University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

CS 150 J. Wawrzynek
Fall 1996 N. Weaver

Lab 3
Finite State Machine on Xilinx

1 Objective

Put your combination lock from Lab 2 onto a chip. To do this, you will

1. Use XMake to compile and route your design.
2. Use XChecker to do real-time hardware debugging.

1.1 Overview

For a design to be useful, it must make its way into hardware. To do this we will be using Xilinx Field Programmable Gate Arrays (FPGAs), chips that can be configured as many different circuits. Specifically, we will be using the Xilinx XC4000 Design Demonstration Board, which contains an XC4005 FPGA, switches, and light-emitting diodes.

These boards are very expensive and not easily replaced. **Be very careful with these. Make sure that they stay on the anti-static mat. Make sure also that the ground plug is plugged in. Before touching the Xilinx board, ground yourself: touch any metal you can find or the static mat.**

For now, **do not take the Xilinx boards or anything else from 204B Cory!**.

2 Prelab

1. **Make sure your Lab 2 design works.**
2. Make sure your schematic does not use any components from the `builtin` library. If you followed the instructions in Lab 2, this should not be a problem.
3. Make sure none of your symbols or components have the same name as components in the (`xc4000`) library, such as `DFF` or `CLB`. If you did name anything identically, rename both the schematic and the symbol, and change the references to the symbol in all schematics in which it appears.
4. **Read through this lab** and answer the questions on the checkoff sheet that are answerable at this time.

2.1 Xilinx Interface

To compile your design for the Xilinx, some interface components need to be added. We have done the buisy work for you. Look at the design at your leisure.

2.1.1 I/O Pads

I/O pads and buffers are special cells in the (`xc4000`) library, connect the Xilinx to the outside world of buttons, LEDs, and other chips:

`IPAD`	Input Pad	`OPAD`	Output Pad
`IBUF`	Input Buffer	`OBUF`	Output Buffer

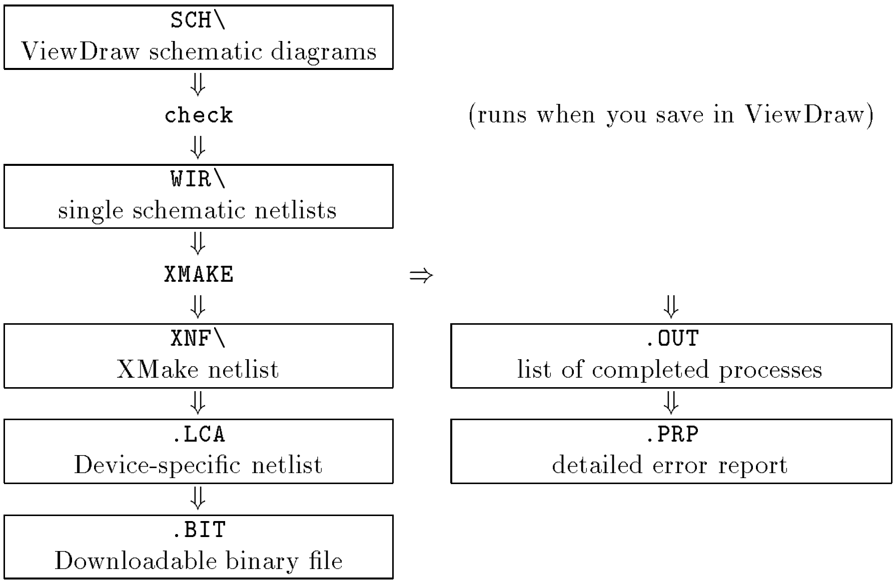

Figure 1: XMake's flow

Each `IPAD` and `OPAD` is connected to a particular pin on the Xilinx, designated by the `LOC` attribute. For example, in our schematic, the `IPAD` in the upper-left corner, `SW5-1`, has the attribute `LOC=P27`, indicating it connects to pin 27 of the Xilinx.

2.1.2 DEBOUNCE

Your lock expects `ENTER` and `RESET` to be high for exactly one clock cycle every time it is pressed. The `DEBOUNCE` circuit, a simple state machine, ensures this.

3 Add the Xilinx Interface Circuits

We have entered the required interface circuits; you need to copy them into your design. We'll use the ability of ViewDraw schematics to be divided into sheets.

A single schematic, i.e., one that can be enclosed in a symbol, can be divided into multiple sheets. Until now, you have only used one-sheet schematics (hence the `.1` extension on their names). To add the interface circuitry, we will put it on a second sheet.

Like on one-sheet schematics, nets with the same label on different sheets of the same schematic are connected implicitly.

1. Start ViewDraw. Open the Lab 3 schematic in the (cs150) library.

2. Choose File⇒Save Copy. Enter FSM.2[1].

3. Open FSM.2 in Viewdraw. Do File⇒Save. Examine the design demonstration board schematic in your reader and note how the XC4005[A] is connected to the LEDs, DIP switches, etc.

[1] or substitute the name of your Lab 2 design for FSM. I will assume that you named the top level of your Lab 2 combination lock "FSM."

`load` *design*	Download *design*`.BIT` to the FPGA.
`clock -int`	Use the XChecker clock.
`clock -ext`	Use an external clock.
`clock -stop`	Stop the XChecker clock.
`clock -resume`	Enable the XChecker clock.
`clock -speed` n	Set the speed of the XChecker clock. The slowest is 1 ($\approx$ 920kHz).
`clock` n	Step n clock cycles.
`help`	List XChecker commands.
`quit`	Terminate XChecker.

Table 1: Some XChecker commands.

4 Compile Your Design with XMake

XMake compiles your schematic into a form that can be downloaded into a Xilinx FPGA. This process can take up to fifteen minutes for this lab. The flow is shown in Figure 1.

1. **Start up the special DOS prompt for working with the Xilinx tools.** You need to run the Xilinx tools from a special dos prompt, in the Xilinx program group. It is called NTXILINX.

2. Run `ViewVsm` manually on your design[2]. This should report `0 errors and 0 warnings`.

3. Run XMake with `XMAKE -P 4005PC84-6 FSM`. The syntax is `XMAKE -P` *part schematicName*. The part number can be seen on the xilinx chip in the demoboard. If you ever have to deal with a different type of xilinx chip, you must rerun xmake.

If errors occur during XMake, check the `.OUT` and the `.PRP`[3] files. At least fifteen other files are also created, of varying utility. The `.RPT` file is interesting if the XMake is successful.

5 Debug Your Design with XChecker

XChecker is Xilinx's real-time hardware debugging tool, something like `gdb` for debugging hardware. The XChecker program communicates with the XC4003[A] through the XChecker cable. Table 1 lists some useful XChecker commands.

5.1 Power the Design Demonstration Board

The power supplies have a cable with two banana plugs, red and black, and a yellow lead. The black is ground, the red is +5V, and the yellow is −5V. **Don't attach the −5V connector to anything, and make sure it doesn't touch anything metal!**

1. Plug in the power supply.

2. Ground yourself.

3. Connect the black banana plug to the board.

4. Connect the red banana plug. The decimal point on `CR3` should light, indicating the power is on.

[2] This is not really a prerequisite for running XMake, but it's a cheap form of insurance. If the design doesn't vsm, it won't XMake.

[3] We are having some minor difficulties getting Xmake to create the .prp file. We hope to have this worked out before the project begins

5.2 Download and Test the Circuit

1. **Make sure Windows is not running.**
2. Run XChecker at the NTXILINX prompt:
   ```
   U:
   CD \username \lab2
   XCHECKER
   ```
   If this complains about the cable not being connected correctly, make sure the board has powered.
3. Make sure `SW 4-7` (marked `RST` on the board) is **closed.** Otherwise, the `RESET` button won't work.
4. Type `load FSM`, and press `Enter` to confirm the load. XChecker will download `FSM.BIT` into the XC4005. A `0` should appear on `CR4`, indicating the machine is in State 0.
5. Test a successful combination using the buttons and switches on the design demonstration board (see Figure ?? for connections).
6. Test an unsuccessful combination.

Have your TA check off your working lock.

5.3 Observe the Circuit in Operation

Try stopping and restarting the clock:

1. Stop the clock with `clock -stop`. The decimal point on `CR3` goes off and the lock stops working.
2. Restart the clock with `clock -resume`. The decimal point on `CR3` illuminates and the lock works.

Name: ______________________ Name: ______________________
Lab Section (Check one)
M: ☐11 ☐12 T: ☐13 W: ☐14 ☐15 Th: ☐16

6 Checkoffs

1. What is `ENTER` connected to (include direct and indirect sources)? Why?
2. What do you expect to see to see when the lock is opened?
3. To where does the clock feed out? Why?
4. Why are `CODE0` and `CODE1` not debounced?
5. In terms of buttons, switch settings, and lights illuminating, give instructions for opening the lock.
6. Explain the three wire-wrap wires on the board. Why are they there? Why does `SW 4-7` have to be closed? (Hint: look at a board and the demonstration board schematics)
7. By looking at the report files, how many CLBs does your design use?

1. Working Xilinx Lock TA: ______________(50%)
2. Questions answered TA: ______________(40%)
3. Turned in on time: TA: ______________(10%)