# CS/ECE 3700 Introduction to the Lab Kits

ATTENTION: Please do not ever use the UV (+5V) pin or any other power source above 3.3V in this lab kit, you can burn the FPGA instantly !

### Introduction

You will receive a lab kit (a "lunchbox" kit filled with circuit boards, wires, integrated circuits, etc.) in your first lab session. Most of the labs for the remainder of the semester will require you to use the lab kit to construct simple digital circuits, either with individual integrated circuits or on the FPGA. The purpose of this handout is to give you some familiarity with what's provided in the kit and to give you some advice about wiring up circuits. The material that follows is intended to introduce you to your lab kit. There are a number of precautions that you should take to avoid damaging the kit, so *be sure you read this document thoroughly!* 

### Wiring

Typically, IC's are mounted on boards or cards that hold the circuits in place, and provide a means of interconnecting the chips with signal lines. In large systems, many boards may be required and a chassis is used to hold the boards in place and to provide connections between them. The board provides a rigid base on which integrated circuits and components are mounted, and it provides a means of interconnecting chips. Two common strategies for interconnecting chips on a single board are printed circuit boards (PCBs) and, especially for prototyping, wire wrap boards. We will concentrate on wire wrap boards since this is what is provided in the lab kit.

A wire wrap board is, at its simplest, a board covered with a grid of holes. Sockets are always used in wire wrap systems (see Figure 1). The pins of the chip fit into the holes on top of the socket, and make electrical contact with the square pin-like posts. The base of the socket sits on the board with the pins going through the holes and protruding out the other side. Thin wires wrap around the posts and provide connections between IC's. The height of the post determines the number of wires that can be wrapped around it (typically, sockets provide enough room for two or three wires). A tool called a wire wrap tool is used to wrap the wire tightly around the post 5 or 10 times, providing up to 40 contact points to the square post. It's important that the wire-wrap posts are square. This is so the corners of the post can "dig in" to the wires to make a more secure connection that doesn't just fall off as might happen with a round post.



Figure 1: A wire-wrap socket

The side of the wire wrap board with the body of the socket is called the component side, and the other is called the wiring side. The wire wrapped around the posts is sufficient to keep the sockets from falling out of the board so no other attachment mechanism is needed. You will use a 5 volt power supply for your experiments this semester, so you will have to manually wire the 5v power pin and 0v ground pin to each socket on the component side. The location of the power and ground connections to the socket are found on the chip's data sheet.

The perfect wire wrap connection is one where the first wrap or first two wraps on the post (that is, the wrap closest to the board) is insulated wire, and the rest of the wrap is bare wire making contact with the post. This holds the socket on the board in a way that doesn't make electrical contact between the board and the post.

Wire wrap is often used to prototype systems before manufacturing a pc board. Wire wrap boards are not as easy to mass-produce as pc boards however (this is, perhaps, a bit of an understatement...), so they are not used in high volume production. The main drawback of wire wrap boards (besides the extremely high production costs in dollars, person-hours and in time) is that they are much thicker than pc boards because of the posts protruding from the board. Thus, fewer boards can be placed into a cabinet of some given size. There are also issues of how fast the signals can make transitions on a wire-wrapped prototype vs. a PCB or FPGA prototype. But, for small systems, it's a good solution for prototyping, and makes more secure wiring connections than the white proto-boards that you also see used for this purpose.

Large systems, especially prototypes, may consist of several boards, so some framework is necessary to mount them. It is important that the circuit boards are rigidly mounted into the card cage so that they cannot move around and accidentally short some part of the circuit (the box is often grounded). Some framework is also necessary to allow interboard connections. The framework for performing this function is usually referred to as the backplane. A typical enclosure consists of a box, a built in power supply, perhaps some switches and lights, and a card cage into which the boards are mounted. A fan is also sometimes required if the circuits generate an excessive amount of heat. There should always be proper ventilation since even low power devices can build up a fair amount of heat if completely enclosed. If a fan is provided, it should circulate air over the temperature sensitive logic circuits before the less sensitive power supply. Finally, a metal shield is often placed between the power supply and the logic circuits to shield the latter from electrical noise generated by the supply. The inside of a desktop PC tower with PCI and memory slots, power a power supply, and cages for disks and optical drives is a good example of this type of enclosure.

As for connections between boards, it is **NOT** a good idea to solder or wire wrap wires from one board to another. This type of approach makes it very difficult to service the system, since boards cannot be removed or replaced without bringing out a soldering iron or a wire wrap tool. Instead, the preferred approach is to use connectors that are easily inserted or removed. The plug/wall socket on home appliances is a good example of this type of connector.

# The CS/ECE 3700 lab kit

Each of you will receive a lab kit that you will use to construct your circuits. The lab kit is a lunchbox kit (actually a fabric-wrapped CD case) that contains:

- 1. A Digilent Nexys 3 FPGA board (with a Xilinx Spartan 6 FPGA)
- 2. A VmodWW wire wrapping extender board (prototyping area)
- 3. A USB cable to power and program the FPGA board
- 4. A number of black wire wrap sockets for mounting chips.
- 5. Several packages of wire wrap wire.
- 6. Several jumper cables.
- 7. A manual wrap-unwrap tool.
- 8. A number of integrated circuits mounted on two "bug rugs"
- 9. Miscellaneous items: toggle switches, push button switches, LEDs, resistor IC, etc

# **Integrated Circuits and Sockets**

Let's start at the smallest components and work our way up to bigger things. First, numerous chips (the black ICs with short legs) and wire wrap sockets (the black plastic carriers with long legs that the chips fit into) are provided. Note that many chips have 14 pins, but no 14-pin sockets are provided. You should use 16-pin sockets for these chips.

Some precautions must be taken in handling the chips. CMOS chips are particularly sensitive to static electricity, and can easily be destroyed if care is not taken. When you walk across a rug, your body acts like a capacitor and collects charge. The "spark" which occurs when you touch something metal is this charge being transferred to that object. If an IC is in this current path, its circuitry may be permanently damaged. Thus, you should never work on your kit in a room with carpets. Further, you should be careful in a chair

with a cloth seat, as you can pick up charge as you shift positions. Semiconductor manufacturers often have their employees wear conductive bracelets around their wrists which are wired to ground to avoid picking up charge; however this should not be necessary here. Finally, you should do your wire wrapping before you insert chips into your board.

The pads or "bug rugs" on which the IC's are supplied in your kits conduct electricity. The idea is to short the pins together so no pin is at a potential much higher than any other pin, protecting the chip from damage. To preserve this protection when you remove the chip from the pad, you should first touch a grounded object to discharge any charge you might be carrying, pick up the chip, and place it in the palm of your hand so that all of the pins contact your palm, and are (more or less) shorted together.

The function and pin assignments of the integrated circuits in your lab kit are documented in separate data sheets for each IC. These data sheets are linked to the class web page.

# Switches

Some components are provided in your kit are not documented in the data sheets linked to the class web page and thus require further explanation. First, examine the "pushbutton switches" each of which is mounted on a 16-pin wire wrap socket. The pinout for this switch (as mounted in the 16-pin socket) is shown in Figure 2a. Note that the "top" (pins 1 and 16) corresponds to the exposed portion of the wire wrap socket. The switch is shown in its normal position. Similarly, the pinout for the red four-pack of toggle switches is shown in Figure 2b. The arrow on each switch indicates the direction of the switch lever.

Throughout the semester you will use mechanical switches to provide input information to your hardware. Switches either make or break an electrical connection, and can therefore be used to connect a wire to ground (logic '0') or power, i.e. +5 volts (logic '1'). A typical circuit for using a switch to generate a '0' or '1' signal is shown in Figure 3a. When the switch is closed, a '0' is generated, and when it is open, a '1' is created. The resistor limits the current that flows through the switch when the switch is closed. It also pulls the X signal high when the switch is open. Without the pull-up resistor the X signal would be floating in that case. Although there are other ways of achieving the same effect without using a resistor, the circuit in Figure 3a is a good, common solution.

The switch in Figure 3a may be used to implement a Boolean variable that the user can set to 0 or 1 at will. It is often desirable to be able to generate such a variable as well as its complement. The circuit in Figure 3b achieves this effect, i.e. when one signal line is '0', the other is '1' and vice versa. Note that separate pull-up (current-limiting) resistors are required for each of the X and Xbar signals. You cannot use the same resistor for both wires.



Figure 2: Two types of switches - (a) single pushbutton, (b) four toggle switch



Figure 3: Circuits that use the red toggle switches to generate (a) A Boolean variables and (b) A Boolean variable and its complement

# LEDs

Several LED (light emitting diode) packs are also provided in your kit. A diode is a semiconductor device that conducts current when a positive voltage V is applied biased as shown in Figure 4a. The top (input) terminal of the LED in Figure 4 is the anode, and the bottom (output) terminal is the cathode. No current flows if the diode is biased in the opposite direction. It is essentially a one-way valve for electricity: current can flow in one direction but not the other (current only flows from anode to cathode).

An LED is a diode that emits light when it is conducting current. Typically the more current that flows, the brighter the light (until it reaches its standard current limit, or burns up. In our LED packets each LED pack in your kit a four-bar LED array. Each "bar" actually has two LEDs in it. Wiring up a single LED with the current provided by the Nexys 3 board to the VmodWW extender board has sufficient brightness. If you wire up more than one LED then both will need more current to keep the same brightness, so they will be dim.

The pinout for the LED pack is shown in Figure 4b, as well as circuits to use them. The two circuits demonstrate how you can have the LED on to indicate a high or a low logic signal. Note that the LED packs are symmetrical so it is impossible to insert one upside down. Figure 4b shows how you can wire up an LED. The rectangle that has the diode pictures inside represents the 16-pin LED package in your lab kit. You have to add the wires shown in the figure to connect the pins of the LED package in the right way to get it to light up.

To be even more specific (there has been confusion in the past), to make the top LED in Figure 4b light up (this will light the entire top "bar" on the LED package) do the following:

- 1. Add a wire from the active high output pin of the circuit to pin 15 of the LED package.
- 2. Add a wire from pin 16 of the LED package to ground.

Or

- 1. Add a wire from the active low output pin of the circuit to pin 16 of the LED package
- 2. Add a wire from pin 15 of the LED package to 3.3 VCC

In the first case when the signal at the input to the LED is high it will light up, and when that signal is low the LED will be dark, and we have the opposite effect in the second case as shown in Figure 4b.



Figure 4: (a) A diode (b) LED pack from your kit with example driver circuits

### **Resistor Packs**

A number of 16 pin resistor packs are included on the bug rug. These are meant to be used as "pull up" resistors (i.e. one end tied to 5 volts), so one terminal of all of the resistors are tied together (pin 16). The pinout is shown in Figure 5. Each resistor is 3.3K ohms. If you connect pin 16 to vdd you then have 15 separate pull-up resistors to use in your circuit connecting to any of the other 15 pins on the package. These resistors are especially useful for the pull-up/current-limiting resistors for switches as shown in Figure 3.



Figure 5: Resistor pack pinout showing the common connection on pin 16

## Wire Wrapping

A wire wrap-unwrap tool is provided in your lab kit for wiring circuits. The tool has a small aluminum handle with metal posts on each end (See Figure 6). The tool has two uses, a wire wrapping side (the longer side) and a wire unwrapping side (the shorter side). Examine the wrap tool side and note that there are two holes in the center of the tool as you're looking end-on: a large one in the center (for the post), and a smaller one towards the perimeter (for the wire). Take a small piece of stripped wire (be sure to take a piece of wire wrap wire, and not one of the larger jumper cables), and place it into the smaller hole as far in as it will go. A small portion of the insulation should fit into the hole. Now bend the wire 90 degrees. Push the barrel of the tool onto one of the posts on a board or socket (it should fit into the large hole), hold the wire in place against the board near the tool's barrel, and start turning in the clockwise direction with a steady, even motion. The tool will turn, and wrap the wire around the post (see Figures 7-8). The portion of wire wrapped around the base of the post should have insulation on it. This helps to avoid any contact between the wire and the board.

If your wire is bunched up into a messy coil around the base of the post, then you probably held the wrap tool too tightly against the board. If there is space between the coils on the post so that you can see the post, then you applied too little pressure. Occasionally, the wire will break, leaving part of the conductor inside the side groove of the tool. This will happen if you apply too much pressure while wrapping. The broken piece of the wire will be in the slot on the side of the too. To remove the wire, use your fingernail or another piece of wire.



Figure 6: A wire wrap and unwrap tool



Figure 7: Close up view of wire-wrapped connections with some common errors. From the left, the first two wires (blue and red) are good. The second red wire has too few turns. The third red wire has an overlap. The fourth red wire has open wraps, and the fifth red wire has a "pigtail." (<u>http://ecee.colorado.edu/~mathys/ecen1400/labs/lab08</u>



Figure 8: Close up view of some good wire wrapped connections. You can see that multiple wires (up to 3 if you're careful) can be wrapped onto a single post. (http://ecee.colorado.edu/~mathys/ecen1400/labs/lab08/)

Practice wire wrapping a few wires into the board. Now take a look at the unwrap part of the tool (the shorter side), a tip which is somewhat similar to the wire wrap side. The unwrap tool has a single hole which fits over a post. Push the tool into a post with a wire, and turn the tool counterclockwise. The wire wrapped around the post should come undone so that you can easily remove it.

If several points must be wired together, then a "daisy chain" of connections could be used, as shown in Figure 9a. Each additional wire is added to the end of the chain. However, suppose we find that we have made a mistake, and must now remove the first (leftmost) wire. Since we must unwrap the top wire to get to the one below it, and since unwrapped wires cannot be reused reliably, the entire chain will have to be removed to remove this one wire. A better way to wire a chain of points together is shown in Figure 9b. With this approach, we need to unwrap at most three wires to remove any wire in the chain. You should always use this second method to wire several points together.





Be careful when handling your wire wrap-unwrap tool. Even though it's small, it's surprisingly expensive. If the tips get bent they will not work reliably. If the tool is bent (perhaps by dropping it) it requires special care. See the DSL staff or your TA if any maintenance is required on your equipment.

There are also wire-wrap "guns" available in the DSL. These are like the wire-wrap tool, but have a trigger that spins the wrapping head. When you pull back on the trigger, the wrapping head spins the wire around the post. They're a little tricky to get the hang of, but can be quite a bit faster than the "pencil" tools that are in your kit once you figure them out. If you'd like to try one of the guns, you can check one out from the DSL staff. Note that these guns are even more delicate than the pencil tools so care is required (the DSL staff will show you how they work).

## Wire Wrap Boards

A VmodWW – VHDC wire warp extender board is provided in your kit. For the first part of the semester you will be using the prototyping area (see Figure 10) of the extender board to wire up your circuits. If you take a close look, you will find the power VCC (+3.3V) and ground (gnd) points. Notice that there is also a VU+5V point, which you shouldn't be using for this class whatsoever. If you turn the board towards the wiring side, there will be power, ground, and pin posts corresponding to the marked points. Use these posts to make your power, ground, and logic connections to your chip sockets.



Figure 10: VmodWW Extender Board from Digilent

# **Integrated Circuits**

Throughout this semester you will build your hardware using Boolean logic gates like NAND, NOR, INV, etc. These gates can be implemented as programmable functions on an FPGA (more about that later) or as stand-alone integrated circuits that implement standard logic functions. These stand-alone chips can be as simple as Inverters and NAND gates, or be slightly more complex like counters, decoders, memories, or be as complex as a full processor. It is very difficult to determine what function is performed by an integrated circuit just by looking at it (very difficult, but not impossible; many manufacturers pry open their competitor's IC's to see how they are implementing certain functions. This process is widespread, and is referred to as reverse engineering).

Manufacturers label IC's with a part number to identify it, and publish descriptions of the part in datasheets. We have links to the data sheets of each of the ICs in your lab kit on our class web page.

The part number usually specifies the manufacturer, the type of technology used, and the function performed. For example, in the part number MM74HC00, the MM indicates that the manufacturer is National Semiconductor, (each company has a different letter sequence, e.g. National uses MM, Motorola uses MC, Texas instruments uses SN, etc. I don't know why they chose those codes.). The "74" is the chip series or family of related chips. The "HC" refers to the circuit technology, in this case High-Speed CMOS. Most of the chips provided in your lab kit are 74xx series CMOS chips. Finally, the 00 indicates that it is a quad 2-input NAND gate package, i.e. the chip has four circuits, each one a two input NAND gate. For this course, this is the only number of any significance since it indicates the function provided by the chip.

How do we know that 74HC00 refers to a NAND gate? The same way you memorized your phone number. It's just a number that was assigned to this type of IC so you can tell it apart from others. The IC equivalent of a phone book that tells you what the 00 is vs. the 04 is a databook or a datasheet. Each manufacturer publishes a databook: a catalog indicating the logic function performed by each part the company manufactures. The databook also includes the pinouts for the part, i.e. which pins are the inputs to the first NAND gate, which is the output, etc., as well as timing and electrical specifications for the chip (how fast it is, what are the input/output voltage and current levels, etc.). Some of the packages remain constant across the 74 chip series. See Figure 11 for an example pinout diagram for an MM74HC00.



Figure 11: Pinout diagram for an MM74HC00 quad two-input NAND

A small notch is usually placed at the top of each integrated circuit to denote the top. Alternatively, a small mark may be placed in the upper left hand corner of the chip. When the chip is oriented this way, pin 1 refers to the pin in the upper left hand corner, pin 2 to the one below it, and so on going down the left edge of the chip, and then back up the right edge. A 14-pin chip has 7 pins on each edge. By convention, the pin on the lower left hand corner of the chip (pin 7 for a 14 pin chip) is usually connected to ground, and the pin in the upper right hand corner (pin 14 on 14-pin chip) is used for power. This is not always the case however, so you should always check the datasheets to be sure.

WARNING – when you connect wires using the wire wrap tool on the wiring side of the board, the pins are mirror-reversed from the view you're seeing from the component side! This is a really common source of wiring errors. Make SURE you think about where the pins are on the wiring side. From the top side (component side) pin 1 is in the upper left. When you flip the board over to put a wire on, pin 1 is now in the upper right!



Figure 12: Remember that things look different on the component and wiring sides of the board

In this class we'll start out using Boolean gates from the 74xx series of parts. The Xilinx ISE schematic tool does not have specific libraries for these parts. But, they have Boolean gates that look like the common 74xx series parts. So, if you need a NAND gate from a 74hc00 package, use the nand2 component in the ISE library.

## **FPGA Board**

The FPGA board is the Nexys3, from Digilent, bearing a Xilinx Spartan 6 FPGA is a complete, ready-to-use digital circuit development platform based on the Xilinx Spartan-6 LX16 FPGA. The Spartan-6 is optimized for high performance logic. The board offers a broad range of peripherals, but we will be using only the LEDs, switches, pushbuttons, the quad 7 segment display and the Vmod extender connections. The board is powered via the provided USB cable, and it is programmed using the Xilinx ISE tools (iMpact) via the same cable.

The VmodWW connector is used to power the wire wrapping board in your kit for using external IC chips, switches and LEDs. You can also use the 12 pin Pmod connectors for quick access to the FPGA pins, all providing dual 3.3 VCC GND pins, and 8 logic pins.



Figure 13: Spartan 6 FPGA board from Digilent, the Nexys 3.

# A Design Method

A haphazard design strategy usually leads to hardware that is difficult to modify and/or debug. Thus, we will now discuss a procedure you should follow in constructing your circuits.

- 1. Prepare a high level description of the major components of your design. Components at this level might include a finite state machine, a circuit for performing serial addition, etc. Get a clear picture in your mind of exactly what each module does, how they are interconnected, and what control and data signals must flow between them. In a finite state machine for example, the description would include what operations the machine performs, what inputs are required (e.g. to start the machine through a sequence of steps), and what outputs are generated. If the circuits are sufficiently simple, then this step may be omitted.
- 2. Prepare a detailed logic design of your circuits, using a databook to identify precisely what functions are to be performed by each chip. Use a schematic editor to draw the schematics for your design. Each block in the design should correspond to a component available in your lab kit. Make sure you have all of the components you need in your design. Once you have the detailed logic design entered into the schematic, use the ISE simulator to simulate the circuit. Simulating before constructing the circuit can track down many errors before it becomes necessary to re-wire circuits on the wire wrap board! Never even think of wiring until the simulation works!
- 3. Figure out a placement of your chips on the board.
- 4. Indicate on your logic diagram (schematic) the chip position of each gate and the pin number corresponding to each input/output (Pin numbers can be found on the data sheet for the part). The end result is that just by looking at the logic diagram, you should be able to locate the location of any signal on your board.
- 5. Make photocopies of your schematics. You can then use highlighters or other marks to show each wire that you've put on the board and keep track of what you've wired and
- 6. Insert the sockets (no chips yet!) into the board and wire up power and ground connections. Be careful if some 16 pin sockets contain 14 pin chips! You might want to wire some more dummy wires to these unused posts as a reminder that they are not to be used.
- 7. Check your power and ground wiring. Make sure that the power and ground pins of the socket connect to the power and ground pins on the prototyping board. Taking a few minutes to check that each chip gets power and ground will save you countless hours of headaches later on, as these sorts of mistakes are not

easy to track down. Further, connecting the ground pin of the chip to +5 volts and the power pin to ground will probably destroy the chip, so you'll want to make sure you get it right.

- 8. Use your schematic as a guide to wire wrap the rest of the board. I like to use a highlighter to mark off wires as I go so that I can see which wires have been added, and which are left to be added.
- 9. Before the chips have been inserted into the board, check your wiring from the component side of the board. Use a copy of your schematic to do this check, marking off lines as they are checked. This will detect errors made in generating the wiring list as well as wiring errors. Perform this test with an ohmmeter, or if you don't have one, you can use your logic probe and a long piece of wire with one end wrapped to ground. In the latter case connect your XST board to power. To test for a connection from point A to point B, first connect the probe to point A. It should not produce any signal (if it does, you have erroneously connected the line to power or ground). Now attach the grounded wire to point B. The logic probe should register a logic zero.

If you're using an ohm meter, note that it will not give useful readings after the chips have been inserted. Wiring errors are among the most frustrating, so you should always check your wiring before inserting any chips into the board.

A few minutes now will often save you hours of headaches and frustrations later on.

- 10. Insert the chips into the socket, and begin debugging the circuit.
- 11. If you didn't check your wiring and your circuit doesn't work because of a wiring error, don't say we didn't tell you so!
- 12. Make all changes on a copy of your schematic, and when the circuit works, update the originals.

### **Some Final Notes**

There are some miscellaneous facts you should be aware of about the lab kit. First, if you plug the XST in and the light on the power switch does NOT come on, then this probably means you have a short circuit from power to ground. Turn the power off immediately, and test the board to locate the short. If you turn power off and the power light remains on for a few seconds don't be alarmed. The circuitry on the boards stores up charge that keeps the light on until the charge is dissipated.

When you wire your board, note that the orientation of the chips is reversed from when you look at them from the component side. From the wiring side, pin number 1 is in the upper right hand corner. Also be sure you insert the chip with the notch at the top, or the dot over pin 1 on the upper left hand corner.

Chips inserted upside down will usually be permanently damaged.

In all labs you should pack your circuits onto the wire wrap board as tightly as is reasonable so that you don't waste space between chips. Don't spread your chips all over the board.

# **Parts and Wire**

For your labs, you should not use any parts beyond what is provided in the lab kit. You are probably doing something wrong if you run out of parts. You may not use parts from your partner's or anyone else's lab kit. You will be allowed to replace damaged parts however. If you have a part that you think has gone bad for any reason, turn it in to the DSL staff and they'll give you a new one. You must not swap wire wrap boards with anyone else. You will not receive any credit for work done on a wire wrap board that was not checked out to you.

The wire packets in your kit should be far more than you'll need for this semester. We'll be doing some wire wrapping, but also a lot of circuits on the FPGA. If you run out of a particular length, you can trade unopened packets of wire of one length for a different length – check with the DSL staff. You will probably find you will need much more wire of the shorter lengths than the longer lengths. You are also free to exchange wire with other students to obtain lengths you need. There is also a box of free wire in the DSL that you can use at your own risk. There is a small chance that the bare wire in the open packets will corrode and therefore not make good electrical contact. Most of this free wire will be fine though. I've used it for years and never had a problem!