XStend Board Manual

XESS Corporation

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.

Table of Contents

Getting Help!

If you follow the instructions in this manual and you encounter problems, here are some places to get help:

- If you can't get the XS 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/FPGA`.

- If you can't get your XILINX 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/searchtd.htm`.

1 XStend Overview

The XS40 and XS95 Boards offer a flexible, low-cost method of prototyping FPGA and CPLD designs. However, their small physical size limits the amount of support circuitry they can hold. The XStend Board removes this limitation by providing additional support circuitry that the XS40 and XS95 Boards can access through their breadboard interfaces.

The XStend Board contains resources that extend the range of applications of the XS Boards into three areas:

- The pushbuttons, DIP switches, LEDs, and prototyping area are useful for basic lab experiments. These features in combination with the XS Boards replicates the functionality of the older HW/UW FPGABOARD.

- The VGA monitor interface, PS/2 keyboard/mouse interface, and static RAM let the XS Boards be used in video and computing experiments.

- The stereo codec and dual-channel analog input/output circuitry are useful for processing of audio signals in combination with DSP circuits synthesized with XILINX's CORE generation software.

2 XStend Board Features

The XStend Board extends the capabilities of the XS40 and XS95 Boards by providing

- mounting receptacles for both an XS40 and an XS95 Board;

- a 3"×3" prototyping area;

- a 42×2 connector for add-on daughterboards.

- additional LEDs and LED displays;

- pushbutton and DIP switches;
- an interface to VGA monitors;
- an interface to a PS/2-style keyboard or mouse;
- an additional 64 Kbytes of static RAM;
- a stereo codec with left/right input and output channels;

Each of these resources will be described below.

2.1 XS40/XS95 Board Mounting Area

An XS40 or XS95 Board is mounted on the XStend Board using mounting receptacle J1 or J2, respectively. These receptacles mate with the breadboard interface of the XS Boards to give them access to all the resources of the XStend Board. The XS Boards also provide power to the XStend Board through these receptacles.

☛***Warning: Version 1.0 of the XS40 Board with a 3.3V XC4000XL FPGA will not work with the XStend Board! You must replace the XC4000XL FPGA with an XC4000E FPGA and remove the J8 jumper to switch the board to 5V operation.***

To use an XS40 Board with the XStend Board, insert it into the right-most columns of the mounting receptacles. When using an XS95 Board, however, you should insert it into the left-most columns. There are markings on the XStend Board to indicate which column is ocuppied by each type of XS Board.

2.2 LEDs

The XStend Board provides an additional eight LEDs (D1—D8) and two more LED displays (U1 and U2) for use by the XS Boards. All of these LEDs are active-low meaning that an LED or LED display segment will glow when a logic-low is applied to it.

The LEDs are disabled by removing the shunts on the following jumpers:

Table 1: Jumper settings for XStend LEDs.

Jumper	Setting
J8	Removing the shunt on this jumper disconnects the power from LEDs D1—D8.
J4	Removing the shunt on this jumper disconnects the power to the left LED display U1.

J7	Removing the shunt on this jumper disconnects the power to the right LED display U2.

Here are the connections from the XS40 and XS95 Boards to the LEDs on the XStend Board (expressed as UCF constraints):

Listing 1: Connections between the XStend LEDs and the XS40.

```
# LEFT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
NET LS_0          LOC=P3;
NET LS_1          LOC=P4;
NET LS_2          LOC=P5;
NET LS_3          LOC=P78;
NET LS_4          LOC=P79;
NET LS_5          LOC=P82;
NET LS_6          LOC=P83;
NET LDP_          LOC=P84;
#
# RIGHT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
NET RS_0          LOC=P59;
NET RS_1          LOC=P57;
NET RS_2          LOC=P51;
NET RS_3          LOC=P56;
NET RS_4          LOC=P50;
NET RS_5          LOC=P58;
NET RS_6          LOC=P60;
NET RDP_          LOC=P28;
#
# INDIVIDUAL LED CONNECTIONS (ACTIVE-LOW)
NET D_1           LOC=P41;
NET D_2           LOC=P40;
NET D_3           LOC=P39;
NET D_4           LOC=P38;
NET D_5           LOC=P35;
NET D_6           LOC=P81;
NET D_7           LOC=P80;
NET D_8           LOC=P10;
```

Listing 2: Connections between the XStend LEDs and the XS95.

```
# LEFT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
NET LS_0          LOC=P1;
NET LS_1          LOC=P2;
NET LS_2          LOC=P3;
NET LS_3          LOC=P75;
NET LS_4          LOC=P79;
NET LS_5          LOC=P82;
NET LS_6          LOC=P83;
NET LDP_          LOC=P84;
#
# RIGHT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
NET RS_0          LOC=P58;
NET RS_1          LOC=P56;
NET RS_2          LOC=P54;
NET RS_3          LOC=P55;
NET RS_4          LOC=P53;
NET RS_5          LOC=P57;
NET RS_6          LOC=P61;
NET RDP_          LOC=P34;
#
# INDIVIDUAL LED CONNECTIONS (ACTIVE-LOW)
NET D_1           LOC=P44;
NET D_2           LOC=P43;
NET D_3           LOC=P41;
NET D_4           LOC=P40;
NET D_5           LOC=P39;
NET D_6           LOC=P37;
NET D_7           LOC=P36;
NET D_8           LOC=P35;
```

2.3 Switches

The XStend has a bank of eight DIP switches and two pushbuttons (labelled SPARE and RESET) that are accessible from the XS Boards. (There is a third pushbutton labelled PROGRAM which is used to initiate the programming of the XS40 Board. It is not intended to be a general-purpose input.)

When closed, each DIP switch pulls the connected pin of the XS Board to ground. When the DIP switch is open, the pin is pulled high through a 10KWresistor.

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

When pressed, each pushbutton pulls the connected pin of the XS Board to ground. Otherwise, the pin is pulled high through a 10 KW resistor.

Here are the connections from the XS40 and XS95 Boards to the switches on the XStend Board expressed as UCF constraints (for the UCF syntax and usage tips, check out http://www.xilinx.com/techdocs/2449.htm):

Listing 3: Connections between the XStend DIP and pushbutton switches and the XS40.

```
# DIP SWITCH CONNECTIONS
NET DIPSW1        LOC=P7;
NET DIPSW2        LOC=P8;
NET DIPSW3        LOC=P9;
NET DIPSW4        LOC=P6;
NET DIPSW5        LOC=P77;
NET DIPSW6        LOC=P70;
NET DIPSW7        LOC=P66;
NET DIPSW8        LOC=P69;
#
# PUSHBUTTON SWITCH CONNECTIONS (ACTIVE-LOW)
NET PUSH_SPARE_   LOC=P67;
NET PUSH_RESET_   LOC=P37;
```

Listing 4: Connections between the XStend DIP and pushbutton switches and the XS95.

```
# DIP SWITCH CONNECTIONS
NET DIPSW1        LOC=P6;
NET DIPSW2        LOC=P7;
NET DIPSW3        LOC=P11;
NET DIPSW4        LOC=P5;
NET DIPSW5        LOC=P72;
NET DIPSW6        LOC=P71;
NET DIPSW7        LOC=P66;
NET DIPSW8        LOC=P70;
#
# PUSHBUTTON SWITCH CONNECTIONS (ACTIVE-LOW)
NET PUSH_SPARE_   LOC=P67;
NET PUSH_RESET_   LOC=P10;
```

2.4 VGA Interface

The XStend Board provides the XS Board with an interface to a VGA monitor through connector J5. The XS Board can drive the active-low horizontal and vertical sync signals and control the width and height of the video frame. The XS Board also has access to two bits each of red, green, and blue color signals so it can generate pixels in any of $2^2 \times 2^2 \times 2^2 = 64$ different colors.

Here are the connections from the XS40 and XS95 Boards to the VGA interface of the XStend Board (expressed as UCF constraints):

Listing 5: Connections between the XStend VGA interface and the XS40.

```
# VGA CONNECTIONS
NET VSYNC_        LOC=P67;
NET HSYNC_        LOC=P19;
NET RED1          LOC=P18;
NET RED0          LOC=P23;
NET GREEN1        LOC=P20;
NET GREEN0        LOC=P24;
NET BLUE1         LOC=P26;
NET BLUE0         LOC=P25;
```

Listing 6: Connections between the XStend VGA interface and the XS95.

```
# VGA CONNECTIONS
NET VSYNC_        LOC=P24;
NET HSYNC_        LOC=P15;
NET RED1          LOC=P14;
NET RED0          LOC=P18;
NET GREEN1        LOC=P17;
NET GREEN0        LOC=P19;
NET BLUE1         LOC=P23;
NET BLUE0         LOC=P21;
```

2.5 PS/2 Keyboard Interface

The XStend Board provides the XS Board with a PS/2-style interface (mini-DIN connector J6) to either a keyboard or a mouse. The XS Board 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.

Here are the connections from the XS40 and XS95 Boards to the PS/2 interface of the XStend Board (expressed as UCF constraints):

Listing 7: Connections between the XStend PS/2 interface and the XS40.

```
# PS/2 KEYBOARD CONNECTIONS
NET KB_DATA       LOC=P69;
NET KB_CLK        LOC=P68;
```

Listing 8: Connections between the XStend PS/2 interface and the XS95.

```
# PS/2 KEYBOARD CONNECTIONS
NET KB_DATA       LOC=P70;
NET KB_CLK        LOC=P26;
```

2.6 RAMs

The XStend Board adds an additional 64 KBytes of RAM to the 32 KBytes already on the XS Board. The XStend RAM connects to the same pins as the XS Board RAM for the address bus, data bus, write-enable, and output-enable. The chip-selects of the XStend Board RAMs are connected to different pins so all the RAMs can be individually selected.

The XStend RAMs are disabled by removing the shunts on the following jumpers:

Table 2: Jumper settings for XStend RAMs.

Jumper	Setting
J16	Removing the shunt on this jumper disables the left RAM U5 by pulling its chip-select pin high.
J17	Removing the shunt on this jumper disables the right RAM U6 by pulling its chip-select pin high.

Here are the connections from the XS40 and XS95 Boards to their own RAMs and the RAMs of the XStend Board (expressed as UCF constraints):

Listing 9: Connections between the XStend RAMs and the XS40.

```
NET AD0           LOC=P41;    # DATA BUS
NET AD1           LOC=P40;
NET AD2           LOC=P39;
NET AD3           LOC=P38;
NET AD4           LOC=P35;
NET AD5           LOC=P81;
NET AD6           LOC=P80;
NET AD7           LOC=P10;
NET A0            LOC=P3;     # LOWER BYTE OF ADDRESS
NET A1            LOC=P4;
NET A2            LOC=P5;
NET A3            LOC=P78;
NET A4            LOC=P79;
NET A5            LOC=P82;
NET A6            LOC=P83;
NET A7            LOC=P84;
NET A8            LOC=P59;    # UPPER BYTE OF ADDRESS
NET A9            LOC=P57;
NET A10           LOC=P51;
NET A11           LOC=P56;
NET A12           LOC=P50;
NET A13           LOC=P58;
NET A14           LOC=P60;
NET A15           LOC=P28;
NET WR_           LOC=P62;    # ACTIVE-LOW WRITE-ENABLE FOR ALL RAMS
NET OE_           LOC=P61;    # ACTIVE-LOW OUTPUT-ENABLE FOR ALL RAMS
NET CE_           LOC=P65;    # ACTIVE-LOW CHIP-ENABLE FOR XS40 RAM
NET LCE_          LOC=P7;     # ACTIVE-LOW CHIP-ENABLE FOR LEFT XSTEND RAM
NET RCE_          LOC=P8;     # ACTIVE-LOW CHIP-ENABLE FOR RIGHT XSTEND RAM
```

Listing 10: Connections between the XStend RAMs and the XS95.

```
NET AD0           LOC=P44;    # DATA BUS
NET AD1           LOC=P43;
NET AD2           LOC=P41;
NET AD3           LOC=P40;
NET AD4           LOC=P39;
NET AD5           LOC=P37;
```

```
NET AD6           LOC=P36;
NET AD7           LOC=P35;
NET A0            LOC=P1;     # LOWER BYTE OF ADDRESS
NET A1            LOC=P2;
NET A2            LOC=P3;
NET A3            LOC=P75;
NET A4            LOC=P79;
NET A5            LOC=P82;
NET A6            LOC=P83;
NET A7            LOC=P84;
NET A8            LOC=P58;    # UPPER BYTE OF ADDRESS
NET A9            LOC=P56;
NET A10           LOC=P54;
NET A11           LOC=P55;
NET A12           LOC=P53;
NET A13           LOC=P57;
NET A14           LOC=P61;
NET A15           LOC=P34;
NET WR_           LOC=P63;    # ACTIVE-LOW WRITE-ENABLE FOR ALL RAMS
NET OE_           LOC=P62;    # ACTIVE-LOW OUTPUT-ENABLE FOR ALL RAMS
NET CE_           LOC=P65;    # ACTIVE-LOW CHIP-ENABLE FOR XS95 RAM
NET LCE_          LOC=P6;     # ACTIVE-LOW CHIP-ENABLE FOR LEFT XSTEND RAM
NET RCE_          LOC=P7;     # ACTIVE-LOW CHIP-ENABLE FOR RIGHT XSTEND RAM
```

2.7 Stereo Codec

The XStend Board has a stereo codec that accepts two analog input channels from jack J9, digitizes the analog values, and sends the digital values to the XS Board 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 XStend Board through jack J10.

The codec is configured by placing shunts on the following jumpers:

Table 3: Jumper settings for XStend codec.

Jumper	Setting
J11	Placing a shunt on this jumper disables the codec by holding it in the reset state.
J13, J15	Placing shunts across two of the three pins of these jumpers selects the digital de-emphasis for different sampling rates:
0 0	De-emphasis for 32 KHz
0 1	De-emphasis for 44.1 KHz
1 0	De-emphasis for 48 KHz
1 1	De-emphasis off

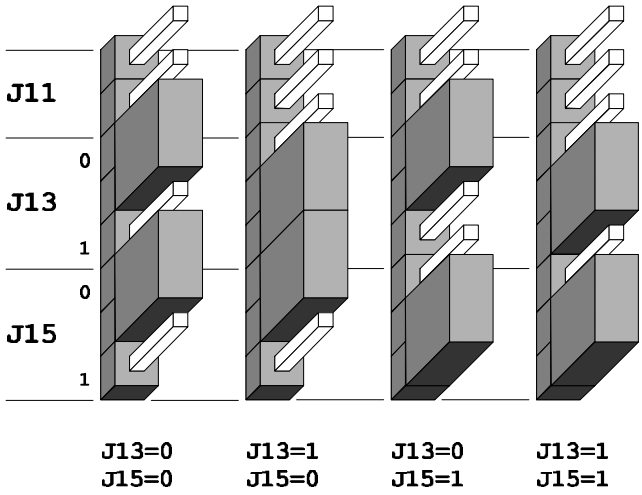


Figure 1: Jumper settings for codec sampling rate de-emphasis

Here are the connections from the XS40 Board to the codec interface on the XStend Board (expressed as UCF constraints):

Listing 11: Connections between the XStend stereo codec and the XS40.

```
# STEREO CODEC CONNECTIONS
NET SDOUT         LOC=P66;
NET MCLK          LOC=P9;
NET LRCK          LOC=P6;
NET SCLK          LOC=P77;
NET SDIN          LOC=P70;
NET CCLK          LOC=P44;
NET CDIN          LOC=P45;
NET CS_           LOC=P46;
```

Listing 12: Connections between the XStend codec and the XS95.

```
# STEREO CODEC CONNECTIONS
NET SDOUT         LOC=P66;
NET MCLK          LOC=P11;
NET LRCK          LOC=P5;
NET SCLK          LOC=P72;
NET SDIN          LOC=P71;
NET CCLK          LOC=P46;
NET CDIN          LOC=P47;
NET CS_           LOC=P48;
```

2.8 Prototyping Area

The XStend Board has a prototyping area consisting of component through-holes on an 0.1"×0.1" grid interspersed with a network of +5V and GND buses.

☛***To prevent damage, remove the XS40 or XS95 Board from the XStend Board when constructing circuitry in the prototyping area.***

Connections from the XS Board to the prototyping area are made through connector J3. The arrangement of pins on this connector exactly matches the arrangement of pins on the XS40 Board. For example, the pin at the bottom-left of J3 on the XStend Board corresponds to pin 21 at the bottom-left of the XS40 Board.

The XS95 Board has a completely different pin arrangement than the XS40. Therefore each pin on J3 is explicitly labelled with the corresponding pin number on the XS95 Board. For example, the pin at the bottom-left of J3 on the XStend Board is connected to pin 68 near the top-left of the XS95 Board.

2.9 Daughterboard Connector

Daughterboards with specialized circuitry can be connected to the XStend board through connector J18. This 42×2 connector brings all the I/O and power/GND from the XS40 or XS95 Board to the daughterboard.

3 Example Designs for the XStend Board

Here are several examples of designs built using the XStend Board coupled with an XS40 or XS95 Board.

3.1 Using the LEDs and Switches

This example creates a circuit that displays the settings of the DIP switches on the LEDs and LED displays. The particular set of LEDs which is activated is selected by the SPARE and RESET pushbuttons. The ABEL code for this example is shown in **Listing 13**; **Listing 14** and **Listing 15** show the UCF files for using the XS40 and XS95 Boards with the XStend Board, respectively.

The design files for this example should be found in C:\XST-PROJ\EXP1. The steps for compiling and testing the design using an XS40 combined with an XStend Board are as follows:

- Synthesize the ABEL code in the EXP1.ABL for an XC4005XL FPGA.
- Compile the synthesized netlist using the EXP1_40.UCF constraint file.
- Mount an XS40 Board in the XStend Board and attach the downloading cable from the XS40 to the PC parallel port. Apply 9VDC though jack J9 of the XS40. Place shunts on jumpers J4, J7, and J8 of the XStend Board to enable the LED displays. Place a shunt on jumper J11 to keep the XStend codec disabled.
- Download the EXP1.BIT file into the XS40/XStend combination with the command: `XSLOAD EXP1.BIT`.

- Set the DIP switches and press the SPARE and RESET pushbuttons. Observe the results on the LEDs.

The steps for compiling and testing the design using an XS95 combined with an XStend Board are as follows:

- Synthesize the ABEL code in the EXP1.ABL for an XC95108 CPLD.
- Compile the synthesized netlist using the EXP1_95.UCF constraint file.
- Generate an SVF file for the design.
- Mount an XS95 Board in the XStend Board and attach the downloading cable from the XS95 to the PC parallel port. Apply 9VDC though jack J9 of the XS95. Place shunts on jumpers J4, J7, and J8 of the XStend Board to enable the LED displays. Place a shunt on jumper J11 to keep the XStend codec disabled.
- Download the EXP1.SVF file into the XS95/XStend combination with the command: `XSLOAD EXP1.SVF`.
- Set the DIP switches and press the SPARE and RESET pushbuttons. Observe the results on the LEDs.

Listing 13: ABEL code for testing the XStend LEDs and switches.

```
001- MODULE EXP1
002- TITLE 'EXP1'
003-
004- // This example shows the settings of the DIP switches using each
005- // of the LED displays as selected by the SPARE and RESET pushbuttons
006-
007- DECLARATIONS
008-
009- OE_ PIN;                        // output enable for the RAMs
010- RST PIN;                        // uC reset
011- DIPSW8..DIPSW1 PIN;             // DIP switch inputs
012- DIPSW = [DIPSW8..DIPSW1];
013- PUSH_SPARE_ PIN;               // SPARE pushbutton input
014- PUSH_RESET_ PIN;               // RESET pushbutton input
015- S6..S0 PIN;                    // LED display on XS Board
016- S = [S6..S0];
017- LS_6..LS_0 PIN;                // left LED display on XStend Board
018- LS_ = [LS_6..LS_0];
019- LDP_ PIN;                      // decimal point on left LED display
020- RS_6..RS_0 PIN;                // right LED display on Xstend board
021- RS_ = [RS_6..RS_0];
022- RDP_ PIN;                      // decimal point on right LED display
023- D_8..D_1 PIN;                  // string of LEDs on XStend Board
024- D_ = [D_8..D_1];
025-
026- EQUATIONS
027-
028- RST = 1;     // keep the uC in the reset state
```

```
029- OE_ = 1;     // disable the outputs from all the RAMs
030-
031- // the state of the pushbuttons will select which LED display will
032- // show the settings of the DIP switches.  A bright LED corresponds
033- // to a DIP switch which is set to '1'.
034-
035- // When both pushbuttons are pressed, the DIP switch settings are
036- // shown on the LED display of the XS Board.
037- WHEN ((PUSH_SPARE_==0) & (PUSH_RESET_==0)) THEN {
038-        S = [DIPSW7..DIPSW1];
039-        [LDP_,RS_6..RS_0] = ^B11111111;
040-        [RDP_,LS_6..LS_0] = ^B11111111;
041-        D_ = ^B11111111;
042-        }
043- // When only the RESET pushbutton is pressed, the DIP switch settings
044- // are shown on the left LED display on the XStend Board.
045- ELSE WHEN ((PUSH_SPARE_==1) & (PUSH_RESET_==0)) THEN {
046-        S = ^B0000000;
047-        [LDP_,LS_6..LS_0] = !(DIPSW);
048-        [RDP_,RS_6..RS_0] = ^B11111111;
049-        D_ = ^B11111111;
050-        }
051- // When only the SPARE pushbutton is pressed, the DIP switch settings
052- // are shown on the right LED display on the XStend Board.
053- ELSE WHEN ((PUSH_SPARE_==0) & (PUSH_RESET_==1)) THEN {
054-        S = ^B0000000;
055-        [LDP_,LS_6..LS_0] = ^B11111111;
056-        [RDP_,RS_6..RS_0] = !(DIPSW);
057-        D_ = ^B11111111;
058-        }
059- // When neither pushbuttons is pressed, the DIP switch settings are
060- // shown on the string of LEDs of the XStend Board.
061- ELSE WHEN ((PUSH_SPARE_==1) & (PUSH_RESET_==1)) THEN {
062-        S = ^B0000000;
063-        [LDP_,LS_6..LS_0] = ^B11111111;
064-        [RDP_,RS_6..RS_0] = ^B11111111;
065-        D_ = !(DIPSW);
066-        }
067-
068- END EXP1
```

Listing 14: UCF file for LED/switch XStend example with XS40.

```
001- # XS40 BOARD LED CONNECTIONS (ACTIVE_HIGH)
002- NET S0       LOC=P25;
003- NET S1       LOC=P26;
004- NET S2       LOC=P24;
005- NET S3       LOC=P20;
006- NET S4       LOC=P23;
007- NET S5       LOC=P18;
008- NET S6       LOC=P19;
009- #
010- # MICROCONTROLLER PINS
011- NET RST      LOC=P36; # ACTIVE-HIGH RESET
012- #
013- # XS40 BOARD RAM CONTROL PINS
014- NET OE_      LOC=P61; # ACTIVE-LOW OUTPUT ENABLE
015- #
016- #
```

```
017- # XSTEND BOARD CONNECTIONS
018- #
019- # DIP SWITCH CONNECTIONS
020- NET DIPSW1   LOC=P7;
021- NET DIPSW2   LOC=P8;
022- NET DIPSW3   LOC=P9;
023- NET DIPSW4   LOC=P6;
024- NET DIPSW5   LOC=P77;
025- NET DIPSW6   LOC=P70;
026- NET DIPSW7   LOC=P66;
027- NET DIPSW8   LOC=P69;
028- #
029- # PUSHBUTTON SWITCH CONNECTIONS (ACTIVE-LOW)
030- NET PUSH_SPARE_    LOC=P67;
031- NET PUSH_RESET_    LOC=P37;
032- #
033- # LEFT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
034- NET LS_0     LOC=P3;
035- NET LS_1     LOC=P4;
036- NET LS_2     LOC=P5;
037- NET LS_3     LOC=P78;
038- NET LS_4     LOC=P79;
039- NET LS_5     LOC=P82;
040- NET LS_6     LOC=P83;
041- NET LDP_     LOC=P84;
042- #
043- # RIGHT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
044- NET RS_0     LOC=P59;
045- NET RS_1     LOC=P57;
046- NET RS_2     LOC=P51;
047- NET RS_3     LOC=P56;
048- NET RS_4     LOC=P50;
049- NET RS_5     LOC=P58;
050- NET RS_6     LOC=P60;
051- NET RDP_     LOC=P28;
052- #
053- # INDIVIDUAL LED CONNECTIONS (ACTIVE-LOW)
054- NET D_1      LOC=P41;
055- NET D_2      LOC=P40;
056- NET D_3      LOC=P39;
057- NET D_4      LOC=P38;
058- NET D_5      LOC=P35;
059- NET D_6      LOC=P81;
060- NET D_7      LOC=P80;
061- NET D_8      LOC=P10;
```

Listing 15: UCF file for LED/switch XStend example with XS95.

```
001- # XS95 BOARD LED CONNECTIONS (ACTIVE_HIGH)
002- NET S0       LOC=P21;
003- NET S1       LOC=P23;
004- NET S2       LOC=P19;
005- NET S3       LOC=P17;
006- NET S4       LOC=P18;
007- NET S5       LOC=P14;
008- NET S6       LOC=P15;
009- #
010- # MICROCONTROLLER PINS
011- NET RST      LOC=P45; # ACTIVE-HIGH RESET
```

```
012- #
013- # XS95 BOARD RAM CONTROL PINS
014- NET OE_      LOC=P62; # ACTIVE-LOW OUTPUT ENABLE
015- #
016- #
017- # XSTEND BOARD CONNECTIONS
018- #
019- # DIP SWITCH CONNECTIONS
020- NET DIPSW1   LOC=P6;
021- NET DIPSW2   LOC=P7;
022- NET DIPSW3   LOC=P11;
023- NET DIPSW4   LOC=P5;
024- NET DIPSW5   LOC=P72;
025- NET DIPSW6   LOC=P71;
026- NET DIPSW7   LOC=P66;
027- NET DIPSW8   LOC=P70;
028- #
029- # PUSHBUTTON SWITCH CONNECTIONS (ACTIVE-LOW)
030- NET PUSH_SPARE_    LOC=P67;
031- NET PUSH_RESET_    LOC=P10;
032- #
033- # LEFT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
034- NET LS_0     LOC=P1;
035- NET LS_1     LOC=P2;
036- NET LS_2     LOC=P3;
037- NET LS_3     LOC=P75;
038- NET LS_4     LOC=P79;
039- NET LS_5     LOC=P82;
040- NET LS_6     LOC=P83;
041- NET LDP_     LOC=P84;
042- #
043- # RIGHT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
044- NET RS_0     LOC=P58;
045- NET RS_1     LOC=P56;
046- NET RS_2     LOC=P54;
047- NET RS_3     LOC=P55;
048- NET RS_4     LOC=P53;
049- NET RS_5     LOC=P57;
050- NET RS_6     LOC=P61;
051- NET RDP_     LOC=P34;
052- #
053- # INDIVIDUAL LED CONNECTIONS (ACTIVE-LOW)
054- NET D_1      LOC=P44;
055- NET D_2      LOC=P43;
056- NET D_3      LOC=P41;
057- NET D_4      LOC=P40;
058- NET D_5      LOC=P39;
059- NET D_6      LOC=P37;
060- NET D_7      LOC=P36;
061- NET D_8      LOC=P35;
```

3.2 Using the VGA Interface

This example creates a circuit that displays the contents of the XS Board RAM on a monitor through the VGA interface of the XStend Board. The ABEL code for this example is shown in **Listing 16**; **Listing 17** and **Listing 18** show the UCF files for using the XS40 and XS95 Boards with the XStend Board, respectively.

The design files for this example should be found in C:\XST-PROJ\EXP2. The steps for compiling and testing the design using an XS40 combined with an XStend Board are as follows:

- Synthesize the ABEL code in the VGACORE.ABL for an XC4005XL FPGA.
- Compile the synthesized netlist using the EXP2_40.UCF constraint file.
- Mount an XS40 Board in the XStend Board and attach the downloading cable from the XS40 to the PC parallel port. Apply 9VDC though jack J9 of the XS40. Place a shunt on jumper J11 to keep the XStend codec disabled. Attach a video monitor to the VGA connector of the XStend Board.
- Download the EXP2.BIT file into the XS40/XStend combination and initialize the RAM with a pattern to display on the monitor with the command: `XSLOAD TESTPATT.HEX EXP2.BIT`.
- Observe the results on the video monitor.

The steps for compiling and testing the design using an XS95 combined with an XStend Board are as follows:

- Synthesize the ABEL code in the EXP2.ABL for an XC95108 CPLD.
- Compile the synthesized netlist using the EXP2_95.UCF constraint file.
- Generate an SVF file for the design.
- Mount an XS95 Board in the XStend Board and attach the downloading cable from the XS95 to the PC parallel port. Apply 9VDC though jack J9 of the XS95. Place a shunt on jumper J11 to keep the XStend codec disabled. Attach a video monitor to the VGA connector of the XStend Board.
- Download the EXP2.SVF file into the XS95/XStend combination and initialize the RAM with a pattern to display on the monitor with the command: `XSLOAD TESTPATT.HEX EXP2.SVF`.
- Observe the results on the video monitor.

Listing 16: ABEL code for testing the XStend VGA.

```
001- MODULE VGACORE
002- TITLE 'VGA signal generator'
003-
004- " This implementation does a 256x240 screen with 4-bit pixels
005-
006- DECLARATIONS
007- "------ external signals -------
008- rst                PIN;              "uC reset control
009- push_reset_        PIN;              "reset
```

```
clk                PIN;               "VGA dot clk
hsync_             PIN ISTYPE 'REG'; "horizontal sync
vsync_             PIN ISTYPE 'REG'; "vertical (frame) sync
red1..red0         PIN ISTYPE 'REG'; "red component
green1..green0     PIN ISTYPE 'REG'; "green component
blue1..blue0 PIN ISTYPE 'REG';       "blue component
a14..a0            PIN;              "address into video RAM
ad7..ad0           PIN;              "data from video RAM
ce_                PIN;              "video RAM chip select
oe_                PIN;              "video RAM output enable
wr_                PIN;              "video RAM write enable
"------ internal storage and signals -------
hcnt8..hcnt0       NODE ISTYPE 'REG'; "horizontal column counter
vcnt9..vcnt0       NODE ISTYPE 'REG'; "vertical line counter
pixrg7..pixrg0     NODE ISTYPE 'REG'; "byte register for four pixels
video_blank        NODE ISTYPE 'COM'; "video blanking signal
delayed_blank      NODE ISTYPE 'REG'; "delayed video blanking signal
"------ synonyms for various items -------
hcnt = [hcnt8..hcnt0];
vcnt = [vcnt9..vcnt0];
vram_addr = [a14..a0];
vram_data = [ad7..ad0];
pixrg = [pixrg7..pixrg0];
pixel = [pixrg3..pixrg0];
rgb = [red1..red0,green1..green0,blue1..blue0];
X = .X.;

EQUATIONS

@CARRY 1;

rst = 1;     " keep uC in reset state so it doesn't interfere

hcnt.ACLR = !push_reset_; "clear counter on active-low reset
vcnt.ACLR = !push_reset_; "clear counter on active-low reset
hcnt.CLK = clk;          "horiz cnt increments on each dot clk
vcnt.CLK = hsync_;       "inc vert line cnt after every horizontal line

hsync_.ASET = !push_reset_;
vsync_.ASET = !push_reset_;
hsync_.CLK = clk;
vsync_.CLK = hsync_;

"column counter rolls-over after 379
WHEN (hcnt<380) THEN hcnt:=hcnt+1 ELSE hcnt:=0;
"horiz sync is low during this interval to signal the start of new line
WHEN ((hcnt>=291)&(hcnt<337)) THEN hsync_:=0 ELSE hsync_:=1;
"horizontal counter rolls-over after 524
WHEN (vcnt<524) THEN vcnt:=vcnt+1 ELSE vcnt:=0;
"vert sync is low during this interval to signal the start of a frame
WHEN ((vcnt>=492)&(vcnt<494)) THEN vsync_:=0 ELSE vsync_:=1;
"blank video outside of visible region: (0,0)->(255,479)
WHEN ((hcnt>=256)#(vcnt>=480)) THEN video_blank=1 ELSE video_blank=0;

"video RAM control signals
ce_ = 0;           "enable the RAM
oe_ = video_blank; "enable the RAM output when video is not blanked
wr_ = 1;           "disable writing to the RAM
"the video RAM address is built from the bits 8-1 of the vert line cnt
"and bits 7-1 of the horiz column cnt.  Each RAM byte contains two
```

```
070- "4-bit pixels so make a new RAM address every two dot clks and ignore
071- "hcnt0 when addressing the RAM.  Also repeat each line of pixels twice
072- "(so the pixels are approximately square on the screen), so ignore vcnt0
073- "when addressing the RAM.
074- vram_addr = [vcnt8..vcnt1,hcnt7..hcnt1];
075-
076- pixrg.ACLR = !push_reset_;     "clear pixel register on reset
077- pixrg.CLK = clk;         "pixel clk controls changes in pixel register
078- "the pixel reg is loaded with from the RAM when the lower bit of the
079- "horizontal counter is zero.  The active pixel is in the lower four
080- "bits of the pixel reg.  On the next clk, the pixel reg is left-shifted
081- "by four bits to bring the other pixel into the active position.
082- WHEN (hcnt0==0)
083-  THEN pixrg := vram_data                 "load 2 pixels from RAM
084-  ELSE pixrg := [0,0,0,0,pixrg7..pixrg4]; "left-shift pixel reg four bits
085-
086- "delay the video blanking by one clk to account for RAM access delay
087- delayed_blank.ACLR = !push_reset_;
088- delayed_blank.CLK = clk;
089- delayed_blank := video_blank;
090-
091- "color mapper that translates each 4-bit pixel into a 6-bit RGB value.
092- "when the video signal is blanked, the RGB value is forced to 0.
093- rgb.ACLR = !push_reset_;
094- rgb.CLK = clk;
095- TRUTH_TABLE ([delayed_blank, pixel] :>    rgb)
096-              [      0     ,^b0000 ] :> ^b000000; "black
097-              [      0     ,^b0001 ] :> ^b101010; "light-gray
098-              [      0     ,^b0010 ] :> ^b111111; "white
099-              [      0     ,^b0011 ] :> ^b110000; "red
100-              [      0     ,^b0100 ] :> ^b001100; "green
101-              [      0     ,^b0101 ] :> ^b000011; "blue
102-              [      0     ,^b0110 ] :> ^b111100; "yellow
103-              [      0     ,^b0111 ] :> ^b110011; "magenta
104-              [      0     ,^b1000 ] :> ^b001111; "cyan
105-              [      0     ,^b1001 ] :> ^b100000; "dark-red
106-              [      0     ,^b1010 ] :> ^b001000; "dark-green
107-              [      0     ,^b1011 ] :> ^b000010; "dark-blue
108-              [      0     ,^b1100 ] :> ^b101000; "tan
109-              [      0     ,^b1101 ] :> ^b100010; "purple
110-              [      0     ,^b1110 ] :> ^b001010; "teal
111-              [      0     ,^b1111 ] :> ^b010101; "dark-gray
112-              [      1     ,X      ] :> ^b000000; "black during blanking
113-
114- END VGACORE
```

Listing 17: UCF file for VGA XStend example with XS40.

```
001- # CLOCK FROM XS40 OSCILLATOR
002- NET CLK      LOC=P13;
003- #
004- # MICROCONTROLLER PINS
005- NET RST      LOC=P36; # ACTIVE-HIGH RESET
006- NET WR_      LOC=P62; # ACTIVE-LOW WRITE (ALSO CONTROLS RAM)
007- NET AD0      LOC=P41; # MULTIPLEXED ADDRESS/DATA BUS
008- NET AD1      LOC=P40;
009- NET AD2      LOC=P39;
010- NET AD3      LOC=P38;
011- NET AD4      LOC=P35;
```

```
012- NET AD5      LOC=P81;
013- NET AD6      LOC=P80;
014- NET AD7      LOC=P10;
015- NET A0       LOC=P3;  # DEMUXED LOWER BYTE OF ADDRESS
016- NET A1       LOC=P4;
017- NET A2       LOC=P5;
018- NET A3       LOC=P78;
019- NET A4       LOC=P79;
020- NET A5       LOC=P82;
021- NET A6       LOC=P83;
022- NET A7       LOC=P84;
023- NET A8       LOC=P59; # UPPER BYTE OF ADDRESS
024- NET A9       LOC=P57;
025- NET A10      LOC=P51;
026- NET A11      LOC=P56;
027- NET A12      LOC=P50;
028- NET A13      LOC=P58;
029- NET A14      LOC=P60;
030- #
031- # XS40 BOARD RAM CONTROL PINS
032- NET OE_      LOC=P61; # ACTIVE-LOW OUTPUT ENABLE
033- NET CE_      LOC=P65; # ACTIVE-LOW CHIP ENABLE
034- #
035- #
036- # XSTEND BOARD CONNECTIONS
037- #
038- # VGA CONNECTIONS
039- NET VSYNC_   LOC=P67;
040- NET HSYNC_   LOC=P19;
041- NET RED1     LOC=P18;
042- NET RED0     LOC=P23;
043- NET GREEN1   LOC=P20;
044- NET GREEN0   LOC=P24;
045- NET BLUE1    LOC=P26;
046- NET BLUE0    LOC=P25;
047- #
048- # PUSHBUTTON SWITCH CONNECTIONS (ACTIVE-LOW)
049- NET PUSH_RESET_    LOC=P37;
```

Listing 18: UCF file for VGA XStend example with XS95.

```
001- # CLOCK FROM XS95 OSCILLATOR
002- NET CLK      LOC=P9;
003- #
004- # MICROCONTROLLER PINS
005- NET RST      LOC=P45; # ACTIVE-HIGH RESET
006- NET WR_      LOC=P63; # ACTIVE-LOW WRITE (ALSO CONTROLS RAM)
007- NET AD0      LOC=P44; # MULTIPLEXED ADDRESS/DATA BUS
008- NET AD1      LOC=P43;
009- NET AD2      LOC=P41;
010- NET AD3      LOC=P40;
011- NET AD4      LOC=P39;
012- NET AD5      LOC=P37;
013- NET AD6      LOC=P36;
014- NET AD7      LOC=P35;
015- NET A0       LOC=P1;  # DEMUXED LOWER BYTE OF ADDRESS
016- NET A1       LOC=P2;
017- NET A2       LOC=P3;
018- NET A3       LOC=P75;
```

```
019- NET A4       LOC=P79;
020- NET A5       LOC=P82;
021- NET A6       LOC=P83;
022- NET A7       LOC=P84;
023- NET A8       LOC=P58; # UPPER BYTE OF ADDRESS
024- NET A9       LOC=P56;
025- NET A10      LOC=P54;
026- NET A11      LOC=P55;
027- NET A12      LOC=P53;
028- NET A13      LOC=P57;
029- NET A14      LOC=P61;
030- #
031- # XS95 BOARD RAM CONTROL PINS
032- NET OE_      LOC=P62; # ACTIVE-LOW OUTPUT ENABLE
033- NET CE_      LOC=P65; # ACTIVE-LOW CHIP ENABLE
034- #
035- #
036- # XSTEND BOARD CONNECTIONS
037- #
038- # VGA CONNECTIONS
039- NET VSYNC_   LOC=P24;
040- NET HSYNC_   LOC=P15;
041- NET RED1     LOC=P14;
042- NET RED0     LOC=P18;
043- NET GREEN1   LOC=P17;
044- NET GREEN0   LOC=P19;
045- NET BLUE1    LOC=P23;
046- NET BLUE0    LOC=P21;
047- #
048- # PUSHBUTTON SWITCH CONNECTIONS (ACTIVE-LOW)
049- NET PUSH_RESET_    LOC=P10;
```

3.3 Using the PS/2 Keyboard Interface

This example creates a circuit that accepts scan codes from a keyboard attached to the PS/2 interface of the XStend Board and displays it on the LEDs. In addition, if a scan code for one of the keys '0'—'9' arrives, then the numeral will be displayed on the right LED display of the XStend Board. The ABEL code for this example is shown in **Listing 19**; **Listing 20** and **Listing 21** show the UCF files for using the XS40 and XS95 Boards with the XStend Board, respectively.

The design files for this example should be found in C:\XST-PROJ\EXP3. The steps for compiling and testing the design using an XS40 combined with an XStend Board are as follows:

- Synthesize the ABEL code in the EXP3.ABL for an XC4005XL FPGA.
- Compile the synthesized netlist using the EXP3_40.UCF constraint file.
- Mount an XS40 Board in the XStend Board and attach the downloading cable from the XS40 to the PC parallel port. Apply 9VDC though jack J9 of the XS40. Place shunts on jumpers J4, J7, and J8 to enable the LEDs. Place a shunt on jumper J11 to keep the XStend codec disabled. Attach a keyboard to the PS/2 connector of the XStend Board.

- Download the EXP3.BIT file into the XS40/XStend combination with the command: `XSLOAD EXP3.BIT`.

- Press keys on the keyboard and observe the results on the LED displays.

The steps for compiling and testing the design using an XS95 combined with an XStend Board are as follows:

- Synthesize the ABEL code in the EXP3.ABL for an XC95108 CPLD.

- Compile the synthesized netlist using the EXP3_95.UCF constraint file.

- Generate an SVF file for the design.

- Mount an XS95 Board in the XStend Board and attach the downloading cable from the XS95 to the PC parallel port. Apply 9VDC though jack J9 of the XS95. Place shunts on jumpers J4, J7, and J8 to enable the LEDs. Place a shunt on jumper J11 to keep the XStend codec disabled. Attach a keyboard to the PS/2 connector of the XStend Board.

- Download the EXP3.SVF file into the XS95/XStend combination with the command: `XSLOAD EXP3.SVF`.

- Press keys on the keyboard and observe the results on the LED displays.

Listing 19: ABEL code for testing the XStend PS/2 example.

```
001- MODULE EXP3
002- TITLE 'EXP3'
003-
004- DECLARATIONS
005-
006- RST PIN;                        // uC reset control
007- OE_ PIN;                        // RAM output enable
008- KB_DATA PIN;                    // serial data from PS/2 keyboard
009- KB_CLK PIN;                     // clock from PS/2 keyboard
010- D_8..D_1 PIN;                   // LED string on XStend Board
011- D_ = [D_8..D_1];
012- RS_6..RS_0 PIN ISTYPE 'COM';   // right LED display on XStend Board
013- RS_ = [RS_6..RS_0];
014- [MASTER9..MASTER0] NODE ISTYPE 'REG';      // master register for
015- MASTER = [MASTER9..MASTER0];               // storing keyboard data
016- [SLAVE9..SLAVE0] NODE ISTYPE 'REG';        // slave register for
017- SLAVE = [SLAVE9..SLAVE0];                  // storing keyboard data
018-
019- EQUATIONS
020-
021- RST = 1;     // disable uC
022- OE_ = 1;     // disable RAM outputs
023-
024- // Scan codes from the keyboard arrive LSB-first on the falling
025- // edge of the clock.  Each bit is shifted into the MSB of the
026- // master register.  The lower bits of the master are filled with
027- // slave contents shifted right by one bit.  Then the updated
```

```
028- // contents of the master are loaded into the slave on the rising
029- // edge of the clock.
030- MASTER.CLK = !KB_CLK;
031- MASTER := [KB_DATA,SLAVE9..SLAVE1];
032- SLAVE.CLK  =  KB_CLK;
033- SLAVE  := MASTER;
034-
035- // Show the bit string for the scan code.  Ignore the upper two
036- // bits because these are the parity bit and the stop bit.
037- D_ = ![SLAVE7..SLAVE0];
038-
039- // If one of the '0'-'9' keys is pressed, display the digit on the
040- // right LED display of the XStend Board.
041- TRUTH_TABLE ([SLAVE7..SLAVE0]  -> [RS_6..RS_0])
042-                     ^H16              ->    ^B1101101;
043-                     ^H1E              ->    ^B0100010;
044-                     ^H26              ->    ^B0100100;
045-                     ^H25              ->    ^B1000101;
046-                     ^H2E              ->    ^B0010100;
047-                     ^H36              ->    ^B0010000;
048-                     ^H3D              ->    ^B0101101;
049-                     ^H3E              ->    ^B0000000;
050-                     ^H46              ->    ^B0000100;
051-                     ^H45              ->    ^B0001000;
052-
053- END EXP3
```

Listing 20: UCF file for PS/2 XStend example with XS40.

```
001- # MICROCONTROLLER PINS
002- NET RST      LOC=P36; # ACTIVE-HIGH RESET
003- #
004- # XS40 BOARD RAM CONTROL PINS
005- NET OE_      LOC=P61; # ACTIVE-LOW OUTPUT ENABLE
006- #
007- #
008- # XSTEND BOARD CONNECTIONS
009- #
010- # PS/2 KEYBOARD CONNECTIONS
011- NET KB_DATA  LOC=P69;
012- NET KB_CLK   LOC=P68;
013- #
014- # RIGHT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
015- NET RS_0     LOC=P59;
016- NET RS_1     LOC=P57;
017- NET RS_2     LOC=P51;
018- NET RS_3     LOC=P56;
019- NET RS_4     LOC=P50;
020- NET RS_5     LOC=P58;
021- NET RS_6     LOC=P60;
022- #
023- # INDIVIDUAL LED CONNECTIONS (ACTIVE-LOW)
024- NET D_1      LOC=P41;
025- NET D_2      LOC=P40;
026- NET D_3      LOC=P39;
027- NET D_4      LOC=P38;
028- NET D_5      LOC=P35;
029- NET D_6      LOC=P81;
```

```
030- NET D_7      LOC=P80;
031- NET D_8      LOC=P10;
```

Listing 21: UCF file for PS/2 XStend example with XS95.

```
001- # MICROCONTROLLER PINS
002- NET RST      LOC=P45; # ACTIVE-HIGH RESET
003- #
004- # XS95 BOARD RAM CONTROL PINS
005- NET OE_      LOC=P62; # ACTIVE-LOW OUTPUT ENABLE
006- #
007- #
008- # XSTEND BOARD CONNECTIONS
009- #
010- # PS/2 KEYBOARD CONNECTIONS
011- NET KB_DATA  LOC=P70;
012- NET KB_CLK   LOC=P26;
013- #
014- # RIGHT LED DIGIT SEGMENT CONNECTIONS (ACTIVE-LOW)
015- NET RS_0     LOC=P58;
016- NET RS_1     LOC=P56;
017- NET RS_2     LOC=P54;
018- NET RS_3     LOC=P55;
019- NET RS_4     LOC=P53;
020- NET RS_5     LOC=P57;
021- NET RS_6     LOC=P61;
022- #
023- # INDIVIDUAL LED CONNECTIONS (ACTIVE-LOW)
024- NET D_1      LOC=P44;
025- NET D_2      LOC=P43;
026- NET D_3      LOC=P41;
027- NET D_4      LOC=P40;
028- NET D_5      LOC=P39;
029- NET D_6      LOC=P37;
030- NET D_7      LOC=P36;
031- NET D_8      LOC=P35;
```

3.4 Using the RAMs

3.5 Using the Stereo Codec

4 Connections between XStend/XS Board resources

Table 4 and **Table 5** show the connections between the XS40 Board and XS95 Board and the resources on the XStend Board, respectively. Both tables contain the same information but are sorted according to the ordering of the XS40 or XS95 Board pins.

Table 4: Connections between the XS40 Board and XStend Board resources.

XS40 Bus (J1,J3,J18)	XS95 Pins (J2)	Power / GND	DIP Switch	Push-buttons	LEDs	VGA Interface	PS/2 Interface	RAMs	Stereo Codec	8051Uc	PC Parallel Port	Oscillator	Function	UW-FPGA BOARD Pin
1	77												Uncommitted XS95 I/O pin	
2	78	+5V											+5V power source	
3	1				LS_0			A0					LED segment; address line	P35
4	2				LS_1			A1					LED segment; address line	P36
5	3				LS_2			A2					LED segment; address line	P29
6	5		DIPSW4						LRCK	P13			DIP switch; codec control; uC I/O	P24
7	6		DIPSW1					LCE_		P10			DIP switch; RAM chip-enable, uC I/O port	P19
8	7		DIPSW2					RCE_		P11			DIP switch; RAM chip-enable, uC I/O port	P20
9	11		DIPSW3						MCLK	P12			DIP switch; codec clock; uC I/O port	P23
10	35				D_8			AD7		P07			LED; RAM data line; uC muxed address/data line	P61
11	31												Uncommitted XS95 I/O pin	
12	69												Uncommitted XS95 I/O pin	
13	9											CLK	XS40/95 oscillator output	
14	13									PSEN_			uC program storage-enable	
15	28												JTAG TDI; DIN	
16	30												JTAG TCK; CCLK	
17	29												JTAG TMS	
18	14				S5	RED1							XS Board LED segment; VGA color signal	
19	15				S6	HSYNC_							XS Board LED segment; VGA horiz. sync.	
20	17				S3	GREEN1							XS Board LED segment; VGA color signal	
21	68												Uncommitted XS95 I/O pin	
22	33												Uncommitted XS95 I/O pin	
23	18				S4	RED0							XS Board LED segment; VGA color signal	
24	19				S2	GREEN0							XS Board LED segment; VGA color signal	
25	21				S0	BLUE0							XS Board LED segment; VGA color signal	
26	23				S1	BLUE1							XS Board LED segment; VGA color signal	
27	32									RD_			uC read line	
28	34				RDP_			A15		P27			LED decimal-point; address line; uC I/O port	P41
29	20									ALE_			uC address-latch-enable	
30													Serial EEPROM chip-enable	
31	12												Uncommitted XS95 I/O pin	
32	81										PC_D6		PC parallel port output	
33	25												Uncommitted XS95 I/O pin	
34	80										PC_D7		PC parallel port output	
35	39				D_5			AD4		P04			LED; RAM data line; uC muxed address/data line	P66
36	45									RST			uC reset line	
37	10			RESET_						XTAL1			Pushbutton; uC clock line	P56
38	40				D_4			AD3		P03			LED; RAM data line; uC muxed address/data line	P57
39	41				D_3			AD2		P02			LED; RAM data line; uC muxed address/data line	P58
40	43				D_2			AD1		P01			LED; RAM data line; uC muxed address/data line	P59
41	44				D_1			AD0		P00			LED; RAM data line; uC muxed address/data line	P60
42	4												Uncommitted XS95 I/O pin	
43													Unconnected	
44	46								CCLK		PC_D0		Codec control line; PC parallel port output	
45	47								CDIN		PC_D1		Codec control line; PC parallel port output	
46	48								CS_		PC_D2		Codec control line; PC parallel port output	
47	50										PC_D3		PC parallel port output	
48	51										PC_D4		PC parallel port output	
49	52										PC_D5		PC parallel port output	
50	53				RS_4			A12		P24			LED segment; address line; uC I/O port	P48
51	54				RS_2			A10		P22			LED segment; address line; uC I/O port	P45
52	49	GND											Power supply ground	
53													Unconnected	
54		+3.3V											+3.3V/+5V power supply (4000E/4000XL)	
55				PROGRAM									XS40 configuration control	P55
56	55				RS_3			A11		P23			LED segment; address line; uC I/O port	P51
57	56				RS_1			A9		P21			LED segment; address line; uC I/O port	P47
58	57				RS_5			A13		P25			LED segment; address line; uC I/O port	P50
59	58				RS_0			A8		P20			LED segment; address line; uC I/O port	P46
60	61				RS_6			A14		P26			LED segment; address line; uC I/O port	P49
61	62							OE_					RAM output-enable	
62	63							WR_		P36			RAM write-enable; uC I/O port	
63													Unconnected	
64													Unconnected	
65	65							CE_					XS Board RAM chip-enable	
66	66		DIPSW7						SDOUT	P16			DIP switch; codec output data; uC I/O port	P27
67	24,67			SPARE_		VSYNC_				P17			Pushbutton; VGA horiz. sync.; uC I/O port	P18
68	26						KB_CLK			P34			PS/2 keyboard clock; uC I/O port	
69	70		DIPSW8				KB_DATA			P31			DIP switch; keyboard serial data; uC I/O port	P28
70	71		DIPSW6						SDIN	P15			DIP switch; codec input data; uC I/O port	P26
71	28												JTAG TDI; DIN	
72	59												JTAG TDO; DOUT	
73	30												JTAG TCK; CCLK	
74	74												Uncommitted XS95 I/O pin	
75	59												JTAG TDO; DOUT	
76	76												Uncommitted XS95 I/O pin	
77	72		DIPSW5						SCLK	P14			DIP switch; codec serial clock; uC I/O port	P25
78	75				LS_3			A3					LED segment; address line	P44
79	79				LS_4			A4					LED segment; address line	P38
80	36				D_7			AD6		P06			LED; RAM data line; uC muxed address/data line	P62
81	37				D_6			AD5		P05			LED; RAM data line; uC muxed address/data line	P65
82	82				LS_5			A5					LED segment; address line	P40
83	83				LS_6			A6					LED segment; address line	P39
84	84				LDP_			A7					LED decimal-point; address line	P37

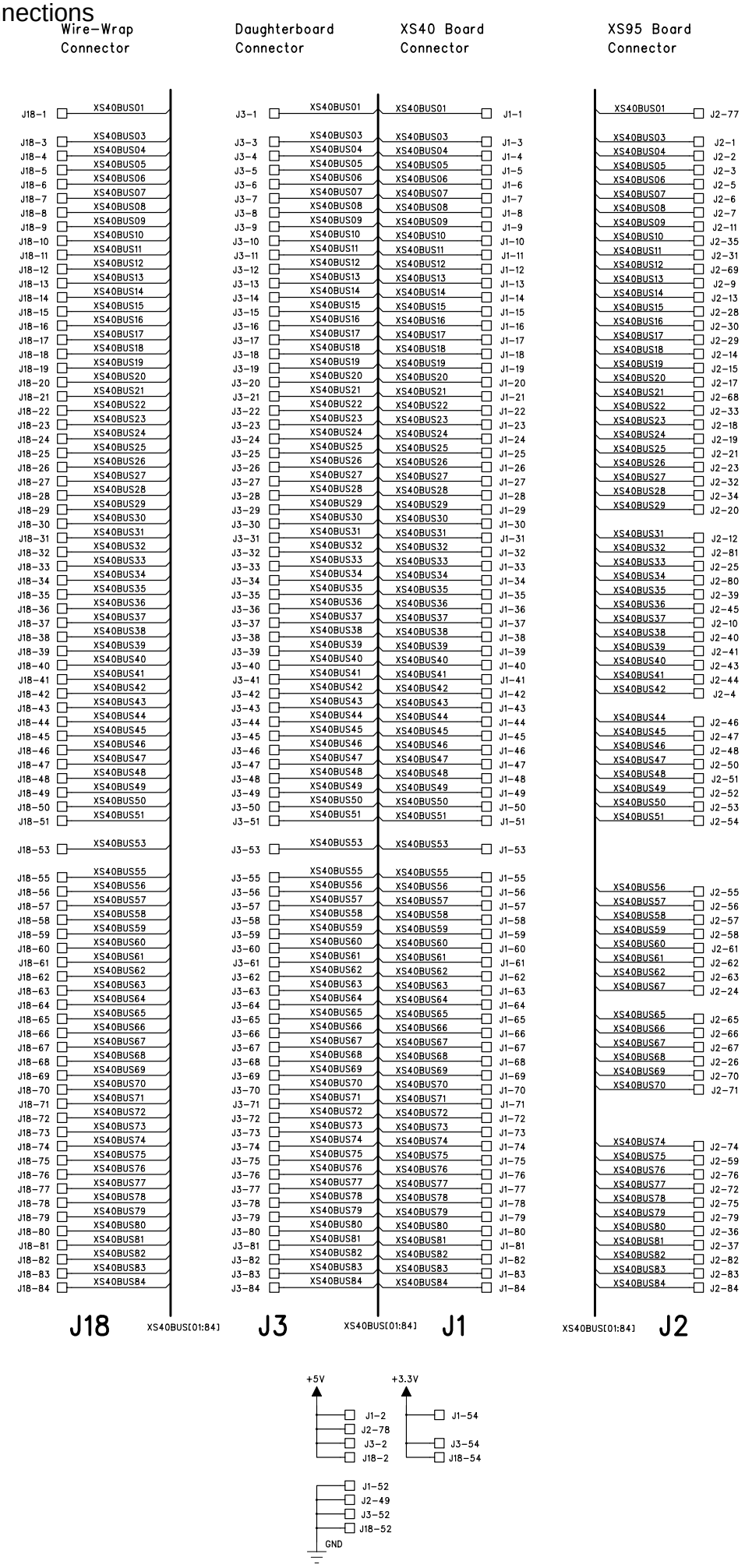
Table 5: Connections between the XS95 Board and XStend Board resources.

XS95 Pins (J2)	XS40 Bus (J1,J3,J18)	Power/ GND	DIP Switch	Push-buttons	LEDs	VGA Interface	PS/2 Interface	RAMs	Stereo Codec	8051 Uc	PC Parallel Port	Oscillator	Function	UW-FPGA BOARD Pin
1	3				LS_0			A0					LED segment; address line	P35
2	4				LS_1			A1					LED segment; address line	P36
3	5				LS_2			A2					LED segment; address line	P29
4	42												Uncommitted XS95 I/O pin	
5	6		DIPSW4						LRCK	P13			DIP switch; codec control; uC I/O	P24
6	7		DIPSW1					LCE_		P10			DIP switch; RAM chip-enable, uC I/O port	P19
7	8		DIPSW2					RCE_		P11			DIP switch; RAM chip-enable, uC I/O port	P20
9	13											CLK	XS40/95 oscillator output	
10	37			RESET_						XTAL1			Pushbutton; uC clock line	P56
11	9		DIPSW3						MCLK	P12			DIP switch; codec clock; uC I/O port	P23
12	31												Uncommitted XS95 I/O pin	
13	14									PSEN_			uC program storage-enable	
14	18				S5	RED1							XS Board LED segment; VGA color signal	
15	19				S6	HSYNC_							XS Board LED segment; VGA horiz. sync.	
17	20				S3	GREEN1							XS Board LED segment; VGA color signal	
18	23				S4	RED0							XS Board LED segment; VGA color signal	
19	24				S2	GREEN0							XS Board LED segment; VGA color signal	
20	29									ALE_			uC address-latch-enable	
21	25				S0	BLUE0							XS Board LED segment; VGA color signal	
23	26				S1	BLUE1							XS Board LED segment; VGA color signal	
25	33												Uncommitted XS95 I/O pin	
26	68						KB_CLK			P34			PS/2 keyboard clock; uC I/O port	
28	15												JTAG TDI; DIN	
28	71												JTAG TDI; DIN	
29	17												JTAG TMS	
30	16												JTAG TCK; CCLK	
30	73												JTAG TCK; CCLK	
31	11												Uncommitted XS95 I/O pin	
32	27									RD_			uC read line	
33	22												Uncommitted XS95 I/O pin	
34	28				RDP_			A15		P27			LED decimal-point; address line; uC I/O port	P41
35	10				D_8			AD7		P07			LED; RAM data line; uC muxed address/data line	P61
36	80				D_7			AD6		P06			LED; RAM data line; uC muxed address/data line	P62
37	81				D_6			AD5		P05			LED; RAM data line; uC muxed address/data line	P65
39	35				D_5			AD4		P04			LED; RAM data line; uC muxed address/data line	P66
40	38				D_4			AD3		P03			LED; RAM data line; uC muxed address/data line	P57
41	39				D_3			AD2		P02			LED; RAM data line; uC muxed address/data line	P58
43	40				D_2			AD1		P01			LED; RAM data line; uC muxed address/data line	P59
44	41				D_1			AD0		P00			LED; RAM data line; uC muxed address/data line	P60
45	36									RST			uC reset line	
46	44								CCLK		PC_D0		Codec control line; PC parallel port output	
47	45								CDIN		PC_D1		Codec control line; PC parallel port output	
48	46								CS_		PC_D2		Codec control line; PC parallel port output	
49	52	GND											Power supply ground	
50	47										PC_D3		PC parallel port output	
51	48										PC_D4		PC parallel port output	
52	49										PC_D5		PC parallel port output	
53	50				RS_4			A12		P24			LED segment; address line; uC I/O port	P48
54	51				RS_2			A10		P22			LED segment; address line; uC I/O port	P45
55	56				RS_3			A11		P23			LED segment; address line; uC I/O port	P51
56	57				RS_1			A9		P21			LED segment; address line; uC I/O port	P47
57	58				RS_5			A13		P25			LED segment; address line; uC I/O port	P50
58	59				RS_0			A8		P20			LED segment; address line; uC I/O port	P46
59	72												JTAG TDO; DOUT	
59	75												JTAG TDO; DOUT	
61	60				RS_6			A14		P26			LED segment; address line; uC I/O port	P49
62	61							OE_					RAM output-enable	
63	62							WR_		P36			RAM write-enable; uC I/O port	
65	65							CE_					XS Board RAM chip-enable	
66	66		DIPSW7						SDOUT	P16			DIP switch; codec output data; uC I/O port	P27
68	21												Uncommitted XS95 I/O pin	
69	12												Uncommitted XS95 I/O pin	
70	69		DIPSW8				KB_DATA			P31			DIP switch; keyboard serial data; uC I/O port	P28
71	70		DIPSW6						SDIN	P15			DIP switch; codec input data; uC I/O port	P26
72	77		DIPSW5						SCLK	P14			DIP switch; codec serial clock; uC I/O port	P25
74	74												Uncommitted XS95 I/O pin	
75	78				LS_3			A3					LED segment; address line	P44
76	76												Uncommitted XS95 I/O pin	
77	1												Uncommitted XS95 I/O pin	
78	2	+5V											+5V power source	
79	79				LS_4			A4					LED segment; address line	P38
80	34										PC_D7		PC parallel port output	
81	32										PC_D6		PC parallel port output	
82	82				LS_5			A5					LED segment; address line	P40
83	83				LS_6			A6					LED segment; address line	P39
84	84				LDP_			A7					LED decimal-point; address line	P37
24,67	67			SPARE_		VSYNC_				P17			Pushbutton; VGA horiz. sync.; uC I/O port	P18
	30												Serial EEPROM chip-enable	
	43												Unconnected	
	53												Unconnected	
	54	+3.3V											+3.3V/+5V power supply (4000E/4000XL)	
	55			PROGRAM									XS40 configuration control	P55
	63												Unconnected	
	64												Unconnected	

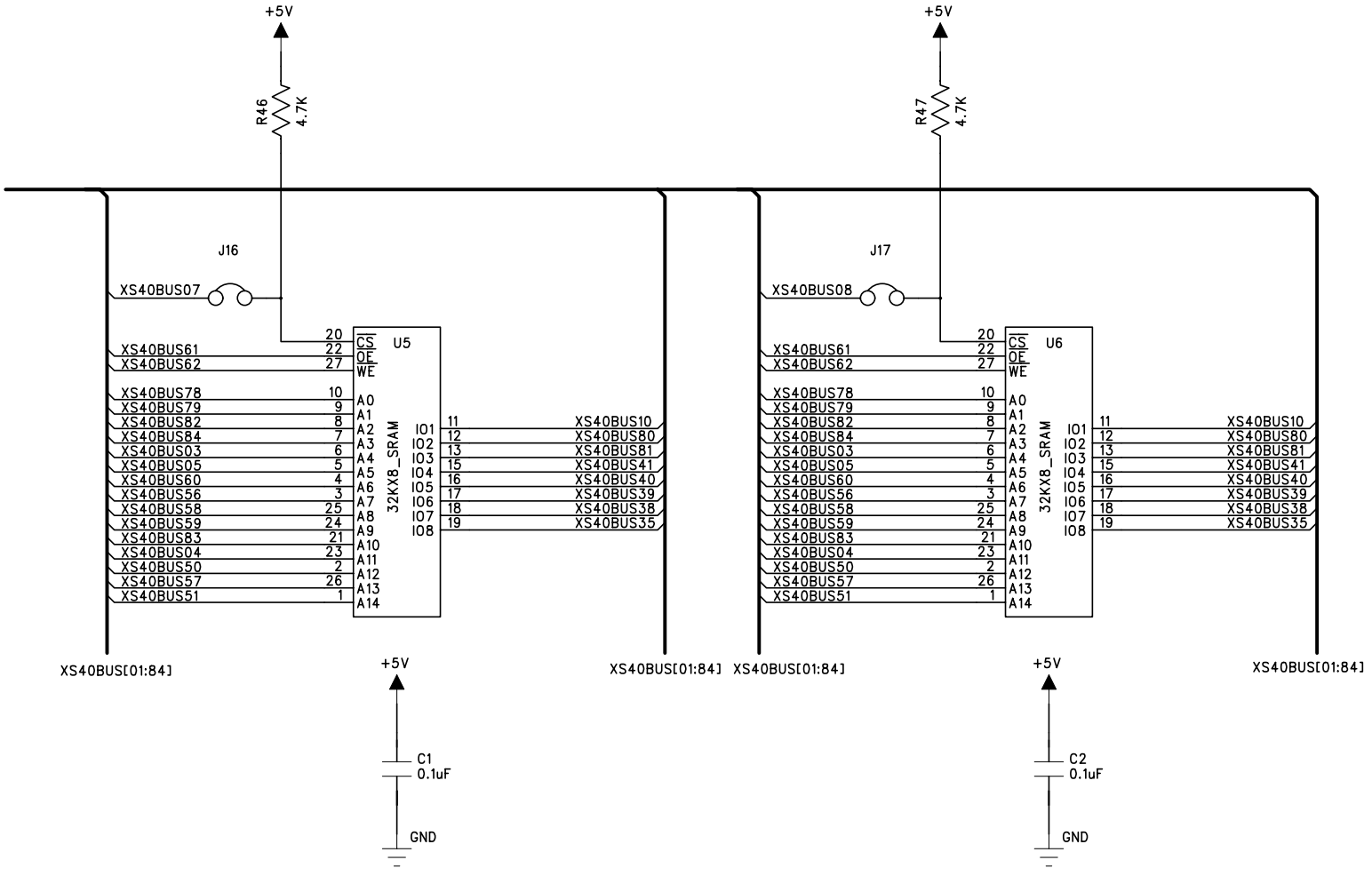
Table 6: Jumpers for the XStend Board.

Jumper	Setting
J8	Removing the shunt on this jumper disconnects the power from LEDs D1—D8.
J4	Removing the shunt on this jumper disconnects the power to the left LED display U1.
J7	Removing the shunt on this jumper disconnects the power to the right LED display U2.
J16	Removing the shunt on this jumper disables the left RAM U5 by pulling its chip-select pin high.
J17	Removing the shunt on this jumper disables the right RAM U6 by pulling its chip-select pin high.
J11	Placing a shunt on this jumper disables the codec by holding it in the reset state.
J13, J15	Placing shunts across two of the three pins of these jumpers selects the digital de-emphasis for different sampling rates:
0 0	De-emphasis for 32 KHz
0 1	De-emphasis for 44.1 KHz
1 0	De-emphasis for 48 KHz
1 1	De-emphasis off

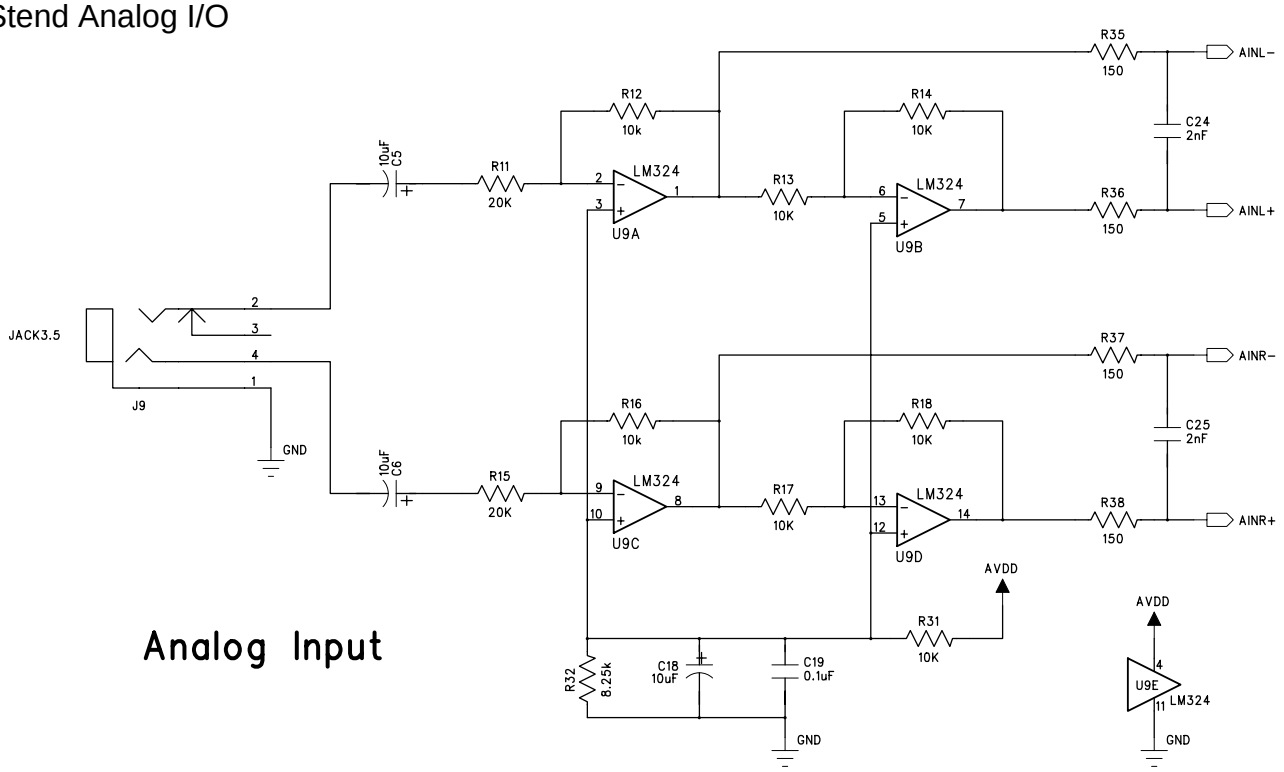
XStend Bus Connections



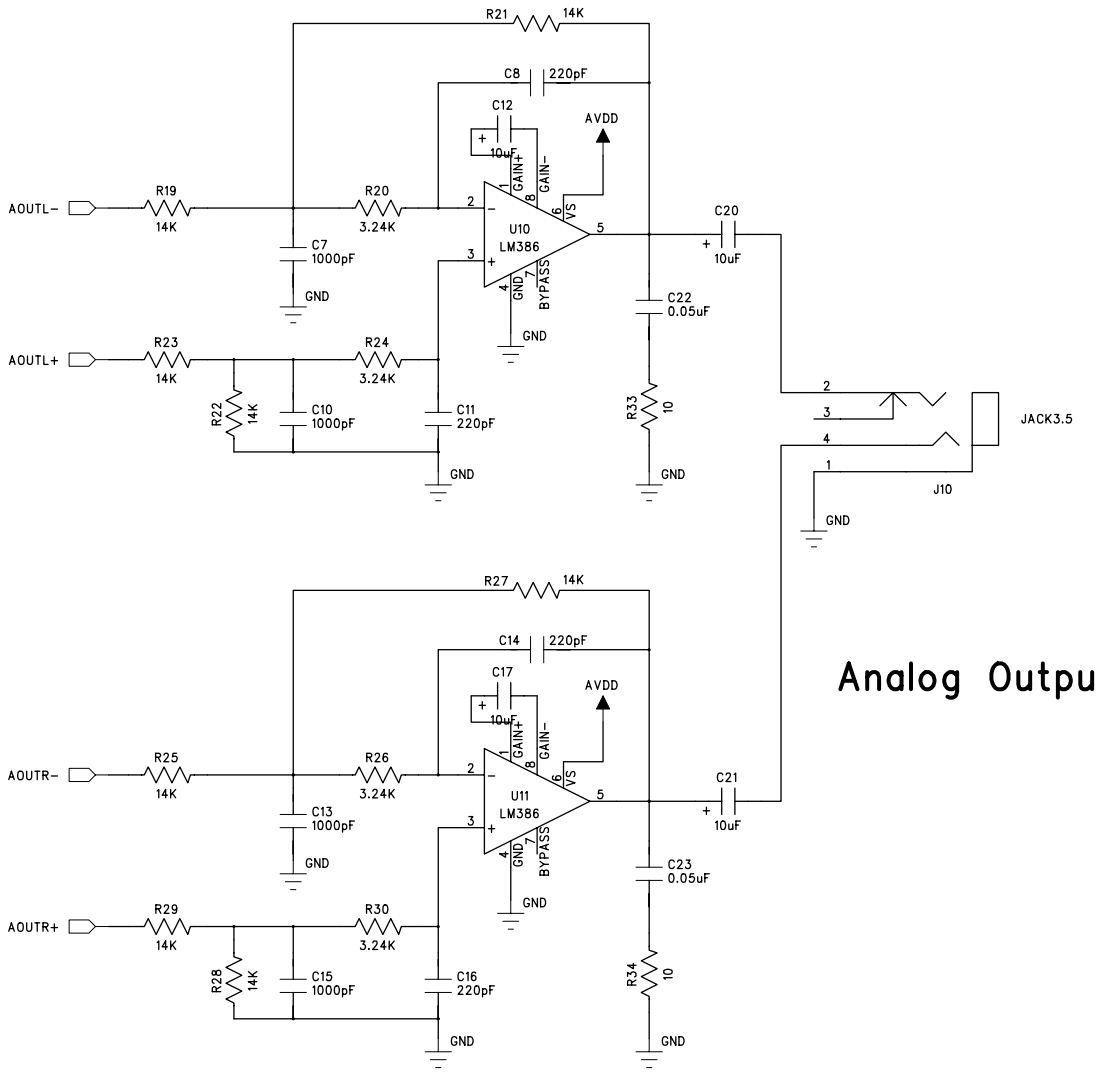
XStend RAMs


+5V
R46
4.7K
+5V
R47
4.7K
J16
XS40BUS07
J17
XS40BUS08
U5
U6
32KX8_SRAM
CS
OE
WE
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
XS40BUS61
XS40BUS62
XS40BUS78
XS40BUS79
XS40BUS82
XS40BUS84
XS40BUS03
XS40BUS05
XS40BUS60
XS40BUS56
XS40BUS58
XS40BUS59
XS40BUS83
XS40BUS04
XS40BUS50
XS40BUS57
XS40BUS51
XS40BUS10
XS40BUS80
XS40BUS81
XS40BUS41
XS40BUS40
XS40BUS39
XS40BUS38
XS40BUS35
XS40BUS[01:84]
XS40BUS[01:84]
XS40BUS[01:84]
XS40BUS[01:84]
+5V
C1
0.1uF
GND
+5V
C2
0.1uF
GND

XStend Analog I/O

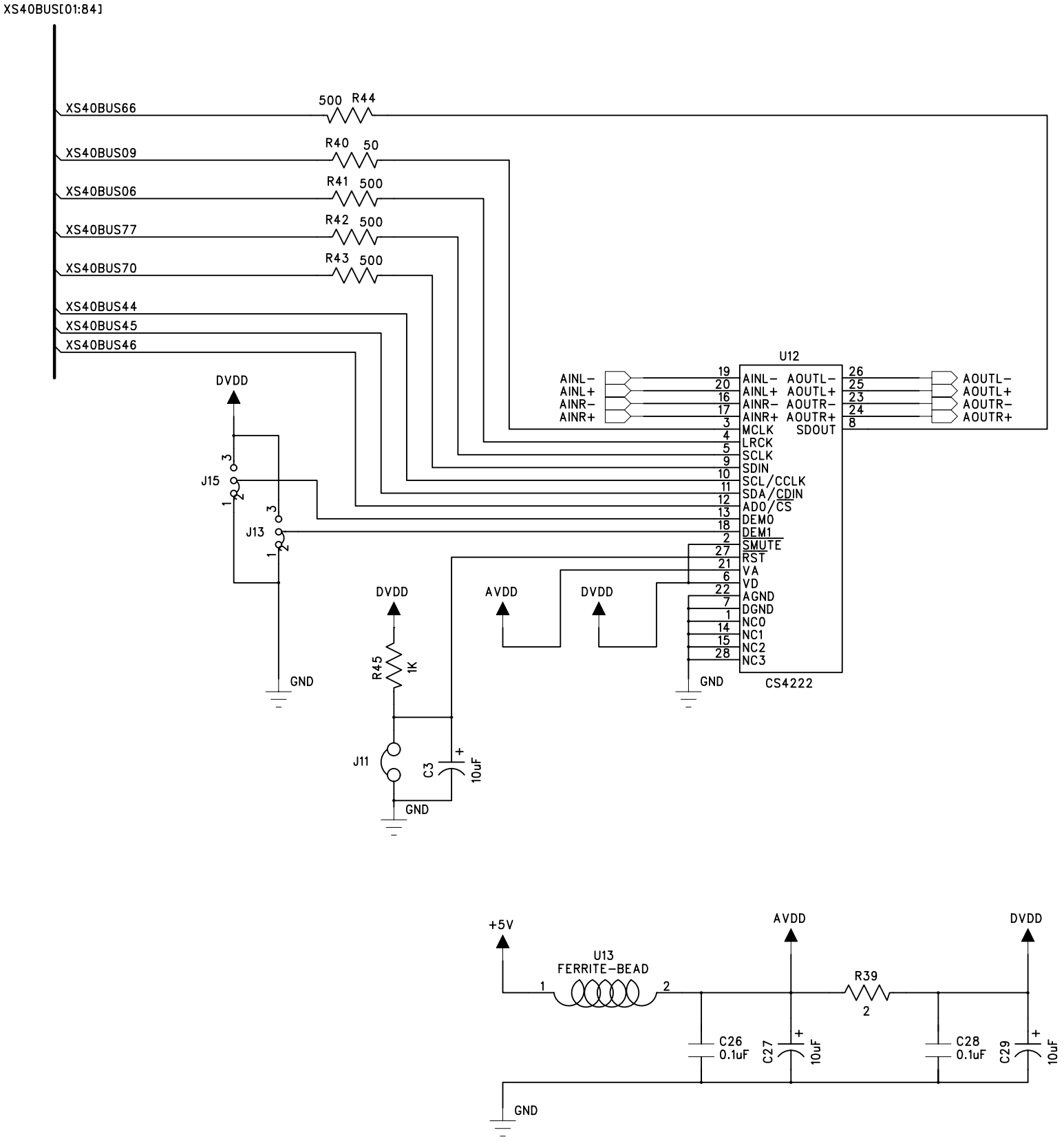


Analog Input

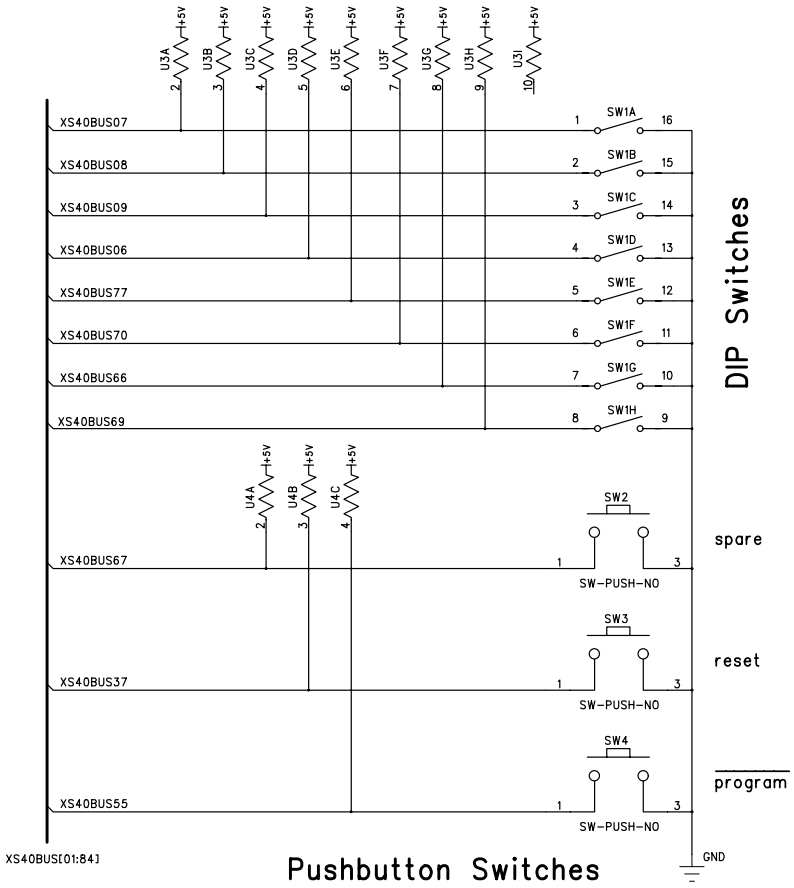


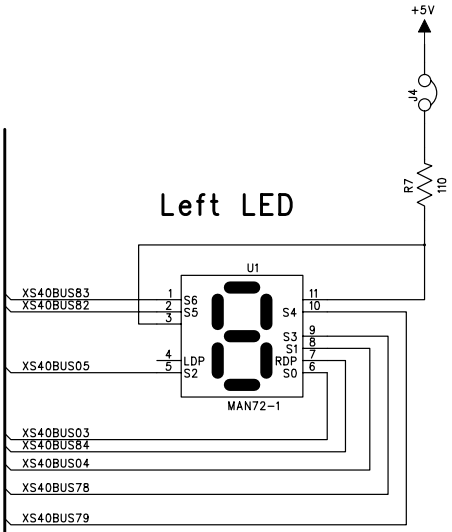
Analog Output

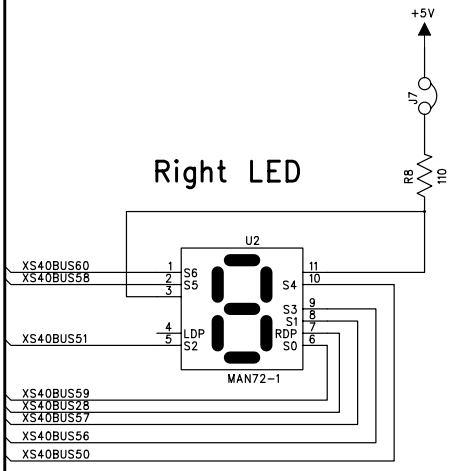
XStend Stereo Codec

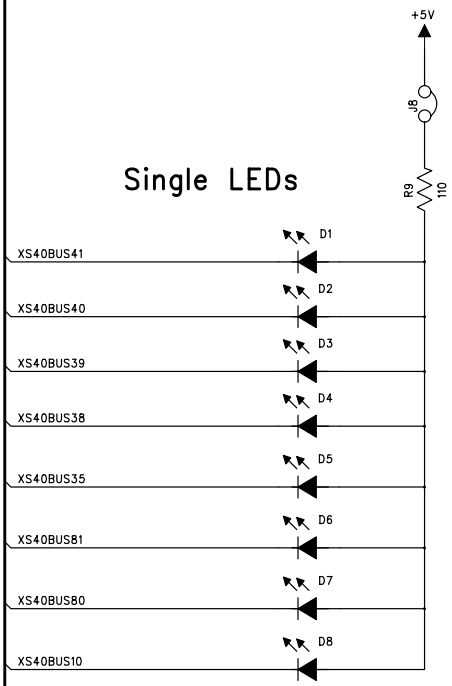


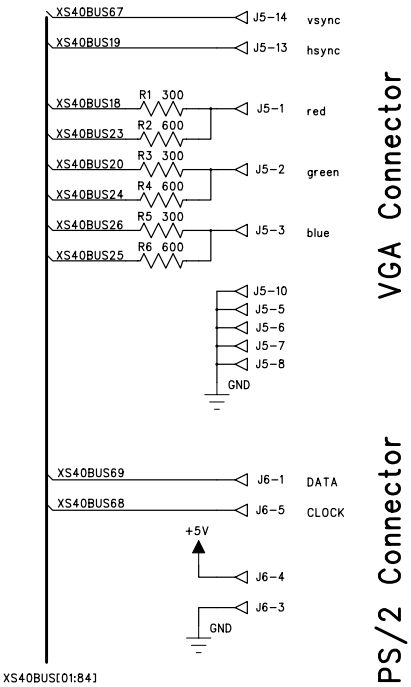
XStend Switches, LEDs, VGA Interface, and PS/2 Interface









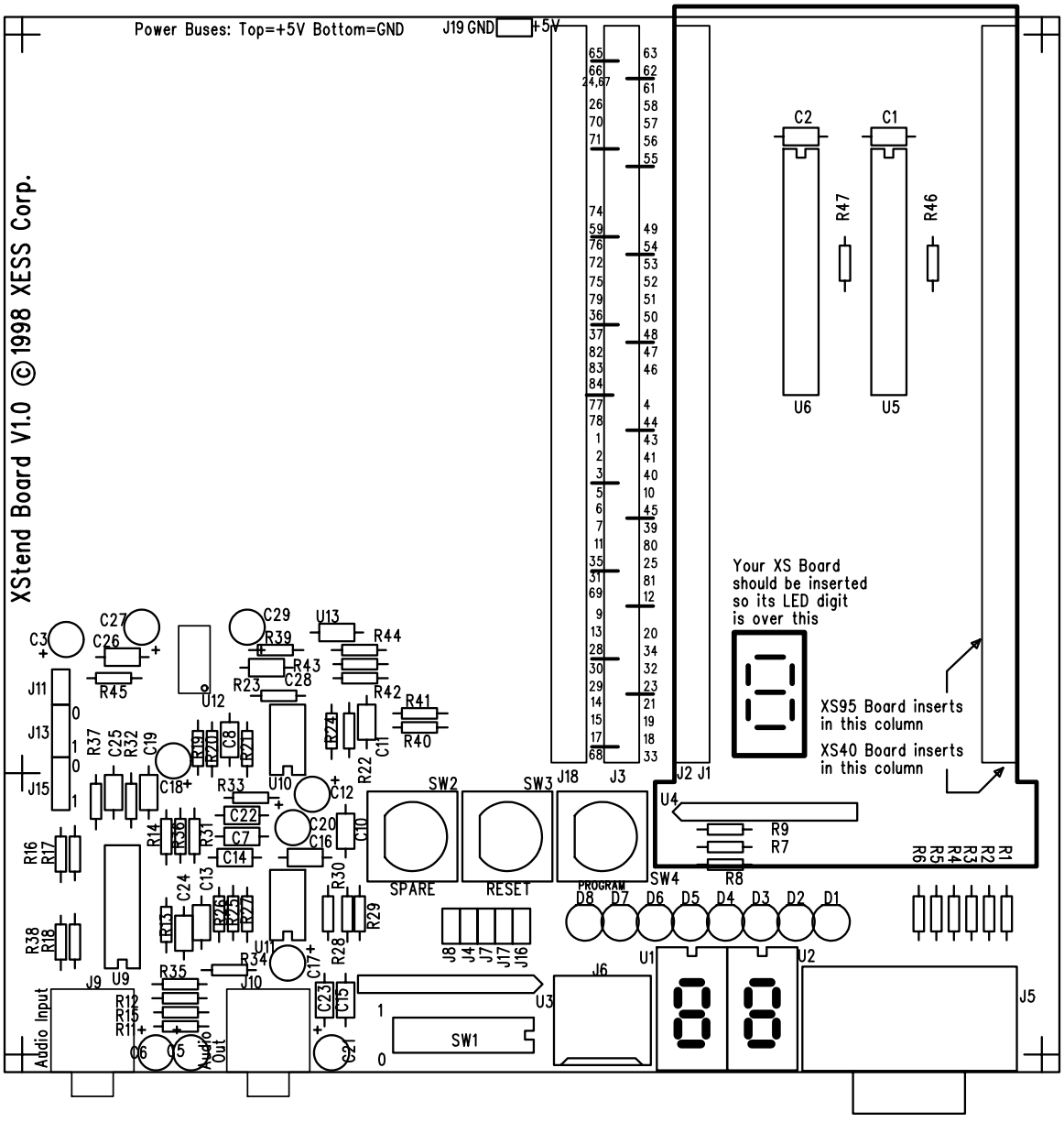


XStend Board Layout

H1

H5 H3
H2

H4



Silkscreen