Lab #2

Introduction

Objectives

1- Binary Logic Basics
2- Logic Gates Basics (Signal voltages for 0 &1, Logic Operations: AND, OR, NAND, NOR and NOT—T. Tables)
3- Familiarize with IC's (IC Pins: inputs, outputs, VCC & GND, applying inputs and monitoring outputs)
4- Partial familiarization with FPGA board, pins, switches, LEDs and power supply connections.
5- Verify Basic gates operation

Overview

The main purpose of this lab is to get familiar with binary logic and voltages the basic logic gates and their truth tables as well as gain some basic knowledge of integrated circuits (IC's).

To achieve this, the bread-boarding part of the FPGA board, switches, LEDs and power supply pins will be discussed and the used with integrated circuits to verify the truth tables of basic logic gates.

Design Description

In COE 200 lab, you are required to work on digital circuits. Digital circuits are hardware components that are implemented using transistors and interconnections in complex semiconductor devices called *integrated circuits*. Digital circuits work on the binary logic domain which uses two discrete values. These two values can be considered as **TRUE** or **FALSE**, High Voltage (+5V) or Low Voltage (0V). For our purpose, it is convenient to think in terms of binary values which are **1** or **0**.

Basic digital circuits are referred to as a logic gates. Using these logic gates, complex functions or larger digital circuits can be built. Examples of the basic logic gates are **AND, OR, NOT, NAND and NOR**. A complex gate such as an **XOR** gate can be built out of these basic gates. An example of the larger digital circuits is the digital clock.

The operation of each logic gate can be described using a truth table. A truth table is a tabular listing of all possible values that the inputs of a logic gate can take and the corresponding. Following are the truth tables of the AND gate, the OR gate and the Inverter (NOT gate):

Input 1 (X)	Input 2 (Y)	Output (Z)
0	0	0
0	1	0
1	0	0
1	1	1

Table 2. 1: Truth table of an AND gate

Input 1 (X)	Input 2 (Y)	Output (Z)
0	0	0
0	1	1
1	0	1
1	1	1

Table 2. 2: Truth table of an OR gate

Input (X)	Output (Z)
0	1
1	0

Table 2. 3: Truth table of an Inverter (NOT) gate

The graphical symbols used to designate the three types of gates (AND, OR, and NOT) is shown in Figure 2. 1.

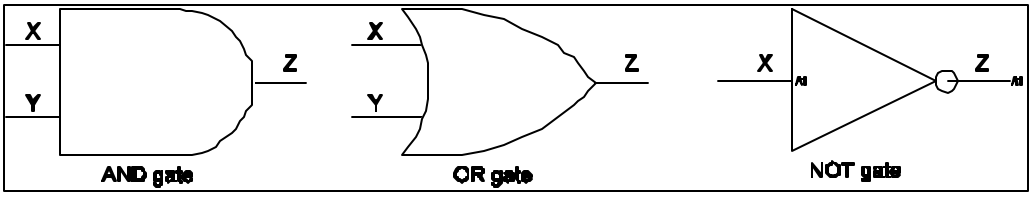

Figure 2.1: Basic logic gates graphics symbols

These basic gates are encapsulated in black boxes called integrated circuits (ICs). Each IC or chip has an ID number that you can check for by using the TTL data book. From the book, you can get the pin configuration of each chip. An example is the 7408 which contains four 2-input AND gate.

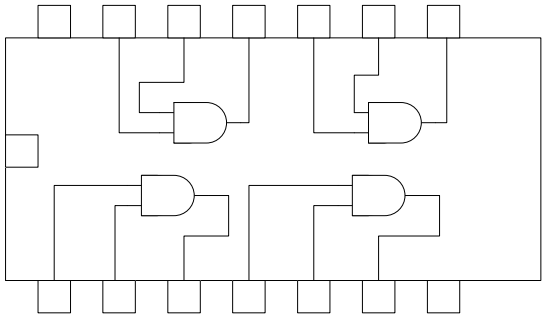

Figure 2.2: Quadruple 2-input AND gate

ICs must be provided with power and ground connections. In complex ICs more than one pin can be dedicated for power (VCC) and ground. In the simple gates we will be using in this experiment, the ICs require only one pin for power (VCC) and another for ground (GND). The power supply (VCC) voltage is typically +5olts, 3.3 volts or 2.5 volts, while to the ground is the reference node and is connected to 0V. For more information on other ICs, you may refer to the TTL data book or the appendix at the end of the book.

Digilab Board

The Digilab is a handy board to implement digital circuits on a Xilinx FPGA. The board contains a Spartan XCS10 (5V) FPGA. The board also provides 4 seven-segment displays, 8 LEDs, 8 general-purpose switches, four button switches, a serial port, a parallel port, PS/2 mouse and keyboard connector, VGA port, an 1/8" audio connector, a BNC connector, 2 on-board clocks, a ROM, a regulated power supply, a breadboard and several connectors for easy access to the FPGA pins. A simplified schematic of the board is shown in Figure 2.3 below.

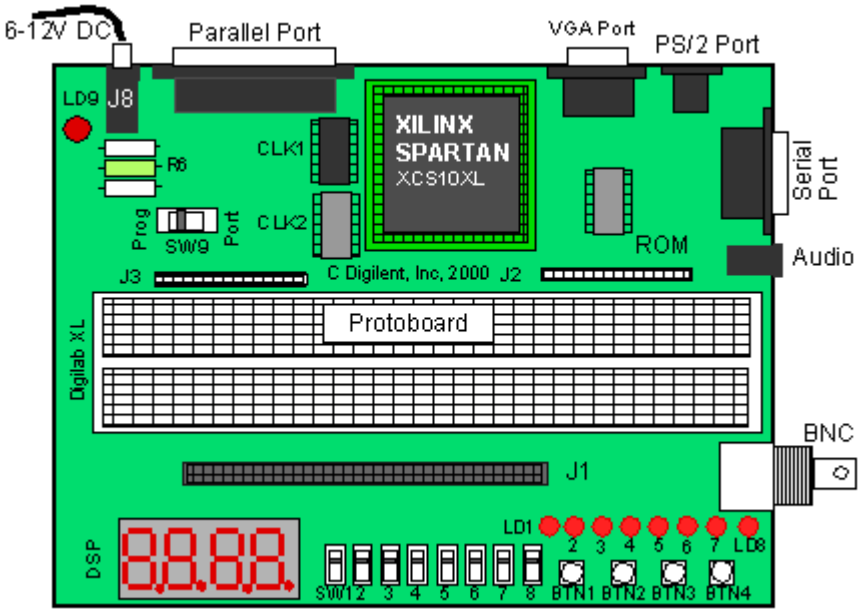

Figure 2.3: Digilab board

General Purpose Slide Switches SW1-SW8

These switches (SW1-SW8) can be used to supply a "1" (Vdd) or a "0" (GND) to the pins of the FPGA. The switches connect both to the FPGA pins and the pins of the J1 connector as indicated in the table-4 below. The switch will provide a "1" (Vdd) when the slider is in the down position (towards the edge of the board) and a "0" (GND) when in the up position.

Slide Switch #	Pin # on FPGA	Pin on J1 Connector
1	28	SW1
2	27	SW2
3	26	SW3
4	25	SW4
5	24	SW5
6	23	SW6
7	20	SW7
8	19	SW8

Table 2.1: Slide switches Connections

Push Buttons BTN1-BTN4

The Digilab board provides four push buttons which are normally connected to GND ("0") and which can be momentarily connected to Vdd ("1"). The buttons are connected to both to the FPGA pins and the pins of the J1 connector as indicated in the table-5.

Push Button #	Pin # on FPGA	Pin on J1 Connector
1	59	BTN1
2	58	BTN2
3	57	BTN3
4	56	BTN4

Table 2.2: Push Buttons Connections

LEDs

Eight LEDs are provided to display the value of output signals. The LED will light up when a "1" (Vdd) is applied to its input (anode) and will be off when a "0" (0V, low or GND) is applied. The signals can come either from the FPGA or from the pins on the J1 connector, as shown in the table below. One should be careful not to drive the LEDs from both the FPGA and the J1 connector.

	LED drive signal LD		Gate Signal LDG	
LED#	Pin # on FPGA	Pin J1	Pin # on FPGA	Pin J1
1	69	LD1	70	LDG
2	68	LD2		
3	67	LD3		
4	66	LD4		
5	65	LD5		
6	62	LD6		
7	61	LD7		
8	60	LD8		

Table 2.3: LED Connections

Seven-Segment Displays

The Digilab provides four seven-segment displays that can be used to display data from the FPGA or from other circuits through the J1 connector. The segments of each display are called A, B, up to G as is standard. In order to reduce the number of connections needed to address each of the LEDs, the anodes of the LEDs of each seven-segment display have been connected together. The common anode for the first seven-segment display is called A1; A2 for the second display, etc. In addition, the cathode pins from each display have been connected together to form seven common terminals, called A, B, C, D, E, F and G, corresponding to the seven-segments. Thus to illuminate a segment of a particular display, e.g. segment E of the third display, one has to apply a "1" (High or Vdd) to anode A3 and a "0" (GND or Low) to cathode node E. In addition to the seven segments LED, there is also a decimal point available (DP). However, the decimal point is not connected to the FPGA but to a pin of the J1 connector. If one needs to drive the decimal point, one can use a general-purpose output of the FPGA and connect this output to the DP pin on the J1 connector.

Seven-segment Terminal	Pin # of FPGA	Pin on J1 connector
A1 (Anode digit 1)	44	A1
A2 (Anode digit 2)	40	A2
A3 (Anode digit 3)	39	A3
A4 (Anode digit 3)	38	A4
A (Cathode of segment A)	51	CA
B	50	CB
C	49	CC
D	48	CD
E	47	CE
F	46	CF
G	45	CG
Dec. point	-	DP

Table 2.4: Seven segment displays' connections

Clocks

The Digilab board has two clocks, CLK1 and CLK2. CLK1 clock is the general system clock that has been connected to pin 13 (PGCK1) of the FPGA. It provides a clock of 25.175MHz. CLK2 is connected to pin 35 (PGCK2) of the FPGA as well as to pin CLK2 on the J1 connector. The second clock can come in handy for peripheral devices that need a clock.

We would be using CLK2 to provide manual pulses in many experiments. For doing so, we would be connecting CLK2 pin with some push-button like BTN1.

Clock	Pin # on FPGA	Pin on J1 connector
CLK1 (25.175 MHz	13 (PGCK1)	*No Connection*
CLK2	35 (PGCK2)	CLK2

Table 2.5: Clock Connections

Parallel Port (J7)

The parallel port can be used for two purposes: data communication and for programming the FPGA. The choice between the two modes are set by setting switch SW9 in the "PORT" (for parallel data communication) or in the "PROG" (programming) position. We would mostly be concerned with programming through the parallel port.

The parallel port can be used to program the FPGA. Make sure that switch SW9 is in the "PROG" position. The Xilinx Project Manager will automatically detect the parallel cable and enable programming over this port. However, for the first time it may be required to manually set the cable type in the Hardware Debugger Window (more on it in section 4.0).

Detailed Pin Description of the Digilab Board

Shaded boxes represent dedicated pins that are not available for use.

Pin#	**Function**	***Pin#***	**Function**	***Pin#***	**Function**
1	**GND**	*29*	O1	*57*	BTN3
2	VDD	*30*	M1_NC	*58*	BTN2
3	PWE	*31*	GND	*59*	BTN1
4	PD0	*32*	MODE	*60*	LD8
5	PD1	*33*	Vdd	*61*	LD7
6	PD2	*34*	M2_NC	*62*	LD6
7	PD3	*35*	CLK2	*63*	Vdd
8	PD4	*36*	O2	*64*	GND
9	PAS	*37*	O3	*65*	LD5
10	PRS	*38*	A4	*66*	LD4
11	Vdd	*39*	A3	*67*	LD3
12	GND	*40*	A2	*68*	LD2
13	CLK1	*41*	O4	*69*	LD1
14	PDS	*42*	Vdd	*70*	LDG
15	PWT	*43*	GND	*71*	O5
16	PD5	*44*	A1	*72*	RXD
17	PD7	*45*	CG	*73*	CCLK
18	PD6	*46*	CF	*74*	Vdd
19	SW8	*47*	CE	*75*	PINT
20	SW7	*48*	CD	*76*	GND
21	GND	*49*	CC	*77*	R
22	Vdd	*50*	CB	*78*	G
23	SW6	*51*	CA	*79*	B
24	SW5	*52*	GND	*80*	HS
25	SW4	*53*	DONE	*81*	VS
26	SW3	*54*	Vdd	*82*	PS2C
27	SW2	*55*	PROG	*83*	PS2D
28	SW1	*56*	BTN4	*84*	PINT

Pre-lab

1- Read the experiment
2- Derive the truth table of the NAND and NOR gates (NAND is an AND followed by an inverted)
3- (Bonus) Draw the graphic symbol of the NAND and NOR gate
4- Refer to chapter 2 in your text book.

In-lab

1- You are required to build a 4 inputs AND gate.
2- Use one TTL AND (7408) chip as in Figure 2.2.
3- Use the IC tester to test the 7408 chip.
4- Place the chip on the bread board part of the FPGA board.
5- Connect a wire from the VDD pin on the J1 connector to the bread board.
6- Connect pin 14 (Vcc) of the chip to the VDD on the board.
7- Connect a wire from the VGND pin on the J1 connector to the bread board.
8- Connect pin 7 (VGND) of the chip to the VGND on the board.
9- Use SW1, SW2, SW3, SW4 input pins on the J1 connector as your inputs to build a 4 input AND gate.
10- Connect the gate output to LD1 on the J1 connector of the digilab board.
11- Use the switches on the board to verify your gate.

Post-lab

1- Submit the truth table of the 4 input AND gate
2- Draw a circuit diagram of your implementation of the 4 input AND gate
3- (Bonus) How can you build a 4 input NAND gate using the above circuit?