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

XS40, XSP Board V1.3 User Manual

How to install, test, and use
your new XS40 or XSP Board

RELEASE DATE: 9/24/1999

Limited Warranty

X Engineering Software Systems Corp. (XESS) warrants that the Product, in the course of its normal use, will be free from defects in material and workmanship for a period of one (1) year and will conform to XESS's specification therefor. This limited warranty shall commence on the date appearing on your purchase receipt.

XESS shall have no liability for any Product returned if XESS determines that the asserted defect a) is not present, b) cannot reasonably be rectified because of damage occurring before XESS receives the Product, or c) is attributable to misuse, improper installation, alteration, accident or mishandling while in your possession. Subject to the limitations specified above, your sole and exclusive warranty shall be, during the period of warranty specified above and at XESS's option, the repair or replacement of the product. The foregoing warranty of XESS shall extend to repaired or replaced Products for the balance of the applicable period of the original warranty or thirty (30) days from the date of shipment of a repaired or replaced Product, whichever is longer.

THE FOREGOING LIMITED WARRANTY IS XESS'S SOLE WARRANTY AND IS APPLICABLE ONLY TO PRODUCTS SOLD AS NEW. THE REMEDIES PROVIDED HEREIN ARE IN LIEU OF a) ANY AND ALL OTHER REMEDIES AND WARRANTIES, WHETHER EXPRESSED OR IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, AND b) ANY AND ALL OBLIGATIONS AND LIABILITIES OF XESS FOR DAMAGES INCLUDING, BUT NOT LIMITED TO ACCIDENTAL, CONSEQUENTIAL, OR SPECIAL DAMAGES, OR ANY FINANCIAL LOSS, LOST PROFITS OR EXPENSES, OR LOST DATA ARISING OUT OF OR IN CONNECTION WITH THE PURCHASE, USE OR PERFORMANCE OF THE PRODUCT, EVEN IF XESS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

In the United States, some statutes do not allow exclusion or limitations of incidental or consequential damages, so the limitations above may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

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 fpga-bugs@xess.com or check our web site at http://www.xess.com. Our web site also has
  - answers to frequently-asked-questions,
  - example designs 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 Foundation software tools installed properly, send an e-mail message describing your problem to hotline@xilinx.com or check their web site at http://support.xilinx.com.

Take notice!!

- The XS40 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 9VDC power supply to your XS40 Board, please make sure the center terminal of the plug is positive and the outer sleeve is negative.
- The V1.3 version of the XS40 Board now uses a programmable oscillator with a default frequency of 50 MHz. You must reprogram the oscillator if you want to use another frequency. The procedure for doing this is described on page 8.

Packing List

Here is what you should have received in your package:

- an XS40 or XSP Board (note that your XSP Board will be labeled as an XS40 but the socket will contain a Xilinx Spartan FPGA with an "XCS" prefix);
- a 6' cable with a 25-pin male connector on each end;
- a 3.5" floppy diskette or CDROM with software utilities and documentation for using the XS40 Board.

2
Installation

Installing the XSTOOLs Utilities and Documentation

XILINX currently provides the Foundation tools for programming their FPGAs and CPLDs. Any recent version of XILINX software should generate bitstream configuration files that are compatible with your XS40 Board. Follow the directions XILINX provides for installing their software.

XESS Corp. provides the additional XSTOOLs utilities for interfacing a PC to your XS40 Board. Run the SETUP.EXE program on the 3.5" diskette or CDROM to install these utilities.

Once the XSTOOLs are installed you will see the following subdirectories:

XSTOOLS\BIN contains the executable programs for downloading to the XS40 Board and for applying signals to the XS40 Board through the printer port. An assembler for the microcontroller on the XS40 Board is also included.

XSTOOLS\DOCS contains the documentation and schematics for the XS40 Board.

Applying Power to Your XS40 Board

You can use your XS40 Board in two ways, distinguished by the method you use to apply power to the board.

Using a 9VDC wall-mount

You can use your XS40 Board all by itself to experiment with logic and microcontroller designs. Just place the XS40 Board on a non-conducting surface as shown in Figure 1. Then apply power to jack J9 of the XS40 Board from a 9V DC wall transformer with a 2.1 mm female, center-positive plug. (See Figure 2 for the location of jack J9 on your XS40 Board.) The on-board voltage regulation circuitry will create the voltages required by the rest of the XS40 Board circuitry.

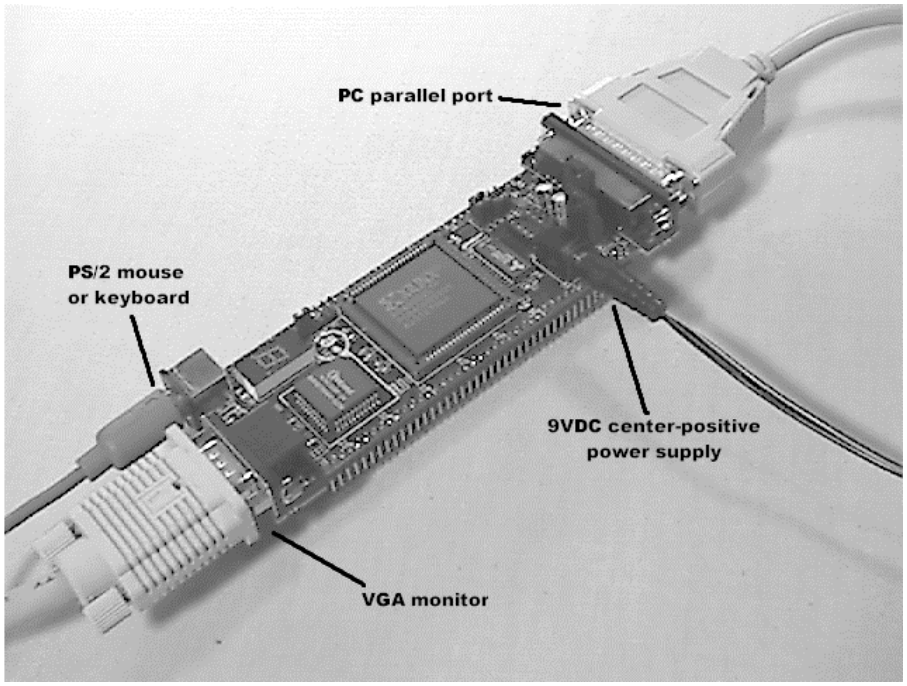
Solderless Breadboard Installation

The two rows of pins from your XS40 Board can be plugged into a solderless breadboard with holes spaced at 0.1" intervals. (One of the A.C.E. protoboards from 3M is a good choice.) Once plugged in, all the pins of the FPGA and microcontroller, and SRAM are accessible to other circuits on the breadboard. (The numbers printed next to the rows of pins on your XS40 Board correspond to the pin numbers of the FPGA.) Power can still be

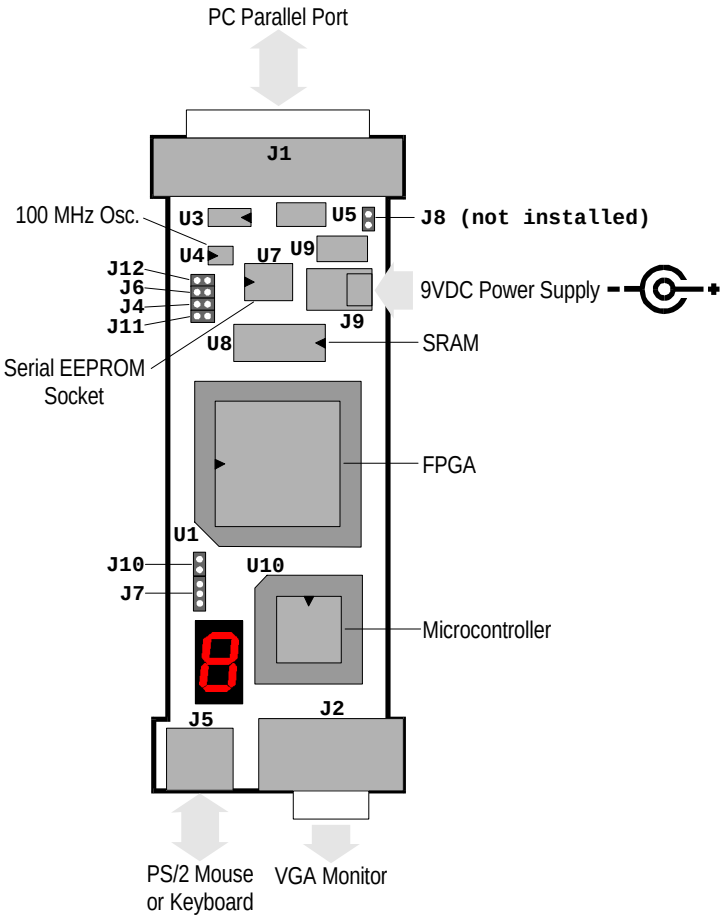
supplied to your XS40 Board though jack J9, or power can be applied directly through several pins on the underside of the board. Just connect +5V, +3.3V, and ground to the following pins for your particular type of XS40 Board. (You will need +3.3V only if your XS40 Board contains an XC4000XL type of FPGA.)

- Table 1: Power supply pins for the various XS40 Boards.

XS40 Board Type	GND Pin	+5V Pin	+3.3V Pin
XS40-005E V1.3	52	2, 54	none
XS40-005XL V1.3	52	2	54
XS40-010E V1.3	52	2, 54	none
XS40-010XL V1.3	52	2	54
XSP-010 V1.3	52	2,54	none



- Figure 1: External connections to the XS40 Board.



• Figure 2: Arrangement of components on the XS40 Board.

Connecting a PC to Your XS40 Board

The 6' cable included with your XS40 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 (J1) at the top of the XS40 Board as shown in Figure 1.

Connecting a VGA Monitor to Your XS40 Board

You can display images on a VGA monitor by connecting it to the 15-pin J2 connector at the bottom of your XS40 Board (see Figure 1). You will have to download a VGA driver circuit to your XS40 Board to actually display an image. You can find an example VGA driver at http://www.xess.com.

Connecting a Mouse or Keyboard to Your XS40 Board

You can accept inputs from a keyboard or mouse by connecting it to the J5 PS/2 connector at the bottom of your XS40 Board (see Figure 1). You can find an example keyboard driver at http://www.xess.com.

Setting the Jumpers on Your XS40 Board

The default jumper settings shown in Table 2 configure your XS40 Board for use in a logic design environment. You will need to change the jumper settings only if you are:

- using your XS40 in a stand-alone mode where it is unconnected from the PC parallel port (see page 10),
- reprogramming the clock frequency on your XS40 Board (see page 8),
- executing microcontroller code from internal ROM instead of the external SRAM on the XS40 Board. (You will have to replace the ROMless microcontroller on the XS40 Board with a ROM version to use this feature.)

• Table 2: Jumper settings for XS40 and XSP Boards.

Jumper	Setting	Purpose
J4	On (default)	A shunt should be installed if you are downloading the XS40 or XSP Board through the parallel port.
	Off	The shunt should be removed if the XS40 or XSP Board is being configured from the on-board serial EEPROM (U7).
J6	On	The shunt should be installed when the on-board serial EEPROM (U7) is being programmed.
	Off (default)	The shunt should be removed during normal board use.
J7	1-2 (ext) (default)	The shunt should be installed on pins 1 and 2 (ext) if the 8031 microcontroller program is stored in the external 32 KByte SRAM (U8) of the XS40 Board.
	2-3 (int)	The shunt should be installed on pins 2 and 3 (int) if the program is stored internally in the microcontroller.
J8	On	The shunt should be installed in XS40 or XSP Boards which use the 3.3V XC4000XL type of FPGAs.
	Off	The shunt should be removed on XS40 or XSP Boards which use the 5V XC4000E type of FPGAs.
J10	On	The shunt should be installed if the XS40 or XSP Board is being configured from the on-board serial EEPROM.
	Off (default)	The shunt should be removed if the XS40 or XSP Board is being downloaded from the PC parallel port.
J11	On (default)	The shunt should be installed if the XS40 or XSP Board is being downloaded from the PC parallel port.
	Off	The shunt should be removed if the XS40 or XSP Board is being configured from the on-board serial EEPROM.
J12	Off (default)	The shunt should be removed during normal operations when the programmable oscillator is generating a clock signal.
	On	The shunt should be installed when the programmable oscillator frequency is being set.

Testing Your XS40 Board

Once your XS40 Board is installed and the jumpers are in their default configuration, you can test the board by typing one of the commands listed in Table 3 into a DOS window.

- Table 3: Commands for testing the various types of XS40 Boards.

XS40 Board Type	Test Command
XS40-005E	XSTEST XS40-005E
XS40-005XL	XSTEST XS40-005XL
XS40-010E	XSTEST XS40-010E
XS40-010XL	XSTEST XS40-010XL
XSP-010	XSTEST XSP-010

The test procedure programs the FPGA, loads the SRAM with a test program for the microcontroller, and then the microcontroller executes this program. The total test period (including programming the board) is about 15 seconds for an XS40 Board. If the test completes successfully, then you will see a **O** displayed on the LED digit.

However, if the test program detects an error, then the LED digit displays an **E** or remains blank. In this case, check the following items:

- Make sure the XS40 Board is receiving power from a 9V DC power supply through jack J9 or through the VCC and GND pins.
- Check that the XS40 Board is sitting upon a non-conducting surface and that there are no connections to any of the pins (except for the VCC and GND pins if this is the way you are powering the board).
- Verify that the jumpers are in their default configuration.
- Make sure the downloading cable is securely attached to the XS40 Board and the PC parallel port.
- Verify that the parallel port is in ECP mode. (The mode is usually set in the BIOS as either SPP, EPP, ECP, or bidirectional. ECP mode works most reliably while bidirectional mode is not recommended.)

If all these checks are positive, then test the board using another PC. In our experience, 99.9% of all problems are due to the parallel port. If you cannot get your XS40 Board to pass the test even after taking these steps, then contact XESS Corp. to get a replacement board.

Programming the XS40 Board Clock Oscillator

The XS40 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 FPGA 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 XS40 Board. The following steps will store a particular divisor into the oscillator chip memory:

1) In a DOS window, use the following command with the type of XS40 Board and the clock divisor you want listed as arguments:

```
C:\> XSSETCLK XS40-005XL 8
```

The example shown above will set the programmable oscillator on an XS40-005XL Board to a frequency of 100 MHz / 8 = 12.5 MHz You may use any divisor between 1 and 2052 depending upon the clock frequency you want to use.

2) The XSSETCLK program will prompt you to remove the power and download cables from your XS40 Board. Then you should place a shunt on jumper J12. Then re-attach the download cable. Then reattach the power cable **only after the download cable is attached!**. When power is restored to the XS40 Board, the programmable oscillator will power up in its programming mode instead of generating a clock signal.

3) Press RETURN and the clock divisor will be programmed into the oscillator chip. If you wish to change the value of the divisor, you may re-issue the XSSETCLK command at this point with a new divisor without having to power-down the XS40 Board.

4) Finally, remove the power and download cables from your XS40 Board. Then remove the shunt from jumper J12. Then re-attach the download cable and the power cable. When power is restored to the XS40 Board, the programmable oscillator will power up in its active mode and output a clock signal at the programmed frequency.

3

Programming

This section will show you how to download a logic design from a PC into your XS40 Board and how to store a design in its optional serial EEPROM that will become active when power is applied.

Downloading Designs into Your XS40 Board

During the development and testing phases, you will usually connect the XS40 Board to the parallel port of a PC and download your circuit each time you make changes to it. You can download an FPGA design into your XS40 Board as follows:

```
C:\> XSLOAD CIRCUIT.BIT
```

where CIRCUIT.BIT is an XC4000 or Spartan bitstream file that contains the configuration for the XC4000 or XCS FPGA. This file is created using the XILINX Foundation software tools. Make sure the file contains a bitstream for the type of FPGA chip installed on your XS40 Board.

Use one of the following commands if you need to configure the FPGA and also download an Intel-formatted HEX file into the SRAM of the XS40 Board:

```
C:\> XSLOAD FILE.HEX CIRCUIT.BIT
```

where CIRCUIT.BIT is a bitstream file and FILE.HEX is a file containing hexadecimal data. The HEX file could contain microcontroller object code generated by the ASM51 assembler, or it could be arbitrary data from some other source. Whatever its source, the hexadecimal data is downloaded into the XS40 Board SRAM.

XSLOAD assumes the XS40 Board is connected to parallel port #1 of your PC. You can specify another port number using the -P option like so:

```
C:\> XSLOAD -P 2 FILE.HEX CIRCUIT.BIT
```

Storing Non-Volatile Designs in Your XS40 Board

Once your design is finished, you may want to store the design on the XS40 Board so that it is configured for operation as soon as power is applied.

The XC4000 or XCS FPGA on the XS40 Board stores its configuration in an on-chip SRAM which is erased whenever power is removed. You can place an external serial EEPROM in socket U7 which stores the FPGA configuration and reloads it on power-up. The XILINX XC1700 series of serial EEPROMs is a good choice for this, but you will need an external programmer to download your bitstream into the XC1700 chip. Also the XC1700 is one-time programmable (OTP), so you will need a new chip every time you change your logic design. Table 4 lists the serial EEPROM chip you need for storing the bitstream files for each type of XS40 Board.

- Table 4: Recommended XILINX serial EEPROMS for various types of XS40 Boards.

XS40 Board Type	Bitstream Size	XILINX EEPROM
XS40-005E	95,008	XC17128E
XS40-005XL	151,960	XC17256E
XS40-010E	178,144	XC17256E
XS40-010XL	283,424	XC1701
XSP-010	95,008	XC17S10

You also have the option of storing your design into an AT17C256 Atmel reprogrammable serial EEPROM if you have an XS40-005E, XS40-005XL, or XS40-010E Boards. The XS40 Board can directly program the Atmel chip and the FPGAs on these boards have bitstream files which are small enough to fit in the AT17C256. You can load your design into the Atmel EEPROM by following these steps:

1) Turn off power to the XS40 Board.

2) Place the Atmel AT17C256 EEPROM chip into the U7 socket.

3) Place a shunt on jumper J6. This enables the programming circuitry in the Atmel EEPROM chip.

4) Apply power to the XS40 Board.

5) Use the following command to load the FPGA bitstream file into the EEPROM:

```
C:\> XSLOAD –SERIAL_EEPROM CIRCUIT.BIT
```

It will take less than a minute to program the contents of the bitstream in CIRCUIT.BIT into the Atmel EEPROM.

6) Turn off power to the XS40 Board.

7) Remove the shunt on jumper J6. This disables the programming circuitry in the Atmel EEPROM chip so your design cannot be overwritten.

Once your design is loaded into an EEPROM, the following steps will make the XS40 Board configure itself from the EEPROM in socket U7 instead of the PC parallel port interface:

1) Remove the downloading cable from connector J1 of the XS40 Board. (As an alternative, you can use the command `XSPORT 0` to make sure the upper two data bits of the parallel port are at logic 0. These bits are connected to the mode pins of the FPGA and must be at logic 0 or the FPGA will not power-up in the active-serial mode.)

2) Place a shunt on jumper J10. This sets the FPGA into the active-serial mode so it will provide a clock signal to the EEPROM which sequences the loading of the configuration from the EEPROM into the FPGA.

3) Remove the shunts on jumpers J4 and J11. This prevents the PC interface circuitry on the XS40 Board from interfering with the clock and data signals from the FPGA.

4) Apply power to the XS40 Board. The FPGA will be configured from the serial EEPROM. You may reattach the downloading cable if you need to inject test signals into your design using the XSPORT program.

4

Programmer's Models

This section discusses the organization of components on the XS40 Board and introduces the concepts required to create applications that use both the microcontroller and the FPGA. Building FPGA-based designs is covered in detail in the *Practical Xilinx Designer Lab Book* by Prentice-Hall.

Microcontroller + FPGA Design Flow

The basic design flow for building microcontroller+FPGA applications is shown in Figure 3. Initially you have to get the specifications for the system you are trying to design. Then you have to determine what inputs are available to your system and what outputs it will generate.

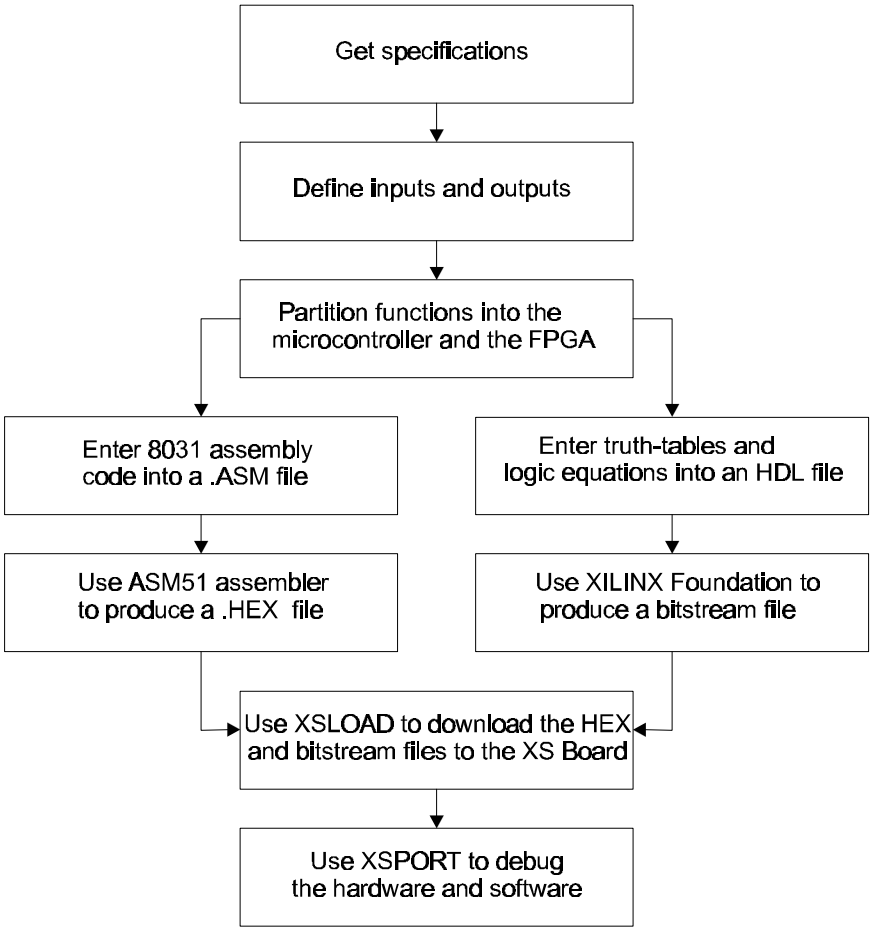
At this point, you have to partition the functions of your system between the microcontroller and the FPGA. Some of the input signals will go to the microcontroller, some will go to the FPGA, and some will go to both. Likewise, some of the outputs will be computed by the microcontroller and some by the FPGA. There will also be some new intra-system inputs and outputs created by the need for the microcontroller and the FPGA to cooperate.

In general, the FPGA will be used mainly for low-level functions where signal transitions occur more frequently and the control logic is simpler. A specialized serial transmitter/receiver would be a good example. Conversely, the microcontroller will be used for higher-level functions where the responses occur less quickly and the control logic is more complex. Reacting to commands passed in by the receiver is a good example. Once the design has been partitioned and you have assigned the various inputs, outputs, and functions to the microcontroller and the FPGA, then you can begin doing detailed design of the software and hardware. For the software, you can use your favorite editor to create a .ASM assembly-language file and assemble it with ASM51 to create a .HEX file for the microcontroller on the XS40 Board. For the FPGA hardware portion, you will enter truth-tables and logic equations into a .ABL or .VHDL file and compile it into an .BIT bitstream file using the XILINX Foundation software.

You can download the .HEX program file and the .BIT bitstream file to the XS40 Board using the XSLOAD program. XSLOAD stores the contents of the .HEX file into the SRAM on the XS40 Board and then it reconfigures the FPGA by loading it with the bitstream file.

When the XS40 Board is loaded with the hardware and software, you need to test it to see if it really works. The answer usually starts as "No" so you need a method of injecting test signals and observing the results. XSPORT is a simple program that lets you send test

signals to the XS40 Board through the PC parallel port. You can trace the reaction of your system to signals from the parallel port by programming the microcontroller and the FPGA to output status information on the LED digit (much like placing "printf" statements in your C language programs). This is admittedly crude but will serve if you don't have access to a programmable stimulus generator or logic analyzer.



• Figure 3: FPLD+microcontroller design flow.

XS40 Board Component Interconnections

The microcontroller and the FPGA on the XS40 Board are already connected together. These pre-existing connections save you the effort of having to wire them yourself, but they also impose limitations on how your microcontroller program and the FPGA hardware will interact. A high-level view of how the microcontroller, SRAM, and FPGA on the XS40 Board are connected is shown on the following pages. A more detailed schematic is also presented at the end of this manual.

The programmable oscillator output goes directly to a synchronous clock input of the FPGA. The FPGA uses this clock to generate a clock that it sends to the XTAL1 clock input of the microcontroller.

The microcontroller multiplexes the lower eight bits of a memory address with eight bits of data and outputs this on its P0 port. Both the SRAM data lines and the FPGA are connected to P0. The SRAM uses this connection to send and receive data to and from the microcontroller. The FPGA is programmed to latch the address output on P0 under control of the ALE signal and send the latched address bits to the lower eight address lines of the SRAM.

Meanwhile, the upper eight bits of the address are output on the P2 port of the microcontroller. The 32 KByte SRAM on the XS40 Board uses the lower seven of these address bits. The FPGA also receives the upper eight address bits and decodes these along with the PSENB and read/write control line (from pin P3.6 of port P3) from the microcontroller to generate the CEB and OEB signals that enable the SRAM and its output drivers, respectively. Either of the CEB or OEB signals can be pulled high to disable the SRAM and prevent it from having any effect on the rest of the XS40 Board circuitry.

One of the outputs of the FPGA controls the reset line of the microcontroller. The microcontroller can be prevented from having any effect on the rest of the circuitry by forcing the RST pin high through the FPGA. (When RST is active, the microcontroller pins are weakly pulled high.)

Many of the I/O pins of ports P1 and P3 of the microcontroller connect to the FPGA and can be used for general-purpose I/O between the microcontroller and the FPGA. In addition to being general-purpose I/O, the P3 pins also have special functions such as serial transmitters, receivers, interrupt inputs, timer inputs, and external SRAM read/write control signals. If you aren't using a particular special function, then you can use the associated pin for general-purpose I/O between the microcontroller and the FPGA. In many cases, however, you will program the FPGA to make use of the special-purpose microcontroller pins. (For example, the FPGA could generate microcontroller interrupts.) If you want to drive the special-purpose pin from an external circuit, then the FPGA I/O pin connected to it must be tristated.

A seven-segment LED digit connects directly to the FPGA. (These same FPGA pins can also drive a VGA monitor.) The FPGA can be programmed so the microcontroller can control the LEDs either through P1 or P3 or by memory-mapping a latch for the LED into the memory space of the microcontroller.

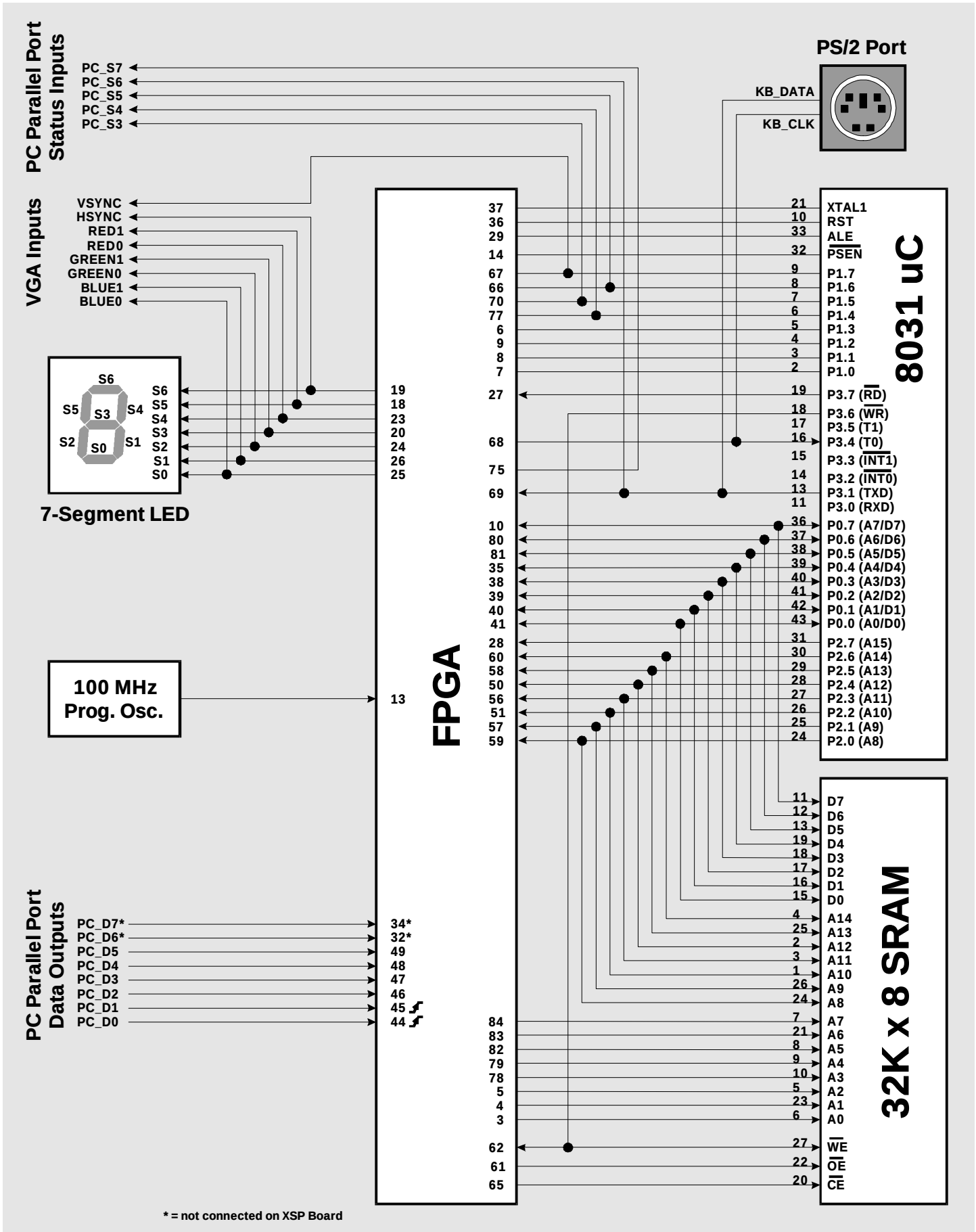
The PC can transmit signals to the XS40 Board through the eight data output bits of the parallel port. The FPGA has direct access to these signals. The microcontroller can also access these signals if you program the FPGA to pass them onto the FPGA I/O pins connected to the microcontroller.

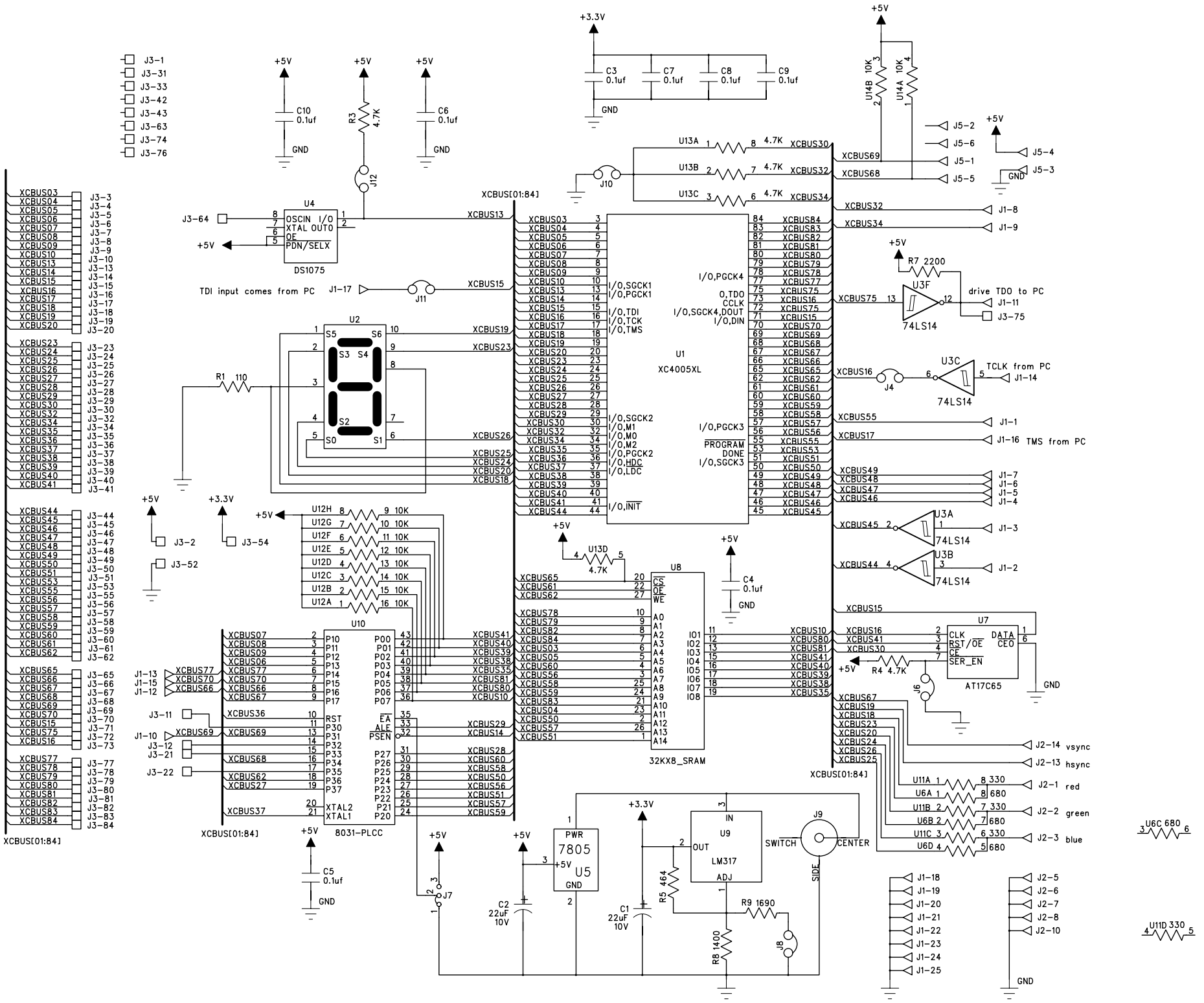
Communication from the XS40 Board back to the PC also occurs through the parallel port. The parallel port status pins are connected to pins of microcontroller ports P1 and P3 . Either the microcontroller or the FPGA can drive the status pins. The PC can read the status pins to fetch data from the XS40 Board.

The FPGA also has access to the clock and data lines of a keyboard or mouse attached to the PS/2 port of the board.

- Table 5: XS40 Board pin descriptions.

XS40 Pin	Connects to...	Description
25	S0, BLUE0	These pins drive the individual segments of the LED display (S0-S6). They also drive the color and horizontal sync signals for a VGA monitor.
26	S1, BLUE1	
24	S2, GREEN0	
20	S3, GREEN1	
23	S4, RED0	
18	S5, RED1	
19	S6, HSYNCB	
13	CLK	An input driven by the 100 MHz programmable oscillator.
44	PC_D0	These pins are driven by the data output pins of the PC parallel port. Clocking signals can only be reliably applied through pins 44 and 45 since these have additional hysterisis circuitry. Pins 32 and 34 are mode signals for the FPGA so you must adjust your design to account for the way that the Foundation tools handle these pins.
45	PC_D1	
46	PC_D2	
47	PC_D3	
48	PC_D4	
49	PC_D5	
32	PC_D6	
34	PC_D7	
37	XTAL1	Pin that drives the uC clock input.
36	RST	Pin that drives the uC reset input.
29	ALEB	Pin that monitors the uC address latch enable.
14	PSENB	Pin that monitors the uC program store enable.
7	P1.0	These pins connect to the pins of Port 1 of the uC. Some of the pins are also connected to the status input pins of the PC parallel port. Pin 67 drives the vertical sync signal for a VGA monitor.
8	P1.1	
9	P1.2	
6	P1.3	
77	P1.4, PC_S4	
70	P1.5, PC_S3	
66	P1.6, PC_S5	
67	P1.7, VSYNCB	
69	P3.1(TXD), PC_S6,	These pins connect to some of the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated in parentheses. Pin 62 connects to the data write pin of the uC and the write-enable pin of the SRAM. Pin 69 connects to a status input pin of the PC parallel port and the PS/2 data line. Pin 68 connects to the PS/2 clock line.
68	P3.4(T0),PS/2 CLK	
62	P3.6(WRB), WEB	
27	P3.7(RDB)	
41	P0.0(AD0), D0	These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to the data pins of the SRAM.
40	P0.1(AD1), D1	
39	P0.2(AD2), D2	
38	P0.3(AD3), D3	
35	P0.4(AD4), D4	
81	P0.5(AD5), D5	
80	P0.6(AD6), D6	
10	P0.7(AD7), D7	
59	P2.0(A8), A8	These pins connect to Port 2 of the uC which also outputs the upper address byte. These pins also connect to the 7 upper address bits of the SRAM.
57	P2.0(A9), A9	
51	P2.0(A10), A10	
56	P2.0(A11), A11	
50	P2.0(A12), A12	
58	P2.0(A13), A13	
60	P2.0(A14), A14	
28	P2.0(A15)	
3	A0	These pins drive the 8 lower address bits of the SRAM.
4	A1	
5	A2	
78	A3	
79	A4	
82	A5	
83	A6	
84	A7	
61	OEB	Pin that drives the SRAM output enable.
65	CEB	Pin that drives the SRAM chip enable.
75	PC_S7	Pin that drives a status input pin of the PC parallel port.


PC Parallel Port Status Inputs
PC_S7
PC_S6
PC_S5
PC_S4
PC_S3
PS/2 Port
KB_DATA
KB_CLK
VGA Inputs
VSYNC
HSYNC
RED1
RED0
GREEN1
GREEN0
BLUE1
BLUE0
S6
S5
S4
S3
S2
S1
S0
7-Segment LED
100 MHz Prog. Osc.
FPGA
8031 uC
XTAL1
RST
ALE
PSEN
P1.7
P1.6
P1.5
P1.4
P1.3
P1.2
P1.1
P1.0
P3.7 (RD)
P3.6 (WR)
P3.5 (T1)
P3.4 (T0)
P3.3 (INT1)
P3.2 (INT0)
P3.1 (TXD)
P3.0 (RXD)
P0.7 (A7/D7)
P0.6 (A6/D6)
P0.5 (A5/D5)
P0.4 (A4/D4)
P0.3 (A3/D3)
P0.2 (A2/D2)
P0.1 (A1/D1)
P0.0 (A0/D0)
P2.7 (A15)
P2.6 (A14)
P2.5 (A13)
P2.4 (A12)
P2.3 (A11)
P2.2 (A10)
P2.1 (A9)
P2.0 (A8)
32K x 8 SRAM
D7
D6
D5
D4
D3
D2
D1
D0
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
WE
OE
CE
PC Parallel Port Data Outputs
PC_D7*
PC_D6*
PC_D5
PC_D4
PC_D3
PC_D2
PC_D1
PC_D0
* = not connected on XSP Board


XS40, XSP Board V1.3 Schematic
U1
XC4005XL
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
DS1075
74LS14
AT17C65
32KX8_SRAM
8031-PLCC
LM317
7805
XCBUS[01:84]
TDI input comes from PC
TMS from PC
TCLK from PC
drive TDO to PC
vsync
hsync
red
green
blue
SWITCH
CENTER
SIDE