



2608 Sweetgum Drive  
Apex NC 27502  
Toll-free: 800-549-9377  
International: 919-387-0076  
FAX: 919-387-1302

---

# **XSV Board V1.1 Manual**

---

How to install and use  
your new XSV Board

Copyright ©1999-2001 by X Engineering Software Systems Corporation.

All XS-prefix product designations are trademarks of XESS Corp.

All XC-prefix product designations are trademarks of Xilinx.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

## Table of Contents

Limited Warranty .....	<b>Error! Bookmark not defined.</b>
Preliminaries .....	4
Getting Help! .....	4
Packing List .....	4
XSV Overview .....	5
XSV Board Features .....	5
Installation .....	8
Installing the XSVTOOLS Software .....	8
Unpacking the Board .....	8
Configuring the Jumpers .....	8
Applying Power .....	8
Connecting to a PC .....	9
Testing the XSV Board .....	9
Setting the Oscillator Frequency .....	<b>Error! Bookmark not defined.</b>
Programming the Interface .....	11
Downloading Virtex Configuration Bitstreams .....	13
Downloading Virtex Configuration Bitstreams to Flash .....	<b>Error! Bookmark not defined.</b>
XSV Circuitry .....	18
Programmable logic: XCV50-XCV800 Virtex FPGA and XC95108 CPLD .....	18
100 MHz programmable oscillator .....	18
16 Mbit Flash RAM .....	19
SRAM Banks .....	21
Video Decoder .....	22
RAMDAC and VGA Monitor Interface .....	23
Stereo Codec .....	25

---

Ethernet PHY.....	25
Expansion Headers.....	28
Pushbuttons and Eight-Position DIP Switch.....	30
Digit and Bargraph LEDs.....	31
PS/2 Port.....	33
USB Port.....	33
Parallel Port.....	34
Serial Port.....	36
Xchecker Interface.....	37
Power Connectors.....	38
XSV Pin Connections.....	39
XSV Schematics.....	40

# 1

## Preliminaries

### Getting Help!

Here are some places to get help if you encounter problems:

- If you can't get the XS40 Board hardware to work, send an e-mail message describing your problem to [help@xess.com](mailto:help@xess.com) or submit a problem report at <http://www.xess.com/reqhelp.html>. Our web site also has
  - [answers to frequently-asked-questions](#),
  - [example designs for the XSV Boards](#),
  - [application notes](#),
  - [a place to sign-up for our email forum](#) where you can post questions to other XS Board users.
- If you can't get your XILINX Foundation software tools installed properly, send an e-mail message describing your problem to [hotline@xilinx.com](mailto:hotline@xilinx.com) or check their web site at <http://support.xilinx.com>.

### Packing List

Here is what you should have received in your package:

- an XSV Board;
- a 6-foot, 25-wire cable with a male DB25 connector at each end;
- an XSTOOLS CDROM with software utilities and documentation for using the XSV Board.

# 2

## XSV Overview

The XSV Board brings you the power of the XILINX Virtex FPGA embedded in a framework for processing video and audio signals. The XSV Board has a single Virtex FPGA from 50K to 800K gates in size. The XSV can digitize PAL, SECAM, or NTSC video with up to 9-bits of resolution on the red, green, and blue channels and can output video images through a 110 MHz, 24-bit RAMDAC. The XSV can also process stereo audio signals with up to 20 bits of resolution and a bandwidth of 50 KHz. Two independent banks of 512K x 16 SRAM are provided for local buffering of signals and data.

The XSV Board has a variety of interfaces for communicating with the outside world: parallel and serial ports, Xchecker cable, a USB port, PS/2 mouse and keyboard port, and 10/100 Ethernet PHY layer interface. There are also two independent expansion ports, each with 38 general-purpose I/O pins connected directly to the Virtex FPGA.

You can configure the XSV Board through a PC parallel port, serial port, Xchecker cable or from a bitstream stored in the 16 Mbit Flash RAM. The Flash RAM can also store data for use by the FPGA after configuration is complete.

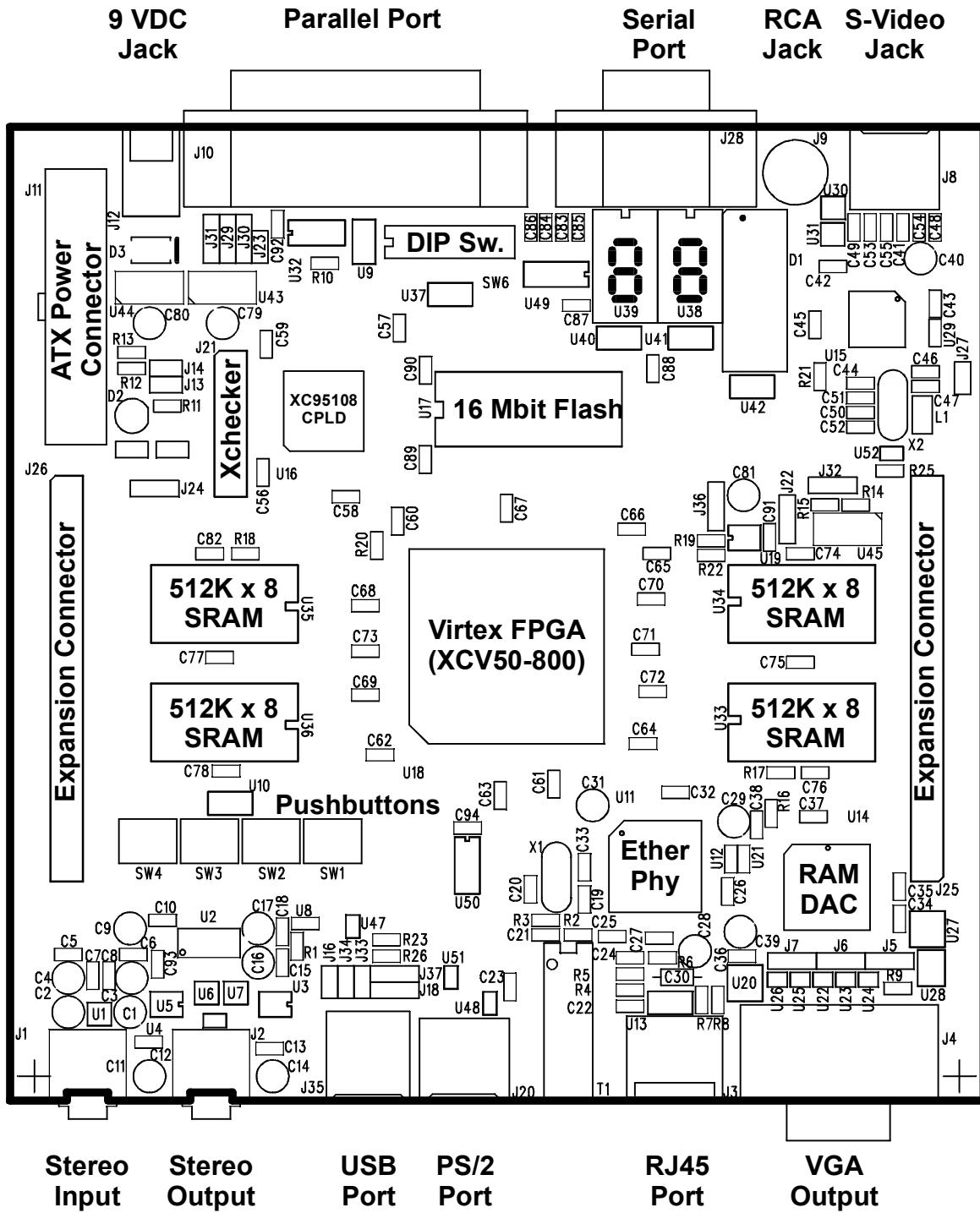
### XSV Board Features

The XSV Board includes the following resources:

- Programmable logic chips:
  - XILINX Virtex FPGA:** Virtex FPGAs from 57 Kgates (XCV50) up to 888 Kgates (XCV800) in a 240-pin PQFP or HQFP package are compatible with the XSV Board. The Virtex FPGA is the main repository of programmable logic on the XSV Board.
  - XILINX XC95108 CPLD:** The CPLD is used to manage the configuration of the Virtex FPGA via the parallel port, serial port, or Flash RAM. The CPLD also controls the configuration of the Ethernet PHY chip.
- Programmable oscillator that provides a clock signal to the FPGA and CPLD derived from a 100 MHz base frequency.
- 16 Mbit Flash RAM that can store multiple configurations or general-purpose data for the FPGA.
- Two independent 512K x 16 SRAM banks used by the FPGA for general-purpose data storage.

- Video decoder that accepts NTSC/PAL/SECAM signals through an RCA jack or S-video connector and outputs the digitized signal to the FPGA.
- RAMDAC with a 256-entry, 24-bit colormap that is used by the FPGA to output video to a VGA monitor.
- Stereo codec that lets the FPGA digitize and generate 0-50 KHz audio signals with up to 20 bits of resolution.
- 10BASE-T/100BASE-TX Ethernet PHY that allows the FPGA to access a LAN at up to 100 Mbps.
- Two expansion headers interface the FPGA to external circuitry through 76 general-purpose I/Os.
- Four pushbuttons and one eight-position DIP switch provide general-purpose inputs to the FPGA and CPLD.
- Two LED digits and one LED bargraph let the FPGA and CPLD display status information.
- Mouse/keyboard PS/2 port gives the FPGA access to common PC input devices.
- Single USB port provides the FPGA with a serial I/O channel with bandwidths of 1.5 to 12 Mbps.
- Parallel/serial port interfaces let the CPLD send and receive data in a parallel or serial format similar to a PC.
- Xchecker cable interface allows downloading and readback of the FPGA configuration.
- ATX power connector or 9 VDC power jack lets the XSV Board receive power from a standard ATX power supply or a 9 VDC power supply.

The locations of these resources are indicated in the simplified view of the XSV Board shown below. Each of these resources will be described in the following section.





# 3

## Installation

### Installing the XSVTOOLS Software

Run the SETUP.EXE file on the XSTOOLS CDROM. This will install the utilities and configuration files for testing and programming your XSV Board.

### Unpacking the Board

You should place the XSV Board on a non-conducting surface.

### Configuring the Jumpers

Your XSV Board should arrive with the shunts set on the jumpers in their default arrangement. The minimal shunt arrangement to allow testing of your XSV Board is as follows:

1. Place a shunt on jumper J23.
2. Place a shunt on pins 2 and 3 of jumper J31.
3. Place a shunt on pins 2 and 3 of J22.
4. Place a shunt on pins 1 and 2 of J36.

### Applying Power

You can supply the XSV Board with power in two ways:

1. **Recommended!** Attach an ATX PC power supply to connector J11. Remove any shunts on jumpers J13 and J14. Place a shunt on pins 1 and 2 of jumper J32.
2. Attach a 9 VDC power supply with a 2.1mm, center-positive plug to jack J12. The power supply must be able to source at least 1.5 A. Place shunts on jumpers J13 and J14. Place a shunt on pins 1 and 2 of jumper J32.

LED D2 will glow when the power is on.

## Connecting to a PC

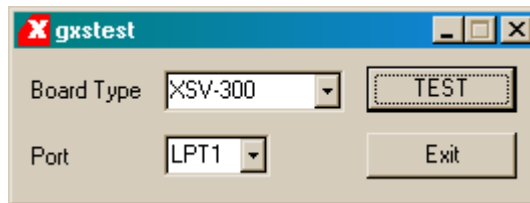
One DB25 connector on the 6-foot cable should be attached to connector J10 on the XSV Board and the other end should plug into the parallel port connector of a PC.

## Testing Your XSV Board

Once your XSV Board is installed and the jumpers are in their default configuration, you can test the board using the GUI-based GXSTEST utility as follows.



You start GXSTEST by clicking on the GXSTEST icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below.



Next you select the parallel port that your XSV Board is connected to from the Port pulldown list. GXSTEST starts with parallel port LPT1 as the default, but you can also select LPT2 or LPT3 depending upon the configuration of your PC.

After selecting the parallel port, you select either the XSV-50, XSV-100, XSV-300 or XSV-800 item in the Board Type pulldown list. Then click on the TEST button to start the testing procedure. GXSTEST will configure the FPGA to perform a test procedure on your XSV Board. After several seconds you will see a O displayed on the LED digit if the test completes successfully. Otherwise an E will be displayed if the test fails. A status window will also appear on your PC screen informing you of the success or failure of the test.

If your XSV Board fails the test, you will be shown a checklist of common causes for failure. If none of these causes applies to your situation, then test the XSV Board using another PC. In our experience, 99.9% of all problems are due to the parallel port. If you cannot get your board to pass the test even after taking these steps, then contact XESS Corp for further assistance.

As a result of testing the XSV Board, the CPLD is programmed with a special-purpose parallel port interface found in the dwnldtst.svf bitstream file located within the XSTOOLSXSV folder. You should reprogram the CPLD with the standard parallel port interface stored in the dwnldpar.svf file in the same folder when you want to use your XSV Board with the GXSLOAD utility. The procedure for doing this is described later.

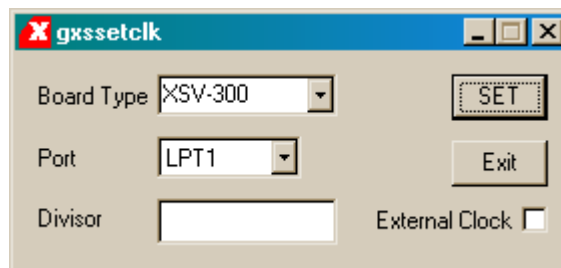
## Setting the XSV Board Clock Oscillator Frequency

The XSV Board has a 100 MHz programmable oscillator (a Dallas Semiconductor DS1075Z-100). The 100 MHz master frequency can be divided by factors of 1, 2, ... up to 2052 to get clock frequencies of 100 MHz, 50 MHz, ... down to 48.7 KHz, respectively. The divided frequency is sent to the rest of the XSV Board circuitry as a clock signal.

The divisor is stored in non-volatile memory in the oscillator chip so it will resume operation at its programmed frequency whenever power is applied to the XSV Board. You can store a particular divisor into the oscillator chip by using the GUI-based GXSSSETCLK as follows.



You start GXSSSETCLK by clicking on the GXSSSETCLK icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below.



Your next step is to select the parallel port that your XSV Board is connected to from the Port pulldown list. Then select your particular XSV Board from the Board Type pulldown list.

Next you enter a divisor between 1 and 2052 into the Divisor text box and then click on the SET button. Then follow the sequence of instructions given by XSSSETCLK for moving shunts and removing and restoring power during the oscillator programming process. At the completion of the process, the new frequency will be programmed into the DS1075.

An external clock signal can be substituted for the internal 100 MHz oscillator of the DS1075. Checking the External Clock checkbox will enable this feature in the programmable oscillator chip. If this option is selected, you are then responsible for providing the external clock to the XSV Board through header J27.

Note that GXSSSETCLK reprograms the CPLD on the XSV Board in order to access the programmable oscillator. So you will need to reprogram the CPLD with the standard parallel port interface circuit if you want to program the FPGA. (See the next section for details on this.)

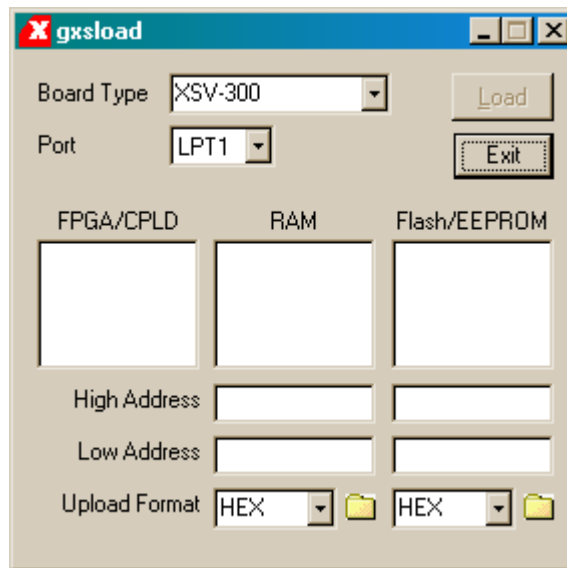
## Programming the Interface

The Virtex FPGA is the main repository of programmable logic on the XSV Board. The CPLD manages the configuration of the FPGA via the parallel port or from the Flash memory. Therefore, the CPLD must be configured so that it implements the necessary interface. The CPLD stores its configuration in its internal non-volatile memory so the interface is restored each time power is applied to the XSV Board (unless the interface circuit is erased).

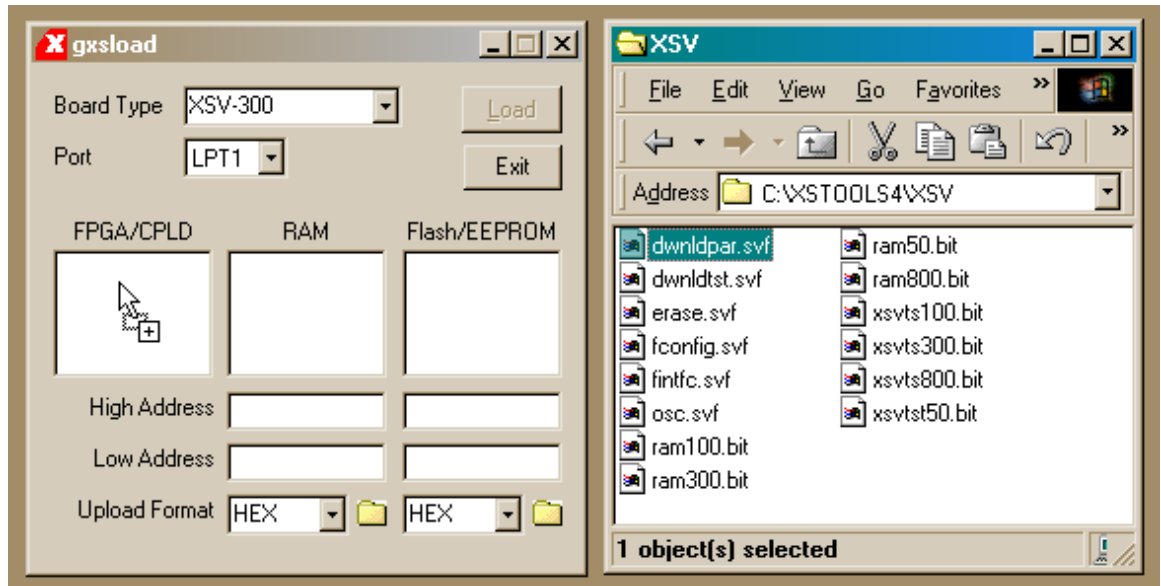
The CPLD is enabled for configuration by placing a shunt on jumper J23. The CPLD is configured with an interface by using the GXLOAD software utility. You start GXLOAD



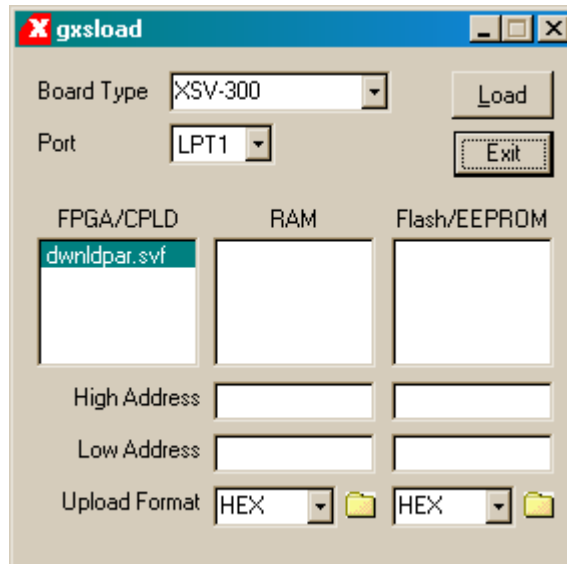
by clicking on the GXLOAD icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below. Then select the type of XSV Board you are using and the parallel port to which it is connected as described previously.



After setting the board type and parallel port, you can download an .SVF file to the CPLD on your XSV Board simply by dragging it into the FPGA/CPLD area of the GXLOAD window as shown below. To program the CPLD with the parallel port interface, drag the dwnldpar.svf file from the XSTOOLS\XSV folder.



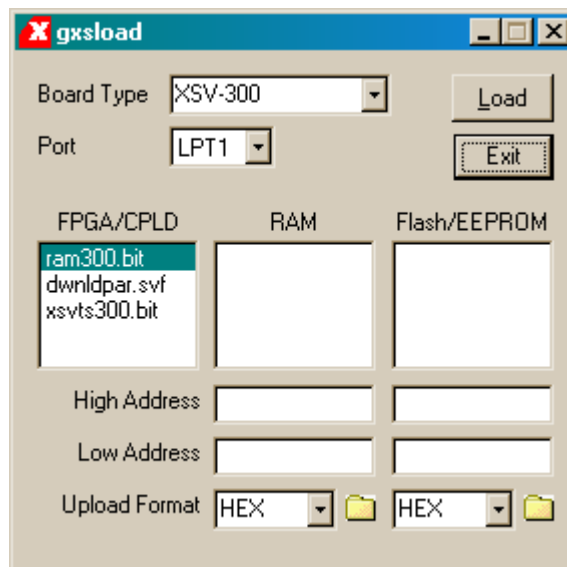
Once you release the left mouse button and drop the file, the highlighted file name appears in the FPGA/CPLD area and the Load button in the GXSLOAD window is enabled. Clicking on the Load button will begin sending the .SVF file to the CPLD on the XSV Board through the parallel port connection. During the downloading process, GXSLOAD will display the name of the file and the progress of the current download.



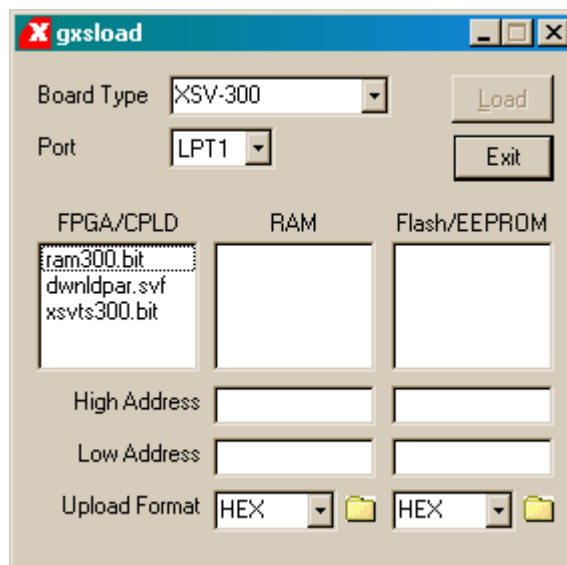
Once the CPLD is programmed with the parallel port interface circuit, you can remove the shunt from jumper J23 to prevent accidental reprogramming of the CPLD.

## Downloading Virtex Configuration Bitstreams

Once the CPLD is programmed with the downloading interface circuit, you can download bitstreams into the Virtex FPGA using the GXSLOAD utility. Make sure there is a shunt across pins 2 and 3 of jumper J31. Then drag & drop one or more .BIT files for the type of Virtex FPGA on your XSV Board into the FPGA/CPLD area of the GXSLOAD window. Clicking your mouse on a file name will highlight the name and select the file for downloading. Only one bitstream file at a time can be selected. Clicking on the Load button causes the highlighted Virtex configuration bitstream to pass through the parallel port and CPLD and then into the FPGA.



Double-clicking the highlighted file will deselect it so no file will be downloaded. Doing this disables the Load button.



## Storing Non-Volatile Designs in Your XSV Board

The Virtex FPGA on the XSV Board stores its configuration in an on-chip SRAM which is erased whenever power is removed. Once your design is finished, you may want to store the bitstream in the 2 MByte Flash device on the XSV Board which configures the FPGA for operation as soon as power is applied.

Before downloading to the Flash, the FPGA .BIT file must be converted into a .EXO or .MCS format using one of the following commands:

```
promgen -u 0 file.bit -p exo -s 2048
```

```
promgen -u 0 file.bit -p mcs -s 2048
```

In the commands shown above, the bitstream in the file.bit file is transformed into an .EXO or .MCS file format starting at address zero and proceeding upward until an upper limit of 2048 KBytes is reached.

**Before attempting to program the Flash, you must place all eight DIP switches into the OFF position!**

After the .EXO or .MCS file is generated, it is loaded into the Flash device by dragging it into the Flash/EEPROM area and clicking on the Load button. This activates the following sequence of steps:

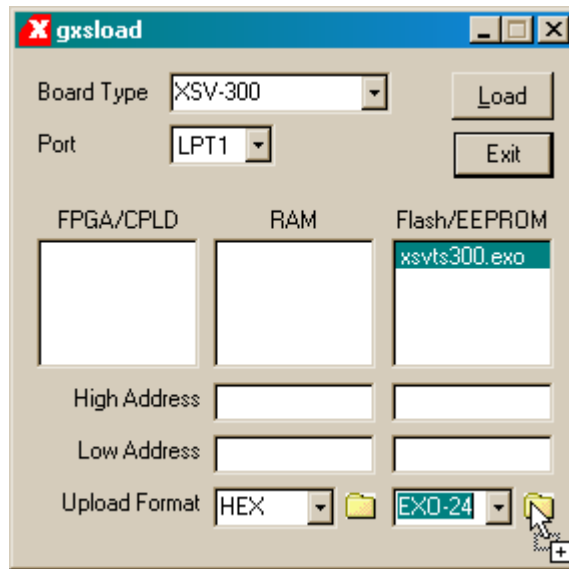
1. The entire Flash device is erased.
2. The CPLD on the XSV Board is reprogrammed to create an interface between the Flash device and the PC parallel port. (This interface is stored in the fintfc.svf bitstream file located within the XSTOOLS\XSV folder.)
3. The contents of the .EXO or .MCS file are downloaded into the Flash through the parallel port.
4. The CPLD is reprogrammed to create a circuit that configures the FPGA with the contents of the Flash when power is applied to the XSV Board. (This configuration loader is stored in the fconfig.svf bitstream file located within the XSTOOLS\XSV folder.)

Multiple files can be stored in the Flash device just by dragging them into the Flash/EEPROM area, highlighting the files to be downloaded and clicking the Load button. (Note that anything previously stored in the Flash will be erased by each new download.) This is useful if you need to store information in the Flash in addition to the FPGA bitstream. Files are selected and de-selected for downloading just by clicking on their names in the Flash/EEPROM area. **The address ranges of the data in each file should not overlap or this will corrupt the data stored in the Flash device!**

You can also examine the contents of the Flash device by uploading it to the PC. To upload data from an address range in the Flash, type the upper and lower bounds of the range into the High Address and Low Address fields below the Flash/EEPROM area, and

select the format in which you would like to store the data using the Upload Format pulldown list. Then click on the file icon and drag & drop it into any folder. This activates the following sequence of steps:

1. The CPLD on the XSV Board is reprogrammed to create an interface between the Flash device and the PC parallel port.
2. The Flash data between the high and low addresses (inclusive) is uploaded through the parallel port.
3. The uploaded data is stored in a file named FLSHUPLD with an extension that reflects the file format.



The uploaded data can be stored in the following formats:

MCS: Intel hexadecimal file format. This is the same format generated by the promgen utility with the `-p mcs` option.

HEX: Identical to MCS format.

EXO-16: Motorola S-record format with 16-bit addresses (suitable for 64 KByte uploads only).

EXO-24: Motorola S-record format with 24-bit addresses. This is the same format generated by the promgen utility with the `-p exo` option.

EXO-32: Motorola S-record format with 32-bit addresses.

XESS-16: XESS hexadecimal format with 16-bit addresses. (This is a simplified file format that does not use checksums.)

XESS-24: XESS hexadecimal format with 24-bit addresses.

XESS-32: XESS hexadecimal format with 32-bit addresses.



After the data is uploaded from the Flash, the CPLD on the XSV Board is left with the Flash interface programmed into it. You will need to reprogram the CPLD with either the parallel port or Flash configuration circuit before the board will function again. The CPLD configuration bitstreams are stored in the following files:

XSTOOLS\XSV\dwndpar.svf: Drag & drop this file into the FPGA/CPLD area and click on the Load button to put the XSV Board in a mode where it will configure the FPGA through the parallel port.

XSTOOLS\XSV\ fconfig.svf: Drag & drop this file into the FPGA/CPLD area and click on the Load button to put the XSV Board in a mode where it will configure the FPGA with the contents of the Flash device upon power-up.

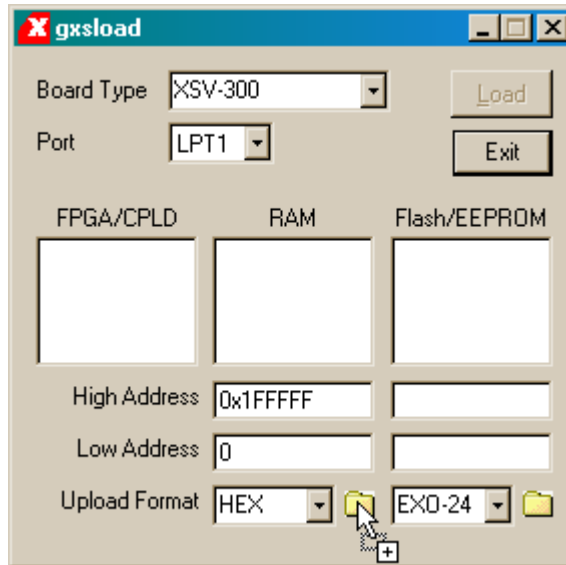
## Downloading and Uploading Data to the RAM in Your XSV Board

The XSV Board contains two banks of 512K x 16 RAM whose contents can be downloaded and uploaded by GXSLD. This is useful for initializing the RAM with data for use by the FPGA and then reading the RAM contents after the FPGA has operated upon it. The RAM is loaded with data by dragging & dropping one or more .EXO, .MCS, .HEX, and/or .XES files into the RAM area of the GXSLD window and then clicking on the Load button. This activates the following sequence of steps:

1. The Virtex FPGA on the XSV Board is reprogrammed to create an interface between the RAM device and the PC parallel port. (This interface is stored in the ram50.bit, ram100.bit, ram300.bit or ram800.bit bitstream file located within the XSTOOLS\XSV folder. **The CPLD must have previously been loaded with the dwndpar.svf file found in the same folder.**)
2. The contents of the .EXO, .MCS, .HEX or .XES files are downloaded into the RAM through the parallel port. **The data in the files will overwrite each other if their address ranges overlap.**
3. If any file is highlighted in the FPGA/CPLD area, then this bitstream is loaded into the FPGA or CPLD on the XSV Board. Otherwise the FPGA remains configured with the interface to the RAM.

You can also examine the contents of the RAM device by uploading it to the PC. To upload data from an address range in the RAM, type the upper and lower bounds of the range into the High Address and Low Address fields below the RAM area, and select the format in which you would like to store the data using the Upload Format pulldown list. Then click on the file icon and drag & drop it into any folder. This activates the following sequence of steps:

1. The Virtex FPGA on the XSV Board is reprogrammed to create an interface between the RAM device and the PC parallel port.
2. The RAM data between the high and low addresses (inclusive) is uploaded through the parallel port.
3. The uploaded data is stored in a file named RAMUPLD with an extension that reflects the file format.



The 16-bit data words in the SDRAM are mapped into the eight-bit data format of the .HEX, .MCS, .EXO and .XES files using a Big Endian style. That is, the 16-bit word at address  $N$  in the SDRAM is stored in the eight-bit file with the upper eight bits at location  $2N$  and the lower eight bits at location  $2N+1$ . This byte-ordering applies for both RAM uploads and downloads.

The XSV RAM is organized into a 512K x 16 right bank (RAM chips U33 and U34) and a 512K x 16 left bank (RAM chips U35 and U36). With respect to the GXSLOAD upload/download process, the right bank is located in the byte address range [0x000000–0x0FFFFFF] and the left bank is between byte addresses [0x100000–0x1FFFFFF].

# 4

## XSV Circuitry

This section describes the various sections of the XSV Board and shows how the I/O of the FPGA and CPLD are connected to the rest of the circuitry. The schematics which follow are less detailed so as to simplify the descriptions. Please refer to the complete schematics at the end of this document if you need more details.

### Programmable logic: XCV50-XCV800 Virtex FPGA and XC95108 CPLD

The XSV Board contains two programmable logic chips:

- A XILINX Virtex FPGA in a 240-pin QFP package. Virtex FPGAs from 57 Kgates (XCV50) up to 888 Kgates (XCV800) are compatible with the XSV Board. The Virtex FPGA is the main repository of programmable logic on the XSV Board.
- A XILINX XC95108 CPLD that is used to manage the configuration of the Virtex FPGA via the parallel port, serial port, or Flash RAM. The CPLD also controls the configuration of the Ethernet PHY chip.

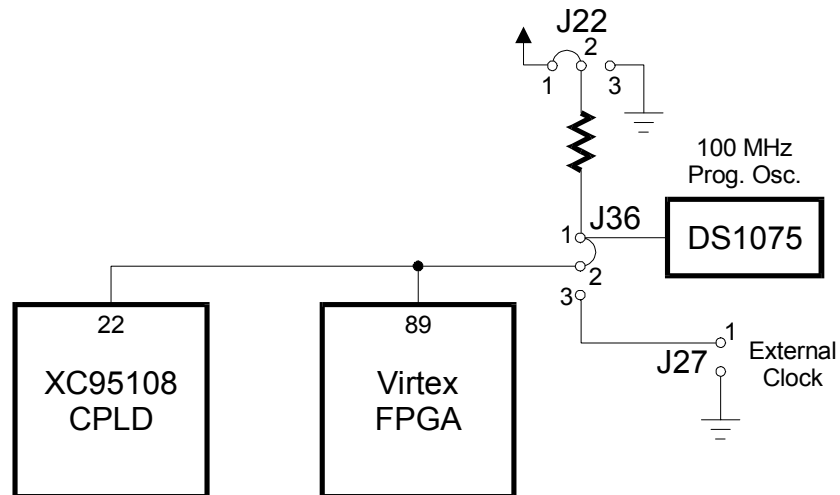
### 100 MHz programmable oscillator

A Dallas DS1075 programmable oscillator (<http://www.dalsemi.com/DocControl/PDFs/1075.pdf>) provides a clock signal to both the FPGA and the CPLD. The DS1075 has a maximum frequency of 100 MHz that is divided to provide frequencies of 100 MHz, 50 MHz, 33.3 MHz, 25 MHz, ..., 48.7 KHz. The clock signal is connected to dedicated clock inputs of both the CPLD and FPGA as follows:

DS1075 Output	Virtex FPGA Pin	XC95108 CPLD Pin
CLK	89	22

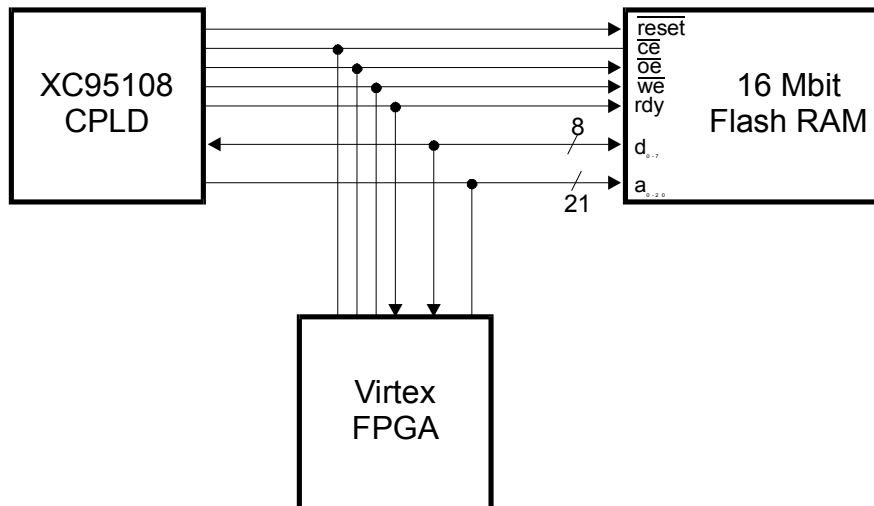
To set the divisor value, the DS1075 must be placed in its programming mode. This is done by pulling the clock output to Vcc on power-up with a shunt across pins 1 and 2 of jumper J22. Then programming commands to set the divisor can be sent to the DS1075 by either the CPLD or FPGA. The divisor is stored in EEPROM in the DS1075 so it will be restored whenever power is applied to the XSV Board. The shunt on jumper J22 must be across pins 2 and 3 to make the oscillator output a clock signal upon power-up.

To get a precise frequency value or to sync the XSV circuitry with an external system, you can insert an external clock signal through pin 1 of connector J27 and place a shunt across pins 2 and 3 of jumper J36. This external clock replaces the output from the DS1075 oscillator.



## 16 Mbit Flash RAM

An Intel 28F016S5 Flash RAM (<http://developer.intel.com/design/flcomp/datashts/290597.htm>) with 16 Mbits of storage ( $2M \times 8$ ) is connected to both the Virtex FPGA and XC95108 CPLD as follows:



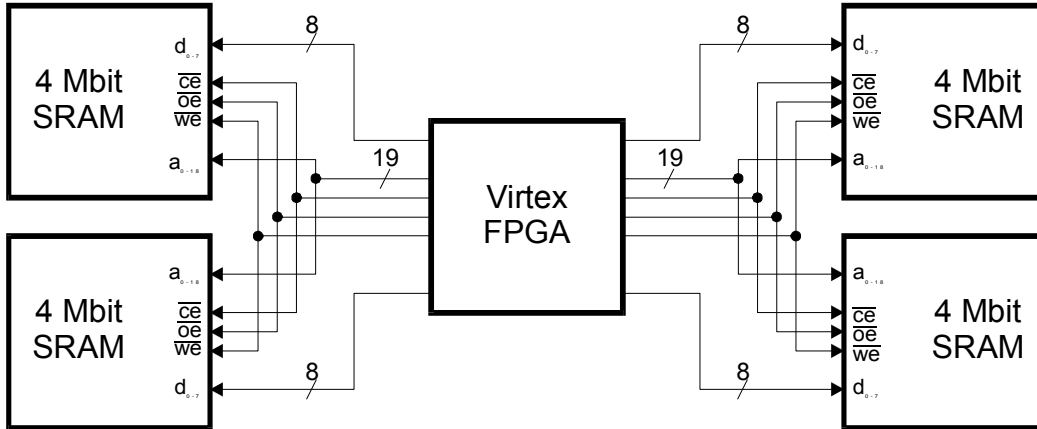
The CPLD and FPGA both have access to the Flash RAM. Typically, the CPLD will program the Flash with data passed through the parallel or serial port. If the data is an FPGA configuration bitstream then the CPLD can be configured to program the FPGA with the Flash bitstream whenever the XSV Board is powered up. After power-up, the FPGA can read and/or write the Flash. (Of course, the CPLD and FPGA have to be programmed such that they do not conflict if both are trying to access the Flash.) The Flash can be disabled by raising the /CE pin to Vcc in which case the I/O lines connected to the Flash can be used for general-purpose communication between the FPGA and the CPLD.

The pins of the FPGA and CPLD connected to the Flash RAM are listed below:

<b>Flash RAM Pin</b>	<b>Virtex FPGA Pin</b>	<b>XC95108 CPLD Pin</b>
/RESET	N/A	3
/CE	170	46
/OE	173	42
/WE	131	43
RDY	171	41
D0	177	32
D1	167	33
D2	163	34
D3	156	35
D4	145	36
D5	138	37
D6	134	39
D7	124	40
A0	132	16
A1	133	17
A2	139	18
A3	141	19
A4	144	20
A5	147	23
A6	152	24
A7	154	25
A8	157	27
A9	160	28
A10	162	29
A11	169	30
A12	168	49
A13	161	50
A14	159	52
A15	155	53
A16	153	54
A17	149	55
A18	146	56
A19	142	58
A20	140	59

## SRAM Banks

The FPGA has access to two independent banks of SRAM as shown below: Each SRAM bank is organized as 512K × 16 bits. Each bank is made from two Winbond AS7C4096 SRAMs ([ftp://ftp14.ba.best.com/pub/pai/FTP\\_site/pdf/sram.pdf/as7c34096.pdf](ftp://ftp14.ba.best.com/pub/pai/FTP_site/pdf/sram.pdf/as7c34096.pdf)). The FPGA pins connected to the SRAM banks are shown in the accompanying table.

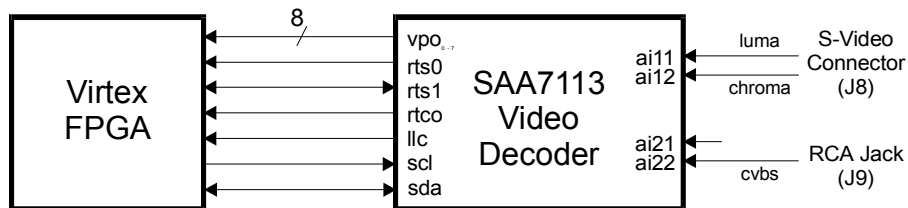


SRAM Pin	Virtex FPGA Pin to Left Bank	Virtex FPGA Pin to Right Bank
/CE	186	109
/OE	228	95
/WE	201	68
D0	202	70
D1	203	71
D2	205	72
D3	206	73
D4	207	74
D5	208	78
D6	209	79
D7	215	80
D8	216	81
D9	217	82
D10	218	84
D11	220	85
D12	221	86
D13	222	87
D14	223	93
D15	224	94

SRAM Pin	Virtex FPGA Pin to Left Bank	Virtex FPGA Pin to Right Bank
A0	200	67
A1	199	66
A2	195	65
A3	194	64
A4	193	63
A5	192	57
A6	191	56
A7	189	55
A8	188	54
A9	187	53
A10	238	108
A11	237	107
A12	236	103
A13	235	102
A14	234	101
A15	232	100
A16	231	99
A17	230	97
A18	229	96

## Video Decoder

The XSV Board can digitize NTSC, SECAM, and PAL video signals using the SAA7113 video decoder (<http://www-us.semiconductors.philips.com/pip/SAA7113H>). The digitized video arrives at the FPGA over the VPO bus. The arrival of video data is synchronized with the rising edge of the LLC (line-locked clock) from the video decoder. The FPGA programs the video options of the SAA7113 using the I<sup>2</sup>C bus (SCL and SDA).

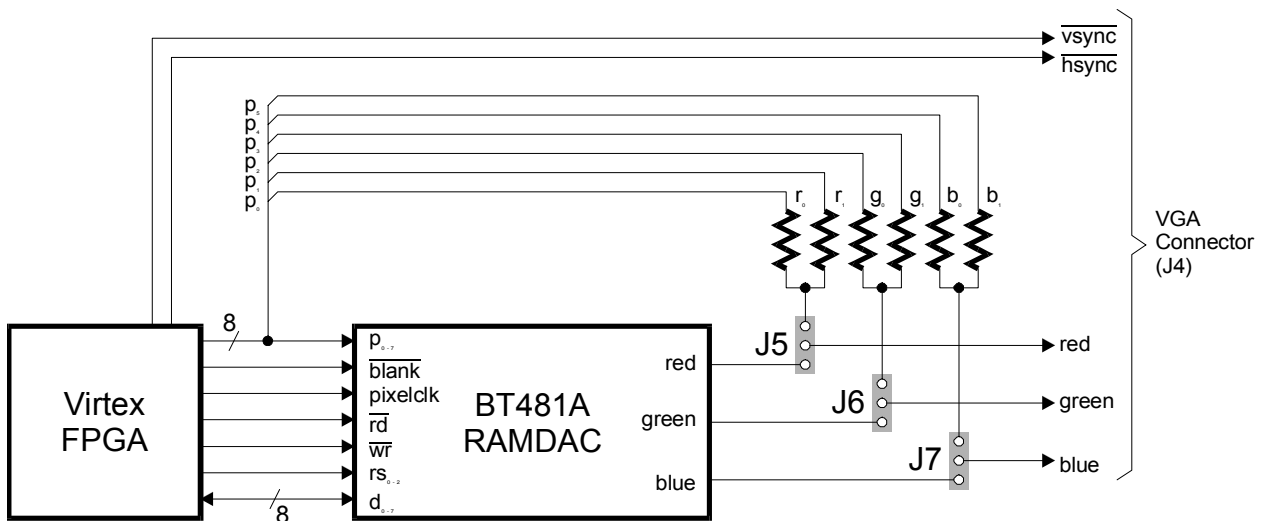


SAA7113 Pin	Virtex FPGA Pin
LLC	92
RTS0	111
RTS1	110

SAA7113 Pin	Virtex FPGA Pin
RTCO	113
VPO0	116
VPO1	117
VPO2	118
VPO3	125
VPO4	126
VPO5	127
VPO6	128
VPO7	130
SCL	114
SDA	115

## RAMDAC and VGA Monitor Interface

The FPGA can generate a video signal for display on a VGA monitor either directly or using a BT481A RAMDAC ([http://www.erc.msstate.edu/~reese/EE4993/data\\_sheets/btl481a\\_c.pdf](http://www.erc.msstate.edu/~reese/EE4993/data_sheets/btl481a_c.pdf)) depending upon the arrangement of the shunts on jumpers J5, J6, and J7.



When the FPGA is directly generating VGA signals, the lower six bits of the P bus provide two-bits of red, green, and blue color information to a simple resistor-ladder DAC. The outputs of the DAC are sent to a VGA monitor along with the horizontal and vertical sync pulses (/HSYNC, /VSYNC) from the FPGA.

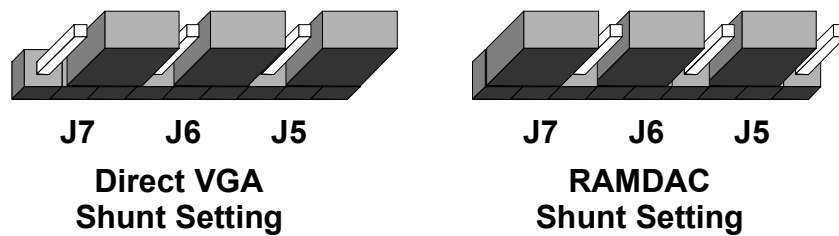
When the RAMDAC generates the VGA color signals, then the FPGA uses the full eight-bit P bus to pass the index of the color for the current pixel. The index is used to look up the 24-bit color value (eight bits for the red, green, and blue components) stored in the



256-entry colormap of the RAMDAC chip. The transfers over the P bus are synchronized with the PIXELCLK generated by the FPGA. The FPGA lowers the /BLANK signal when the pixels fall outside the desired visible area of the monitor screen.

The colormap of the RAMDAC is initialized by the FPGA using the D bus along with the RS, /WR, and /RD signals. The 24-bit colormap entries are passed in groups of three bytes over the eight-bit D bus synchronized by the /WR signal. The register-select signals (RS0, RS1, RS2) select the staging register for writing the colormap. The contents of the staging register are written into the colormap after the last byte of color information arrives over the D bus, and then the internal colormap address is incremented to point to the next entry.

The shunt placement to enable the FPGA to generate VGA signals directly or through the RAMDAC is shown below.



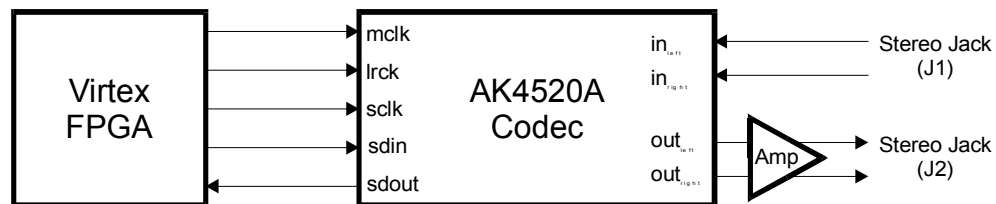
The pin assignments for the connection of the FPGA to the VGA signal generation circuitry are shown below. Note that the FPGA shares some connections between the RAMDAC and the chip which interfaces to the Ethernet (LXT970A). The RAMDAC pins are used to load the colormap and should not be active except during system initialization. The other connections are used for Ethernet data transmission and reception and are usually only active after system initialization.

Direct VGA Pin	RAMDAC Pin	Virtex FPGA Pin	LXT970A Function
	PIXELCLK	52	
/HSYNC	/HSYNC	48	
/VSYNC	/VSYNC	49	
	/BLANK	50	
RED0	P0	70	
RED1	P1	71	
GREEN0	P2	72	
GREEN1	P3	73	
BLUE0	P4	74	
BLUE1	P5	78	
	P6	79	
	P7	80	
	/RD	47	
	/WR	46	
	RS0	31	TXD4
	RS1	28	RX_ER
	RS2	26	RX_DV
	D0	42	TXD0

Direct VGA Pin	RAMDAC Pin	Virtex FPGA Pin	LXT970A Function
	D1	41	TXD1
	D2	40	TXD2
	D3	39	TXD3
	D4	38	RXD0
	D5	36	RXD1
	D6	35	RXD2
	D7	34	RXD3

## Stereo Codec

The XSV Board has an AK4520A stereo codec (<http://www.akm.com/ProductPages/ak4520a.html>) that accepts two analog input channels from jack J1, digitizes the analog values, and sends the digital values to the FPGA as a serial bit stream. The codec also accepts a serial bit stream from the XS Board and converts it into two analog output signals, which exit the XSV Board through jack J2. The serial bit streams are synchronized with a clock from the FPGA that enters the codec on SCLK signal. The FPGA uses the LRCK signal to select the left or right channel as the source/destination of the serial data. The master clock from the FPGA (MCLK) synchronizes all the internal operations of the codec.

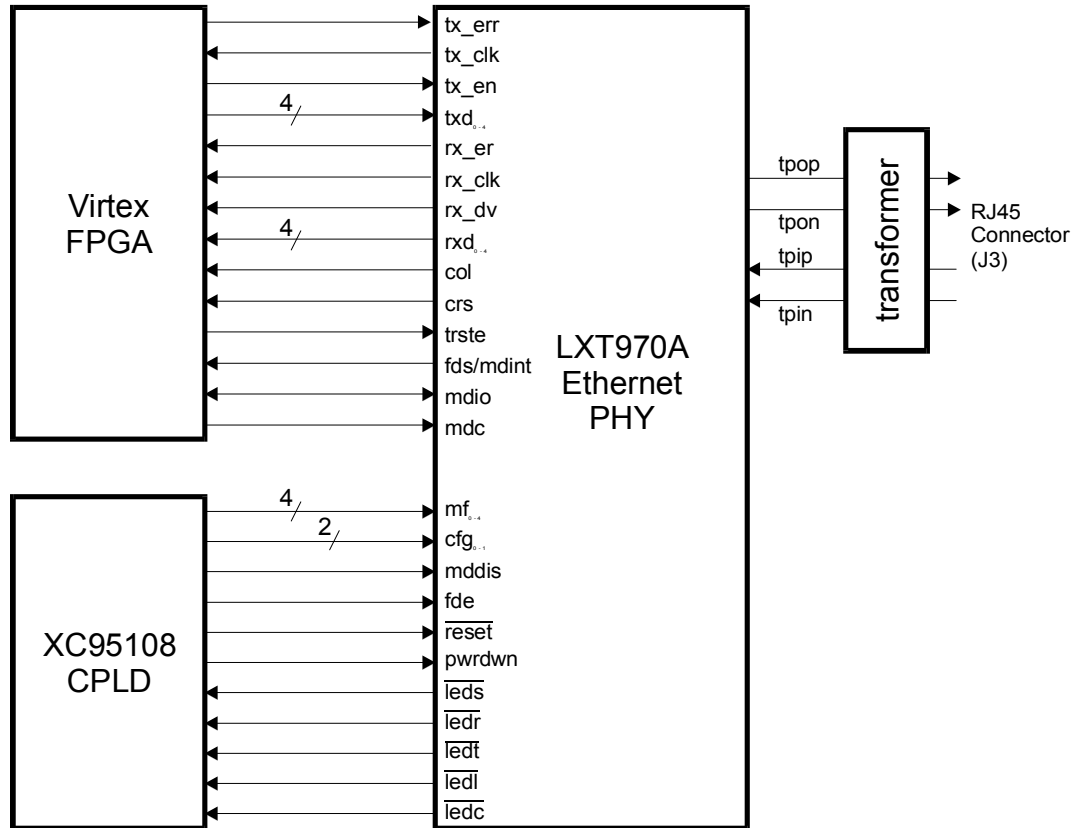


The FPGA pins which connect to the codec are as follows:

Stereo Codec Pin	Virtex FPGA Pin
MCLK	3
LRCK	4
SCLK	5
SDIN	6
SDOUT	7

## Ethernet PHY

The XSV Board interfaces to an Ethernet LAN at 10 or 100 Mbps. The LXT970A Ethernet PHY chip (<http://128.11.21.45/scripts/mardev/product/lxt970.asp>) connects to both the FPGA and the CPLD. The FPGA acts as a MAC (media access controller) and manages the transfer of data packets to and from the PHY chip, while the CPLD controls the configuration pins that determine the operational mode of the PHY chip.



The FPGA enables the transmitter with TX\_EN and sends bits on TXD<sub>4:0</sub> in sync with the transmit clock (TX\_CLK) generated by the PHY chip. The PHY chip is alerted to transmission errors that occur in the MAC when the TX\_ERR signal is asserted. The FPGA also receives an indication when valid data has been received (RX\_DV) and the data (RXD<sub>0:4</sub>) in sync with the receiver clock (RX\_CLK) from the PHY chip. Any reception errors are indicated to the FPGA via the RX\_ER signal. The CRS signal indicates when the receiver is non-idle. The COL signal is asserted when data collides on the Ethernet.

The FPGA can disable the interface to the PHY chip by asserting the tristate control (TRSTE). Otherwise, the FPGA passes management information to and from the PHY chip over the serial data line (MDIO) in sync with a clock (MDC). The FPGA can be alerted to changes in PHY chip status by the FDS/MDINT interrupt line.

The CPLD sets the static values on pins which control the configuration of the PHY chip. Pins MF0-4 set the modes for auto-negotiation, repeating, symbol transmission, scrambling, etc. Likewise, the configurations signals (CFG0-1) select the 10 Mbps or 100 Mbps operating speed of the PHY chip. MDDIS enables/disables the management information interface. FDE selects either full-duplex or half-duplex communication mode. The reset (/RESET) and power-down (PWRDWN) signals do exactly what they say.

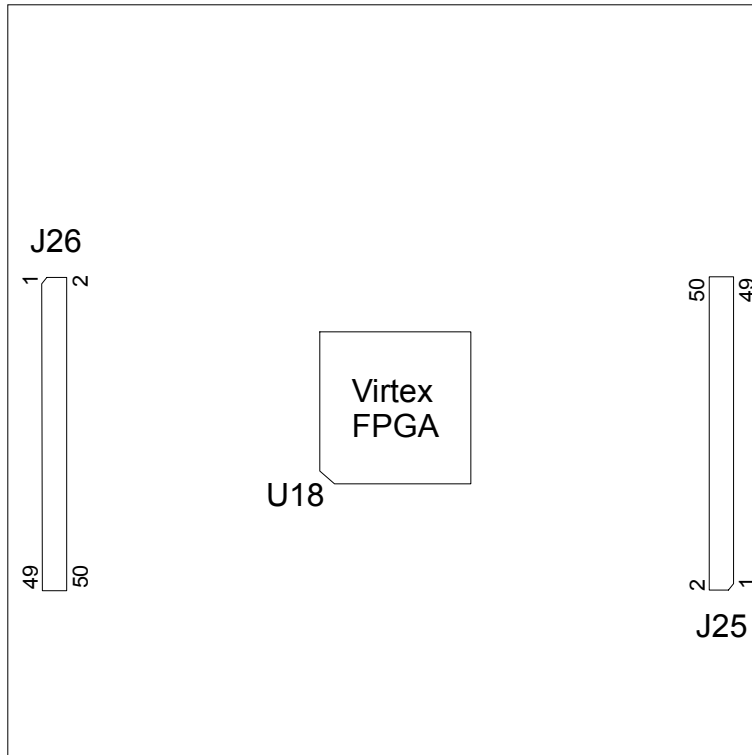
The CPLD also gets receives the status outputs from the PHY chip that normally drive LEDs. The outputs are active-low and indicate when 100 Mbps operation is selected (/LEDS), the receiver is active (/LEDR), the transmitter is active (/LEDT), the link is active (/LEDL), and a collision is detected (/LEDC). The CPLD can relay these signals to the LEDs on the XSV Board if you wish to display the Ethernet status.

The connections of the PHY chip to the FPGA and CPLD are listed below. Note that the FPGA shares some connections between the PHY chip and the RAMDAC. The RAMDAC pins are used to load the colormap and should not be active except during system initialization. The PHY connections are used for data transmission and reception and are usually only active after system initialization.

<b>LXT970A Pin</b>	<b>Virtex FPGA Pin</b>	<b>XC95108 CPLD Pin</b>	<b>RAMDAC</b>
COL	23		
CRS	21		
TRSTE	24		
TX_CLK	210		
TX_EN	25		
TX_ER	27		
TXD0	42		D0
TXD1	41		D1
TXD2	40		D2
TXD3	39		D3
TXD4	31		RS0
RX_CLK	213		
RX_DV	26		RS2
RX_ER	28		RS1
RXD0	38		D4
RXD1	36		D5
RXD2	35		D6
RXD3	34		D7
RXD4	33		
FDS/MDINT	18		
MDC	19		
MDIO	20		
MDDIS		94	
MF0		91	
MF1		90	
MF2		89	
MF3		87	
MF4		86	
CFG0		93	
CFG1		2	
FDE		92	
/RESET		3	
/LEDS		1	
/LEDR		95	
/LEDT		96	
/LEDL		97	
/LEDC		99	

## Expansion Headers

The XSV Board has two 50-pin headers (J25 and J26) which connect the FPGA to external systems. The arrangement of the headers is shown below:



The connections between the FPGA and the expansion headers are listed below. The FPGA pins which connect to the left and right expansion headers are also connected to the left and right banks of SRAM, respectively. The SRAM bank chip-enable should be raised to disable the SRAMs on that side if the associated expansion header is being used for external I/O.

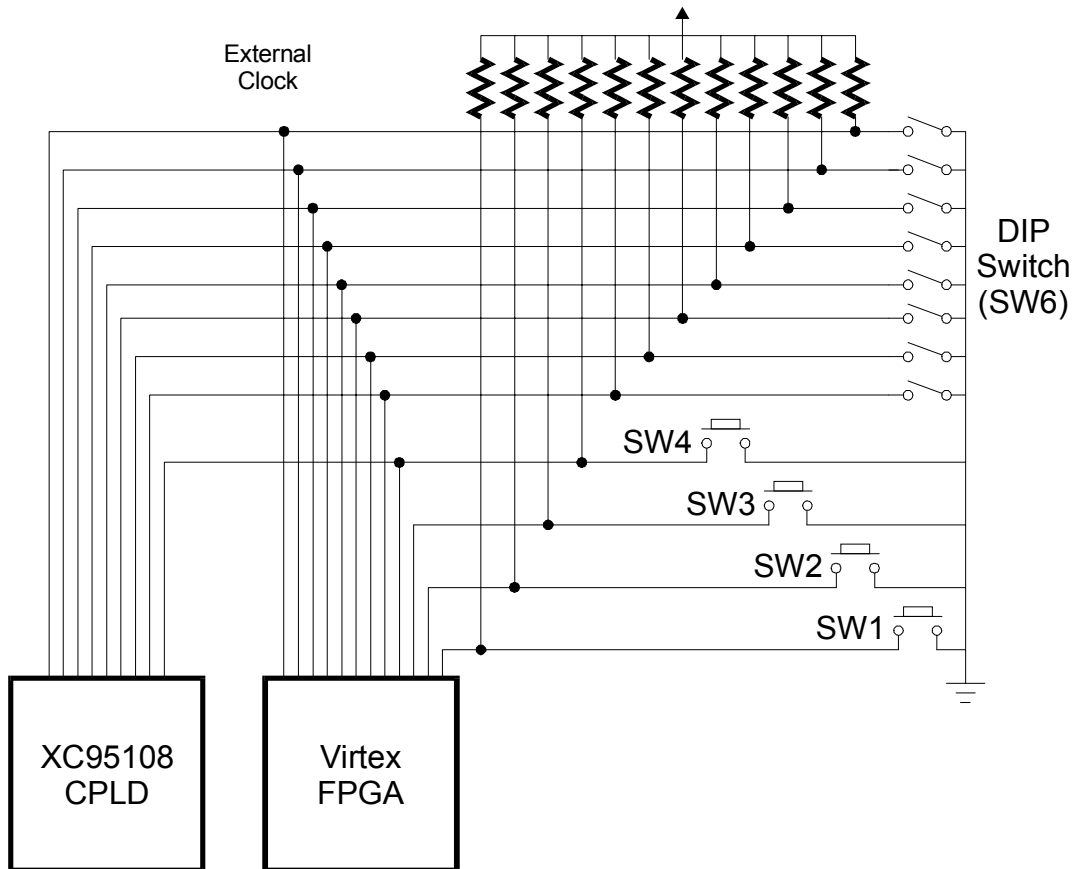
Expansion Connector Pin	Virtex FPGA Pin to Left Connector	Virtex FPGA Pin to Right Connector	SRAM Function
1	186	109	/CE
2	187	53	A9
3			+5V
4	188	54	A8
5	189	55	A7
6	191	56	A6

<b>Expansion Connector Pin</b>	<b>Virtex FPGA Pin to Left Connector</b>	<b>Virtex FPGA Pin to Right Connector</b>	<b>SRAM Function</b>
7			GND
8	192	57	A5
9	193	63	A4
10	194	64	A3
11			+5
12	195	65	A2
13	199	66	A1
14	200	67	A0
15			GND
16	201	68	/WE
17	202	70	D0
18	203	71	D1
19			+5
20	205	72	D2
21	206	73	D3
22	207	74	D4
23			GND
24	208	78	D5
25	209	79	D6
26	215	80	D7
27			+3.3
28	216	81	D8
29	217	82	D9
30	218	84	D10
31			GND
32	220	85	D11
33	221	86	D12
34	222	87	D13
35			+3.3
36	223	93	D14
37	224	94	D15
38	228	95	/OE
39			GND
40	229	96	A18
41	230	97	A17
42	231	99	A16
43			+3.3
44	232	100	A15
45	234	101	A14
46	235	102	A13
47			GND
48	236	103	A12
49	237	107	A11
50	238	108	A10

## Pushbuttons and Eight-Position DIP Switch

The XSV Board has a bank of eight DIP switches and four pushbuttons that are accessible from the FPGA. The CPLD is also connected to the DIP switches and one of the pushbuttons. When pressed, each pushbutton pulls the connected pin of the FPGA and CPLD to ground. Otherwise, the pin is pulled high through a resistor. Likewise, each DIP switch pulls the connected pin of the FPGA or CPLD to ground when it is closed or ON. When the DIP switch is open or OFF, the pin is pulled high through a resistor.

**When not being used, the DIP switches should be left in the open or OFF configuration so the pins of the FPGA and CPLD are not tied to ground and can freely move between logic low and high levels.**



The table below lists the connections from the FPGA and CPLD to the switches. The DIP switches also share the same pins as the uppermost eight bits of the Flash RAM address bus. If the Flash RAM is programmed with several FPGA bitstreams, then the DIP switch can be used to select a particular bitstreams which will be loaded into the FPGA by the CPLD on power-up.

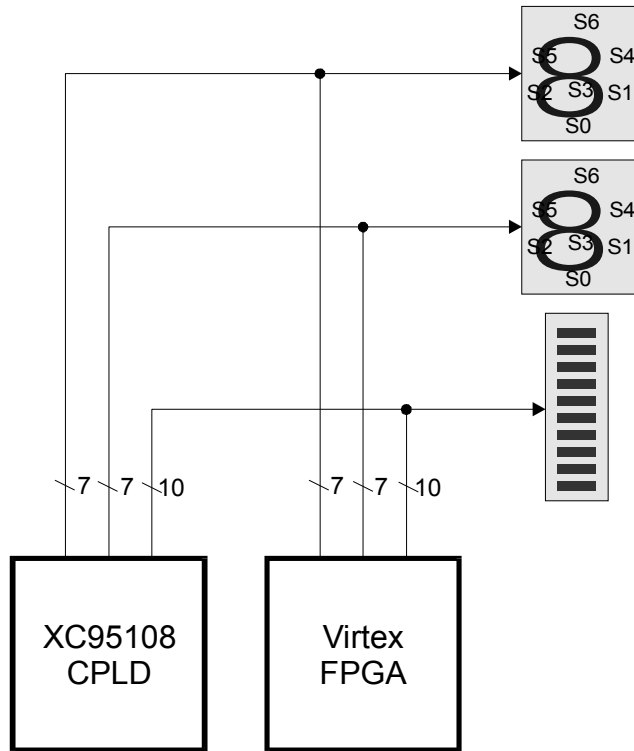
Switch	Virtex FPGA Pin	XC95108 CPLD Pin	Flash RAM Function
SW1	174		
SW2	175		
SW3	176		
SW4	185	7	
DIPSW1	161	50	A13
DIPSW2	159	52	A14
DIPSW3	155	53	A15
DIPSW4	153	54	A16
DIPSW5	149	55	A17
DIPSW6	146	56	A18
DIPSW7	142	58	A19
DIPSW8	140	59	A20

## Digit and Bargraph LEDs

The XSV Board has a 10-segment bargraph LED and two more 7-segment LED digits for use by the FPGA and CPLD. All of these LEDs are active-high meaning that an LED segment will glow when a logic-high is applied to it.

The table below lists the connections from the FPGA and CPLD to the LEDs. The LEDs also share the same pins as the uppermost eight bits of the Flash RAM address bus. If the Flash RAM is programmed with several FPGA bitstreams, then the DIP switch can be used to select a particular bitstreams which will be loaded into the FPGA by the CPLD.



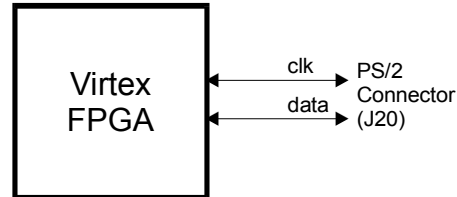


	LED	Virtex FPGA Pin	XC95108 CPLD Pin	Flash RAM Function
Left Digit	SL0	177	32	D0
	SL1	167	33	D1
	SL2	163	34	D2
	SL3	156	35	D3
	SL4	145	36	D4
	SL5	138	37	D5
	SL6	134	39	D6
Right Digit	SR0	124	40	D7
	SR1	132	16	A0
	SR2	133	17	A1
	SR3	139	18	A2
	SR4	141	19	A3
	SR5	144	20	A4
	SR6	147	23	A5
Bargraph	B0	152	24	A6
	B1	154	25	A7
	B2	157	27	A8
	B3	160	28	A9
	B4	162	29	A10
	B5	169	30	A11
	B6	168	49	A12

	<b>LED</b>	<b>Virtex FPGA Pin</b>	<b>XC95108 CPLD Pin</b>	<b>Flash RAM Function</b>
	B7	173	42	/OE
	B8	131	43	/WE
	B9	171	41	RDY

## PS/2 Port

The XSV Board provides a PS/2-style interface (mini-DIN connector J20) to either a keyboard or a mouse. The FPGA receives two signals from the PS/2 interface: a clock signal and a serial data stream that is synchronized with the falling edges on the clock signal.

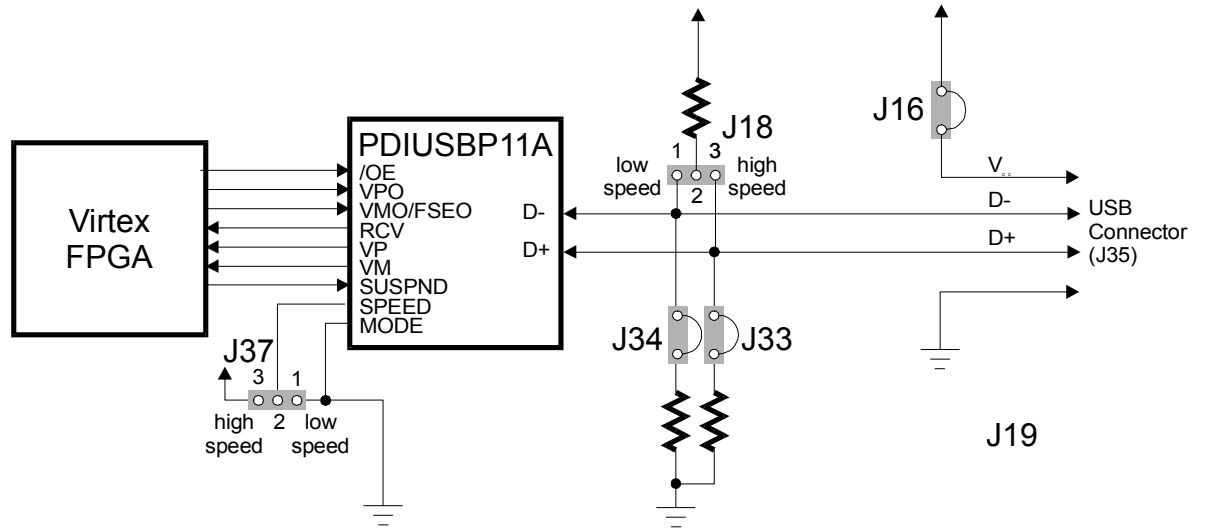


The following table shows the connections from the FPGA to the PS/2 interface.

<b>PS/2 Port Pin</b>	<b>Virtex FPGA Pin</b>
CLK	13
DATA	17

## USB Port

The XSV Board has a USB interface (J35) that can be connected to a variety of high-speed or low-speed USB peripherals. The FPGA interfaces to the two differential data signals from the USB port through a PDIUSBP11A USB interface chip ([http://www-us.semiconductors.philips.com/pip/PDIUSBP11A\\_2](http://www-us.semiconductors.philips.com/pip/PDIUSBP11A_2)).



The USB port is set to high (12 Mbps) or low speed (1.5 Mbps) by shunts on jumpers J18 and J37. A 15K load can be placed on the D+ and D- USB signals by placing shunts across jumpers J33 and J34. If the USB peripheral connected to the port needs to draw power from the XSV Board, then a shunt should be placed on jumper J16.

The connections of the FPGA to the USB interface chip are listed below. Note that the FPGA shares some of its pins between the USB interface, the PS/2 interface and one pushbutton switch.

PDIUSB11A Pin	Virtex FPGA Pin	Other Functions
/OE	12	
VPO	13	PS/2 CLK
VMO/FSEO	17	PS/2 DATA
RCV	11	
VP	10	
VM	9	
SUSPND	176	SW3

## Parallel Port

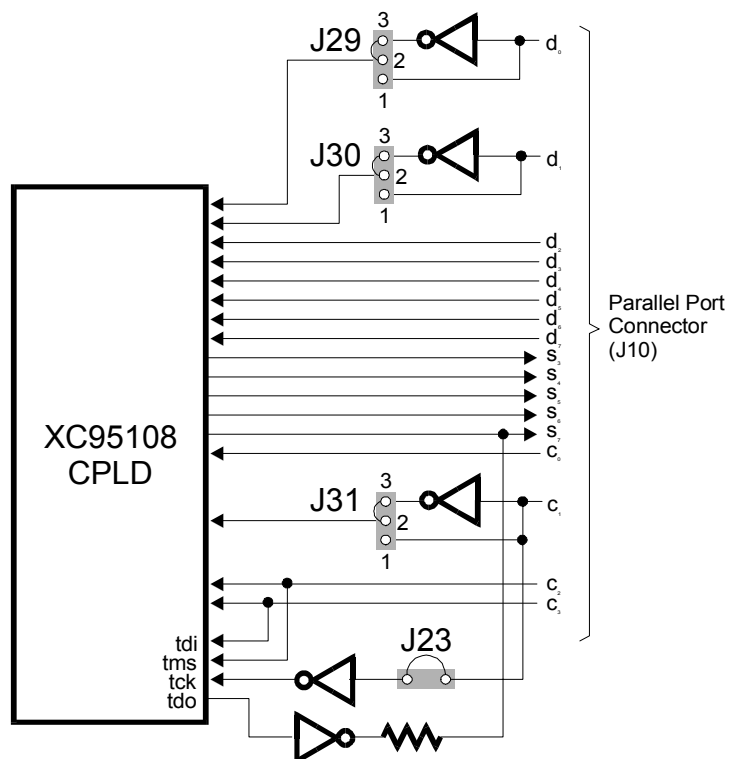
The CPLD handles the interface to the parallel port. The seventeen active lines of the parallel port connect to general-purpose I/O pins on the CPLD.

Four of the parallel port lines also connect to the JTAG pins through which the CPLD is programmed. The TCK signal clocks configuration data in through the TDI pin while the TMS signal steers the actions of the programming state machine. The TDO pin outputs

information back through the parallel port. Removing the shunt from jumper J23 isolates the TCK pin from the parallel port so the CPLD will not be inadvertently reprogrammed during routine parallel port operations. The series resistor prevents the TDO output from interfering with the general-purpose I/O pin during routine parallel port operations.

The CPLD can be programmed to act as an interface between the FPGA and the parallel port (the `dwnldpar.svf` file is an example of such an interface). Schmitt-trigger inverters can be inserted into the  $d_0$ ,  $d_1$ , and  $c_1$  signal lines by placing shunts on pins 2 and 3 of jumpers J29, J30, and J31, respectively. Along with the parallel port interface circuitry in the CPLD, these inverters make the XSV Board compatible with the GXSPORT and GXLOAD software utilities. If your application requires direct access to these signal lines, then you can move the shunts on one or more of these jumpers to pins 1 and 2. But GXLOAD will no longer work if you remove the inverter from the  $c_1$  signal line.

**Be careful if you program the CPLD with outputs that drive the  $c_1$ ,  $c_2$ ,  $c_3$ , and  $s_7$  pins of the parallel port.** These parallel port pins are also connected to the JTAG pins of the CPLD and are used to reprogram the CPLD. **As an ultra-conservative rule-of-thumb, you should never configure CPLD pins 63, 71, 73, or 78 as outputs.** If you must use these pins as outputs, then place them in a high-impedance mode whenever the Virtex FPGA is unconfigured (i.e., the Virtex DONE pin is at logic 0). This will allow you to regain control of the parallel port by toggling the power to the board.



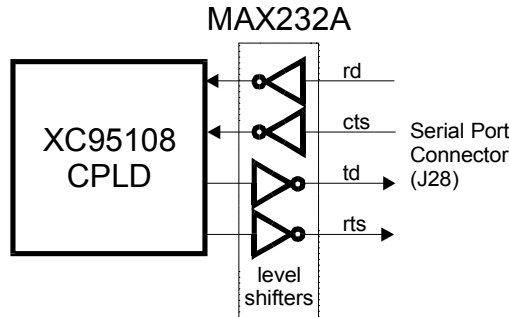
The table below lists the connections from the parallel port to the general-purpose I/O pins of the CPLD:

<b>Parallel Port Pin</b>	<b>XC95108 CPLD Pin</b>
1 (C0)	79
2 (D0)	77
3 (D1)	74
4 (D2)	72
5 (D3)	70
6 (D4)	68
7 (D5)	67
8 (D6)	66
9 (D7)	65
10 (S6)	64
11 (S7)	63
12 (S5)	61
13 (S4)	60
14 (C1)	78
15 (S3)	76
16 (C2)	73
17 (C3)	71

## Serial Port

The CPLD handles the interface to the serial port. The four active lines of the serial port connect to general-purpose I/O pins on the CPLD as follows.

<b>Serial Port Pin</b>	<b>XC95108 CPLD Pin</b>
RTS	82
TD	81
CTS	85
RD	80

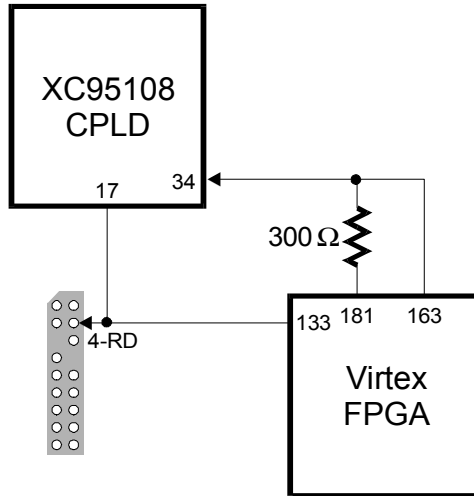


## Xchecker Interface

Header J21 provides an interface between the FPGA and an Xchecker cable. The Xchecker cable can be used to perform configuration and readback operations on the FPGA. To prevent interference, the shunt should be removed from jumper J36 to disconnect the DS1075 and any external clocks from the clock input through the Xchecker cable (CLKI). You must also make sure the CPLD pins in the table below are in a high-impedance state. In addition, the CPLD should place a high logic level on the chip-enable pin of the Flash RAM to prevent the LSB of its data pins from interfering with the DIN pin of the Xchecker interface.

Xchecker Pin	Virtex FPGA Pin	CPLD Pin
1 – VCC (+5V)	N/A	N/A
2 – RT	132	16
3 – GND	N/A	N/A
4 – RD	133	17
6 – TRIG	139	18
7 – CCLK	179	12
9 – DONE	120	10
10 – TDI	167	33
11 – DIN	177	32
12 – TCK	239	4
13 – PROGRAM	122	11
14 – TMS	156	35
15 – INIT	123	9
16 – CLKI	89	22
17 – RST	144	20
18 – CLKO	141	19

If you also want to access the JTAG port of the FPGA, all the requisite pins are already connected to the Xchecker interface except for TDO. The TDO pin of the FPGA connects to pin 34 of the CPLD, so you must route the signal through the CPLD and onto the RD pin of the Xchecker interface (which the Xchecker uses for TDO). You must also make sure that pins 133 and 163 on the FPGA are tristated in your design or else they will override TDO.



You can get a complete Xchecker interface for the CPLD from <http://www.xess.com/appnotes/an-050101-xsvxchk.pdf>.

## Power Connectors

A standard ATX PC power supply can be connected to the XSV Board through connector J11. The connector is keyed so power cannot be applied with the wrong polarity. The shunts should be removed from jumpers J13 and J14 to prevent the 9 VDC converter circuitry from interfering with the ATX power supply. We recommend using the ATX power supply due to its stability and power capacity.

The XSV Board can also be powered from a 9 VDC power supply through jack J12. The power supply must have a 2.1mm, center-positive plug. Two voltage regulators will generate the 5V and 3.3V voltages for the other XSV Board components. Shunts should be placed on jumpers J13 and J14 to connect the outputs from the voltage regulators to the rest of the XSV Board. We do not recommend the 9 VDC power input for general use!

The 2.5V for the Virtex FPGA core logic can be generated on the XSV Board or supplied from an external source. Placing a shunt across pins 1 and 2 of jumper J32 will use the on-board regulator to generate the 2.5V from the 5V supply. You can inject 2.5V from an external source by attaching the positive terminal to pin 2 of jumper J32 and ground to pin 3.

# A

---

## XSV Pin Connections

The following tables list the pin numbers of the Virtex FPGA and the XC95108 CPLD along with the pin names of the other chips that they connect to. These connections correspond with the pin assignments in the user-constraint files VIRTEX.UCF and CPLD.UCF.























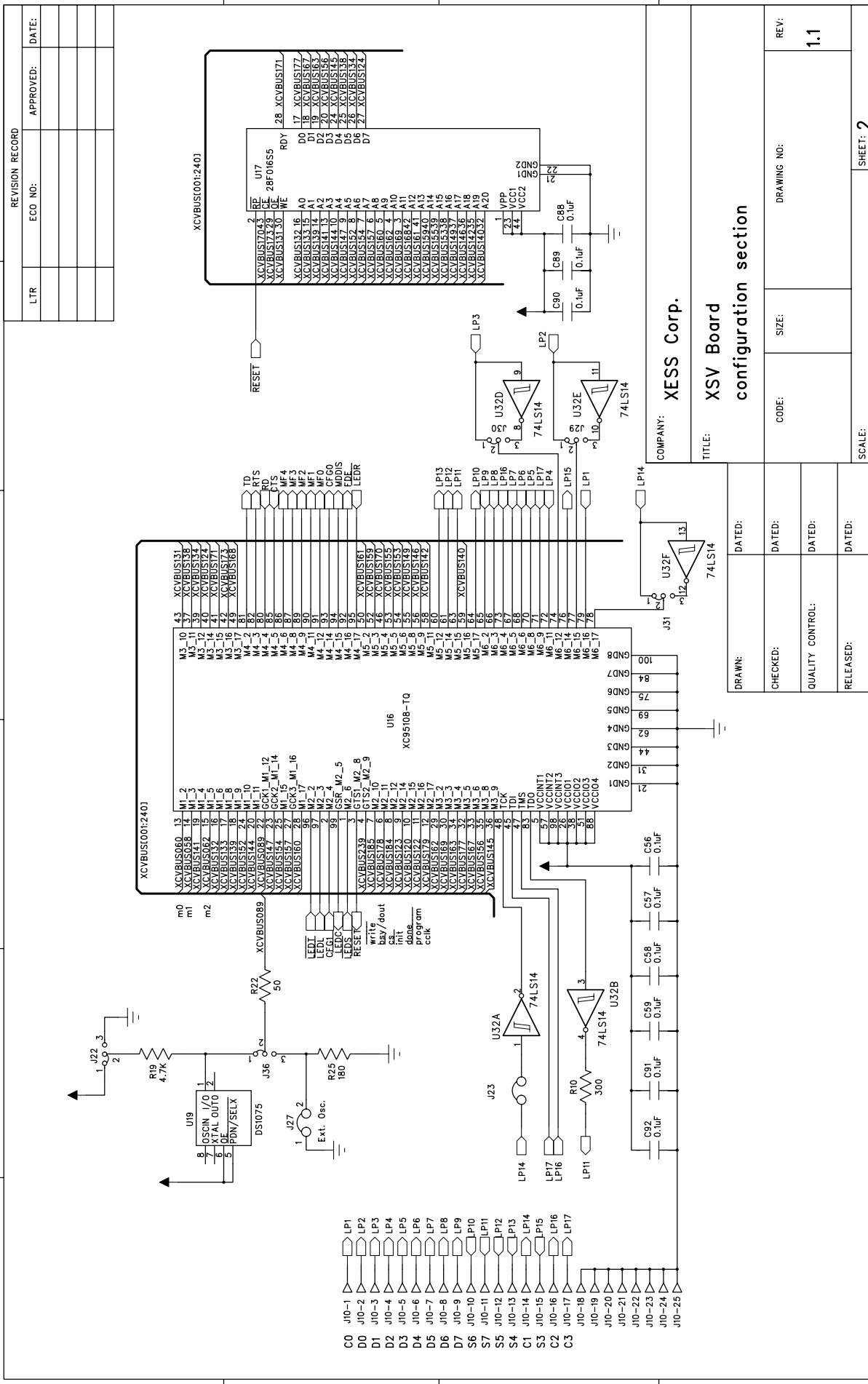


---

# **XSV Schematics**

The following pages show the detailed schematics for the XSV Board.





REVISION RECORD	
LTR	APPROVED:
ECO NO:	DATE:

1 2 3 4 5 6

D C B A

COMPANY: XESS Corp.

TITLE: XSV Board configuration section

DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

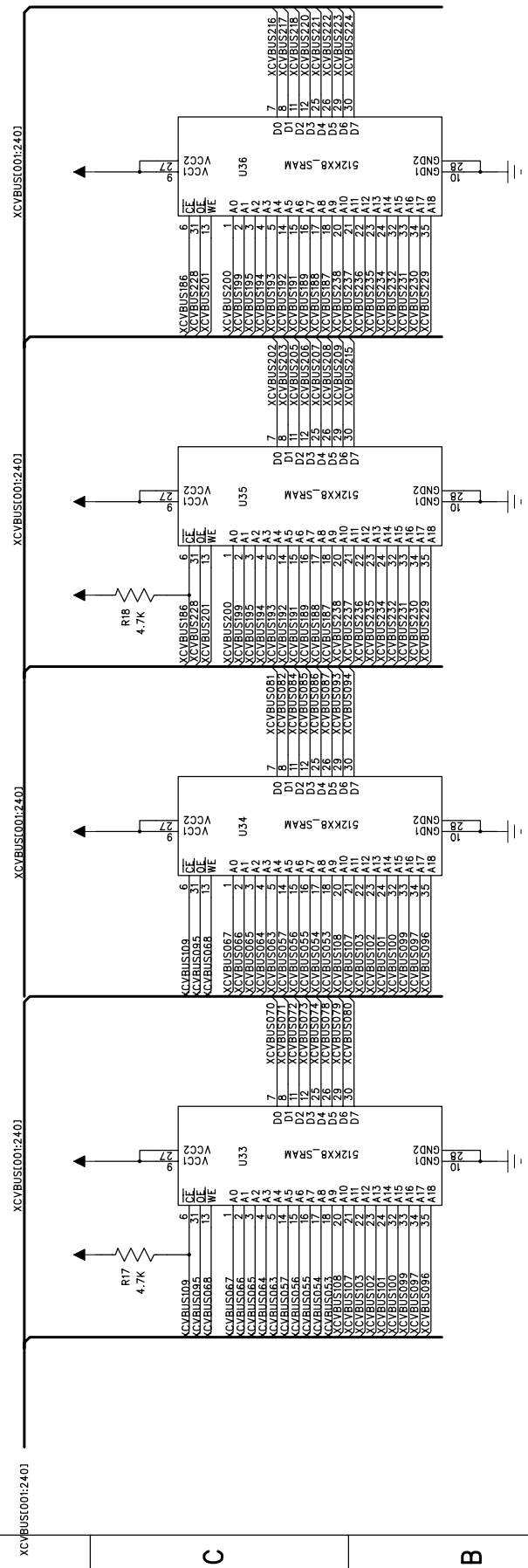
CODE:	SIZE:	DRAWING NO:	REV:
			1.1

SHEET: 2

REVISION RECORD		
LTR	ECO NO:	APPROVED:
		DATE:

D

D

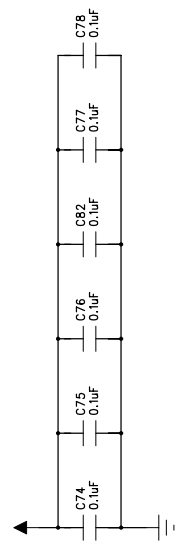


C

C

B

B



COMPANY: XESS Corp.

TITLE: XSV Board  
RAM section

DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

A

A

CODE:	SIZE:	DRAWING NO:	REV:
			1.1
SCALE:		SHEET: 3	

6

5

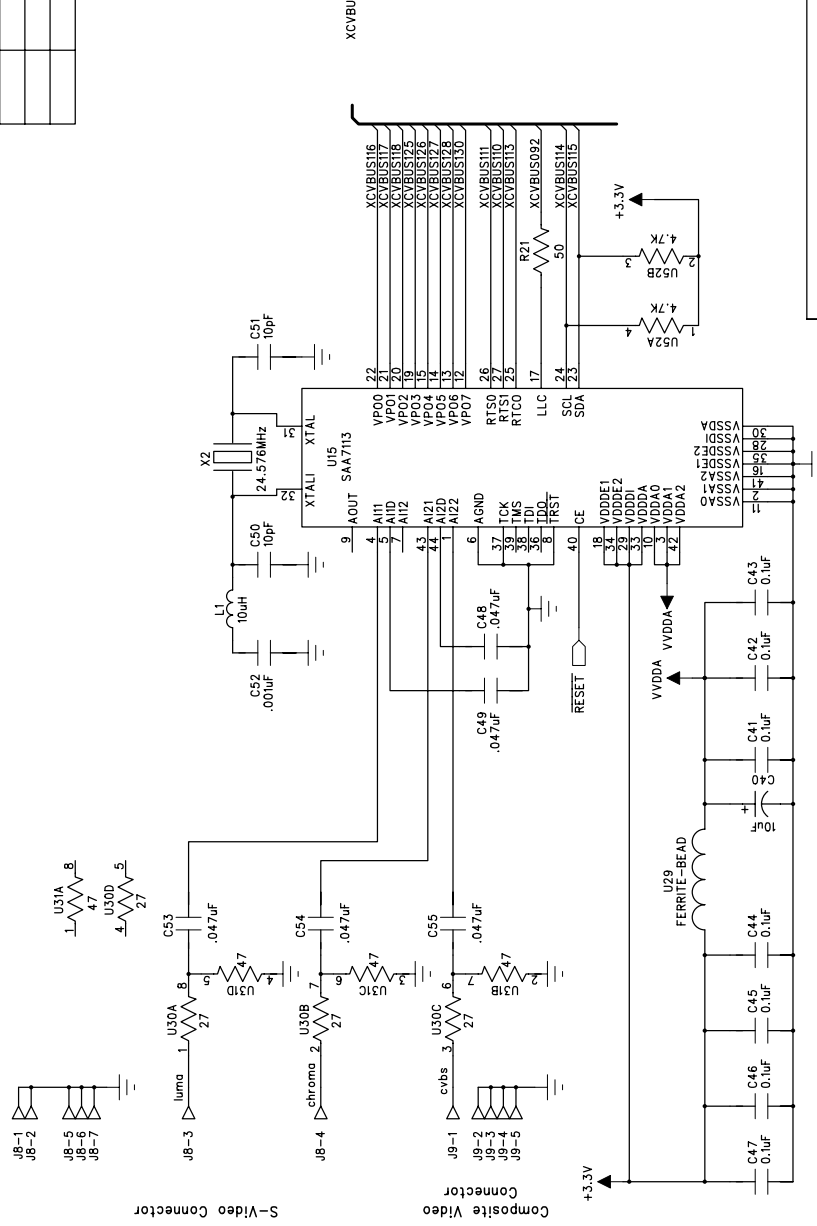
4

3

2

1

REVISION RECORD	
LTR	DATE:



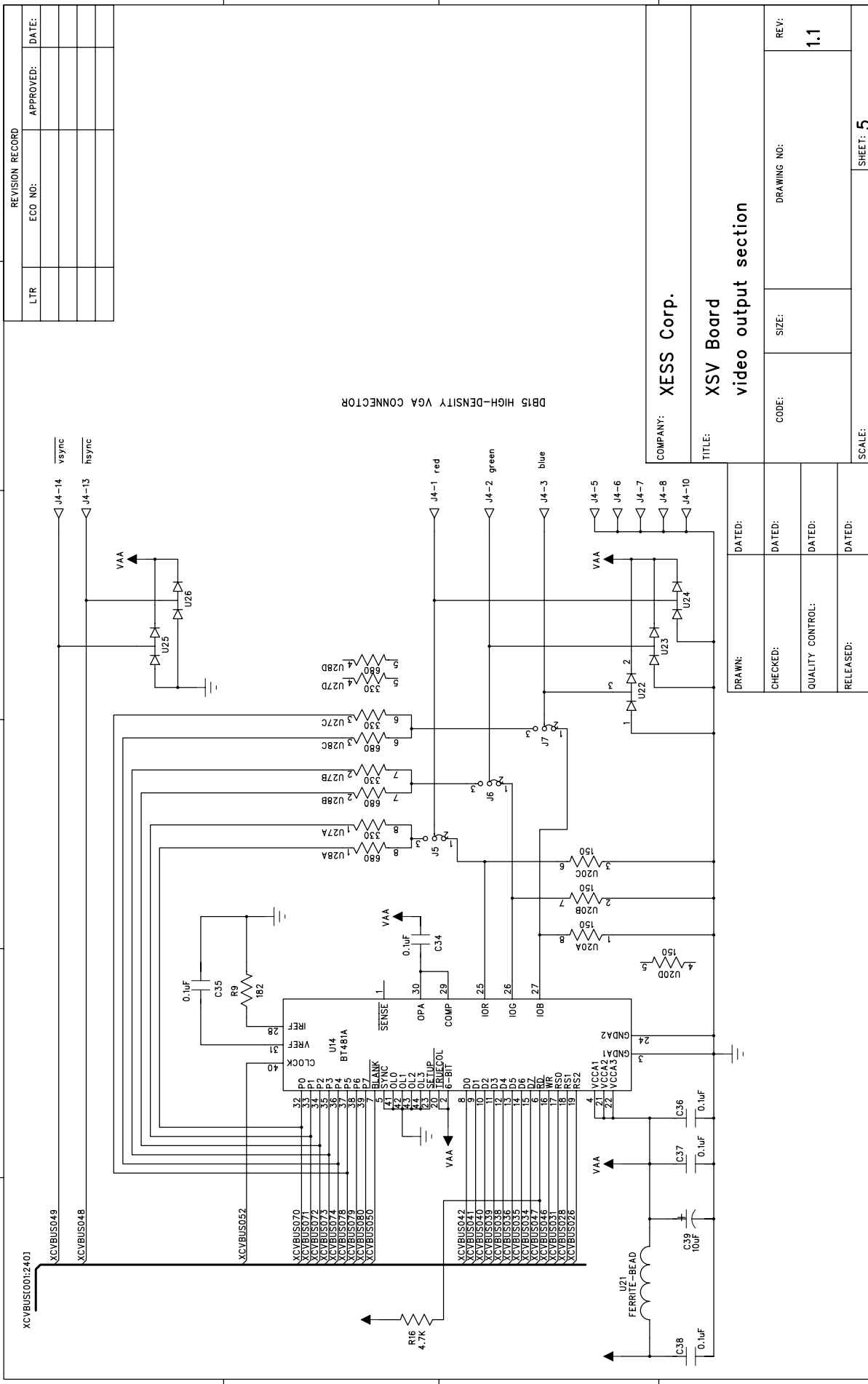
COMPANY: XESS Corp.	
TITLE: XSV Board video decoder section	
DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:
CODE:	SIZE:
DRAWING NO:	REV: 1.1
SCALE: 4	SHEET: 4

D

C

B

A



REVISION RECORD			
LTR	ECO NO:	APPROVED:	DATE:

6 5 4 3 2 1

D C B A

COMPANY: XESS Corp.	
TITLE: XSV Board video output section	
DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:
CODE:	SIZE:
DRAWING NO:	REV: 1.1
SCALE:	SHEET: 5

DB15 HIGH-DENSITY VGA CONNECTOR

FERRITE-BEAD  
U21

1

2

3

4

5

6

REVISION RECORD	
ECO NO:	APPROVED:
LTR	DATE:

D

C

B

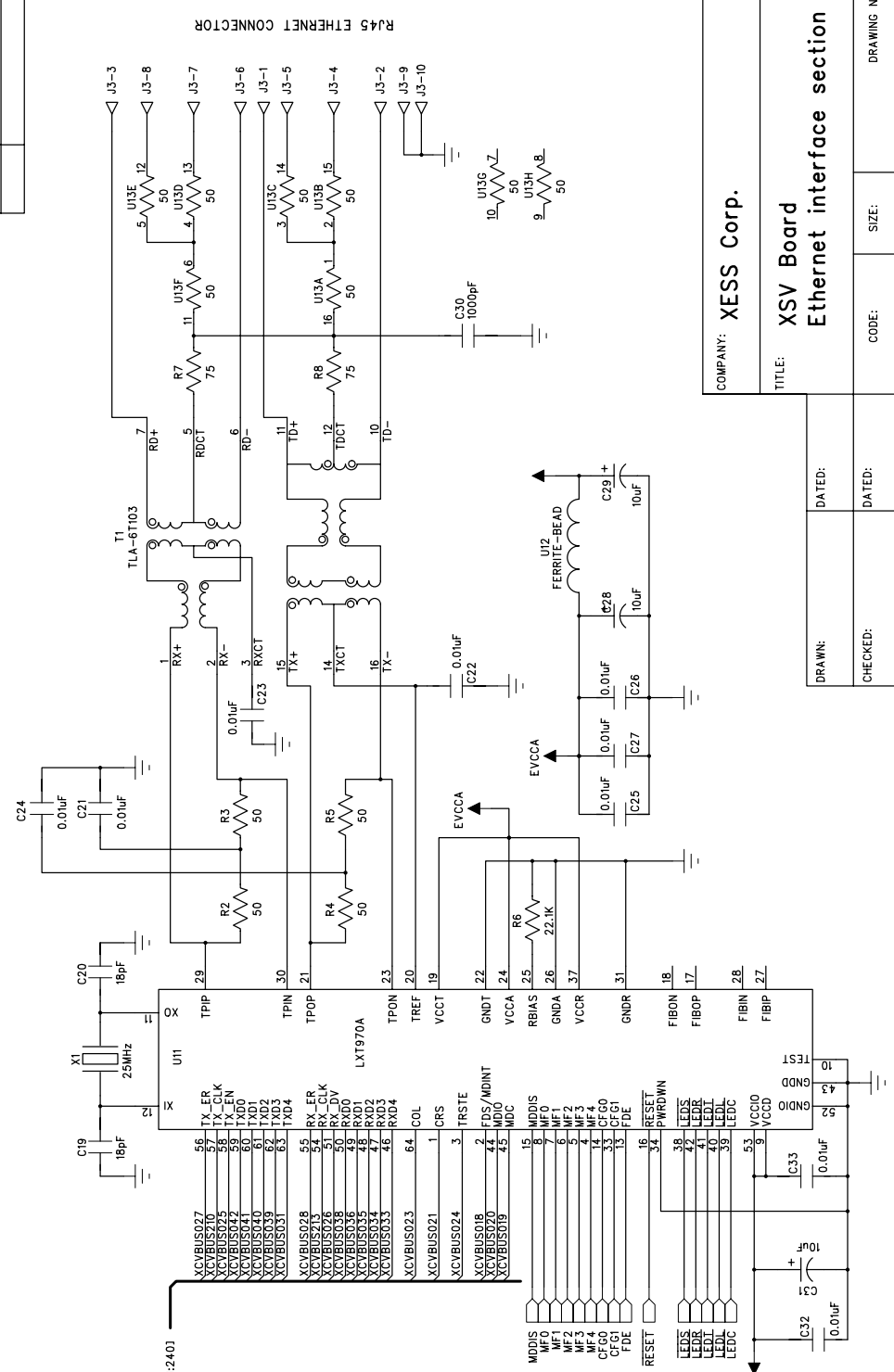
A

D

C

B

A



R445 ETHERNET CONNECTOR

COMPANY: XESS Corp.

TITLE: XSV Board Ethernet interface section

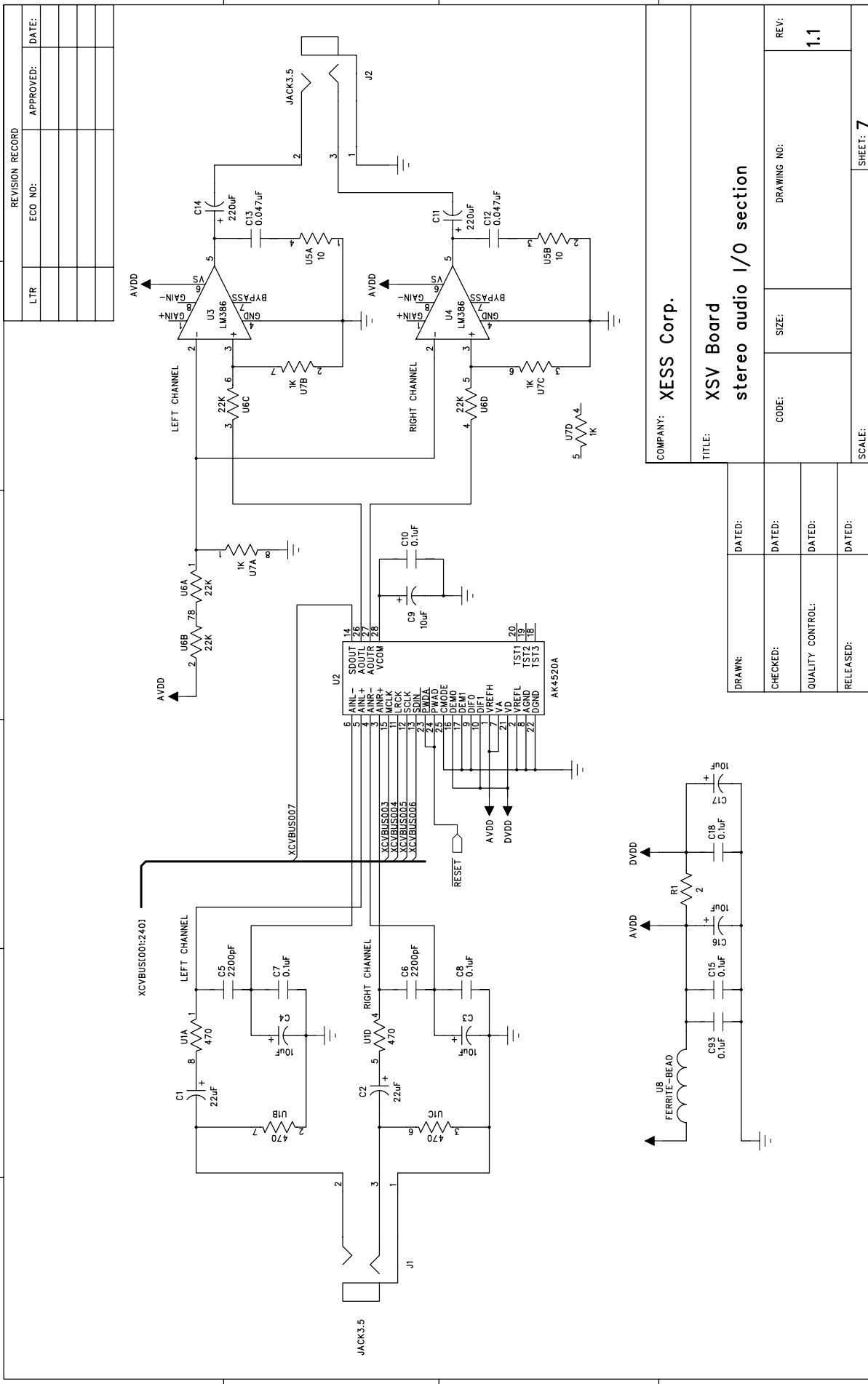
DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

CODE:	SIZE:	DRAWING NO:	REV:
			1.1

SCALE: SHEET: 6

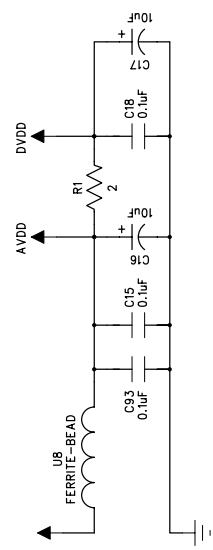


6 5 4 3 2 1



REVISION RECORD	
LTR	APPROVED:
ECO NO:	DATE:

COMPANY: XESS Corp.	
TITLE: XSV Board stereo audio I/O section	
DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:
CODE:	SIZE:
DRAWING NO:	
REV: 1.1	
SCALE:	SHEET: 7



D C B A

D C B A

1

2

3

4

5

6

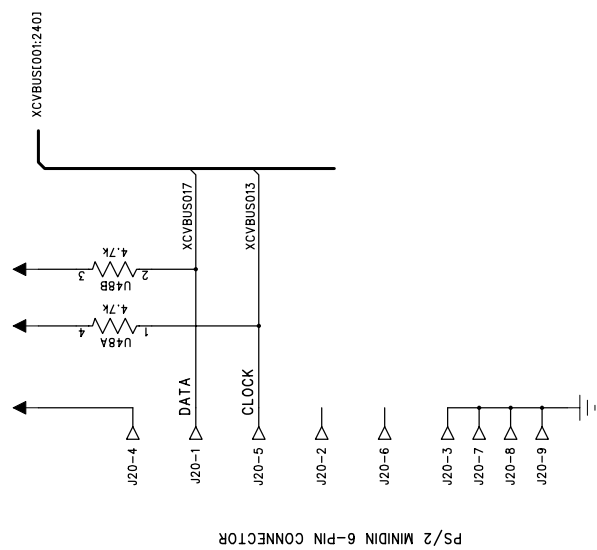
REVISION RECORD			
LTR	ECO NO:	APPROVED:	DATE:

D

C

B

A



PS/2 MINDIN 6-PIN CONNECTOR

COMPANY: XESS Corp.

TITLE: XSV Board PS/2 interface section

DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

REV: 1.1

DRAWING NO:

SIZE:

CODE:

SHEET: 8

SCALE:

D

C

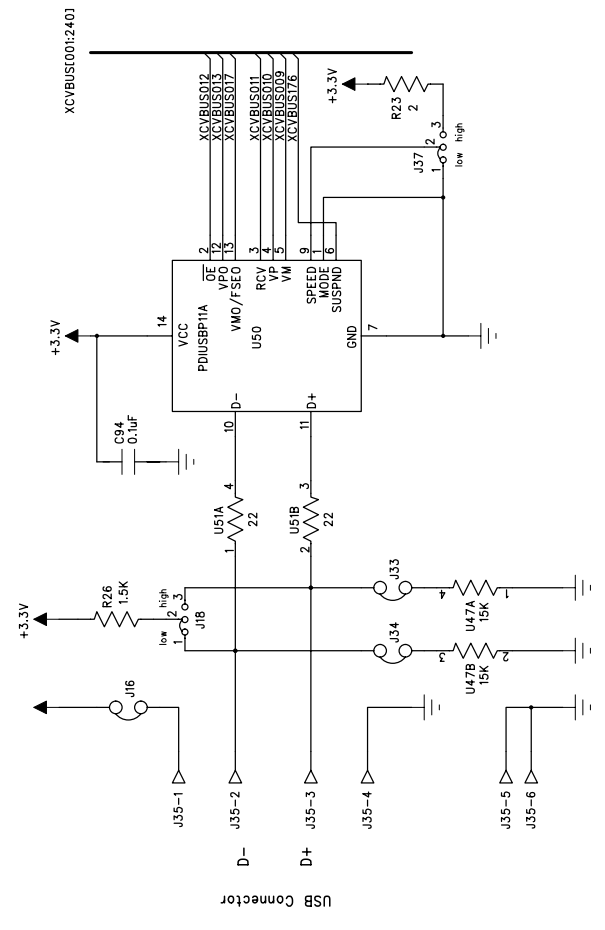
B

A

6 5 4 3 2 1

REVISION RECORD			
LTR	ECO NO:	APPROVED:	DATE:

D C B A



COMPANY: XESS Corp.

TITLE: XSV Board  
USB interface section

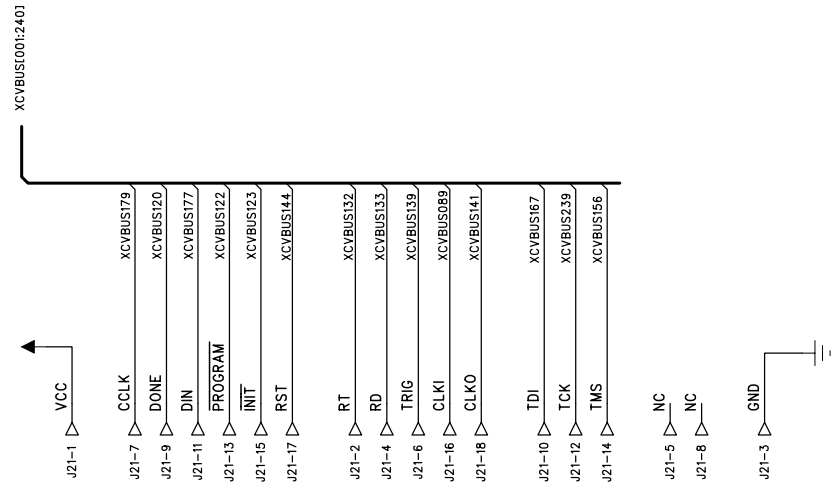
DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

CODE:	SIZE:	DRAWING NO:	REV:
			1.1
SCALE:			SHEET: 9

6 5 4 3 2 1

D C B A

REVISION RECORD			
LTR	ECC NO:	APPROVED:	DATE:



XCHECKER HEADER

COMPANY: XESS Corp.

TITLE: XSV Board  
XCHECKER interface section

DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

CODE:	SIZE:	DRAWING NO:	REV:
			1.1
SCALE:			SHEET: 10

REVISION RECORD		
LTR	ECCO NO:	APPROVED: DATE:

D

C

B

A

XCVBUSI001:2401

XCVBUSI001:2401

- J26-1 ▷ XCVBUSI86
- J26-2 ▷ XCVBUSI87
- J26-3 ▷ XCVBUSI88
- J26-4 ▷ XCVBUSI89
- J26-5 ▷ XCVBUSI90
- J26-6 ▷ XCVBUSI91
- J26-7 ▷ XCVBUSI92
- J26-8 ▷ XCVBUSI93
- J26-9 ▷ XCVBUSI94
- J26-10 ▷ XCVBUSI95
- J26-11 ▷ XCVBUSI96
- J26-12 ▷ XCVBUSI97
- J26-13 ▷ XCVBUSI98
- J26-14 ▷ XCVBUSI99
- J26-15 ▷ XCVBUS200
- J26-16 ▷ XCVBUS201
- J26-17 ▷ XCVBUS202
- J26-18 ▷ XCVBUS203
- J26-19 ▷ XCVBUS204
- J26-20 ▷ XCVBUS205
- J26-21 ▷ XCVBUS206
- J26-22 ▷ XCVBUS207
- J26-23 ▷ XCVBUS208
- J26-24 ▷ XCVBUS209
- J26-25 ▷ XCVBUS210
- J26-26 ▷ XCVBUS211
- J26-27 ▷ XCVBUS212
- J26-28 ▷ XCVBUS213
- J26-29 ▷ XCVBUS214
- J26-30 ▷ XCVBUS215
- J26-31 ▷ XCVBUS216
- J26-32 ▷ XCVBUS217
- J26-33 ▷ XCVBUS218
- J26-34 ▷ XCVBUS219
- J26-35 ▷ XCVBUS220
- J26-36 ▷ XCVBUS221
- J26-37 ▷ XCVBUS222
- J26-38 ▷ XCVBUS223
- J26-39 ▷ XCVBUS224
- J26-40 ▷ XCVBUS225
- J26-41 ▷ XCVBUS226
- J26-42 ▷ XCVBUS227
- J26-43 ▷ XCVBUS228
- J26-44 ▷ XCVBUS229
- J26-45 ▷ XCVBUS230
- J26-46 ▷ XCVBUS231
- J26-47 ▷ XCVBUS232
- J26-48 ▷ XCVBUS233
- J26-49 ▷ XCVBUS234
- J26-50 ▷ XCVBUS235

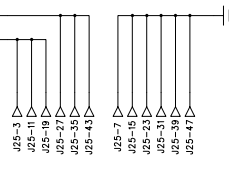
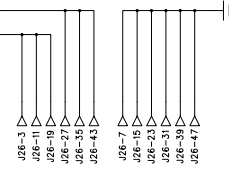
EXPANSION HEADER

- J25-1 ▷ XCVBUS109
- J25-2 ▷ XCVBUS110
- J25-3 ▷ XCVBUS111
- J25-4 ▷ XCVBUS112
- J25-5 ▷ XCVBUS113
- J25-6 ▷ XCVBUS114
- J25-7 ▷ XCVBUS115
- J25-8 ▷ XCVBUS116
- J25-9 ▷ XCVBUS117
- J25-10 ▷ XCVBUS118
- J25-11 ▷ XCVBUS119
- J25-12 ▷ XCVBUS120
- J25-13 ▷ XCVBUS121
- J25-14 ▷ XCVBUS122
- J25-15 ▷ XCVBUS123
- J25-16 ▷ XCVBUS124
- J25-17 ▷ XCVBUS125
- J25-18 ▷ XCVBUS126
- J25-19 ▷ XCVBUS127
- J25-20 ▷ XCVBUS128
- J25-21 ▷ XCVBUS129
- J25-22 ▷ XCVBUS130
- J25-23 ▷ XCVBUS131
- J25-24 ▷ XCVBUS132
- J25-25 ▷ XCVBUS133
- J25-26 ▷ XCVBUS134
- J25-27 ▷ XCVBUS135
- J25-28 ▷ XCVBUS136
- J25-29 ▷ XCVBUS137
- J25-30 ▷ XCVBUS138
- J25-31 ▷ XCVBUS139
- J25-32 ▷ XCVBUS140
- J25-33 ▷ XCVBUS141
- J25-34 ▷ XCVBUS142
- J25-35 ▷ XCVBUS143
- J25-36 ▷ XCVBUS144
- J25-37 ▷ XCVBUS145
- J25-38 ▷ XCVBUS146
- J25-39 ▷ XCVBUS147
- J25-40 ▷ XCVBUS148
- J25-41 ▷ XCVBUS149
- J25-42 ▷ XCVBUS150
- J25-43 ▷ XCVBUS151
- J25-44 ▷ XCVBUS152
- J25-45 ▷ XCVBUS153
- J25-46 ▷ XCVBUS154
- J25-47 ▷ XCVBUS155
- J25-48 ▷ XCVBUS156
- J25-49 ▷ XCVBUS157
- J25-50 ▷ XCVBUS158

EXPANSION HEADER

+3.3V

+3.3V



COMPANY: XESS Corp.	
TITLE: XSV Board expansion interface section	
DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:
CODE:	SIZE:
DRAWING NO:	REV:
	1.1
SCALE:	SHEET: 11

D

C

B

A

1

2

3

4

5

6

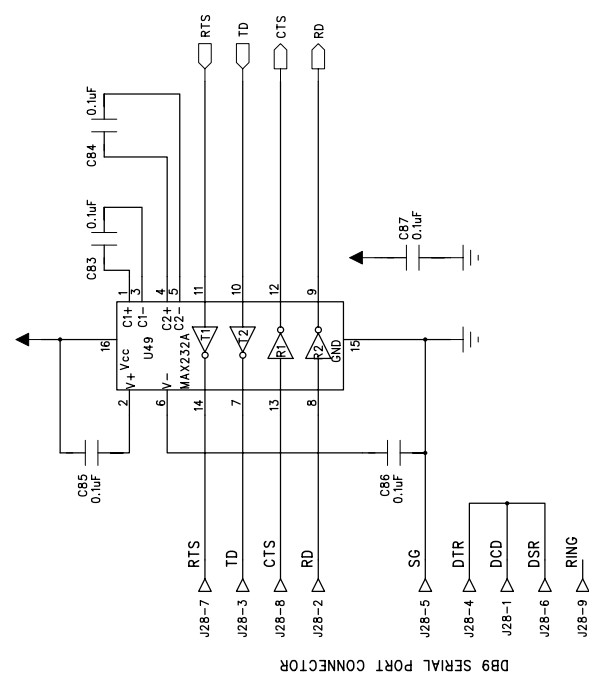
REVISION RECORD			
LTR	ECO NO:	APPROVED:	DATE:

D

C

B

A



COMPANY: XESS Corp.

TITLE: XSV Board serial port interface

DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

REV: 1.1

DRAWING NO:

CODE:

SIZE:

SCALE:

SHEET: 12

D

C

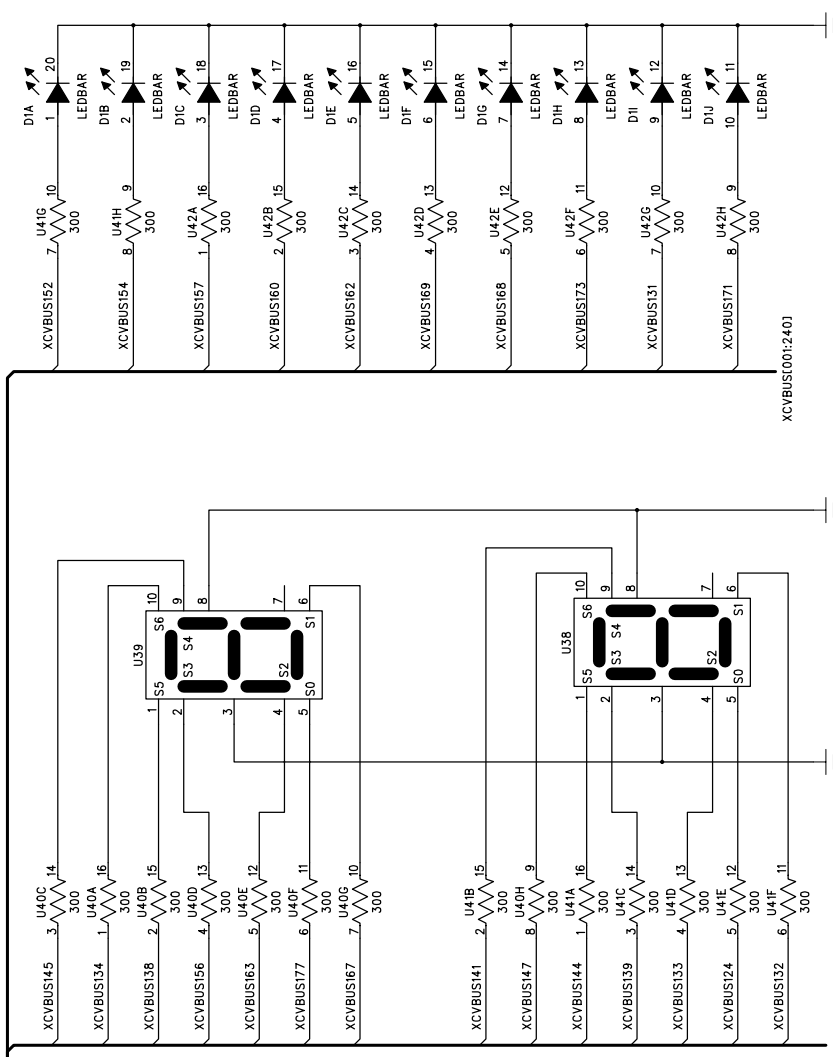
B

A

6 5 4 3 2 1

REVISION RECORD		
LTR	ECO NO:	APPROVED:

XCVBUS001:2401



COMPANY: XESS Corp.

TITLE: XSV Board LED section

DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

CODE:	SIZE:	DRAWING NO:	REV:
			1.1
SCALE:			SHEET: 13

D C B A

1

2

3

4

5

6

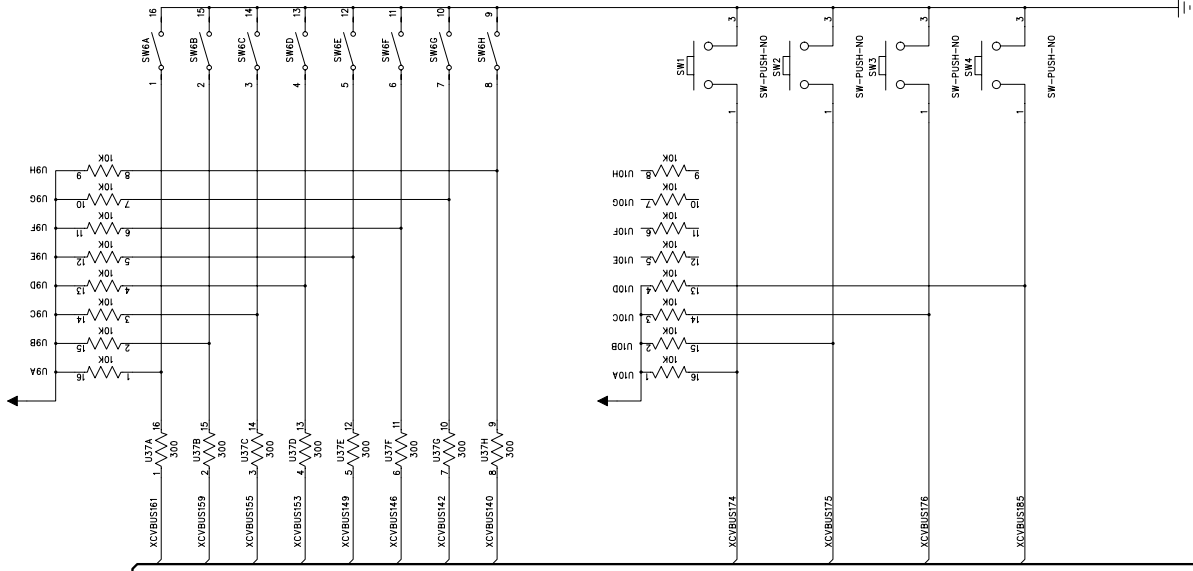
D

C

B

A

REVISION RECORD		
LTR	ECCO NO:	APPROVED: DATE:



COMPANY: XESS Corp.	
TITLE: XSV Board	
dip switches and pushbuttons	
DATED:	DRAWING NO:
CHECKED:	CODE:
QUALITY CONTROL:	SIZE:
RELEASED:	DATED:
SCALE: 1:1	SHEET: 14

D

C

B

A

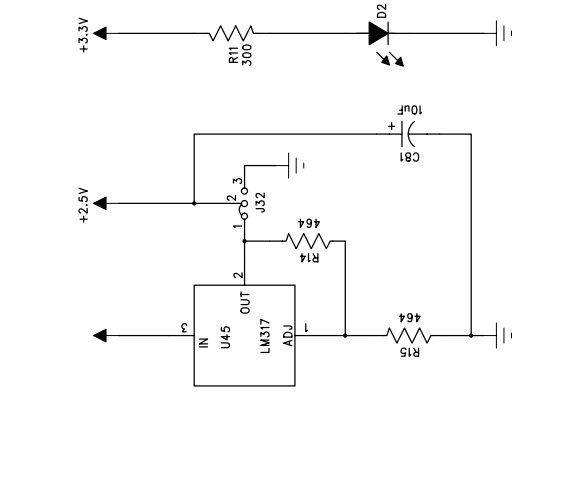
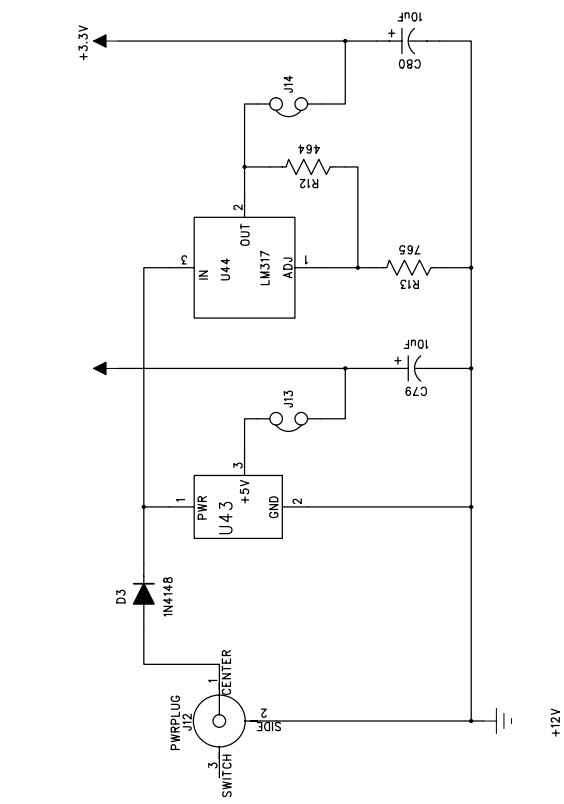
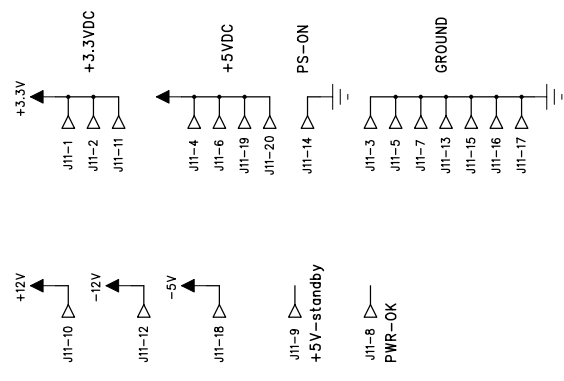


6 5 4 3 2 1

REVISION RECORD	
LTR	APPROVED:
ECO NO:	DATE:

D C B A

D C B A



ATX Power Connector

9VDC Power Plug

Cooling Fan Connector

COMPANY: XESS Corp.

TITLE: XSV Board power input & regulation

DRAWN:	DATED:
CHECKED:	DATED:
QUALITY CONTROL:	DATED:
RELEASED:	DATED:

CODE:	SIZE:	DRAWING NO:	REV:
			1.1
SCALE:			SHEET: 15