. Under “Build” menu, select “Configurations..”
 - b. Click on Add button
 - c. Pick a matching configuration from “Copy Settings from...”
 - For debug select “Win32 (WCE xxxx) Debug”
 - For release select “Win32 (WCE xxxx) Release”
(xxxx signifies don't care)
 - d. Pick a platform
 - Choose WCE x86 to compile for target
 - Choose WCE x86em to compile for emulator
 - e. Click on OK
 - f. Click on Close

Appendix A. Schematics

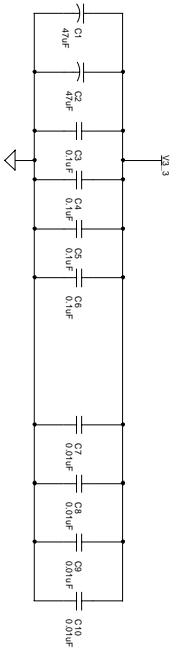
A.1 POS Terminal Baseboard Schematics

Schematics are provided for the following items:

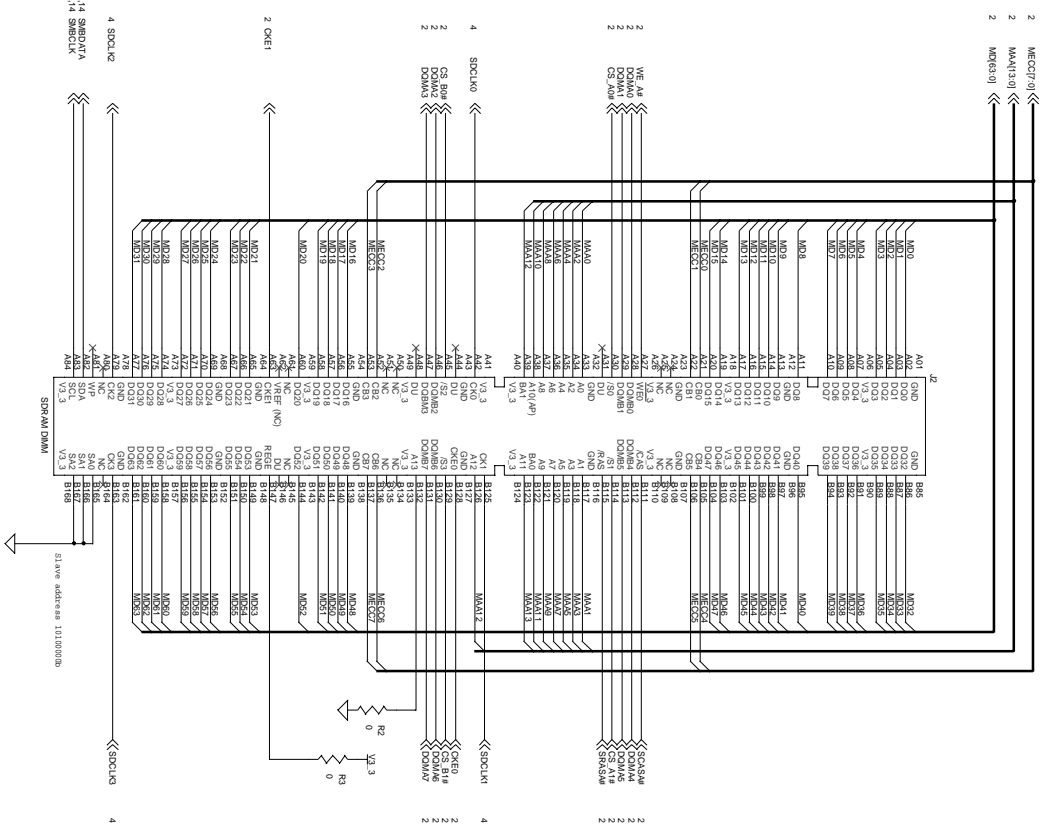
- Embedded Processor Assembly Connector
- SDRAM DIMM socket
- Clocks
- ISA/PCI pullups
- PCI slots 0 and 1
- 69000 part 1 and 2
- AC '97 Audio part 1 and 2
- PCI1221
- StrataFlash
- PIIX4 part 1 and 2
- IDE connector
- USB connectors
- Ultra I/O
- ISA connector
- COMx, DB25, floppy
- Touch screen controller
- Super I/O (extra serial ports)
- BIOS
- AXT power connector
- Unused gates



Note: Distances from J10A to contacts may be 3.2" longer than from CONTACT to J10 Connector

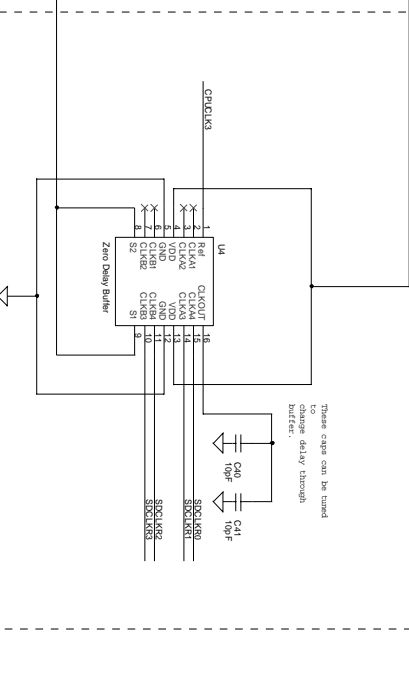
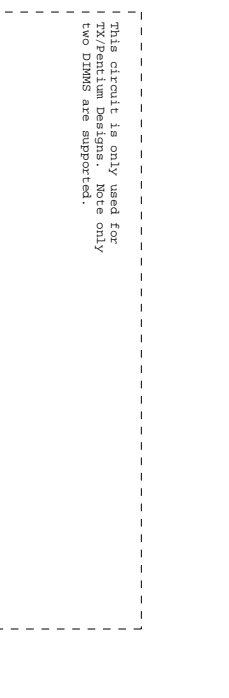
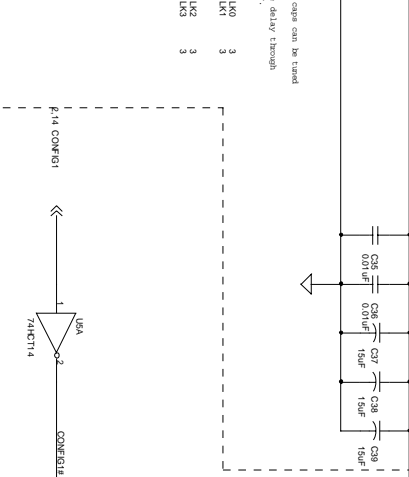
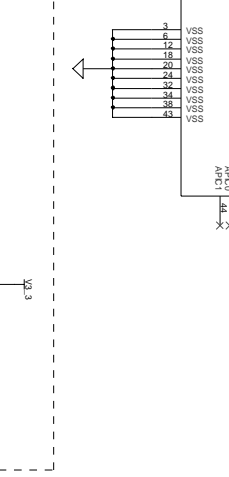
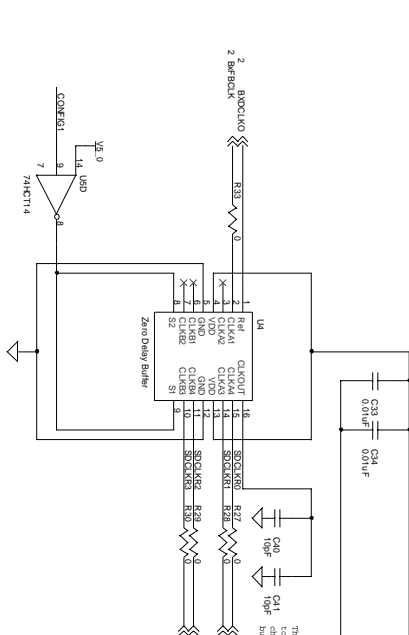
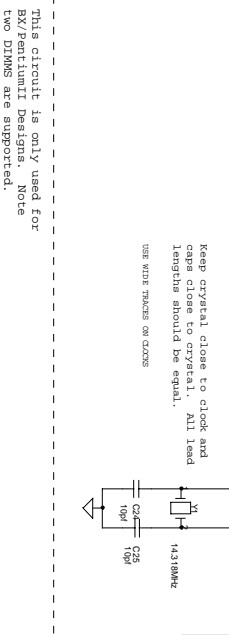
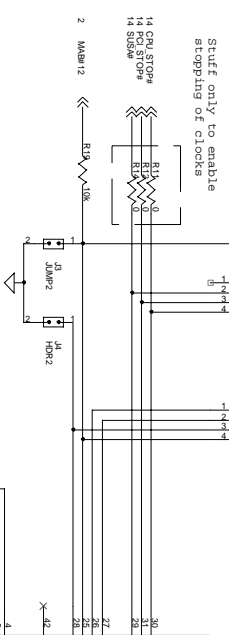
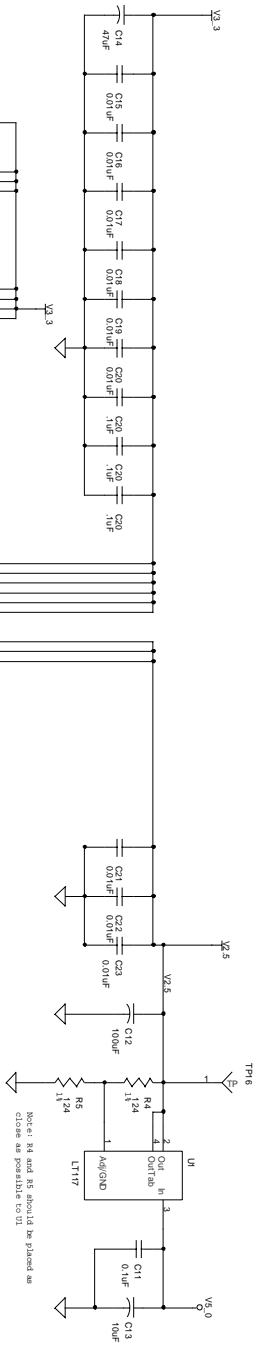


Socket 0



Slave address 1010000

File	DMM0
Size	Document Number
Doc	
Issue	Thursday, October 27, 1999
Sheet	3 of 24
Rev	A



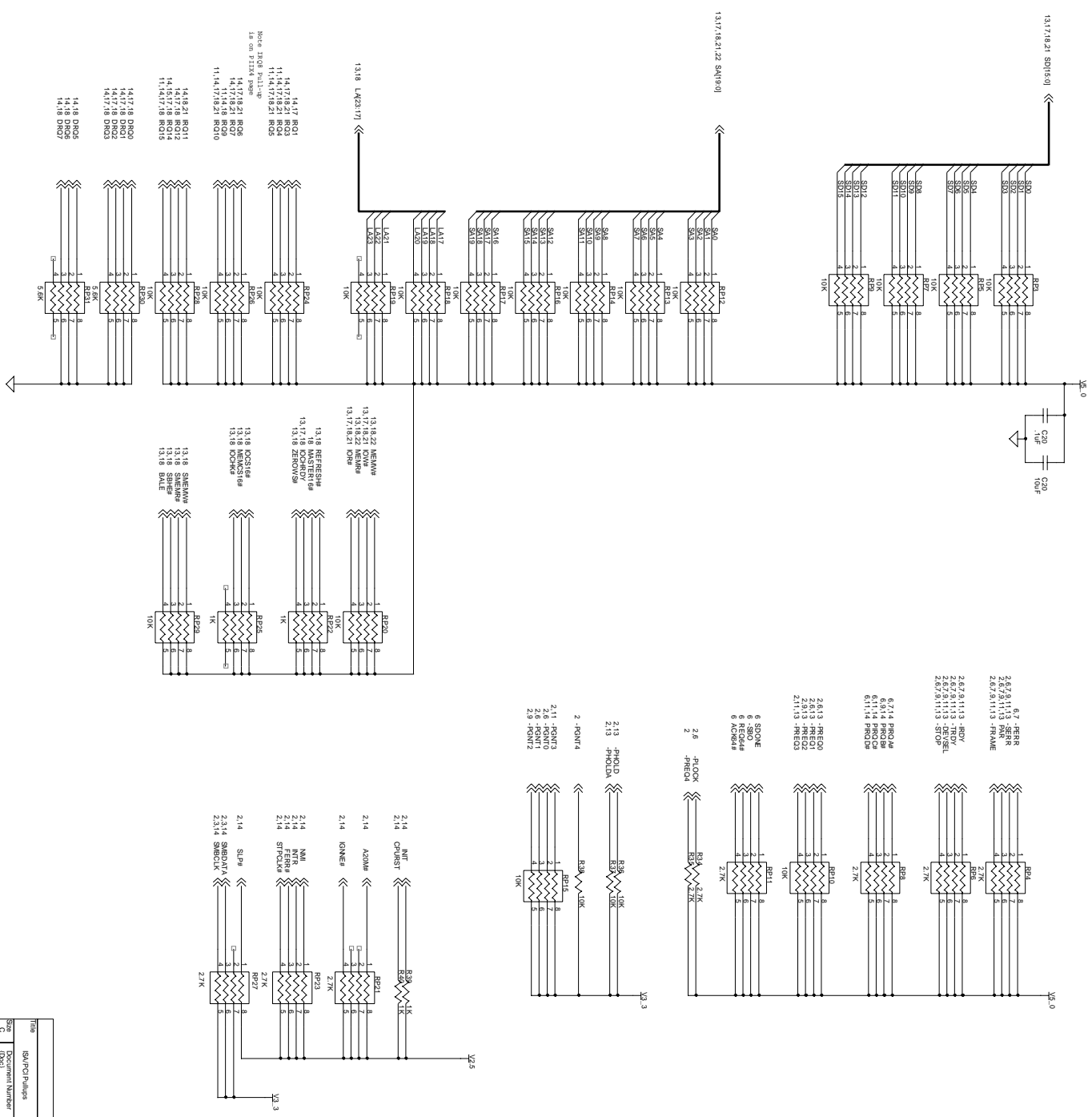
This circuit is only used for TX/Pentium Designs. Note only two DIMMs are supported.

This circuit is only used for TX/Pentium Designs. Note only two DIMMs are supported.

File	Size	Document Number	Rev
Clocks	C		A
Doc1			

ISA Pullups

PCI Pullups



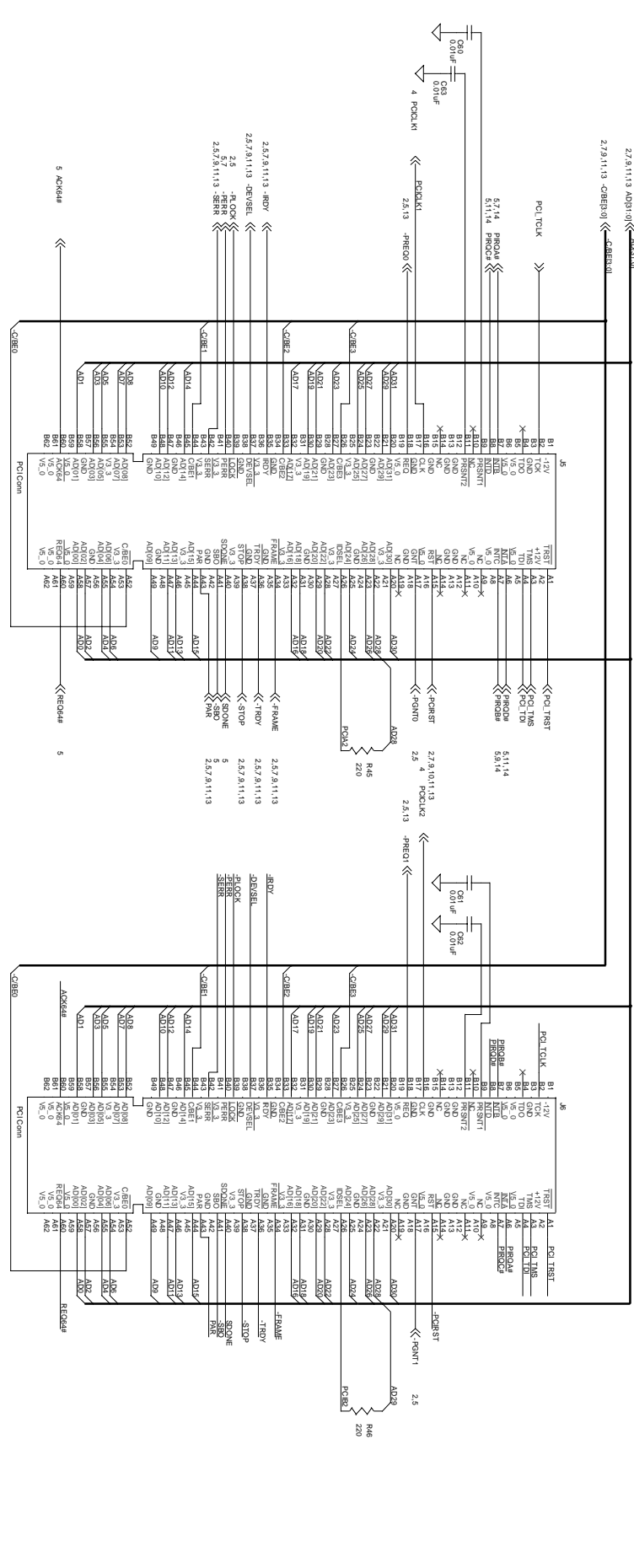
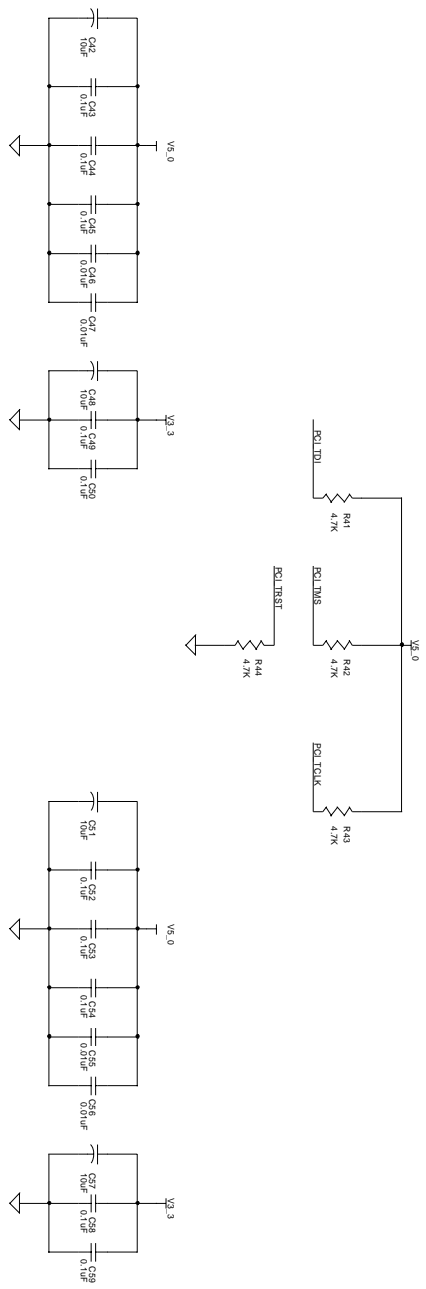
Title	ISA/PCI Pullups
Size	C
Doc#	
Issue	Thursday, October 27, 1994
Scale	5 of 21
Rev	A

J8/J9 V5_0:
 A5, A8, A10, A16, A59, A61, A62 | AL, A3, A4
 B5, B6, B19, B22, B59, B61, B62

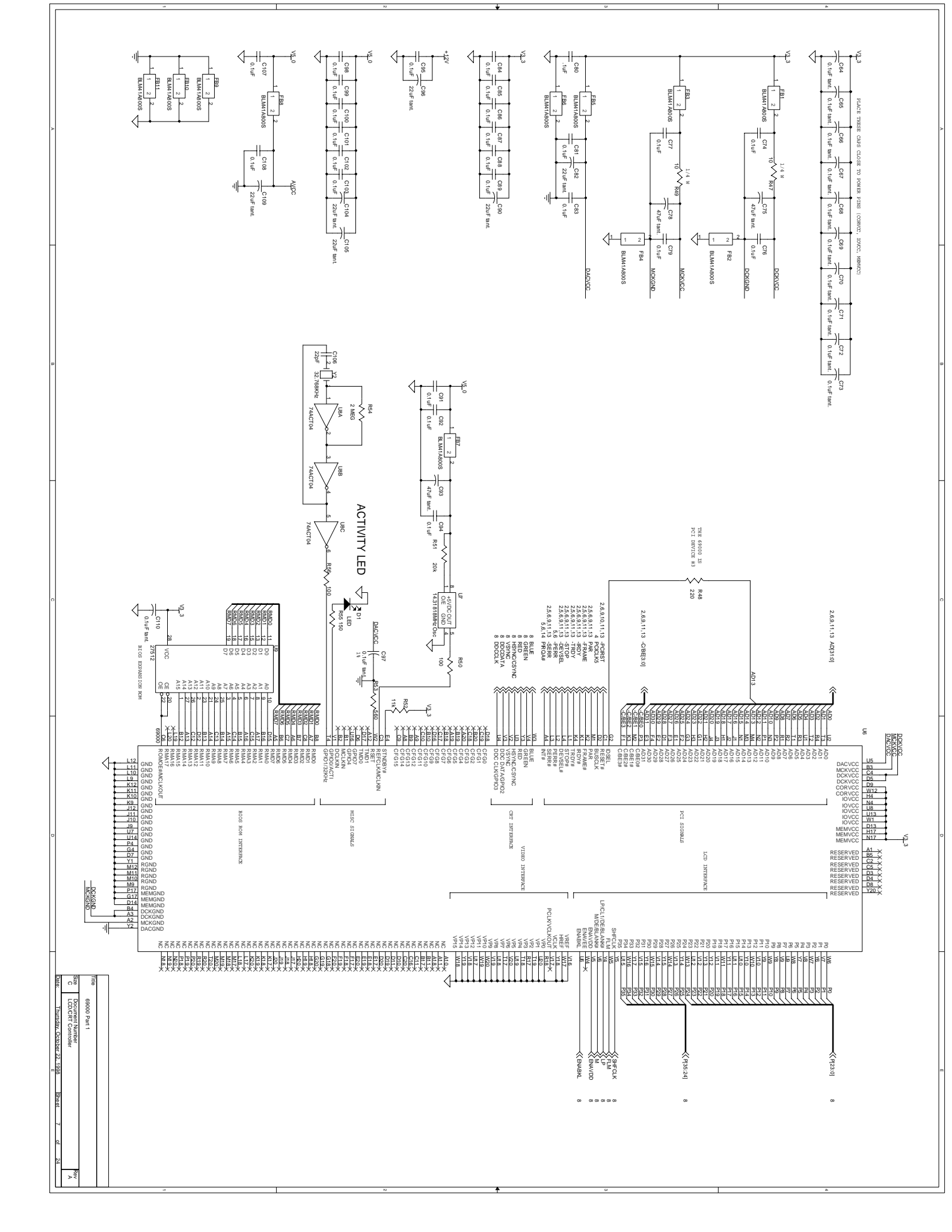
J8/J9 V3_3:
 A21, A27, A33, A39 A45, A53
 B25, B31, B36, B41, B43, B54

J8/J9 NC:
 A9, A11, A14, A19
 B10, B14

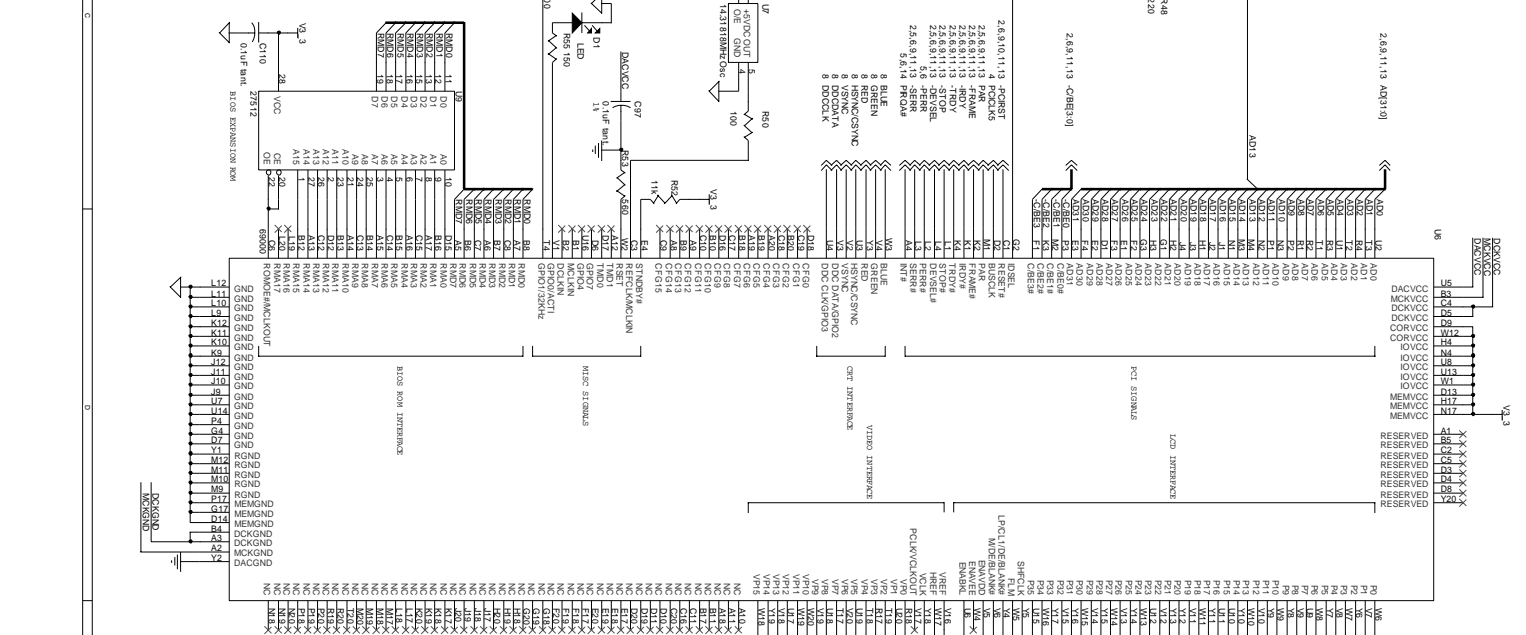
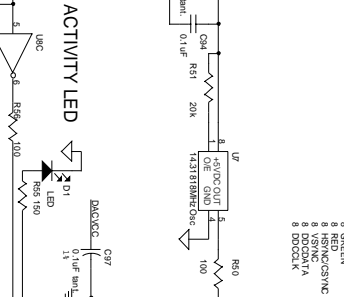
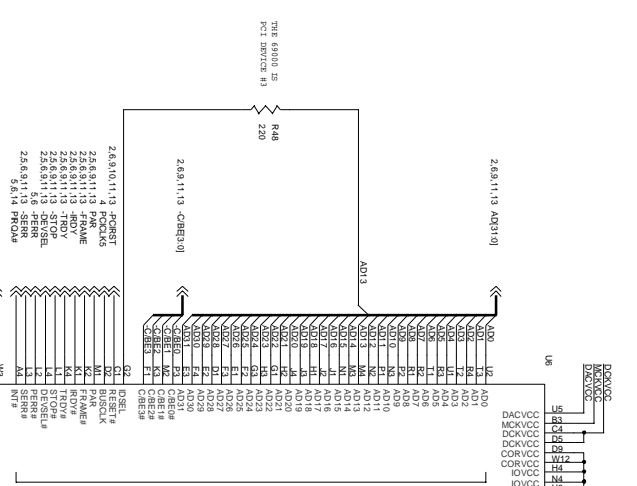
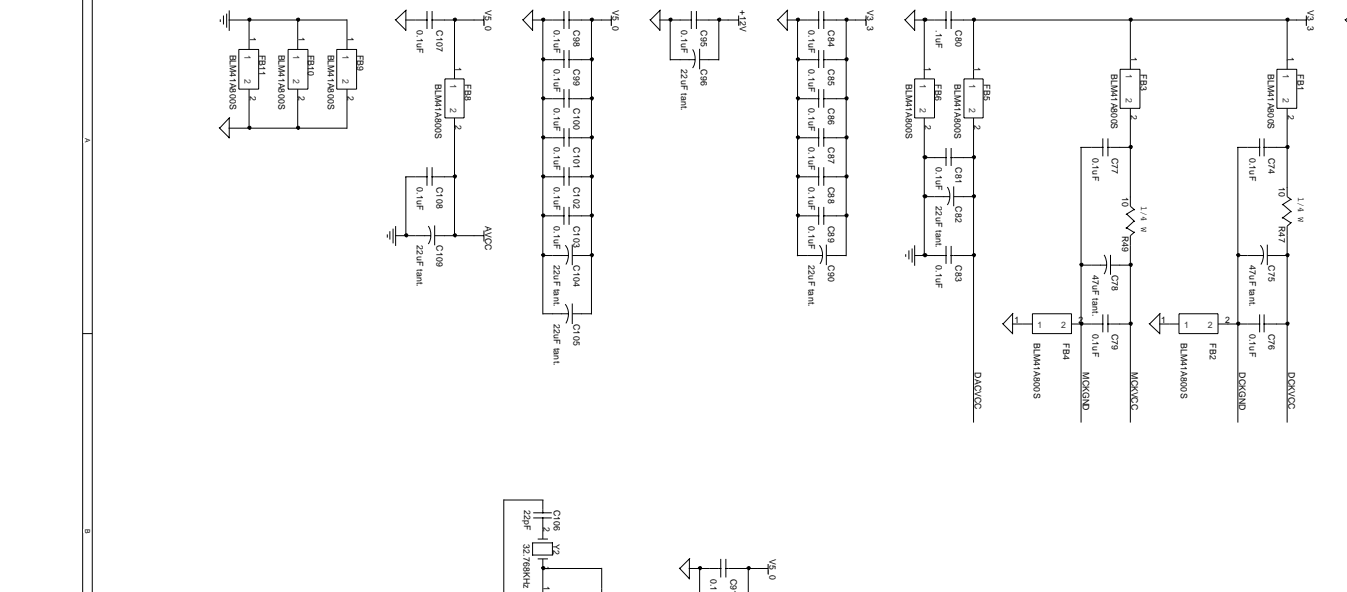
J8/J9 GND:
 A12, A13, A18, A24, A30, A35, A37, A42, A48, A56
 B3, B12, B13, B15, B17, B28, B34, B38, B46, B49,
 B57 | B2
 J8/J9 +12V: A2
 -12V: B1



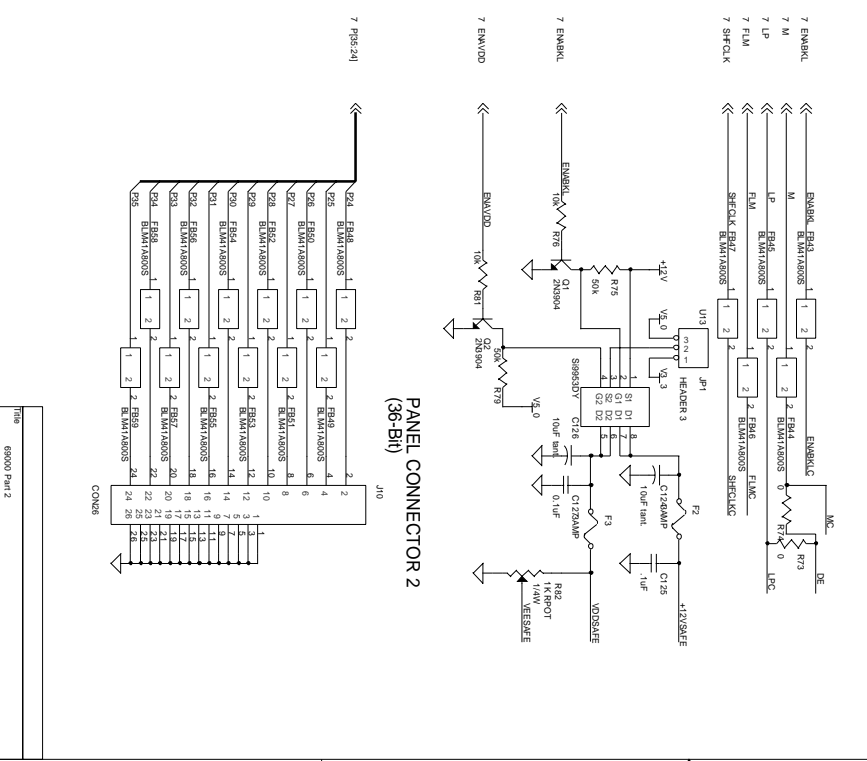
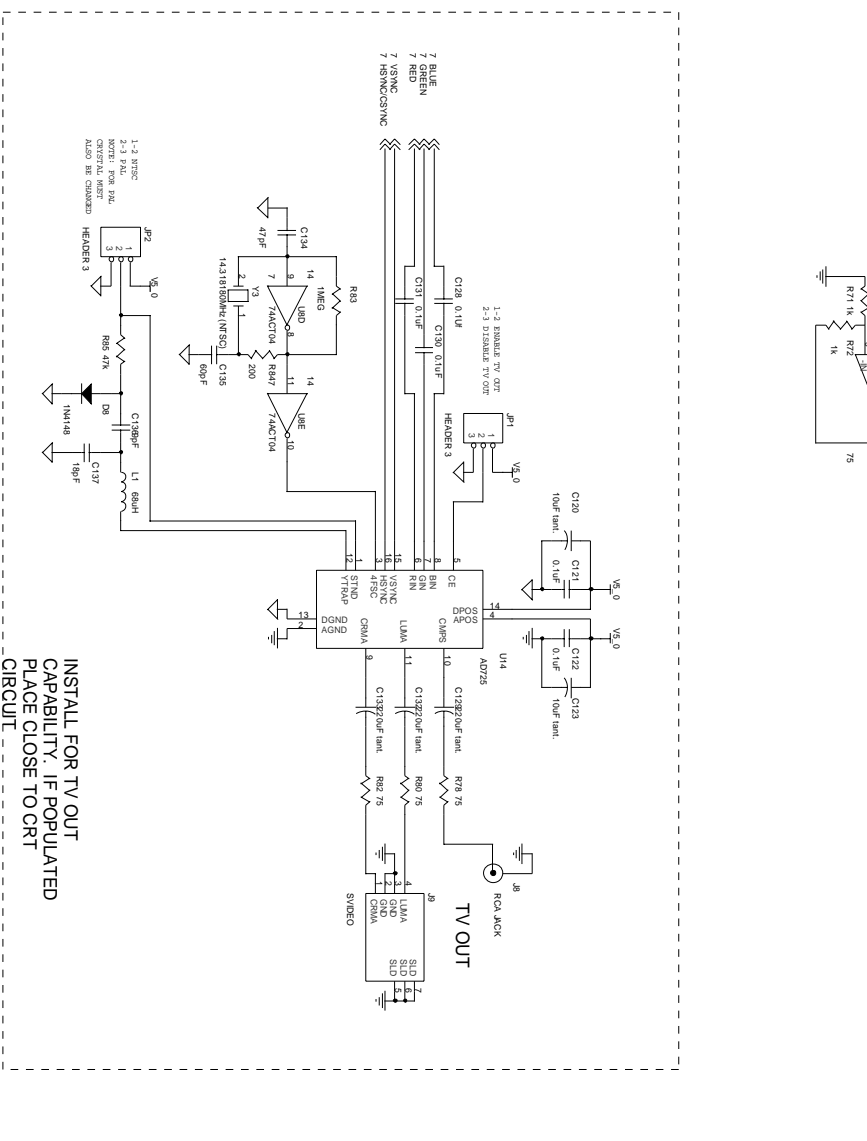
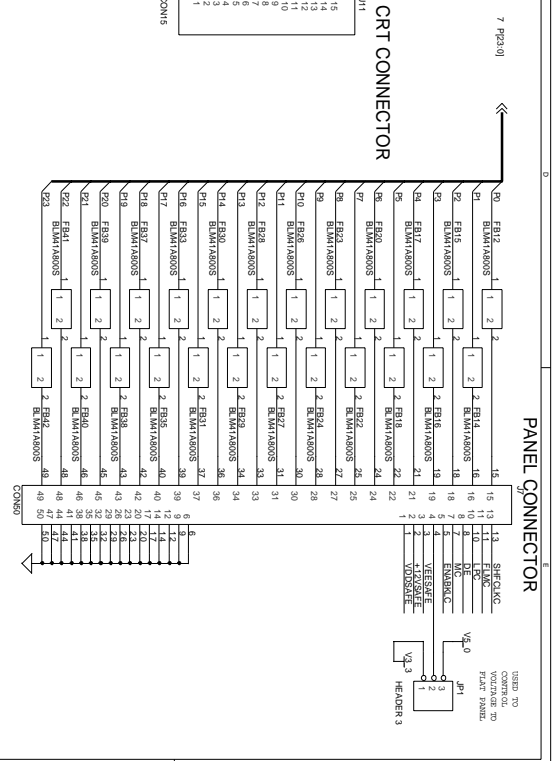
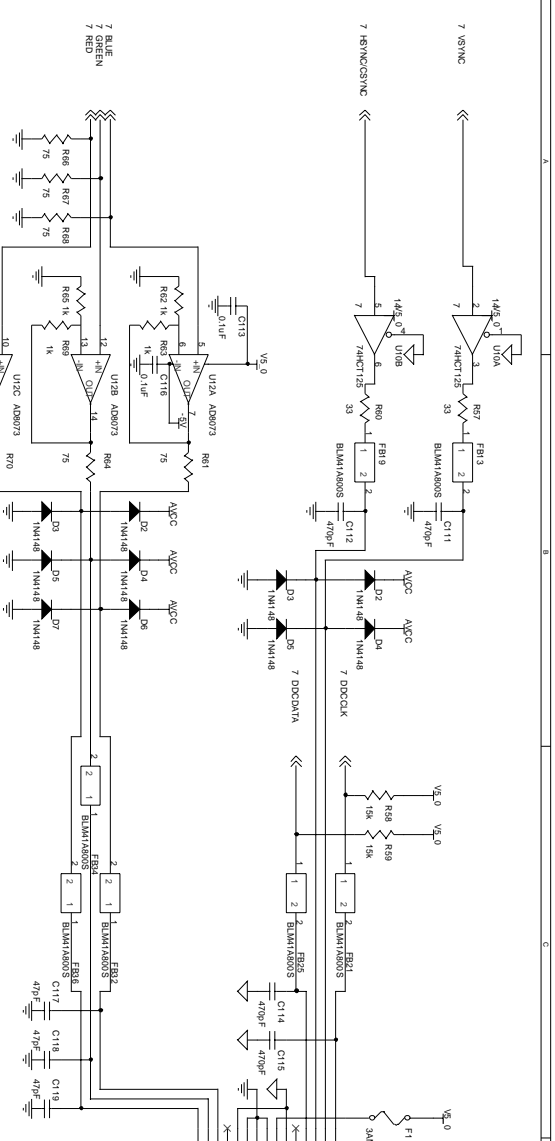
File	PCI Slot 0 & 1
Size	Document Number
Doc	
Revision	Rev A
Date	Thursday, October 24, 1998
Sheet	6 of 21



PLACE THESE CAPS CLOSE TO POWER PINS (CORRECT, INVERT, INVERT)

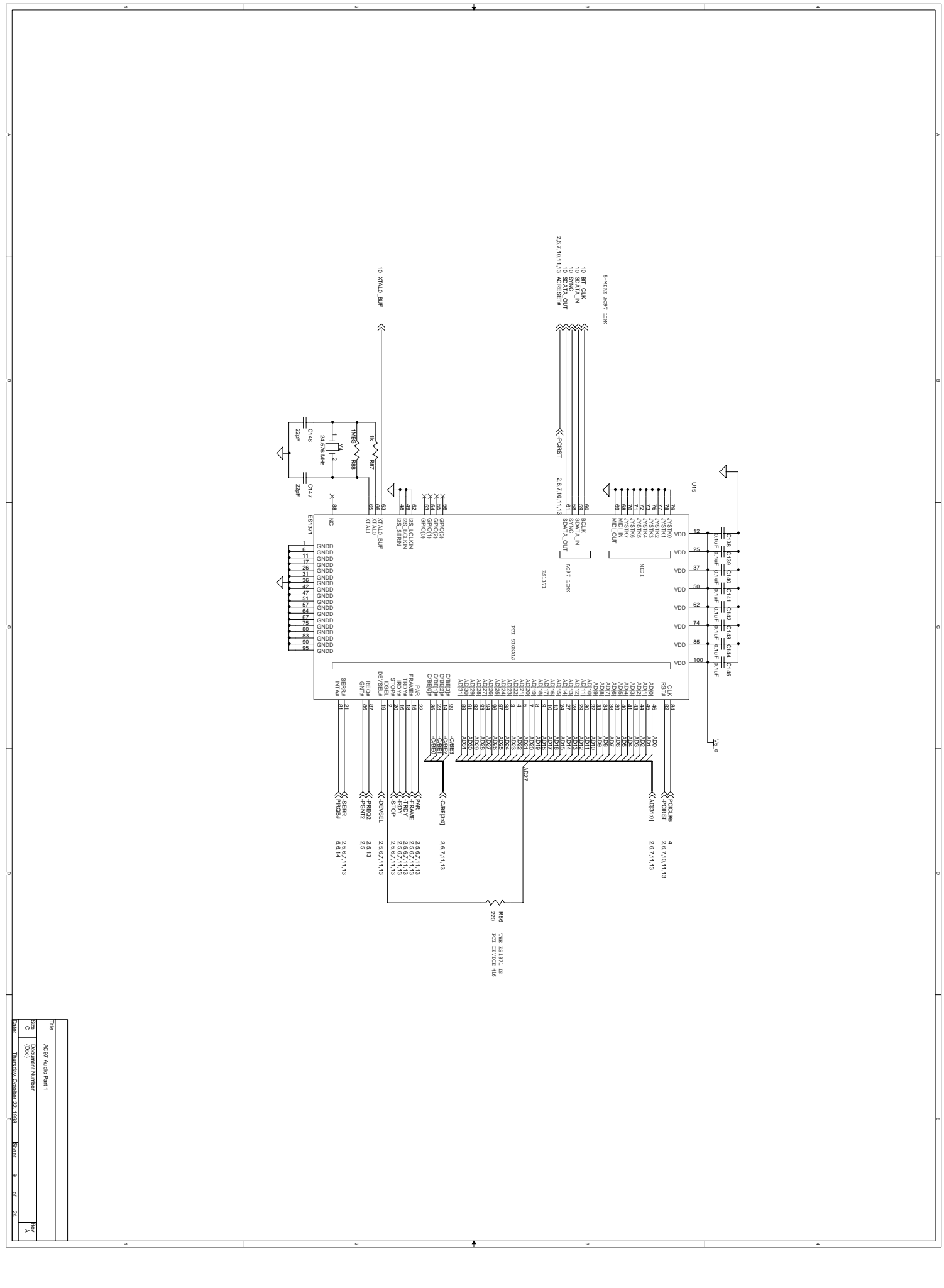


File	68000 Part 1
Docu	Docu
Sheet	1 of 21
Date	Thursday, October 21, 1999
Scale	7
Rev	A

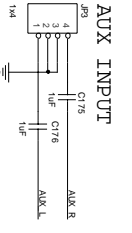


INSTALL FOR TV OUT
CAPABILITY. IF POPULATED
PLACE CLOSE TO CRT
CIRCUIT.

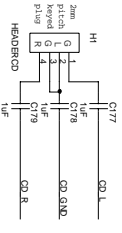
File	68000 Part 2
Size	Document Number
Doc	Doc
Rev	A



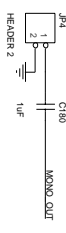
Title	AC97 Audio Part 1
Size	Document Number
C	(Doc)
Revision	Thursday, October 27, 1994
Page	9 of 24
Rev	A



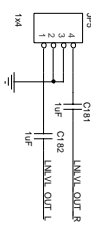
CDROM AUDIO HEADER



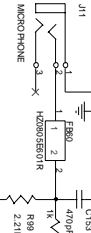
MONO OUT



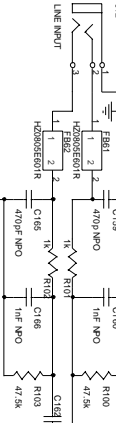
LINE LEVEL OUT



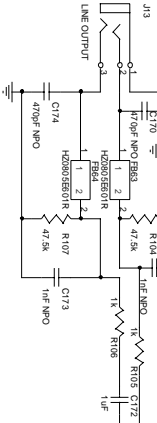
MICROPHONE



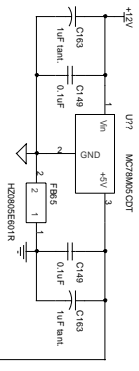
LINE INPUT

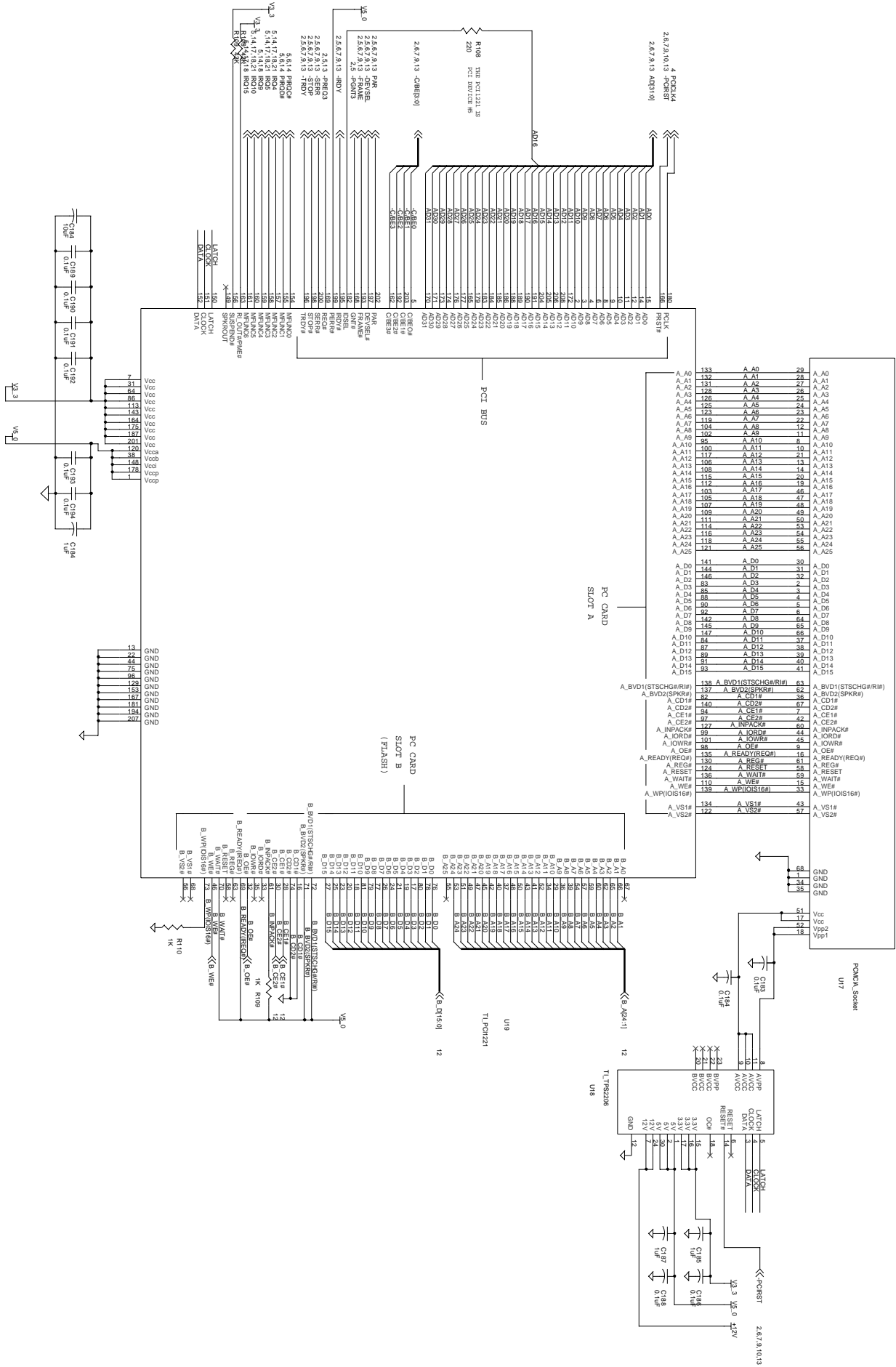


LINE OUTPUT

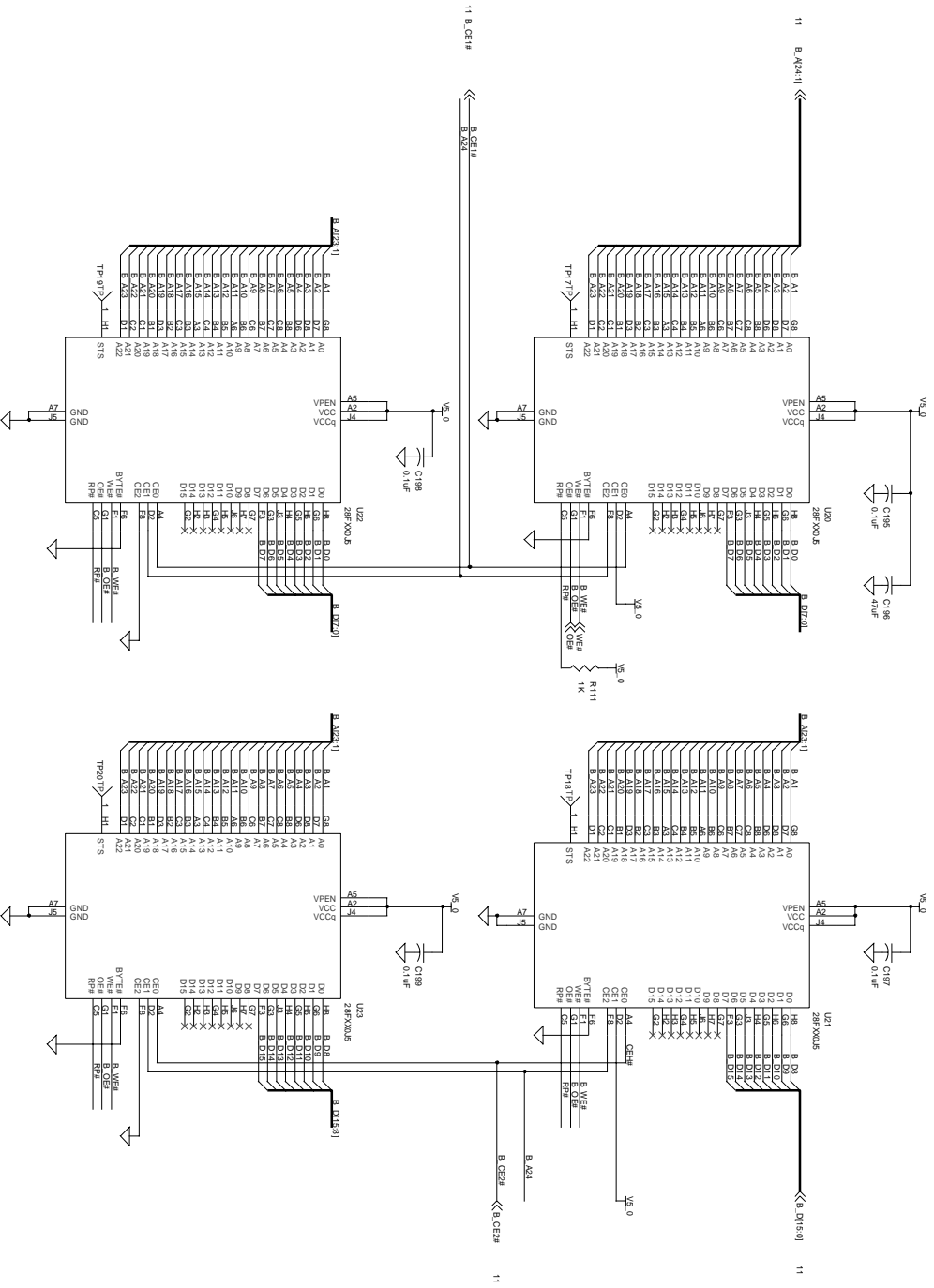


ALL RESISTORS AND CAPACITORS ARE 1% W
SEE APPLICATION NOTE FOR LAYER DEFINITIONS

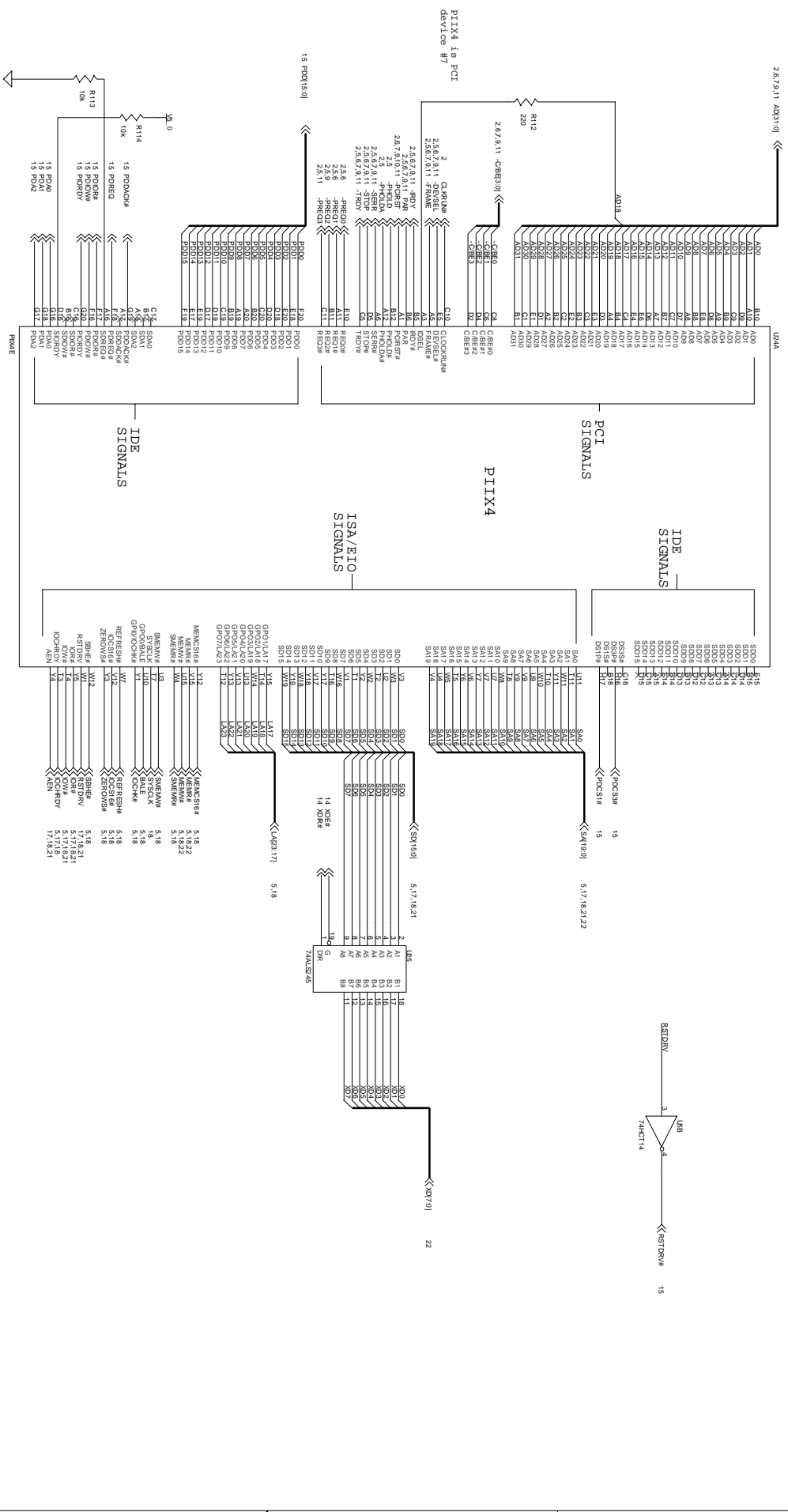




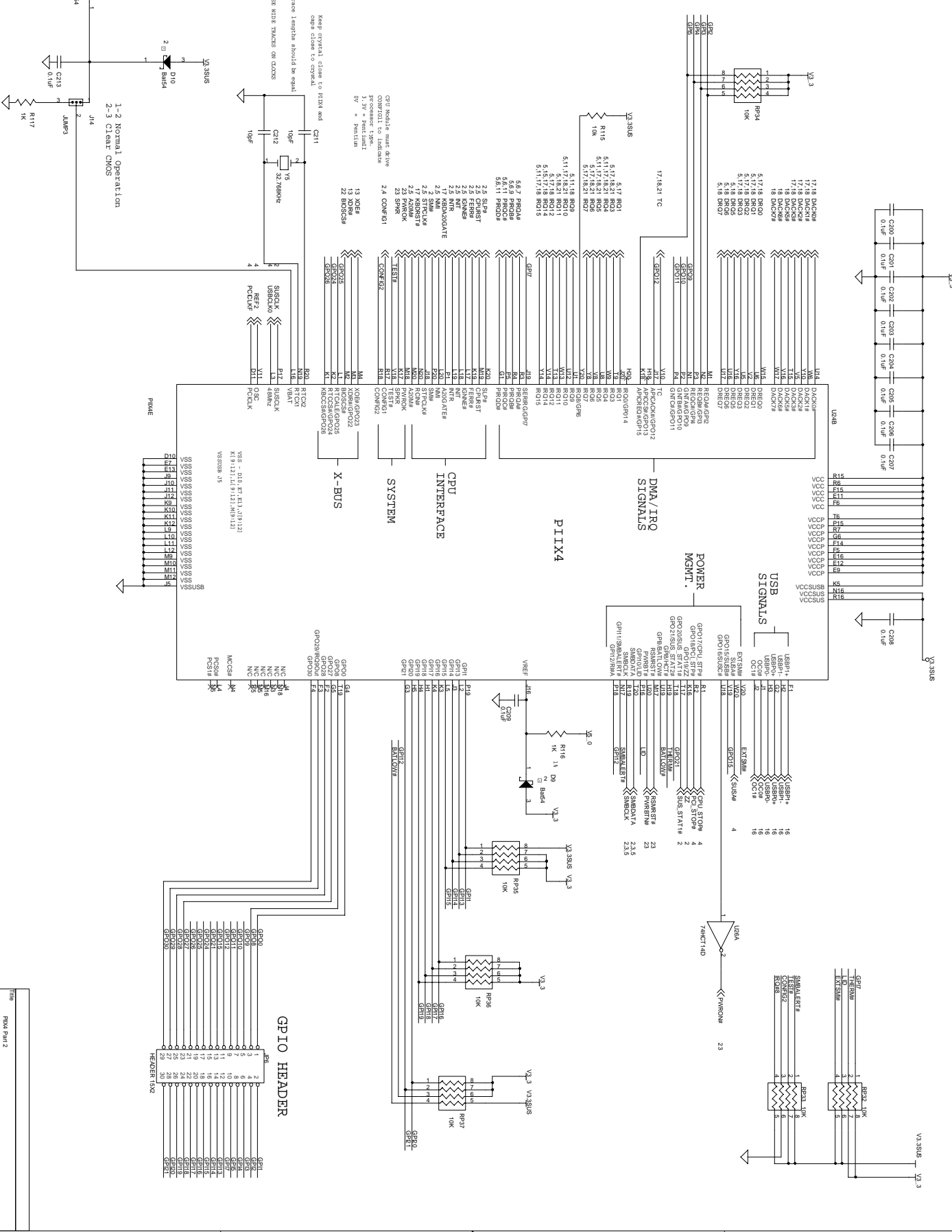
File	FCI1221
Size	Document Number
C	(Doc)
Thursday, October 24, 1996	Sheet 11 of 24
Rev	A



File	StrataFlash
Size	Document Number
C	(Doc)
Issue	Thursday, October 27, 1994
Page	12 of 24
Rev	A



Title	Print Part 1
Size	Document Number
Doc	C
Date	Thursday, October 27, 1999
Sheet	13 of 21
Rev	A



1-2 Normal Operation
2-3 Clear CMOS

Trace lengths should be equal
Use wide traces on clocks

Keep crystal close to fixed and
caps close to crystal

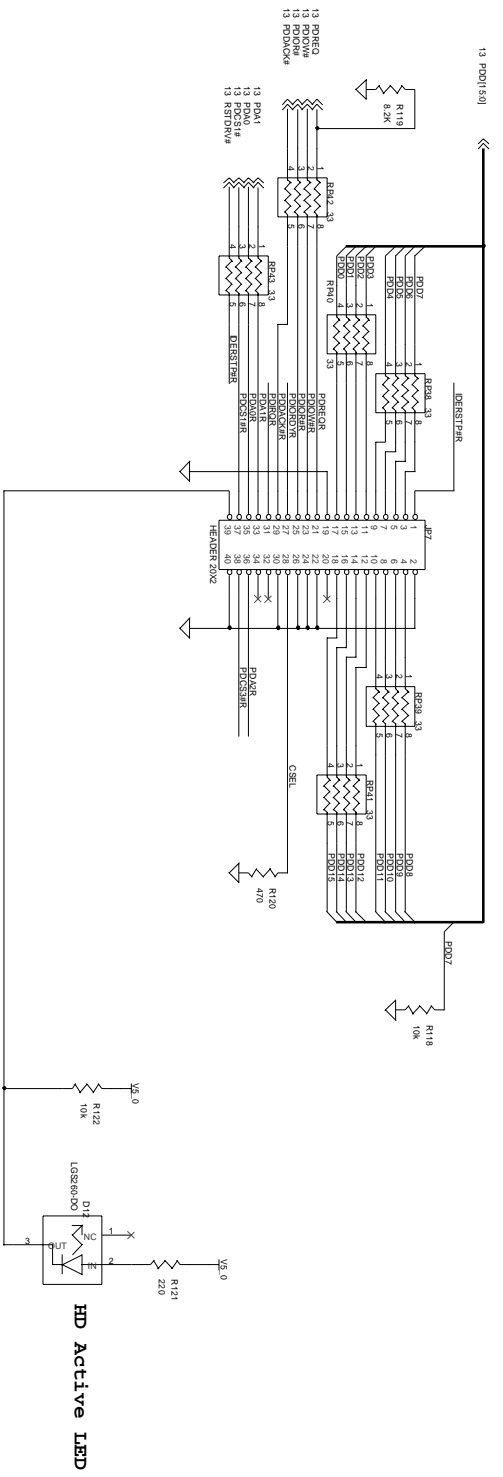
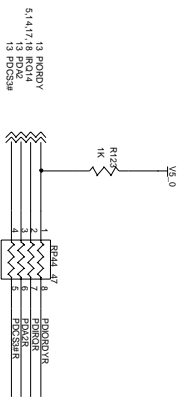
CPU Module must active
CONTROL to indicate
1.3V = Pentium I
1V = Pentium II

USB
POWER MGMT.
DMA/IRQ SIGNALS
PIIX4
CPU INTERFACE
SYSTEM
X-BUS

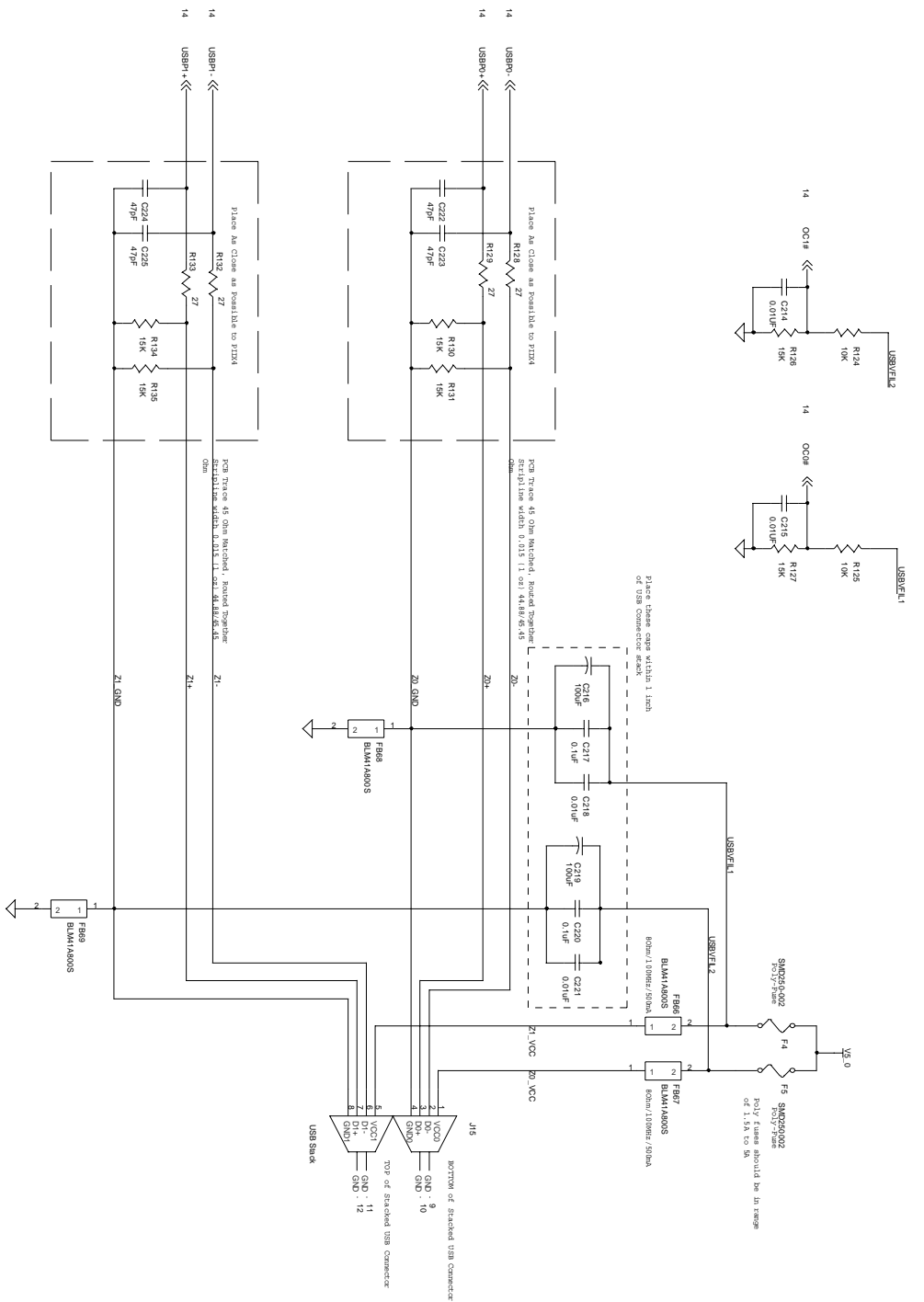
GPIO HEADER

Pin	Function
1	GPIO0
2	GPIO1
3	GPIO2
4	GPIO3
5	GPIO4
6	GPIO5
7	GPIO6
8	GPIO7
9	GPIO8
10	GPIO9
11	GPIO10
12	GPIO11
13	GPIO12
14	GPIO13
15	GPIO14
16	GPIO15
17	GPIO16
18	GPIO17
19	GPIO18
20	GPIO19
21	GPIO20
22	GPIO21
23	GPIO22
24	GPIO23
25	GPIO24
26	GPIO25
27	GPIO26
28	GPIO27
29	GPIO28
30	GPIO29
31	GPIO30
32	GPIO31
33	GPIO32
34	GPIO33
35	GPIO34
36	GPIO35
37	GPIO36
38	GPIO37
39	GPIO38
40	GPIO39
41	GPIO40
42	GPIO41
43	GPIO42
44	GPIO43
45	GPIO44
46	GPIO45
47	GPIO46
48	GPIO47
49	GPIO48
50	GPIO49
51	GPIO50
52	GPIO51
53	GPIO52
54	GPIO53
55	GPIO54
56	GPIO55
57	GPIO56
58	GPIO57
59	GPIO58
60	GPIO59
61	GPIO60
62	GPIO61
63	GPIO62
64	GPIO63
65	GPIO64
66	GPIO65
67	GPIO66
68	GPIO67
69	GPIO68
70	GPIO69
71	GPIO70
72	GPIO71
73	GPIO72
74	GPIO73
75	GPIO74
76	GPIO75
77	GPIO76
78	GPIO77
79	GPIO78
80	GPIO79
81	GPIO80
82	GPIO81
83	GPIO82
84	GPIO83
85	GPIO84
86	GPIO85
87	GPIO86
88	GPIO87
89	GPIO88
90	GPIO89
91	GPIO90
92	GPIO91
93	GPIO92
94	GPIO93
95	GPIO94
96	GPIO95
97	GPIO96
98	GPIO97
99	GPIO98
100	GPIO99

Primary IDE Connector



Title	IDE Connector
Size	Document Number
Rev	C
Date	Thursday, October 27, 1994
Sheet	16 of 24
Rev	A



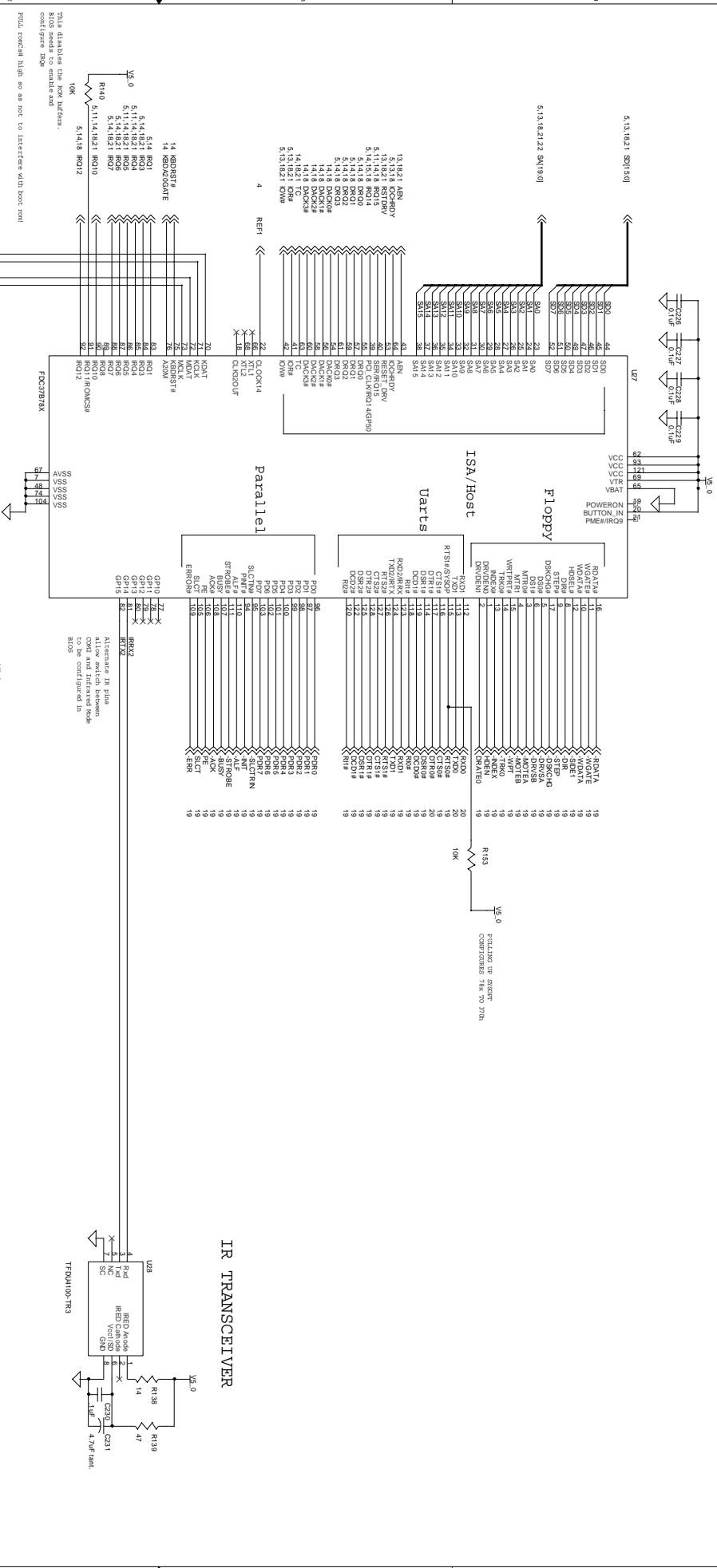
NOTE 1: USB differential traces route together (Z0- & Z0+) and (Z1- & Z1+).
 Must be 45 Ohm Matched
 Stripline width 0.015 (For 1 oz)->44.88/45.45 Ohm.

NOTE 2: Protect differential traces w/ guard traces or
 double space to any other signal.

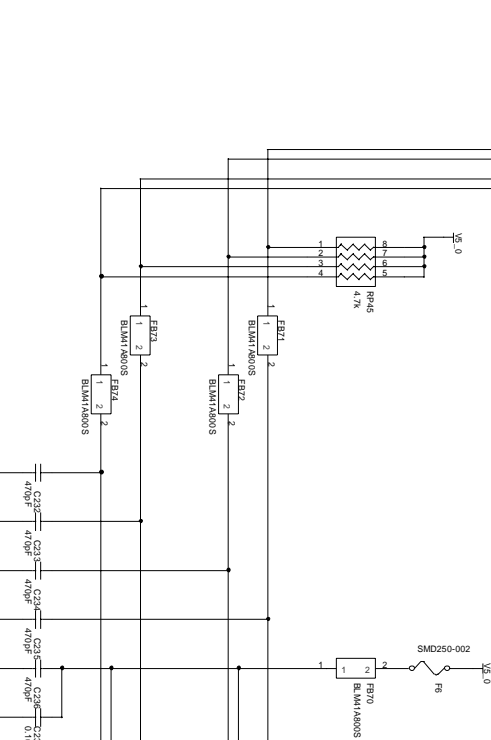
NOTE 3: Place ferrites at connector.

NOTE 4: Poly-fuse min 1.5A
 max 5A.

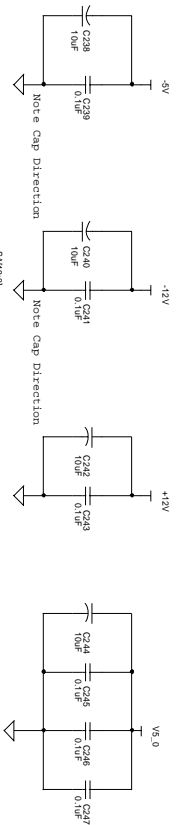
File	USB Connectors
Size	Document Number
C	Doc
Thursday, October 24, 1996	Sheet 16 of 21
Rev	A



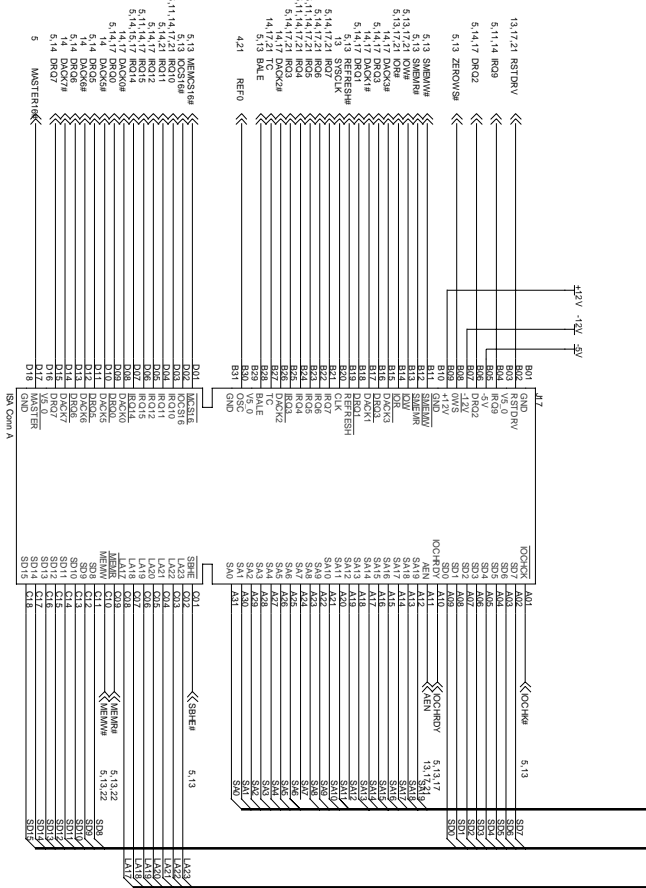
This disables the ROM buffers.
 ROM needs to be enabled
 configure 00h
 Pull, normally high so as not to interfere with boot rom



ISA Slots

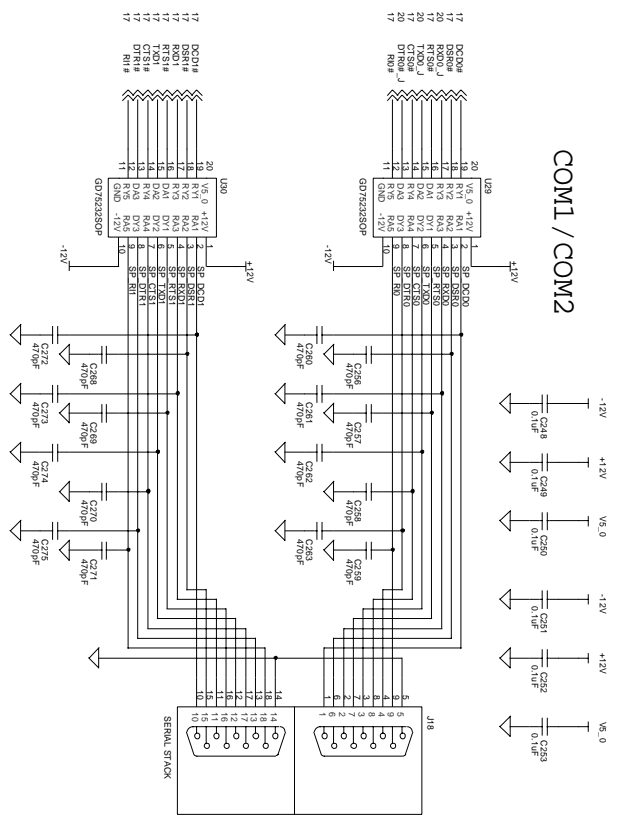


5.13.17.21.22 SW19/90 << SW19/90 >>
 5.13.17.21 SW19/90 << SW19/90 >>
 5.13 LA2317 << LA2317 >>

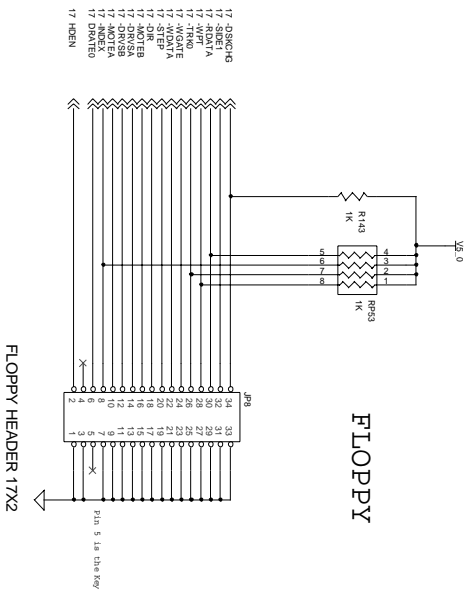


J17/J18 V5.0:
 B31, D16
 J17/J18 GND:
 B01, B10, D18
 J17/J18: +12V B09
 -12V B07
 -5V B05

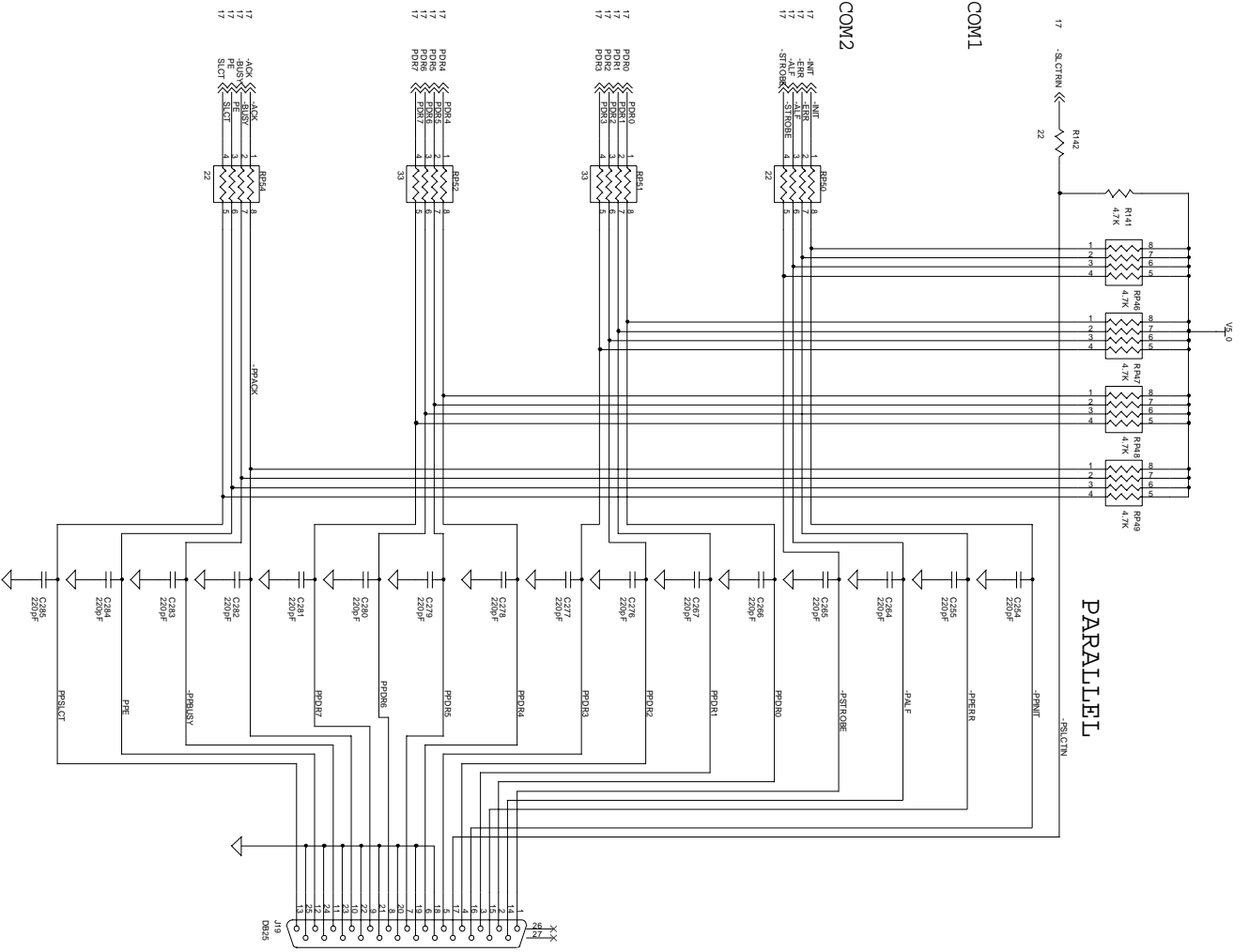
COM1 / COM2



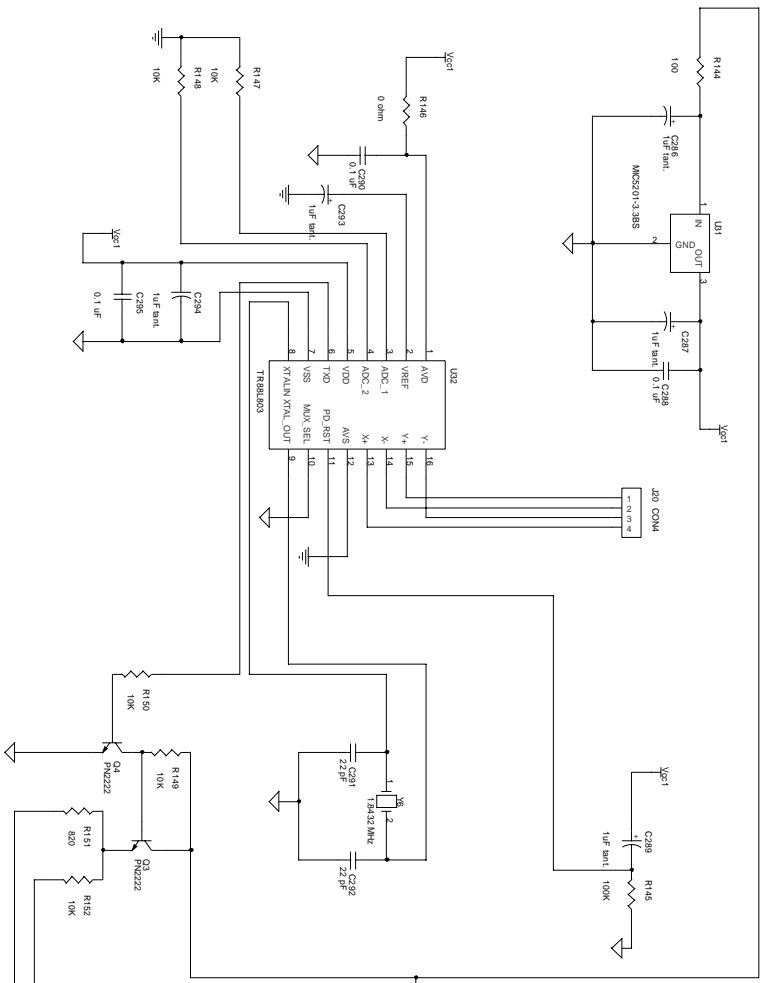
FLOPPY



PARALLEL



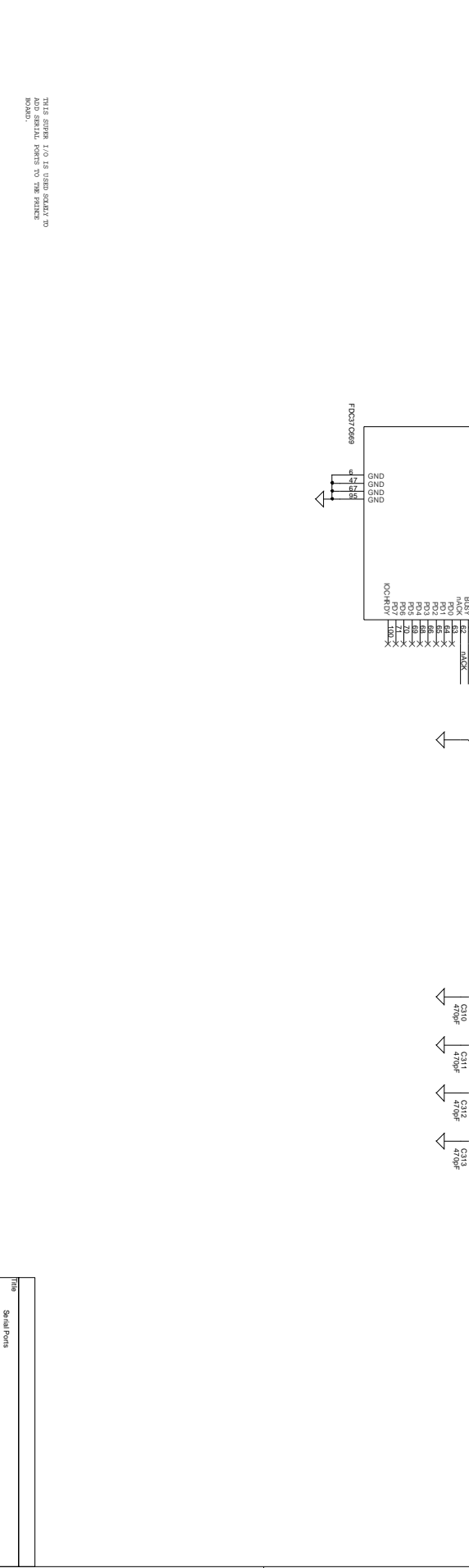
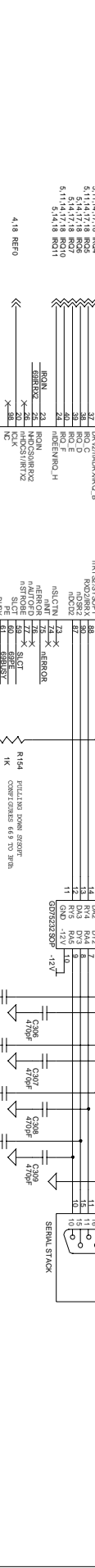
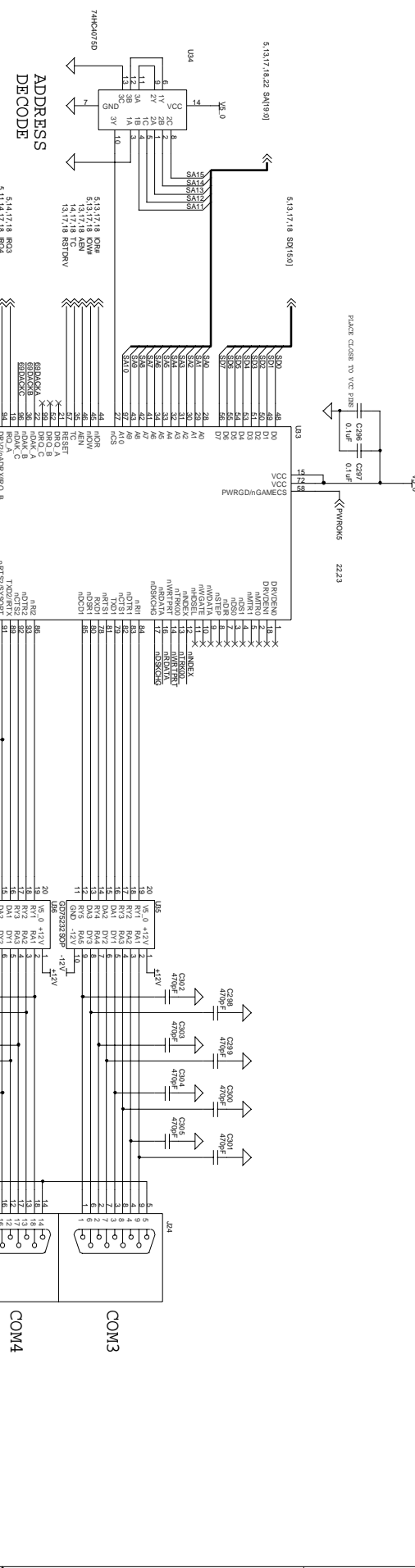
File	COM1, DREG, Floppy
Size	Document Number
Doc	C
Date	Thursday, October 24, 1996
Sheet	10 of 21
Rev	A



THESE JUMPER ARE USED TO SWITCH BETWEEN COM1 AND TOUCH FOR TOUCH SCREENS 2.3" FOR COM1

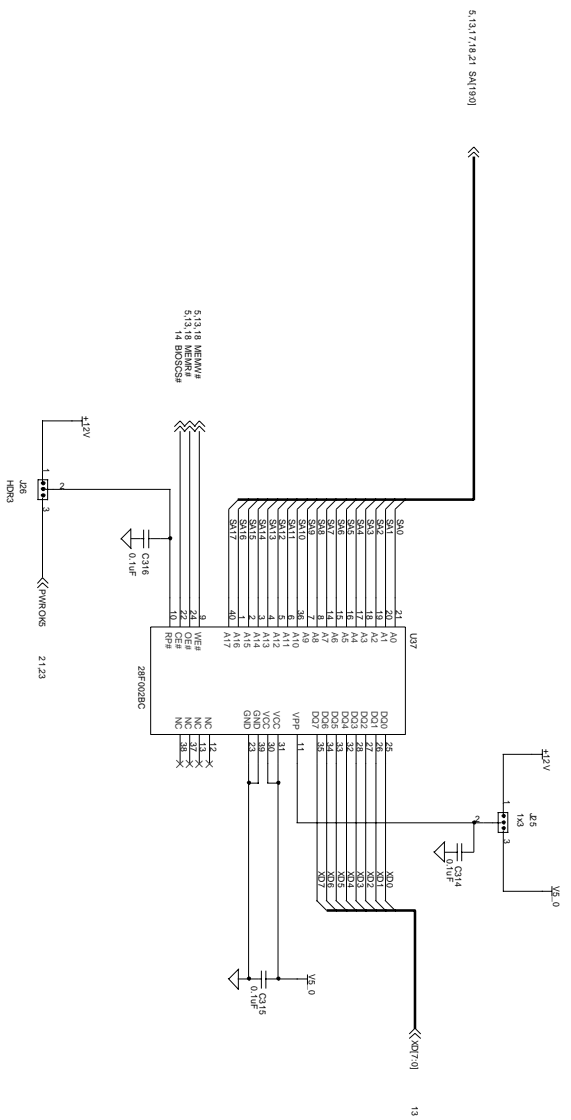


File	Touch Screen Controller
Size	Document Number
C	(Doc)
Issue	Thursday, October 24, 1998
Scale	20 of 24
Rev	A

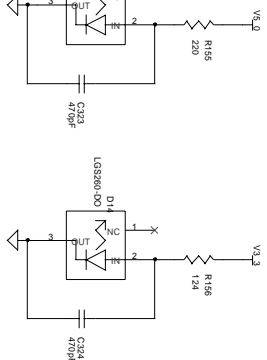


THIS SKETCH IS TO BE USED EXACTLY AS SHOWN. NO CHANGES SHOULD BE MADE TO THE PRINTED BOARD.

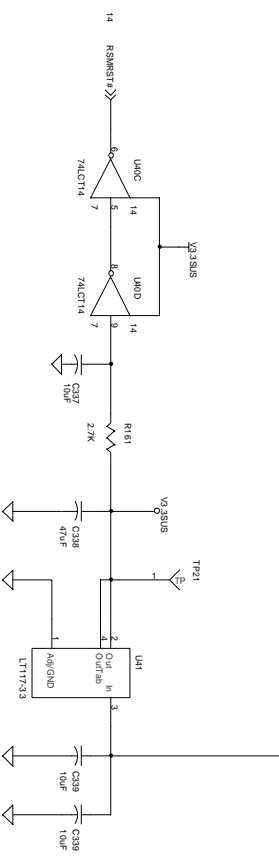
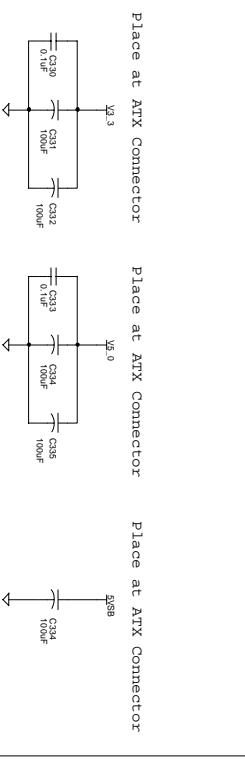
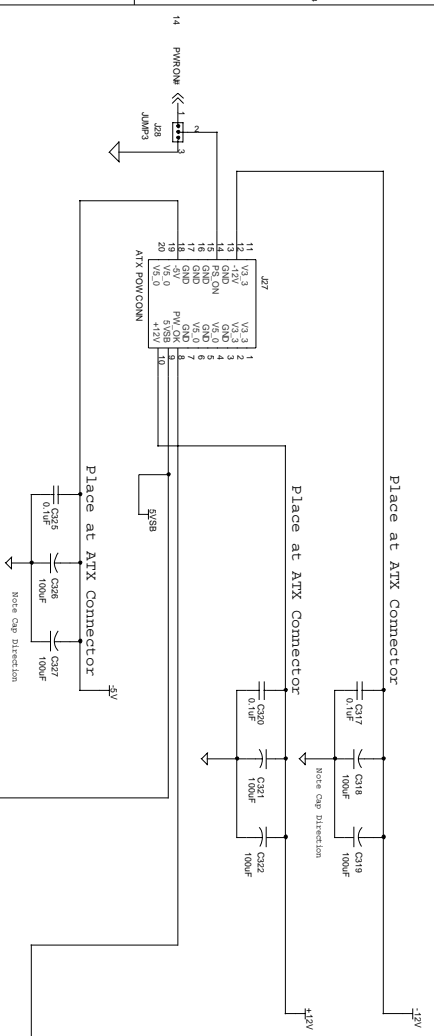
File	Serial Ports
Size	Document Number
C	(Doc)
Thursday, October 24, 1998	Sheet 21 of 24
Rev	A



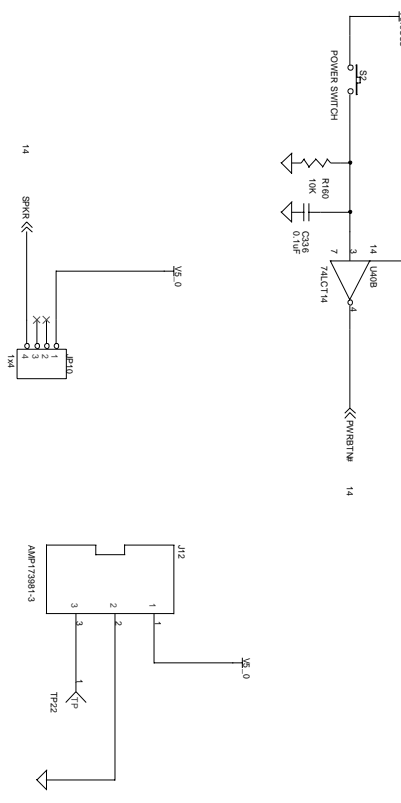
Power Indicators



Note: Add screen marking for V5_0 LED, V3_3 LED

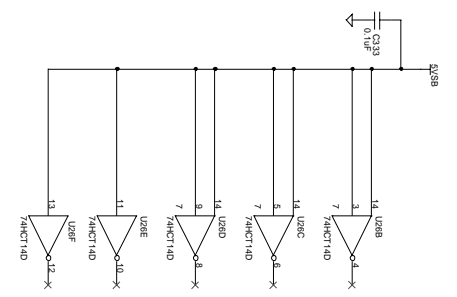
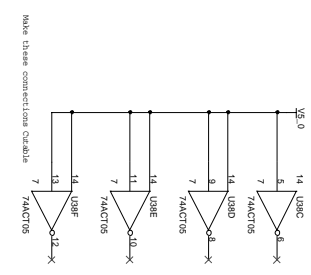
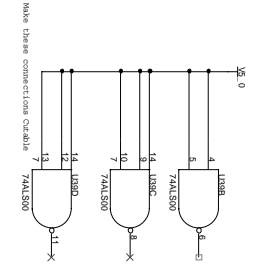
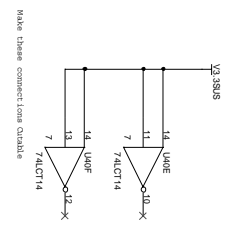
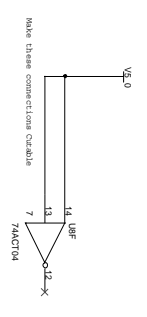
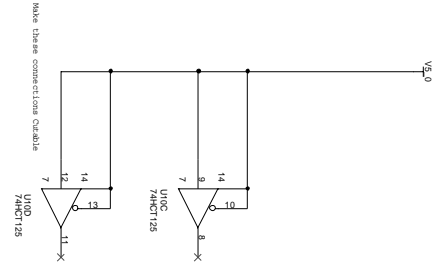
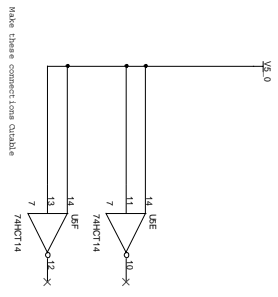


SPEAKER HEADER



- NOTE 1: Locate Fan Header next to Processor Assembly
- Note 2: Connect V5_0 feed for analog supplies AVDD1 and CLKAVDD1 at one point only on PCB
- Note 3: Place power connector decoupling at ATX connector
- NOTE 4: Note direction of caps on -12V, -5V supplies.

File	Rev
ATX Power Connector	A
Document Number	
Size C (Doc)	
Thursday, October 24, 1996	
Page 20 of 24	



File	Unused Gates
Size	Document Number
C	(Doc)
Issue	Issue
24	24
Rev	A