CIRRUS LOGIC®

CL-PS7500FE Development Kit

Hardware User's Guide

Embedded Processors Division

This document describes the CL-PS7500FE development board provided by Cirrus Logic Inc. No warranty is given for the suitability of this design for any purpose other than demonstrating functional operation of the CL-PS7500FE. The information contained in this document is subject to change without notice.

Version 1.0 March 1999

Table of Contents

List of Tables

List of Figures

1 Introduction

The CL-PS7500FE development board is targeted at system designers who are developing CL-PS7500FE based hardware platforms. This document defines the baseline board-level hardware elements that comprise the Cirrus Logic CL-PS7500FE development board. This document references other documents for more specific details of processors and other devices (ICs) used on the design. Scope is restricted to covering the printed circuit board, connections between components on that board, and between the board and other components intended to connect to the development board.

2 Terms and Definitions

- ARM® Advanced RISC Machines Limited. The company that developed the ARM710 processor core in the CL-PS7500FE.
- baud The number of bit transitions per second on a serial line. This is different from bits per second since there are control bits that surround the data bits in a serial transmission stream. It is there always the case that bps < baud on a serial line.
- bps bits per second.
- bpp bits per pixel.
- COM1 Serial port #1 on the development board.
- COM2 Serial port #2 on the development board.
- DAC Digital to Analog Converter.
- DMA Direct Memory Access.
- EDO Extended Data Output. A more efficient means of DRAM data access than FPM.
- FIFO First In First Out. A memory block where data is read out from a single location in the exact sequence in which it was written.
- FLASH A type of non-volatile memory.
- FPM Fast Page Mode. Method of DRAM data access.
- IrDA Infrared Data Association. This body sets the standard for infrared communication.
- ISA Industry Standard Architecture. This is an older bus common to the IBM® PC and its compatibles.
- LCD Liquid Crystal Display.
- LPT Parallel port (much like in a PC).
- LSB Least Significant Bit.
- Mbps Million of bits per second.
- MSB Most Significant Bit.
- NTSC Television video data broadcast format common to North America. Actually stands for National Television System Committee.
- PAL Television video data broadcast format common to Europe. Actually stands for Phase Alternation by Line.
- SIMM Single In-Line Memory Module. SIMMs are available with either DRAM or FLASH on them; both used on this design.
- Qword Quad Word. Four 32-bit words of data.
- Raster A line of pixels on a display.
- UART Universal Asynchronous Receiver Transmitter.

3 Development Board

The development board is designed as a prototype with limited debug capabilities, such as test headers for logic analyzer interfacing, option memory and ISA type bus. It is not intended to be a production design.

3.1 Main Feature Set

The development board has the following features:

- Processor
  - CL-PS7500FE running at 56MHz

- Memory
  - Two 72 PIN SIMM sockets for Fast Page Mode or EDO DRAM, capable of supporting up to 64MBytes per socket.
  - User selectable boot ROM
  - FLASH SIMM up to 16MBytes in size, configured as 32-bit words (not included in kit)

- Peripherals
  - Keyboard interface - PS/2
  - Mouse interface - PS/2
  - Two serial ports (16C550A-compatible)
  - IrDA compliant serial port (serial port back-end) (shared with COM2)
  - Parallel port
  - 10BaseT Ethernet interface (CS8900A)
  - SVGA port
  - LCD module site for supporting custom LCDs
  - Two ISA type slot connectors
  - Stereo headset port (CS4333)
  - User controllable 8-LED bar for OS porting/debugging

3.2 Population Options

DRAM The board has two 72 pin DRAM SIMM sockets supporting from 4MBytes to 128MBytes of memory. Typically most applications will require between 8MBytes to 16MBytes of DRAM. Both Fast Page Mode (FPM) and Extended Data Output (EDO) DRAM are supported. Note: software must be aware of the type of DRAM used.

FLASH The board has a single FLASH SIMM socket on it. This is a standard 80-pin socket and can support up to 16Mbyes of FLASH. The type of FLASH used is critical to application development as FLASH devices do not have common erase/write state machines. The FLASH SIMM option has been selected to permit developers to install modules populated with their own choice of FLASH. A seven bit register is available, connected to the SIMM's information bits. This register is intended to permit interrogation of the SIMM to determine its size. Further information on this standard can be obtained from Global Engineering, by referring to JEDEC standard No. 21-C.

Note: Although JEDEC standard No. 21-C states that SIMMs may be sized from 128KBytes to 32MBytes, this design does not support SIMMs larger than 16MBytes. Note also that EPROM emulators are available that are able to connect 80-pin SIMM sockets. This can be used in place of a SIMM during code development.

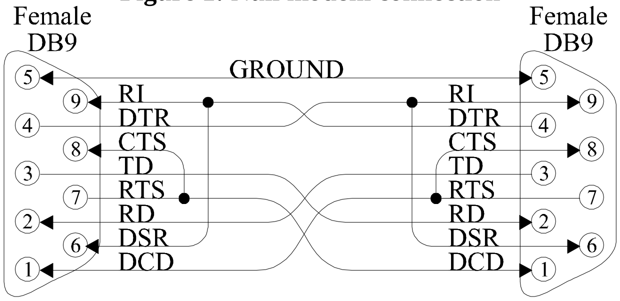
ROM The development board comes with an EPROM that contains the Angel™ debug monitor. The boot ROM is enabled via a jumper and when selected, moves FLASH from an initial location of 0x0000:0000 in the CL-PS7500FE's memory to 0x0100:0000, while switching the boot ROM into location 0x0000:0000. When the boot ROM is not selected, it will not be addressable by the CPU.

3.3 Board Set-up

A Null modem cable is supplied to allow communication between the development board and the host PC. Communication is set-up by connecting COM1 of the development board to any available COM port on the host PC.

Note: Null modem cables differ substantially from a straight-through cable. See Figure 1 below:

Figure 1: Null modem connection



3.4 Power

A 12V 1500mA AC adapter is provided with the development board. The AC adapter is plugged into J23.

Alternatively, a standard PC/AT power supply can be connected to the development board for supplying power. This may be a more attractive alternative during development if ISA plug-in boards with high power consumption are installed.

3.5 Jumpers

There are five jumpers on the development board. Their functions are summarized below (JP3-JP10 are logic analyzer headers):

JP1: SnA status select. This line is a processor clock control line used to set the relationship between MCLK and FCLK for the CL-PS7500FE processor. If a jumper is installed on this pin, the memory system clock and the CPU clock run at different rates. If low, the two clocks are driven by the clock signal entering on the MCLK pin. For further details on this jumper, refer to Section 4.4.

JP2: Video and CPU system clock setting. This is a 2x7 jumper that is used to configure the clock that the video system runs from. Please refer to Sections 4.4 and 4.13 of this document for details on this jumper.

JP11: Boot ROM size select. This is a 1x3 jumper. When a shunt is put on pins 1-2, 4 MBytes of EPROM is assumed. When a shunt is put on 2-3, 1 MByte of boot ROM is assumed. For further details on this jumper, refer to Section 4.3.

JP12: Boot mode select. If a shunt is installed on this 1x2 jumper, the system will boot from the Boot ROM. Otherwise, the system will boot from the FLASH SIMM. For further details on this jumper, refer to Section 4.3.

JP15: -5V power source select. This jumper is used to configure a supply or an external supply of -5V. For further details, refer to Section 4.15.4.

4 Hardware Design Details

Figure 2 shows a block diagram of the development board. Only a limited number of buses and physical connections are shown as a full illustration here would be prohibitively complex.

Figure 2: Development Board Functional Block Diagram

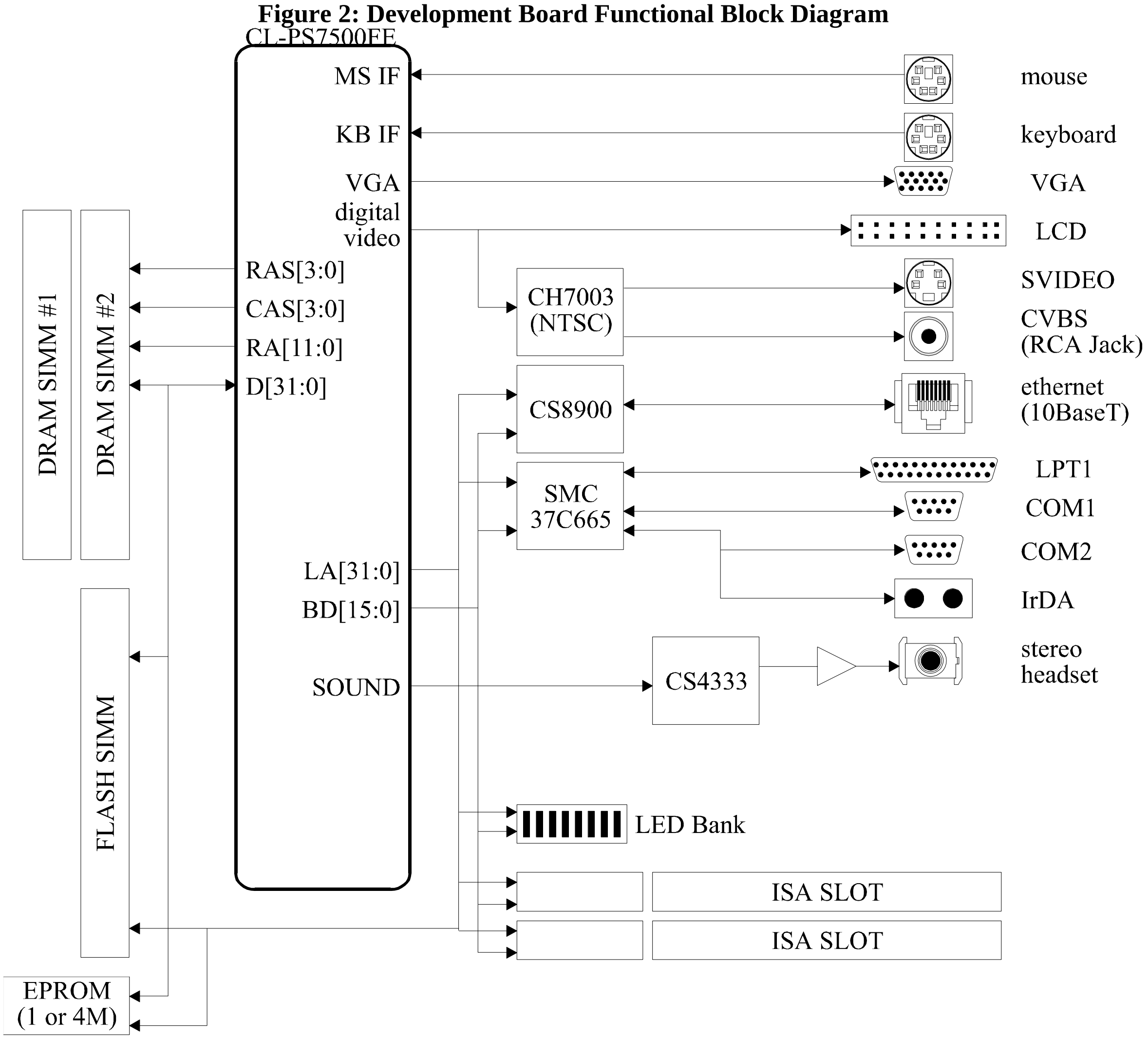
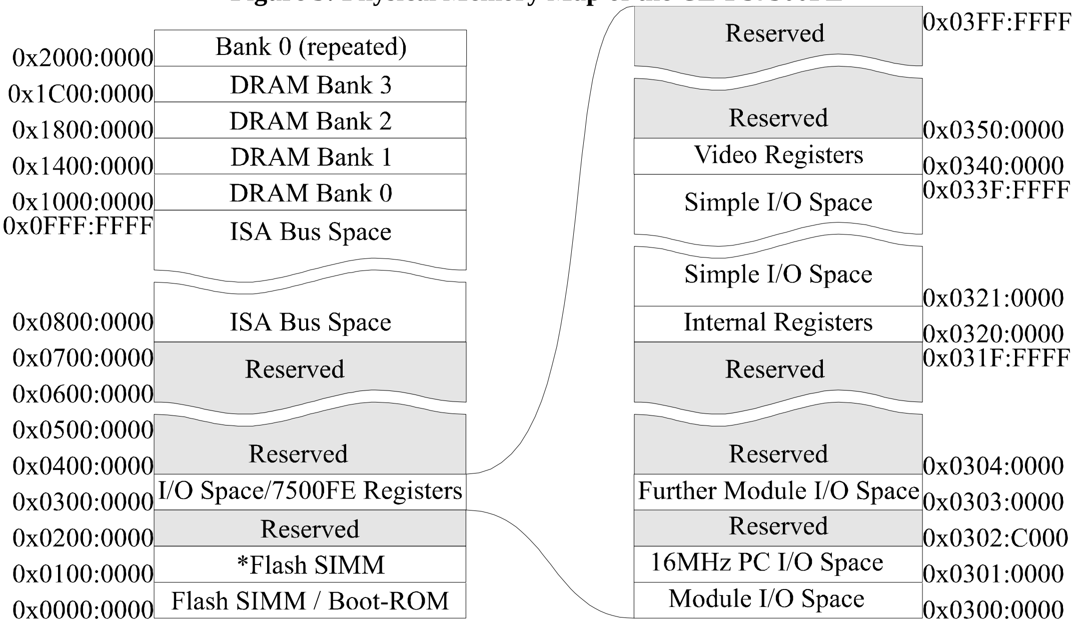


Figure 3 illustrates how physical devices are mapped into the CL-PS7500FE's memory space.

Figure 3: Physical Memory Map of the CL-PS7500FE



*only when Boot-ROM is enabled

Table 1 lists the peripheral chip selects and interrupt request lines to the CL-PS7500FE used by each external device.

Table 1: Chip Select Usage

Device	*Chip Select*	*Address Range*
ISA DMA Acknowledge reg.	nEACS	0x0Enn:nn2n..0x0Fnn:nn3n
ISA DMA Acknowledge reg.	nEACS	0x0E00:0000..0x0FFF:FFFF
ISA Bus I/O access	nEACS	0x0C00:0000..0x0DFF:FFFF
ISA Bus memory access	nEACS	0x0800:0000..0x0BFF:FFFF
CS8900A I/O decode (fast)	nSIOCS2	0x0334:0C00..0x0334:8FFF
CS8900A I/O decode (med.)	nSIOCS2	0x032C:0C00..0x032C:0CFF
CS8900A I/O decode (slow)	nSIOCS2	0x0324:0C00..0x0324:0CFF
LED Bar/FLASH SIMM status	nPCCS1	0x0302:BA00..0x0302:BAFF
ISA DMA/IRQ register (read)	nPCCS2	0x0302:B000..0x0302:B7FF
CS8900A DMA Ack.	nCDACK	0x0301:2000..0x0302:AFFF
Serial Ports (37C665)	nCCS	0x301:0000..0x0301:FFFF
Parallel Port (37C665)	nCCS	0x0301:0000..0x0301:1FFF
FLASH (when ROM enabled)	nROMCS	0x0100:0000..0x01FF:FFFF
ROM (when enabled)	nROMCS	0x0000:0000..0x003F:FFFF
FLASH (when ROM disabled)	nROMCS	0x0000:0000..0x00FF:FFFF

Table 2 shows the interrupt number assignment on the CL-PS7500FE development board.

Table 2: Interrupt request usage on the CL-PS7500FE

IRQ #	*Used by*
1	ISA DMA Requests (all come to this IRQ)
2	Parallel Port (LPT1)
3	Unused
4	CS8900A Interrupt request (Ethernet)
5	CS8900A DMA request (Ethernet)
6	Serial Port #1 (COM1)
7	ISA Interrupt requests (all come to this IRQ)
8	General Purpose IRQ (J4)
9	Serial Port #2 (COM2)

As can be seen in the above table, interrupts 1 and 7 are used to represent exceptions from multiple sources on the ISA bus. These are discussed in further detail in Section 4.10.2.

4.1 DRAM

DRAM is mapped to the range of 0x1000:0000 to 0x1FFF:FFFF (maximum DRAM space is 256MBytes) in the CL-PS7500FE's address space.

There are two 72-pin SIMM sockets on the development board. The sockets may be populated with either FPM or EDO SIMMs. The maximum SIMM size is 128MBytes. SIMMs need not be populated in pairs.

If less than a full 128MByte SIMM is used, the memory will likely be fragmented into smaller segments throughout the memory space. Enabling the MMU in the CL-PS7500FE and re-mapping all of the fragments into a contiguous space can solve this problem.

4.2 FLASH

FLASH is mapped to the range of 0x0000:0000 to 0x0100:0000 on the board (16MBytes). The CL-PS7500FE will begin executing at location 0x0000:0000 under normal circumstances.

FLASH is populated using an 80-pin FLASH SIMM. This approach has several advantages. First, the SIMM can be sized according to the requirements of the application. Second, the speed of the SIMM will be dictated by whether the application is to run or load out of FLASH. Perhaps most importantly, FLASH SIMMs can be procured with different manufacturer's parts (i.e. the SIMM may be populated with Intel's FLASH, or AMD's FLASH). If a developer already has experience with one particular device, or a device with a more feature-rich erase/write state-machine becomes available, a SIMM can be procured with that device on it.

Although standard FLASH SIMMs are available up to 32MBytes, this design does not support SIMMs larger than 16MBytes. The reason for this is that although the ROM space of the CL-PS7500FE has a maximum of 32MBytes, when booting from the on-board boot ROM, a FLASH burning utility may be loaded. In this event, the FLASH is re-mapped to the upper 16MBytes of the ROM space.

In order to be compatible with the evaluation software provided in the kit, one of the following Sharp FLASH SIMMs should be purchased:

Table 3: Recommended FLASH SIMMs

Part number	*Size*	*Memory Organization*
LH59F3211SCTAA	4 MBytes	1M X 32
LH59F3212SCTAA	8 MBytes	2 X 1M X 32
LH59F3221S5TAA	8 MBytes	2M X 32
LH59F3222S5TAA	16 MBytes	2 X 2M X 32

These Sharp FLASH SIMMs can be purchased from Marshall Industries (www.marshall.com).

4.3 Boot ROM

The Boot ROM is normally not addressable by the CL-PS7500FE. If JP12 is installed, the FLASH discussed in the preceding section will be re-mapped to 0x0100:0000, and the boot ROM will be addressed at location 0x0000:0000. The boot ROM can be either a 1MByte or 4MByte device.

Table 4: Boot ROM Enable/Disable Jumper - JP12

Jumper	*Description*
[■ ■]	FLASH located at 0x0100:0000, Boot ROM located at 0x0000:0000.
■ ■	FLASH located at 0x0000:0000, Boot ROM not addressable.

Due to the possibility of the boot ROM taking on different tasks in-system, the option of changing the size of the device has been included. The ROM may be either 1 MByte or 4 MBytes. Size is determined by JP11 and is set according to the following table. Caution – do not leave the jumper off. The jumper must be in one of these two positions or damage to the EPROM or random read errors could occur.

Table 5: Boot ROM Size Selection Jumper – JP11

Jumper	*Description*
1 2 3 [■ ■] ■	4 MBytes.
1 2 3 ■ [■ ■]	1 MByte.

4.4 Clocks

The CL-PS7500FE supports different clocks for the memory CPU and I/O subsystems. The simplest means of clocking the CL-PS7500FE is to use the same clock for all three of the above inputs. In the case of the development board, the CPU clock is run from a Chrontel CH9294G clock generator chip whose frequency is determined by JP2, while the MEMORY clock runs from Y1 at 56MHz. The I/O clock is driven by Y2, running at 64MHz. The intent of running Y2 at 64MHz is so that the divide-by-two mode of operation for the I/O subsystem can be used. Clocks are configured by programming the Clock prescalar register, located at 0x0320:003C:

Table 6: Clock Prescalar Register

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Name	X	X	X	X	X	C	M	I

Table 7: Clock Configuration Register

Bit	*Description*
X	Unspecified.
C	CPUCLK – If 1, internal CPU clock = CPUCLK input, otherwise internal CPU clock = ½ CPUCLK input.
M	MEMCLK - If 1, internal memory clock = MEMCLK input, otherwise internal memory clock = ½ MEMCLK input.
I	I_OCLK – If 1, internal I/O clock = I_OCLK input, otherwise internal I/O clock = ½ I_OCLK input.

The SnA pin on the CL-PS7500FE is used to determine the source of the CPU clock. The SnA pin is jumper configurable according to the following:

Table 8: JP1 (1x2) Definition (Control of SnA pin)

Jumper	*Description*
	SnA pulled low. The CL-PS7500FE treats the MEMCLK and CPUCLK as independent clocks and runs the memory subsystem asynchronous to the CPU.
	SnA pulled high. The CL-PS7500FE uses MEMCLK to drive both the internal memory and CPU clocks.

As stated earlier, the Chrontel clock generator Chip (CH9294G) drives the CPU clock. If the developer would prefer not to use this clock, simply remove JP1 and the 56MHz MEMCLK input will drive the CPU. Otherwise, the CPU speed is configured by JP2. JP2 actually configures both the CPUCLK and the VCLKI inputs to the CL-PS7500FE; however, the jumper fields for setting up VCLKI are discussed in Section 4.13.1.

The following figure illustrates the layout of JP2:

Figure 4: JP2 Jumper Pin Numbering

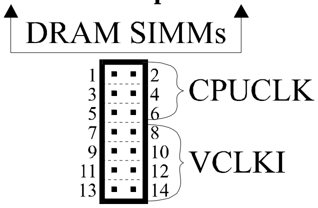
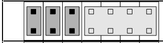
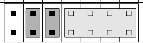
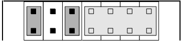
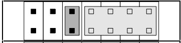
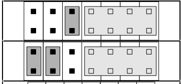
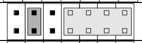
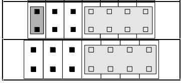
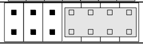


Table 9: CPU Clocking Options (if the CH9294G is used for CPU clocking)

(1-6)	*Freq. (MHz)*
	55.0
	65.0
	70.0
	80.0
	45.0
	40.0
	60.0
	50.0

Note that although clock frequencies as high as 80MHz can be specified for the CPU clock, anything above 55MHz is currently not supported.

4.5 Power Management Modes

The CL-PS7500FE supports three different modes of operation: Normal, Suspend, and Stop. The development board has only limited capability in this regard (the Normal mode is the only one fully supported).

The key missing point is that the oscillators on the board cannot be stopped under software control. If the developer's application requires this functionality, please contact Cirrus Logic regarding specific implementation details.

Although the oscillators cannot be stopped, the processor can still enter the Suspend and Stop modes.

In the Suspend mode, barring the fact that the oscillators are still running, the DMA controller still operates within the CL-PS7500FE so things like the display can continue to be refreshed and updated.

In the Stop mode, a fully functional system would stop all clocks completely, and consume only leakage current. Caution should be exercised when entering and exiting this mode as the LCD may become damaged if not properly shut down. Consult the documentation that comes with the specific LCD being connected. Specifically, the problem is that if the bias voltage to the display (supplied by the customer) is left on and the pixel clocks to the LCD are stopped, the display will burn up. This is an unfortunate problem with some LCDs.

An active low pulse on either of the nEVENT pins or on the nPOR pin of the CL-PS7500FE will cause the processor to exit the Suspend or Stop mode. One difference between the Stop and Suspend modes is that the Suspend mode may be exited via an IRQ or a FIQ. This means that an interrupt from the serial port, mouse, keyboard, or any other interrupt driven peripheral may cause the processor to transition to the Normal mode.

For full details on entering the alternate modes, please consult the CL-PS7500FE Advance Data Book, V2.0, Chapter 21.

4.6 Logic Analyzer Headers

There are eight logic analyzer headers on this design (JP3 to JP10). The headers provide the developer with direct access to many important signals on the board. The pin-outs for the headers are shown on the schematics and listed in the following tables.

Table 10: JP3 Logic Analyzer Header Definition

Description	*#*	*#*	*Description*
VCC (+5V)	1	2	VCC (+5V)
KBCLK (Keyboard shift clock)	3	4	KBDATA (Keyboard serial data)
MSDATA (Mouse port serial data)	5	6	MSCLK (Mouse port shift clock)
SDCLK (Sound port data shift clock)	7	8	SDO (Sound port data)
SCLK (Sound system clock)	9	10	WS (Sound port channel (left/right))
ECLK (Video data shift clock)	11	12	ED7 (Video data)
ED6 (Video data)	13	14	ED5 (Video data)
ED4 (Video data)	15	16	ED3 (Video data)
ED2 (Video data)	17	18	ED1 (Video data)
ED0 (Video data)	19	20	Ground

Table 11: JP4 Logic Analyzer Header Definition

Description	*#*	*#*	*Description*
VCC (+5V)	1	2	VCC (+5V)
NSIOCS1 (simple I/O chip select)	3	4	HCLK (Optional video input clock)
NSIOCS2 (simple I/O chip select)	5	6	16MHz (16MHz IO system clock)
NEASCS (ISA Bus space select)	7	8	CPUCLK (CL-PS7500FE processor clock)
OD1 (general purpose open drain I/O)	9	10	IOCLK (CL-PS7500FE I/O system clock
OD0 (general purpose open drain I/O)	11	12	IOP7 (general purpose open drain I/O)
IOP6 (general purpose open drain I/O)	13	14	IOP5 (general purpose open drain I/O)
IOP4 (general purpose open drain I/O)	15	16	IOP3 (general purpose open drain I/O)
IOP2 (general purpose open drain I/O)	17	18	IOP1 (general purpose open drain I/O)
IOP0 (general purpose open drain I/O)	19	20	Ground

Table 12: JP5 Logic Analyzer Header Definition

Description	#	#	*Description*
VCC (+5V)	1	2	VCC (+5V)
D16 (main data bus bit)	3	4	D15 (main data bus)
D14 (main data bus)	5	6	D13 (main data bus)
D12 (main data bus)	7	8	D11 (main data bus)
D10 (main data bus)	9	10	D9 (main data bus)
D8 (main data bus)	11	12	D7 (main data bus)
D6 (main data bus)	13	14	D5 (main data bus)
D4 (main data bus)	15	16	D3 (main data bus)
D2 (main data bus)	17	18	D1 (main data bus)
D0 (main data bus)	19	20	Ground

Table 13: JP6 Logic Analyzer Header Definition

Description	#	#	*Description*
VCC (+5V)	1	2	VCC (+5V)
D31 (main data bus)	3	4	D30 (main data bus)
D29 (main data bus)	5	6	D28 (main data bus)
D27 (main data bus)	7	8	D26 (main data bus)
D25 (main data bus)	9	10	D24 (main data bus)
D23 (main data bus)	11	12	D22 (main data bus)
D21 (main data bus)	13	14	D20 (main data bus)
D19 (main data bus)	15	16	D18 (main data bus)
D17 (main data bus)	17	18	READY (CL-PS7500FE Ready line)
nROMCS (FLASH/ROM chip select)	19	20	Ground

Table 14: JP7 Logic Analyzer Header Definition

Description	#	#	*Description*
VCC (+5V)	1	2	VCC (+5V)
nCCS (PC Combo Chip Select)	3	4	BD15 (16-bit I/O data)
BD14 (16-bit I/O data)	5	6	BD13 (16-bit I/O data)
BD12 (16-bit I/O data)	7	8	BD11 (16-bit I/O data)
BD10 (16-bit I/O data)	9	10	BD9 (16-bit I/O data)
BD8 (16-bit I/O data)	11	12	BD7 (16-bit I/O data)
BD6 (16-bit I/O data)	13	14	BD5 (16-bit I/O data)
BD4 (16-bit I/O data)	15	16	BD3 (16-bit I/O data)
BD2 (16-bit I/O data)	17	18	BD1 (16-bit I/O data)
BD0 (16-bit I/O data)	19	20	Ground

Table 15: JP8 Logic Analyzer Header Definition

Description	#	#	*Description*
VCC (+5V)	1	2	VCC (+5V)
LA28 (main address bus)	3	4	LA27 (main address bus)
LA26 (main address bus)	5	6	LA25 (main address bus)
LA24 (main address bus)	7	8	LA23 (main address bus)
LA22 (main address bus)	9	10	LA21 (main address bus)
LA20 (main address bus)	11	12	LA19 (main address bus)
LA18 (main address bus)	13	14	LA17 (main address bus)
LA16 (main address bus)	15	16	LA15 (main address bus)
LA14 (main address bus)	17	18	LA13 (main address bus)
LA12 (main address bus)	19	20	Ground

Table 16: JP9 Logic Analyzer Header Definition

Description	*#*	*#*	*Description*
VCC (+5V)	1	2	VCC (+5V)
LA10 (main address bus)	3	4	LA11 (main address bus)
LA8 (main address bus)	5	6	LA9 (main address bus)
LA6 (main address bus)	7	8	LA7 (main address bus)
LA4 (main address bus)	9	10	LA5 (main address bus)
LA2 (main address bus)	11	12	LA3 (main address bus)
LA0 (main address bus)	13	14	LA1 (main address bus)
nRAS2 (DRAM Row strobe)	15	16	nRAS3 (DRAM Row strobe)
nRAS0 (DRAM Row strobe)	17	18	nRAS1 (DRAM Row strobe)
nIOW (I/O and FLASH write strobe)	19	20	Ground

Table 17: JP10 Logic Analyzer Header Definition

Description	*#*	*#*	*Description*
VCC (+5V)	1	2	VCC (+5V)
NIOR (I/O and FLASH read strobe)	3	4	nCAS3 (DRAM Column strobe)
NCAS2 (DRAM Column strobe)	5	6	nCAS1 (DRAM Column strobe)
NCAS0 (DRAM Column strobe)	7	8	RA11 (DRAM address bus)
RA10 (DRAM address bus)	9	10	RA9 (DRAM address bus)
RA8 (DRAM address bus)	11	12	RA7 (DRAM address bus)
RA6 (DRAM address bus)	13	14	RA5 (DRAM address bus)
RA4 (DRAM address bus)	15	16	RA3 (DRAM address bus)
RA2 (DRAM address bus)	17	18	RA1 (DRAM address bus)
RA0 (DRAM address bus)	19	20	Ground

4.7 Keyboard and Mouse

The keyboard and mouse ports in the CL-PS7500FE are identical. Each is intended for connection to a PS/2 style mouse or keyboard.

The mouse port consists of MSCLK (CL-PS7500FE pin 165) and MSDATA (CL-PS7500FE pin 167), which are brought out to J3. The keyboard port consists of KBCLK (CL-PS7500FE pin 168) and KBDATA (CL-PS7500FE pin 169), which are brought out to J2.

From a programmer's perspective, each interface consists of the following registers:

Table 18: Keyboard and Mouse Registers

Name	*Location*	*Description*
KBDCR	0x0320:0008	Keyboard control/status register.
KBDAT	0x0320:0004	Keyboard data read/write register.
MSECR	0x0320:00AC	Mouse control/status register.
MSEDAT	0x0320:00A8	Mouse data read/write register.

Each data register is used both for reading and writing information to and from each of the two interfaces.

Each of the above control/status registers has a different bit definition for read vs. write. The following tables outline the differences.

Table 19: PS2 Control Register Write Bit Mapping

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Name	X	X	X	X	ENA	X	DAT	CLK

Table 20: PS2 Control Register Write Bit Definition

Bit	*Description*
X	Unspecified.
ENA	Enable line. Set this pin high to enable the PS2 interface being used. Set low to disable.
DAT	Data line. Setting this bit to 1 forces the data pin of the specified PS2 interface low. Setting this bit to 0 floats the data pin. Assuming a pull-up is on this open-drain pin, the data pin will float high.
CLK	Clock line. Setting this bit to 1 forces the clock pin of the specified PS2 interface low. Setting this bit to 0 floats the clock pin. Assuming a pull-up is on this open-drain pin, the clock pin will float high.

Table 21: PS2 Control Register Read Bit Mapping

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Name	TXE	TXB	RXF	RXB	ENA	RXP	DAT	CLK

Table 22: PS2 Control Register Write Bit Definition

Bit	*Description*
TXE	Transmit shift register status. If 0, the transmit shift register is busy. If 1, the transmit shift register is ready to accept data.
TXB	Transmitter busy bit. If 0, the transmitter is NOT busy. If 1, the transmitter is busy.
RXF	Receive shift register status bit. If 0, the receive shift register is NOT full. If 1, the receive shift register is full.
RXB	Receiver busy bit. If 0, the receiver is NOT busy. If 1, the receiver is busy.
ENA	Enable status. If 1, the associated PS2 interface is enabled. If 0, the interface is disabled.
RXP	Received data parity bit. This bit is an odd parity bit associated with the last received byte.
DAT	Status of the data pin of the selected PS2 interface. If the pin is high, this bit will be 1, otherwise it will be 0.
CLK	Status of the clock pin of the selected PS2 interface. If the pin is high, this bit will be 1, otherwise it will be 0.

In addition to the above control and status register definitions, each register has the following data registers:

The keyboard interface may be polled or interrupt driven. Both "transmit register empty" and "receive register full" interrupts are available. These are found in the IRQB register set. The locations and bits associated with IRQB are shown in the following table:

Table 23: Keyboard Interrupt Control

Register	*Location*	*Description*	*Bits used*
IRQSTB	0x0320:0020	IRQB interrupt status (enabled/disabled).	7 = keyboard receive interrupt. 6 = keyboard transmit interrupt.
IRQRQB	0x0320:0024	IRQB interrupt requests	7 = keyboard receive interrupt. 6 = keyboard transmit interrupt.
IRQMSKB	0x0320:0028	IRQB interrupt enable mask.	7 = keyboard receive interrupt. 6 = keyboard transmit interrupt.

The mouse interface may be polled or interrupt driven in exactly the same fashion as the keyboard interface. Both "transmit register empty" and "receive register full" interrupts are available. These are found in the IRQD register set. The locations and bits associated with IRQD are shown in the following table:

Table 24: Mouse Interrupt Control

Register	*Location*	*Description*	*Bits used*
IRQSTD	0x0320:0070	IRQD interrupt status (enabled/disabled).	1 = mouse transmit interrupt. 0 = mouse receive interrupt.
IRQRQD	0x0320:0074	IRQD interrupt requests	1 = mouse transmit interrupt. 0 = mouse receive interrupt.
IRQMSKD	0x0320:0078	IRQD interrupt enable mask.	1 = mouse transmit interrupt. 0 = mouse receive interrupt.

4.8 Sound

4.8.1 Physical Details

The sound system in the CL-PS7500FE consists of a serial interface intended for driving a 16-bit stereo DAC. In the case of the development board, the sound system pins are connected to a Crystal Semiconductor CS4333 16-bit stereo DAC. This DAC outputs data to the headset port, J1.

The serial arrangement in the sound system consists of four pins, the functions of which are summarized in the table below:

Table 25: Sound System External Interface

Name	*Pin #*	*Description*
SDO	68	Serial Data. This line shifts out the serial data in 32-bit words, 16-bits for the left and 16-bits for the right channel. Values on the data pin change on the falling edge of SDCLK.
SDCLK	70	Serial Data Shift Clock. This is the synchronous serial clock for shifting out data. Data is shifted out MSB first.
WS	71	Word Select. This pin is high during the transmission of the left channel data, low otherwise. The state of the pin changes on the MSB of the left channel data. This is connected to the LRCK or left/right clock of the CS4333
SCLK	69	This is the master clock used by the DAC.

In the table above, the "Japanese" mode of operation of the sound macrocell is assumed. If the developer is interested in the alternate mode (Normal) please refer to Chapter 13 of the CL-PS7500FE Advance Data Book V2.0. This mode is not used because the default mode of the CS4333 is not compatible with it.

The sound portion of the video macrocell in the CL-PS7500FE supports operating the sound system clock either off of a clock derived from the I/O clock, or off of an external clock. Since the desired sample rate of the CS4333 is 44.1kHz, an external clock is required. This clock must be either 256, 384 or 512 times the desired sample rate. In this case, 256x was the chosen value, so the external clock for the sound subsystem is 11.2896MHz. This clock is an input to the SCLK pin of the CL-PS7500FE, as well as the MCLK pin of the CS4333.

In the case of the CS4333, the DAC will be operated in the externally clocked mode. This is the only mode available, as the CL-PS7500FE will not accept a clock on the SDCLK line. The CS4333's de-emphasis filter will also be unavailable, as it is not supported when running the DAC in externally clocked mode.

The CS4333 has a register that is programmed by setting the serial data line high, using the left/right clock as a shift clock, and shifting data in via the serial data clock line. The CL-PS7500FE does not support this type of communication, so the CS4333 must be operated in its default mode. The setting is as follows:

- 16 bits per channel, meaning at least 32 shift clocks per combined left and right sample.
- Right justified. That is, the left/right clock changes on the falling edge of SCLK on the least significant data bit.
- Data is latched into the CS4333 on the rising edge of SDCLK. The CL-PS7500FE changes data on the falling edge of SDCLK, so set-up/hold times are comfortably met.
- The left/right clock (LRCK) is defined as high on the left channel, low on the right channel. The only way to support this is to use the "Japanese" mode of the sound macrocell.

4.8.2 Programming Details

Programming the following values into these registers, configures the sound macrocell in the CL-PS7500FE:

Table 26: Sound Macrocell Configuration Parameters

Register	*Location*	*Value*	*Description*
SFR	0x0340:0000	0xB000:0002	Sound Frequency Register
SCTL	0x0340:0000	0xB100:0002	Sound Control Register
VIDMUX	0x0320:006C	VIDMUX[1]=1	Video LCD and serial sound mux control. This sets Japanese Mode.

In addition, there are two sets of DMA pointers to ensure the DAC never runs out of data. This permits software to fill one block of the sound page in memory while the DMA controller is reading data from another. All sound data must be written into a 4Kbyte page in the CL-PS7500FE's memory. DMA data is transferred in blocks of four 32-bit words, or Qwords; the 4Kbyte block of memory must be Qword aligned. To configure the DMA pointer registers, write the following:

Table 27: Sound System DMA and Interrupt Registers

Register	*Location*	*Description*
SDCURA	0x0320:0180	Sound DMA Current Address Frequency Register, page A.
SDENDA	0x0320:0184	Last Qword location in sound page A.
SDCURB	0x0320:0188	Sound DMA Current Address Frequency Register, page B.
SDENDB	0x0320:018C	Last Qword location in sound page B.

Specific values are not listed in the above table, as the location of the sound data pages in memory is entirely up to the programmer. For information on the pointer registers, please consult the CL-PS7500FE Advance Data Book V2.0, pages 99-101.

In addition to the above address pointer configuration, there are control and status registers that must be configured:

Table 28: SDCR – Sound DMA Control Register Configuration (0x0320:0190)

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Name	CLR	X	ENA	1	X	X	X	X

Table 29: SDCR Register Write Bit Definition

Bit	*Description*
CLR	Clear bit. Setting this bit resets the sound state machine. Writing a 0 to this bit won't explicitly clear the state machine.
ENA	Writing a 0 to this bit shuts down the sound system, writing a 1 enables it.
Bit 4	This bit is not relevant when writing to the control register, but when reading, it should be noted that the bit is always 1.

The sound DMA status register is read only, and contains only 3 bits.

Table 30: SDST – Sound DMA Status Register Configuration (0x0320:0190)

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Name	X	X	X	X	X	O	I	W

Table 31: SDCR Register Write Bit Definition

Bit	*Description*
O	Overrun bit. This bit is set if the DMA state machine has been overrun.
I	Interrupt bit. If 1, this bit indicates that a DMA interrupt request is pending.
W	Which buffer? This bit indicates 0 for buffer A, 1 for buffer B as the buffer currently being accessed by the DMA controller.

For further details on the programming the sound system, please consult the CL-PS7500FE Advance Data Book V2.0, sections 9.3.3-4, 10.3.28, 10.3.51-56, 13, 14.1.7, 16.29-30, 21.1.1, and 22.8.

4.9 Event Switches

There are three key-switches on the development board, S1-S3. These switches are connected to the following subsystems on the board:

Table 32: Event Switch Functions

Switch	*Function(s)*
S1	nEVENT1 on the CL-PS7500FE. Asserts the nEVENT1 interrupt in the CL-PS7500FE. If the CL-PS7500FE is in Stop or Suspend mode, pressing this switch will wake up the processor.
S2	nEVENT2 on the CL-PS7500FE. Asserts the nEVENT2 interrupt in the CL-PS7500FE. If the CL-PS7500FE is in Stop or Suspend mode, pressing this switch will wake up the processor.
S3	nPOR. This switch is connected to the power on reset circuitry of the CL-PS7500FE. Pressing this switch will reset the entire system.

For details on the event pins and how they are used to exit Stop and Suspend modes, please refer to Section 4.5 of this document, and the CL-PS7500FE Advance Data Book V2.0, Chapter 21.

4.10 ISA Interface

The ISA interface on the board is fully buffered and will support most ISA bus transactions. Key differences between a PC ISA bus and the development board ISA bus are discussed here.

4.10.1 CL-PS7500FE Memory Mapping

Whereas the ISA bus in a PC has I/O mapped registers at the bottom of the x86's memory map, the ISA bus is mapped into a section of memory of the CL-PS7500FE. Although the organization of ISA address space was hinted at in Figure 3, further details on decoding are given here.

Table 33: ISA Map to CS-PS7500FE Memory Space

Address Range	*Description*
0x0800:0000..0x0BFF:FFFF	ISA memory mapped device access. Devices requiring memory read and write strobes are addressed in this range.
0x0C00:0000..0x0DFF:FFFF	ISA I/O devices are addressed in this range. During I/O access to this range, the AEN line of the ISA bus is asserted.
0x0E00:0000..0x0FFF:FFFF	Addressing this portion of ISA space will produce a DMA Acknowledge as well as the appropriate I/O strobe. CL-PS7500FE address bits LA[4:2] decode to the specific DMA channel being decoded. For an example of how this decoding works, see Table 35.
0x0Enn:nn2n..0x0Fnn:nn3n	Note that this space is a subset of the above space. Accessing addresses that have LA[7:5] = [0:0:1] will assert the TC pin to the ISA bus as well. This is part of the DMA Acknowledge space. For an example of how this decoding works, see Table 35.

Note that these address spaces rely on address bits that normally do not decode onto the ISA bus (any address > LA25). Note also that LA25 translates to A23 on the ISA bus. Memory mapped devices should therefore be configured to decode at an address with A23 set to 0.

The ISA data bus is only 16-bits wide and an important point must be observed about communicating with this space. All accesses to this space must be treated as 32-bit, even though only 16-bits of data are being used. This data is read from and written to the lower 16-bits of each 32-bit word.

4.10.2 Exception Support

Exceptions from the ISA bus come in the form of interrupts and DMA requests. Since both types of exception are quite similar, and the CL-PS7500FE does not directly support DMA requests and acknowledges from the ISA bus subsystem, DMA requests and interrupts are treated as a common form of exception. When a DMA request is made, the CL-PS7500FE will receive an interrupt request on IRQ1. When an ISA device asserts an interrupt request, the CL-PS7500FE will receive an interrupt on IRQ7.

Determining which device asserted the DMA/IRQ is a matter of reading a register. The register is located at any address in the range 0x0302:B700...0x0302:B7FF. The register is 16-bits wide and is read only. The definition of each bit is as follows:

Table 34: ISA Exception Register Bit Definition

Bit #	*D15*	*D14*	*D13*	*D12*	*D11*	*D10*	*D9*	*D8*
Exception:	DRQ7	DRQ6	DRQ5	DRQ3	DRQ2	DRQ1	DRQ0	IRQ15
Bit #	*D7*	*D6*	*D5*	*D4*	*D3*	*D2*	*D1*	*D0*
Exception:	IRQ14	IRQ11	IRQ10	IRQ9	IRQ7	IRQ5	IRQ4	IRQ3

As noted in the previous section, DMA requests are acknowledged by writing to portions of the address space reserved for this activity. Table 35 below illustrates how the DMA acknowledge "space" is addressed to acknowledge the desired DMA.

Table 35: DMA Acknowledge Memory Space

DACK	*Address*	*Address (w. TC)*
0	0x0E00:0000	0x0E00:0020
1	0x0E00:0004	0x0E00:0024
2	0x0E00:0008	0x0E00:0028
3	0x0E00:000C	0x0E00:002C
5	0x0E00:0010	0x0E00:0030
6	0x0E00:0014	0x0E00:0034
7	0x0E00:0018	0x0E00:0038

4.11 SMSC 37C665 PC Combo Chip

The PC Combo chip on this board is an SMSC product model 37C665, and is common to many PC motherboards. Although the Combo chip supports connections to IDE and floppy drives, these have not been implemented on the development board. The only portions of the combo chip that have been implemented are the two serial ports and the one parallel port. Each of these three interfaces is discussed in detail in the following sections.

Prior to attempting to use the ports available, the 37C665 must be configured. The device has two addresses listed in the following table.

Table 36: PC Combo Chip Set-Up Registers

Name	*Address*	*Description*
CFGA	0x0301:0FC0	Chip Configuration register A. Write internal PC combo chip configuration register addresses here.
CFGB	0x0301:0FC4	Chip Configuration register B. Read and write configuration register data here.

The 37C665 has an IDE interface and floppy disk interface and both are enabled by default, so some minor programming of the configuration registers is required. Note that the programming of configuration registers as discussed here is not exhaustive. Developers who want to move devices around in the CL-PS7500FE's address space, or enable/disable different devices should consult the 37C665 specification.

When configuring the 37C665, first the device must first be placed in configuration mode. To do this, write the following sequence to CFGA: 0x55 0x55. These two writes must happen in immediate succession and there must not be any bus cycle activity between the writes. For this reason, it is highly recommended that interrupts be disabled prior to writing this sequence.

There are a total of 16 internal configuration registers in the 37C665. Data written to CFGA is the index of the configuration register to be read or written, with the actual register being read or written via CFGB.

The following table outlines the write sequence and what is being set up on the combo chip:

Table 37: PC Combo Chip Configuration Register Default Values

CFGA	*CFGB*	*Description*
0x00	0x20	This value will disable both the IDE and floppy drive interfaces within the combo chip. It will also set the serial port baud rate generator to be active only when the PWRGD pin is set high. This is the case in normal operation. If the pin is pulled low for any reason, the baud rate generator for the serial ports will be shut down.
0x01	0x9F	This is actually the default value of this register at power-up. The value is discussed here to explain what is being set up. This register sets both serial port interrupts and the parallel port interrupt to be active high. The register is also used to configure the decoding addresses of the parallel port and COM 3/4. COM 3 and 4 are not implemented in this design, but the corresponding bits for their decode should be left at their default values. The following sections of this document list addresses where each port may be found. The default decoding is assumed. The parallel port is configured for normal mode. For extended modes, please consult the 37C665 data sheet. Also, the most significant bit of this register is used to lock out accesses to the registers. By default, the bit is high, meaning unlocked. If this bit is set low, there will be no way to regain access to any of the configuration registers without a hard reset or power down.
0x02	0xDC	This register enables and sets the base addresses of the two serial ports on the development board. The value listed at left is actually the default, but its meaning is discussed here. The value shown puts the serial port addresses at the locations shown in the following sections on serial ports. The ports are also enabled and powered up under the control of this register. The default is to have both ports on.
0x03	0x78	This register controls the operation of the floppy drive. Since this part of the chip should be powered down, the register should either be programmed with the default value at left, or left alone.
0x04	0x00	This register controls the parallel port mode and the serial port clocking. The default is correct for standard operation of the serial ports. Developers have the option of setting the serial ports to run at MIDI data rates and to run the parallel port in one of its extended modes.

Although there are 16 registers, those not listed are either related to the floppy and IDE drive interfaces, or are reserved.

After the configuration registers have been programmed, the 37C665 is taken out of configuration mode by writing 0xAA to the CFGA register. Be sure to do this prior to attempting to communicate with any of the ports.

4.11.1 Serial Port #1 (COM1)

COM1 is a standard 16C550A-compatible UART. The maximum data rate of this port is 115,200 baud. Port registers are addressed in the CL-PS7500FE's memory at the following locations:

Table 38: COM1 Address Registers

Name	*Location*	*Description*
DLAB (LSB)	0x0301:0FE0	Baud Rate Divisor Latch LSB
DLAB (MSB)	0x0301:0FE4	Baud Rate Divisor Latch MSB
RBR/THR	0x0301:0FE0	Receive Buffer Register / Transmitter Holding Register
IER	0x0301:0FE4	Interrupt Enable Register
IIR	0x0301:0FE8	Interrupt Identification Register
LCR	0x0301:0FEC	Line Control Register
MCR	0x0301:0FF0	Modem Control Register
LSR	0x0301:0FF4	Line Status Register
MSR	0x0301:0FF8	Modem Status Register
SR	0x0301:0FFC	Scratch Register

COM1 uses Interrupt Request 6 on the CL-PS7500FE.

4.11.2 Serial Port #2 (COM2)

COM2 is also a standard 16C550A-compatible UART. The maximum data rate of this port is 115,200 baud. Port registers are addressed in the CL-PS7500FE's memory at the following locations:

Table 39: COM2 Address Registers

Name	*Location*	*Description*
DLAB (LSB)	0x0301:0BE0	Baud Rate Divisor Latch LSB
DLAB (MSB)	0x0301:0BE4	Baud Rate Divisor Latch MSB
RBR/THR	0x0301:0BE0	Receive Buffer Register / Transmitter Holding Register
IER	0x0301:0BE4	Interrupt Enable Register
IIR	0x0301:0BE8	Interrupt Identification Register
LCR	0x0301:0BEC	Line Control Register
MCR	0x0301:0BF0	Modem Control Register
LSR	0x0301:0BF4	Line Status Register
MSR	0x0301:0BF8	Modem Status Register
SR	0x0301:0BFC	Scratch Register

COM2 uses Interrupt Request 9 on the CL-PS7500FE.

4.11.3 IrDA Support

In addition to being a standard serial port, COM2 can be optionally connected to the on-board IrDA transceiver.

JP13 is used to control whether COM2 is connected to J14 on the development board, or is connected to the IrDA transceiver. The function of JP13 is shown in the following table:

Table 40: JP13 (1x2) Definition (Control of Serial Port #2)

Jumper	*Description*
▣	Serial Port #2 is wired to J14 and functions as a standard serial port.
▪ ▪	Serial Port #2 is connected to the IrDA transceiver and is able to interface to IrDA compliant equipment.

The components used to form the IrDA interface on the development board are the TOIM3232 and the TFDS4500, both manufactured by Temic Semiconductors.

The TFDS4500 is the most recognizable component, as it contains the Ir diodes used for transmitting and receiving data (U26). This device has a theoretical maximum data rate of 4Mbaud. However because the TDFS4500 is connected to COM2, which has a maximum data rate of 115,200 baud, the maximum data rate is thus 115,200 baud.

The TOIM3232 is a device used to adjust outgoing serial port pulses to the correct width for the TFDS4500. The maximum data rate of the TOIM3232 is 115,200 baud. The TOIM3232 expects to receive data from the serial port in 8 data bits, no parity, 1 stop bit format.

Once the jumper above has been set to configure COM2 for infrared communication, the TOIM3232 must be programmed with the baud rate at which the serial port will be operating. To do this, perform the following steps:

1) Reset the TOIM3232 by asserting and then de-asserting the DTR signal. After the signal has been de-asserted, wait for at least 7μs.

2) Put the TOIM3232 into programming mode by asserting the RTS signal. After the signal has been asserted, wait for at least 7µs.

3) Send the control byte to the TOIM3232. The serial port must be configured for 9600 baud as this is the rate at which the TOIM3232 operates after being reset. Sending 0x1Z will change the baud rate of the TOIM3232 to the desired rate (where Z = 0 for 115,200 baud, Z = 1 for 57,600 baud, Z = 2 for 38,400 baud…see the TOIM3232 data sheet for a complete list). After the control byte has been sent, wait for at least 1µs.

4) Take the TOIM3232 out of programming mode by de-asserting the RTS signal.

All further data sent to and received from the TOIM3232 will be at the baud rate specified in step 3. The serial port must therefore be reconfigured for the required baud rate.

After the TOIM3232 has been configured with the correct baud rate, from a software perspective the IrDA port can be treated as if it were a standard COM port.

4.11.4 Parallel Port (LPT)

The parallel port used in this design is a standard PC parallel port with its address space mapped as follows:

Table 41: LPT Register Definitions

Name	*Address*	*Description*
PDOUT	0x0301:09E0	Parallel Data Output/Input Register
PSTAT	0x0301:09E4	Parallel Status Register
PCON	0x0301:09E8	Parallel Control Register
PEPPA	0x0301:09EC	Parallel EPP address Port
PEPPD0	0x0301:09F0	Parallel EPP mode data port 0
PEPPD1	0x0301:09F4	Parallel EPP mode data port 1
PEPPD2	0x0301:09F8	Parallel EPP mode data port 2
PEPPD3	0x0301:09FC	Parallel EPP mode data port 3

In addition to the above registers, the LPT port uses interrupt request 2 on the CL-PS7500FE.

This port is capable of supporting normal, ECP, EPP and PS2 type operation.

4.12 CS8900A Ethernet (10BASE-T)

The Ethernet controller on the development board is a Cirrus Logic CS8900A 10BASE-T controller. The device has an ISA compatible interface, and most of the capability of the interface has been implemented on this design. Notably, the device supports an ISA style I/O mapped set of registers and DMA capability. Both of these interfaces are available on the development board.

Note: A detailed discussion of how data is transferred to and from the Ethernet port is beyond the scope of this document. Developers wishing to know more about this should consult the CS8900A data sheet. Also note that although the I/O mapped registers are outlined, the internal register set of the CS8900A (PacketPage register set) is not discussed in thorough detail in this document. Again, the developer is directed to the CS8900A data sheet for full details on the inner workings of this part.

4.12.1 I/O Mapped Interface

The I/O mapped interface on the CS8900A is a set of eight registers, each 16-bits wide. Definitions of each register are given here, but for a more thorough explanation, please consult the CS8900A data sheet. I/O registers have a default mapping into the CL-PS7500FE's address space as follows:

Table 42: CS8900A I/O Mapped Registers

Name	*Address*	*R/W*	*Description*
RTDAT0	0x0324:0C00	R/W	Receive/Transmit data (Port 0). Port holds transmitted and received data in 16-bit words.
RTDAT1	0x0324:0C08	R/W	Receive/Transmit data (Port 1). Port holds most significant half-word of data in 32-bit mode. The CL-PS7500FE treats this interface as strictly 16-bit, so don't use this register.
TxCMD	0x0324:0C10	W	Transmit command. This register is written to indicate data is to be transmitted, and how it is to be transmitted. This register controls how many bytes must be in the CS8900A buffer before transmission begins, whether CRC data is to be sent with the transmission, tolerance for data collisions, whether data currently in the buffer memory should be flushed, and whether data should be padded.
TxLength	0x0324:0C18	W	Transmit length. This register is written immediately after the transmit command. The length of the frame to be transmitted, in bytes, is written to this register.
ISQ	0x0324:0C20	R	Interrupt Status Queue. See the section on exceptions.
PPPTR	0x0324:0C28	R/W	PacketPage Pointer. The CS8900A has an internal set of registers called a PacketPage. The PacketPage pointer points to the current address in the page. Writing to this location also optionally sets up an auto-incrementing mode where each read or write of the PacketPage registers increments the pointer.
PDAT0	0x0324:0C30	R/W	PacketPage Data (Port 0). Reading or writing this register will access the internal PacketPage register pointed to by the PPPTR register. This register is used for both 16-bit and 32-bit data transfers. As mentioned above, the interface to the CL-PS7500FE is 16-bit only and as such, this register should be the only one used when addressing the CS8900A's internal registers.
PDAT1	0x0324:0C38	R/W	PacketPage Data (Port 1). This register is only used when communicating with the CS8900A in 32-bit mode, and is not used on the development board.

When reading through the CS8900A data sheet, location 0x0300 is listed as the default base address. Although these registers can be moved around, due to the connection scheme to the CL-PS7500FE, a base address of 0x0300 is the only valid setting.

4.12.2 Memory Mapped Interface

The CL-PS7500FE development board does not support the memory-mapped interface of the CS8900A.

4.12.3 Exceptions from the CS8900A

The CS8900A is capable of producing interrupt requests on any one of its four IRQ lines. IRQ0 from the CS8900A is the only interrupt connected to the CL-PS7500FE (INT4). All programming of internal registers on the CS8900A should reflect this.

When an event requiring interrupt service occurs, the CS8900A will assert the appropriate interrupt to the CL-PS7500FE and place a 16-bit value in the ISQ register. The CL-PS7500FE is then able to read this value and perform the appropriate action. This register is similar to a FIFO in that the CS8900A may write several interrupt events to this queue. Interrupts are read from the queue in an order of priority, not occurrence. The CL-PS7500FE should continue to read this register until the contents are set to zero by the CS8900A. This signals that all interrupts have been serviced.

The CS8900A is also capable of using any one of three possible DMA channels to off-load some of the overhead associated with data transfer. DRQ0 from the CS8900A is the only channel connected to the CL-PS7500FE. The DMA channel is only used for receiving data. Since the CL-PS7500FE does not possess an external set of DMA request/acknowledge lines, the DMA request line is connected to INT5. To acknowledge requests, a write or read must be performed to 0x0301:2000. The CL-PS7500FE will decode an external strobe to this space that is used as an acknowledge.

4.12.4 Local EEPROM

The on-board serial EEPROM is used to automatically configure the CS8900A on power-up or reset. The EEPROM used on the development board contains 1024 bits of non-volatile storage space divided into 64 16-bit words. The EEPROM used on the development board is a Fairchild NM93C46, a non-sequential device.

The EEPROM can be programmed through the CS8900A registers directly, making it possible to re-configure in-system. The EEPROM used supports 7 different operations, each of which can be initiated via writes to the internal register set of the CS8900A. Each step in the process of programming the EEPROM requires several writes to the CS8900A. Programming steps are outlined below:

1. Verify that the SI-Busy bit is clear:

- Write PPTR = 0x0136
- Read PDAT0 and check bit 8 for set/clear status. If set, wait until clear.

The above sequence must be performed before attempting to issue any EEPROM commands to the CS8900A.

2. Erase/write enable command:

- Write PPPTR = 0x0000:0040
- Write PDAT0 = 0x0000:0030

The above sequence must only be issued once at the beginning of programming the EEPROM.

3. Load configuration data into the EEPROM:

- Write PPPTR = 0x0000:0042 (this step only needs to be done once)
- Write PDAT0 = 0x0000:XXXX (this step should be repeated for each word of data to be written)

The above configuration data makes up the data being burned into the EEPROM and will, of course, be different each time data is to be written.

4. Write command:

- Write PPPTR = 0x0000:0040
- Write PDAT0 = 0x0000:00XX

In the above command sequence, the last byte represents the internal serial EEPROM address being written.

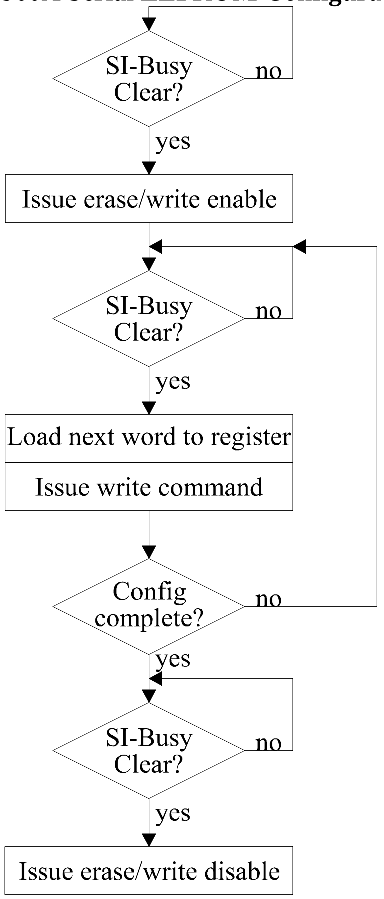
5. Erase/write disable command

- Write PPPTR = 0x0000:0040
- Write PDAT0 = 0x0000:0000

The above command is the last step in the programming process and is only issued once.

The figure below illustrates the procedure for writing data to the EEPROM:

Figure 5: CS8900A Serial EEPROM Configuration Sequence



Following the above flow chart, and noting that the "issue write command" step involves setting the address in the EEPROM being written to, write the following sequence to the EEPROM:

Table 43: EEPROM Data for Default Ethernet Operation

Data	*Write cmd*	*Description*
0x0000:A112	0x0000:0100	Configuration Word. Contains flag indicating presence of EEPROM, plus the number of words in configuration sequence.
0x0000:2020	0x0000:0101	I/O interface info
0x0000:0300	0x0000:0102	I/O base address
0x0000:0000	0x0000:0103	Interrupt
0x0000:0000	0x0000:0104	DMA channel
0x0000:2158	0x0000:0105	Individual address data
0x0000:0010	0x0000:0106	Ethernet address half word #0 (fake value shown here)
0x0000:0000	0x0000:0107	Ethernet address half word #1 (fake value shown here)
0x0000:0000	0x0000:0108	Ethernet address half word #2 (fake value shown here)
0x0000:8100	0x0000:0109	Checksum: 2's complement of the sum of the all bytes of data written, ignoring carries. When added to the sum, result should be zero.
0x0000:FFFF	0x0000:010A	End of data flag

In the above table, the individual address data is the unique IEEE Ethernet address assigned to the specific development board. The EEPROM on the development board has been pre-burned with a unique IEEE Ethernet address assigned by Cirrus Logic.

4.13 Video Subsystem

The video subsystem on the development board is partly built into the CL-PS7500FE and partly made up of external components. The following sections detail each of the operating modes of the video interface. Since there are so many different interfaces, each of which can be connected to different devices, this text is not to be considered exhaustive.

The video controller within the CL-PS7500FE can be configured to operate as a direct VGA controller, or as an LCD controller.

All internal CL-PS7500FE video registers are addressed at any address in the range of 0x0340:0000 to 0x034F:FFFF. Strictly speaking, there is only one video register, and it is shadowed at each word-aligned address throughout the aforementioned video address range. For developers, there will only be one pointer to the video page, and no offsets.

In reality, there are 296 individual registers. Before panicking, realize that 256 of them are palette registers, leaving only 40 to really be worried about. The registers are generally all 32-bits wide, and are addressed by using the most significant byte of each data word as an index into the internal video macrocell registers. The following diagram perhaps better explains this:

Figure 6: Video Data Word Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	A	A	A	A/D	A/D	A/D	A/D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

In the above diagram, the upper 4 bits are labeled A to represent that in the case of all writes to the video registers, the upper four bits of the data word represent an index address. The next four bits are shown as either A or D to indicate that in some cases these bits represent part of the index address, and in some cases they represent data. The remaining bits are always data, but in many cases, bits [15:8] are 0.

An example of how this works is the video palette register. This register consists of 28 bits of actual palette data, with the upper four bits representing the address of the palette (D[31:28] = 0 in this case). Each write to the video register with the upper four bits set to 0 will write a value into the video palette. This register is unique in that it is auto-incrementing, but that issue is best addressed by referring to the CL-PS7500FE Advance Data Book V2.0.

4.13.1 Video System Clock Options

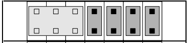
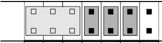
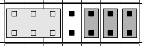
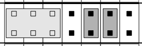
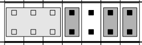
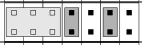
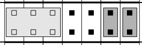
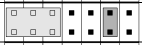
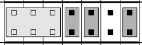
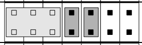
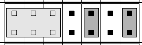
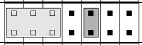
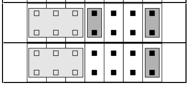
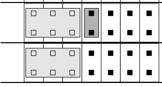
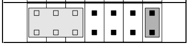
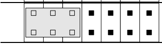
The importance of selecting the proper clock and setting the horizontal and vertical sync rates of the video system cannot be emphasized enough. Incorrect settings can and will damage video equipment connected to these interfaces. Most commercial video driver boards also have this capability, although operating such products outside of spec. is not easily done. This is largely because the developers of those products have taken the time to insulate the end-user from potentially damaging clock rates.

First, the video system within the CL-PS7500FE can be driven from three different clock sources:

- I_OCLK: 32MHz, supplied internally.
- HCLK: Supplied by the Chrontel CH7003 NTSC chip.
- VCLKI: Supplied from the Chrontel CH9294G, and discussed presently.

The jumper JP2 determines VCLKI. JP2 was discussed in Section 4.4. In that section, only the top 3 jumpers of JP2 were discussed, as the rest of the header was not relevant to determining the CPU clock. JP2 is a 2x7-header post. The remaining jumper positions control the frequency of VCLKI. Note that although the CL-PS7500FE Advance Data Book discusses building a PLL circuit at length, the development board uses a clock generator IC for generating different video clocks. The following table outlines the available frequencies:

Table 44: VCLKI Available Frequencies (via JP2)

(7-14)	Freq. (MHz)	(7-14)	Freq. (MHz)
	25.18		130.0*
	28.32		120.0**
	40.0		80.0
	72.0		31.5
	50.0		110.0
	77.0		65.0
	36.0		75.0
	44.9		94.5

* DO NOT ATTEMPT TO RUN THE CL-PS7500FE IN THIS SETTING.

** THIS IS THE MAXIMUM VIDEO CLOCK FREQUENCY OF THE CL-PS7500FE.

Refer to Figure 4 for orientation of the jumper field.

In addition to the above available clock options, each of the clocks listed may be divided by an integer between 1 and 8. The clock source is selected via the CONREG register in the video page. As different developers will have wildly differing requirements, please consult the CL-PS7500FE Advance Data Book V2.0, Section 16.27 for the details of programming this register. Note also that if using the video macrocell, the CONREG register must be programmed first.

4.13.2 VGA

The CL-PS7500FE has the ability to directly drive a VGA monitor. The controller supports multiple resolutions and display formats. A standard video set-up of a 640x480, 256-color display is presented here. In order to keep the discussion simple the color scheme presented will be a simple grey scale.

Since this discussion is of limited scope, the border of the display will be assumed to be black, and the hardware cursor will not be configured.

If the developer wishes to operate the VGA controller in a different mode, please consult the CL-PS7500FE Advance Data Book V2.0, Chapter 16 for details on programming the video macrocell.

4.13.2.1 Setting the Control Registers

As stated above, the first register that must be programmed is the CONREG register in the video macrocell. The most significant 16 bits of the register are set to 0xE000 for all writes. The remaining 16 bits govern the pixel clock source and divide down rate, number of bits per pixel, number of FIFO preloads per DMA service, as well as whether the display is configured for a dual panel LCD display.

A typical display running in VGA mode will have a horizontal refresh rate of 31.47KHz, and a vertical refresh rate of 60Hz. In order to meet the 31.47KHz horizontal refresh rate, the pixel clock will have to be a minimum of 640 times this rate, or 20.14MHz. There is an additional complexity in that the monitor requires time to retrace the beam on the inside of the tube from the right edge of the screen to the left edge, plus there is usually what is referred to as a front porch and back porch. The porches are dead times in which the beam is not writing visible pixels to the screen. These are required for stable performance of the display. Normally retrace time is several pixel clock cycles, in our case an arbitrary 52 clock cycles with a 54 cycle front and back porch (retrace cycles and porch cycles must typically be either a multiple of 2 or 4). The horizontal refresh rate will not change, but rather than shift out 640 pixel clocks per horizontal raster, 640 + 54 + 54 + 52 = 800 pixel clock cycles will be required. This translates to an actual pixel clock frequency of 25.18MHz.

This clock rate is readily available by setting JP2 to produce a 25.18MHz clock output according to Table 44. Once the clock rate jumpers have been set, CONREG must be programmed to use VCLKI.

Set the number of FIFO periods to 16.

Since the macrocell should not be trying to display data while being set-up, CONREG should have the value 0xE000:4260 written to it prior to attempting to configure the macrocell or the DMA controller. After all has been configured, CONREG should be set to 0xE000:0260.

The External Register (EREG) must also be programmed. This register sets up the external functions of the video macrocell. This register is addressed by writing the following:

- Address 0x0340:0000 = 0xC005:1000.

The above data word turns on the on-board video DACs, sets the VSYNC and HSYNC pulses to be active low, leaves hires mode off (intended for displays > 1024x768), turns the LCD gray scaler off, turns pedestal currents off, and disables the external pixel shift clock.

4.13.2.2 Configuring the Palette

The next issue is configuring the video palette. The palette has 256 entries, and each write to the palette register will automatically increment the palette table pointer.

Prior to beginning this exercise, the palette table pointer should be set to zero. Do this by writing 0x1000:0000 to the video macrocell address (0x0340:0000).

Each write to the palette address will be an entry into the table. Each palette data entry consists of the 4-bit palette address in the macrocell, followed by a 28-bit word of data. The most significant 4 bits of palette data are from an external look-up table and will be zero in this case. The remaining 24 bits are broken down as 8 bits for each of blue, green, and red (in that order). A discussion of how RGB elements combine to produce different colors is not critical to the set-up of the palette. The configuration of each palette entry is entirely up to the developer. In the interest of keeping this discussion as simple as possible, set the palette up for 256 color gray scale. Do this by writing the following sequence to the palette (each entry is of the form 0x0EBB:GGRR):

0x0000:0000
0x0001:0101
0x0002:0202
0x0003:0303…

If different color schemes are desired, please feel free to experiment with the palette entries.

4.13.2.3 Horizontal and Vertical Configuration

In the above discussion on clocking, the rate was set to 25.18MHz. In order to set a horizontal refresh rate of ~31.47KHz, a total of 800 pixels per raster will be required. Arbitrarily setting the retrace time to 52 clock cycles, this translates to a front and back porch of 54 clock cycles each. The effective horizontal cycle time is then 31.475KHz; this is close enough for our purposes. Setting the horizontal parameters of the display will require programming seven different video registers (remember, each of these 32-bit values is written to 0x3400:0000 in the CL-PS7500FE's memory space):

1. Horizontal cycle time (0x8000:nnnn = 0x8000:0318)
2. Horizontal sync width (0x8100:nnnn = 0x8100:002C)
3. Horizontal border start (0x8200:nnnn = 0x8200:0068)
4. Horizontal display start (0x8300:nnnn = 0x8300:0062)
5. Horizontal display end (0x8400:nnnn = 0x8400:02E2)
6. Horizontal border end (0x8500:nnnn = 0x8500:02E8)
7. Horizontal cursor start (0x8600:nnnn = 0x8600:004F)

The details of how each of calculating each of these values can be found in the CL-PS7500FE Advance Data Book, Chapter 16.

For the purposes of video register programming, the term 'time' is always measured in pixel shift clocks, not conventional time. The discussion on clock frequency above should give some insight into how these values were calculated.

In terms of vertical configuration, an additional eight registers must be programmed. In the case of vertical settings, the important parameters are the number of lines in the display, the time required for vertical retrace (vertical sync. pulse width) and, if used, the border width in rasters. In this case, the number of lines is 525 (resulting in a vertical refresh of 60Hz), the border is set to zero and the retrace time is set to three cycles. The register locations and values are as follows:

1.	Vertical cycle	(0x9000:nnnn = 0x9000:020B)
2.	Vertical sync width	(0x9100:nnnn = 0x9100:0001)
3.	Vertical border start	(0x9200:nnnn = 0x9200:0017)
4.	Vertical display start	(0x9300:nnnn = 0x9300:0017)
5.	Vertical display end	(0x9400:nnnn = 0x9400:01F7)
6.	Vertical border end	(0x9500:nnnn = 0x9500:01F7)
7.	Vertical cursor start	(0x9600:nnnn = 0x9600:0017)
8.	Vertical cursor end	(0x9700:nnnn = 0x9700:0036)

4.13.2.4 Video DMA Configuration

Video data to be displayed on the screen is normally stored in a section of the CL-PS7500FE's memory. The DMA controller in the CL-PS7500FE must be configured to point to the location of the display information in memory and how many pixels worth of information are to be maintained in the video macrocell FIFO.

This section of DRAM is commonly referred to as a frame buffer, and will be a requirement regardless of which display medium is used (VGA or LCD). In the immediate term, we'll assume the frame buffer is located at 0x1400:0000, or at the bottom of DRAM Bank 1. The availability of this bank on a given development board will depend on the size of DRAM SIMM being used. The location of the buffer is entirely at the discretion of the developer. The size of the buffer can be calculated based on the number of bits per pixel (bpp) and, of course the number of pixels in the display. In this case, 640x480x8bpp=307200, or 0x0004:B000.

About quad words: The DMA controller transfers data in bursts of four 32-bit words or 16 bytes. These are referred to as Qwords. The Video DMA End address below is an example of where this is an important point. Since data in the frame buffer is treated as indivisible quad words, the Video DMA End Address must contain the first address of the last Qword in the display.

There is a second buffer governing the display of the cursor. Although the hardware cursor is not configured in this example, the registers governing its position on the screen must still be initialized.

Be sure to turn the video controller off prior to programming the DMA addresses, or set up the DMA before programming any DMA addresses. If the DMA is configured while the macrocell is trying to grab data from memory and display it, the processor will most likely crash.

There are a total of seven locations that must be configured in order to get the DMA controller working properly. There is also one last register in the macrocell that must be configured. First, the DMA configuration registers:

1.	Cursor DMA initial address	(0x0320:01C4 = 0x1400:0000)
2.	Video DMA current address A	(0x0320:01D0 = 0x1400:0000)
3.	Video DMA end	(0x0320:01D4 = 0x1404:AFF0)
4.	Video DMA start	(0x0320:01D8 = 0x1400:0000)
5.	Video DMA initial address A	(0x0320:01DC = 0x1400:0000)
6.	Video cursor DMA control register	(0x0320:01E0 = 0x0000:0020)
7.	Duplex LCD video initial address B	(0x0320:01E8 = 0x1400:0000)

For more complete information on these registers, please refer to the CL-PS7500FE Advance Data Book, Chapter 10.

The final register in the DMA configuration is the Data Control Register (DCTL) in the video macrocell. This register identifies the number of 32-bit words in a given line of data on the display. In this example, the value would be 640/4 or 160, or 0xA0. In keeping with the conventions used in this section, set (0x3400:0000 = 0x0000:00A0).

One final point about configuring the DMA controller. The CL-PS7500FE contains an MMU which will most likely be turned on by the developer (it is extremely useful). When programming the locations of the frame buffer into the DMA controller, use physical addresses for the locations. Addresses generated by the DMA controller do not go through the MMU.

4.13.2.5 VIDMUX Register

There is one last register that must be configured. This register is used to select the sound format used for the 16-bit stereo DAC as discussed in the sound section above. The register is also used to set the video multiplexor for VGA or LCD mode. Bit 0 of this register (address 0x0320:006C) should be set to 0 for normal VGA operation.

4.13.3 LCD

The LCD support on this design requires an additional daughter board connected to J19 on the board. This option has been chosen because each LCD has a different interface and different power requirements. This precludes standardizing on a common connection scheme such as the VGA monitor's high-density DB15. Connections to an actual display should be made on the LCD interface board and not directly on the development board. The pin-out of J19 is as follows:

Table 45: J19 LCD Adapter Module Header Pin-Out

Description	*#*	*#*	*Description*
ED0	1	2	ED1
ED2	3	4	ED3
ED4	5	6	ED5
ED6	7	8	ED7
ECLK	9	10	HCLK
VSYNC	11	12	HSYNC
XOUT	13	14	IOP4
IOP5	15	16	IOP6
VCC	17	18	VCC
GND	19	20	GND

All of these signals are connected directly to the CL-PS7500FE except for the XOUT signal. XOUT is a 14.31818Mhz clock generated by the Chrontel CH9294G.

Details of how to connect an LCD panel to this header are specific to the particular LCD panel being used, as well as the programming of the video macrocell necessary to correctly drive the LCD panel.

4.14 OS Porting Aid

The development board includes a bank of 8 LEDs (actually 10, but only 8 are connected). The bank can be accessed by writing to any location in the range 0x0302:BA00 to 0x0302:BAFF in the CL-PS7500FE's memory. When a bit is set to 0, the corresponding lamp in the array will be on, otherwise it will be off. This is an I/O mapped address in the bit positions D[15:8].

4.15 Power

Two power options are available. The first option is a set of local power supplies on the board for providing all required voltages. The second option is to use a conventional AT power supply. If the second option is the desired one, read no further. The limits and functions of the supply used will be written on the supply case.

The local supplies on the board are a combination of both switching and linear regulators.

4.15.1 Local 5V Switching Regulator

The local regulator for producing +5V is a switching type configured for operation in a "Buck" topology that operates at ~500kHz by default. The device is capable of producing up to 4.5A of current at +5V. This is unlikely, as the wall mounted transformer will only source 1.5A at 12V unregulated.

The inductor used in this design is really a small toroidal transformer from Coiltronics. The transformer is configured as an inductor in the parallel mode, and is capable of efficient coupling at up to several Megahertz switching frequencies.

For more information on this converter, or on efficient switching supplies in general, please consult the Linear Technology data sheet on the LT1374.

4.15.2 Local +/-12V Switching Regulator

The +/-12V supplies are actually generated from the same switching regulator, configured as a dual flyback converter. This supply is capable of sourcing up to up to 1.5A total. As with the 5V switching supply mentioned above, this supply is limited by the amount of current that can be sourced from the wall-mounted transformer. This regulator switches at ~500kHz by default.

The switching circuit uses an off-the-shelf switching transformer with six independent windings. The VersaPAC (VP3 series) from Coiltronics is capable of efficient energy coupling at up to 1MHz switching rates.

For more information on this converter, or on efficient switching supplies in general, please consult the Linear Technology data sheet on the LT1372.

4.15.3 Synchronous Operation of Switching Regulator

The two switching supplies used in this design operate at approximately the same frequency. This gives rise to a potential problem: If the two converters happen to switch at frequencies within a few Hz to a few kHz of each other, an audible beat tone may be produced on the supply rails of the board. This tone may be audible in the headset circuit.

To prevent this problem from occurring, each of the two switching regulators is available with an option for synchronous operation. This option allows an external oscillator to drive the switching circuit.

Each regulator requires a clock source that is above its natural operating frequency. The LT1374 and LT1372 both have a maximum natural operating frequency of 550kHz. Both regulators should not be operated much above 700kHz as this will make them less stable. The oscillator used to drive both of these is hence configured for operation at ~650kHz. The oscillator is built from a 555 timer and a local linear regulator producing +5V from the unregulated supply.

4.15.4 Linear Regulator for –5V

-5V is supplied via two possible sources: The on-board supplies or an AT power supply. The local regulator produces –5V by regulating up from the –12V regulated supply voltage and is capable of sinking up to 500mA.

If an AT power supply is used, it will produce its own –5V supply for the development board. In this case, the possibility exists that the local –5V supply will be in contention with the external –5V supply. To prevent this situation, there is an additional jumper on the board for disconnecting the local regulator from the –5V rail on the board. JP15 is described further in the following table:

Table 46: –5V Power Source Selection

Jumper	*Description*
[■■]	On-board –5V regulator supplying –5V rail.
▪ ▪	External, AT power supply used for supplying –5V rail.

4.15.5 Optional PC/AT Power Supply Connection

If an AT supply is used to power the development board, three power connections are required. A standard supply will have all required connectors.

The first is J24, which is a floppy drive power supply connector (4 pins on 0.200” centers). This connector supplies +12V and +5V.

The remaining two are 6-pin connectors, and are keyed such that plugging them in the wrong orientation will require effort. However, the two connectors are side by side, and it is possible to cross the two connectors. Use caution when connecting these to the board. The easiest way to make sure the orientation is correct is to ensure that all ground wires are in the center of the two connectors; these are easily distinguished as their wires are black.

5 Physical Specification

5.1 PCB Form Factor

The PCB form factor is 8.5” x 10”. This is similar in size to a standard PC motherboard. Although similar in size, the board will not be easily mountable in a PC case, as many of the connectors located around the periphery of the board have incompatible locations.

5.2 PCB Construction Materials

The PCB is constructed from materials with a flame rating of 94V0. This rating meets the self-extinguishing characteristics required by safety agencies in countries of sale.

5.3 Connector Details

DRAM SIMM Sockets (J6 and J7)
The SIMM sockets used in this design are a vertical mount to the PCB. The sockets are designed for 72-pin, 5V FPM or EDO style DRAM SIMMs. For dimension details on this socket, see AMP part # 7-382698-2.

FLASH SIMM Socket (J5)
The FLASH SIMM socket is of the same family as the DRAM SIMM socket with the key difference that it has 80-pins. This socket is intended to connect 5V FLASH SIMMs up to 16MBytes in size, configured as 32-bit wide data words. For dimension details on this socket, see AMP part # 8-382698-0.

Logic Analyzer headers (JP3-JP10)
The logic analyzer headers on this design are 2x10, 0.100” spacing, 0.025” square posts. For an example of the strip-header used for this port, see 3M part # 929710-10-36.

ISA Bus (J8, J9, J10, J11)
Although shown in the schematics as four discrete connectors, these form two ISA slots on the development board. The spacing of these connectors is twice the standard distance (0.800”) between ISA slots. For dimensional details of this socket, see AMP part # 176139-2.

Keyboard (J2)
The keyboard connector is a standard right-angle PS/2 keyboard connector. This interface is intended to have a standard PC keyboard connected to it. For dimension data, see CUI-Stack part # MD60-SM.

Mouse (J3)
The mouse connector is identical to the keyboard connector and is intended to have a PS/2 mouse connected to it. For dimension data, see CUI-Stack part # MD60-SM.

LCD (J19)

The LCD connector is a standard 0.100” spaced 2x10 header intended for mounting an LCD interface board to the development board. For an example of the strip-header used for this port, see 3M part # 929710-10-36.

VGA (J18)

The VGA connector for this design is a standard right-angle high-density DB15 used for connecting to VGA monitors. For dimensions, see AMP part # 748390-5.

Serial Ports (J13, J14)

The serial port connectors are male, right angle standard density 9-contact DSUBs. For dimension information on these connectors see AMP part # 787203-2.

Parallel Port (J12)

The parallel port connector used in this design is a right-angle female 25-contact DSUB connector. For dimension details on this connector, see AMP part # 787202-1.

Ethernet (J15)

The Ethernet connector used in this design is a standard right angle 10Base-T RJ45. For dimensional details of this connector, please see CorCom part # RJ45-8L2-S.

Headset (J1)

The headset jack is a 3.5mm stereo jack. Connect a standard headset to this port. For dimension data, see CUI-Stack part # SJ-3545N.

Power (wall-mounted) (J23)

The power connector is a two-contact right-angle male DIN style connector. The center post for this connector is 2.1mm in diameter with polarity not specified as the board has a full-wave bridge rectifier to prevent mishaps. For dimension data, see CUI-Stack part # PJ-002A.

Power (AT) (J24, J25, J26)

If the customer elects not to use a wall mounted supply for this board, the option is available to use a standard AT power supply. The connectors used to make up this set are: J24 – AMP part #174520, J25 and J26: Molex part # 15-48-0212.

6 Operating and Storage Environments

6.1 Temperature

The CL-PS7500FE development board can be stored safely at temperatures ranging from -20-85°C, inclusive. The CL-PS7500FE development board will function reliably in an ambient operating temperature of 0-70°C, inclusive.

6.2 Humidity

The CL-PS7500FE development board is reliable in storage and operating in relative humidity of 10% to 95% (non-condensing) in the appropriate temperature ranges specified above.

6.3 Air Flow

The CL-PS7500FE development board is reliable in storage and operating in static air at temperatures and relative humidity specified above.

6.4 Regulatory Compliance (FCC Part 15 Class B, and Part 68)

As this product is sold as an evaluation platform only, no EMI testing was performed.

7 Sources of Additional Information

Web page addresses for suppliers of parts used on the development board:

3M	http://www.mmm.com/interconnects/
Advanced Micro Devices	http://www.amd.com/
Advanced RISC Machines	http://www.arm.com/
AMP	http://connect.amp.com/
Coiltronics	http://www.coiltronics.com/
Chrontel	http://www.chrontel.com/
Cirrus Logic, Inc	http://www.cirrus.com/
Crystal Semiconductor	http://www.cirrus.com
CUI Stack	http://www.cuistack.com/
Cypress Semiconductor	http://www.cypress.com/
Global Engineering	http://global.ihs.com/
Hitachi	http://www.halsp.hitachi.com/
Linear Technology	http://www.linear-tech.com/
Molex	http://www.molex.com/
Standard Microsystems Corp.	http://www.smsc.com/

SMART Analog™

Preliminary product information describes products which are in production, but for which full characterization data is not yet available. Advance product information describes products which are in development and subject to development changes. Cirrus Logic, Inc. has made best efforts to ensure that the information contained in this document is accurate and reliable. However, the information is subject to change without notice and is provided “AS IS” without warranty of any kind (express or implied). No responsibility is assumed by Cirrus Logic, Inc. for the use of this information, nor for infringements of patents or other rights of third parties. This document is the property of Cirrus Logic, Inc. and implies no license under patents, copyrights, trademarks, or trade secrets. No part of this publication may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc. Items from any Cirrus Logic website or disk may be printed for use by the user. However, no part of the printout or electronic files may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc.Furthermore, no part of this publication may be used as a basis for manufacture or sale of any items without the prior written consent of Cirrus Logic, Inc. The names of products of Cirrus Logic, Inc. or other vendors and suppliers appearing in this document may be trademarks or service marks of their respective owners which may be registered in some jurisdictions. A list of Cirrus Logic, Inc. trademarks and service marks can be found at http://www.cirrus.com.