UNIVERSITY OF CALIFORNIA, RIVERSIDE Department of Electrical Engineering Department of Computer Science Written by Hyung-Jun Kim in 1999 Modified by Enoch Hwang Used and distributed with the author's permission

EE/CS 120A : Logic Design Laboratory

Instructor:	Dr. Enoch Hwang
Office:	A303 Bourns Hall
Email:	ehwang@cs.ucr.edu
Homepage:	www.cs.ucr.edu/~ehwang

Lab Overview

It involves design, assembly and test of combinational and sequential logic circuits. Logic designs will be done using simple CAD tools and implemented using field-programmable gate arrays (FPGA). In the laboratory digital logic circuits will be designed and implemented using the Foundation Series Tools and FPGAs from Xilinx, Inc.

Grading Policy

Grading will be based on laboratory reports and in-lab performance (i.e., quizzes and laboratoryrelated questions). TAs and/or the instructor will ask pertinent questions to individual members of a group at random. The tentative grading policy is:

- Pre-lab Work: 20%
- In-lab Performance: 40%
- Laboratory Report: 40%

Lab Instructions

- Seven lab experiments and Projects are to be conducted, in addition to the lab orientation. Groups of 2 students will be formed. The lab report, <u>written individually</u>, is due one or two weeks after the day of the experiment at the *beginning* of the next lab. Every person must submit their own lab report for each lab *before* each session.
- You are excepted to read the lab description, and complete the pre-lab *before* each session. Read each week's assignment and complete any pre-lab calculations and/or design work *before* coming to the lab. Try to understand the objectives of the experiment and determine what information you should record in your laboratory notebook. If you have any questions, please see the instructor or your TA before the scheduled time of your lab.
- Be prepared to show your pre-lab work to your TA. If you have VHDL code, schematic and wiring diagrams ready before you come to lab, you will be able to finish quite easily during the lab time. If you are not prepared, you may not be able to finish during the lab period and consequently, your grade will suffer. Each lab session lasts three hours. TA or instructor is not required to give you extra time to complete your work.
- Reports that appear to be copied will be returned ungraded. <u>No late report will be accepted</u>.

Lab Reports

Please use the following format for your lab reports:

- *Cover:* Includes course number, topic of the experiment, section number, names of members, and your name and student id number.
- Abstract: Brief introduction to the experiment and a summary of results.
- *Analysis:* Design techniques, implementation details, e.g., K-maps, boolean equations, schematics, etc.
- *Records:* Simulation results, VHDL input and output, FPGA pin-maps, interconnections, etc.
- *Discussion:* Comments, circuit debugging and problem solving issues, and conclusions.

EE/CS 120A : Logic Design Lab 1: Up/Down Counter with 7-segment display

Design Problems

In this lab, you will construct an Up/Down counter with 7-segment display as shown in Figure 1.

The MC14029B Binary/Decade up/down counter consists of D-type flip-flops with a gating structure to provide toggle flip-flop capability. The counter can be used in either Binary or BCD operation. It also can be used either as a Down-counter (when you connect pin 10 to ground) or as an Up-counter (when you connect pin 10 to 5V) as shown in Figure 1.

The MC14511B BCD-to-seven segment decoder also has an output drive capability.

Figures 2 and 3 show the detail logic diagrams of the MC14029B (Binary/Decade Up/Down counter) and MC1451B (BCD-to-Seven segment Latch/Decoder/Driver), respectively. You may need the circuit for the final project.





Figure 2: Logic diagram of MC14029B(Binary/Decade Up/Down counter)

LOGIC DIAGRAM



Figure 3: Logic diagram of MC14511B(BCD-to-Seven segment Latch/Decoder/Driver)

EE/CS 120A : Logic Design Lab 2: Combinational Logic Design Using Xilinx Foundation Tools

Objectives

The objective of this laboratory assignment is:

- To get familiar with the Xilinx Foundation Series Tools to design logic circuits.
- To design and implement simple combinational logic circuits using Schematic editor and simulator.

Laboratory Instructions

- Create a directory with your name on the C drive of your lab PC. Use this directory to create your project, store your results, bitstreams, etc. during the lab session.
- You can bring complete project files on a floppy disk and then use the **Copy Project** command from the Project Manager menu to copy it into the directory you created above.
- Alternatively, you can create a new project in your directory on the C drive and then copy your files to that new project directory. Remember to Add your .SCH file to the project.
- Perform functional simulation of your design and have it checked by the lab instructor or your TA.
- In case you modify your source file, remember to copy it back to your floppy disk.
- Test and demonstrate your circuit to the lab instructor or your TA.
- Before you leave the lab please **remove** the files and directories that you created on your lab PC and leave your workplace clean and tidy.

EE/CS 120: Logic Design Xilinx Schematic Simulation Procedures

- 1. Double-click on the Xilinx Foundation Project Manager icon.
- You will get the Getting Started window. Click on Create a New Project and then click OK. You will get the New Project window. Enter your project name in the box Name: and click on Schematic, and then choose XC9500 (lower left device box). Click OK after this.
- 3. You will see a screen that shows you the directory and some default files that have been created. There are three main panes in the Project Manager window. On the upper right-hand pane, you can find a box **Design Entry**. Click on the rightmost icon which has the shape of an AND gate (icon for the schematic editor). You now see the schematic editor in a new window. This is where you will draw your schematic.
- 4. The first thing you need to do is add the gates for your project. On the left side of the screen you will see an icon in the form of an AND gate called a symbol toolbox. Click on this and the SC Symbols window will appear with a list of all types of components you can use. You will get a list of components in the form of a drop down menu or alternately you can go to Mode → Symbols menu item on the options bar on the top and click on symbols to get the same menu. You now have to only type the name of the component that you need. For a 2-input AND gate case, an AND gate at the bottom of the menu or scroll down through the list in the components window till you see an AND2 (2-input AND gate) and select it from the list. You then simply move the mouse and position it to wherever you require it to be placed. To position the component simply left click on the mouse. You can attach another copy of the AND gate to your cursor by simply clicking on the AND gate you just dropped. Then you can drop the new AND gate in the schematic as well. In a similar manner you can get all the components that you require for your circuit. You are now ready to connect the gates for the circuit.
- 5. You now need to add input and output buffers to the circuit. This is commonly referred to as adding IBUFs and OBUFs (short forms for Input Buffer and Output Buffer, respectively). These buffers indicate that the signals attached to them will actually enter and exit the FPLD chip via the I/O pins. Select IBUF for input and OBUF for output from the symbol table. Add the buffer and give it the same name as the input (or output) signal. When this is done we add Terminals. To do this click on the terminal button on the upper left corner of the toolbar in the SC symbols window in which you can type the Terminal Name and Terminal Type of each input and output, e.g. terminal name is A and type is INPUT (you can select input or output from the drop down box). Click OK and an input (or output) terminal will be attached to your cursor. Simply click on the mouse to drop the terminal into your schematic. Then add all other inputs and outputs in the same manner. The terminal names must be unique. Make sure that IBUF's are between the terminals and the logic gates.
- 6. At this point we have all the components we need so you can double-click on the upper left corner of the **SC Symbols** window to get rid of it.

- 7. You might at some point want to rotate the component. Go into Mode → Select and Drag menu item. Select the component you want to rotate with the mouse. Right click on the mouse and select Symbol Properties from the pop up menu, which appears. A new window will appear. Click on Attributes. You will get a menu wherein you can rotate the component by the required degrees. Select the same and click OK.
- 8. The next step is to connect the gates. Select the Mode → Draw Wires menu item to begin the process. Say you want to connect the output of the AND gate to the input of the OR gate. Click on the output of the AND gate followed by clicking on the say upper input of the OR gate. Click on the output of the AND gate followed by clicking on the say upper input of the OR gate. A line will appear connecting the output to the input. You can continue in the manner until all the gates are connected as required by the Boolean equation.
- 9. Now that the schematic is done, we need to check it for errors. First, select Options → Create Netlist. This will check your schematic drawing and generates a machine-readable netlist, which describes what types of gates are used and how they are the connected. Next, select Options → Integrity Test to initiate an error check on the netlist. If the netlist has no errors, save the schematic using File → Save As... menu item. Now we must export the netlist. Go to the option Options → Export Netlist and click on it. An Export Netlist window will appear. Select Edit 200 [*.EDN] in the Netlist Format selection box. Click on the Open button. Now select File → Exit to close the schematic editor. On returning to the Project Manager we must make the filename (your file) .SCH as part of the project. Select the Document → Add... menu item and list items of type Schematic (*.SCH) in the dialog window. Highlight your file and click on OK. You should see your file as part of the project.
- We are now ready for simulation. Click on the visible icon called Simulation at the Project Manager. This will bring up the Logic Simulator Foundation Window and a single, empty Waveform Viewer window.
- 11. The first thing to do is add the inputs and outputs of the circuit to the **Waveform Viewer** so that we can see what is happening as the circuit is simulated. Do this by selecting **Signal** \rightarrow **Add Signals**... menu item. The **Component Selection for Waveform Viewer** window will appear. Click on your input name (say "A") to highlight it and then click on the **Add** button. A waveform labeled "A" will appear in the waveform viewer and a red checkmark will appear by the selected signal. We can repeat this procedure for all the inputs. Similarly do the same thing for all the outputs. Then click on the **Close** button.
- 12. Now the inputs and outputs are displayed, but nothing interesting is happening because all the inputs are set to logic 0. Now we need to apply a stimulus to the circuit, so naturally we select the Signal → Add Stimulators... menu item. This brings up the Stimulator Selection window. There are many number of buttons, but we are only interested in a single item: a 16-bit binary counter labeled Bc.
- 13. We now have to do a one to one mapping of all input signals to this circuit. During a simulation, the right-most 4 bits of this counter will go through every possible combination of inputs, from 0000 to 1111. Starting from the lowest input possible, we label that to the right-most bit of the counter by clicking on one of the bit circles. Do this by clicking on the name of an input in the **Waveform Viewer** window (the selected input is highlighted) and then clicking on one of the bit-circles in the **Bc** section of the simulator Section window. The label of the counter bit attached to the circuit input will appear to the right of the input name in the waveform viewer. Once all the inputs are attached to the counter bits, we can click on **Close** to leave the **Simulator Selection** window. We now need to set up a parameter that controls the speed of the simulation. First, select the **Options** → **Preferences** menu item.

Then in the **Preference** window, set the frequency of the "BO" bit of the binary counter at 50 MHz. Then click on OK. Now you can run the simulation. In case you want to run the simulation again, delete the previous waveforms using **Waveform** \rightarrow **Delete** menu.

- 14. We are now ready for simulation. Set the simulation mode to Functional in the drop-down menu in Logic Simulator window tool-bar. This indicates we are doing a functional simulation that checks only the logical operation of our circuit and ignores detailed physical simulation timing issues. You could go for Step by Step simulation or Long simulation. Step by Step simulation is done by pressing the Step simulator waveform on the menu bar at the top of the screen. This simulates the circuit for each clock pulse. Alternatively we could go for long simulation which simulates the circuit for a fixed period of time (time period decided by you) and display the results. To do this go to Goto options-Start Long simulation and set the running time to 1 sec. Click on start. This will test your circuit for 1 second and the waveform will appear on the screen. Check the waveform for all the input combinations and make sure that the output is correct for all conditions. Similarly check the other circuits and show the grader the complete *one cycle of waveforms* for all the required circuits. Caution: When you print the waveforms, check the Page Setup first. You have to select "Current Page" option before you send any file to the printer. To change the size of the schematic output, check the Print option.
- 15. When you finish the experiment, save your files on your diskette and delete all your files when you leave the lab. Failure to do so will result in marking down of the points for that lab.

Using the Xilinx Foundation Series Tools design, test and demonstrate circuits, which implement the following functions.

- 1. f(a,b,c,) = abc' + a'bc + a'b'c'
- 2. $f(x,y,z) = y \cdot z + x' \cdot y \cdot z' + x (y' \cdot z + y \cdot z')$
- 3. $f(w,x,y,z) = \Sigma(1,2,4,5,9,11,12,13,14,15)$
- 4. $f(w,x,y,z) = \Pi(1,2,6,7,8,10)$
- 5. Design a circuit for the following truth table:

Х	у	Z	m
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

6. Design a circuit with four inputs (a,b,c,d) and one output (x) that produces a "1" at the output if there is an equal number of 0's and 1's at the inputs.

EE/CS 120A : Logic Design Lab 3 : Digital Logic Gates

Objectives

The objectives of this laboratory assignment are:

- To get familiar with the Xilinx Foundation Series Tools to design logic circuits.
- To design and implement full adder and full subtractor circuits using Schematic editor and simulator.
- To download your circuit on the prototype board and test it.

Laboratory Instructions

- Create a directory with your name on the drive C of your lab PC. Use this directory to create your project, store your results, bitstreams, ect. during the lab sessions.
- You can bring complete project files on a floppy disk and then use the **Copy Project** command from the Project Manager menu copy it into the directory you created above.
- Alternatively, you can create a new project in your directory on the C drive and then copy your file to that new project directory. Remember to **Add** your .SCH file to the project.
- Perform functional simulation of your design and have it checked by the lab instructor or your TA.
- In case you modify your source file, remember to copy it back to your floppy disk.
- Test and demonstrate your circuits to the lab instructor or your TA.
- Before you leave the lab please **remove** and files or directories you created on your lab PC and leave your workplace at least as clean and tidy as you found it.

Instruction to download the schematic to the board

- 1. Open a new project one project for each of the files you will be working on.
- 2. Perform a functional simulation that checks the logical operation of your circuit. Check the waveform for all the input combinations and make sure that the output is correct for all conditions. If the output is correct, print your schematic and functional analysis output files. You have to select "Current Page" option before you send any file to the printer.
- 3. Open you schematic file. Delete the input and output terminals. Instead, add **IPAD**s (inputs) and **OPAD**s (outputs) in their places.
- 4. Now, you will have to map the input and output pins on the board to these pads. For this, select the components (IPAD or OPAD) and the click on the right mouse button. You will see a popup menu. Select symbol properties. You come to a square dialog box. In the space for name, type in capital letters, LOC, which stands for location. In the space for description, type the corresponding pin number, say p46 to an IPAD. Then, click add. You will see that the pin number you specified is added to the corresponding IPAD or OPAD. Do this for all inputs and outputs using the following tables which specify the input and output pin numbers. For example, X_i →p48, Y_i→p47, C_i→p46, C_{i+1}→p19, and S_i→p23.

XC95108 pin	Xsport	XC95108
p46	B0	p21
p47	B1	p23
p48	B2	p19
p50	B3	p17
p51	B4	p18
p52	B5	p14
p81	B6	p15
p80	B7	p24

XC95108 pin	LED seg
p21	S0
p23	S1
p19	S2
p17	S3
p18	S4
p14	S5
p15	S 6
n24	S 7

- 5. Save the file. Now, go back to the opening screen. You are ready for implementation.
- 6. Click on the **Implementation Design** in the dialog box. Select device name **95108PC84**. Then, click run. If an error message appears on the screen, then you need to find out the cause of the error. Click on verification and find out the cause of your error. Rectify it.
- 7. Now, you are ready for downloading the schematic on to the board. Click on **Programming**. You will come to a screen **JTAG Programmer**. This window lets us produce a stream of bits that is suitable for programming a set of XC9500 chips. Select the **Output** \rightarrow **Create**

SVF File... menu item. Then an **SVF Options** pop-up window will appear. Select Through Test-Logic-Reset, then click on OK.

- 8. Do not save the file in the rev (revision) directory that appears on the popup box but instead to save the **.svf** file we have to go into the **C:\XSTOOLS\BIN** directory and save the file there as our executable format will be built there. Therefore go to that directory and save the file as **yourfilename.svf**.
- 9. Now turn on the power for the XS prototype board. Make sure that the power supply is in the +6V range and adjust the voltage to about 6V.
- 10. Select **Operation** \rightarrow **Get Device ID** and then click on OK.
- 11. In the JTAG popup window that you are in go to **Operations** → **Program** and select "Erase Before Prog" option. In this step erase the program on the board.
- 12. As the board has a ROM you probably will need to erase the previous program stored on the board. Go to **Start** → **Run** and type **C:\XSTOOLS\BIN\Xstest.batxs95-108**. This will clear the old program from the board.
- **13.** You are now ready to download your program. Go to Start \rightarrow Run and type C:\XSTOOLS\BIN\Xsload.exe yourfilename.svf.
- 14. To test the program, we test it for all possible combinations of the input. For example, if we have 3 inputs the test vector shall range from 000 to 111. To test for each condition go to Start → Run and type C:\XSTOOLS\BIN\Xsport.exe yourtestvector where you have to manually change your test vector from 000 to 111 for a 3 input circuit and so on. The corresponding output will be displayed on the LED on the board. For 4 inputs case, you have to change the test vector from 0000 to 1111.
- 15. When you are ready for the experiment, let the TA check your results.
- 16. When you finish your lab experiment, delete all the .svf files under the BIN directory.

Using the Xilinx Foundation Series Schematic-based Tools design, test and demonstrate circuits based on the specifications given in the following table.

Xi	yi	c _i	c _{i+1}	Si
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

- 1. Using AND, OR, and XOR gates design the full adder that is specified by the above table.
- 2. Using NAND and OR gates design the full adder that is specified by the above table.
- 3. Using multiple-input NAND gates design the full adder that is specified by the above table.

Xi	yi	bi	b_{i+1}	dI
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

- 4. Using the basic logic library implement the full subtractor that is specified by the above table.
- 5. Download your circuit on the prototype board and test it. You need to use one of the three Adder and Subtractor circuits you had designed. The output will be seen in the seven segments display and your input will be given via the keyboard to the input pins specified. There are the steps that you will be going through.

EE/CS 120A : Logic Design Lab 4: Multiplexer and Decoder

Objectives

The objectives of this laboratory assignment are:

- To get familiar with the Xilinx Foundation Series Tools to design logic circuits.
- To design and implement Multiplexer and Decoder circuits using Schematic editor and simulator.
- To download your circuit on the prototype board and test it.

Laboratory Instructions

- Create a directory with your name on the drive C of your lab PC. Use this directory to create your project, store your results, bitstreams, etc. during the lab session.
- You can bring complete project files on a floppy disk and then use the **Copy Project** command from the Project Manager menu to copy it into the directory you created above.
- Alternatively, you can create a new project in your directory on the C drive and then copy your file to that new project directory. Remember to Add your .SCH file to the project.
- Perform functional simulation of your design and have it checked by the lab instructor or your TA.
- In case you modify your source file, remember to copy it back to your floppy disk.
- Test and demonstrate your circuit to the lab instructor or your TA.
- Before you leave the lab please **remove** any files or directories you created on the your lab PC and leave your workplace at least as clean and tidy as you found it.

Using the Xilinx Foundation Series Schematic-based Tools design, test and demonstrate circuits. Download your circuit on the prototype board and test it.

- 1. Use as few 4-to-1 multiplexer modules as possible to design a circuit for the following function. Do not use any other logic gates. $f(w,x,y,z) = \Sigma(1,3,5,12,15)$
- 2. Design a 7-segment decoder that has a 4-bit, hexadecimal input $(I_3I_2I_1I_0)$ and seven outputs $(S_0, S_1, S_2, S_3, S_4, S_5, S_6)$ one for each segment of the display. Your decoder should be designed to use the 7-segment display on your prototype board and should be capable of decoding all hexadecimal inputs.

Laboratory Instructions

- Create the VHDL source file(s) for your designs before coming to the lab. You can use the Xilinx HDL editor or any text editor to create your files. Remember to bring these files to the lab on a floppy disk.
- Create a directory with your name on the drive C of your lab PC. Use this directory to create your project, store your results, bitsteams, ect. during the lab session.
- You can bring a complete project (i.e. *project.pdf* file and *project* directory) on a floppy disk and then use the **Copy Project** command from the Project Manager menu to copy it into the directory you created above.
- Alternatively, you create a new project in your directory on the C drive and then copy your VHDL to that new project directory. Remember to *Add* your VHDL file to the project.
- Perform functional simulation of your design and have it checked by the lab instructor or your TA.
- In case you modify your VHDL source file, remember copy it back to your floppy disk.
- If the circuit works as excepted, implement it using the prototyping board assigned to you.
- Use keyboard and LEDs available to apply input stimuli and observe the outputs. Disconnect the XSPORT (parallel port) when you apply input stimulus from the workbench.
- Test and demonstrate your circuit to the lab instructor or your TA.
- Before you leave the lab please remove any files or directories you created on your lab PC and leave your workplace at least as clean and tidy as you found it.

EE/CS 120A : Logic Design Xilinx VHDL Simulation Procedures

VHDL stands for "VHSIC Hardware Description Language." VHSIC, in turn, stands for "Very High Speed Integrated Circuit," which was a U.S. Department of Defense program to encourage research on high-performance IC technology.

The only real difference between the schematic and VHDL design flows is the method by which the netlist is extracted from the design description. Therefore, this procedure will only discuss the steps of entering the VHDL code for the circuit design and the synthesis of the netlist from the VHDL code. The steps which follow (functional simulation, compiling into a bitstream, downloading the bit stream to an XS board, and testing the downloaded design) are performed exactly as they were in the schematic design flow discussed in the previous section.

- 1. Double-click on the Xilinx Foundation Project Manager icon.
- 2. You will get a screen **Getting Started**. Click on **Create a New Project** and then click OK. You will get a screen **New Project**. Enter your project name in the box **Name:** and click on HDL. Click OK after this.
- 3. You will come to a screen that shows you the directory and some default files that have been created. There are three main panes in the Project Manager window. On the upper right-hand pane, you can find a box-**Design Entry**. Click on the leftmost icon, which has the shape of a paper with HDL (icon for the HDL editor). You now see the HDL editor in a new window.
- 4. In the window that appears, click on Use HDL Design Wizard and then click on OK.
- 5. Then click on Next in the **Design Wizard** window. Here, select the VHDL and click Next to move to the **Design Wizard-Name** window. Enter your file name and click Next.
- 6. Now you should be in the **Design Wizard Ports** window where you specify inputs and outputs for your design. To add an input, click on New and then type your input variable in the Name field. Click on Input to see the port to be an input. Repeat these steps for all your inputs. To add your output of your circuit, click on New, enter your output variable and click on the Output. Repeat these steps for all your outputs. In the case your input or output is a vector, you can use Bus option. Now all the inputs and outputs are defined so click on the Finish.
- 7. If you are familiar with HDL editor, you can click on the **Advanced** button to bring up the **Advance Port Settings** window. The drop-down menu in this window lets you set the type of any of the inputs or outputs. The default setting is std-logic.
- 8. At this point, an **HDL Editor** window will appear with the skeleton program. The first line of the file uses the LIBRARY keyword followed by the names of the libraries you want to use in your design. Libraries are used to encapsulate functions

that are generally useful in a wide variety of designs. You can access the macros, definitions, and functions from the IEEE library. A library can be subdivided into package, which further encapsulate features useful in a certain area of type of application. The USE keyword in line 2 indicates that our design will have access to ALL the features found in the std-logic-1164 package of the IEEE library. The IEEE library and the std_logic_1164 package are standards, which are support by the VHDL tools.

- 9. Following the library access control lines define the interface to your circuit. A VHDL entity is simply a declaration of a module's inputs and outputs, while a VHDL architecture is a detailed description of the module's internal structure or behavior. The interface declares the inputs and outputs which an external circuit can use to gain access to the features and functions of the circuit.
- 10. You will use the std_logic type which allows logic signals to take on the standard 1 and 0 Boolean states as well as the undefined, high-impedance, and other states.
- 11. The interface definition is followed by an architecture definition. The VHDL statement in the architecture section describe how the circuit actually carries out the operations on the input/output values passed through the interface. You should enter your statement here. Like other high-level programming languages, VHDL generally ignores spaces and line breaks, and these may be provided as desired for readability. Comments being with two hyphens and end at the end of a line.
- 12. Now we have to check to make sure we have not made any mistakes. Select the **Synthesis** \rightarrow **Check Syntax** menu item. A small window below the HDL editor will appear informing you that the VHDL code is being examined for errors. Within a few seconds, it will state Check Successful if there is no error. Click OK in the pop-up window. If there are errors, each error will be highlighted in the **HDL Editor** window and an error message will appear at the bottom of the window. You can get error-free VHDL code examples for reference by selecting the **Tools** \rightarrow **Language Assistant** menu item.
- 13. VHDL is not synthesized within the HDL Editor window so the Synthesis \rightarrow Synthesize menu item is blanked out.
- 14. Now that the design entry is complete, select File \rightarrow Save in the HDL Editor window. Then select File \rightarrow Exit.
- 15. Upon returning to the **Project Manager** window, you must make yourfilename.VHD file a part of your project. Select the **Document** \rightarrow **Add...** menu item and list items of type **HDL** (yourfilename.VHD) in the dialog window. You should then see your VHDL file under your project name.
- 16. Now, you have to map your input and output variables to the input and output pins on the board. For this, you should double click on the .ucf file under your project name. You will see **Report Browser** window. Go down to the end of the file and type "NET your *input or output* LOC=*pnumber*,". For example, if you use "A" for your input (MSD), then type "NET A LOC=p48,". Add all other inputs and outputs in the same manner. Now the pin assignment is complete, select **File** \rightarrow **Save** and the **File** \rightarrow **Exit**.

- 17. Once the VHDL source file complete, you next need to extract its netlist. Select the **Synthesis** → **Force Analysis of ALL HDL Source Files** in the **Project Manger** window. This indicates a check of all the VHDL files to detect any errors. If there is no error, a green checkmark will appear by the .VHD file name under your project name.
- 18. Next, click on the SYNTHESIS button in the right-hand pane of the **Project Manager** window. This brings up the **Synthesis/ Implementation** window.
- 19. When the window first appears, the name of the interface for your file will be listed in the Top Level text box. The Version Name box shows ver 1 and this will be incremented each time you change the source code for the design.
- 20. In the Target Device are of the window, you will select the family, particular device type, and device speed in the drop-down menus. This lets the synthesis software know the type of chip architecture you are targeting so it can generate a netlist that takes advantage of the features of the chip. Select Family name **XC9500** and Device name **95108PC84**.
- 21. You can also use the controls in the Synthesis Settings are to direct the synthesis tools to emphasize high-speed or area-efficient circuitry. There is also an Insert I/O Pads checkbox, which controls whether input and output buffers will be placed on all I/O signals. This box should be checked if your .VHD file is at the top level of design. (This box would not be checked if the .VHD circuitry is included as a macro in a larger design.)
- 22. Clicking on the Run button starts the synthesis process. If the synthesis is successful, you should see a green mark in the SYTHESIS box of the Flow tab in the **Project Manager** window. You will see a red cross in event of a failure.
- 23. At this point, you have extracted a netlist from the VHDL code that describes your circuit. With this netlist you can do functional simulation, compile the netlist into a bitstream, and download and test the bitstream to an XS95 Board in exactly the same way as was shown in the previous lab on the schematic design.
- 24. Test a functional simulation.
- 25. Download your circuit on the prototype board and test it.

Design Problems

Using the Xilinx Foundation Series Tools design <u>using VHDL</u>, test and demonstrate circuits, which implement the following functions. Your circuits should be as small as possible.

- 1. Design the full adder using VHDL.
- 2. Design a 4-to-1 multiplexer as a module that can later be used as a building block for other circuits.
- Use as few 4-to-1 multiplexer modules as possible design a circuit for the following function. Do not use any other logic gates. F(w,x,y,z) = Σ(1,3,5,11,15)
- 4. Design a 7-segment decoder that has a 4-bit, hexadecimal input $(I_3I_2I_1I_0)$ and seven outputs $(S_0, S_1, S_2, S_3, S_4, S_5, S_6)$ one for each segment of display. Your decoder should be designed to use the 7-segment display on your prototype board and should be capable of decoding all hexadecimal inputs.

EE/CS 120A : Logic Design Lab 6 : Arithmetic and Logic Unit Design

Design Problems

Using the Xilinx Foundation Series Tools design <u>using VHDL</u>, test and demonstrate circuits, which implement the following functions. Your circuits should be as small as possible.

- 1. Design a 1-bit full adder module.
- 2. Design a 3-bit arithmetic unit using 2's complement number representation. The arithmetic unit has two 3-bit inputs (A_{2-0}, B_{2-0}) , a mode input (m), and produce a 3-bit result (S₂₋₀), and a 1-bit overflow flag, *OVFL*. The circuit should perform 2's complement addition $(A_{2-0} + B_{2-0})$ when the module input is 1 and 2's complement subtraction $(A_{2-0} B_{2-0})$ when the mode input is 0. Use the 7-segment decoder designed in Lab 5 as a module in this design to decode and display the output using the on-board 7-segment display.
- 3. Enhance the capability of the arithmetic unit designed above by making it perform bitwise logical operations, *AND* and *OR* on its inputs *A* and *B*. This new circuit is called *Arithmetic* and *Logic Unit (ALU)*. This ALU circuit has a 2-bit mode input (m_{1-0}) besides all the other inputs and outputs described above (Note: for logical operation output flag *OVFL* is ignored). The table below gives a list of functions performed by the ALU based on the mode input. Use the 7-segement decoder designed in Lab 5 as a module in this design to decode and display output using the on-board 7-segment display.

m_1	m ₀	Function
0	0	bit-wise OR
0	1	bit-wise AND
1	0	Addition
1	1	Subtraction

EE/CS 120A : Logic Design Lab 7 : Traffic Light Controller

Design Problems

Design, using the Xilinx VHDL editor, a street-intersection traffic light controller. Both traffic light units have three lights: Red (stop), Yellow (prepare to stop) and Green (go). Each unit should be triggered by either of the two inputs representing cars or pedestrians waiting at a red light. Before one unit changes from red to green, the other unit must change from green to yellow to red.

Due to I/O limits on the XS95 board, you may simulate one set of the red, yellow, and green pattern by lighting segments on the on-board display. (Segment #S2&S5 = G, S0&S6=Y, S1&S4=R). The other set of lights should be connected to the Red, Yellow and Green LED's. Use the Dip switch for the four inputs.

- Download and demonstrate to the TA.
- Conclusion (Write-up).
- How well did your design perform?
- Explain your choice of Flip-flops.
- What changes and/or improvements could you foresee to make your design closer to an actual traffic control system?

Project 1: Simplified Vending Machine Control

Design, using the Xilinx Schematic and VHDL editors, a simplified vending machine. Assume that 5 and 10-cent coins are sequentially deposited in the machine slot until totaling 25-cents. When 25-cents has been totaled, the dispense signal is turned to one from zero. If 25-cents is exceeded, the dispense signal and the change signal will turn from zero to one. Output these signals to light separate LED's on your breadboard. Display the current number of 5-cents deposited on the on-board 7-segment display.

- Download and demonstrate to the TA by trying at least 6 different input sequences of 5 and 10-cent coins.
- Conclusion (Write-up).
- How well did your design perform?
- Explain your choice of Flip-flops.
- What changes and/or improvements could you foresee to make your design closer to an actual vending machine control?
- Extra credit: Add product selection input and two digit 7-segment display. Display total amount deposited.

Project 2: Up/Down Counter

Design, using the Xilinx Schematic and VHDL editors, an up/down counter with clock (approx. 5Hz) and output to a 7-segment display. (You can use the 7-segment decoder designed earlier.) You can use the 12MHz clock on the XS95 (pin 9) as a source for your 5 Hz clock. You may want to test this in stages, substituting some of the hardware from your breadboard circuit in Lab 1.

- Download and demonstrate to the TA.
- Conclusion (Write-up).
- How well did your design perform?
- How did you derive the 5 Hz clock?
- Extra Credit: Add second digit using another 7-segment display.

Project 3 : Combination Lock

Design, using the Xilinx Schematic and VHDL editors, a combination lock that will display the input setting on the 7-segment display and an unlock signal to an LED when the correct code is set on the input Dip switch. Use a 4-bit BCD input code. Set the lock code to be 0101.

- Download and demonstrate to the TA.
- Conclusion (Write-up).
- How well did your design perform?
- Extra credit: Make the combination code programmable using the Xsport utility and make the display two decimal places using 8-bit input.

APPENDIX



XS95 CPLD Board Programmer's Model

©1999 XESS Corporation

21 50.8UL90 These pins drive the individual segments of the LED display 23 51.8UL81 (S0-S6 and DP). They also drive the color, horizontal, and vertical sync signals for a VGA monitor. 16 53.6REEN vertical sync signals for a VGA monitor. 16 54.8E00 an input driven by the 12 MHz oscillator. 24 DPVSYNCB An input driven by the 12 MHz oscillator. 24 DPVSYNCB percent driven by the 12 MHz oscillator. 24 DPVSYNCB percent driven by the 12 MHz oscillator. 24 DPVSYNCB percent driven by the 12 MHz oscillator. 24 PC.05 parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysteris circuitry. 25 PC.05 circuitry. 26 PC.05 circuitry. 27 P1.0 These pins connect to the UC address latch enable 10 X7A.1 Pin that drives the UC program store enable 11 P1.2 These pins connect to the pins of Port 3 of the uC. Some of the P1.4PC S3 26 P1.6 P1.6 37 P1.8PC S4 connects to the data write pin of the UC shores to a status input pin of the UC shores to a status input pin of the UC sho	XS95 Pin	Connects to	Description	
23 51.8.UE1 (SO-S6 and DP). They also drive the color, horizontal, and vertical sync signals for a VGA monitor. 18 S3.GREENt Vertical sync signals for a VGA monitor. 18 S3.REEDT Vertical sync signals for a VGA monitor. 14 S5.REDT These pins are driven by the data output pins of the PC parallel port. 24 DP/SYNCB These pins are driven by the data output pins of the PC parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysterisi sit 36 PC D3 through pins 46 and 47 since these have additional hysterisi sit 31 PC D4 circuitry. 22 PC D5 Pin that drives the uC clock input 36 PC D7 Pin that drives the uC address latch enable 33 PSEN8 Pin that monitors the uC address latch enable 34 P10 These pins connect to the pins of Port 3 of the uC. Some of the pins pins connect to the data write pin of the PC parallel port. 71 P13,PC D5 P13 72 P14,PC 24 parentheses. Pin 63 connect to the data write pin of the PC parallel port. 71 P13,PC 25 These pins connect to Port 3 of the uC. The uC 24 des p34,001,00 73 P3,0KR00,00 These pins con	21	SO.BLUED	These pips drive the individual assessments of it.	
19 52.6REEN0 Vertical sync signals for a VGA monitor. 17 53.6REEN1 Vertical sync signals for a VGA monitor. 18 54.RED0 14 53.6REEN1 14 54.6REN1 14 54.6REN1 15 FC.02 16 PC.02 17 PC.05 18 PC.05 19 XTA.1 10 XTA.1 11 PIn that drives the UC clock input 12 PC.05 13 PSENB 14 Pin that monitors the UC address latch enable 13 PSENB 14 Pin XpC.34 15 P1.6PC.35 16 P1.8PC.35 17 P1.8PC.35 16	23	S1,BLUE1	(S0-S6 and DP). They also drive the LED disp	
17 S3.0REENt 18 S4.RED0 14 S5.RED1 15 S5.HSTNCB 24 OPVSTNC3 9 CLK 46 PC.00 7 PC.01 9 CLK 46 PC.02 47 PC.01 9 CLK 48 PC.02 49 PC.03 51 PC.04 52 PC.06 54 PC.06 55 PC.06 56 PC.07 7 Pin that drives the UC clock input 45 RST 9 In that drives the UC clock input 20 ALEB 9 Pin that monitors the UC address latch enable 11 P1.2 9 Pin that monitors the UC program store enable 11 P1.2 12 PitsPC.53 66 P1.8/PC.54 71 P1.5/PC.53 72 P1.4/PC.54 73 P3.0KRD) 74 </td <td>19</td> <td>S2,GREEN0</td> <td>vertical superside the store the color, horizontal, and</td>	19	S2,GREEN0	vertical superside the store the color, horizontal, and	
15 54,8500 15 58,8601 15 58,8601 24 OPVSTNCB 9 CLK An input driven by the 12 MHz oscillator. 46 PC 00 These pins are driven by the data output pins of the PC parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysterisis 51 PC 03 circuitry. 52 PC 05 circuitry. 51 PC 05 Pin that drives the uC clock input 45 RST Pin that monitors the uC address latch enable 9 PC 05 Pin that monitors the uC address latch enable 13 PSEN6 Pin that monitors the uC address latch enable 14 P1.1 the pins are also connected to the pars of Port 1 of the uC. Some of the pins are also connected to the status input pins of the PC parallel port. 72 P14/9C 23 Parallel port. 73 P3.07800 These pins connect to the pins of Port 3 of the uC. The uC class pecialized functions for each of the port pins indicated 1 parentheses. Pin 63 connects to the data write pin of the uC class a status input pin of the PC parallel port. 74 P3.07809 An as specialized functions for each of the port pins indicated 1 parentheses. Pin 63 connects to the data write pin	17	S3,GREEN1	vertical sync signals for a VGA monitor.	
13 SSMEVUG 24 OPVSTNCB 24 OPVSTNCB 9 CLK An input driven by the 12 MHz oscillator. 46 PC 00 These pins are driven by the data output pins of the PC parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysterisi 51 9 CLK An input driven by the 12 MHz oscillator. 44 PC 02 through pins 46 and 47 since these have additional hysterisi circuitry. 51 PC 05 FC 05 90 PC 06 Pin that drives the UC clock input 45 RST Pin that monitors the UC adock input 10 XTAL1 Pin that monitors the UC program store enable 11 PSEMB Pin that monitors the UC program store enable 12 P1.4PC 34 Parallel port. 71 P1.8PC 35 P1.8PC 35 60 P2.62 P3.0000 71 P1.8PC 35 P1.8PC 35 61 P3.0000 Parentheses. Pin 63 connects to the data write pin of the uC 72 P1.4PC 35 P3.0000 Parentheses. Pin 63 connects to the atas on a multiplexed address/data port. 73 <td< td=""><td>18</td><td>S4,RED0</td><td></td></td<>	18	S4,RED0		
32 Spirates 24 OPVSTNCS 9 CLK An input driven by the 12 MHz oscillator. 46 PC 00 These pins are driven by the data output pins of the PC parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysterisis in PC 05 50 PC 03 circuitry. 51 PC 05 circuitry. 52 PC 05 circuitry. 51 PC 05 circuitry. 52 PC 06 circuitry. 51 PC 05 circuitry. 52 PC 05 circuitry. 53 P3 (PC 28 Pin that drives the uC clock input 72 P14 PC 284 circuitry. 73 P15 PC 285 circuitry. 54 P3 (PR 200	15	35,RED1		
CL Dr. Varitude An input driven by the 12 MHz oscillator. 46 PC 00 These pins are driven by the data output pins of the PC parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysteris circuitry. 51 PC 03 through pins 46 and 47 since these have additional hysteris circuitry. 52 PC 05 circuitry. 60 PC 07 Pin that drives the uC clock input 51 PC 06 Pin that monitors the uC adoex input 60 PC 07 Pin that monitors the uC adoex input 71 P1.1 Pin that monitors the uC program store enable 72 P1.0 These pins connect to the pins of Port 1 of the uC. Some of the pins of Part 2 of the uC. The uC parallel port. 73 P1.4.PC 34 These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated 1 parentheses. Pin 63 connects to the data write pin of the uC address. Pin 63 connects to the data write pin of the uC address/data port. 74 P1.4.PC 34 These pins connect to Port 0 of the uC which is also a multiplexed address/data port. 75 P3.0000.00 These pins connect to Port 2 of the uC which is also a multiplexed address/data port. 75 P3.0000.00 These pins connect to Port 2 of the uC which is also o a multipl	24	DR VEYNCB		
J CtA An input driven by the 12 MHz oscillator. 46 PC 00 These pins are driven by the data output pins of the PC parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysterisi circuitry. 50 PC 03 circuitry. 51 PC 05 circuitry. 52 PC 05 circuitry. 60 PC 07 Pin that drives the uC clock input 10 XTAL1 Pin that drives the uC reset input 20 ALEB Pin that monitors the uC address latch enable 11 PSENB Pin that monitors the uC address latch enable 6 P10 These pins are also connected to the status input pins of the PC parallel port. 71 P13.FPC 35 parallel port. 72 P14.PC 54 These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated i 71 P13.FPC 35 P34000 68 P13.WR200 These pins connect to Port 0 of the uC which is also a multiplexed address/data port. 72 P14.PC 54 These pins connect to Port 0 of the uC which is also a multiplexed address/data port. 73 P340000, D0 These pins connect to Port 2 of the u	0	OP,VSTNUB		
Bit PC 00 These pins are driven by the data output pins of the PC 48 PC 02 brough pins 46 and 47 since these have additional hysterisi circuitry. 51 PC 05 circuitry. 51 PC 05 circuitry. 61 PC 05 circuitry. 61 PC 05 circuitry. 61 PC 07 Pin that drives the uC clock input 62 PC 05 Pin that drives the uC clock input 64 RST Pin that drives the uC clock input 65 PC 07 Pin that drives the uC clock input 74 Pin that drives the uC clock input parallel port. 75 P1,1 These pins connect to the pins of Port 1 of the uC. Some of the pins of Port 2 of the uC. The uC for P1, FC 25 76 P1,1/PC 254 These pins connect to the pins of Port 3 of the uC. The uC for P1, FC 25 77 P1,1/PC 254 These pins connect to the pins of Port 3 of the uC. The uC clock input pin of the P2 200(PD) and the write-enable pin of the PC parallel port. 78 P14, PC 255 parentheses. Pin 63 connects to the data write pin of the uC which is also a mutiplexed address/data port. These pins also connect to the for 040 of the uC which is also a for 040 (p0), p0 79 P3 (MO1, p1) Th	48	OLK CLK	An input driven by the 12 MHz oscillator.	
16 PC 02 parallel port. Clocking signals can only be reliably applied through pins 46 and 47 since these have additional hysteris circuitry. 17 PC 03 circuitry. 18 PC 05 circuitry. 19 XTAL1 Pin that drives the uC clock input 10 XTAL1 Pin that drives the uC clock input 10 XTAL1 Pin that monitors the uC orgarem store enable 11 PEC 03 Pin that monitors the uC orgarem store enable 12 ALE8 Pin that monitors the uC orgarem store enable 13 PSENB Pin that monitors the uC orgarem store enable 14 P12 parallel port. 13 PSENB Pin that monitors the uC orgarem store enable 6 P10 These pins connect to the pins of Port 1 of the uC. Some of the pins are also connects to the data write pin of the PC parallel port. 72 P14.PC 54 Past(TKD).PC 58 67 P1.7 These pins connect to the port pins indicated 1 71 P15.PC 53 Past(RKD).PC 58 68 P16.PC 53 Past(RKD).PC 58 73 P3.01(AD1).D1 <td>40</td> <td>PC_00</td> <td>These pins are driven by the data output pins of the BC</td>	40	PC_00	These pins are driven by the data output pins of the BC	
90 10 20 <th< td=""><td>' 48</td><td>PC_01</td><td>parallel port. Clocking signals can only be reliably applied</td></th<>	' 48	PC_01	parallel port. Clocking signals can only be reliably applied	
si pc 03 circuitry. si pc 05 circuitry. si pc 06 circuitry. si pc 05 pin that drives the uC clock input 45 RsT Pin that drives the uC clock input 46 RsT Pin that drives the uC clock input 20 ALEB Pin that drives the uC clock input 46 P10 These pins connect to the pins of Port 1 of the uC. Some of the pins pc 35 67 P1.7 These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the PC parallel port. 33 P3.5(TRD) P3.6(WR0).00 These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to a status input pin of the PC parallel port. 34 P3.6(WR0).00 These pins connect to Port 2 of the uC which is also a multiplexed address/data port. These pins also connect to the 7 upper address bits of the RAM. 35 <td>50</td> <td>PC D3</td> <td>through pins 46 and 47 since these have additional busies</td>	50	PC D3	through pins 46 and 47 since these have additional busies	
S2 PC 05 B1 PC 05 B0 PC 07 10 XTAL1 Pin that drives the uC clock input 11 XTAL1 Pin that drives the uC clock input 20 ALEB Pin that drives the uC reset input 20 ALEB Pin that monitors the uC address latch enable 13 PSENB Pin that monitors the uC program store enable 6 P1.0 These pins connect to the pins of Port 1 of the uC. Some of the pins of Port 2 of the uC. Some of parallel port. 71 P1.5 PC 33 These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated 1 parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the RAM. Pin 70 connects to a and the write-enable pin of the RAM. Pin 70 connects to a astatus input pin of the PC parallel port. 73 P3.6(T1) Status input pin of the PC parallel port. 74 P3.8(NRB).WEB These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 75 P2.0(A1), A10 These pins drive the 8 lower address bits of the RAM. 76 P2.0(A3), A14 </td <td>51</td> <td>PC D4</td> <td>circuitry.</td>	51	PC D4	circuitry.	
81 PC 06 80 PC 07 10 XTAL1 Pin that drives the uC clock input 45 RST Pin that drives the uC reset input 20 ALEB Pin that monitors the uC arores latch enable 13 PSEN8 Pin that monitors the uC arores enable 6 P1.0 These pins connect to the pins of Port 1 of the uC. Some of the prist of the pins are also connected to the status input pins of the PC parallel port. 71 P1.5, PC S3 Pin that monitors for each of the port pins indicated i parentheses. Pin 63 connects to the data write pins indicated i and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the PC parallel port. 28 P3.5(MR0) These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to the 44 PO (A00), 00 29 P3.6(MR0), WEB These pins connect to Port 2 of the uC which is also a multiplexed address/data port. These pins also connect to the 7 upper address bits of the RAM. 31 P0.4(A01, 04) These pins connect to Port 2 of the uC which also outputs the upper address bits of the RAM. 32 P2.0(A10, A10 These pins also connect to the 7 upper address bits of the RAM. 33 P0.4(A04, 04) These pins also connect to the 7 upper address bits of the RAM. 34 P0.4(A04, 0	52	PC D5		
80 PC 07 10 XTAL1 Pin that drives the uC clock input 45 RST Pin that drives the uC reset input 20 ALEB Pin that monitors the uC reset input 13 PSEN8 Pin that monitors the uC program store enable 6 P10 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. 71 P1.5,PC.54 P1.6,PC.54 71 P1.5,PC.55 Fin that monitors for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the PC parallel port. 26 P3.4(T0) These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to a status input pin of the PC parallel port. 33 P0.4(AD4),D These pins connect to Port 2 of the uC which is also a multiplexed address/data port. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 34 P0.4(AD1),01 These pins drive the 8 lower address bits of the RAM. 35 P0.7(AD3),05 These pins drive the 8 lower address bits of the RAM. 36 P2.0(A13),A13 These pins are not connected to other de	81	PC_D6		
10 XTAL1 Pin that drives the uC clock input 45 RST Pin that drives the uC reset input 20 ALEB Pin that monitors the uC arget saltch enable 13 PSENB Pin that monitors the uC argers store enable 6 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 PO for 1 P1.7 71 P1.4, PC, S4 P1.7 71 P1.6, PC, S5 P1.7 66 P1.8, PC, S3 P1.7 70 P3.0(RXD) These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the RAM. 68 P3.3(NTB1) and the write-enable pin of the RAM. 33 P3.6(WRB), WEB These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 34 P0.6(ADG), D0 These pins drive the 8 lower address bits of the RAM. 35 P2.0(A13), A10 These pins drive the 8 lower address bits of the RAM. 36 P0.6(ADG), D0 These pins are not connected to other	80	PC_D7		
45 RST Pin that drives the UC clock input 20 ALEB Pin that drives the UC clock input 13 PSENB Pin that monitors the UC program store enable 6 P10 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. 71 P1.2 parallel port. 72 P1.4,PC,S4 Pin.7 71 P1.5,PC,S3 Pinsep pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the UC and the write-enable pin of the UC which is also a multiplexed address/data port. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 33 P0.4(AD4), D4 These pins drive the 8 lower address bits of the RAM. 34 P2.0(A13), A13 These pins and cutput enable. 35 P2.0(A14), A14 These pins are not connected to other devices and can be used as general purpose I/O. 36 P2.0(A15), A13 These pins are condeconected to other devices and can be used as general purpose I/O.	10	XTAL1	Pin that drives the uC shall include	
20 ALEB Plin that monitors the uC reset input 13 PSENB Plin that monitors the uC program store enable 6 P10 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. 71 P1.5/PC S3 86 P13.6/PC S3 87 P1.7 71 P3.6/PC S3 86 P13.7/PC S4 71 P3.1(TED) PC S6 87 P3.3(INTB) 98 P3.3(INTB) 93 P3.4(INTB) 93 P3.4(INTB) 94 P0.4(AD0).D0 74 P3.4(ROB).00 75 P0.3(AD0).00 76 P1.4(AD1, D1 77 These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to the 37 79 P0.3(AD3).03 41 P0.3(AD1, D1 77 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 byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 33 P2.0(A13), A13 64 P2.0(A21), A12 75 P2.0(A13), A13 76 P10 that drives the RAM output enable. 77	45	RST	Pin that drives the UC clock input	
13 Psexe Pin that monitors the uC address latch enable 6 P10 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. 71 P15,PC 53 Pin that monitors for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC the write-enable pin of the PC parallel port. 70 P3.1(TXD), PC 96 These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC 26 71 P3.0(RXD) These pins connect to PC parallel port. 73 P3.0(RXD) These pins connect to PC parallel port. 74 P3.0(RXD) These pins connect to PC parallel port. 75 P3.0(RXD), D1 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 RAM. 76 P2.0(A10, A10 These pins drive the 8 lower address bits of the RAM. 77 P3.0(A05, D6 These pins drive the 8 lower address bits of the RAM. 78 P2.0(A11, A11 These pins are not connected to other devices and can be used as general purpose I/O. 78 P2.0(A13, A3 Pin that drives the RAM output enable. 79 A1 A4 <	20	ALER	Pin that drives the uC reset input	
Price Print that monitors the uC program store enable 6 P1.0 These pins connect to the pins of Port 1 of the uC. Some of the pins prevailed port. 77 P1.1 the pins are also connected to the status input pins of the PC parallel port. 77 P1.4, PC, S4 P1.7 71 P1.5, PC, S3 P1.7 86 P1.8, PC, S5 P1.7 87 P3.2 (ITXD), PC, S6 Parentheses. Pin 63 connects to the data write pin of the uC has specialized functions for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the PA.4M. Pin 70 connects to a status input pin of the PC parallel port. 33 P3.6(WRB), WEB These pins connect to Port 0 of the uC which is also a multiplexed address/data port. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 34 P2.0(AB), A5 These pins drive the 8 lower address bits of the RAM. 35 P2.0(AB), A11 These pins are not connected to other devices and can be upper address bits of the RAM. 35 P2.0(AB), A3 These pins are not connected to other devices and can be upper address the RAM cutput enable. 36 P2.0(AB), A11 These pins are not connected to other devices and can be used as general purpose I/O.	12	ALEO	Pin that monitors the uC address latch enable	
* F1.0 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 PO parallel port. 71 P1.4 PC 54 P1.7 71 P1.5 PC 33 P1.6 PC 55 67 P1.7 These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the PC parallel port. 73 P3.3(INTED) These pins connect to PC parallel port. 74 P3.4(WRB), WEB P3.4(WRD).00 75 P3.4(WRD).01 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 RAM. 75 P2.0(A10, A10 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 RAM. 75 P2.0(A10, A10 These pins drive the 8 lower address bits of the RAM. 75 A0 These pins are not connected to other devices and can be used as general purpose I/O. 75 P2.0(A13, A12 Pin that drives the RAM output enable. 75 P2.0(A13, A14 Pin that drives the RAM output enable. 75 P2.0(A13, A14 Pin that drives the RAM output enable. 75 <td< td=""><td>13</td><td>PSENB</td><td>Pin that monitors the uC program store enable</td></td<>	13	PSENB	Pin that monitors the uC program store enable	
11 P1.1 P1.2 11 P1.3 P1.3 72 P1.4 PC S4 P1.7 71 P1.5 PC S3 P1.7 96 P1.8 PC S5 P1.7 31 P3.0(RX0) P3.1(TX0) PC S6 67 P1.7 P1.6 PC S3 96 P3.2(IXD) PC S6 Passing Connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the PC parallel port. 73 P3.3(IXD) P3.5(IXD) 74 P3.5(IXD) P3.5(IXD) 75 P3.6(ADS), D5 These pins connect to Port 0 of the uC which is also a multiplexed address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 76 P2.0(A10, A10 These pins drive the 8 lower address bits of the RAM. 77 P2.0(A10, A10 These pins are not connected to other devices and can be used as general purpose I/O. 77 P2.0(A10, A11 P3.4 P3.4 78 P2.0(A10, A11 P3.4 79 P2.0(A15), A12 P3.4 77 P2.0(A13), A13 P3 78<	7	P1.0	These pins connect to the pins of Port 1 of the uC. Some of	
1 F1.2 72 P1.4.PC_S4 71 P1.5.PC_S3 66 P1.6.PC_S5 67 P1.7 31 P3.0(RX0) 70 P3.1(TX0), PC_S6 68 P3.2(INT80) 69 P3.2(INT80) 68 P3.2(INT80) 69 P3.2(INT80) 70 P3.1(TX0), PC_S6 68 P3.3(INT80) 70 P3.1(TX0), PC_S6 68 P3.3(INT80) 70 P3.1(TX0), PC_S6 71 P3.6(TT1) 72 P3.4(TO1) 73 P3.6(NR0), WEB 74 P3.6(NR0), 00 74 P3.6(NR0), 02 75 A0 76 P2.0(A13), A13 75 A0	11	P1.1	- the pins are also connected to the status input airs of the	
72 PI.4.PC S4 71 PI.5.PC S3 66 PI.6.PC S5 67 PI.7 31 P3.0(RXD) 70 P3.1(TXD).PC S6 68 P3.2(INTB0) 71 parentheses. Pin 63 connects to the data write pin of the uC 72 P3.7(RD) 73 P3.6(WRB).WEB 74 P3.7(RD) 75 A0 76 P2.0(A10).A10 77 P2.0(A10).A11 78 P3.6(WRB).WEB 79 P3.6(A06).00 70 P0.5(A05).05 77 P2.0(A10).A10 78 P2.0(A10).A10 79 P2.0(A10).A10 79 P2.0(A10).A11 79 P2.0(A10).A11 </td <td></td> <td>P1.2</td> <td>parallel port.</td>		P1.2	parallel port.	
71 P13,072,35 86 P13,87C,35 67 P1,7 31 P3,0(RXD) 70 P3.1(TXD), PC,36 68 P3.3(INTB0) 71 parentheses. Pin 63 connects to the data write pin of the uC 71 parentheses. Pin 63 connects to the data write pin of the uC 71 parentheses. Pin 63 connects to the data write pin of the uC 72 P3.7(RD8) 73 P3.6(WRB), WEB 74 P8.0(AD0, D0 75 P0.1(AD1), D1 76 P0.1(AD1), D1 77 P0.5(AD3), D3 78 P2.0(A10), A10 79 A1 79 A1 82 A2 79 A1 83 A5 79 A1 84 P2.0(A13), A12 75 P2.0(A13), A12 76 P2.0(A13), A13 76 P2.0(A13), A13 76 P2.0(A13), A14 77 P2.0(A13), A13 76 P2.0(A13), A13 77 P2.0(A13), A14	72	PLADC CA		
66 P18/PC 35 67 P1.7 31 P3.0(RxD) 70 P3.1(TXD), PC 56 69 P3.2(INTB) 68 P3.3(INTB) 68 P3.4(T0) 26 P3.4(T0) 31 P3.6(NTB) 63 P3.6(NTB) 63 P3.6(NTB) 64 P3.8(NTB) 32 P3.7(IDB) 33 P3.6(T1) 34 P0.1(AD1), D1 44 P0.2(AD2), D2 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 RAM. 43 P0.1(AD1), D1 multiplexed address/data port. These pins also connect to the data pins of the RAM. 36 P0.6(AD6), D6 D6 37 P0.5(AD6), D5 D5 36 P2.0(A13), A11 D1 57 P2.0(A13), A12 These pins drive the 8 lower address bits of the RAM. 58 P2.0(A13), A13 These pins drive the 8 lower address bits of the RAM. 57 P2.0(A13), A13 These pins drive the 8 lower address bits of the RAM. 58 P2.0(A13), A13 </td <td>71</td> <td>P15.PC \$3</td> <td>- A A A A A A A A A A A A A A A A A A A</td>	71	P15.PC \$3	- A A A A A A A A A A A A A A A A A A A	
67 P1.7 31 P3.0(RXD) 70 P3.1(IXD), PC.36 69 P3.2(INTB0) 28 P3.4(INTB0) 28 P3.4(INTB0) 33 P3.5(T1) 34 P3.4(INTB0) 35 P3.4(INTB0) 36 P3.4(INTB0) 37 P3.6(INTB0) 44 P0.0(AD0), D0 45 P0.4(AD1), D1 70 P0.4(AD4), D4 37 P0.4(AD4), D4 38 P0.6(AD6), D6 39 P0.4(AD4), D4 37 P0.4(AD4), D4 38 P2.0(A13), A13 41 P2.0(A13), A13 58 P2.0(A13), A13 59 P2.0(A13), A13	66	P1.6.PC 35	-	
31 P3.0(RX0) These pins connect to the pins of Port 3 of the uC. The uC has specialized functions for each of the port pins indicated i parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the PC parallel port. 33 P3.5(11) parentheses. Pin 63 connects to the data write pin of the uC and the write-enable pin of the PC parallel port. 63 P3.4(T0) and the write-enable pin of the PC parallel port. 63 P3.6(WRB), WEB status input pin of the PC parallel port. 32 P3.7(R0B) These pins connect to Port 0 of the uC which is also a mutiplexed address/data port. These pins also connect to the data pins of the RAM. 44 P0.0(AD0), D0 These pins connect to Port 2 of the uC which is also a mutiplexed address/data port. These pins also connect to the data pins of the RAM. 39 P0.4(AD6), D4 P0.6(AD6), D6 36 P0.6(AD6), D6 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 RAM. 54 P2.0(A11), A11 address bits of the RAM. 55 P2.0(A13), A13 These pins drive the 8 lower address bits of the RAM. 75 A0 These pins drive the 8 lower address bits of the RAM. 79 A1 A3 84 A3<	67	P1.7	-	
70 P3.1(TXD), PC_S6 Inass perialized functions for each of the port pins indicated in parentheses. Pin 63 connects to the data write pins of the uC 26 P3.4(T0) and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the PC parallel port. 33 P3.5(T1) status input pin of the PC parallel port. 32 P3.7(ROB) 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 RAM. 41 P0.2(A02), 02 These pins connect to Port 2 of the uC which is also a multiplexed address/data port. These pins also connect to the data pins of the RAM. 39 P0.4(A04), 04 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 byte. These pins also connect to the 7 upper address bits of the RAM. 54 P2.0(A10), A10 These pins drive the 8 lower address bits of the RAM. 55 P2.0(A11), A11 address bits of the RAM. 54 P2.0(A15), A13 These pins drive the 8 lower address bits of the RAM. 55 P2.0(A15), A13 These pins are not connected to other devices and can be used as general purpose I/O. 62 OEB Pin that drives the RAM chip enable. 75 FREE1 These pins are not connected to other devices and can be used as general purpose I/O. <td>31</td> <td>P3.0(RXD)</td> <td>These nine connect to the size of Days to the</td>	31	P3.0(RXD)	These nine connect to the size of Days to the	
69 P3.2(INTB0) Parentheses. Pin 63 connects to the data write pins indicated i parentheses. Pin 63 connects to the data write pins of the uC and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the PC parallel port. 63 P3.4(T0) and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the PC parallel port. 63 P3.4(WB).WEB status input pin of the PC parallel port. 63 P3.4(NB).WEB 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 RAM. 44 P0.0(AD0).00 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 RAM. 39 P0.4(AD4).04 parentheses byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 36 P0.0(AD6).05 These pins drive the 8 lower address bits of the RAM. 54 P2.0(A10).A10 These pins drive the 8 lower address bits of the RAM. 55 P2.0(A15). These pins drive the 8 lower address bits of the RAM. 61 P2.0(A15). These pins are not connected to other devices and can be used as general purpose I/O. 74 FREE1 used as general purpose I/O.	70	P3.1(TXD), PC_S6	has special and the pins of Port 3 of the uC. The uC	
68 P3.3(INTB1) parentheses. Pin 63 connects to the data write pin of the uC 28 P3.4(T0) and the write-enable pin of the RAM. Pin 70 connects to a 33 P3.5(T1) status input pin of the PC parallel port. 63 P3.6(WRB), WEB and the write-enable pin of the PC parallel port. 32 P3.7(RDB) These pins connect to Port 0 of the uC which is also a 44 P0.0(AD0, D0 These pins connect to Port 0 of the uC which is also a 41 P0.3(AD3, D2 multiplexed address/data port. These pins also connect to 40 P0.3(AD3, D3 multiplexed address/data port. These pins also connect to 41 P0.6(AD6), D6 These pins connect to Port 2 of the uC which also outputs the data pins of the RAM. 37 P0.6(AD6), D6 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 RAM. 58 P2.0(A1), A11 address bits of the RAM. 53 P2.0(A12), A12 These pins drive the 8 lower address bits of the RAM. 54 P2.0(A13), A13 These pins drive the 8 lower address bits of the RAM. 52 A2 These pins are not connected to other devices and can be 62 OEB Pin that drive	69	P3.2(INTB0)	has specialized functions for each of the port pins indicated in	
28 P3.4(T0) and the write-enable pin of the RAM. Pin 70 connects to a status input pin of the PC parallel port. 33 P3.5(T1) status input pin of the PC parallel port. 32 P3.7(RDB) 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 RAM. 44 P0.3(AD3), D2 multiplexed address/data port. These pins also connect to the data pins of the RAM. 40 P0.3(AD3), D3 po.5(AD6), D6 36 P0.5(AD6), D5 po.5(AD6), D6 36 P0.5(AD6), D6 per address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 58 P2.0(A11), A11 address bits of the RAM. 53 P2.0(A12), A12 address bits of the RAM. 57 P2.0(A13), A13 address bits of the RAM. 57 P2.0(A13), A14 address bits of the RAM. 34 P2.0(A15) These pins drive the 8 lower address bits of the RAM. 61 P2.0(A15) These pins drive the 8 lower address bits of the RAM. 62 A2 Pin that drives the RAM output enable. 63 A6 Pin that drives the RAM cutput enable. 64 FREE0 <td>68</td> <td>P3.3(INTB1)</td> <td>parentneses. Pin 63 connects to the data write pin of the uC</td>	68	P3.3(INTB1)	parentneses. Pin 63 connects to the data write pin of the uC	
33 P3.8(11) status input pin of the PC parallel port. 53 P3.7(RDB) status input pin of the PC parallel port. 44 P0.0(AD0), 00 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 RAM. 40 P0.3(AD3), 03 These pins connect to Port 2 of the uC which is also a multiplexed address/data port. These pins also connect to the data pins of the RAM. 39 P0.4(AD4), D4 These pins connect to Port 2 of the uC which also outputs the data pins of the RAM. 36 P0.6(AD6), D6 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 RAM. 58 P2.0(A10), A10 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 57 P2.0(A13), A13 These pins drive the 8 lower address bits of the RAM. 57 P2.0(A13), A14 These pins drive the 8 lower address bits of the RAM. 34 P2.0(A13), A13 These pins drive the 8 lower address bits of the RAM. 52 A2 Pin that drives the RAM cutput enable. 64 A3 These pins are not connected to other devices and can be used as general purpose I/O. 74 FREE2 These pins are not connected to other devices and can be used as gen	26	P3.4(T0)	and the write-enable pin of the RAM. Pin 70 connects to a	
63 P3.6(WRB), WEB 32 P3.7(R0B) 44 P0.0(AD0), 00 43 P0.1(AD1), D1 44 P0.2(AD2), 02 45 P0.3(AD3), 03 46 P0.3(AD3), 03 47 P0.3(AD3), 03 39 P0.4(AD4), D4 37 P0.5(AD5), 05 36 P2.0(AB), A8 58 P2.0(AB), A8 58 P2.0(AB), A8 58 P2.0(AB), A9 59 P2.0(A10), A10 56 P2.0(A12), A12 57 P2.0(A13), A13 61 P2.0(A13), A13 61 P2.0(A13), A13 61 P2.0(A14), A14 34 P2.0(A13), A13 61 P2.0(A13), A13 61 P2.0(A14), A14 34 P2.0(A15) 77 A1 78 A2 84 A3 1 A4 3 A5 83 A6 2 A7 62 OEB Pin that drive	33	P3.5(T1)	status input pin of the PC parallel port.	
32 P3./(R08) 44 P0.0(AD0). D0 These pins connect to Port 0 of the uC which is also a 43 P0.1(AD1). D1 multiplexed address/data port. These pins also connect to 41 P0.2(AD2). D2 the data pins of the RAM. 39 P0.4(AD4). D4 the data pins of the RAM. 37 P0.5(AD5). D5 the data pins of the RAM. 36 P0.6(AD6). D6 the data pins of the RAM. 35 P0.7(AD7). D7 the data 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 RAM. 54 P2.0(A10). A10 address bits of the RAM. 53 P2.0(A11). A11 address bits of the RAM. 53 P2.0(A12). A12 address bits of the RAM. 57 P2.0(A13). A13 these pins drive the 8 lower address bits of the RAM. 61 P2.0(A14). A14 these pins drive the 8 lower address bits of the RAM. 82 A2 these pins are not connected to other devices and can be 75 A0 These pins are not connected to other devices and can be 1 A4 Pin that drives the RAM chip enable. 62 OEB Pin that drives the RAM chip e	12	P3.6(WR8), WEB		
13 P0.1(AD1), 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 RAM. 39 P0.4(AD4), D4 He data pins of the RAM. 39 P0.4(AD4), D4 He data pins of the RAM. 31 P0.5(AD5), D5 He data pins of the RAM. 36 P0.6(AD6), D6 He data pins of the RAM. 35 P0.7(AD7), D7 Hese 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 RAM. 58 P2.0(A10), A10 Hese pins drive the 8 lower address bits of the RAM. 53 P2.0(A11), A11 Hese pins drive the 8 lower address bits of the RAM. 53 P2.0(A13), A13 Hese pins drive the 8 lower address bits of the RAM. 54 P2.0(A13), A13 Hese pins drive the 8 lower address bits of the RAM. 55 P2.0(A13), A13 Hese pins drive the 8 lower address bits of the RAM. 52 A2 Hese pins are not connected to other devices and can be used as general purpose I/O. 56 CEB Pin that drives the RAM chip enable. 57 P2.0(A15) These pins are not connected to other devices and can be used as general purpose I/O. 58 REE2 Hese p	44	P3./(RUB)		
41 P0.2(A02), 02 multiplexed address/data port. These pins also connect to 40 P0.3(A03), 03 multiplexed address/data port. These pins also connect to 39 P0.4(A04), 04 multiplexed address/data port. These pins also connect to 37 P0.5(A05), 05 multiplexed address/data port. These pins also connect to 36 P0.4(A04), 04 multiplexed address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 58 P2.0(A10, A10 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 53 P2.0(A12), A11 address bits of the RAM. 54 P2.0(A12), A12 multiplexed address bits of the RAM. 57 P2.0(A13), A13 multiplexed address bits of the RAM. 61 P2.0(A14), A14 multiplexed address bits of the RAM. 34 P2.0(A15) These pins drive the 8 lower address bits of the RAM. 75 A0 multiplexed training the reset of the RAM output enable. 62 0EB Pin that drives the RAM output enable. 63 CEB Pin that drives the RAM output enable. 64 FREE0 These pins are not connected to other devices and can be used as general purpose I/O. <td>43</td> <td>P01(AD1) D1</td> <td>These pins connect to Port 0 of the uC which is also a</td>	43	P01(AD1) D1	These pins connect to Port 0 of the uC which is also a	
40 P0.3(A03), D3 the data pins of the RAM. 39 P0.4(A04), D4 37 P0.5(A05), D5 36 P0.7(A07), D7 58 P2.0(A8), A8 56 P2.0(A8), A9 57 P2.0(A11), A11 53 P2.0(A12), A12 57 P2.0(A12), A12 57 P2.0(A13), A13 61 P2.0(A12), A12 57 P2.0(A12), A12 57 P2.0(A12), A12 57 P2.0(A13), A13 61 P2.0(A14), A14 34 P2.0(A14), A14 35 A0 75 A0 79 A1 82 A2 84 A3 1 A4 3 A5 83 A6 2 A7 62 DEB Pin that drives the RAM output enable. 65 CEB Pin that drives the RAM chip enable. 12 FREE0 These pins are not connected to other devices and can be used as general purpose I/O. 74 FREE3 Th	41	P0 2(AD2) D2	multiplexed address/data port. These pins also connect to	
39 P0.4(AD4), D4 37 P0.5(AD5), D5 36 P0.6(AD6), D6 35 P0.7(AD7), D7 58 P2.0(A8), A9 54 P2.0(A1), A10 55 P2.0(A1), A11 53 P2.0(A13), A13 61 P2.0(A14), A14 34 P2.0(A15) 75 A0 79 A1 82 A2 84 A3 1 A4 3 A5 83 A6 2 A7 62 OEB Pin that drives the RAM output enable. 65 CEB Pin that drives the RAM chip enable. 12 FREE0 These pins are not connected to other devices and can be used as general purpose I/O. 74 FREE3 FREE4	40	P0.3(AD3), D3	the data pins of the RAM.	
37 P0.5(AD5), D5 36 P0.6(AD6), 06 35 P0.7(AD7), D7 58 P2.0(A8), A8 58 P2.0(A9), A9 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 53 P2.0(A10), A10 53 P2.0(A11), A11 53 P2.0(A12), A12 57 P2.0(A13), A13 61 P2.0(A13), A13 79 A1 79 A1 82 A2 84 A3 1 A4 3 A5 83 A6 2 A7 62 OEB Pin that drives the RAM output enable. 65 CEB Pin that drives the RAM chip enable. 12 FREE0 These pins are not connected to other devices and can be used as general purpose I/O. 76<	39	P0.4(AD4), D4		
36 P0.6(AD6), D6 35 P0.7(AD7), D7 58 P2.0(AB), A9 upper address byte. These pins also connect to the 7 upper address byte. These pins also connect to the 7 upper address bits of the RAM. 54 P2.0(A10, A10 address bits of the RAM. 55 P2.0(A12), A12 address bits of the RAM. 57 P2.0(A13), A13 address bits of the 8 lower address bits of the RAM. 61 P2.0(A14), A14 address bits of the 8 lower address bits of the RAM. 75 A0 These pins drive the 8 lower address bits of the RAM. 79 A1 address bits of the RAM. 81 A3 A5 83 A5 address bits of the RAM output enable. 62 DEB Pin that drives the RAM output enable. 64 FREE0 These pins are not connected to other devices and can be used as general purpose I/O. 74 FREE3 FREE4	37	P0.5(AD5), D5		
35 P0.7(AD7), D7 58 P2.0(A8), A8 56 P2.0(A8), A8 57 P2.0(A10), A10 57 P2.0(A12), A12 57 P2.0(A12), A12 57 P2.0(A12), A13 61 P2.0(A13), A13 61 P2.0(A14), A14 34 P2.0(A15) 75 A0 79 A1 82 A2 84 A3 1 A4 3 A5 83 A6 2 A7 62 DEB 63 Pin that drives the RAM output enable. 65 CEB 9 These pins are not connected to other devices and can be 12 FREE1 12 FREE3 78 FREE3 78 FREE3	36	P0.6(AD6), D6		
38 P2:0(AB), A8 58 P2:0(AB), A9 54 P2:0(A10), A10 55 P2:0(A10), A10 55 P2:0(A11), A11 53 P2:0(A12), A12 57 P2:0(A13), A13 51 P2:0(A12), A12 57 P2:0(A13), A13 51 P2:0(A12), A12 57 P2:0(A13), A13 51 P2:0(A12), A12 57 P2:0(A13), A13 34 P2:0(A15) 75 A0 79 A1 82 A2 84 A3 1 A4 3 A5 63 A6 2 A7 62 OEB 63 A6 2 A7 64 FREE0 76 PREE1 12 FREE2 76 FREE3 76 FREE3 76 FREE3	35	P0.7(AD7), D7		
30 P2.0(A9), A9 54 P2.0(A10), A10 55 P2.0(A11), A11 53 P2.0(A12), A12 57 P2.0(A13), A13 61 P2.0(A13), A14 79 A1 79 A1 84 A3 1 A4 3 A5 83 A6 2 A7 62 OEB 9in that drives the RAM output enable. 65 CEB 9in that drives the RAM chip enable. 12 FREE0 12 FREE1 25 FREE2 76 FREE3 77 FREE3 78 FREE4	58	P2.0(A8), A8	These pins connect to Port 2 of the uC which also and a	
34 P20(A10), A10 55 P20(A11), A11 53 P20(A12), A12 57 P2.0(A12), A12 57 P2.0(A13), A13 61 P2.0(A14), A14 34 P2.0(A15) 75 A0 79 A1 82 A2 84 A3 1 A4 3 A5 62 OEB 21 A7 65 CEB 91n that drives the RAM output enable. 65 CEB 91n that drives the RAM chip enable. 12 FREE0 12 FREE1 25 FREE3 76 FREE3 77 FREE3 78 FREE4	58	P2.0(A9), A9	Upper address byte. These pice also outputs the	
53 P2.0(A11), A11 53 P2.0(A12), A12 57 P2.0(A13), A13 61 P2.0(A13), A13 61 P2.0(A13), A14 34 P2.0(A15) 75 AQ 79 A1 82 A2 84 A3 1 A4 3 A5 83 A6 2 A7 62 OEB 91 that drives the RAM output enable. 65 CEB 91 that drives the RAM chip enable. 12 FREE0 12 FREE1 25 FREE2 76 FREE3 77 FREE3 78 FREE4	54	P2.0(A10), A10	address bits of the PAM	
57 P2.0(A13), A12 61 P2.0(A13), A13 61 P2.0(A14), A14 34 P2.0(A15) 75 A0 79 A1 82 A2 84 A3 1 A4 3 A5 62 OEB Pin that drives the RAM output enable. 62 OEB Pin that drives the RAM chip enable. 4 PREE0 These pins are not connected to other devices and can be used as general purpose I/O. 74 PREE3 76 PREE4	53	P2.0(A11), A11		
61 P2.0(A12), A13 34 P2.0(A15) 75 A0 79 A1 84 A3 1 A4 33 A5 83 A6 2 A7 62 OEB Pin that drives the RAM output enable. 65 CEB Pin that drives the RAM chip enable. 12 FREE1 These pins are not connected to other devices and can be used as general purpose I/O. 74 FREE3 76 FREE4	57	P2 (VA12), A12	-la in a second se	
34 P2.0(A15) 75 A0 79 A1 82 A2 84 A3 1 A4 3 A5 83 A5 62 OEB Pin that drives the RAM output enable. 65 CEB 12 FREE0 12 FREE1 25 FREE3 78 FREE4	61	P2 0(A14) A14		
75 AQ These pins drive the 8 lower address bits of the RAM. 82 A2 84 A3 1 A4 3 A5 83 A6 2 A7 62 OEB 9in that drives the RAM output enable. 65 CEB 12 FREE0 12 FREE1 25 FREE2 76 FREE4	34	P2 0(A15)		
79 A1 82 A2 84 A3 1 A4 3 A5 63 A6 2 A7 62 OEB 91 that drives the RAM output enable. 65 CEB 4 FREE0 12 FREE1 25 FREE2 76 FREE3	75	AO	These size drive the O.L.	
82 A2 84 A3 1 A4 3 A5 83 A6 2 A7 62 OEB 9 Pin that drives the RAM output enable. 65 CEB 4 FREE0 12 FREE1 25 FREE2 74 FREE3 76 FREE4	79	A1	These pins drive the 8 lower address bits of the RAM.	
84 A3 1 A4 3 A5 83 A6 2 A7 62 OEB Pin that drives the RAM output enable. 65 CEB Pin that drives the RAM chip enable. 12 FREE0 These pins are not connected to other devices and can be used as general purpose I/O. 74 FREE3 76 FREE4	82	A2		
1 A4 3 A5 83 A6 2 A7 62 OEB Pin that drives the RAM output enable. 65 CEB Pin that drives the RAM chip enable. 4 FREE0 These pins are not connected to other devices and can be used as general purpose I/O. 74 FREE3 76 FREE4	84	A3		
3 A5 83 A6 2 A7 62 OEB 65 CEB 4 FREE0 12 FREE1 25 FREE2 76 FREE4	1	A4		
53 As 2 A7 62 OEB 65 CEB 4 FREE0 12 FREE1 25 FREE2 76 FREE4	3	A5		
2 A7 62 OEB 65 CEB 9 Pin that drives the RAM output enable. 4 FREE0 12 FREE1 25 FREE2 74 FREE3 76 FREE4	83	A6		
62 OEB Pin that drives the RAM output enable. 65 CEB Pin that drives the RAM chip enable. 4 FREE0 These pins are not connected to other devices and can be 12 FREE1 used as general purpose I/O. 74 FREE3 76 FREE4	2	'A7		
65 CEB Pin that drives the RAM chip enable. 4 FREE0 These pins are not connected to other devices and can be 12 FREE1 used as general purpose I/O. 74 FREE3 76 FREE4	62	OEB	Pin that drives the RAM output enable	
4 FREE0 These pins are not connected to other devices and can be 12 FREE1 used as general purpose I/O. 25 FREE2 used as general purpose I/O. 76 FREE4 FREE4	65	CEB	Pin that drives the RAM chin enable	
12 FREE1 Freese 25 FREE2 used as general purpose I/O. 74 FREE3 FREE4	4	FREEO	These pine are not connected to attend	
25 FREE2 Used as general purpose I/O. 74 FREE3 76 FREE4	12	FREE1	used as are not connected to other devices and can be	
74 FREE3 76 FREE4	25	FREE2	useu as general purpose I/O.	
75 FREE4	74	FREE3		
	76	FREE4		

XS95 CPLD Board Programmer's Model

C1999 XESS Corporation