



XSA-3S1000 Board V1.0 User Manual

How to install, test, and use
your new XSA-3S1000 Board

Copyright © 2001-2005 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

Table of Contents	2
Preliminaries	4
Getting Help!	4
Take notice!!	4
Packing List	5
Installation	6
Installing the XSTOOLS Utilities and Documentation	6
Applying Power to Your XSA-3S1000 Board	6
Using a 5V - 9V DC wall-mount power supply	6
Powering Through the PS/2 Connector	6
Solderless Protoboard Installation.....	7
Connecting a PC to Your XSA-3S1000 Board.....	8
Connecting a VGA Monitor to Your XSA-3S1000 Board.....	8
Connecting a Mouse or Keyboard to Your XSA-3S1000 Board	9
Inserting the XSA-3S1000 Board into an XStend Board	9
Setting the Jumpers on Your XSA-3S1000 Board	9
Testing Your XSA-3S1000 Board	9
Setting the XSA-3S1000 Board Clock Oscillator Frequency.....	10
Programming	11
Generating Bitstreams for the FPGA	11
Downloading Bitstreams into the FPGA and CPLD	12
Downloading Using GXSLD.....	12
Downloading Using Xilinx iMPACT	15
Storing Non-Volatile Bitstreams in the Flash	15
Downloading and Uploading Data to the SDRAM	18

Programmer's Models	20
XSA-3S1000 Board Organization	20
Programmable logic: FPGA and CPLD	21
100 MHz Fixed-Frequency Oscillator.....	22
Synchronous DRAM	23
Flash RAM.....	23
Seven-Segment LED	25
DIP Switches and Pushbuttons.....	25
PS/2 Port.....	26
VGA Port.....	26
Parallel Port	27
XILINX Parallel Cable IV Connector	30
Prototyping Header	30
XSA-3S1000 Pin Connections	32
XSA-3S1000 Schematics.....	33

1

Preliminaries

Getting Help!

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

- If you can't get the XSA-3S1000 Board hardware to work, send an e-mail message describing your problem to help@xess.com or submit a problem report at <http://www.xess.com/help.html>. Our web site also has
 - [answers to frequently-asked-questions](#),
 - [example designs, application notes and tutorials for the XS Boards](#),
 - [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 WebPACK software tools installed properly, send an e-mail message describing your problem to hotline@XILINX.com or check their web site at <http://www.xilinx.com/support/support.htm>.
- If you need help using the WebPACK software to create designs for your XSA-3S1000 Board, then check out this [tutorial](#).

Take notice!!

- The XSA-3S1000 is not 5V-tolerant. Do not connect 5V logic signals to the prototyping header. Do not insert your XSA-3S1000 Board into an XST-1 or XST-2.x XStend Board. These versions of the XStend Boards contain some 5V logic.
- The XSA-3S1000 Board requires an external power supply to operate! It does not draw power through the downloading cable from the PC parallel port.
- If you are connecting a 5V - 9V DC power supply to your XSA-3S1000 Board, please make sure the center terminal of the plug is positive and the outer sleeve is negative.
- Do not power your XSA-3S1000 Board with a battery! This will not provide enough current to insure reliable operation of the XSA-3S1000 Board.
- Even if you have experience with the XILINX software tools, please read this [section on setting the bitstream generation options for the XSA-3S1000 Board](#).

Packing List

Here is what you should have received in your package:

- an XSA-3S1000 Board;
- a 6' cable with a 25-pin male connector on each end;
- an XSTOOLS CDROM with software utilities and documentation for using the XSA-3S1000 Board.

2

Installation

Installing the XSTOOLS Utilities and Documentation

XILINX currently provides the WebPACK tools for programming their CPLDs and Spartan-series FPGAs. The XESS CDROM contains a version of WebPACK that will generate bitstream configuration files compatible with your XSA-3S1000 Board. You can also [download](#) the most current version of the WebPACK tools from the XILINX website.

In addition, XESS Corp. provides the XSTOOLS utilities for interfacing a PC to your XSA-3S1000 Board. These utilities should be installed automatically when you insert the XSTOOLS CDROM into your CDROM drive. If not, then manually run the SETUP.EXE installation program on the CDROM.

Applying Power to Your XSA-3S1000 Board

You can use your XSA-3S1000 Board in three ways, distinguished by the method you use to apply power to the board. **Only use one of these methods to power your XSA-3S1000 Board!** Supplying power from multiple sources can damage the board and/or power supplies.

Using a 5V - 9V DC wall-mount power supply

You can use your XSA-3S1000 Board all by itself to experiment with logic designs. Just place the XSA-3S1000 Board on a non-conducting surface as shown in Figure 1. Then apply power to the XSA-3S1000 Board from a DC wall-mount power supply with a 2.1 mm female, center-positive plug and a voltage in the range 5V – 9V. (See Figure 2 for the location of the 5V – 9V DC power jack on your XSA-3S1000 Board.) The shunt on jumper J7 should be in the “9V” position. The on-board voltage regulation circuitry will create the voltages required by the rest of the XSA-3S1000 Board circuitry. **Be careful!! The voltage regulator on the XSA-3S1000 Board can become hot.** To reduce the power dissipation in the regulator and keep it cool, use a power supply with a voltage near 5V. You can also attach a heat sink to the regulator if necessary.

Powering Through the PS/2 Connector

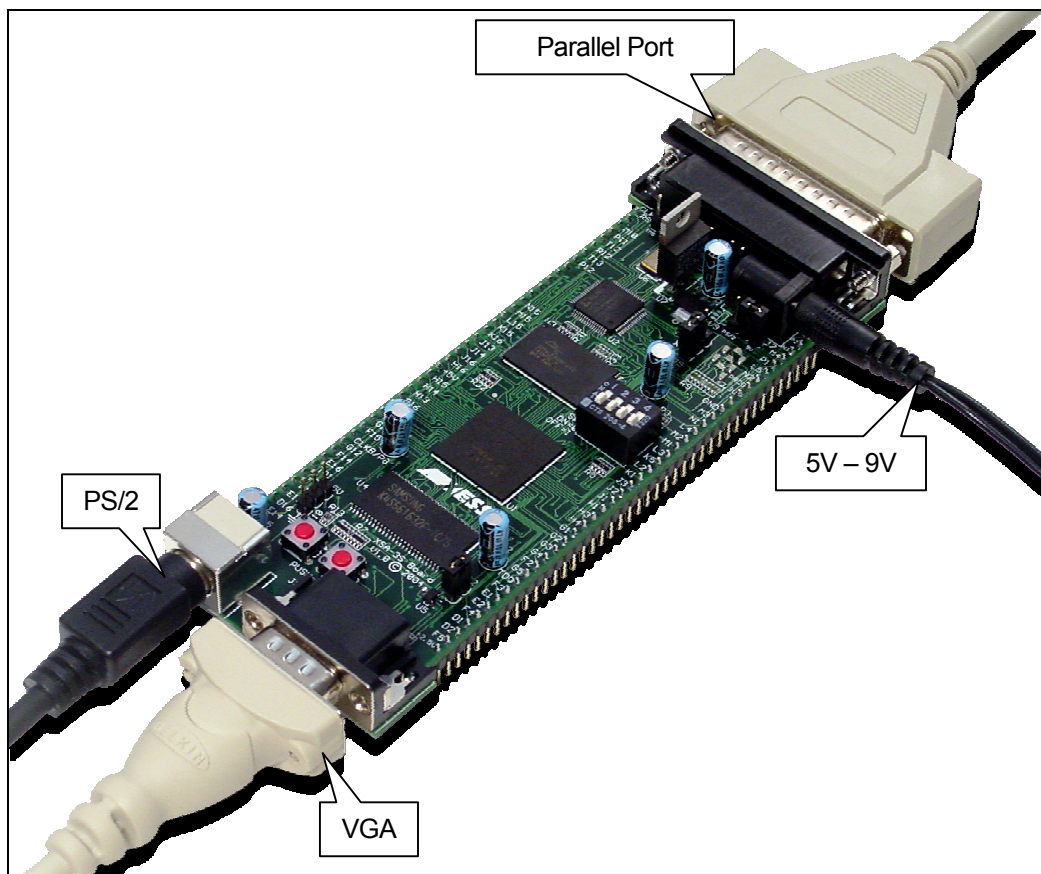
You can use your XSA-3S1000 Board with a laptop PC by connecting a PS/2 male-to-male cable between the PS/2 ports of the laptop and the board. The shunt on jumper J7 should be in the “PS/2” position. The on-board voltage regulation circuitry will create the voltages required by the rest of the XSA-3S1000 Board circuitry. **Many PS/2 ports cannot supply more than 0.5A so large, high-frequency FPGA designs may not work when using this power source!**

Solderless Protoboard Installation

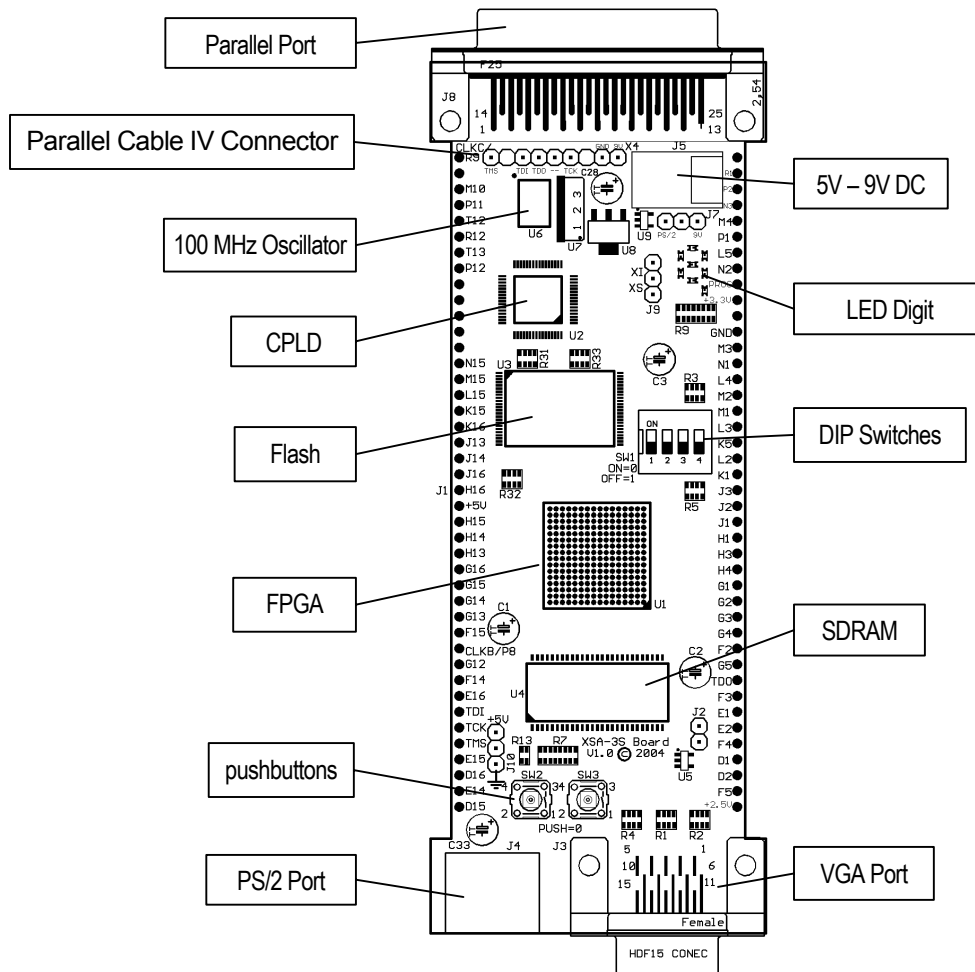
The two rows of pins from your XSA-3S1000 Board can be plugged into a solderless protoboard with holes spaced at 0.1" intervals. (One of the A.C.E. protoboards from 3M is a good choice.) Once plugged in, many of the pins of the FPGA are accessible to other circuits on the protoboard. (The labels printed next to the rows of pins on your XSA-3S1000 Board correspond to the pin numbers of the FPGA.) Power can still be supplied to your XSA-3S1000 Board through the 5V – 9V DC jack, or power can be applied directly through several pins on the underside of the board. Just connect +5V, +3.3V, +2.5V and ground to the pins of your XSA-3S1000 Board listed in Table 1. (Remove the shunt on jumper J2 if you supply +2.5V from an external source.)

• Table 1: Power supply pins for the XSA-3S1000 Board.

Voltage	Pin	Note
+5V	2	This pin is labeled "+5V".
+3.3V	54	This pin is labeled "+3.3V".
+2.5V	22	This pin is labeled "+2.5V".
GND	52	This pin is labeled "GND".



• Figure 1: External connections to the XSA-3S1000 Board.



• Figure 2: Arrangement of components on the XSA-3S1000 Board.

Connecting a PC to Your XSA-3S1000 Board

The 6' DB25 male-to-male cable included with your XSA-3S1000 Board connects it to a PC. One end of the cable attaches to the parallel port on the PC and the other connects to the female DB-25 connector at the top of the XSA-3S1000 Board as shown in Figure 1.

Connecting a VGA Monitor to Your XSA-3S1000 Board

You can display images with up to 512 colors on a VGA monitor by connecting it to the VGA port at the bottom of your XSA-3S1000 Board (see Figure 1). You will have to create a VGA display circuit for your XSA-3S1000 Board to actually display an image. See [this section](#) for details on the VGA port circuitry and an example of a VGA display circuit.

Connecting a Mouse or Keyboard to Your XSA-3S1000 Board

You can accept inputs from a keyboard or mouse by connecting it to the PS/2 port at the bottom of your XSA-3S1000 Board (see Figure 1). You will have to create a keyboard or mouse interface circuit to actually receive information on keystrokes or mouse movements. See [this section](#) for details on the PS/2 port circuitry and an example of a keyboard interface.

Inserting the XSA-3S1000 Board into an XStend Board

Do not insert an XSA-3S1000 Board into an XST-1 or XST-2.x XStend Board. The XSA-3S1000 Board is not compatible with the voltages on these versions of the XStend Board. An XST-3.0 XStend Board will be available in 2005.

Setting the Jumpers on Your XSA-3S1000 Board

The default jumper settings shown in Table 2 configure your XSA-3S1000 Board for use in a logic design environment. You will need to change the jumper settings only if you are:

- downloading FPGA bitstreams to your XSA-3S1000 Board using the XILINX iMPACT software;
- changing the power sources for the XSA-3S1000 supply voltages.

• Table 2: Jumper settings for XSA-3S1000 Boards.

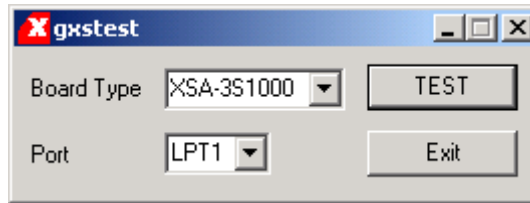
Jumper	Setting	Purpose
J2	On (default)	A shunt should be installed if the +2.5V supply voltage is derived from the +3.3V supply.
	Off	The shunt should be removed if the +2.5V supply voltage is applied from an external source through pin 22 of the XSA-3S1000 Board (labeled "+2.5V" at the lower right-hand corner of the board).
J7	1-2 (9V)	The shunt should be installed on pins 1 and 2 if the board supply voltages are derived from the an external 5V – 9V DC supply.
	2-3 (PS/2)	The shunt should be installed on pins 2 and 3 if the board supply voltages are derived from the +5V supply applied through the PS/2 keyboard/mouse connector (J4).
J9	1-2 (XI)	The shunt should be installed on pins 1 and 2 (XI) if the XSA-3S1000 Board is to be downloaded using the XILINX iMPACT software.
	2-3 (XS) (default)	The shunt should be installed on pins 2 and 3 (XS) if the XSA-3S1000 Board is to be downloaded using the XESS GXLOAD software.
J10	N/A	This is a header that provides access to the +5V and GND references on the board. No shunt should be placed on this header.

Testing Your XSA-3S1000 Board

Once your XSA-3S1000 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 XSA-3S1000 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, pick the XSA-3S1000 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 XSA-3S1000 Board. Within thirty 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 XSA-3S1000 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 try some of the solutions listed in the XSTOOLS\README.TXT file. 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 XSA-3S1000 Board, the CPLD is programmed with the standard parallel port interface found in the XSTOOLS\XSA\3S1000\dwlnldpar.svf bitstream file. This is the interface that should be loaded into the CPLD when you want to use it with the GXLOAD utility.

Setting the XSA-3S1000 Board Clock Oscillator Frequency

Unlike previous versions of the XSA Board, your XSA-3S1000 Board has a fixed-frequency oscillator of 100 MHz. The GXSETCLK utility cannot be used to change the frequency of the clock sent to the FPGA and CPLD. You can lower the clock frequency by placing a clock-divider circuit in the FPGA or CPLD. See the [section on the XSA-3S1000 Board clock circuitry](#) for more details.

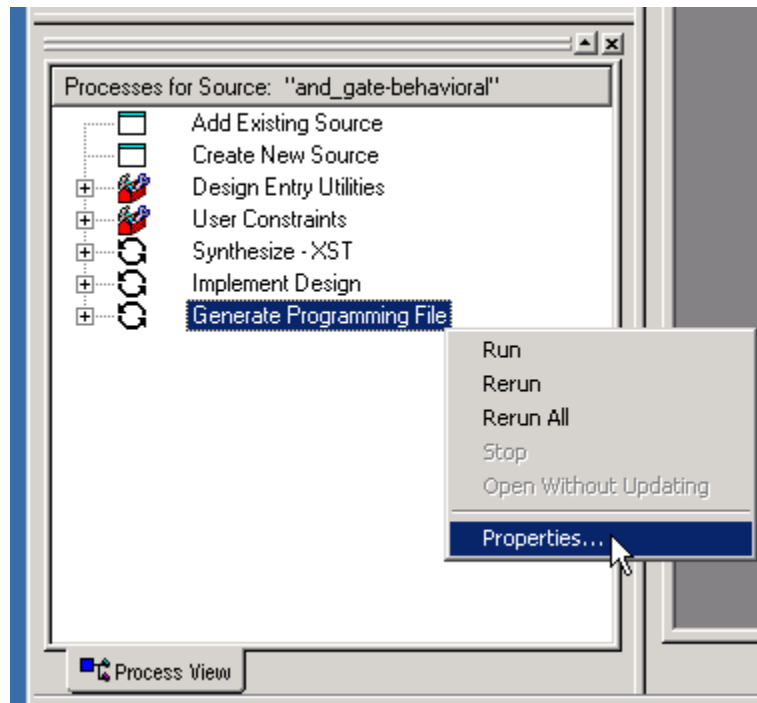
3

Programming

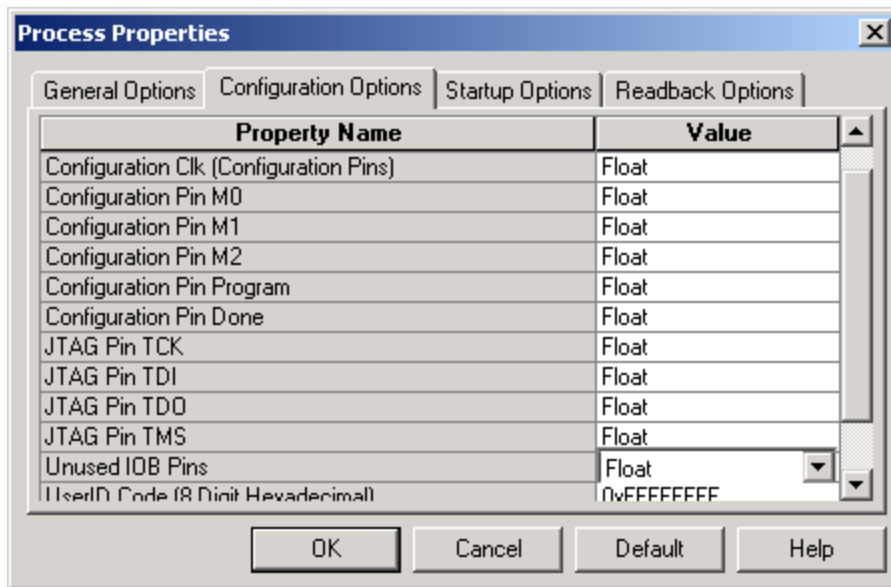
This section will show you how to download logic designs into the FPGA and CPLD of your XSA-3S1000 Board and how to download and upload data to and from the SDRAM and Flash devices on the board.

Generating Bitstreams for the FPGA

Before downloading a bitstream to the FPGA on your XSA-3S1000 Board, you will use the XILINX development software to generate the .BIT file. Steps for doing this are given in the XILINX documentation, but there is one detail that is specific to your board. The Spartan3 FPGA has relatively low-impedance, internal pullup and pulldown resistors on its pins that become active after the FPGA is configured. These resistors can overpower the 4.7K Ω external pullup and pulldown resistors on the board and prevent it from functioning correctly. Therefore, you have to set the bitstream generation options to disable these internal resistors. This is done by right-clicking on the Generate Programming File item in the Processes pane and selecting Properties... from the pop-up menu.



Then select the Configuration Options tab in the Process Properties window and change all the pin settings to float. Then click on the OK button. The internal pullup and pulldown resistors will now be disabled in the bitstream generated for this project.



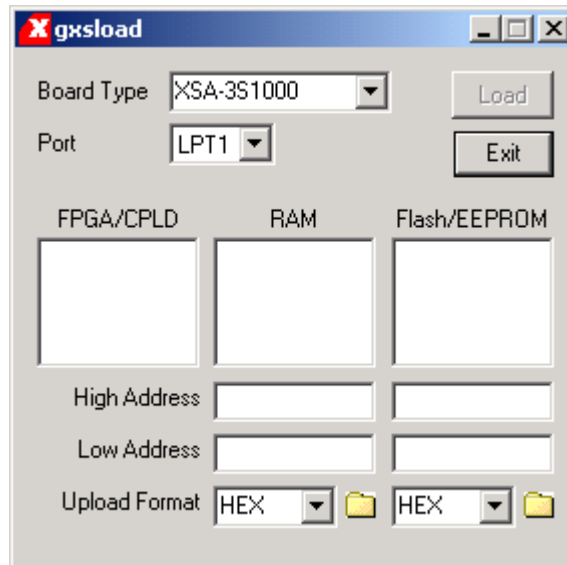
Downloading Bitstreams into the FPGA and CPLD

Downloading Using GXSLOAD

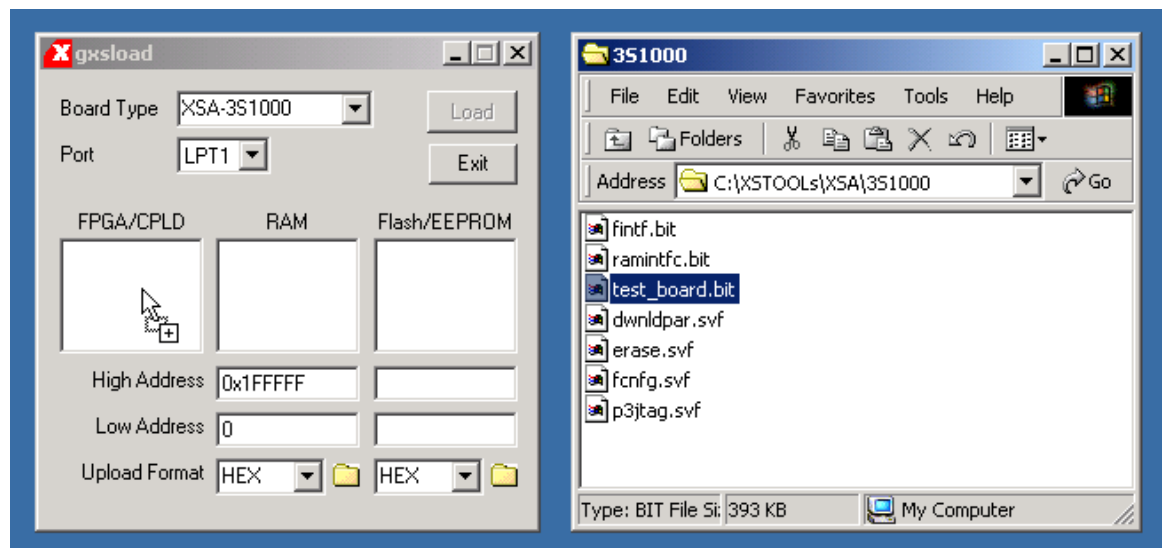
As you develop and test a logic design, you will usually connect the XSA-3S1000 Board to the parallel port of a PC and download the configuration bitstream each time you make changes. You can download a bitstream into your XSA-3S1000 Board using the GXSLOAD utility.



You start GXSLoad by clicking on the **GXSLoad** icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below. Select the XSA-3S1000 Board and the parallel port to which it is connected as indicated below.

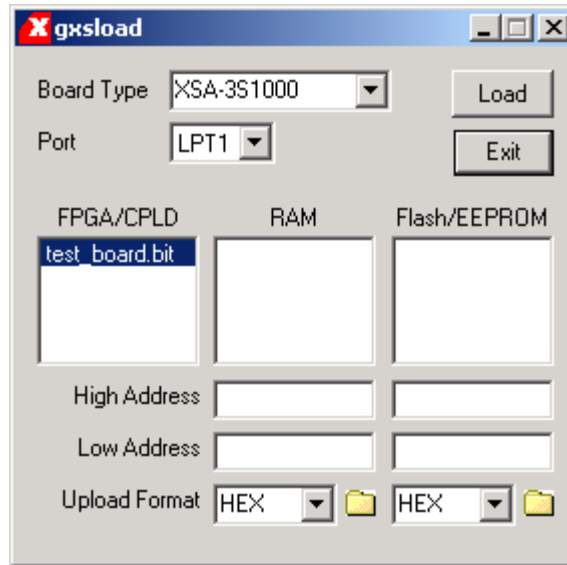


Now you can download bitstream files to the FPGA or CPLD simply by dragging them from their folder and dropping them into the FPGA/CPLD pane of the GXSLoad window as shown below.

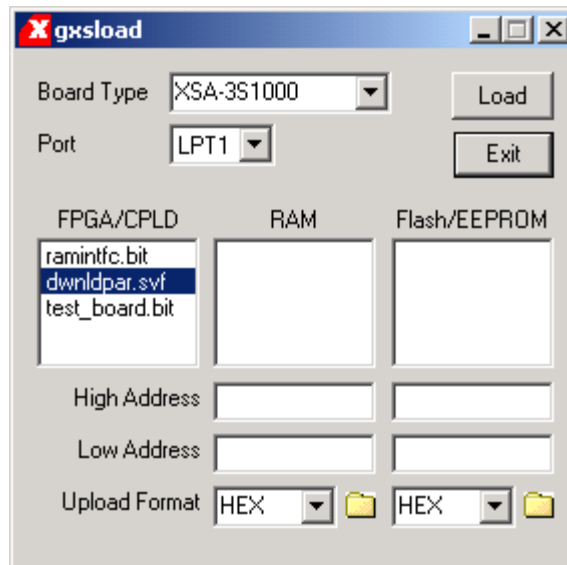


Once you drop the file, the highlighted file name appears in the FPGA/CPLD pane and the Load button in the GXSLoad window is enabled. Clicking on the Load button will begin sending the bitstream in the file to the XSA-3S1000 Board through the parallel port connection. .BIT files contain configuration bitstreams that are loaded into the FPGA while .SVF files will go to the CPLD. GXSLoad will reject any non-downloadable files (ones

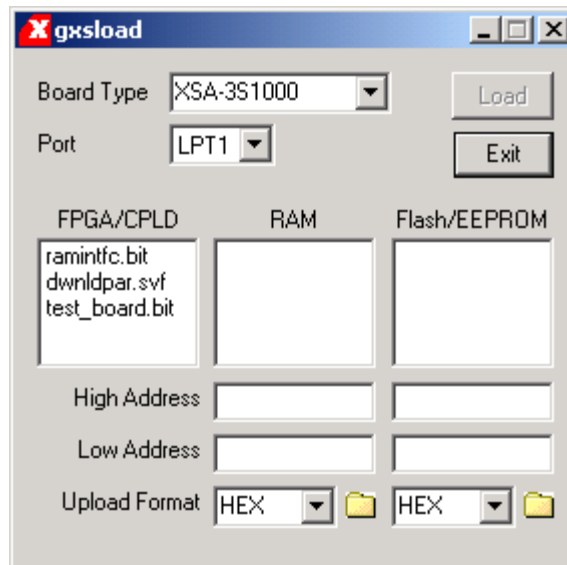
with a suffix other than .BIT or .SVF). During the downloading process, GXSLD will display the name of the file and the progress of the current download.



You can drag & drop multiple files into the FPGA/CPLD area. Clicking your mouse on a filename will highlight the name and select it for downloading. Only one file at a time can be selected for downloading.



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



Downloading Using Xilinx iMPACT

You can use the Xilinx iMPACT software to download bitstreams to the XSA-3S1000 Board. The iMPACT programming tool downloads bitstreams through the JTAG interface of the FPGA so we need to change the parallel port interface by reprogramming the CPLD. Drag & drop the p3jtag.svf file from the XSTOOLS\XSA\3S1000 folder into the FPGA/CPLD pane of the GXSLOAD window. Then click on the Load button and the CPLD will be reprogrammed in less than a minute. Then move the shunt on jumper J9 from the XS to the XI position. At this point you can start iMPACT and it will believe it is connected to the XSA-3S1000 Board through a Xilinx Parallel Cable III in boundary-scan mode. Follow the instructions for iMPACT to download bitstreams to the FPGA.

You can also use a Xilinx Parallel Cable IV with the XSA-3S1000 Board by downloading the XSTOOLS\XSA\3S1000\p4jtag.svf file into the CPLD. Then disconnect the XESS downloading cable and connect the Xilinx Parallel Cable IV to the X4 connector. Now start iMPACT to download bitstreams to the FPGA in boundary-scan mode.

Note that the CPLD only needs to be reprogrammed once to support iMPACT because it retains its configuration even when power is removed from the board. (If you want to go back to using the GXSLOAD programming utility, just must move the shunt on J9 back to the XS position and download the XSTOOLS\XSA\dwnldpar.svf file into the CPLD.)

Storing Non-Volatile Bitstreams in the Flash

The FPGA on the XSA-3S1000 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 16 Mbit Flash device on the XSA-3S1000 Board from which the FPGA will be configured each time power is applied.

The Flash is partitioned into four quadrants, each of which can hold a bitstream for the FPGA. Before a bitstream can be downloaded into a quadrant of the Flash, the .BIT file must be converted into an .EXO or .MCS format using one of the following commands:

Quadrant	Address Range	Conversion Command	DIP Switch Setting	
			SW1-1	SW1-2
0	0x000000 – 0x07FFFF	promgen -u 0 file.bit -p exo -w promgen -u 0 file.bit -p mcs -w	ON	ON
1	0x080000 – 0x0FFFFFFF	promgen -u 80000 file.bit -p exo -w promgen -u 80000 file.bit -p mcs -w	ON	OFF
2	0x100000 – 0x17FFFF	promgen -u 100000 file.bit -p exo -w promgen -u 100000 file.bit -p mcs -w	OFF	ON
3	0x180000 – 0x1FFFFFFF	promgen -u 180000 file.bit -p exo -w promgen -u 180000 file.bit -p mcs -w	OFF	OFF

In the commands shown above, the bitstream in file.bit is transformed into an .EXO or .MCS formatted file starting at the first address in each quadrant and proceeding upward.

The .EXO or .MCS file is downloaded into the Flash device by dragging it into the Flash/EEPROM pane and clicking on the Load button. This activates the following sequence of steps:

1. The FPGA and CPLD on the XSA-3S1000 Board are reprogrammed to create an interface between the Flash device and the PC parallel port.
2. The entire Flash device is erased.
3. The contents of the .EXO or .MCS file are downloaded into the Flash through the parallel port.
4. The CPLD is reprogrammed with a circuit that configures the FPGA with the contents of the Flash whenever power is applied to the XSA-3S1000 Board. (This configuration loader is stored in the XSTOOLS\XSA\3S1000\fcnfsg.svf file.)

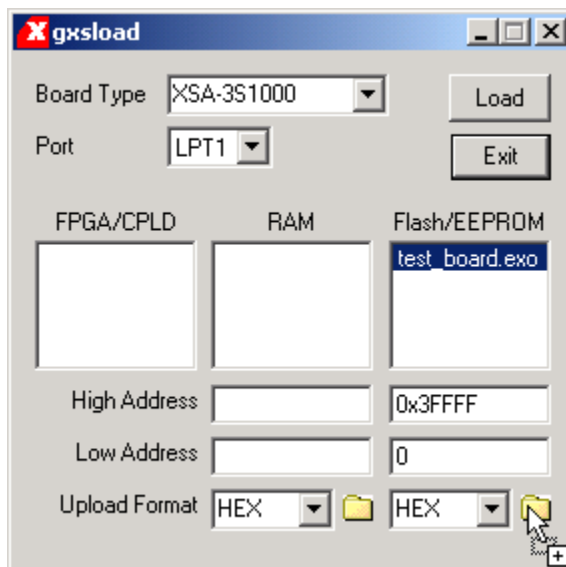
Once the Flash download is complete, you must set the DIP switches to select the Flash quadrant containing the FPGA bitstream (see the switch settings in the table above). The FPGA will be configured with the bitstream in that quadrant whenever power is applied to the board. You can download multiple bitstreams to the Flash and use the switches to select the one to be loaded into the FPGA on power-up.

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 located below the Flash/EEPROM pane, and select the format for the uploaded data from 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 and FPGA on the XSA-3S1000 Board are 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 default parallel port interface remains in the CPLD. You will need to reprogram the CPLD with the configuration loader bitstream in XSTOOLS\XSA\3S1000\fcnfg.svf if you want the FPGA to be configured from Flash whenever power is applied.

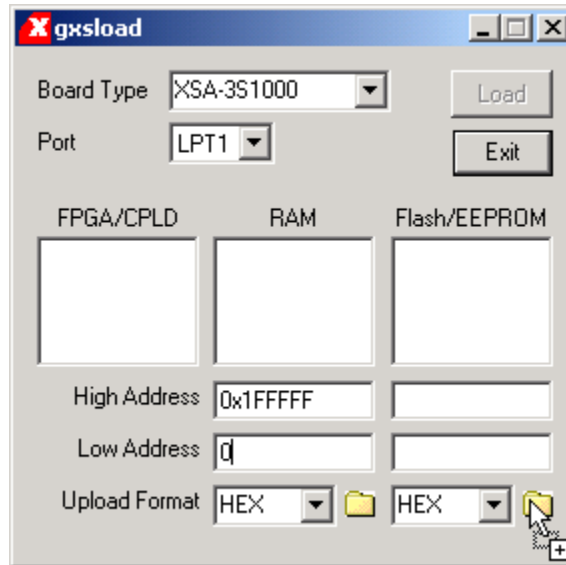
Downloading and Uploading Data to the SDRAM

The XSA-3S1000 Board contains a 256 Mbit, synchronous DRAM (16M x 16 SDRAM) whose contents can be downloaded and uploaded by GXSLOAD. This is useful for initializing the SDRAM with data for use by the FPGA and then reading the SDRAM contents after the FPGA has operated upon it. The SDRAM is loaded with data by dragging & dropping one or more .EXO, .MCS, .HEX, and/or .XES files into the RAM pane of the GXSLOAD window and then clicking on the Load button. This activates the following sequence of steps:

1. The FPGA is reprogrammed to create an interface between the SDRAM and the PC parallel port. (This interface is stored in the XSTOOLS\XSA\3S1000\ramintfc.bit bitstream file. **The CPLD must have previously been loaded with the dwnldpar.svf file found in the same folder.**)
2. The contents of the .EXO, .MCS, .HEX or .XES files are downloaded into the SDRAM 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 pane, then this bitstream is loaded into the FPGA or CPLD on the XSA-3S1000 Board. Otherwise the FPGA remains configured as an interface between the PC and the SDRAM.

You can also examine the contents of the SDRAM device by uploading it to the PC. To upload data from an address range in the SDRAM, type the upper and lower bounds of the range into the High Address and Low Address fields below the RAM pane, and select the format for the uploaded data from 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 FPGA is reprogrammed to create an interface between the SDRAM device and the PC parallel port.
2. The SDRAM 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 location N in the SDRAM is stored in the eight-bit file with the upper eight bits at address $2N$ and the lower eight bits at address $2N+1$. This byte-ordering applies for both RAM uploads and downloads.

4

Programmer's Models

This section describes the various sections of the XSA-3S1000 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. For more information, you can find a table of FPGA and CPLD pin connections and detailed schematics at the end of this manual.

XSA-3S1000 Board Organization

The XSA-3S1000 Board contains the following components:

FPGA: This is the main repository of programmable logic on the XSA-3S1000 Board.

CPLD: This manages the interface between the PC parallel port and the rest of the XSA-3S1000 Board. It can also configure the FPGA with a bitstream from Flash.

Oscillator: A fixed-frequency oscillator generates the master clock for the XSA-3S1000 Board.

SDRAM: A 256 Mbit SDRAM provides volatile data storage accessible by the FPGA.

Flash: A 16 Mbit Flash device provides non-volatile storage for data and FPGA configuration bitstreams.

LED: A seven-segment LED allows visible feedback as the XSA-3S1000 Board operates.

DIP switch: A four-position DIP switch passes settings to the XSA-3S1000 Board and controls the upper address bits of the Flash device.

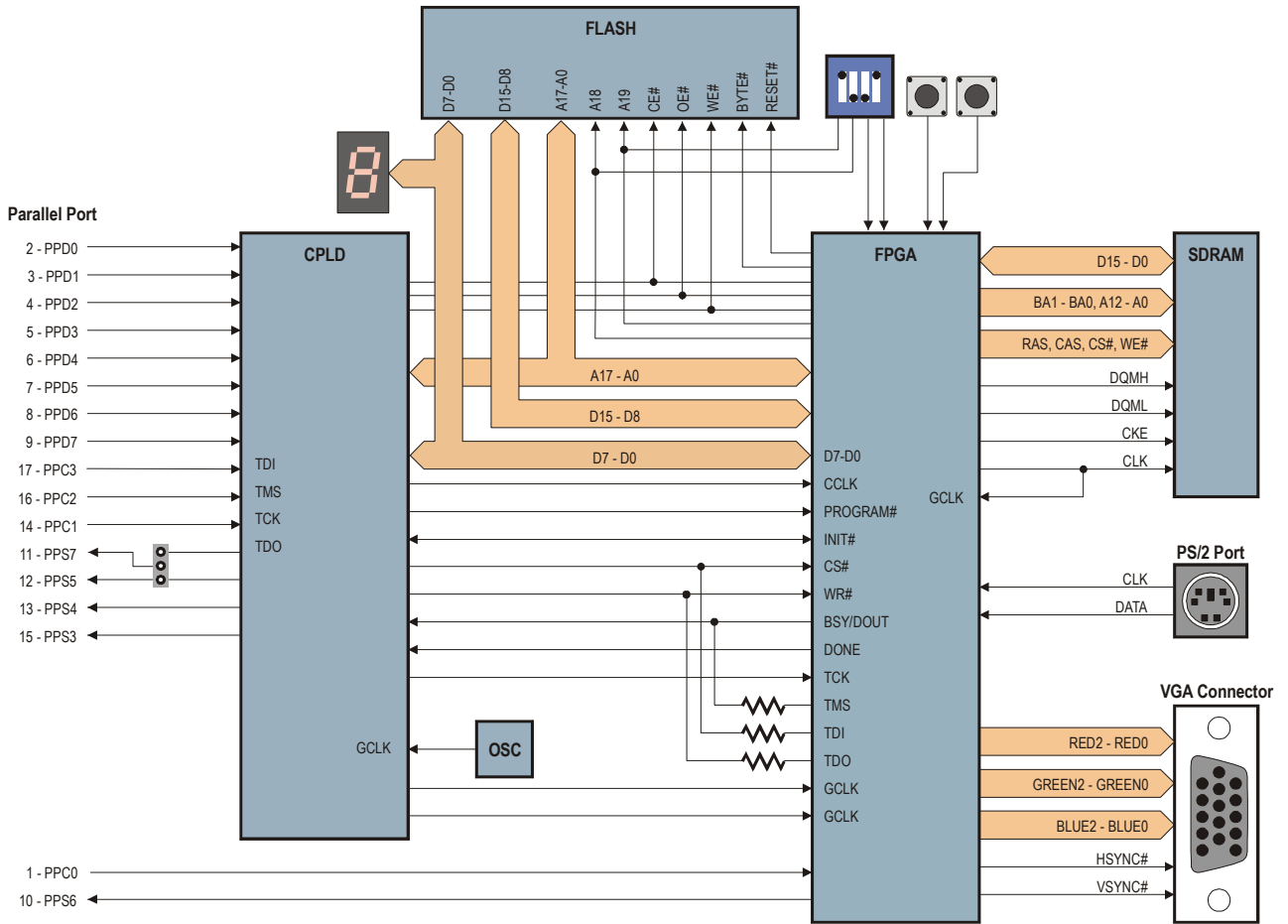
Pushbuttons: Two pushbuttons send momentary contact information to the FPGA.

PS/2 Port: A keyboard or mouse can interface to the XSA-3S1000 Board through this port.

VGA Port: The XSA-3S1000 Board can send signals to display 512-color graphics on a VGA monitor through this port.

Parallel Port: This is the main interface for passing configuration bitstreams and data to and from the XSA-3S1000 Board.

Prototyping Header: Many of the FPGA I/O pins are connected to the 84 pins on the bottom of the XSA-3S1000 Board that are meant to mate with solderless breadboards or an XST-3 Board.



• Figure 3: XSA-3S1000 Board programmer's model.

Programmable logic: FPGA and CPLD

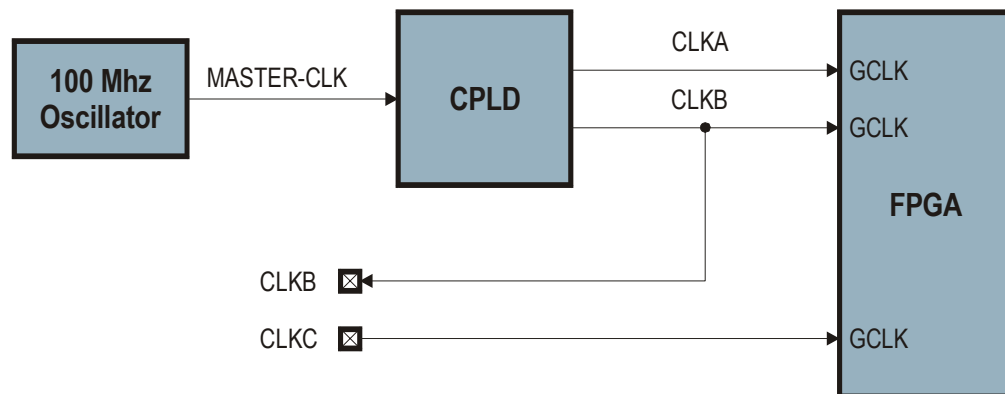
The XSA-3S1000 Board contains two programmable logic chips:

- A 1,000,000-gate XILINX Spartan3 FPGA in a 256-pin BGA package ([XC3S1000-4FT256](#)) is the main repository of programmable logic on the XSA-3S1000 Board.
- A XILINX XC9500XL CPLD ([XC9572XL-10VQ64](#)) is used to manage the configuration of the FPGA via the parallel port. In stand-alone mode, the CPLD also configures the FPGA with a bitstream from the Flash RAM.

100 MHz Fixed-Frequency Oscillator

An oscillator provides a fixed, 100 MHz clock signal to a dedicated clock input of the CPLD. From this clock, the CPLD generates two clock signals, CLKA and CLKB, that go to dedicated clock inputs of the FPGA. This allows the CPLD to control the FPGA clocks. By default, the CPLD outputs 100 MHz and 50 MHz clocks on CLKA and CLKB, respectively. The clock-divider circuit in the CPLD can be reprogrammed to send lower-frequency clocks to the FPGA if desired.

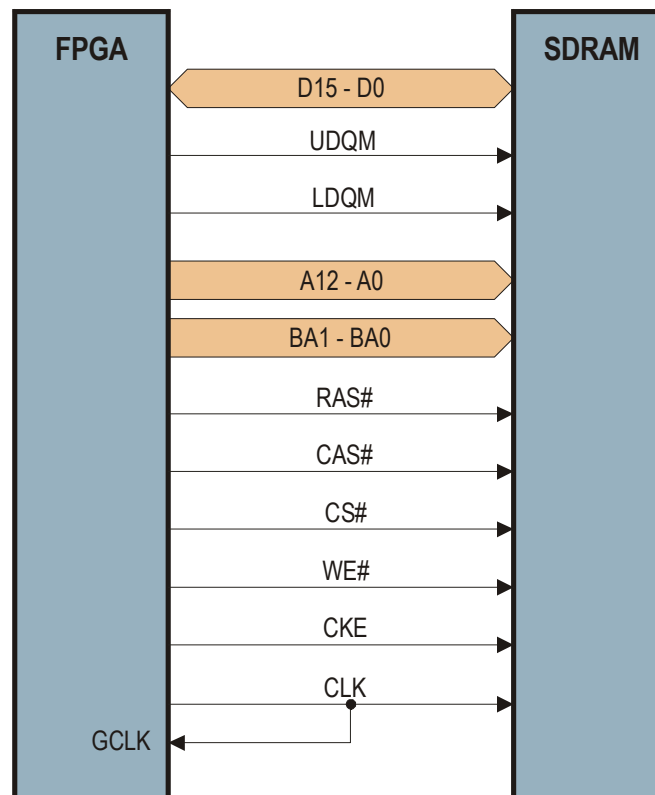
The CLKB signal also exits through a pin on the prototyping header, so it can be used as a clock for an external system connected to the XSA-3S1000 Board. Or the external system can send a clock directly to the FPGA through the dedicated CLKC pin of the prototyping header.



Synchronous DRAM

The XSA-3S1000 Board incorporates a 16M x 16 SDRAM ([K4S561632ETC75](#)) that connects solely to the FPGA as shown below. Note that the clock signal is re-routed back to a dedicated clock input of the FPGA to compensate for clock delays to the SDRAM, thus allowing synchronization of the FPGA's internal operations with the SDRAM operations.

This [application note](#) describes an SDRAM controller that makes the SDRAM appear like a simple static RAM to the rest of the circuitry in the FPGA.



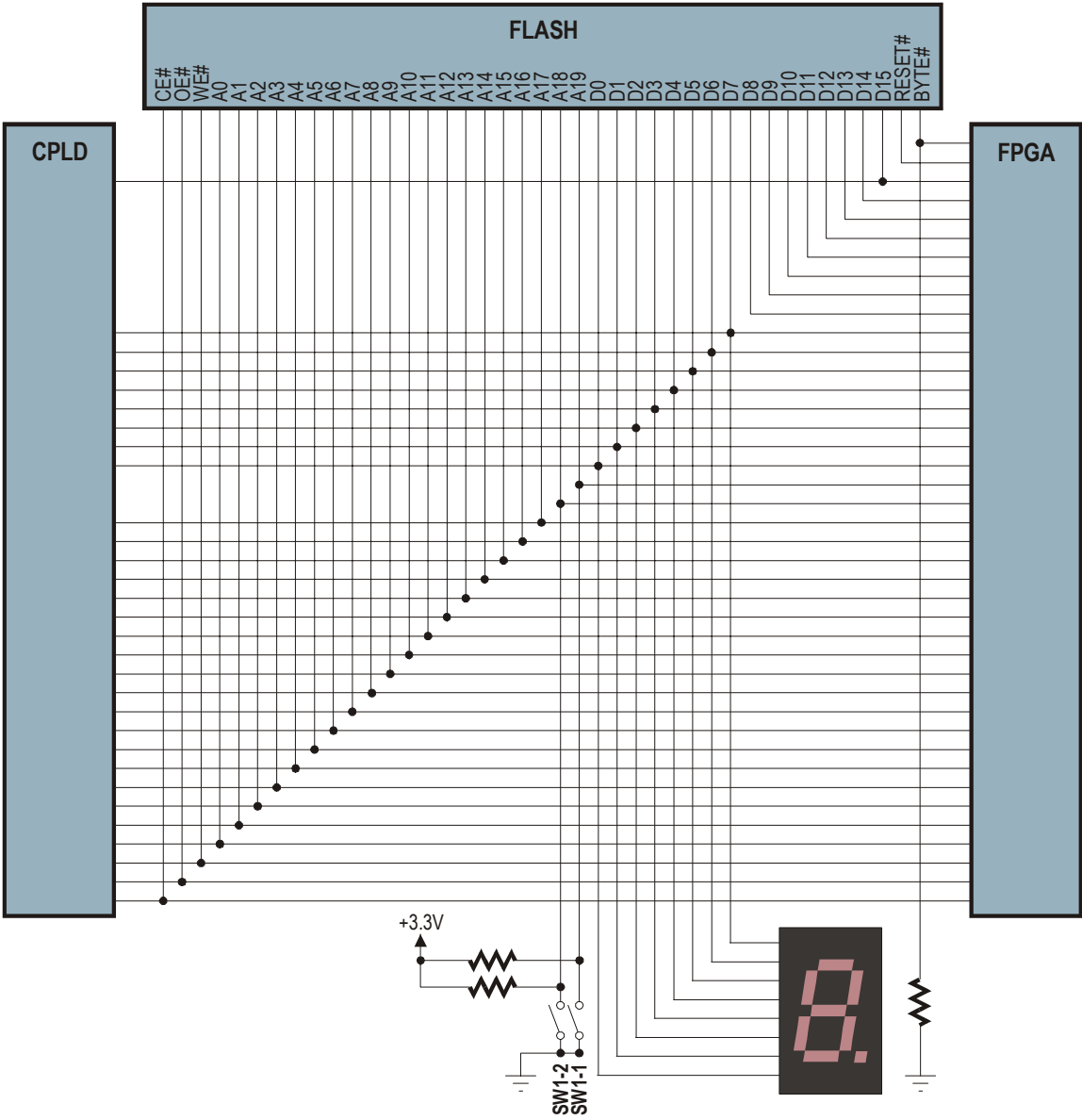
Flash RAM

The CPLD and FPGA connect to a 16 Mbit Flash RAM ([S29AL016M10TAI020](#)) that operates in either byte mode (2M x 8) or word mode (1M x 16). The CPLD uses the byte mode as it only has access to the lower eight bits of the Flash data bus, while the FPGA connects to the entire 16-bit data bus and can select either mode using the BYTE# control line.

The FPGA has access to the entire Flash address bus so it can read or write any location. For this reason, the FPGA is used to pass data between the parallel port and the Flash when GXLOAD downloads/uploads files to/from the Flash. The CPLD, however, is not connected to the upper two address lines so it can only access a quadrant of the Flash. The quadrant is selected by two DIP switches connected to the upper address lines. On power-up in stand-alone mode, the CPLD configures the FPGA with a bitstream retrieved

from the selected quadrant, so the DIP switches can be used to select between four separate bitstreams stored in the Flash. (See the [application note](#) on the XSA Board Flash configuration circuit for more details on this.)

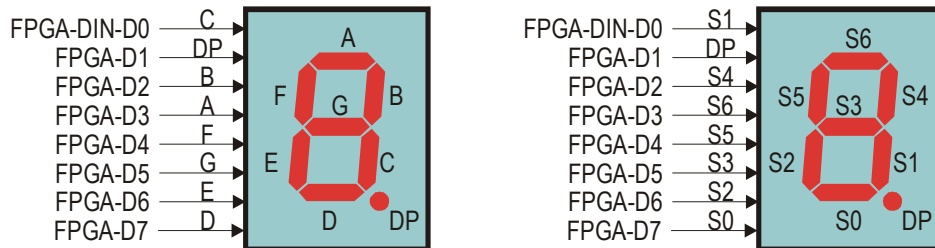
After power-up, any application circuit loaded into the FPGA can read and/or write the Flash. To avoid contention, the CPLD is programmed to release control of all Flash address/data/control lines whenever the FPGA lowers the Flash CE# line. When the Flash is disabled by raising CE#, the I/O lines connected to the Flash are available for general-purpose communication between the FPGA and the CPLD.



Seven-Segment LED

The XSA-3S1000 Board has a 7-segment LED digit for use by the FPGA or the CPLD. Segments of the LED glow when a logic-high level is applied to them.

The LED shares the same eight-bit data bus that interconnects the CPLD, the FPGA configuration port and the lower-byte of the Flash RAM data bus. The connections between the LED segments and the data bus are shown below. (We use two distinct labelings of the LED segments in our documentation and design examples, so we show the connections for both.)



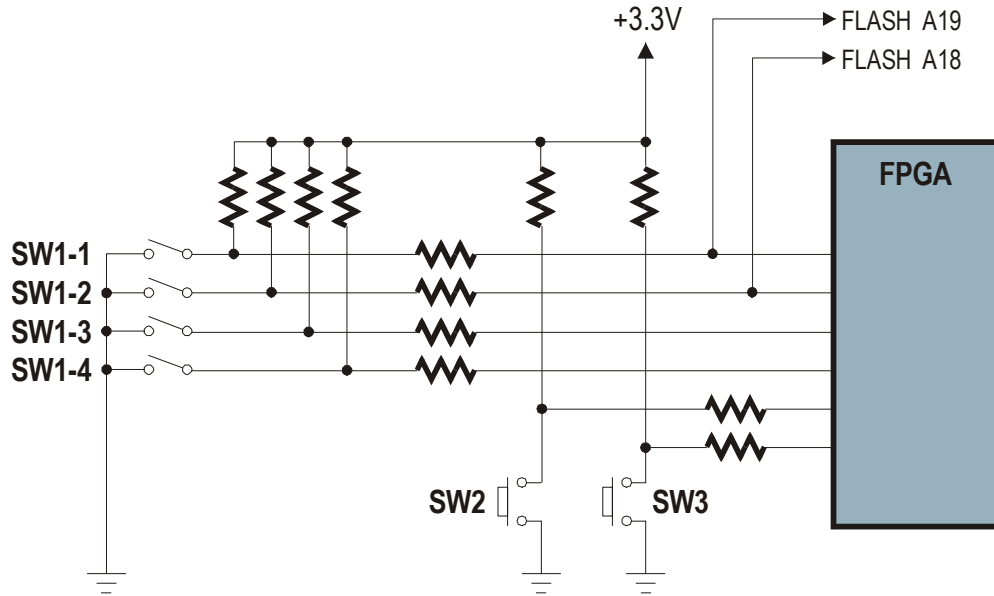
DIP Switches and Pushbuttons

Four DIP switches are attached to the FPGA. When closed (ON), each switch pulls the connected pin of the FPGA to ground. The pin is pulled high through a resistor when the switch is open (OFF).

Two of the DIP switches also connect to the upper two bits of the Flash address bus. These DIP switches are used to select the Flash quadrant holding a bitstream that will be loaded into the FPGA by the CPLD on power-up.

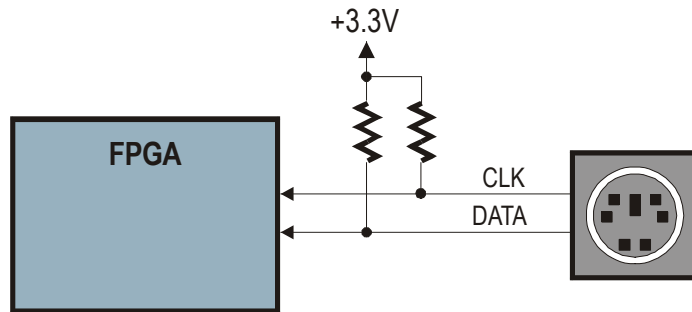
The FPGA also connects to two pushbuttons. Each pushbutton applies a low level to its FPGA pin when pressed and a resistor pulls the pin to a high level when the pushbutton is released.

Small resistors are placed in series between the FPGA and the switches and pushbuttons to prevent damage if the FPGA tries to drive a pin that is being pulled low.



PS/2 Port

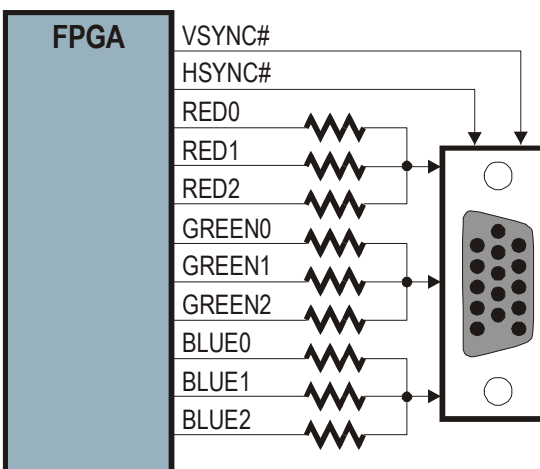
A PS/2 port provides the FPGA with an interface to either a keyboard or a mouse. The FPGA receives two signals from the PS/2 interface: a clock signal and a serial data stream that is synchronized with the falling edge of the clock. (For more details on using the PS/2 port and a simple circuit for receiving keystroke information from a keyboard, see this [application note](#).)



VGA Port

The FPGA can generate a video signal for display on a VGA monitor. The FPGA outputs three bits each of red, green, and blue color information to a simple resistor-ladder DAC. This provides a palette of $2^3 \times 2^3 \times 2^3 = 512$ colors. The outputs of the DAC are sent to the

RGB inputs of a VGA monitor. The FPGA also generates the horizontal and vertical sync pulses (HSYNC#, VSYNC#). (See this [application note](#) for more details on a simple circuit for generating VGA signals that displays an image stored in SDRAM.)



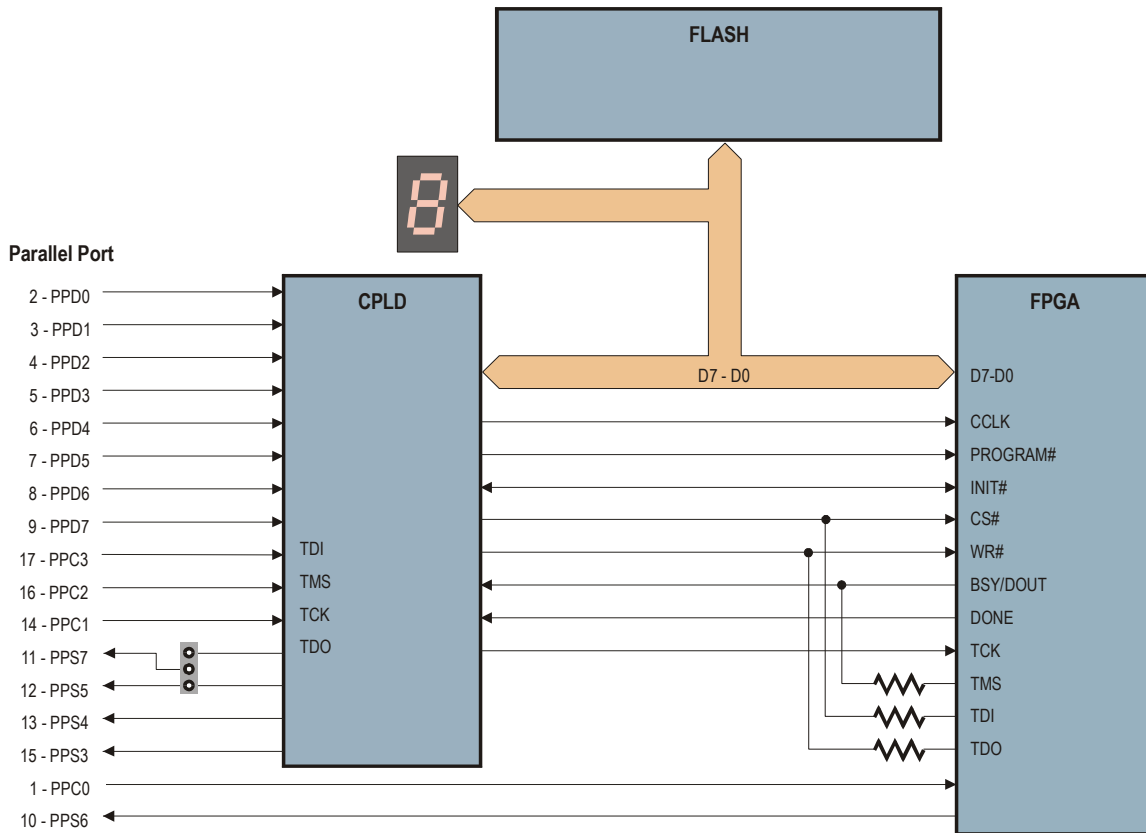
Parallel Port

The parallel port is the main interface for communicating between the XSA-3S1000 Board and a PC. Control line C0 and status line S6 connect directly to the FPGA and can be used for bidirectional communication between the FPGA and PC. The CPLD handles the fifteen remaining active lines of the parallel port as follows.

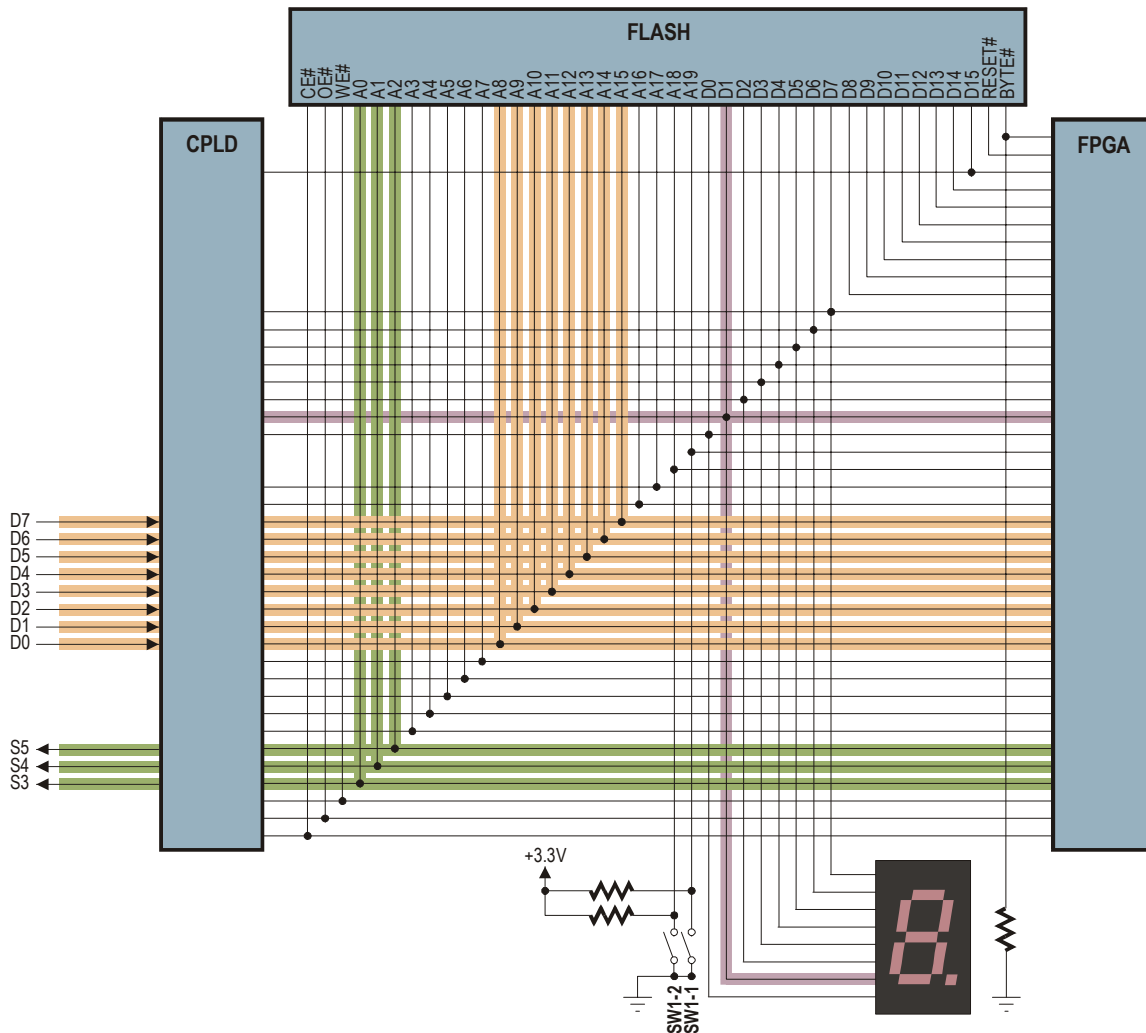
Three of the parallel port control lines, C1–C3, connect to the JTAG pins through which the CPLD is programmed. The C1 control line clocks configuration data presented on the C3 line into the CPLD while the C2 signal steers the actions of the CPLD programming state machine. Meanwhile, information from the CPLD returns to the PC through status line S7.

The eight data lines, D0–D7, and the remaining three status lines, S3–S5, connect to general-purpose pins of the CPLD. The CPLD can be programmed to act as an interface between the FPGA and the parallel port. The CPLD connects to the FPGA configuration pins so it can pass bitstreams from the parallel port to the FPGA. The actual configuration data is presented to the FPGA on the same 8-bit bus that also connects to the Flash and seven-segment LED. The CPLD also drives the configuration pins (CCLK, PROGRAM#, CS#, and WR#) that sequence the loading of a bitstream into the FPGA. The CPLD can monitor the status of the bitstream download through the INIT#, DONE, and BSY/DOUT pins.

The CPLD also has access to the FPGA's JTAG pins: TCK, TMS, TDI, and TDO. The TMS, TDI, and TDO pins share the connections with the BSY/DOUT, CS#, and WR# pins. The CPLD can be programmed with an interface that allows configuration of the FPGA through the JTAG pins using the XILINX iMPACT software (see this [application note](#) for more details). Jumper J9 allows the connection of status pin S7 to the general-purpose CPLD pin that also drives status pin S5. This is required by the iMPACT software so it can check for the presence of the downloading cable.



After the FPGA is configured with a bitstream and the DONE pin goes high, the CPLD switches into a mode that connects the parallel port data and status pins to the FPGA. This lets the PC pass data to the FPGA over the parallel port data lines while receiving data from the FPGA over the status lines. The active connections between the FPGA, CPLD and the parallel port after configuration are shown below.



The FPGA sends data to the PC by driving logic levels onto the A0, A1 and A2 Flash address lines which pass through the CPLD and onto the parallel port status lines S3, S4 and S5, respectively. Conversely, the PC sends data to the FPGA on parallel port data lines D7–D0 and the data passes through the CPLD and ends up on the A15–A8 Flash address lines, respectively. The FPGA should never drive A15–A8 unless it is accessing the Flash, otherwise the CPLD and/or FPGA could be damaged. The CPLD can sense when the FPGA lowers the Flash CE# and it will release the Flash address lines so the FPGA can drive them without contention.

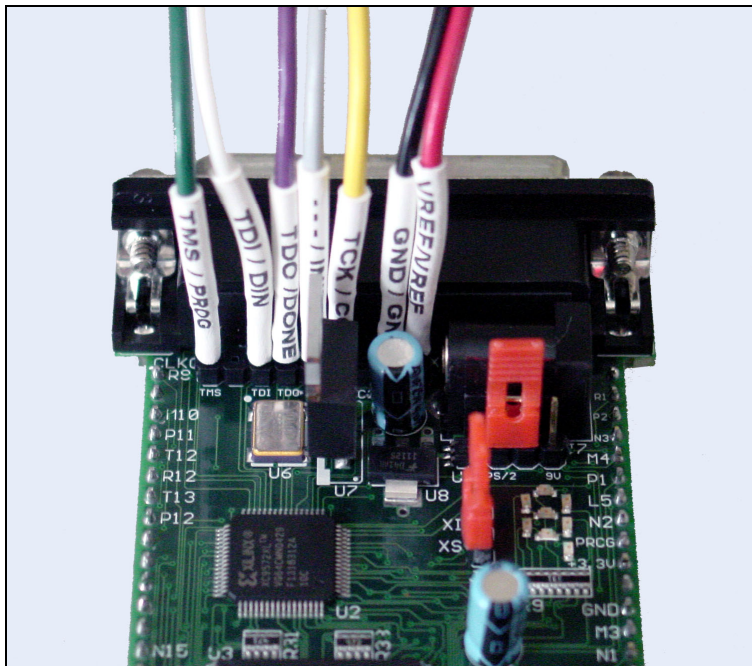
The CPLD also drives the decimal-point of the LED display (connected to Flash data line D1) to give a visual indication when the FPGA is configured with a valid bitstream. Unless it is accessing the Flash RAM, the FPGA should never drive Flash data line D1 to a low logic level or it may damage itself or the CPLD. But when the FPGA lowers the Flash CE#, the CPLD will stop driving the LED decimal-point to allow the FPGA access to data line D1 of the Flash.

For more details on how the CPLD manages the interface between the parallel port and the FPGA both before and after device configuration, see [this application note](#).

XILINX Parallel Cable IV Connector

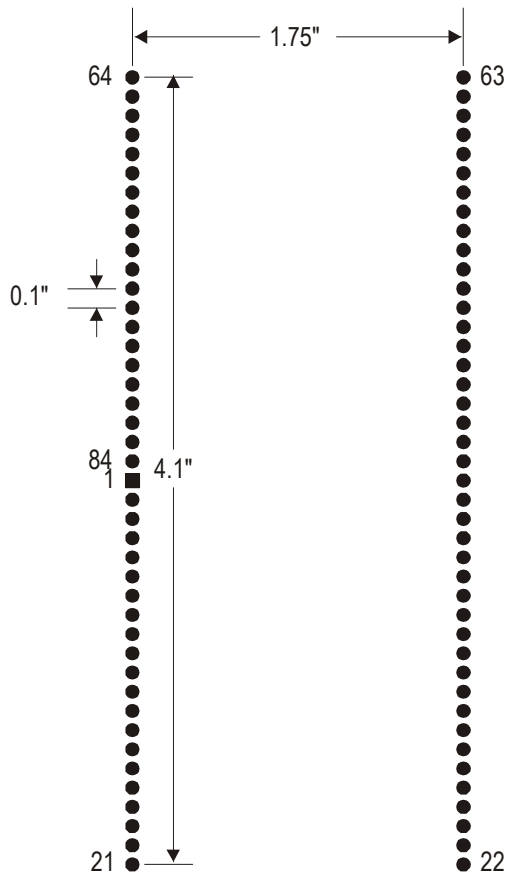
The XSA-3S1000 Board has a connector (X4) for attaching a XILINX Parallel Cable IV. The connections between the Parallel Cable IV and the XSA-3S1000 Board are shown below.

You will need to reprogram the CPLD on the XSA-3S1000 Board with the XSTOOLS\XSA\3S1000\p4jtag.svf file before you can use the Parallel Cable IV. Then disconnect the XESS downloading cable and connect the Parallel Cable IV to the X4 connector. **You should never simultaneously attach a Parallel Cable IV and the XESS parallel port downloading cable!** After connecting the Parallel Cable IV, you can use iMPACT to download bitstreams to the FPGA in boundary-scan mode.



Prototyping Header

The pins of the FPGA are accessible through the 84-pin prototyping header on the underside of the XSA-3S1000 Board. Pin 1 of the header (denoted by a square pad) is located in the middle of the left-hand edge of the board and the remaining 83 pins are arranged counter-clockwise around the periphery. The physical dimensions of the prototyping header and the pin arrangement are shown below.



A subset of the FPGA pins connects to the prototyping header. These pins are not connected to any of the other components on the XSA-3S1000 Board so they are completely free to use for I/O operations with external systems without any restrictions.

The number of the FPGA pin connected to a given header pin is printed next to the header pin on the board. This makes it easier to find a given FPGA pin when you want to connect it to an external system.

A

XSA-3S1000 Pin Connections

The following tables list the pin numbers of the FPGA and CPLD along with the pin names of the other chips that they connect to on the XSA-3S1000 Board.

Connections Between the FPGA, CPLD and other Components on the XSA-3S1000 Board

Net Name	FPGA Pin (U1)	CPLD Pin (U2)	Parallel Port	LEDs	Switch / Buttons	SDRAM (U4)	Flash (U3)	Osc (U6)	VGA	PS/2	Proto. Pin (J1)
PROTO34	G3										34
PROTO35	G2										35
PROTO36	G1										36
PROTO37	H4										37
PROTO38	H3										38
PROTO39	H1										39
PROTO40	J1										40
PROTO41	J2										41
PROTO42	J3										42
PROTO43	K1										43
PROTO44	L2										44
PROTO45	K5										45
PROTO46	L3										46
PROTO47	M1										47
PROTO48	M2										48
PROTO49	L4										49
PROTO50	N1										50
PROTO51	M3										51
PROTO53											53
PROTO56	N2										56
PROTO57	L5										57
PROTO58	P1										58
PROTO59	M4										59
PROTO60	N3										60
PROTO61	P2										61
PROTO62	R1										62
PROTO63											63
PROTO65											65
PROTO66	M10										66
PROTO67	P11										67
PROTO68	T12										68
PROTO69	R12										69
PROTO70	T13										70
PROTO71	P12										71
PROTO72											72
PROTO73											73
PROTO74											74
PROTO75											75
PROTO76											76
PROTO77	N15										77
PROTO78	M15										78
PROTO79	L15										79
PROTO80	K15										80
PROTO81	K16										81
PROTO82	J13										82
PROTO83	J14										83
PROTO84	J16										84
PS2-CLK	B16									CLK (5)	
PS2-DATA	E13									DATA (1)	
SDRAM-A0	B5					A0 (23)					
SDRAM-A1	A4					A1 (24)					
SDRAM-A2	B4					A2 (25)					
SDRAM-A3	E6					A3 (26)					

Connections Between the FPGA, CPLD and other Components on the XSA-3S1000 Board

Net Name	FPGA Pin (U1)	CPLD Pin (U2)	Parallel Port	LEDs	Switch / Buttons	SDRAM (U4)	Flash (U3)	Osc (U6)	VGA	PS/2	Proto. Pin (J1)
SDRAM-A4	E3					A4 (29)					
SDRAM-A5	C1					A5 (30)					
SDRAM-A6	E4					A6 (31)					
SDRAM-A7	D3					A7 (32)					
SDRAM-A8	C2					A8 (33)					
SDRAM-A9	A3					A9 (34)					
SDRAM-A10	B6					A10 (22)					
SDRAM-A11	C5					A11 (35)					
SDRAM-A12	C6					A12 (36)					
SDRAM-BA0	A7					BA0 (20)					
SDRAM-BA1	C7					BA1 (21)					
SDRAM-CAS#	A10					CAS# (17)					
SDRAM-CKE	D7					CKE (37)					
SDRAM-CLK	N8					CLK (38)					
SDRAM-CLK	E10										
SDRAM-CS#	B8					CS# (19)					
SDRAM-D0	C15					DQ0 (2)					
SDRAM-D1	D12					DQ1 (4)					
SDRAM-D2	A14					DQ2 (5)					
SDRAM-D3	B13					DQ3 (7)					
SDRAM-D4	D11					DQ4 (8)					
SDRAM-D5	A12					DQ5 (10)					
SDRAM-D6	C11					DQ6 (11)					
SDRAM-D7	D10					DQ7 (13)					
SDRAM-D8	B11					DQ8 (42)					
SDRAM-D9	B12					DQ9 (44)					
SDRAM-D10	C12					DQ10 (45)					
SDRAM-D11	B14					DQ11 (47)					
SDRAM-D12	D14					DQ12 (48)					
SDRAM-D13	C16					DQ13 (50)					
SDRAM-D14	F12					DQ14 (51)					
SDRAM-D15	F13					DQ15 (53)					
SDRAM-LDQM	C10					LDQM (15)					
SDRAM-RAS#	A9					RAS# (18)					
SDRAM-UDQM	D9					UDQM (39)					
SDRAM-WE#	B10					WE# (16)					
SW1-3	K2				SW1-3						
SW1-4	J4				SW1-4						
SW2	E11				SW2						
SW3	A13				SW3						
VGA-BLUE0	C9								BLU0		
VGA-BLUE1	E7								BLU1		
VGA-BLUE2	D5								BLU2		
VGA-GREEN0	A8								GRN0		
VGA-GREEN1	A5								GRN1		
VGA-GREEN2	C3								GRN2		
VGA-RED0	C8								RED0		
VGA-RED1	D6								RED1		
VGA-RED2	B1								RED2		
VGA-HSYNC#	B7								HSYNC#		
VGA-VSYNC#	D8								VSYNC#		

B

XSA-3S1000 Schematics

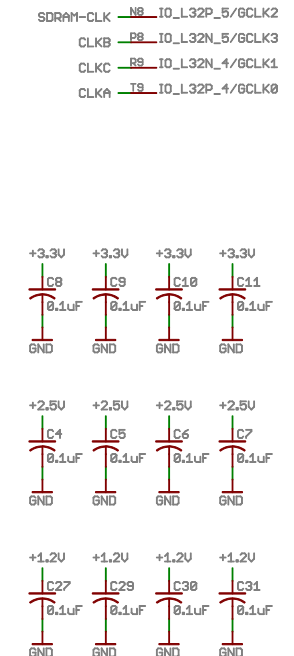
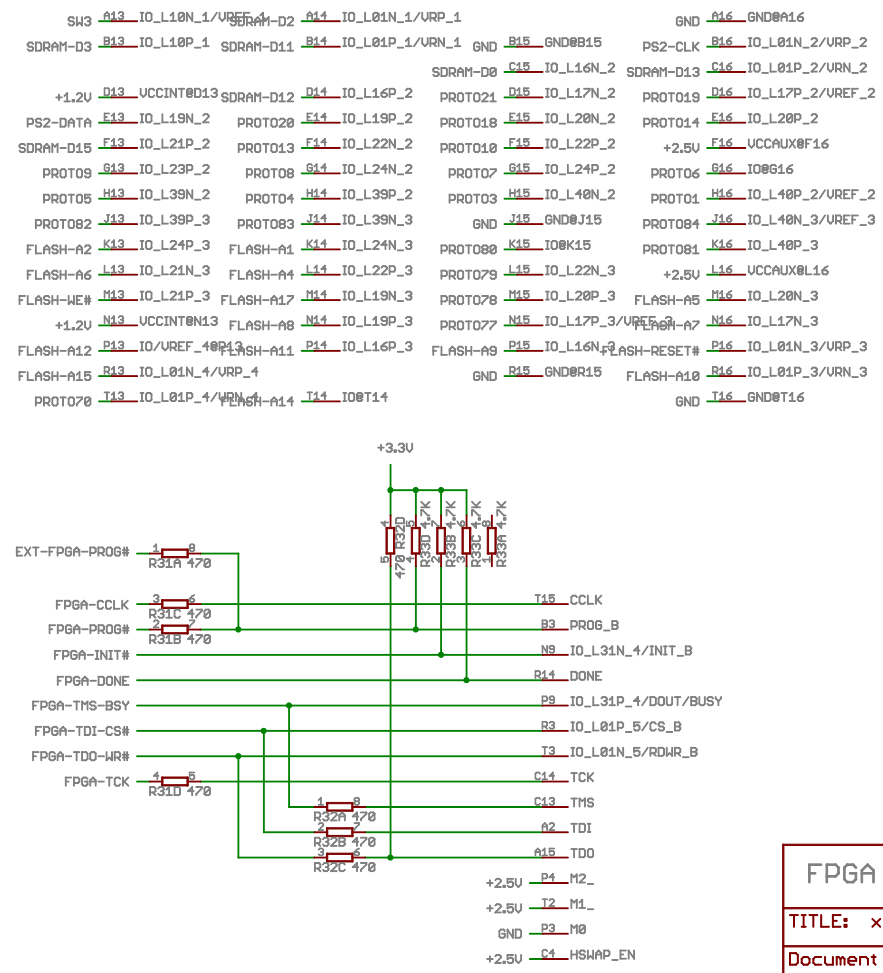
The following pages show the detailed schematics for the XSA-3S1000 Board.

GND A1 GND0A1 SDRAM-A9 A3 IO/VREF_00A3 SDRAM-A1 A4 IO_L01P_0/URN_0 UGA-GREEN1 A5 IO0A5 +2.5V A6 UCCAUX0A6
 UGA-RED2 B1 IO_L01P_7/URN_7 GND B2 GND0B2 SDRAM-A10 B4 IO_L01N_0/URP_0 SDRAM-A0 B5 IO_L25P_0 SDRAM-A10 B6 IO_L28P_0
 SDRAM-A5 C1 IO_L01N_7/URP_7 SDRAM-A8 C2 IO_L16N_7 UGA-GREEN2 C3 IO_L16P_7/UREF_7 SDRAM-A11 C5 IO_L25N_0 SDRAM-A12 C6 IO_L26N_0
 PROT025 D1 IO_L17N_7 PROT024 D2 IO_L17P_7 SDRAM-A7 D3 IO_L19P_7 +1.2V D4 UCCINT0D4 UGA-BLUE2 D5 IO/VREF_00D5 UGA-RED1 D6 IO_L27P_0
 PROT028 E1 IO_L20N_7 PROT027 E2 IO_L20P_7 SDRAM-A4 E3 IO_L19N_7/UREF_7 SDRAM-A6 E4 IO_L21P_7 +1.2V E5 UCCINT0E5 SDRAM-A3 E6 IO_L27N_0
 +2.5V F1 UCCAUX0F1 PROT032 F2 IO_L22N_7 PROT029 F3 IO_L22P_7 PROT026 F4 IO_L22N_7 PROT023 F5 IO_L23P_7 GND F6 GND0F6
 PROT036 G1 IO_L40P_7 PROT035 G2 IO0G2 PROT034 G3 IO_L24N_7 PROT033 G4 IO_L24P_7 PROT031 G5 IO_L24N_7 +3.3V G6 UCCO_70G6
 PROT039 H1 IO_L40N_7/UREF_7 GND H2 GND0H2 PROT038 H3 IO_L39N_7 PROT037 H4 IO_L39P_7 +3.3V H5 UCCO_70H5 +3.3V H6 UCCO_70H6
 PROT040 J1 IO_L40P_6/UREF_6 PROT041 J2 IO_L40N_6 PROT042 J3 IO_L39P_6 SW1-4 J4 IO_L39N_6 +3.3V J5 UCCO_60J5 +3.3V J6 UCCO_60J6
 PROT043 K1 IO0K1 SW1-3 K2 IO_L24P_6 FLASH-A18 K3 IO_L24N_6/UREF_6 FLASH-A19 K4 IO_L23P_6 PROT045 K5 IO_L23N_6 +3.3V K6 UCCO_60K6
 +2.5V L1 UCCAUX0L1 PROT044 L2 IO_L22P_6 PROT046 L3 IO_L22N_6 PROT049 L4 IO_L21P_6 PROT057 L5 IO_L21N_6 GND L6 GND0L6
 PROT047 M1 IO_L20P_6 PROT048 M2 IO_L20N_6 PROT051 M3 IO_L19P_6 PROT059 M4 IO_L19N_6 +1.2V M5 UCCINT0M5 FPGA-D7 M6 IO_L28P_5/D7
 PROT050 N1 IO_L17P_6/UREF_6 PROT056 N2 IO_L17N_6 PROT060 N3 IO_L16P_6 +1.2V N4 UCCINT0N4 FLASH-A0 N5 IO0N5 FPGA-D6 N6 IO_L28N_5/D6
 PROT058 P1 IO_L01P_6/URP_6 PROT061 P2 IO_L16N_6 FLASH-DE# P5 IO_L27P_5 FLASH-D11 P6 IO_L29P_5/UREF_5
 PROT062 R1 IO_L01N_6/URP_6 GND R2 GND0R2 FLASH-CE# R4 IO_L10P_5/URP_5 FLASH-D9 R5 IO_L27N_5/UREF_5 FLASH-D13 R6 IO_L29N_5
 GND T1 GND0T1 FLASH-D8 T4 IO_L10N_5/URP_5 FLASH-D10 T5 IO0T5 +2.5V T6 UCCAUX0T6

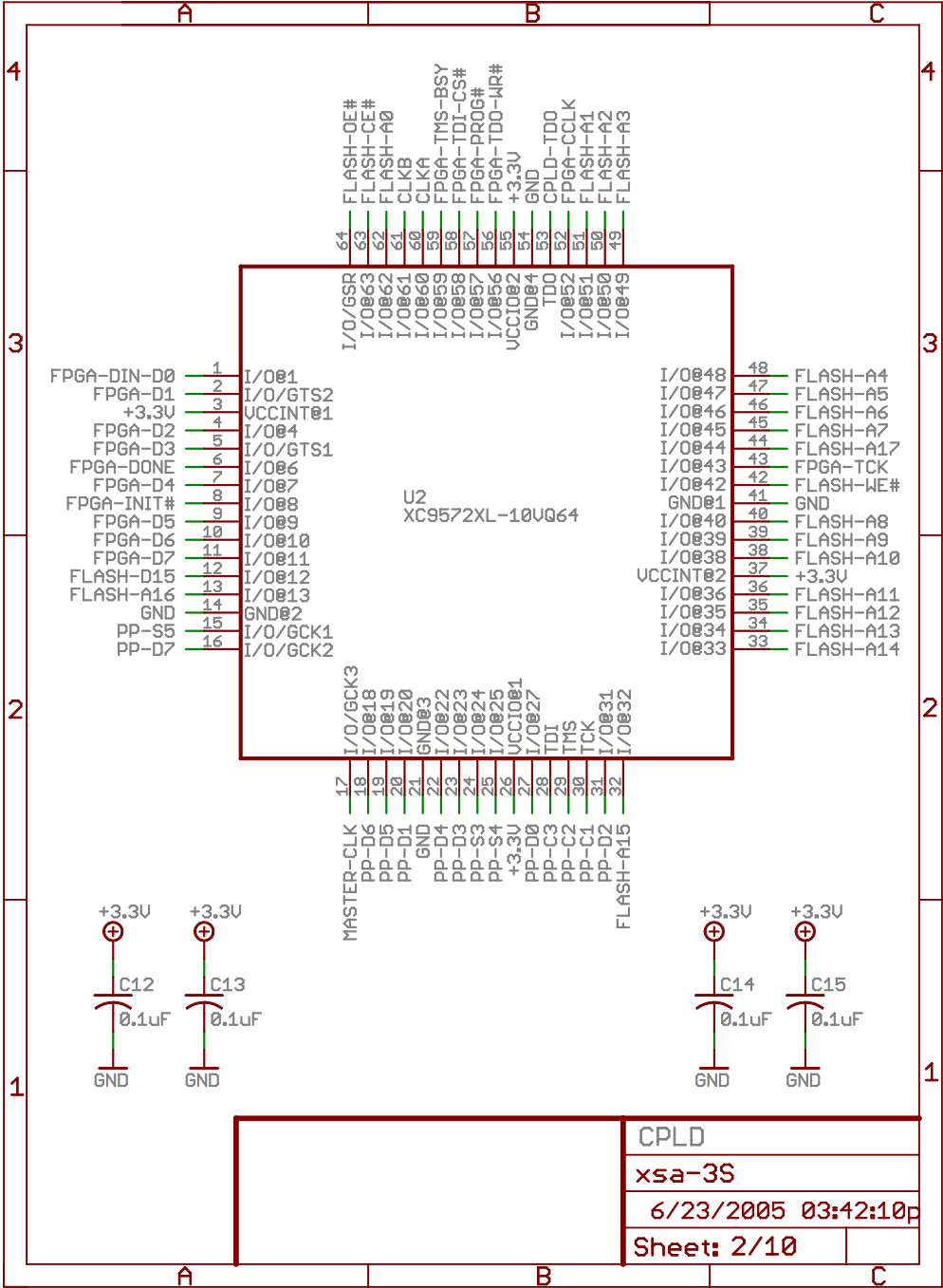
SDRAM-BA0 A7 IO0A7 UGA-GREEN0 A8 IO_L32P_0/CLK7 FLASH# A9 IO0A9 SDRAM-CAS# A10 IO_L31N_1/UREF_1 +2.5V A11 UCCAUX0A11 SDRAM-D5 A12 IO0A12
 UGA-HSYNC# B7 IO_L30P_0 SDRAM-CS# B8 IO_L32N_0/GCLK7 GND B9 GND0B9 SDRAM-WE# B10 IO_L31P_1 SDRAM-DB B11 IO_L29N_1 SDRAM-D9 B12 IO_L27N_1
 SDRAM-BA1 C7 IO_L30N_0 UGA-RED0 C8 IO_L31P_0/UREF_0 UGA-BLUE0 C9 IO_L32N_1/GCLK7 SDRAM-LDQM C10 IO0C10 SDRAM-D6 C11 IO_L29P_1 SDRAM-D10 C12 IO_L27P_1
 SDRAM-C0E D7 IO_L29P_0 UGA-USYNC# D8 IO_L31N_0 SDRAM-UQM# D9 IO_L32P_1/GCLK7 SDRAM-D7 D10 IO_L30N_1 SDRAM-D4 D11 IO_L28N_1 SDRAM-D11 D12 IO/VREF_1
 UGA-BLUE1 E7 IO_L29N_0 +3.3V E8 UCCO_00E8 +3.3V E9 UCCO_10E9 SDRAM-CLK F10 IO_L30P_1 +3.3V F10 UCCO_10F10 GND F11 GND0F11 SDRAM-D14 F12 IO_L21N_2
 +3.3V F7 UCCO_00F7 +3.3V F8 UCCO_00F8 +3.3V F9 UCCO_10F9 +3.3V F10 UCCO_10F10 GND F11 GND0F11 SDRAM-D14 F12 IO_L21N_2
 GND G7 GND0G7 GND G8 GND0G8 GND G9 GND0G9 GND G10 GND0G10 +3.3V G11 UCCO_20G11 PROT012 G12 IO_L23N_2/UREF_2
 GND H7 GND0H7 GND H8 GND0H8 GND H9 GND0H9 GND H10 GND0H10 +3.3V H11 UCCO_20H11 +3.3V H12 UCCO_20H12
 GND J7 GND0J7 GND J8 GND0J8 GND J9 GND0J9 GND J10 GND0J10 +3.3V J11 UCCO_30J11 +3.3V J12 UCCO_30J12
 GND K7 GND0K7 GND K8 GND0K8 GND K9 GND0K9 GND K10 GND0K10 +3.3V K11 UCCO_30K11 FLASH-A3 K12 IO_L23N_3
 +3.3V L7 UCCO_50L7 +3.3V L8 UCCO_50L8 +3.3V L9 UCCO_40L9 +3.3V L10 UCCO_40L10 GND L11 GND0L11 FLASH-RDY L12 IO_L23P_3/UREF_3
 FLASH-D12 M7 IO_L30P_5 +3.3V M8 UCCO_50M8 +3.3V M9 UCCO_40M9 PROT066 M10 IO_L29N_4/P0A-DIN-D0 M11 IO_L27N_4/DIN/D01.2V M12 UCCINT0M12
 FLASH-D14 N7 IO_L30N_5 FLASH-A16 N10 IO_L29P_4 FPGA-D1 N11 IO_L27P_4/FLASH-A13 N12 IO/VREF_40N12
 FLASH-D15 P7 IO0P7 FPGA-D2 P10 IO_L30N_4/D2 PROT067 P11 IO_L28N_4 PROT071 P12 IO_L25N_4
 FPGA-D5 R7 IO_L31P_5/D5 GND R8 GND0R8 FPGA-D3 R10 IO_L30P_4/D3 pp-C0 R11 IO_L28P_4 PROT069 R12 IO_L25P_4
 FPGA-D4 T7 IO_L31N_5/FLASH-BYTE# T8 IO/VREF_5 pp-S6 T10 IO/VREF_40T10 +2.5V T11 UCCAUX0T11 PROT068 T12 IO0T12

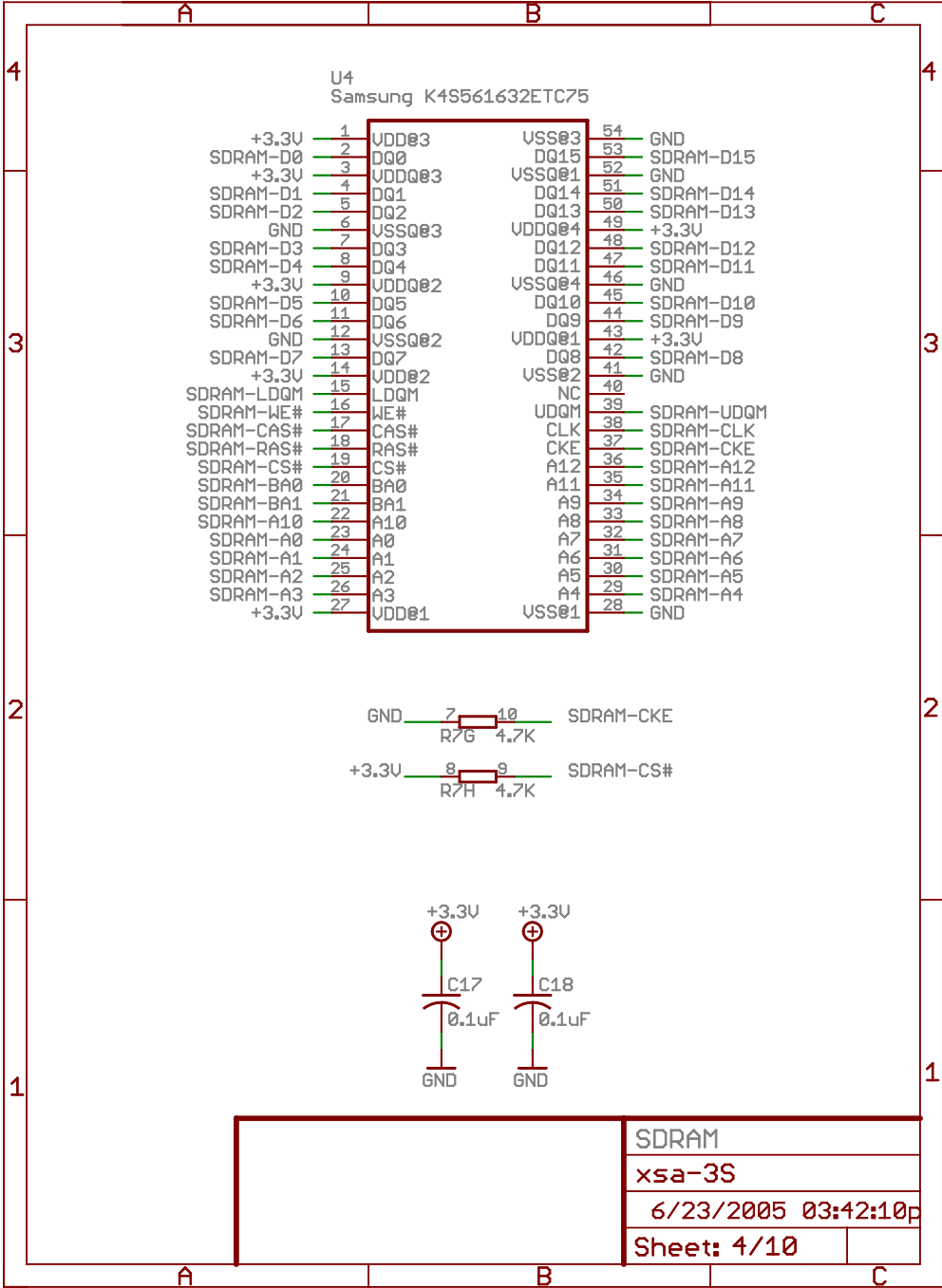
SW3 A13 IO_L10N_1/UREF_1 SDRAM-D2 A14 IO_L01N_1/URP_1 GND A16 GND0A16
 SDRAM-D3 B13 IO_L10P_1 SDRAM-D11 B14 IO_L01P_1/URN_1 GND B15 GND0B15 PS2-CLK B16 IO_L01N_2/URP_2
 +1.2V D13 UCCINT0D13 SDRAM-D12 D14 IO_L16P_2 PROT021 D15 IO_L17N_2 PROT019 D16 IO_L17P_2/UREF_2
 PS2-DATA F13 IO_L19N_2 PROT020 F14 IO_L19P_2 PROT018 F15 IO_L20N_2 PROT014 F16 IO_L20P_2
 SDRAM-D15 F13 IO_L21P_2 PROT013 F14 IO_L21N_2 PROT010 F15 IO_L22P_2 +2.5V F16 UCCAUX0F16
 PROT09 G13 IO_L23P_2 PROT08 G14 IO_L24N_2 PROT07 G15 IO_L24P_2 PROT06 G16 IO0G16
 PROT05 H13 IO_L39N_2 PROT04 H14 IO_L39P_2 PROT03 H15 IO_L40N_2 PROT01 H16 IO_L40P_2/UREF_2
 PROT082 J13 IO_L39P_3 PROT083 J14 IO_L39N_3 GND J15 GND0J15 PROT084 J16 IO_L40N_3/UREF_3
 FLASH-A2 K13 IO_L24P_3 FLASH-A1 K14 IO_L24N_3 PROT080 K15 IO0K15 PROT081 K16 IO_L40P_3
 FLASH-A6 L13 IO_L21N_3 FLASH-A4 L14 IO_L19N_3 PROT079 L15 IO_L22P_3 +2.5V L16 UCCAUX0L16
 FLASH-WE# M13 IO_L21P_3 FLASH-A17 M14 IO_L19N_3 PROT078 M15 IO_L20P_3 FLASH-A5 M16 IO_L20N_3
 +1.2V M13 UCCINT0M13 FLASH-A8 M14 IO_L19P_3 PROT077 M15 IO_L17P_3/UREF_3 FLASH-A7 M16 IO_L17N_3
 FLASH-A12 P13 IO/VREF_40P13 FLASH-A11 P14 IO_L16P_3 FLASH-A9 P15 IO_L16N_3 FLASH-RESET# P16 IO_L01N_3/URP_3
 FLASH-A15 R13 IO_L01N_4/URP_4 GND R15 GND0R15 FLASH-A10 R16 IO_L01P_3/URN_3
 PROT070 T13 IO_L01P_4/URN_4 FLASH-A14 T14 IO0T14 GND T16 GND0T16

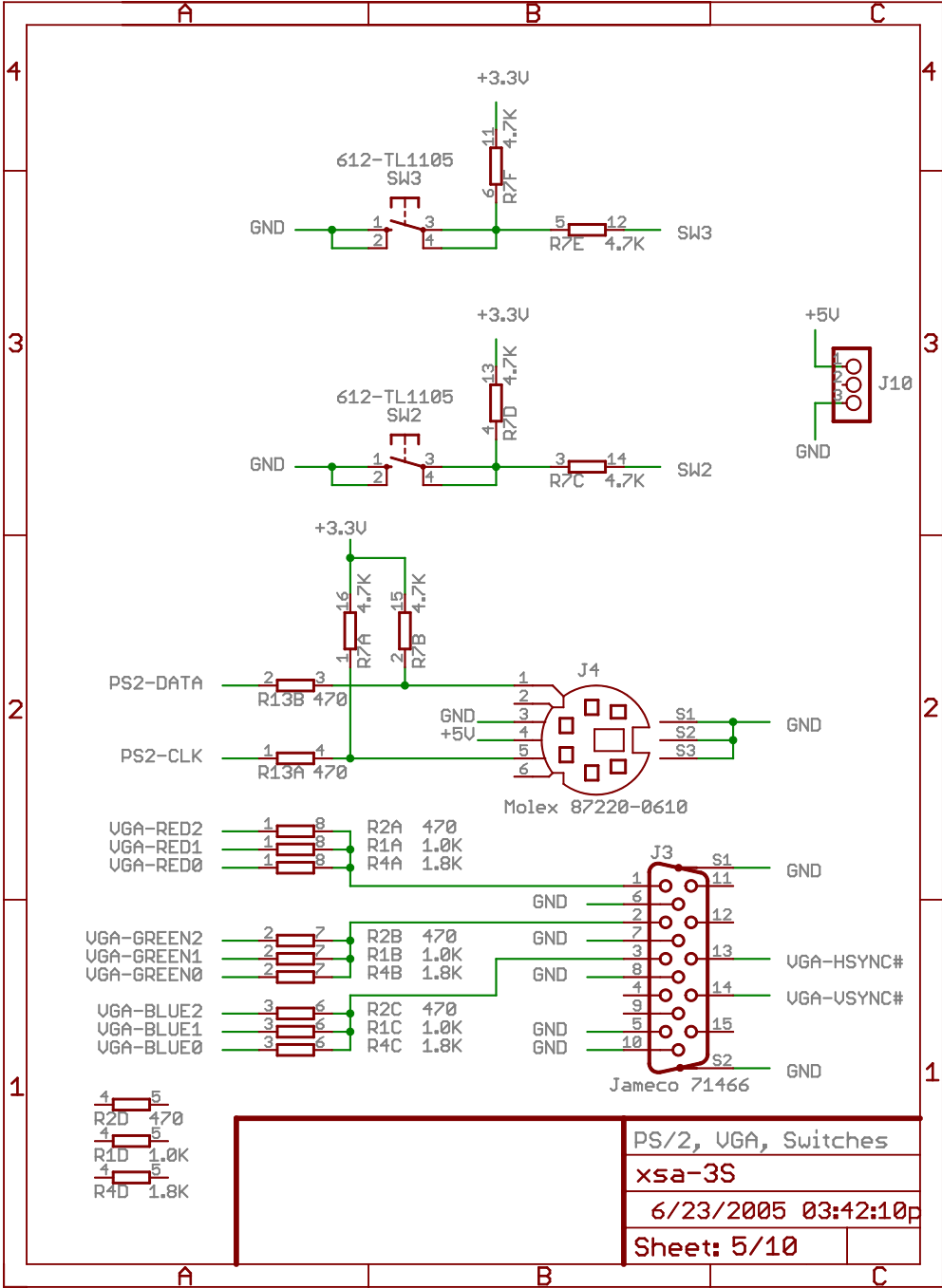
U1
XC3S1000-4FT256



FPGA	
TITLE: xsa-3S	
Document Number:	REV:
Date: 6/23/2005 03:42:10p	Sheet: 1/10

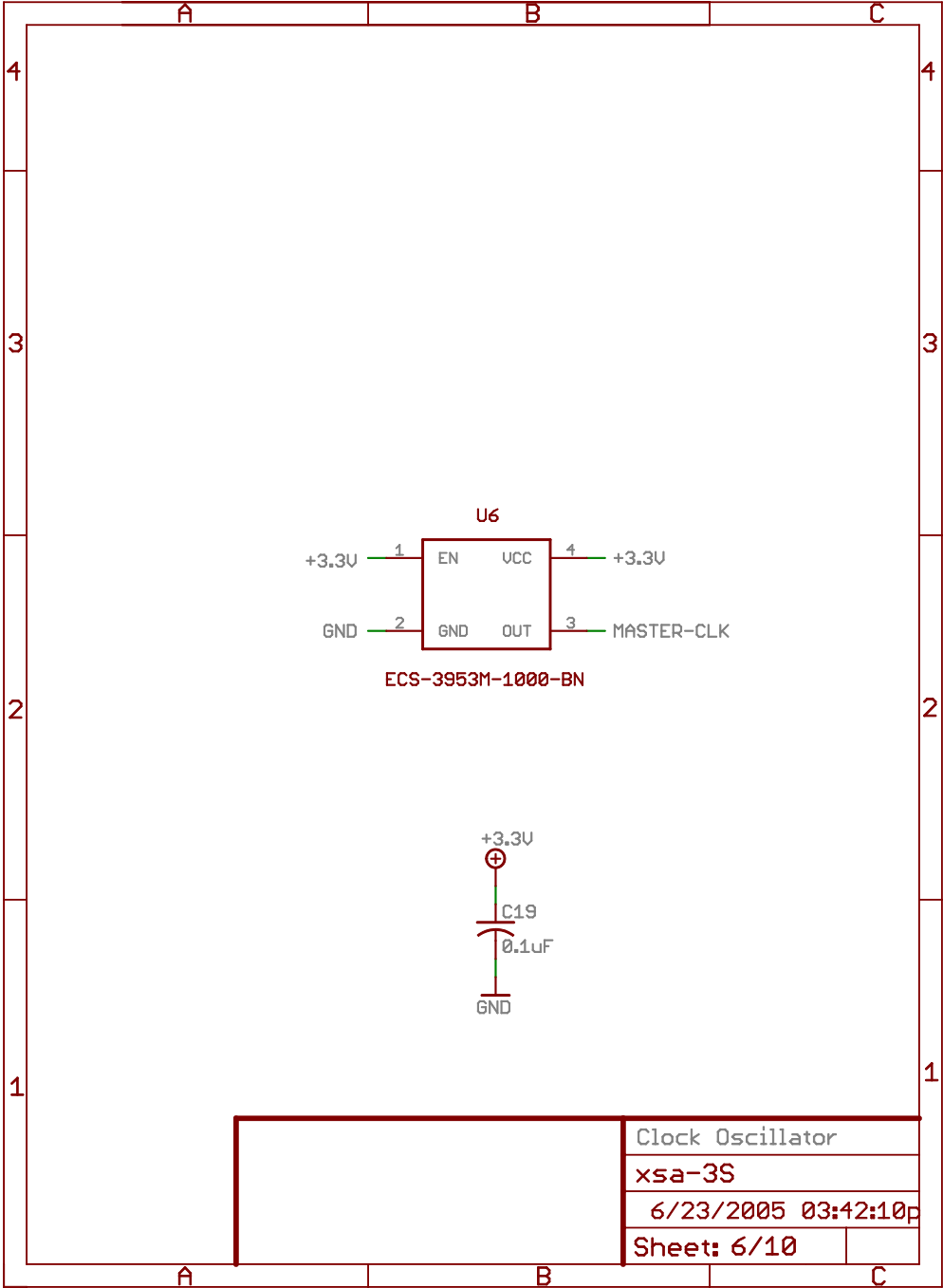


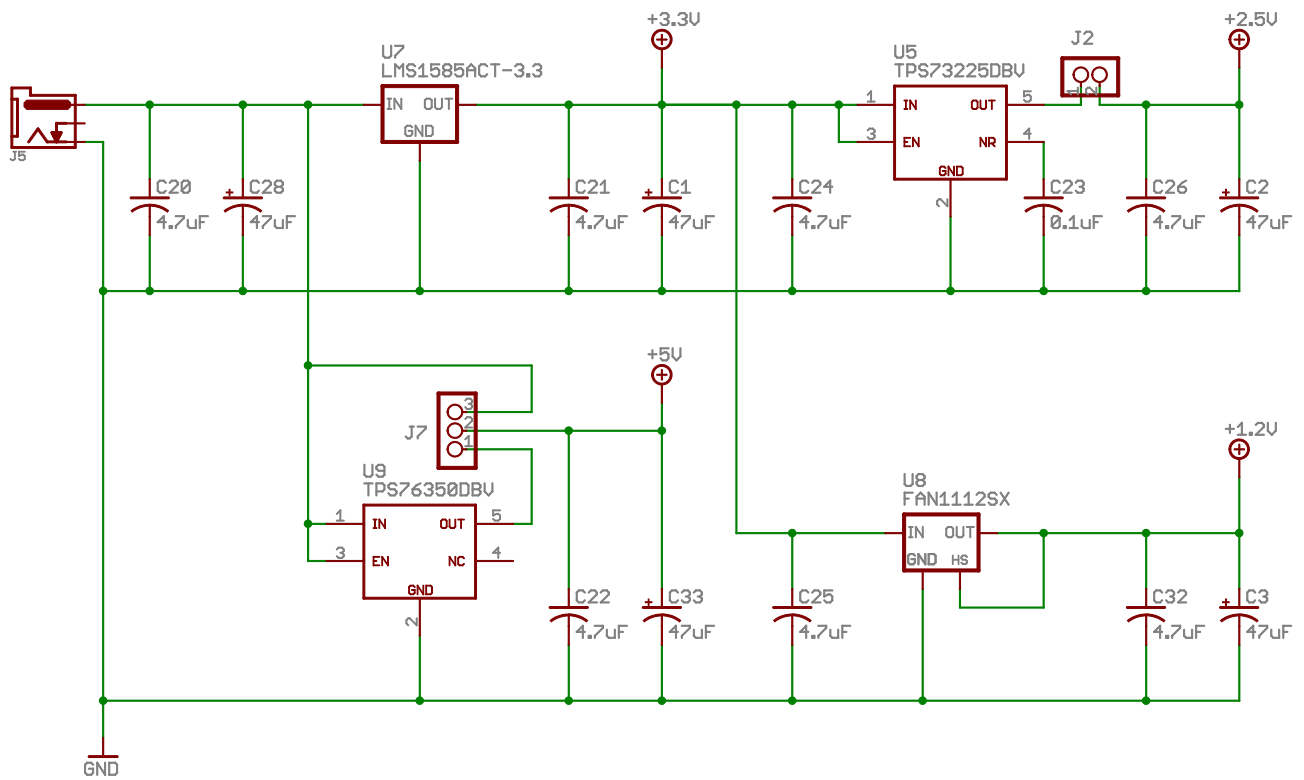




4 5
 R2D 470
 4 5
 R1D 1.0K
 4 5
 R4D 1.8K

PS/2, UGA, Switches
 xsa-3S
 6/23/2005 03:42:10p
 Sheet: 5/10





Power Supply

TITLE: xsa-3S

Document Number:

REV:

Date: 6/23/2005 03:42:10p

Sheet: 7/10

